

Primus PLM



Primus PLM ist ein Daten- und Dokumentenverwaltungssystem von [CADAdapt](#). Dieses Wiki dient dazu eine Dokumentation für das [Primus_Framework](#) zu erstellen. Es sollen neben einer [Befehlsreferenz](#) auch [Beispiele](#), sowie [Tipps_Tricks](#) für die [Konfiguration](#) bestimmter Aufgaben aus dem Bereich Datenverwaltung und Lebenszyklus Management. Es werden u.a. die Befehle zum Konfigurieren der [Benutzeroberfläche](#) beschrieben.

Folgende Seiten sind von besonderem Interesse:

- [Menübefehl](#)
- [Workflow Status](#)
- [Addins](#)
- [Abschnitte](#) in der [CFG.XML](#)
- [Primus.XML](#)
- [Tipps & Tricks](#)
- [Beispiele](#)

Das Konfigurationshandbuch als [PDF exportieren](#).

From:

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

Permanent link:

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

Last update: **2024/08/14 12:52**



Abschnitte

Die Primus Konfiguration gliedert sich in mehrere Abschnitte z.T. mit Unterabschnitten. Der prinzipielle XML-Aufbau eines Abschnitts ist im [Glossar](#) beschrieben. Alle Abschnitte müssen sich innerhalb des [Stammelements](#) befinden.

Folgende Abschnitte gibt es:

Name	Pflicht	Beschreibung
IncludeConfigLayers	Nein	Liste der beim Start dazu zu linkenden <i>.CFGLAYER.CFG</i>
SearchServiceDB	Ja	Verbindung zum zentralen PrimusSearchService
ReplicationService	Nein	Definiert den Dienst und den Standort für die Replikation
FileAccessManagementServer	Nein	Verbindung zum FileAccessManager
Folder_Paths	Ja	Definiert alle Daten- und Konfigurationsverzeichnisse
RelationsParentChild_by_ID	Nein	Definiert die Parent Child Zuordnung über unterschiedliche DataContext hinweg
SubFiles	Nein	Definiert zu anderen Dateien zugehörige Dateien (z.B. <i>.CFG</i> zu jeder Solid Edge Baugruppe)
DataContext	Nein	Definiert für bestimmte Folder_Paths abweichende GUI Konfigurationen
AddIns	Nein	Für Primus zur Verfügung stehende externe Erweiterungen
JobServer	Nein	Ablagepfad für Jobs, Jobserver Host und Timer
XML_DataTemplates	Nein	Definition für den Aufbau von Daten-XML
GUI	Ja	In diesem Abschnitt wird die Primus Oberfläche konfiguriert
Options	Ja	Globale Einstellungen für das Primus Framework
StatusWorkflow_Settings	Nein	Definition der möglichen Workflowstatus
Templates	Nein	Vorlagen für das neu Anlegen von Dateien durch Primus
Lists	Nein	Definition der Listen für Combobox und Listbox Elemente der GUI

From:

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

Permanent link:

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

Last update:

2024/03/22 09:37



Die Umleitung zur Seite [Addins](#) erfolgt in ungefähr 3 Sekunden.</doku.php?id=Addins;3000>

From:

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

Permanent link:

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

Last update: **2025/11/25 12:14**



AddIns

Im [Abschnitt AddIns](#) wird definiert welche Zusatzprogramme Primus zur Verfügung stehen. Die einzelnen AddIns werden als [Leer-Tag](#) definiert. Der Name des Tag ist der Name mit dem das AddIn innerhalb der Konfiguration, z.B. für [Menüeinträge](#) mit dem Attribut **addin**, angesprochen werden kann. Es werden zusätzlich noch [Attribute](#) angegeben.

Attribut	Wert	Funktion
dll	<DATEIPFAD>	Relativer oder absoluter Pfad zur DLL
class	<Klasse>	Programmklasse für den Aufruf
loadonstart		Soll das AddIn beim Primus Start mit geladen werden
	0	nicht beim Start laden
	1	beim Start von Primus laden

Beispiel

```
<!-- Definition des Primus SmartCopy AddIns-->
<PrimusSmartCopy dll=".\\Resources\\AddIns\\PrimusSmartCopy"
class="PrimusSmartCopy.AppInterface" loadonstart="1" />
```

Aufruf eines AddIn

Call

Ein AddIn kann innerhalb eines [Attributs](#) über **Call()** aufgerufen werden. Der Aufruf erfolgt über **ADDINNAME.KLASSE**, wobei **ADDINNAME** der im Abschnitt AddIns definiert Name ist, während **KLASSE** eine der Funktionen ist, welche das AddIn zur Verfügung stellt. Dazu wird die Dokumentation des AddIns benötigt. Um beispielsweise einen Wahrheitswert für **visible** zu erhalten. Eventuell nötige Aufrufparameter werden mit Leerzeichen getrennt übergeben.

Beispiel

```
<!--Blendet das Untermenü "GUI_SubMenu_Element" ein, wenn keine SmartCopy
Session aktiv ist-->
<INCLUDE menu="GUI_SubMenu_Element" name="Element"
visible="CALL(PrimusSmartCopy.IsSessionNotActive)" icon="" />
```

Menüeintrag

Die von einem Addin zur Verfügung gestellten Funktionen können auch direkt als Menüeintrag eingebunden werden. Dazu wird ein Leer-Tag mit dem Namen des Addin Befehls angelegt und im erforderlichen Attribut **addin**, der o.g. Name eingetragen.

Beispiel

```
<seCopyAllPropertiesToClipboard addin="SEInterface"
visible="'CALL(IsCtrlKey)'" enabled="TRUE" parameter="" text="Copy all
properties to clipboard" icon="" />
```

Liste der Standard AddIns von Primus

AddIn	Funktion
SEInterface	Interface zum CAD System Solid Edge
swinterface	Interface zum CAD System SolidWorks
msointerface	Interface für Microsoft Office
SEJob_Export	Arbeitet Jobs mit Solid Edge ab
PrimusSmartCopy	Stellt die SmartCopy Funktionen für Primus bereit
islprimusaddin	AddIn für die ISL-Normteile
sebom2xlsaddin	Übergibt Solid Edge Stücklisten an Excel

Siehe auch

- [unitec_addin](#)
- [DataConnect](#)

[klaerungsbedarf](#), [addin](#)

From:

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

Permanent link:

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

Last update: **2023/02/08 12:01**



AppAttributes

Im Verzeichnis **AppAttributes** liegen die Ini-Dateien mit denen der Abgleich der Datenfelder zwischen [Solid Edge](#) und Primus konfiguriert wird. Eine Liste der Felder und ihrer Schreibweise in Solid Edge, liegt im Verzeichnis und heit *Properties.txt*.

se_getprop_XXX.ini

Steuert den Abgleich der mit dem [Menbefehl seGetAllProperties](#) aufgerufen wird. Das XXX oben steht fr den jeweiligen Dateityp von Solid Edge.

Syntax

Die Ini-Datei besteht nur aus einer Section mit dem Namen **[PropertiesToPrimus]**. Jedes Zuweisungsprchen mu in einer eigenen Zeile stehen. Links steht das Primus [Datenfeld](#) und rechts durch ein = getrennt die Solid Edge Dateieigenschaft. Diese wird mit dem Schema **Eigenschaftsbereich . Eigenschaftsname** angegeben.

Bereich	Eigenschaft
SummaryInformation	Titel
	Thema
	Autor
	Kommentar
ProjectInformation	Revisionsnummer
	Dokumentnummer
	Projektname
FileProperties	CreatingDate
Custom	Benutzerdefinierte Eigenschaft

Beispiel

```
[PropertiesToPrimus]
Title=SummaryInformation.Titel
Revision=ProjectInformation.Revisionsnummer
Dimension=Custom.Dimension
```

se_getprop_XXX_ADDIN.ini

Steuert den Abgleich fr **Speichern unter** mit dem [Solid Edge Addin](#) damit werden bestimmte [predata](#) Felder von der bestehenden XML-Datei und nicht nur aus den Solid Edge Dateieigenschaften gelesen. Dadurch werden auch nicht nur die Felder in die neue Datei bertragen, die in dem [GUI_NewDocument_DataSheet](#) definiert sind, sondern auch die Felder die in der **_ADDIN.ini** zustzlich angegeben worden sind. Z.B. Note=# (Note). Die Syntax ist identisch zu

[se_getprop_XXX.ini](#)

se_prop_XXX.ini

Diese Ini Datei steuert den Abgleich der vom Solid Edge Addin beim Öffnen über Primus vorgenommen wird. Die Ini-Datei besteht aus mehreren Sections die für die einzelnen Eigenschaftsbereiche der Solid Edge Dateieigenschaften stehen, sowie dem Bereich [\[PropertiesToPrimus\]](#). Letzterer ist wie unter [se_getprop_XXX.ini](#) beschrieben aufgebaut und steuert welche Werte beim Speichern einer Solid Edge Datei in Primus zurück geschrieben werden sollen.

[PropertiesToPrimus]

In der `se_prop_par.ini`, `se_prop_psm.ini` und `se_prop_asm.ini` kann `[PropertiesToPrimus] Weight=Weight.kg(##0.000)` ausgelassen werden, wenn nicht bei jedem Speichern das Gewicht übertragen werden soll. Zusätzlich gibt es die Menü-Funktion [seUpdateWeight](#), die in dem Fall für alle selektierten Dateien das Gewicht aktualisiert.

Listeninhalte verwenden

Beim Übertragen der Werte von Primus an [Solid Edge](#), ist es möglich statt des eigentlichen Wertes des Datenfelds stattdessen einen zugeordneten Wert aus einer Liste zu übertragen. Dies ist beispielsweise sinnvoll bei dem Benutzernamen. Dieser ist in Primus der Windows-Loginname des Anwenders ohne vorgestellte Domäne. Für den Eintrag in Solid Edge kann es aber sinnvoll sein z.B. für die Historientabelle auf der Zeichnung eine abweichende Schreibweise zu verwenden. Die Definition welche Liste zu verwenden ist, wird dabei aus der [GUI_Document_List](#) der [CFG.XML](#) genommen. ~~Abweichende Werte für die Ersetzung müssen hier jedoch mit einem Pipe (|) und nicht mit einem Doppelpunkt abgetrennt werden, wie dies in der CFG.XML üblich ist.~~¹⁾ Dafür stehen folgende Funktionen zur Verfügung:

- [GetListText](#)
- [GetListTextFromDict](#)

Mittels Listeninhalten ist es auch möglich [Bedingte Werte von Primus an Solid Edge zu übertragen](#).

Beispiele

- `CheckedName=CALL(GetListText CheckedName)`
- `Autor=CALL(GetListText CreatedName)` - Weist dem *Autor* in Solid Edge den im Attribut text in der zugehörigen Liste hinterlegten Wert zu.
- `ChangeName#=#8#CALL(GetListText ChangeName)` - Weist dem *ChangeName* in Solid Edge den im Attribut histext in der zugehörigen Liste hinterlegten Wert zu.

Die Definition der Liste selbst, wird aus der [GUI_Document_List](#) der [CFG.XML](#) genommen, in diesem Fall beidesmal `List_User`.

```
<CreatedName data="CreatedName" type="Text" list="List_User"
```

```
position="19" width="104" visible="1" />
    <ChangeName data="ChangeName" type="List" list="List_User:hистext"
position="39" width="104" visible="0" />
```

Sonderfall `se_prop_dft.ini`

Für die Übergabe der Werte der Versionshistorie in die Benutzerdefinierten Eigenschaften (`[Custom]`) der Draftdatei, kann man steuern wie viele Revisionen in Draft angezeigt werden und ob die Startversion berücksichtigt werden soll oder nicht. Dazu dient das Nummernzeichen (`#`) als Platzhalter am Eigenschaftsnamen. `Rev#=#8#Revision` bedeutet das die 8 neuesten Revisionen in die Draftdatei eingetragen werden und das Feld mit `Rev` beginnt und eine der fortlaufenden Nummern direkt angehängt wird. Soll dabei mit der Startversion statt der ersten Revision begonnen werden, dann wird der Angabe der Anzahl der Versionen ein Punkt (`.`) vorangestellt. `Rev#=#.8#Revision`

Ein typischer Eintrag mit Kommentarzeilen (`#`) in der `se_prop_dft.ini` sieht dann so aus:

```
# Das #-Zeichen steht für die Nummer der jeweiligen Revision.
# Die 8 steht für die letzten 8 Dateien. Ein Punkt vor der 8 (.8) heißt eine
Version niedriger.
Rev#=#8#Revision
Change#=#8#Change
ChangeDate#=#8#ChangeDate
ChangeName#=#8#CALL(GetListText ChangeName)
```

Siehe auch

- [Listenmapping](#)

¹⁾

Dies ist eine Fehlinterpretation aus der Anfangszeit

From:

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

Permanent link:

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

Last update: **2025/02/12 12:33**



AssignDocumentDataFields

Im [Abschnitt AssignDocumentDataFields](#) unter [PreAssignData](#) wird definiert, welche Felder beim Auslesen der Dateieigenschaften der [Solid Edge](#) Dokumente in die XML von Primus übertragen werden sollen.

[Klärungsbedarf](#)

Liste der Primuswerte

Primus stellt folgende Werte in Variablen zur Verfügung. Alle Variablen sind in %-Zeichen eingeschlossen. Die Liste erhebt keinen Anspruch auf Vollständigkeit.

Variable	Funktion
%CREATEDTIME%	Das Datum des Dokuments im Dateisystem
%COUNTER%	Liefert eine fortlaufende Nummer aus dem Primus Nummerngenerator
%REVISION%	Den Weet der Revision aus den Solid Edge Dateieigenschaften
%TODAY%	Das heutige Datum
%USER%	Der angemeldete Benutzer

Beispiel

```
<AssignDocumentDataFields load="XML_Project">
  <Revision value="%REVISION%"/>
  <Number value="%COUNTER%"/>
  <WriteStatus value="Write"/>
  <WorkflowStatus value="001"/>
  <CreatedName value="%USER%"/>
  <CreateDate value="%CREATEDTIME%"/>
  <Change value=""/>
  <ChangeName value="%USER%"/>
  <ChangeDate value="%CREATEDTIME%"/>
</AssignDocumentDataFields>
```

From:

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

Permanent link:

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

Last update: **2023/06/20 14:24**



AssignDocumentDataFields_COPY

Im [Abschnitt AssignDocumentDataFields_COPY](#) werden die Datenfelder definiert die beim Kopieren einer Datei gefüllt werden sollen.

Attribute

Name	Wert	Funktion
load	XML_Project	Klaerungsbedarf

Einträge

Die Einträge der einzelnen Felder sind Leer-Tags, der Tag-Name ist der Name des zu füllenden XML-Datenfelds, das einzige Attribut ist **value**.

Beispiel

```
<Number value="%COUNTER%" />
```

Liste der Standardfelder

```
<AssignDocumentDataFields_COPY load="XML_Project">
  <Revision value="%REVISION%" />
  <Number value="%COUNTER%" />
  <WriteStatus value="Write" />
  <WorkflowStatus value="001" />
  <CreatedName value="%USER%" />
  <CreatedDate value="%TODAY%" />
  <Change value="" />
  <ChangeName value="%USER%" />
  <ChangeDate value="%TODAY%" />
</AssignDocumentDataFields_COPY />
```

Wertzuweisung

Konstante Werte

Dem kopierten Dokument können konstante Werte zugewiesen werden, wie z.B. WorkflowStatus value=„001“ oder leere Felder Change value=„“.

Primus Parameter

Darüber hinaus können von Primus bereitgestellte [Parameter](#) verwendet werden.

Externe Funktionen

Es können auch externe Funktionen mit [Call](#) aufgerufen werden, deren Rückgabewert dann in dem Feld gespeichert wird.

```
<TP_GUID value="CALL(UnitecNumGen.GetNewGuid B)" />
```

From:

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

Permanent link:

https://dokuwiki.unitec.de/doku.php?id=primus:assigndocumentdatafields_copy

Last update: **2023/05/08 11:31**



AssignDocumentDataFields_SMARTCOPY

Im [Abschnitt AssignDocumentDataFields_SMARTCOPY](#) werden die [Datenfelder](#) definiert, die beim Ausführen des Befehls SmartCopy vorgegeben und eventuell mit Vorgabedaten befüllt werden sollen. Der [Tag-Name AssignDocumentDataFields_SMARTCOPY](#) ist fest vorgegeben.

Attribute

Name	Wert	Funktion
load	XML_Project	klaerungsbedarf

Einträge

Die Einträge der einzelnen Felder sind Leer-Tags, der Tag-Name ist der Name des zu füllenden XML-Datenfelds, es können folgende Attribute zugewiesen werden.

Attribute der Einträge

Attribut ²⁾	Funktion
condition	Bedingung mit der geprüft wird ob die Zuweisung erfolgen soll
value	Zuweisung des Werts

Beispiel

```
<-- Die Wertzuweisung darf nur bei Solid Edge Dokumenten erfolgen,
da das SEInterface auf andere Dokumentarten nicht zugreifen kann -->
<WorkflowStatus value="CALL(SEInterface.seSetSolidEdgeFileStatusByPath
%FILEPATH%|0)|001" condition="'$(Condition.isSEDocument)'" />
```

[SmartCopy](#), [Wertzuweisung](#), [Hardcoded](#)

²⁾

Pflichtfelder in Fettschrift

From:

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

Permanent link:

https://dokuwiki.unitec.de/doku.php?id=primus:assigndocumentdatafields_smartcopy

Last update:

2024/09/05 08:06



AttachmentFiles

Im Eintrag **AttachmentsFiles** im Abschnitt **Public** wird der Ablagepfad und der Filter für die angehängten Neutralformate definiert. Die Werte werden in **Attributen** mit dem Namen **pattern** und einer angehängten Zahl übergeben, es sind mehrere Attribute möglich die aber fortlaufend nummeriert werden müssen. Die Verwendung von Windows **Wildcards** im Attributwert ist zulässig. Die angegebenen Pfad- und Dateinamensschemen werden auch von **SmartCopy Move** benutzt um die Anhänge ebenfalls zu verschieben.

Beispiel

```
<-- Nur Anhänge zur Dateirevision-->
<AttachmentFiles pattern1=".\\Attachments\\%FILENAME%*%EXTENSION%\\*.*" />
<-- Alle Attachments-Unterordner für alle Revisionen zu erfassen -->
<AttachmentFiles pattern1=".\\Attachments\\%FILENAME_NOREV%*%EXTENSION%\\*.*"
/>
```

Siehe auch

- [Attachments](#)
- [Liste aller Platzhalter](#)

[Public](#), [SmartCopy](#)

From:

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

Permanent link:

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

Last update: **2024/08/15 12:05**



Bedingte Werte von Primus an Solid Edge übertragen

Aufgabenstellung

Die Aufgabenstellung ist, wenn in der Primusdatenbank die Revision leer ist, dann soll in die Dateieigenschaften von [Solid Edge](#) ein Unterstrich () bei der Revision eingetragen werden. Dies wird benötigt, da die Zeichnungsnummer aus mehreren Felder zusammengesetzt werden soll, bei leerem Feld Revision würde aber eine Stelle in der Nummer fehlen.

Lösung

Da eine Bedingte Zuordnung über die [AppAttributes](#) nicht direkt möglich ist, muß ein anderer Weg gefunden werden.

Aufruf mittels CALL

Das ginge über einen [CALL](#) Aufruf, mit dem eine speziell dazu programmierte Funktion aufgerufen wird. Es geht aber

Ersetzung mittels Liste

Man erstellt dazu eine Liste mit dem Listeneintrag

```
<List_Revision>
  <entry value="" text="_"/>
</List_Revision>
```

In der entsprechenden `se_prop_XXX.ini` wird dann für die Revision dieser Eintrag hinzugefügt: `Revision=CALL(GetListText Revision)`. Dazu muß noch in der [GUI_Document_List](#) der Eintrag für das Revisionsfeld angepasst werden:

```
<Revision data="Revision" label="$Primus.COL_REVISION" type="List"
list="List_Revision"/>
```

Dadurch wird ein leerer Revisionseintrag in Primus in einen Unterstrich übersetzt, alle anderen Werte für die Revision bleiben unverändert da sie nicht in der Liste stehen.

Siehe auch

- [Tipps & Tricks](#)

Tipps&Tricks

From:

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

Permanent link:

https://dokuwiki.unitec.de/doku.php?id=primus:bedingte_werte_von_primus_an_solid_edge_uebertragen

Last update: **2023/12/19 15:14**



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	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	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	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>

Last update:

2026/01/20 07:12



Beispiele

In **Beispiele** werden typische Kundenanpassungen aufgelistet und beschrieben, wie diese konfiguriert werden können.

Liste der Beispiele

Allgemeine Anpassungen

- [Weiteres Datenfeld anlegen](#)

SmartCopy

- [Der Zeichnungsname bei SmartCopy soll der Name des Models sein](#)
- [Bestimmte Ordner per Default von SmartCopy ausnehmen](#)
- [Bei kopierten Solid Edge Dateien soll der Status zurück gesetzt werden](#)

Siehe auch

- [Tipps & Tricks](#)

[Liste](#), [Beispiel](#), [HowTo](#)

From:

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

Permanent link:

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

Last update: **2024/07/31 14:45**



button

Mit dem Wert **button** für das [Attribut ctrltype](#) kann man Schaltflächen in [DataSheets](#) definieren. Dem Button kann eine Funktion zugewiesen werden, die beim Klick darauf, aufgerufen wird. Bei den Funktionen kann es sich um Primus interne Funktionen z.B. [Menuebefehl](#) oder um [externe Programme](#) oder [AddIns](#) handeln.

Zugehörige Attribute

Um eine Schaltfläche in einem Formular zu definieren müssen neben dem Attribut **ctrltype** noch weitere Attribute ausgefüllt sein. Prinzipiell können alle Attribute aus DataSheets verwendet werden.

Attribut	Pflicht	Funktion
data	Ja	Definiert das aufzurufende Programm und übergibt ggf. noch Parameter (Trennzeichen) an dieses
type	Ja	Muss button sein
height	Nein	Darf nicht zu klein sein, sonst ist der Button nicht zu sehen und die Beschriftung wird nicht vollständig angezeigt
width		
labelposleft		Kann entfallen da der Labeltext immer auf der Schaltfläche liegt
visible		Gibt an ob der Befehl im Menü sichtbar ist
enabled		Wird nicht unterstützt

Beispiel

```
<ButtonGetNr ctrltype="button" data="UnitecAddIn.unitecGetControlText_Click  
Number|Title" label="ERP-Nummer holen" type="Button" mode="W" left="393"  
top="170" width="150" height="22" />
```

Aufruf von NamedCalls

Es ist möglich mit einem Button auch [NamedCalls](#) aufzurufen. Es ist jedoch zu beachten dass,

- [%FILENAME%](#)
- [%FILEPATH%](#)

sich immer auf die Auswahl in der [Haupt-Dokumentenliste](#) beziehen und nicht auf die Auswahl in der [Komponentenliste](#). Der Hintergrund ist, dass nicht feststellbar ist, ob der Button im [DataSheet](#) oder im [SubDataSheet](#) gedrückt wurde.

Beispiel

```
<ButtonClickDemo ctrltype="button" data="prmExecuteNamedCalls |DraftExport"  
label="ButtonClickDemo" type="Text" mode="W" left="382" top="713"
```

```
width="100" height="22" />
```

Wichtig ist hier das Leerzeichen hinter `prmExecuteNamedCalls`.

Rückgabewert

klaerungsbedarf

gui, datasheet, ctrltype

From:

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

Permanent link:

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

Last update: **2024/03/12 13:13**



callafterchange

callafterchange ist ein Attribut eines [Datenblatt](#) Feldes. Ändert sich der Feldinhalt dann wird eine im Attribut hinterlegte [Addinmethode](#) aufgerufen.

Beispiel

```
<StkVP ctrltype="textbox" data="StkVP" label="VP/Stk. €" type="Text"
mode="W" left="88" top="220" width="61" height="20" labelposleft="15"
multisel="1" callafterchange="CustomAddIn.progrDeckungsbeitragFeldUpdate" />
```

Beispiel für das Addin

```
public void progrDeckungsbeitragFeldUpdate(object oControl, DataGridViewRow
oRow, string oParams)
{
    if (oControl is TextBox)
    {
        TextBox oTB = (TextBox)oControl;
        if (oTB.Tag is PrimGUIDataField)
        {
            PrimGUIDataField oPrimGUIDataField = (PrimGUIDataField)oTB.Tag;

            string sFilePath = (string)oRow.Tag;
            if (!File.Exists(sFilePath)) return;

            IDictionary<string, string> oDataDict = PrimData.ReadDataXML(sFilePath
+ ".xml");
            if (oDataDict != null)
            {
                string sEPValue, sValue;
                if (oDataDict.TryGetValue("StkEPAuftr", out sEPValue) &&
sEPValue.Length > 0 && oDataDict.TryGetValue(oPrimGUIDataField.sField, out
sValue) && sValue.Length > 0)
                {
                    decimal dEPValue, dValue;
                    try
                    {
                        dEPValue = Convert.ToDecimal(sEPValue);
                        dValue = Convert.ToDecimal(sValue);
                    }
                    catch { return; }

                    bool bWriteData = true;
                    if (oPrimGUIDataField.sField == "StkVP")
```



```
        {
            decimal dDeckungsbEUR = dValue - dEPValue;
            oDataDict["DeckungsbEUR"] = dDeckungsbEUR.ToString("F2");
            oDataDict["DeckungsbProz"] = (dDeckungsbEUR / (dEPValue /
100)).ToString("F1");
        }
        else bWriteData = false;

        if (bWriteData)
        {
            string sRetMsg;
            PrimData.WriteDataXML(oPrimData.oActivePrimProject,
oPrimData.oActivePrimFolder.sCFGPath, sFilePath + ".xml", oDataDict,
"XML_Position", true, out sRetMsg);
            oPrimData.UpdateListViewRow(oRow.DataGridView, oRow, true);
        }
    }
}
}
```

Siehe auch

- [GUI_Document_DataSheet](#)

[GUI](#), [DataSheet](#), [Addin](#)

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

Permanent link:
<https://dokuwiki.unitec.de/doku.php?id=primus:callafterchange>

Last update: **2023/08/08 12:20**



CFG.XML

Zur Konfiguration werden in Primus **XML** -Dateien verwendet. In Primus wird, wie aber auch allgemein in XML, strikt zwischen Groß- und Kleinschreibung unterschieden. Die beiden folgenden Einträge sind nicht identisch:

```
<Number value="%COUNTER%" />
<number value="%COUNTER%" />
```

Darüber hinaus kann es für Primus problematisch sein wenn doppelte Leerzeichen, also zwei Leerzeichen hintereinander verwendet werden. Für XML ist dies prinzipiell zulässig, der **XML-Parser** von Primus ist jedoch auf Geschwindigkeit optimiert und erkennt deswegen solche Einträge in XML u.U. nicht richtig. Insbesondere in Attributwerten mit denen Parameter an Primusfunktionen übergeben werden, führt dies dann zu Fehlfunktionen. In **Notepad++** können diese jedoch über einen regulären Ausdruck gefunden werden. Dieser lautet

```
\S\K[[:blank:|]]{2,} 'Sucht nach zwei und mehr Leerzeichen hintereinander,
sofern diese nicht am Zeilenanfang sind
```

Kodierung

Die XML-Dateien müssen die Kodierung **UCS-2-LE-BOM** haben. Dateien die nicht diese Kodierung haben werden von Primus u.U. nicht als gültig anerkannt. Dies gilt insbesondere für die **Primus Lizenzdatei**.

XML-Header

Der Header der XML Dateien sieht folgendermaßen aus.

```
<?xml version="1.0" encoding="UTF-16"?>
```

Das führt allerdings beim Speichern mit **Notepad++** immer zu der Fehlermeldung **Document labeled UTF-16 but has UTF-8 content**.

Stammelement

Jede **.CFG.XML** muss ein und nur ein **Stammelement** enthalten. Dabei handelt es um ein **Tag**, das alle **Abschnitte** der Konfiguration enthält. Der Tag-Name lautet **PrimusFolderConfig**, und darf das Attribut **name** enthalten. Ist kein Name angegeben, dann wird in Primus am obersten Projektknoten auch kein Name angezeigt.

Beispiel

```
<PrimusFolderConfig name="Unitec">
```

XML Dateien verlinken

Hauptartikel [Verlinken von Primus Konfigurationen](#)

Es ist möglich die **CFG.XML** Datei in mehrere Dateien aufzuteilen und diese mit einem Kommando beim Starten von Primus zur Konfiguration dazu zu linkern. Dadurch ist es möglich auf einer Basiskonfiguration aufbauend z.B. unterschiedliche Menüs für CAD, Office oder Viewer Arbeitsplätze zu definieren. Allgemeine Änderungen, z.B. ein zusätzliches Datenfeld in der XML oder ein [Pflichtfeld](#), sind dann automatisch in allen angepassten Konfigurationen enthalten.

Kommentare

Kommentare sind in den Primus Konfigurationsdateien nur eingeschränkt zulässig. Innerhalb bestimmter [Abschnitte](#), sind sie nicht zulässig, zwischen einigen Abschnitten schon. Wo ein [Kommentar](#) erlaubt ist und wo nicht, wird in den einzelnen Abschnitten beschrieben. Kommentare werden am Beginn mit `<!--` gekennzeichnet und mit `-->` abgeschlossen. Ein Kommentar kann über mehrere Zeilen gehen. kommentierbare Bereiche werden mit dem Tag *Kommentierbar* gekennzeichnet.

```
{{tag>Kommentierbar}}
```

Platzhalter

Primus stellt zur Laufzeit Informationen in Form verschiedener [Platzhalter](#) zur Verfügung, die für die Konfiguration von Befehlen gebraucht werden können. Beispiele sind `%COUNTER%` der die aktuelle Nummer aus dem Zähler liefert oder `%FILEPATH%` der den Namen der aktuell zu verarbeitenden Datei inkl. Pfad und [Dateiendung](#) enthält. Unter [Liste aller Platzhalter](#) findet sich eine komplette Liste der Platzhalter.

Siehe auch

- [Glossar](#)
- [Liste aller Platzhalter](#)

[Konfigurations-XML](#)

From:

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

Permanent link:

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

Last update: **2023/04/25 13:15**



CheckActiveLicenseType

Mit der **Bedingung CheckActiveLicenseType** kann geprüft werden mit welcher Art von Lizenz Primus aktuell gestartet worden ist. Dadurch ist es möglich z.B. **Menübefehle** abhängig von der Lizenz unsichtbar zu machen um z.B. auf Viewer Clients Befehle zum Bearbeiten auszublenden. Die Bedingung kann mittels **Call** abgefragt werden. Dabei wird der gewünschte Lizenztyp als Parameter durch ein Leerzeichen getrennt übergeben. Es wird TRUE zurückgegeben wenn dieser Lizenztyp aktiv ist, FALSE wenn nicht. In Primus 7 lautete der Name der Bedingung **CheckActiveLicense**.

Mögliche Parameter sind:

Parameter	Bedeutung
CAD	Wenn Primus mit einer CAD Lizenz gestartet wurde
VIEWER	Wenn Primus mit einer VIEWER Lizenz gestartet wurde
<LICFILENAME>	Name der Lizenzdatei, z.B. JobServer (ab Primus 2021)

Beispiel

```
<prmFunktion visible="CALL(CheckActiveLicenseType CAD)" enabled="TRUE"
parameter="" text="Funktion CAD" icon="" />
<-- Oder als vordefinierte Bedingung unter Public -->
<Condition isCADLic="'CALL(CheckActiveLicenseType CAD)'" />
```

Anwendungsbeispiel

Es ist auch möglich die **DocumentOpenCall** und **DocumentRefListCall** auf Lizenz zu prüfen. Dadurch werden die **SE Dokumente** nicht mit der **Schnittstelle** geöffnet bzw. es werden keine **Referenzen/Komponenten** aktualisiert, wenn nur **VIEWER** oder **OFFICE** Lizenz geladen ist.

```
<Options>
  <Public>
    <DocumentOpenCall PAR_COND="CALL(CheckActiveLicenseType CLIENT)"
PSM_COND="CALL(CheckActiveLicenseType CAD)"
PWD_COND="CALL(CheckActiveLicenseType CAD)"
ASM_COND="CALL(CheckActiveLicenseType CAD)"
DFT_COND="CALL(CheckActiveLicenseType CAD)"/>
    <DocumentRefListCall PAR_COND="CALL(CheckActiveLicenseType CLIENT)"
PSM_COND="CALL(CheckActiveLicenseType CAD)"
PWD_COND="CALL(CheckActiveLicenseType CAD)"
ASM_COND="CALL(CheckActiveLicenseType CAD)"
DFT_COND="CALL(CheckActiveLicenseType CAD)" />
```

Prüfen der Aktivierung von Lizenzen

Die Primus Lizenzen werden über eine Internetverbindung regelmäßig erneuert. Dazu dient die URL <https://www.primusplm.net/Service/TestConnection>

Siehe auch

- [licenseName](#) als Attribut im Stammelement der Primus.xml

[Bedingung](#), [Beispiel](#), [Umstellung P7 auf P2021](#)

From:

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

Permanent link:

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

Last update: **2023/05/02 13:43**



CheckChildrenStatus

Die Funktion **CheckChildrenStatus** überprüft ob mit einem Dokument verlinkte Dokumente einen bestimmten **Status** haben und kann z.B. in einem **Workflow** als Bedingung verwendet werden. Der Rückgabewert der Funktion ist

Wert	Bedeutung
TRUE	Alle verknüpften Dokumente haben den angegebenen Status
FALSE	Mindestens eines der verknüpften Dokumente hat nicht den gesuchten Status

Attribute

Der Befehl wird über folgende Attribute gesteuert.

Attribut	Wert	Funktion
condition	TRUE/FALSE	Es werden nur Dokumente geprüft, die diese Bedingung erfüllen
parameter	1 %FILEPATH%	Pfad zum zu prüfenden Dokument
	2 <FELDNAME>	Name des zu prüfenden Datenfelds
	3 <WERT>	Mit diesem Wert wird der Inhalt von <FELDNAME> verglichen, Soll mit mehreren Werten möglichen Werten verglichen werden, dann sind diese durch Punkte voneinander zu trennen („.003.004.005.“)
	4 <BEDINGUNG>	Bedingung welche die Komponenten erfüllen müssen

Die Übergabe von **parameter** werden durch ein | voneinander getrennt.

Beispiel

```
<!-- Aufruf der Funktion durch einen Workflow PreCall, mögliche Statuswerte sind 003 oder 004
Der Pfadname der Komponenten darf kein "\normteile\" oder "\standardteile\" enthalten-->
<Call method="CheckChildrenStatus"
parameter="%FILEPATH%|WorkflowStatus|.003.004.|'%CHILD_FILEPATH:cn|\normteile\%=='AND'%CHILD_FILEPATH:cn|\standardteile\%==' "
condition="$ (Condition.isSEDocument)" />
```

Hinweistext

Wenn die Überprüfung fehlschlägt, wird ein Hinweistext ausgegeben. Dabei handelt es sich um einen **Elementtext** eines **Systemdefinierten Platzhalters** mit dem Namen MSG_SELCHILDSTATDIF.

Siehe auch

- [CheckChildrenWriteable](#)
- [CheckFileWriteable](#)
- [CheckParentsStatus](#)
- [CheckParentsWriteable](#)
- [IgnoreStatus](#)
- [SetChildrenToStatus](#)
- [SetPrevVersionToStatus](#)
- [SetSubFilesToStatus](#)

[Menu](#), [Befehlsliste](#), [Workflow](#), [Bedingung](#), [Sprachabhängig](#)

From:

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

Permanent link:

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

Last update: **2025/03/27 13:22**



CheckChildrenWriteable

Die **funktion CheckChildrenWriteable** überprüft ob die zu einem Dokument zugehörigen Dokumente, beschreibbar sind. Damit kann in den **PreCalls** geprüft werden ob eine Datei die während eines **Workflows** verändert werden soll, nicht anderweitig im Zugriff ist. Ein eventuell von Primus gesetzter Schreibschutz, z.B. durch Status 002 (Prüfung), wird dabei nicht berücksichtigt.

Rückgabewert

Der Rückgabewert der Funktion ist:

Wert	Bedeutung
TRUE	Alle verknüpfte Dokumente sind beschreibbar
FALSE	Mindestens eines der verknüpften Dokumente ist nicht beschreibbar

Attribute

Der Befehl wird über folgende **Attribute** gesteuert.

Attribut	Reihenfolge	Wert	Funktion
condition		TRUE/FALSE	Es werden nur Dokumente geprüft, die diese Bedingung erfüllen
parameter	1	%FILEPATH%	Pfad zum zu prüfenden Dokument
	2	<FELDNAME>	Name des zu prüfenden datenfelds
	3	<WERT>	Mit diesem Wert wird der Inhalt von <FELDNAME> verglichen, Soll mit mehreren Werten möglichen Werten verglichen werden, dann sind diese durch Punkte (.) voneinander zu trennen („.003.004.005.“)
	4	???	Klärungsbedarf
	5	SUBCOMP	

Die Übergabewerte von **parameter** werden durch ein | voneinander getrennt und müssen in der angegebenen Reihenfolge übergeben werden.

Beispiel

```
<Call method="CheckChildrenWriteable"
parameter="%FILEPATH%|WorkflowStatus|001|SUBCOMP"
condition="'$(Condition.isSEDocument)'" />
```

Siehe auch

- [CheckChildrenStatus](#)
- [CheckFileWriteable](#)
- [CheckParentsStatus](#)
- [CheckParentsWriteable](#)
- [IgnoreStatus](#)
- [SetChildrenToStatus](#)
- [SetPrevVersionToStatus](#)
- [SetSubFilesToStatus](#)

[Menu](#), [Befehlsliste](#), [Workflow](#), [Status](#), [Bedingung](#)

From:

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

Permanent link:

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

Last update: **2023/02/08 12:01**



checkedlistbox

Eine **checkedlistbox** ist ein [GUI Element](#) für [DataSheets](#), das dem Anwender eine [Auswahlliste](#) mit möglichen Feldwerten zur Verfügung stellt. Der Feldinhalt ist nur auf die zur Auswahl stehenden Listeneinträge begrenzt, es können jedoch im Gegensatz zur [listbox](#) mehrere Elemente ausgewählt werden. Soll der Anwender neben der Auswahl von Werten auch eigene Eingaben machen können, dann ist eine [Combobox](#) zu verwenden. Der Eintrag erfolgt im Attribut **ctrltype**.

Zusätzliche Attribute

Für die der Auswahl zugrunde liegende Liste müssen folgende Attribute angegeben werden:

Attribut	Wert	Pflicht	Funktion
type	List	Ja	Wird stattdessen der Wert Text zugewiesen, dann verhält sich das Feld wie eine textbox
list	<LISTENNAME>	Ja	Definiert welche unter Lists definierte Liste zur Auswahl gestellt werden soll

Beispiel

```
<UserLinks ctrltype="checkedlistbox" data="UserLinks" label="Benutzer"
type="List" list="List_JournalUser" mode="W" left="78" top="270" width="166"
height="120" labelposleft="20" multisel="1" />
```

Siehe auch

- [listbox](#)
- [combobox](#)
- [Hierarchische Listen](#)
- [Listenmapping](#)

[DataSheet](#), [GUI](#), [ctrltype](#), [Liste](#)

From:

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

Permanent link:

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

Last update: **2024/11/11 14:04**



CheckFileWriteable

Die [funktion](#) **CheckFileWriteable** überprüft ob ein Dokument, beschreibbar ist. Damit kann in den [PreCalls](#) geprüft werden ob eine Datei die während eines [Workflows](#) verändert werden soll, nicht anderweitig im Zugriff ist. Ein eventuell von Primus gesetzter Schreibschutz, z.B. durch Status 002 (Prüfung), wird dabei nicht berücksichtigt.

Rückgabewert

Der Rückgabewert der Funktion ist:

Wert	Bedeutung
TRUE	Das Dokument ist beschreibbar
FALSE	Das Dokument ist nicht beschreibbar

Attribute

Der Befehl wird über folgende [Attribute](#) gesteuert.

Attribut	Reihenfolge	Wert	Funktion
condition		TRUE/FALSE	Es werden nur Dokumente geprüft, die diese Bedingung erfüllen
parameter	1	%FILEPATH%	Pfad zum zu prüfenden Dokument

Beispiel

```
<Call method="CheckFileWriteable" parameter="%FILEPATH%"
condition="'$(Condition.isSEDocument)'" />
```

Siehe auch

- [CheckChildrenStatus](#)
- [CheckChildrenWriteable](#)
- [CheckParentsStatus](#)
- [CheckParentsWriteable](#)
- [IgnoreStatus](#)
- [SetChildrenToStatus](#)
- [SetPrevVersionToStatus](#)
- [SetSubFilesToStatus](#)

[Menu](#), [Befehlsliste](#), [Workflow](#), [Status](#), [Bedingung](#)

From:

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

Permanent link:

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

Last update: **2023/02/08 12:01**



checkinghidden

Bei **checkinghidden** handelt es um ein [Attribut](#) des [Tags Folder_Paths](#) der Primus [Konfigurationsdatei](#). Dieses Attribut legt fest ob die von Primus erstellten [XML](#)-Dateien das Dateisystemattribut Versteckt (Hidden) erhalten oder nicht. Versteckte Dateien sind mit den normalen Einstellungen des Windows-Dateiexplorer nicht sichtbar. Dadurch bleiben die Verzeichnisse, auch für Anwender die den Windows-Dateiexplorer weiter neben Primus benutzen, übersichtlich.

Werte

Das Attribut kann folgende Werte annehmen.

Wert	Funktion
0	Die XML-Dateien bleiben sichtbar
1	Die XML-Dateien werden versteckt

[Attribut](#)

From:

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

Permanent link:

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

Last update: **2023/08/02 07:55**



checkiswriteable

Bei **checkiswriteable** handelt es um ein [Attribut](#) des [Tags Folder_Paths](#) der Primus [Konfigurationsdatei](#). Dieses Attribut legt fest wie Primus den Dateischreibschutz setzen soll.

Werte

Das Attribut kann folgende Werte annehmen.

Wert	Funktion
0	Klärungsbedarf
1	
2	Der Dateischreibschutz wird nur über den WriteStatus gesteuert, die Reservierung wird nur angezeigt beeinflusst aber nicht den Dateischreibschutz.

Siehe auch

- [checkiswriteable_directory](#)
- [checkwriteaccess](#)

[Attribut](#), [Klärungsbedarf](#)

From:

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

Permanent link:

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

Last update: **2023/02/08 12:01**



CheckParentsStatus

Die **Funktion CheckParentStatus** überprüft ob die auf ein Dokument verlinkenden Dokumente, einen bestimmten **Status** haben. Dies kann Beispielsweise beim Ungültig machen von Dokumenten als **Bedingung** benutzt werden, um zu prüfen ob das Dokument noch irgendwo in einem gültigen Dokument verwendet wird. In dem Fall wird dem Anwender ein Hinweistext angezeigt. Die Funktion geht alle übergeordneten Dokumente (Parents), nacheinander durch, hat ein Dokument nicht den gewünschten Status, dann wird eine definierte Meldung ausgegeben und als **Rückgabewert** FALSE zurückgegeben. Für die Prüfung jedes Parents werden die betreffenden Werte in **speziellen Platzhaltern** abgelegt und können für Bedingungen und Meldungen benutzt werden.

Rückgabewert

Der Rückgabewert der Funktion ist:

Wert	Bedeutung
TRUE	Alle verknüpfenden Dokumente haben den angegebenen Status
FALSE	Mindestens eines der verknüpfenden Dokumente hat nicht den gesuchten Status

Attribute

Der Befehl wird über folgende **Attribute** gesteuert.

Attribut	Reihenfolge	Wert	Funktion
condition		TRUE/FALSE	Es werden nur Dokumente geprüft, die diese Bedingung erfüllen
parameter	1	%FILEPATH%	Pfad zum zu prüfenden Dokument
	2	<FELDNAME>	Name des zu prüfenden Datenfelds
	3	<WERT>	Mit diesem Wert wird der Inhalt von <FELDNAME> verglichen, Soll mit mehreren Werten möglichen Werten verglichen werden, dann sind diese durch Punkte (.) voneinander zu trennen („.003.004.005.“)
	4	<Bedingung>	Gibt eine Bedingung an mit der die zu prüfenden Parentdokumente eingegrenzt werden können
	5	<MELDUNGSTEXT>	Text der im Fehlerfall dem Anwender angezeigt werden soll. Die Angabe mittels Elementtext ist möglich

Die Übergabewerte von **parameter** werden durch ein | voneinander getrennt und müssen in der angegebenen Reihenfolge übergeben werden.

Spezielle Platzhalter

Diese Funktion stellt spezielle **Platzhalter** zur Verfügung.

Platzhalter	Bedeutung
Parent_Extension	Enthält die Dateiendung des übergeordneten Dokuments
Parent_Filename	Enthält den Dateinamen ohne Dateiendung ...
Parent_Filepath	Enthält den kompletten Pfad mit Dateinamen und Dateiendung ...

Beispiel

- Die übergeordneten Dokumente des Dokuments %FILEPATH% werden geprüft
- Der Inhalt des Felds WorkflowStatus
- Muß 005 sein
- Die Dateiendung des übergeordneten Dokuments darf nicht .dft sein
- Im Fehlerfall wird die Meldung aus \$Primus.MSG_ATLEAST_ONE_PARENT_IS_VALID ergänzt um %PARENT_FILEPATH% ausgegeben.

```
<Call method="CheckParentsStatus"
parameter="%FILEPATH%|WorkflowStatus|.005.|%PARENT_EXTENSION%!=.dft|$Primus.
MSG_ATLEAST_ONE_PARENT_IS_VALID %PARENT_FILEPATH%"
condition="$ (Condition.isSEDocument)" />
```

Siehe auch

- [CheckChildrenStatus](#)
- [IgnoreStatus](#)

[Menu](#), [Befehlsliste](#), [Workflow](#), [Status](#), [Bedingung](#)

From:

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

Permanent link:

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

Last update: **2024/09/05 08:07**



CheckParentsWriteable

Die [Funktion](#) **CheckParentsWriteable** überprüft ob die übergeordneten Dokumente eines Dokument, beschreibbar sind. Damit kann in den [PreCalls](#) geprüft werden ob eine Datei die während eines [Workflows](#) verändert werden soll, nicht anderweitig im Zugriff ist.

Rückgabewert

Der Rückgabewert der Funktion ist:

Wert	Bedeutung
TRUE	Alle übergeordneten Dokumente sind beschreibbar
FALSE	Mindestens eines der übergeordneten Dokumente ist nicht beschreibbar

Attribute

Der Befehl wird über folgende [Attribute](#) gesteuert.

Attribut	Reihenfolge	Wert	Funktion
condition		TRUE/FALSE	Es werden nur Dokumente geprüft, die diese Bedingung erfüllen
parameter	1	%FILEPATH%	Pfad zum zu prüfenden Dokument
	2	<FELDNAME>	Name des zu prüfenden Datenfelds
	3	<WERT>	Mit diesem Wert wird der Inhalt von <FELDNAME> verglichen, Soll mit mehreren Werten möglichen Werten verglichen werden, dann sind diese durch Punkte (.) voneinander zu trennen („.003.004.005.“)
	4	<Bedingung>	Gibt eine Bedingung an mit der die zu prüfenden Parentdokumente eingegrenzt werden können
	5	<MELDUNGSTEXT>	Text der im Fehlerfall dem Anwender angezeigt werden soll. Die Angabe mittels Elementtext ist möglich

Die Übergabewerte von **parameter** werden durch ein | voneinander getrennt und müssen in der angegebenen Reihenfolge übergeben werden.

Beispiel

```
<Call method="CheckParentsWriteable"
parameter="%FILEPATH%|WorkflowStatus|001|$(Condition.isSEDraft)|$Primus.MSG_
PARENTWRITEPROT %PARENT_FILEPATH%" condition=" '$(Condition.isSEDocument)' "
/>
```

Siehe auch

- [CheckChildrenWriteable](#)
- [CheckChildrenStatus](#)
- [CheckParentsStatus](#)
- [IgnoreStatus](#)
- [SetChildrenToStatus](#)
- [SetPrevVersionToStatus](#)
- [SetSubFilesToStatus](#)

[Menu](#), [Befehlsliste](#), [Workflow](#), [Status](#), [Bedingung](#)

From:

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

Permanent link:

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

Last update: **2024/09/05 08:07**



checkwriteaccess

Bei **checkwriteaccess** handelt es um ein [Attribut](#) des [Tags Folder_Paths](#) der Primus [Konfigurationsdatei](#). Dieses Attribut legt fest dass [Daten-XML](#) die Status Read haben automatisch auf Readonly gesetzt werden. In Verbindung mit [checkinghidden](#), können XML versteckt und gegen ungewollte Änderung gesichert werden. Mehr ist mit [PrimusFAMService](#) umsetzbar

Werte

Das Attribut kann folgende Werte annehmen.

Wert	Funktion
0	Deaktiviert
1	Aktiviert

Siehe auch

- [checkinghidden](#)
- [PrimusFAMService](#)

[Attribut](#), [CADAdaptApproved](#)

From:

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

Permanent link:

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

Last update: **2025/08/13 08:35**



CheckXMLValue

Bei **CheckXMLValue** handelt es sich um eine [Methode](#), die mit [call](#) aufgerufen werden kann. Damit können z.B. in einem [Workflow](#), die Inhalte von Datenfeldern auf einen bestimmten Wert geprüft werden. Es kann aber prinzipiell eine beliebige [Bedingung](#) geprüft werden.

Rückgabewert

Der Rückgabewert der Methode ist:

Wert	Bedeutung
TRUE	Das Dokument erfüllt die Bedingung
FALSE	Das Dokument erfüllt nicht die Bedingung

Aufrufparameter

Die Aufrufparameter werden mit dem [Attribut](#) parameter, durch | getrennt übergeben. Die Parameter sind in folgender Reihenfolge zu übergeben:

	Parameter	Funktion
1	<DOKUMENT>	Dokument zu dem z.B. ein Feldwert geprüft werden soll (%FILEPATH%)
2	<BEDINGUNG>	Bedingung die geprüft werden soll
3	<MELDUNG IM FEHLERFALL>	Meldung die bei nicht zutreffender Bedingung ausgegeben wird

Beispiel

Im Beispiel wird der Wert des Gewicht eines 3D Modells geprüft und wenn dieses leer ist, wird eine Meldung ausgegeben.

```
<Call method="CheckXMLValue" parameter="%FILEPATH%|'#(Weight)!='|No weight is calculated" condition="$ (Condition.isSEModel)" />
```

Siehe auch

- [Liste der Methoden des Primus Framework](#)

From:

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

Permanent link:

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

Last update: **2023/02/08 12:01**



combobox

Eine **combobox** ist ein [GUI Element](#) für [DataSheets](#), das dem Anwender eine [Auswahlliste](#) mit möglichen Feldwerten zur Verfügung stellt. Der Anwender kann neben der Auswahl von Werten auch eigene Eingaben machen. Soll der Feldinhalt nur auf die zur Auswahl stehenden Listeneinträge begrenzt werden, dann ist eine [listbox](#) zu verwenden. Der Eintrag erfolgt im Attribut **ctrltype**.

Zusätzliche Attribute

Für die der Auswahl zugrunde liegende Liste müssen folgende Attribute angegeben werden:

Attribut	Wert	Pflicht	Funktion
type	List	Ja	Wird stattdessen der Wert Text zugewiesen, dann verhält sich das Feld wie eine textbox
list	<LISTENNAME>	Ja	Definiert welche unter Lists definierte Liste zur Auswahl gestellt werden soll

Beispiel

```
<Benennung ctrltype="combobox" type="List" list="List_Titles" />
```

Siehe auch

- [listbox](#)

[GUI](#), [DataSheet](#), [ctrltype](#)

From:

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

Permanent link:

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

Last update: **2023/02/08 12:01**



Condition

Mit **Condition** wird im [Abschnitt Public](#) eine [Bedingung](#) definiert, die in der gesamten Konfiguration genutzt und über ihren Namen aufgerufen werden kann. Es handelt sich dabei um einen [Leer-Tag](#) mit dem Namen der Bedingung als einzigem [Attribut](#), die Bedingung wird diesem Attribut als Wert zugewiesen.

Beispiel

```
<!--Die Bedingung prüft ob das Dokument eine Solid Edge DFT ist,
in der Konfiguration kann sie über isSEDraft genutzt werden-->
<Condition isSEDraft="'#(FILE_TYPE)==DFT'" />
```

Oder die Definition einer Liste von Dateitypen ohne expliziten Vergleichsoperator

[klaerungsbedarf](#)

```
<Condition isCheckModifyFileTypes=".PAR.PSM.ASM.PWD.DFT.DOC.DOCX.XLS.XLSX."
/>
```

Verwendung

Bedingungen werden in verschiedenen Primusbefehlen verwendet um die Funktion nur unter bestimmten Voraussetzungen zu aktivieren, anzeigen oder nutzen. Dazu haben die jeweiligen Befehle [Attribute](#) denen ein Wahrheitswert (TRUE, FALSE oder 0, 1) zugewiesen werden kann.

Attribut	Funktion
enabled	Steuert ob ein GUI -Objekt aktiv ist oder nicht
visible	Steuert ob ein GUI-Objekt sichtbar ist oder nicht
condition	Steuert ob ein bestimmter Befehl ausgeführt werden kann

[options](#), [public](#)

From:

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

Permanent link:

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

Last update:

2023/02/08 12:01



ContainsListValue

Die Funktion **ContainsListValue** überprüft ob in einem String ein **Listeneintrag** enthalten ist und liefert als Rückgabewert TRUE oder FALSE. Die Funktion kann mit **Call** als **Bedingung** genutzt werden. Es ist dadurch möglich nach dem Vorhandensein eines Begriffs in einem Wert zu suchen, ohne die einzelnen Vergleiche mit AND oder OR zu verknüpfen, was schnell zu schwer lesbaren Bedingungen führt. Der Vorteil ist, dass eine Liste als separate XML-Datei zentral abgelegt und von einem Admin einfach geändert werden kann, ohne die komplette **Konfiguration** ändern zu müssen. Dadurch entfällt auch die Notwendigkeit, die Clients zu aktualisieren.

Parameter

Die Parameter müssen in der folgenden Reihenfolge übergeben werden:

Parameter	Funktion
1 String	Der String dessen Inhalt verglichen werden soll
2 Liste	Name der Liste, deren Einträge in String auf enthält geprüft werden

Beispiele

Bestimmte Verzeichnisse aus **SmartCopy** ausnehmen.

```
<SmartCopy componentfilter="'CALL(ContainsListValue  
%FILEPATH%|List_ExcludeFromSmartCopy)'" />
```

Prüfen ob der angemeldete Benutzer in der Liste der Administratoren ist, um eine bestimmte Menüfunktion, hier übernehmen der **Solid Edge** Materialtabelle nach Primus, sichtbar zu machen.

```
<seCopySEMaterialsToCFG addin="SEInterface" visible="'CALL(ContainsListValue  
%USER%|List_PrimusAdmins)'" />
```

From:

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

Permanent link:

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

Last update: **2023/08/10 11:45**



CopyFile

Bei **CopyFile** handelt es sich um eine **Methode**, die mit **call** aufgerufen werden kann. Damit können z.B. in einem **Workflow**, Dateien von einer Quelle zu einem Ziel kopiert werden, dabei ist es möglich das sich auch der Dateiname ändert.

Rückgabewert

Der Rückgabewert der Methode ist:

Wert	Bedeutung
TRUE	Der Kopiervorgang war erfolgreich
FALSE	Der Kopiervorgang war nicht erfolgreich

Aufrufparameter

Die Aufrufparameter werden mit dem **Attribut** parameter, durch | getrennt übergeben. Die Parameter sind in folgender Reihenfolge zu übergeben:

	Parameter	Funktion
1	<QUELLE>	Datei mit kompletter Pfadangabe, die kopiert werden soll
2	<ZIEL>	Ablagepfad inkl. Name der Zieldatei

Beispiel

Im Beispiel wird eine Exceltabelle kopiert, welche den gleichen Dateinamen wie die gerade verarbeitete Draftdatei hat.

```
<Call method="CheckXMLValue"
parameter="%DIRPATH%\%FILENAME%.xlsx|$(Exporte.path)\%FILENAME%.xlsx"
condition="$(Condition.isSEDraft)" />
```

Siehe auch

- [Liste der Methoden des Primus Framework](#)

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

Permanent link:
<https://dokuwiki.unitec.de/doku.php?id=primus:copyfile>

Last update: **2023/02/08 12:01**



CreateJob_StatusPostCalls

Mit der [Primus Methode CreateJob_StatusPostCalls](#) können Jobs für einen [JobServer](#) erzeugt werden. Dabei können mehrere Aufgaben in einem mit **CreateJob_StatusPostCalls** erstellten Job zusammengefasst werden. Dazu wird die Methode mit einem [Call Tag](#) aufgerufen, alle in diesem Tag zusammengefassten Calls werden dann als Aufgaben in die Jobdatei geschrieben. Als Parameter wird er Ablagepfad für die Jobdateien übergeben. Dazu wird der String `$(JobServer.path)` Im Abschnitt [Public](#) definiert. Die Methode ist nur in [PostCalls](#) eines [StatusWorkflows](#) verwendbar.

Verwenden von Bedingungen

Die Methode **CreateJob_StatusPostCalls** kann über das Attribut `condition` abhängig von einer [Bedingung](#) aufgerufen werden. Dabei ist zu beachten das die Bedingungen der einzelnen Aufgaben zur Laufzeit des Jobs nicht mehr ausgewertet werden können. Es sollten also nur Aufgaben zusammengefasst werden, auf die die Bedingung des Jobs zutreffen. Es können aber mehrere **CreateJob_StatusPostCalls** Definitionen innerhalb eines PostCalls angelegt werden.

Beispiel

Die im folgende Beispiel in den einzelnen Aufgaben angegebenen [Bedingungen](#) sind wirkungslos, da lediglich die Bedingung von **CreateJob_StatusPostCalls** ausgewertet wird. Sind hier verschiedene Bedingungen zu beachten, dann müssen jeweils eigenständige Jobs angelegt werden.

```
<Call method="CreateJob_StatusPostCalls" parameter="$(JobServer.path)"
condition="$(Condition.isSEDraft)" reverseonerror="ReverseDataValues">
  <Call method="SEJob_Export.seExportJobByPath"
parameter="%FILEPATH%|C300|SE-
&gt;PDF|. \Attachments\%FILENAME%%EXTENSION%\%FILENAME%|XML_DocumentAttachmen
ts" />
  <Call method="SEJob_Export.seExportJobByPath"
parameter="%FILEPATH%|1|DXF|. \Attachments\%FILENAME%%EXTENSION%\%FILENAME%|X
ML_DocumentAttachments" />
  <Call method="CopyFile"
parameter="%DIRPATH%\Attachments\%FILENAME%%EXTENSION%\%FILENAME%.pdf|$(ToER
P.PDFDir)%FILENAME%.pdf" />
  <Call method="CopyFile"
parameter="%DIRPATH%\Attachments\%FILENAME%%EXTENSION%\%FILENAME%.dxf|$(ToER
P.DXFDDir)%FILENAME%.dxf" />
  <Call method="SetXMLValue" parameter="%FILEPATH%|WorkflowStatus|003" />
</Call>
```

Siehe auch

- [seCreateExportJob](#)

[Workflow](#), [Methode](#)

From:

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

Permanent link:

https://dokuwiki.unitec.de/doku.php?id=primus:createjob_statuspostcalls

Last update: **2025/02/05 08:27**



ctrltypes

Bei **ctrltypes** handelt sich um ein [Attribut](#) von [GUI_Data_Sheets](#)-Elementen. Die Namen der ctrltypes sind grundsätzlich klein zu schreiben.

Name	Funktion
button	Eine Schaltfläche im Formular mit der eine Funktion aufgerufen werden kann
checkbox	
checkedlistbox	Eine <code>listbox</code> mit Checkboxes für eine Mehrfachauswahl der Einträge
combobox	Eine Liste von Werten aus denen der Anwender wählen oder einen Wert eingeben kann
datetime	
groupbox	Grafische Gruppierung von Elementen in einem Formular
icon	Bindet ein Icon in ein Formular ein
label	Beschriftungstext im Formular
listbox	Eine Liste von Werten aus denen der Anwender einen wählen kann
picture	Bindet eine Bilddatei in ein Formular ein
textbox	Bietet ein Eingabefeld für Freitexteingabe

Beispiele

```
<Material ctrltype="listbox" />
<Logo ctrltype="picture" />
<FileName ctrltype="textbox" />
```

Siehe auch

- Liste der [types](#)

[datasheet](#), [gui](#), [ctrltype](#)

From:

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

Permanent link:

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

Last update: **2024/11/11 13:55**



PrimusDataConnect

Bei **PrimusDataConnect** handelt es sich um zwei von Unitec entwickelte Anwendungen, die jeweils einen Teil der DataConnectv1 Schnittstelle implementieren. PrimusDataConnect besteht aus einem Primus [Addin](#), welches die Interaktion mit einem anderen System durch die grafische Benutzeroberfläche von Primus ermöglicht. Zum anderem besteht PrimusDataConnect aus einem Dienst der unabhängig von der grafischen Oberfläche, es dem anderen System ermöglicht mit Primus zu interagieren. Ein mögliches anderes System muss ebenfalls die DataConnectv1 Schnittstelle implementieren. Ein Beispiel ist ein [ERP-System](#), dessen Hersteller die DataConnectv1 Schnittstelle in sein implementiert hat.

Funktionen Primus Addin

Funktion	Typ	Bedingung	Parameter	Beschreibung
GetNewUniqueNumber	Menü		Selektiert Datei im Datagrid	GUID als Rückgabe
GetNewUniqueNumberEx	Menü		Selektierte Zeile(n) im DataGrid	ID aus externem Service als Rückgabe
GetNewUniqueNumberEx_Click	Button ³⁾		<DATENFELD>	Holt eine Nummer aus dem ERP und schreibt sie in das <DATENFELD>
SetBillOfMaterials	Menü	Baugruppe		Materialliste an externen Service übergeben
SetBillOfMaterials_Click	Button			Aufruf nur über Button
SetBillOfMaterials_ByPath			%FILEPATH%	Mehrere Dateien per „ “ trennen. Übergibt die Materialliste an einen externen Service
GetDocumentData	Menü	CAD Datei		ERP - Dokument abgleichen Klärungsbedarf
GetDocumentData_Click	Button			
SetDocumentsData				Menu im DataGrid, Selektierte Zeile(n) im DataGrid, Dokument im externen Service anlegen
SetDocumentsData_ByPath			%FILEPATH%	Mehrere Dateien per „ “ trennen. Legt ein Dokument im externen Service an
SetDocumentsData_Click				
ShowVersion				Gibt die Version des externen Services und des Addins zurück
UploadFiles	Menü		Selektierte Zeile(n) im DataGrid	Datei wird an den externen Service übergeben

Funktion	Typ	Bedingung	Parameter	Beschreibung
UploadFiles_ByPath			%FILEPATH%	mehrere Dateien per „ “ trennen, Datei wird an den externen Service übergeben
UploadFiles_Click	Button		Datei wird an den externen Service übergeben	

Beispiel

```
<ButtonGetNr ctrltype="button" data="DataConnect.GetNewUniqueNumber_Click  
Number" label="ERP - Nummer holen" type="Button" mode="W" left="393"  
top="170" width="150" height="22" />
```

[DataConnect](#), [SolidEdge](#), [Addin](#), [Menu](#), [GUI](#)
³⁾

Die Funktion prüft ob sie über einen Button aufgerufen wurde

From:

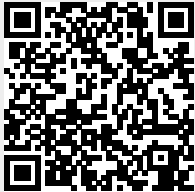
<https://dokuwiki.unitec.de/> - **Unitec Wiki für Produktdokumentationen**

Permanent link:

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

Last update:

2026/01/22 12:45



Daten-XML

Die **Daten-XML** ist das zentrale Element der Primus *No-SQL* Datenbank. In dieser Datei werden alle Informationen zu einem Dokument abgelegt. Die Daten-XML wird von den Primus Clients automatisch angelegt sobald diese ein Dokument ohne zugehörige Daten-XML finden, dabei wird auf die Definition unter [XML_DataTemplates](#) zugegriffen. Der Dateiname einer Daten-XML ist immer der Dateiname des Dokuments inklusive Dateiendung und dieser wird dann noch mit `.xml` erweitert.

Aufbau

Das [Stammelement](#) einer Daten-XML ist `<XML_Document>`.

```
<?xml version="1.0" encoding="utf-16"?>
<XML_Document>
```

Darunter finden sich folgende Abschnitte:

Data

Im Abschnitt `<Data>` sind alle Datenfelder zu einem Dokument abgelegt. Das Tag selbst hat ein Attribut `lastdocwritetime` in dem [Primus](#) eine Zeitseriennummer des letzten Speicherzeitpunkts ablegt. Die Anzahl der Datenfelder kann zwischen verschiedenen Dokumenten variieren.

Datenfeld

Ein **Datenfeld** besteht aus einem [Tag](#) mit dem Namen des Datenfelds und dem [Attribut](#) `value` in dem der Datenfeldinhalt abgelegt wird. Das Datenfeld `<FILE_TYPE>` ist ein Pflichtfeld. Es gibt vom [Systemdefinierte Datenfelder](#), die nicht anderweitig genutzt werden dürfen.

Beispiel

```
<FILE_TYPE value="PAR" />
<MODIFIED value="0" />
<Revision value="00" />
<WriteStatus value="Read" />
<WorkflowStatus value="003" />
```

RefParents

In **RefParents** werden die übergeordneten Dokumente zu einem Dokument aufgelistet. Diese Information wird vom [Verwendungsnachweis](#) ausgewertet.

RefChildren

In **RefChildren** werden die untergeordneten Dokumente eines Dokuments abgelegt. Diese werden für die [Dokumentenliste](#) ausgewertet. Das Tag selbst hat ein Attribut `lastdocwritetime` in dem [Primus](#) eine Zeitseriennummer des letzten Speicherzeitpunkts ablegt.

Nur bei Verwendung von DataConnect

Wenn Primus in Verbindung mit **DataConnect** verwendet wird, dann legt [DataConnect](#) eigene Tags mit weiteren Informationen ab.

VariableList

PropertyList

PhysicalProperties

OccurrenceList

Finden beschädigter Daten-XML

Wenn Daten-XML beschädigt sind, z.B. durch Unterbrechung der Netzwerkverbindung während einer Schreiboperation, dann werden diese Dokument in Primus nicht mehr aufgelistet. Allerdings wird auch keine neue Daten-XML angelegt, da eine solche ja bereits existiert. Dazu gibt es einen [Menübefehl `prmFindCorruptXMLDataFiles`](#), um diese Daten-XML zu finden.

Siehe auch

- [XML_DataTemplates](#)
- [Liste der Systemdefinierten Datenfelder](#)
- [prmFindCorruptXMLDataFiles](#)

From:

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

Permanent link:

<https://dokuwiki.unitec.de/doku.php?id=primus:daten-xml>

Last update: **2025/06/03 10:41**



DocumentOpen

Mit **DocumentOpen** werden im [Abschnitt Public](#), verschiedene Optionen für das Öffnen von Dokumenten definiert.

Attribute

Attribut	Wert	Bedeutung
showWriteProtectedMsg	0 / 1	Definiert ob der Hinweis auf ein schreibgeschütztes Dokument beim Öffnen angezeigt wird. Der Hinweistext wird als Elementtext unter <code>\$Primus.MSG_FILEWRPROT</code> abgelegt

Beispiel

Es soll keine Meldung beim Öffnen schreibgeschützter Dokumente angezeigt werden.

```
<DocumentOpen showWriteProtectedMsg="0" />
```

[options](#), [public](#)

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

Permanent link:
<https://dokuwiki.unitec.de/doku.php?id=primus:documentopen>

Last update: **2023/02/08 12:01**



DocumentOpenCall

Mit **DocumentOpenCall** werden im [Abschnitt](#) Public, die zu verwendenden [Addins](#) für das Öffnen von Dokumenten definiert. Es handelt sich bei **DocumentOpenCall** um ein reserviertes Tag.

Attribute

Attribut	Wert	Bedeutung
<DATEITYP>	Addin.Methode	Definiert den Dateityp ohne Punkt und das zugehörige Addin das zum Öffnen verwendet werden soll. Evtl. noch Parameter für das Addin.
<DATEITYP>_COND	Bedingung	Definiert die Bedingung unter der das Addin verwendet werden soll.

Beispiel

Es soll mit dem [Solid Edge](#) Addin nur dann geöffnet werden wenn es sich um eine CAD Client Lizenz handelt.

```
<DocumentOpenCall PAR="SEInterface.seOpen r"
PAR_COND="$(Condition.isCADLic)"/>
```

Siehe auch

- [DocumentRefListCall](#)
- [seOpen](#)

[Options](#), [Public](#), [Hardcoded](#)

From:

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

Permanent link:

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

Last update:

2025/01/22 14:21



DocumentRefListCall

Mit dem Tag **DocumentRefListCall** im [Abschnitte Public](#) in der [CFG.XML](#) können verschiedene [Attribute](#) festgelegt werden, welche die Verarbeitung von Dokumentverknüpfungen steuern. Die Attribute definieren für jede Dateiendung die Funktion die Primus aufrufen und die [Bedingung](#) unter der dies geschehen soll.

Attribute	Beschreibung
ASM	Geben das Addin und die Methode an mit dem auf die Verknüpfungen zugegriffen werden kann, z.B. SEInterface.seGetSERefList
ASM_COND	Geben die Bedingung an unter der die Verarbeitung erfolgen soll, z.B. nur mit CAD Lizenz
Weitere Attribute mit analoger Bedeutung	
DFT	
DFT_COND	
PAR	
PAR_COND	
PSM	
PSM_COND	
PWD	
PWD_COND	

Beispiel

Siehe hierzu auch das [Anwendungsbeispiel](#).

```
<DocumentRefListCall PAR="SEInterface.seGetSERefList"
PSM="SEInterface.seGetSERefList" PWD="SEInterface.seGetSERefList"
ASM="SEInterface.seGetSERefList" DFT="SEInterface.seGetSERefList"/>
<DocumentRefListCall PAR_COND="$(Condition.isCADLic)"
PSM_COND="$(Condition.isCADLic)" PWD_COND="$(Condition.isCADLic)"
ASM_COND="$(Condition.isCADLic)" DFT_COND="$(Condition.isCADLic)"/>
```

Siehe auch

- [CheckActiveLicenseType](#)
- [DocumentOpenCall](#)
- [prmProjectUpdateAllRefs](#)

[Public](#)

From:

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

Permanent link:

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

Last update: **2025/01/22 14:20**



Elementtext

Mit **Elementtext** ist es möglich Beschriftungen in der [Primusoberfläche sprachabhängig](#) und damit variabel zu halten. Dazu wird anstelle der direkten (hardcoded) Eingabe eines Textes, ein Platzhalter angegeben. Dieser wird beim Start von Primus, abhängig von der eingestellten Sprache, durch einen Text aus der, in einer [Sprachdatei](#) hinterlegten, Tabelle ersetzt.

Sprachdefinition

Die Definition der zu verwendenden Sprache erfolgt in der [Primus.xml](#) im [Stammelement](#) mit dem Attribut **stringspath**.

Aufbau eines Elementtextes

Der Aufbau eines Elementtextes sieht folgendermaßen aus:

`$<SPRACHDATEI>.SAMPLE_ELEMENTTEXT`. Er beginnt immer mit einem Dollarzeichen (\$), gefolgt vom Namen der Sprachdatei ohne Dateiendung, dann ein Punkt (.) und anschließend der eigentliche Platzhaltertext. Der Übersichtlichkeit wegen wird der Platzhaltertext immer in Großbuchstaben geschrieben, Leerzeichen sind nicht erlaubt. Bei der Verwendung von Platzhaltertexten ist die Groß- / Kleinschreibung unbedingt zu beachten.

Mehrere Elementtexte kombinieren

Es ist möglich mehrere Elementtexte zu kombinieren. Allerdings muß zwischen den beiden Elementtexten ein Leerzeichen als Trennung verwendet werden.

`$<SPRACHDATEI>.SAMPLE_ELEMENTTEXT $<SPRACHDATEI>.SAMPLE_ELEMENTTEXT_2`.

Elementtext mit Hardcoded Text ergänzen

Es ist möglich einen Elementtext mit einem festen Text zu ergänzen. Auch hier muß zwischen dem Elementtext und dem festen Text ein Leerzeichen zur Trennung sein.

`$<SPRACHDATEI>.SAMPLE_ELEMENTTEXT` (Nur als Beispiel)

Beispiel

```
<Title label="$Primus.SHEET_TITLE" />
```

Wie man Platzhaltertexte definiert wird unter [Spracheinstellung](#) näher erklärt.

Systemdefinierte Platzhalter

Für einige Funktionen und Meldungen in Primus gibt es vordefinierte Platzhalter. Welche dies sind, steht in der [Liste der Systemdefinierten Platzhalter](#).

Suche nach Hardcoded Texten in der CFG.xml

Mit der [Regular Expression](#)

```
(label=")(?![\\$,\\"])
```

kann in [Notepad++](#) nach Texten in der [CFG.XML](#), die nicht durch einen Eintrag in einer Sprachdatei definiert sind, gesucht werden. Wobei statt text auch

- label
- tooltip
- title

verwendet werden kann.

[GUI](#), [Sprachabhängig](#), [Textersetzung](#), [Case sensitiv](#), [Regular Expression](#)

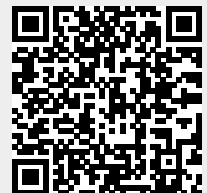
From:

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

Permanent link:

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

Last update: **2023/06/15 12:32**



Farbdefiniton

In Primus werden **Farben** für verschiedene Zwecke eingesetzt, z.B.:

- nicht ausgefüllte Pflichtfelder
- Hintergrundfarbe von Formularen
- Farbe einer [Groupbox](#)
- Farbe von Textlabels
- Farben von [WorkflowStatus](#)⁴⁾

[klaerungsbedarf](#)

RGB-Farben

Müssen Farben als RGB Werte angegeben werden, dann werden die Farbanteile für **R**ot, **G**rün und **B**lau, durch Komma getrennt aufgelistet. Zulässig sind für jeden Farbanteil die Werte von 0 - 255. Beispiel für die Hintergrundfarbe von [Formularen](#):

```
<Sample bgcolor="255,255,255">
```

Vordefinierte Farben

Es gibt in Primus Farben die durch Angabe des Namen verwendet werden können. Diese Farben werden in *List_WorkflowStatus* verwendet.

Farbname
DarkGrey
DarkRed
Green
LightCoral
LightCyan
LightSalmon
White

Für weitere Farben siehe unter [Weblinks](#).

[klaerungsbedarf](#)

Weblinks

- [Übersicht von HTML-Farben](#)

[gui](#), [Menu](#), [DataSheet](#), [DataList](#)

⁴⁾

Nur namentlich vordefinierte Farben

From:

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

Permanent link:

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

Last update: **2023/02/08 12:01**



FastFilter

Im Abschnitt **FastFilter** der [CFG.XML](#) unter [GUI](#), können Filterlisten für die Schnellsuche konfiguriert werden.

Aufbau von FastFilter

Tag			Attribute	Beschreibung
<FastFilter>				Definition des Abschnitts unter GUI
			active	Definiert den aktiven Filter
			columnwidth0	Definiert die Breite der Spalte mit der Feldbezeichnung
			columnwidth1	Definiert die Breite der Spalte mit dem Suchbegriff
			mode	Möglicherweise der Beginn der Suche, Aktueller Ordner, mit Unterordnern, vom obersten Ordner Klärungsbedarf
	Filter1	DATENFELD		Definition einer Filterliste
			name	Anzeigename für die Filterliste
				Bezeichnet die Zeile eines Suchfeldes i.d.R. = data
			data	Name eines Datenfelds nach dem gesucht werden kann
			label	Anzeigename des Datenfelds
			filtertype	Suche nach <i>enthält</i> (1) oder <i>ist gleich</i> (0)
	Filter2			Weitere Filterlisten
	...			

Voraussetzungen für die Suche

Um nach einem Datenfeld zu suchen, muss dieses in der entsprechenden Suchdatenbank als zu indizierendes Feld eingetragen sein. In der [GUI_Document_List](#) muss eine sichtbare Spalte für dieses Feld konfiguriert sein. Andernfalls liefert die Suche kein Ergebnis.

Siehe auch

- [PrimusSearchService](#)

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

Permanent link:
<https://dokuwiki.unitec.de/doku.php?id=primus:fastfilter>

Last update: **2023/06/21 09:53**



FileAccessManagementServer

Beim **FileAccessManagementServer** handelt es sich um einen Abschnitt in Form eines [Leer-Tags](#). Mit diesem Abschnitt wird definiert wie ein Primus Client mit dem [fileaccessmanager](#) Dienst (FAM) kommunizieren kann.

Attribute

Das Leer-Tag hat folgende Attribute:

Name	Wert	Funktion
server		Gibt den Namen des Servers an
port	9009	IP-Port auf dem
writinguser	%USER%	Klaerungsbedarf

From:

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

Permanent link:

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

Last update: **2023/02/08 12:01**



Folder_Paths

Im [Abschnitt Folder_Paths](#) werden die Verzeichnisse die Primus im [Ordnerfenster](#) anzeigen soll soll definiert. Dem [Tag](#) können verschiedene globale [Attribute](#) übergeben werden.

Attribute

Folgende Attribute sind möglich:

Attribut	Funktion
checkinghidden	Setzt das Hidden Attribut des Dateisystems bei den XML-Dateien
checkwriteaccess	XML die Status Read haben, werden automatisch auf Readonly gesetzt CADAdaptApproved
checkiswriteable	Steuert den Dateischreibschutz mittels WriteStatus
checkiswriteable_directory	Steuert den Dateischreibschutz mittels WriteStatus
dirlogicalsort	Aktiviert (1) die logische Sortierung für Verzeichnisse
ignoredirectories	Listet Verzeichnisnamen auf die Primus ignorieren soll
localworkmode	Steuert ob sich Primus im localworkmode befindet.
offlinemode	Steuert ob sich Primus im offlinemode befindet.
parentsrefscheck	Aktiviert / Deaktiviert das automatische Aktualisieren von Parentreferenzen
visibleattachfilescond	Definiert welche Dateitypen in den Anhängen angezeigt werden sollen. Ein - zu Beginn negiert die Auswahl
visiblefilescond	Definiert welche Dateitypen angezeigt werden sollen. Ein - zu Beginn negiert die Auswahl
visibleprojectscond	Definiert unter welchen Bedingungen ein Verzeichnis angezeigt werden soll
watchactivedir	Gibt die Zeit in Millisekunden an mit der das aktive Verzeichnis auf Änderungen überwacht werden soll. 0 - Keine Überwachung, 1 - Update durch Server

Path Einträge

Die einzelnen Pfade werden mit einem [Leer-Tag](#) Namens [Path](#) definiert. Es können beliebig viele Path Einträge vorgenommen werden.

Attribute

Folgende Attribute sind für die PATH Einträge möglich:

Attribut	Funktion
path	Gibt den Datenpfad an, UNC-Pfade sind möglich

Attribut	Funktion
localpath	Gibt den lokalen Speicherort für lokale Arbeitsweise an
name	Anzeigename in der Ordnerfenster
datacontext	Optional eigener Datakontext für diese Datenablage
id	FileServer
db	Name der zugehörige PrimusSearchService Datenbank, siehe SearchServiceDB
iconbytype	Zeigt Dateitypabhängige, in einer Liste definierte, Icons in der Dokumentenliste an
readtopxml	Definiert ob (0/1) nach einer .PRO.XML für den obersten Ordner gesucht werden soll
type	Gibt den Type des Pfades an und steuert für vordefinierte Datacontext auch diesen
condition	Definiert eine Bedingung die bestimmt ob dieser PATH Eintrag gültig ist. Diese Bedingung wird beim Laden der Konfiguration einmalig geprüft.
hidefilecond	Wenn die obige Bedingung erfüllt ist, dann definiert diese Bedingung welche Dokumente ausgeblendet werden sollen. Diese Bedingung wird für jede Datei einer Liste, bei jedem Listenaufbau geprüft.
visiblefilescond	Wenn es sich um einen separaten datacontext handelt, hat dies die gleich Bedeutung wie das Attribut unter Folder_Path

From:

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

Permanent link:

https://dokuwiki.unitec.de/doku.php?id=primus:folder_paths

Last update: **2025/08/13 08:26**



GetListTextByParam

Mit der **Funktion GetListTextByParam** ist es möglich, zu einem Wert das zugehörige Attribut auszulesen. Die Funktion wird mit **CALL** aufgerufen.

Beispiele

Syntax: <LISTENNAME>.<ATTRIBUT>|<GESUCHTER WERT>

Als Bedingung

Zu einer Artikelgruppe wird das Attribut job ausgelesen und als Bedingung verwendet, um nur bestimmte Artikelgruppen bei der Jobverarbeitung zu berücksichtigen.

```
'CALL(GetListTextByParam List_Artikelgruppe:job|#(Artikelgruppe))==1'
```

Siehe auch

- [Listenmapping](#)
- [GetListTextFromDict](#)

[Funktion](#), [Workflow](#), [Liste](#)

From:

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

Permanent link:

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

Last update: **2023/07/14 04:42**



Glossar

Hier werden die Begriffe, die in dieser Dokumentation verwendet werden, näher erklärt.

A

Abschnitt

Mit **Abschnitt** wird ein Bereich in der Primus.CFG.XML bezeichnet. Ein Abschnitt beginnt mit einem **Start-Tag** und endet mit einem **End-Tag**.

```
<GUI_Data_Lists> <!--Start-Tag-->
<!--Inhalt des Abschnitts-->
</GUI_Data_Lists> <!--End-Tag-->
```

Ausnahme sind die Abschnitte die aus einem sogenannten **Leer-Tag** bestehen. Hier wird der Inhalt des Abschnitts zusammen mit Start- und End-Tag in eine Zeile geschrieben. Als Inhalt kommen in dem Fall nur **Attribute** in Frage.

```
<ListViewSettings splitterdist="376" mode="0" showbehinddatalist="1"
splitterorient="Ver" />
```

Administrator

Bei einem Administrator handelt es sich i.d.R. um ein Benutzerkonto mit erhöhten Rechten, dem es möglich ist bestimmte Aufgaben auszuführen.

Attribut

Ein **Attribut** ist Bestandteil eines **Tags**. Ein Attribut muß innerhalb eines Tags immer eindeutig sein, hierbei wird die Groß-/Kleinschreibung unterschieden. Einem Attribut ist stets ein Wert zugewiesen. Die Wertzuweisung erfolgt mittels =, der Wert selbst muß in " gesetzt sein. Die Wertzuweisung kann auch leer sein, die Zuweisung erfolgt dann durch zwei aufeinanderfolgende "".

```
<Preview mode="viewer" />
<Filename label="" /> <!--Zuweisung eines leeren Werts-->
```

Das **Tag** ist in diesem Fall **Preview**, das **Attribut** **mode** und der **Wert** **viewer**. Bestimmte Zeichen sind innerhalb eines Wertes nicht erlaubt. Falls diese in einem Wert jedoch trotzdem benötigt werden, müssen sie durch sog. Entitäten ersetzt werden. Die folgende Tabelle zeigt die nicht zulässigen Zeichen und den Ersatztext.

Verbotenes Zeichen	Entität	Bemerkung
--------------------	---------	-----------

Verbotenes Zeichnen	Entität	Bemerkung
<	<	lower than (kleiner als)
>	>	greater than (größer als)
'	'	apostroph
„	"	quota (Anführungszeichen)
&	&	ampersand (kaufm. und)

D

Dateiendung

Bei der Dateiendung handelt es sich um den rechten Teil des Dateinamens der mit einem Punkt (.) vom linken Teil des Dateinamens getrennt ist. Programme erkennen an der Dateiendung um welche Art Datei es sich handelt und ob sie diese verarbeiten können. In Windows können zu bestimmten Dateiendungen, Programme eingetragen werden die zum Verarbeiten der Datei aufgerufen werden.

Dienst

Ein Dienst, gelegentlich auch Daemon genannt, ist ein Prozess der auf einem Server im Hintergrund ohne Benutzerinteraktion läuft. Dienste werden von Programmen aufgerufen um bestimmte Funktionen, die der Dienst zur Verfügung stellt, zu nutzen oder ihm bestimmte Aufgaben zu übergeben. Dienste können auch über das Netzwerk von anderen Rechnern aus genutzt werden.

DLL

Die Abkürzung **DLL** steht für **Dynamic Link Library**, und bezeichnet eine Programmbibliothek die dynamisch, also nur bei Bedarf, vom aufrufenden Programm in den Speicher geladen wird. Bei Primus werden z.B. [Addins](#) auf diese Weise eingebunden. Dies betrifft sowohl Primus eigene als auch Kundenspezifische Zusatzfunktionen.

E

End-Tag

Ein **End-Tag** kennzeichnet das Ende eines Tag. Es besteht aus `</Tag-Name>`. Ausnahme: Bei einem [Leer-Tag](#) wird ein vereinfachter Abschluss ohne Name, nur mit `</>` verwendet.

ERP-System

Ein **ERP-System** ist ein System für Enterprise-Resource-Planning (ERP). Die Aufgabe eines solchen Systems ist vereinfacht ausgedrückt, die Ressourcen die einem Unternehmen zur Verfügung stehen, zeitlich und mengenmäßig so zu steuern das die betrieblichen Abläufe möglichst störungsfrei ablaufen

können, um so den Wertschöpfungsprozess zu optimieren.

F

Freigabe

Eine **Freigabe** ist ein Verzeichnis auf einem Dateiserver welches unter einem bestimmten Namen aus dem Netzwerk heraus angesprochen werden kann. Die Freigabe kann einen anderen Namen haben als das Verzeichnis im Dateisystem der Platte, auch kann ein Verzeichnis verschiedene Freigaben haben. Jede Freigabe kann eigene Berechtigung für die Benutzer mit sich bringen. Freigaben werden z.B. in der [Primus.CFG.XML](#) im Abschnitt [Folder_Paths](#) verwendet.

Funktion

Eine Funktion ist ein Programm oder eine Programmroutine welche einen Wert an den Aufrufenden zurück gibt.

G

GUID

Die Abkürzung **GUID** steht für *Globally Unique Identifier*, damit ist eine eindeutige Zahlenfolge gemeint. Die Wahrscheinlichkeit das zwei Rechner unabhängig voneinander eine identische GUID erzeugen ist sehr gering ($2,71 \times 10^{18} \approx 2,71$ Trillionen).

I

Icon

Ein Icon ist ein Symbolbild das z.B. einen bestimmten Dateityp darstellt, oder auch als grafisches Element in der Primus Oberfläche verwendet wird.

K

Kommentar

Ein **Kommentar** ist ein speziell markierter Text im Quellcode eines Programms oder einer Konfigurationsdatei. Dieser Text soll die Verständlichkeit für den Bearbeiter erhöhen. Ein Kommentar wird vom jeweils lesenden Programm ignoriert. In [XML](#) beginnt ein Kommentar mit `<!--` und endet mit `-->`. Alles was zwischen Kommentaranfang und -ende steht wird vom [XML-Parser](#) ignoriert.

Beispiel:

```
<GUI_Data_Lists>
  <!-- Das ist ein Kommentar -->
  <GUI_VirtualFolders_List>
```

Kontextmenüs

Um das [Kontextmenü](#) aufzurufen müssen Sie die rechte Maustaste auf einem ausgewählten Objekt kurz drücken. Die Befehle unterscheiden sich je nach ausgewähltem Objekt.

L

Lebenszyklus

Leer-Tag

Bei einem **Leer-Tag**, stehen Anfangs- und End-Tag in einer Zeile. Ein Leer-Tag kann keine weitere Unterteilung enthalten. Ein Leer-Tag kann ausschließlich [Attribute](#) enthalten. Einträge in [Lists](#) sind z.B. Leer-Tags. Beispiel:

```
<entry value="0" text="Normal" />
```

P

Platzhalter

Bei einem Platzhalter handelt es sich um ein in Prozentzeichen (%) eingefassten Text, der zur Laufzeit von Primus mit einem Wert gefüllt wird und zum Beispiel zur Verarbeitung ausgewählter Dokumente (%FILEPATH%) genutzt wird. Es gibt eine [Liste aller Platzhalter](#).

S

Stammelement

Das **Stammelement** ist ein spezielles [Primusspezifisches Tag](#).

Start-Tag

Ein **Start-Tag** definiert den Beginn eines [Tags](#). Es besteht immer aus [<Tag-Name>](#). Ausnahme: Bei

einem **Leer-Tag** entfällt das > hinter dem Tag-Namen und es folgen direkt die **Attribute**.

T

Tag

Bei einem **Tag** handelt es sich um ein XML Strukturelement. Ein Tag hat einen **Anfang**, einen Namen und ein **Ende**. Darüber hinaus können auch noch **Attribute** zugewiesen werden.

```
<Das_Tag das_Attribut="Wert des Attributs" />
```

Tag-Name

Der **Tag-Name** ist ein notwendiger Bestandteil eines **Tag**.

U

UNC-Pfad

Unter UNC (Uniform Naming Convention) versteht man die Angabe eines Datei- oder Verzeichnispfades unter Angabe des Rechner- und des Freigabennamens nebst weiteren Verzeichnissen unterhalb der Freigabe. Der Pfad beginnt immer mit \\.

```
\\<SERVER>\<FREIGABE>\<VERZEICHNIS>\
```

W

Wildcard

Wildcards sind bestimmte Zeichen welche als Platzhalter für die Suche oder Filterung von Datei- und Pfadnamen verwendet werden. Unter Windows sind dies ein ? für ein einzelnes Zeichen und ein * für eine beliebige Zeichenfolge. Das bekannte *.* filtert auf eine beliebige Zeichenfolge der mittels eines Punkts von einer zweiten Zeichenfolge getrennt ist.

X

XML

Die Abkürzung **XML** bedeutet *Extensible Markup Language* und beschreibt eine Textdatei, deren

Inhalt hierarchisch und strukturiert gegliedert ist. Dadurch ist der Inhalt sowohl für Menschen als auch Maschinen gleichermaßen lesbar. Die Daten werden in [Tags](#) und [Attributen](#) abgelegt.

XML-Parser

Bei einem **XML-Parser** handelt es sich um ein Programm oder den Teil eines Programms das/der eine XML-Datei liest und zur weiteren Verarbeitung zur Verfügung stellt. Der XML-Parser ist dabei so aufgebaut, dass er nur die jeweils gewünschten Informationen ausliest und alle anderen Informationen ignoriert. Von Primus werden alle XML-Dateien über StreamReader gelesen. Dadurch werden die XML-Dateien immer sauber geschlossen und blockieren keine anderweitigen Zugriffe.

[Klärungsbedarf](#)

From:

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

Permanent link:

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

Last update: **2025/05/14 07:52**



groupbox

Mit dem Wert **groupbox** für das Attribut **ctrltype** kann man Datenfelder in DataSheets optisch gruppieren. Eine Groupbox hat sonst keine weitere Funktion.



Zugehörige Attribute

Um eine Groupbox in einem Formular zu definieren, müssen neben dem Attribut **ctrltype** noch weitere Attribute ausgefüllt sein. Prinzipiell können alle Attribute aus DataSheets verwendet werden, haben aber keine Bedeutung.

Attribut	Pflicht	Funktion
forecolor	Ja	Definiert die Farbe der Groupbox mittels RGB Werten
labelcolor	Nein	Definiert die Farbe für den Text der Groupbox
height	Ja	Gesamthöhe der Groupbox einschließlich des linken senkrechten Balkens
width		Breite der Groupbox
labelposleft		Kann entfallen da der Labeltext immer auf der Groupbox liegt

Tag-Name

Der Name des **Tag** hat keine Bedeutung, muss aber innerhalb des DataSheet Abschnitts eindeutig sein. Es empfiehlt sich folgende Namenskonvention: `GROUPBOX_<Sammelbegriff für die Datenfelder>`.

Beispiel

```
<GROUPBOX_Sample ctrltype="groupbox" label="SAMPLE" left="7" top="69"
width="359" height="89" labelcolor="255,255,255" forecolor="255,255,195" />
```

[gui](#), [datasheet](#), [ctrltype](#)

From:

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

Permanent link:

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

Last update: **2023/02/08 12:01**



GUI

Im [Abschnitt GUI](#) wird die Benutzeroberfläche von Primus konfiguriert. Der Abschnitt muss direkt im [Stammelement](#) liegen. Die Konfiguration enthält mehrere Unterabschnitte. In diesem Abschnitt sind [Kommentare](#) erlaubt.

Name	Pflicht	Funktion
GUI_Data_Lists	Ja	Definiert die Listenansichten
GUI_Data_Sheets	Ja	Hier werden alle in Primus verwendeten Datenblattansichten definiert
GUI_Menus	Ja	Hier werden alle Kontextmenü von Primus definiert
ListViewSetting	Ja	Darstellungseinstellungen für die ListView
FastFilter	Ja	Vordefinierte Filtereinstellungen
PreviewAddins	Ja	Defintion der externen Viewer für diverse Dateiformate
Preview	Ja	Einstellungen für den Modus des Preview Fensters

Siehe auch

- [Spracheinstellung](#)

[GUI](#), [Kommentierbar](#)

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

Permanent link:
<https://dokuwiki.unitec.de/doku.php?id=primus:gui>

Last update: **2023/06/21 09:47**



GUI_Data_Lists

Im [Abschnitt GUI_Data_Lists](#) werden die Listenansichten in Primus definiert. Einige Listeneinstellungen haben Einfluss auf andere Primusfunktionen. So werden z.B. in [GUI_Document_List](#), die [Textersetzen](#) für den Eigenschaftsabgleich durch das [Solid Edge ApplInterface](#) definiert.

Liste	Funktion
GUI_VirtualFolders_List	Listet die Inhalte von virtuellen Verzeichnissen auf
GUI_Document_List	Dokumentenlistenlistenansicht
GUI_Document_List_BOM	klaerungsbedarf
GUI_Document_List_BOMStruct	klaerungsbedarf
GUI_DocumentAttachments_List	Liste der Anhänge zu einer Datei
GUI_StdParts_List	klaerungsbedarf

Attribute

Einer Liste können optional direkt Attribute für die [Sortierung](#) zugewiesen werden.

Attribut	Wert	Funktion
sort	<SPALTENNAME>	Name der Spalte nach der sortiert werden soll.
sorttype		Art der Sortierung
	A	Aufsteigend (A-Z)
	D	Absteigend (Z-A)

Beispiel

```
<GUI_Document_List sort="FileName" sorttype="A">
```

[gui](#), [datalist](#)

From:

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

Permanent link:

https://dokuwiki.unitec.de/doku.php?id=primus:gui_data_lists

Last update: **2023/02/08 12:01**



GUI_Data_Sheets

Der [Abschnitt GUI_Data_Sheets](#) ist ein Unterabschnitt von **GUI**, in dem die verschiedenen Datenblätter von Primus definiert werden. Die Datenblätter werden z.T. von [Menübefehlen](#) benötigt, z.B. beim Neu anlegen von Dokumenten. In diesem Abschnitt sind [Kommentare](#) erlaubt. Ein Datenblatt kann direkt aus der Primus Oberfläche heraus, mit gedrückter *STRG*-Taste und einem Rechtsklick ins Datenblatt, [editiert](#) werden.

Name	Pflicht	Funktion
GUI_Project_DataSheet	Ja	Zeigt die Eigenschaften von Projekten und Verzeichnissen an
GUI_NewProject_DataSheet	Ja	Anlegen von neuen Projekten
GUI_VirtualFolder_DataSheet	Ja	Anlegen von virtuellen Verzeichnissen
GUI_Document_DataSheet	Ja	Zeigt die Eigenschaft von Dokumenten an
GUI_NewDocument_DataSheet	Ja	Erstellen einer neuen Datei
GUI_DocumentBOM_DataSheet	Ja	klaerungsbedarf
GUI_NewISLDocument_DataSheet	Nein	Neu anlegen von ISL-Normteilen
GUI_NewDocument_DataSheet_OFFICE	Nein	Neu anlegen von Office Dokumenten
GUI_NewDocument_DataSheet_DRAW	Nein	Neu anlegen einer CAD-Zeichnung
GUI_RenameDocument_DataSheet	klaerungsbedarf	klaerungsbedarf
GUI_NewAttachment_DataSheet	klaerungsbedarf	Neu anlegen eines Anhangs
GUI_DocumentAttachments_DataSheet	klaerungsbedarf	Zeigt die Eigenschaften von Anhängen an

Verwendung

Die Anhänge werden von Primus selbst für bestimmte Fensterbereiche oder von Primus [Menübefehlen](#) verwendet.

Attribute

Die DataSheets können [Attribute](#) haben.

Attribut	Funktion
bgcolor	Hintergrundfarbe des Dialogs in RGB Werten
height	Gibt die Höhe des Datenblatts in Pixel an.
title	Überschrift des Datenblatts ⁵⁾
width	Gibt die Breite des Datenblatts in Pixel an

Beispiel:

```
<GUI_NewDocument_DataSheet height="163" title="Dateiname"
bgcolor="255,255,255">
```

Grafische Kontrollelemente

Für den Aufbau eines Datenblatts stellt Primus spezielle Grafische Elemente sog. Control Types ([ctrltype](#)) zur Verfügung.

[datasheet](#), [gui](#), [Kommentierbar](#)

5)

Unterstützt jedoch keine Platzhaltertexte

From:

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

Permanent link:

https://dokuwiki.unitec.de/doku.php?id=primus:gui_data_sheets

Last update: **2023/02/08 12:01**



GUI_Document_DataSheet

Der **Abschnitt GUI_Document_DataSheet**, ist das zentrale **Datenblatt** in dem die **Datenfelder** zu einem in der **Dokumentenliste** markierten Dokument angezeigt und, außer im **Primus-Viewer**, auch geändert werden können. Das Tag *GUI_Document_DataSheet* kann folgende Attribute haben:

Attribut	Funktion
bgcolor	Hintergrundfarbe des Dialogs in RGB Werten
height	Gibt die Höhe des Datenblatts in Pixel an.
width	Gibt die Breite des Datenblatts in Pixel an

Datenblatteinträge

Die einzelnen Datenblatteinträge werden jeweils als einzelnes **Leer-Tag** aufgebaut. Der **Tag-Name** hat keine besondere Bedeutung, er muss aber innerhalb des Abschnitts eindeutig sein. Der Übersichtlichkeit wegen sollte der Tag-Name aber einen Bezug zu der dargestellten Information haben. Die Reihenfolge der Tags im Abschnitt ist für die Reihenfolge der Bearbeitung von Bedeutung wenn man mit Tabulator durch das Formular springt, die Anordnung der Elemente eines Formulars wird jedoch durch entsprechende Attribute gesteuert. Ein Datenblatteintrag kann folgende Attribute haben:

Attribut	Pflicht	Wert	Funktion
callafterchange		Ruft nach direkt erfolgter Änderung ein Programm auf	
ctrltype	X		Definiert den Elementtyp
data			Abhängig vom ctrltype , ein xml-datenfeld oder ein Parameter für das Element
enabled		0 / 1	Definiert ob ein Element aktiv ist oder nicht (siehe auch Bedingung)
height	X	<ZAHL>	Höhe des Datenfelds in Pixel
label		Elementtext	Beschriftung des Elements, die Anzeigeposition ist vom ctrltype abhängig
labelcolor		<RGB-Anteile>	Definiert die Schriftfarbe des Labels Default ist schwarz, weiß wäre „255,255,255“
labelposleft		<ZAHL>	Postion in Pixel, des Labeltextes vom linken Rand des Formulars (ctrltype abhängig)
left		<ZAHL>	Abstand des in Pixel vom linken Rand des Formulars
maxlength		<ZAHL>	Limitiert die maximal mögliche Eingabelänge
mode		R / W	Steuert ob das Element beschreibbar (W) ist oder nicht (R)
mode_singleselect			dito bei Einzelauswahl
multiselect		0 / 1	Gibt an ob das Datenfeld auch bei einer Mehrfachauswahl in der Dokumentenliste aktiv ist.
top	X	<ZAHL>	Positon des Elements in Pixel vom oberen Rand des Formulars

Attribut	Pflicht	Wert	Funktion
tooltip		Elementtext	Definiert den Text der als Tooltip angezeigt wird, wenn man mit der Maus auf das Element deutet
type	X		Elementtyp, der Wert hängt mit ctrltype zusammen und muss entsprechend gewählt werden
visible		0 / 1	Steuert ob ein Element sichtbar oder unsichtbar ist (siehe auch Bedingung)
width	X	<ZAHL>	Breite des Elements in Pixel

Defaultwerte in **fett**.

Datenfelder

Ein normales **Datenfeld** enthält folgende Attribute:

Attribut	Wert
Pflichtattribute	
ctrltype	textbox
data	xml-datenfeld oder primusvariable
height	20
left	130
top	in 22 Pixelschritten steigend
width	255
Optionale Attribute	
label	Elementtext
labelposleft	15
mode	R
multisel	1
tooltip	Elementtext
callafterchange	Externes Programm

Beispiel

```
<Title ctrltype="textbox" data="Title" label="$Primus.SHEET_TITLE"
type="Text" mode="W" left="130" top="193" width="255" height="20"
labelposleft="15" multisel="1" />
<-- Aufruf eines externen Programms bei Änderung des Feldwertes -->
<StkVP ctrltype="textbox" data="StkVP" label="VP/Stk. €" type="Text"
callafterchange="CustomerAddIn.calcStkVP" mode="W" left="88" top="220"
width="61" height="20" labelposleft="15" multisel="1" />
```

GUI, DataSheet

From:

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

Permanent link:

https://dokuwiki.unitec.de/doku.php?id=primus:gui_document_datasheet

Last update: **2023/08/08 11:37**



GUI_Document_List

Im [Abschnitt GUI_Document_List](#) werden die [spalten](#) für die [dokumentenliste](#) definiert. Darüber hinaus werden einige Spalteneinstellung vom [SEInterface](#) benutzt um Feldinhalte über [Listeneinträge](#) umzuwandeln.

Sortierung vordefinieren

Für diese Listenansicht kann die Sortierung vordefiniert werden. Dazu stehen zwei Attribute zur Verfügung.

Attribut	Wert	Funktion
sort	<SPALTENNAME>	Name der Spalte nach der sortiert werden soll.
sorttype		Art der Sortierung
	A	Aufsteigend (A-Z)
	D	Absteigend (Z-A)

Beispiel

```
<GUI_Document_List sort="Title" sorttype="A">
```

Spaltendefinition

Die einzelnen Spalten werden jeweils als eigener [Leer-Tag](#) definiert. Der Name des Tag ist frei wählbar, sollte der Übersichtlichkeit wegen aber mit dem Namen des dargestellten Datenfeldes übereinstimmen. Die Reihenfolge der Tag im Abschnitt hat keine Bedeutung, da die Reihenfolge der Spalten über ein [Attribut](#) definiert wird.

Attribute

Für die Spaltendefinition stehen folgende Attribute zur Verfügung:

Attribut	Pflicht	Funktion
data	Ja	Datenfeld der XML das in dieser Spalte verwendet werden soll
history	Nein	Spaltennummer in der historieliste
label	Nein	Spaltenüberschrift
list	Nein	Listenname aus welcher der Text oder das Symbol zum anzeigen genommen wird
position	Ja	Spaltennummern, dieser Wert bestimmt die Reihenfolge der Spalten in der Listenansicht
type	Ja	Gibt den Datenfeldtyp an
visible	Nein	Steuert ob die Spalte ein- (1 ⁶⁾) oder ausgeblendet (0 ⁷⁾) ist. Es kann nur nach sichtbaren Spalten gesucht oder gefiltert werden.

Attribut	Pflicht	Funktion
width	Ja	Spaltenbreite in Pixel

Beispiel

```
<FileTypeIcon data="FILE_TYPE" label="$Primus.COL_FILE_TYPE_ICONLIST"
type="FileTypeIcon" list="List_FileTypes" position="0" width="23"
visible="1" history="0" />
<Title data="Title" label="$Primus.COL_TITLE" type="Text" position="4"
width="154" visible="1" history="4" />
```

6)

TRUE ist auch zulässig

7)

FALSE ist auch zulässig

From:

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

Permanent link:

https://dokuwiki.unitec.de/doku.php?id=primus:gui_document_list

Last update: **2023/02/08 12:01**



GUI_Menus

Im Abschnitt **GUI_Menus** der [CFG.XML](#) werden die Menüs, die dem Anwender in Primus zur Verfügung gestellt werden, definiert. Es gibt Menüs die fest in der Oberfläche integriert sind und es gibt die [kontextmenus](#). Die Menüs sind in Haupt- und Untermenüs unterteilt.

Name	Datacontext		Pflicht	Funktion
	Context	Attribut		
gui_menu_attachment		attachments	Ja	
gui_menu_element			Ja	
gui_menu_extendedfilter				
gui_menu_folder				
gui_menu_jobserver	jobserver	folder		
gui_menu_linkedvirtualfolders				
gui_menu_newattachment		attachments_empty	Ja	
gui_menu_newelement		document_empty	Ja	
gui_menu_project		folder	Ja	
gui_menu_recyclebin	recyclebin	folder		
gui_menu_refelement_bom				
gui_menu_refelement_bomstruct				
gui_menu_refelement		ref	Ja	
gui_menu_relationships		relationships	Ja	Menüliste des Registers Beziehungen im registerkartenfenster
gui_menu_stdparts_dim				
gui_menu_stdparts_folder	stdparts	folder		
gui_menu_virtualfolders	virtfolders	folder		
Submenüs				
gui_submenu_cad				
gui_submenu_element				
gui_submenu_files				
gui_submenu_jobserver				
gui_submenu_local				
gui_submenu_localattachments				
gui_submenu_office				
gui_submenu_project_local				
gui_submenu_serefjobs				
gui_submenu_smartcopydocs				
gui_submenu_smartcopyprojects				

Menübefehle

Für die Gestaltung der Menüs stellt Primus [Menübefehle](#) zur Verfügung.

Datacontext

Es können spezielle [GUI](#) Elemente für verschiedene DataContext definiert werden. Der [datacontext](#) kann in [Folder_Paths](#) mit dem Attribut datacontext eines [Path](#) Eintrags deklariert werden, darüberhinaus existieren folgende vordefinierte DataContext die mit dem Attribut type des jeweiligen Path Eintrags zugewiesen werden.

Datacontext	Funktion
jobserver	Anzeige der Jobserver Jobs
recyclebin	papierkorb
stdparts	isl_normteile
virtfolders	virtuelle_ordner

Siehe auch

- [ctrltype](#)
- [GUI_Data_Sheets](#)
- [GUI_Data_Lists](#)
- [datacontext](#)

[menu](#), [GUI](#), [Klärungsbedarf](#), [DataContext](#)

From:

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

Permanent link:

https://dokuwiki.unitec.de/doku.php?id=primus:gui_menus

Last update: **2023/02/08 12:01**



GUI_NewDocument_DataSheet

Der [Abschnitt GUI_NewDocument_DataSheet](#) ist ein DataSheet das beim neu anlegen von Dokumenten verwendet wird. Es kann bei dem [Menübefehl prmelementsnew](#) im Attribut **parameter** als Datenblatt für Neuanlagen angegeben werden. Außerdem kann es in [Public](#) im Tag [newfilenamedialog](#) im Attribut **gui** als Wert eingetragen werden.

Besonderheiten

Gegenüber anderen DataSheets gibt es im **GUI_NewDocument_DataSheet** einige Besonderheiten.

PreAssign

Mit dem Attribut [preassign](#) können Werte zur Anzeige im Formular vorbelegt werden. Außerdem kann im [preassign](#) `%predata:Feld%` verwendet werden. Dabei werden zuerst die Dateieigenschaften aus der aktiven Datei laut [se_getprop_par/asm/psm/dft.ini](#) in die `predata`-Liste ausgelesen. So können die Werte aus der aktiven SE-Datei verwendet werden und nicht aus der XML in Primus. Da bei neuen SE-Dateien oft die Dateieigenschaften befüllt sind aber noch keine XML existiert.

Pflichtfeld

Unabhängig von der Definition als Pflichtfeld ist **Title** bei neuen Dokumenten immer ein Pflichtfeld und muß, wenn es im Dialog vorhanden ist ausgefüllt werden.

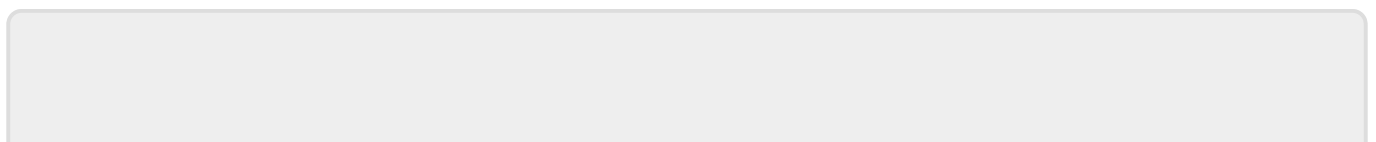
Nicht XML relevante Felder

Die Felder **FileNamePrefix** und **FileNameSuffix** können im Datenblatt mit Werten belegt werden, diese Werte werden nicht in die [dokument-xml](#) eingetragen, können aber in [assigndocumentdatafields](#) verwendet werden um z.B. Dateinamen zu bilden.

Kein ausblenden von Feldern möglich

Das Attribut `visible` wird derzeit (Stand: 06.12.2019) ignoriert. Als Workaround kann mit einem zweiten Datenblatt gearbeitet werden, das mit dem Tag [newfilenamedialog](#) in [Public](#) definiert werden. Über das Attribut [condition](#) kann dann gesteuert werden wann welcher Dialog angezeigt wird.

DataSheet



From:

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

Permanent link:

https://dokuwiki.unitec.de/doku.php?id=primus:gui_newdocument_datasheet

Last update: **2023/02/08 12:01**

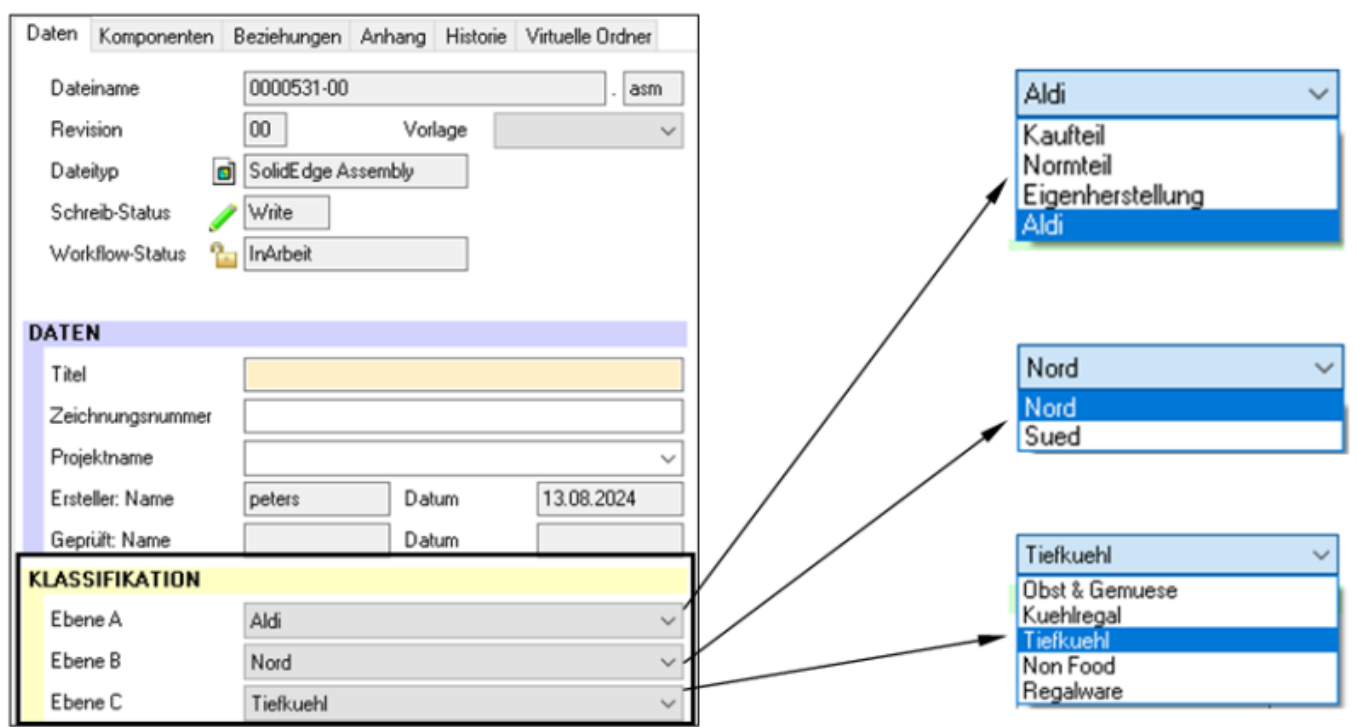


Hierarchische Listen

Listendefinitionen über mehrere Ebenen am Beispiel der Metadaten für die Klassifikation.

Definiert werden die Listen in der **\Ressources\Lists.xml** oder einer eigenen [Listendefinition](#). Die Listen werden über die **PrimusSE.cfg.xml** im Abschnitt [Lists](#) eingebunden

```
<Lists include=".\\Resources\LISTS.xml|.\\Resources\SE_MATERIAL.LIST.xml">
```



Liste der Ebene A

In einer übergeordneten Liste wird mit dem [Attribut](#) **childlist** bei jedem Eintrag angegeben aus welcher untergeordneten Liste bei Auswahl dieses Eintrags etwas ausgewählt werden kann.

Code	Erläuterung
<code><List_Classes1></code>	Name der Liste. Verweis in der PrimusSE.cfg.xml
<code><entry value=„Listeneintrag“ text=„\$Primus.str oder .ustr“ childlist=„Liste für nächste Ebene“ /></code>	Syntax
<code><entry value=„PRCHPART“ text=„\$Primus.LST_PRCHPART“ childlist=„List_Classes2_PurchasedPart“ /></code>	Eintrag 1
<code><entry value=„STDPART“ text=„\$Primus.LST_STDPART“ childlist=„List_Classes2_StandardPart“ /></code>	Eintrag 2
<code><entry value=„SELFPROD“ text=„\$Primus.LST_SELFPROD“ childlist=„List_Classes2_SelfProd“ /></code>	Eintrag 3

Code	Erläuterung
<entry value=„Aldi“ text=„\$Primus.LST_aldi“ childlist=„List_Classes2_Aldi“ />	Eintrag 4
</List_Classes1>	Abschluss der Liste

Liste der Ebene B für Auswahl Aldi

Code	Erläuterung
<List_Classes2_Aldi>	Name der Liste
<entry value=„North“ text=„\$Primus.LST_N“ childlist=„List_Classes3_ALDI“ />	Eintrag 1
<entry value=„South“ text=„\$Primus.LST_S“ childlist=„List_Classes3_ALDI“ />	Eintrag 2
<List_Classes2_Aldi>	

Liste der Ebene C für Auswahl Aldi Nord oder Süd

Code	Erläuterung
<List_Classes3_Aldi>	Name der Liste
<entry value=„Listeneintrag“ text= \$Primus.str oder .ustr />	Syntax Keine Childlist mehr
<entry value=„Fruit and Vegetable“ text=„\$Primus.LST_FRUIT“ />	Eintrag 1
<entry value=„Fresh“ text=„\$Primus.LST_Fresh“ />	Eintrag 2
<entry value=„Frozen“ text=„\$Primus.LST_FROZEN“ />	Eintrag 3
<entry value=„NON_FOOD“ text=„\$Primus.LST_NonFood“ />	Eintrag 4
<entry value=„REST“ text=„\$Primus.LST_REST“ />	Eintrag 5
</List_Classes3_Aldi>	

Listendefinition als Vorlage mit Verweis auf untergeordnete Listen

```
<List_Classes1>\\
<entry value="Wert1" text="$Primus.LST_WERT1"
childlist="List_Classes2_WERT1" />\\
<entry value="Wert2" text="$Primus.LST_WERT2"
childlist="List_Classes2_WERT2" />\\
</List_Classes1>
```

Listendefinition als Vorlage ohne Verweis auf weitere Listen

```
<List_Classes2_WERT1>\\
<entry value="Listeneintrag" text= $Primus.str oder .ustr />\\
<entry value="Fruit and Vegetable" text="$Primus.LST_FRUIT" />\\
</List_Classes2_WERT1>
```

Siehe auch

- [listbox](#)
- [combobox](#)

From:

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

Permanent link:

https://dokuwiki.unitec.de/doku.php?id=primus:hierarchische_listen

Last update: **2024/10/25 14:46**



icon

Ein **icon** ist ein [GUI Element](#) für [DataSheets](#), mit dem eine Grafik vom Dateityp ICO, basierend auf dem Inhalt eines Datenfelds, angezeigt werden kann. Der Eintrag erfolgt im Attribut **ctrltype**. Es muss dazu eine [Liste](#) definiert sein, bei der der Dateiname der Bilddatei im Attribut **icon** hinterlegt ist.

Zusätzliche Attribute

Für die der Auswahl zugrunde liegende Liste müssen folgende Attribute angegeben werden:

Attribut	Wert	Pflicht	Funktion
type	IconList	Ja	Wird stattdessen der Wert Text zugewiesen, dann verhält sich das Feld wie eine textbox
list	<LISTENNAME>	Ja	Definiert welche unter Lists definierte Liste zur Auswahl des Bilddateinamen im Attribut icon verwendet werden soll

Beispiel

Zeigt den Bearbeitungsstandort als Grafik (Landesflagge) im Datenblatt an.

```
<WriteLocation ctrltype="icon" data="WriteLocation"
label="$Primus.SHEET_WRITELOC" type="IconList" list="List_Location" mode="R"
/>
```

Die zugehörige Listendefinition:

```
<List_Location>
  <entry value="de" server="de-30-solidedge"
icon="Resources\Icons\flag_germany"/>
  <entry value="us" server="us-30-solidedge"
icon="Resources\Icons\flag_united_states"/>
</List_Location>
```

Siehe auch

- [combobox](#)
- [Listenmapping](#)

[DataSheet](#), [GUI](#), [ctrltype](#), [Liste](#)

From:

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

Permanent link:

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

Last update: **2023/06/16 14:47**



IgnoreStatus

Die Funktion **IgnoreStatus** überprüft ob das Dokument einen bestimmten [Status nicht](#) hat und kann z.B. in einem [Workflow](#) als Bedingung verwendet werden. Der Rückgabewert der Funktion ist

Wert	Bedeutung
TRUE	Das Dokument hat <u>nicht</u> den angegebenen Status
FALSE	Das Dokument hat den zu ignorierenden Status

Attribute

Der Befehl wird über folgende Attribute gesteuert.

Attribut	Wert	Funktion
condition	TRUE/FALSE	Es werden nur Dokumente geprüft, die diese Bedingung erfüllen
parameter	%FILEPATH%	Pfad zum zu prüfenden Dokument
	<FELDNAME>	Name des zu prüfenden Datenfelds
	<WERT>	Mit diesem Wert wird der Inhalt von <FELDNAME> verglichen. Soll mit mehreren Werten möglichen Werten verglichen werden, dann sind dafür eigene IgnoreStatus Prüfungen vorzusehen.

Die Übergabe von **parameter** werden durch ein | voneinander getrennt.

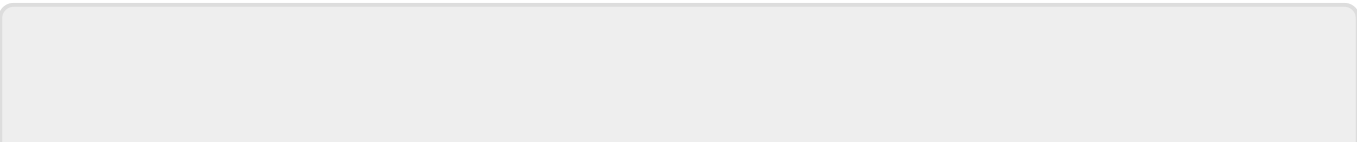
Beispiel

```
<!-- Aufruf der Funktion durch einen Workflow PreCall, ob das Dokument den
gesuchten Status nicht hat!-->
<Call method="IgnoreStatus" parameter="%FILEPATH%|WorkflowStatus|003"
condition="TRUE" />
<!-- Die zweite Prüfung wird nur benötigt wenn auf mehr als einen Status
geprüft werden soll-->
<Call method="IgnoreStatus" parameter="%FILEPATH%|WorkflowStatus|004"
condition="TRUE" />
```

Siehe auch

- [CheckChildrenStatus](#)
- [CheckParentsStatus](#)

[menu](#), [Befehlsliste](#), [workflow](#), [Bedingung](#)



From:

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

Permanent link:

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

Last update: **2023/05/22 13:19**



IncludeConfigLayers

Im optionalen [Abschnitt IncludeConfigLayers](#) können weitere sog. [CFGLAYER.XML](#) angegeben werden, deren Inhalt beim Start von Primus zur Konfiguration dazu geladen wird. Dadurch ist es möglich die CFG.XML Datei in mehrere Dateien aufzuteilen und diese mit einem Kommando beim Starten von Primus zur [PrimusSE.CFG.xml](#) dazu zu linkern. Dadurch ist es möglich auf einer Basiskonfiguration aufbauend z.B. unterschiedliche Menüs für CAD, Office oder Viewer Arbeitsplätze zu definieren. Allgemeine Änderungen, z.B. ein zusätzliches Datenfeld in der XML oder ein [pflichtfeld](#), sind dann automatisch in allen angepassten Konfigurationen enthalten.

Syntax

Der [Tag-Name](#) lautet **IncludeConfigLayers** und liegt in in der Konfiguration auf oberste Ebene direkt unter dem [Stammelement](#). Die Einträge sind [Leer-Tags](#) mit dem Namen **Layer** und dem [Attribut path](#). Das Attribut verweist auf die zu verlinkende [CFGLAYER.XML](#) Datei. Es können mehrere Layer-Tags definiert werden. Der Übersichtlichkeit halber erhalten diese Dateien die Endung **.CFGLAYER.XML** und werden im Verzeichnis **CFGLAYERS** abgelegt.

Beispiel

```
<IncludeConfigLayers>
  <Layer path=".\\CFGLayers\\Menue.CFGLAYER.xml" />
</IncludeConfigLayers>
```

CFGLAYER.XML

Die Dateien die zu einer bestehenden Konfiguration dazu gelinkt werden sollen, müssen den gleichen Aufbau wie eine [CFG.XML](#) haben. Zu Beginn steht der [XML-Header](#) und ein einziges Stammelement. Das Attribut **name** des Stammelements sollte in [.CFGLAYER.XML](#) weggelassen werden. Es wird sowieso nur berücksichtigt wenn das Attribut in der [PrimusSE.cfg.xml](#) leer gelassen wurde.

Beispiel

```
<?xml version="1.0" encoding="UTF-16"?> <!--XML Header -->
<PrimusFolderConfig> <!--Stammelement ohne das Attribut name="Beispiel"-->
  <!--Hier kommt der Inhalt rein-->
</PrimusFolderConfig>
```

Inhalt

In der **.CFGLAYER.XML** müssen die Inhalte exakt so abgelegt sein wie sie auch in der

PrimusSE.CFG.XML abgelegt wurden.

Beispiel

```
<?xml version="1.0" encoding="UTF-16"?>
<PrimusFolderConfig>
<GUI>
  <GUI_Data_Lists>
    <GUI_DocumentAttachments_List>
      <FileTypeIcon data="FILE_TYPE" label="$Primus.COL_FILE_TYPE_ICONLIST"
type="FileTypeIcon" list="List_FileTypes" position="0" width="23"
visible="1" />
      <FileType data="FILE_TYPE" label="$Primus.COL_FILETYPE" type="List"
list="List_FileTypes" position="1" width="88" visible="0" />
      <FileName data="%FILENAME%" label="$Primus.COL_FILENAME" type="Text"
position="2" width="100" visible="1" />
      <FileExt data="%EXTENSION%" label="$Primus.COL_FILE_EXT" type="Text"
position="3" width="29" visible="1" />
      <Title data="Title" label="$Primus.COL_TITLE" type="Text" position="4"
width="150" visible="1" />
      <Description data="Description" label="$Primus.COL_DESC" type="Text"
position="5" width="200" visible="1" />
    </GUI_DocumentAttachments_List>
  </GUI_Data_Lists>
</GUI>
</PrimusFolderConfig>
```

Konfigurations-XML

From:

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

Permanent link:

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

Last update: **2023/02/08 12:01**



IsActiveRegTab

Die [Funktion IsActiveRegTab](#) liefert den Namen des aktuell aktiven Registers der Elementdaten zurück. Mittels der Abfrage des aktiven Register können [Menübefehle bedingt](#) angezeigt bzw. aktiviert werden.

Mögliche Werte

Wert	Beschreibung
tabAttachments	Register mit allen Anhängen zu einem Dokument
tabChildRefs	Enthaltene Komponenten eines Dokuments z.B. Teile einer Baugruppe
tabHistory	Auflistung der Versionen eines Dokuments
tabParentRefs	Das Register Beziehungen
tabSubData	Klärungsbedarf
tabVirtFolders	Virtuelle Ordner in denen das Dokument enthalten ist

[Bedingung](#)

From:

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

Permanent link:

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

Last update:

2024/08/14 09:14



JobServer

Beim **JobServer** handelt es sich um ein [Addin](#) von Primus, mit dem Aufgaben zur Verarbeitung von Dokumenten an einen Rechner übergeben werden können. Dabei kann jeder Rechner selbst seine eigenen Jobs abarbeiten oder dies geschieht zentral auf einem oder mehreren Rechnern. Für jede JobServer Funktion wird eine eigene Lizenz, für den Rechner auf dem die Verarbeitung erfolgt, benötigt. Die einzelnen Funktionen sind abhängig vom CAD System dessen Daten verarbeitet werden sollen.

Primus eigene Methoden

Neben den Jobfunktionen die durch anwendungsspezifische Addins zur Verfügung gestellt werden, bietet auch Primus selbst Methoden die in Jobs benutzt werden können. Siehe dazu [Liste der Methoden des Primus Framework](#).

Lizenzierbare Module

Für [Solid Edge](#) sind folgende Module Lizenzierbar.

Modul	Funktion
Primus -JobServer	Für einen dedizierten JobServer-Rechner, Automatisierung für Druckaufträge, Konvertierungsjobs, Workflows
Drucken der Solid Edge-Zeichnungen (Druckaufträge)	Einstellbar: Drucker/Plotter, Format, Farbe/SW, Priorität, Ziel-JobServer (bei mehreren)
Drucken der Excel-Dateien (Druckaufträge)	Einstellbar: Drucker, Format, Farbe/SW, Priorität, Ziel-JobServer (bei mehreren)
DXF/DWG-Dateien aus Solid Edge-Zeichnungen erstellen	Zielordner und Dateiname, Priorität, Ziel-JobServer (bei mehreren)
PDF-Dateien aus Solid Edge-Zeichnungen erstellen	Zielordner und Dateiname, Multi-Sheet, Farbe/SW, Priorität, Ziel-JobServer (bei mehreren)
TIF-Dateien aus Solid Edge-Zeichnungen erstellen	Zielordner und Dateiname, Multi-Sheet, Farbe/SW, Priorität, Ziel-JobServer (bei mehreren)
3D PDF-Dateien aus Solid Edge-Modellen erstellen	Zielordner und Dateiname, Priorität, Ziel-JobServer (bei mehreren)
DXF-Abwicklungen aus Solid Edge-Blechmodellen erstellen	Zielordner und Dateiname, Priorität, Ziel-JobServer (bei mehreren)
3D-Export aus Solid Edge-Modellen erstellen	STEP, IGES, Parasolid-Einstellbar: Zielordner und Dateiname, Priorität, Ziel-JobServer (bei mehreren)
Schriftkopftausch für Solid Edge-Zeichnungen	Einstellbar: Neue Vorlage, Priorität, Ziel-JobServer (bei mehreren)
Update Properties für Solid Edge-Zeichnungen	Einstellbar: Ansichten, Ziel-JobServer (bei mehreren)

- Alle JobServer-Module sind entweder pro JobServer (dedizierter Rechner mit CAD und Primus Installation) oder pro Benutzerrechner zu lizenzieren.

Einrichtung in Primus

Die JobServer [DLL](#) muß in der Primus [Konfiguration](#) unter [Addins](#) eingetragen werden.

Beispiel

```
<SEJob_Export dll=".\\Resources\\AddIns\\Primus_Export_SE_Job"  
class="Primus_Export_SE_Job.ExportAddIn" loadonstart="1"/>
```

Siehe auch

- [SEJob_Export](#)
- [PrimusJobServerService](#)
- [CreateJob_StatusPostCalls](#)
- [Liste der Methoden des Primus Framework](#)

[Addin](#), [JobServer](#)

From:

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

Permanent link:

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

Last update: **2024/09/09 12:02**



Kopieren

Dieser Befehl wird verwendet um Einzelteile oder einzelne Dokumente zu kopieren.



From:

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

Permanent link:

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

Last update: **2023/02/08 12:01**



Kopieren mit Komponenten

Dieser Befehl wird benutzt um Baugruppen und die beinhalteten Einzelteile zu kopieren. Innerhalb der Session können Einzelteile aus dem Vorgang ausgeschlossen werden.



From:

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

Permanent link:

https://dokuwiki.unitec.de/doku.php?id=primus:kopieren_mit_komponenten

Last update: **2023/02/08 12:01**



Kopieren mit Komponenten und Zeichnungen

Dieser Befehl wird verwendet um die gesamte Baugruppe mit Einzelteilen und allen Zeichnungen zu kopieren. Wählen Sie zum Kopieren immer am besten die oberste Baugruppe, sodass alle verlinkten Zeichnungen und Modelle mitkopiert werden.



From:

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

Permanent link:

https://dokuwiki.unitec.de/doku.php?id=primus:kopieren_mit_komponenten_und_zeichnungen

Last update: **2023/02/08 12:01**



label

Mit dem Wert **label** für das [Attribut ctrltype](#) kann man Beschriftungen in [DataSheets](#) definieren.

Zugehörige Attribute

Um ein Label in einem Formular zu definieren müssen neben dem Attribut **ctrltype** noch weitere Attribute ausgefüllt sein. Prinzipiell können alle Attribute aus DataSheets verwendet werden.

Attribut	Pflicht	Funktion
type	Ja	Muss text sein
height	Nein	Darf nicht zu klein sein, sonst ist das Label nicht zu sehen und die Beschriftung wird nicht vollständig angezeigt
label	Ja	Enthält den Beschriftungstext
left		Beliebiger Wert, wird aber benötigt sonst wird top ignoriert. Die Position wird mit labelposleft definiert
top		Abstand vom oberen Rand in Pixel
width		Beliebiger Wert, wird aber benötigt sonst wird top ignoriert
labelposleft		Textabstand vom linken Rand in Pixel

Beispiel

```
<Rev_Text ctrltype="label" label="-Revision" type="Text" left="148" top="19" width="100" height="20" labelposleft="15" />
```

[gui](#), [datasheet](#), [ctrltype](#)

From:

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

Permanent link:

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

Last update:

2023/02/08 12:01



listbox

Eine **listbox** ist ein [GUI Element](#) für [DataSheets](#), das dem Anwender eine [Auswahlliste](#) mit möglichen Feldwerten zur Verfügung stellt. Der Feldinhalt ist nur auf die zur Auswahl stehenden Listeneinträge begrenzt. Soll der Anwender neben der Auswahl von Werten auch eigene Eingaben machen können, dann ist eine [Combobox](#) zu verwenden. Der Eintrag erfolgt im Attribut **ctrltype**.

Zusätzliche Attribute

Für die der Auswahl zugrunde liegende Liste müssen folgende Attribute angegeben werden:

Attribut	Wert	Pflicht	Funktion
type	List	Ja	Wird stattdessen der Wert Text zugewiesen, dann verhält sich das Feld wie eine textbox
list	<LISTENNAME>	Ja	Definiert welche unter Lists definierte Liste zur Auswahl gestellt werden soll
listchildren	<LISTENNAME>	Optional	Definiert welches Feld <u>desselben</u> DataSheets als Unterauswahl verwendet werden soll. Dieses Feld wird bei Auswahländerung aktualisiert.

Beispiel

```
<Material ctrltype="listbox" />
```

Siehe auch

- [checkedlistbox](#)
- [combobox](#)
- [Hierarchische Listen](#)
- [Listenmapping](#)

[DataSheet](#), [GUI](#), [ctrltype](#), [Liste](#)

From:

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

Permanent link:

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

Last update: **2024/11/11 14:04**



Liste aller Platzhalter

In der **Liste aller Platzhalter** werden alle Platzhalter aufgelistet die **Primus** zur Laufzeit zur Verfügung stellt und die für **Menübefehle** und **Workflows** genutzt werden können. Die Platzhalter sind immer in Prozentzeichen (%) eingefasst.

Name des Platzhalters	Beschreibung
%CALLLEVEL%	
%CHILD_EXTENSION%	Dateiendungen inkl. Punkt davor der enthaltenen Komponenten
%CHILD_FILENAME%	Dateinamen ohne Dateiendungen der enthaltenen Komponenten
%CHILD_FILEPATH%	Dateinamen mit Pfad und Dateiendung der enthaltenen Komponenten
%CLIPBOARD	Inhalt der Zwischenablage
%COUNTER%	
%COUNTER_PREVIEW%	
%DIM%	
%DIRNAME%	Name des aktuell gewählten Verzeichnisses, nicht des kompletten Pfads
%DIRPATH%	Aktueller Pfad der in Primus Pfadliste derzeit angewählt ist
%EXTENSION%	Dateiendung inkl. Punkt davor
%FILEMODIFIED%	
%FILENAME%	Dateiname ohne Dateiendung
%FILENAME_NOREV%	
%FILEPATH%	Dateiname mit Pfad und Dateiendung
%FILESIZE%	
%GUIBYPARAMETER%	
%ignore%	
%LOCATION%	Die Location die im Attribut location des Tags ReplicationService definiert ist
%NEWGUID%	Erzeugt eine neue GUID , wird z.B. für die RepID verwendet
%NewNamePrefix%	
%NewNameSuffix%	
%PARENT_EXTENSION%	Dateiendung inkl. Punkt davor eines Parentdokuments (nur in bestimmten Funktionen verwendbar)
%PARENT_FILENAME%	Dateiname ohne Dateiendung eines ...
%PARENT_FILEPATH%	Dateiname mit Pfad und Dateiendung eines ...
%pos%	
%predata:ISLINSERTNAME%	
%PREVVER_FILEPATH%	Dateiname mit Pfad und Dateiendung zur Vorgängerversion
%quantity_overwrite%	
%quantity_user%	
%quantity%	Menge in der Stückliste
%REVISION%	
%SmartCopyCounter%	
%SmartCopyPrefix%	

Name des Platzhalters	Beschreibung
%SmartCopyStatus%	
%SmartCopySuffix%	
%SmartCopyTarget%	
%SOURCETOPDIRPATH%	
%state%	
%TODAY%	Datum von heute
%USER%	Name des angemeldeten Benutzers
%XML_Document.Ref1:Class1%	
%XML_Document.Ref1:Class2%	
%XML_Document.Ref1:Class3%	
%XML_Document.Ref1:Description%	
%XML_Document.Ref1:Dimension%	
%XML_Document.Ref1:Material%	
%XML_Document.Ref1:Note%	
%XML_Document.Ref1:Number%	
%XML_Document.Ref1:Title%	
%XML_Document.Ref1:Weight%	
%XML_Project:EMailTo%	
%XML_Project:FileNameFormat%	
%XML_VirtualFolder:Class%	

Liste, Klärungsbedarf

From:

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

Permanent link:

https://dokuwiki.unitec.de/doku.php?id=primus:liste_aller_platzhalter

Last update:

2024/07/09 10:56



Liste der Systemdefinierten Datenfelder

In der **Liste der Systemdefinierten Datenfelder** werden Datenfelder aufgelistet, die von **Primus** Funktionen intern benutzt werden und deren Benennung und Inhalte vorgegeben sind. Diese Feldnamen dürfen nicht für andere Zwecke genutzt werden.

Name	Bedeutung
DelUser	Benutzer der ein Dokument gelöscht und in den Papierkorb geschoben hat.
FILE_TYPE	Dateityp zu dem das Dokument gehört
MODIFIED	Gibt an ob Daten eines Dokument in Primus modifiziert aber noch nicht ins Dokument übertragen worden sind.
RepID	Eindeutige GUID für die Replikation
Replicate	Gibt an ob ein Verzeichnis repliziert werden soll.
ResByAddIn	Gibt an ob die Reservierung manuell oder automatisch gesetzt wurde. Nur automatische Reservierung werden auch automatisch zurückgenommen
Reservation	Name des Anwenders der ein Dokument reserviert hat. Wird von jeden Reservierungsvorgang geschrieben. CADAdaptApproved
WorkflowStatus	Der WorkflowStatus eines Dokuments
WriteLocation	Standort an dem das replizierte Dokument bearbeitet werden darf.

Siehe auch

- [Liste aller Platzhalter](#)
- [Liste der Systemdefinierten Platzhalter](#)

[Liste](#)

From:

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

Permanent link:

https://dokuwiki.unitec.de/doku.php?id=primus:list_der_systemdefinierten_datenfelder

Last update:

2025/03/17 09:33



Systemdefinierte Platzhalter

In dieser Liste werden die in Primus und den von [CADAdapt](#) gelieferten [Primus_Addins](#), vordefinierten Platzhalter für [Elementtexte](#) aufgelistet. Der Inhalt dieser Texte kann vom Anwender angepasst und auch übersetzt werden, die Bezeichnung des jeweiligen Platzhalters ist jedoch fest im Programm hinterlegt. Benutzerdefinierten Änderungen sollten jedoch nicht in der jeweiligen [STR-Datei](#) sondern in der extra dafür vorgesehenen USTR-Datei vorgenommen werden.

Liste der vordefinierten Platzhalter

Die Liste enthält verschiedene Abschnitte für die einzelnen [Sprachdateien](#).

Platzhalter	Funktion
Primus.str	
FRM_RENFILECAPT	prmAttachmentRename
	prmElementsRename
	secreaterenamejob
MSG_FILEWRPROT_CANNOTREN	seCreateRenameJob
MSG_SELCHILDSTATDIF	CheckChildrenStatus
MSG_STATREVERSE0NERROR	seUpdatePropertiesByPath
PrimusSE.str	
MSG_YES2REPLALLOCCURS	seReplace

[Liste](#), [Sprachabhängig](#), [Hardcoded](#)

From:

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

Permanent link:

https://dokuwiki.unitec.de/doku.php?id=primus:liste_der_systemdefinierten_platzhalter

Last update:

2025/07/24 07:46



Listenmapping

Hier in **Listenmapping** wird beschrieben wie und wo [Listen](#) und ihre Einträge genutzt werden können.

Formularen

In Formularen können Listen genutzt werden in dem Felder mit dem [ctrltype](#) [listbox](#) oder [combobox](#) verwendet werden. Dazu muß das Attribut [type](#)=„list“ sein und [list](#) muß auf die Liste verweisen. Die Verwendung von [listchildren](#) ist optional. Hinter dem Namen der Liste kann mit : (Doppelpunkt) ein bestimmtes Attribut des Listeneintrags gewählt werden, ist nichts angegeben dann wird text verwendet. Soll in einer Wertzuweisung in [Preassign](#) mit einem Listeneintrag erfolgen, dann kann [GetListTextByParam](#) dazu verwendet werden.

Beispiel

Im Beispiel wird statt des Benutzernamens (USER), das zum Namen zugehörige [Attribut](#) initialen aus der Liste List_User verwendet.

```
<Sample ctrltype="textbox" data="Sample" preassign="CALL(GetListTextByParam
List_User:initialen|%USER%)" />
```

Listenansichten

In Listenansichten muß das Attribut [type](#)=„list“ sein und [list](#) muß auf die Liste verweisen. Die Verwendung von [listchildren](#) ist optional. Hinter dem Namen der Liste kann mit : (Doppelpunkt) ein bestimmtes Attribut des Listeneintrags gewählt werden, ist nichts angegeben dann wird text verwendet.

Beispiel

Im Beispiel werden in der GUI_Document_List statt des Loginnamens des Benutzers, der im Feld CheckedName gespeichert ist, die in List_User zugeordneten Initialen angezeigt.

```
<CheckedName data="CheckedName" label="$Primus.COL_APPROVED_NAME"
type="List" list="List_User:initialen" position="16" width="104" visible="1"
/>
```

AppAttributes

Bei der Wertzuweisung in den INI-Dateien von [AppAttributes](#) können Listeneinträge mit [GetListText](#)

oder `GetListTextFromDict` zugewiesen werden. Die Definition welche Liste zu verwenden ist, wird dabei aus der `GUI_Document_List` der `CFG.XML` genommen. Abweichende Attribute, der Default ist `text`, für die Ersetzung müssen hier jedoch mit einem Pipe (|) und nicht mit einem Doppelpunkt abgetrennt werden⁸⁾.

Beispiel

Die Zuweisung `CheckedName=CALL(GetListText CheckedName)` verwendet die Definition aus folgender Definition:

```
<CheckedName data="CheckedName" type="List" list="List_User:initialen"
position="16" width="104" visible="1" label="$Primus.COL_APPROVED_NAME" />
```

Liste

⁸⁾

Dies ist eine Fehlinterpretation aus der Anfangszeit

From:

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

Permanent link:

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

Last update: **2024/09/05 08:00**



Lists

Im [Abschnitt Lists](#) werden Listen definiert, die in Primus für verschiedene Zwecke genutzt werden können. Da wäre die Möglichkeit mit [combobox](#) oder [listbox](#) in einem Formular Auswahllisten anzuzeigen, oder in Listenansichten um statt der Dateieindung das zugehörige [Icon](#) darzustellen.

Namenskonvention für Listen

Der Name der Liste ist, von Hardcoded Listen abgesehen, frei wählbar, muss aber eindeutig sein. Es wird aber empfohlen Listen nach dem Schema `List_<LISTENINHALT>` aufzubauen. Dadurch bleibt die Konfiguration für den Administrator übersichtlich und lesbar.

Standardlisten in Primus

In Primus sind von Haus aus einige Listen standardmäßig enthalten. Diese Listen sind aber nicht zwingend erforderlich und die Listeneinträge können auch geändert werden.

Liste	Hardcoded	Zweck
List_WorkflowStatus	x	Enthält die Benennung, Symbole und Farben für die Status die ein Dokument haben kann
List_WriteStatus		Enthält den Text und das Symbol für schreibbar / schreibgeschützt
List_MODIFIED		Enthält Text und Symbol für den Modified Status
List_FileTypes	x	Enthält die Benennungen und Symbole für die verwalteten Dokumenttypen
List_FileTemp_docx		Klärungsbedarf
List_BOM_State		Enthält den Text für die verschiedenen Stücklisten Status eines Stücklisteneintrags
List_User		Enthält die Loginnamen der Benutzer sowie deren vollständiger Name, eine Kurzform und z.B. die Initialen
List_Material	x	Liste der Werkstoffe die im CAD System definiert sind, der Abgleich erfolgt über eine Primusfunktion

Für die Hardcoded Listen ist der Name von Primus vorgegeben und kann nicht frei bestimmt werden.

Benutzerdefinierte Listen

Es können unternehmensbezogen weitere Listen nach Bedarf konfiguriert werden. Es empfiehlt sich, diese Ergänzungen in der [Lists.xml](#) einzutragen. Die `LISTS.xml` wird mit Attribut [include](#) in den Abschnitt **Lists** eingebettet. Jede benutzerdefinierte Liste muß einen `<Lists>` Abschnitt haben. Es ist sinnvoll Listen auf mehrere Dateien aufzuteilen.

Listeneinträge

Listeneinträge bestehen immer aus einem [Leer-Tag](#) mit dem Namen **entry**. Das [Attribut value](#) ist erforderlich, dazu mindestens ein weiteres Attribut, i.d.R. **text**, mit dem der zugehörige Listeneintrag verknüpft ist. Es können aber beliebige weitere Attribute definiert werden. Das Attribut `childlist` wird für [Hierarchische Listen](#) benötigt.

Beispiel

Hier im Beispiel wird zusätzlich das Attribut `icon` definiert, welches den Pfad zum zugehörigen Icon angibt.

```
<Lists>
  <List_ProjectTypeIcon>
    <entry value="PROJNODE" text="Projektknoten"
icon="Resources\Icons\project" />
    <entry value="CAD" text="CAD" icon="Resources\Icons\cad" />
    <entry value="DOC" text="Dokumentation" icon="Resources\Icons\office"
/>>
    <entry value="FOLDER" text="Ordner" icon="Resources\Icons\folder" />
    <entry value="STD" text="Standard" icon="Resources\Icons\stdpart" />
  </List_ProjectTypeIcon>
</Lists>
```

include

Mit dem Attribut **include** ist es möglich weitere Listen die in separaten Dateien gespeichert sind, beim Start von Primus dazu zu laden. Es können mehrere Listendateien angegeben werden, diese müssen dann mit einem `|` getrennt werden. Es ist möglich absolute Pfade oder relative Pfade anzugeben, bei letzterem muß ggf. ein `.` vorangestellt werden. Der Vorteil einer absoluten Pfadangabe die auf den Server verweist ist, das Änderungen an einer Liste beim nächsten Start von Primus unmittelbar von den Clients übernommen werden, ohne das ein Update des Primus Clients ausgeführt werden muss. Das `include` Attribut kann in den Listendateien ebenfalls verwendet werden, diese Verschachtelung ist aber zu vermeiden, da beim entfernen einer Liste, so evtl. weitere Listen auch nicht mehr geladen werden.

```
<Lists
include=".\\Resources\\LISTS.xml|\\primus\\PrimusSetup\\Resources\\SE_MATERIAL.LI
ST.xml|.\\Resources\\Lists_Kunde.xml|\\primus\\PrimusSetup\\Resources\\List_Users
.xml">
```

Siehe auch

- [Listenmapping](#)
- [Hierarchische Listen](#)

Liste, Hardcoded

From:

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

Permanent link:

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

Last update: **2024/10/25 14:45**



ListViewSetting

Der Abschnitt ListViewSetting besteht aus einem [einzigen Eintrag](#). Mit diesem Abschnitt wird die Listenansicht, die hinter oder unter der [GUI_Document_List](#) angezeigt wird, definiert. Folgende [Attribute](#) können angegeben werden. Der Anwender kann die Werte im Primus Menü **Ansicht** unter **Listen-Ansicht** bzw. **Listen-Teiler horizontal** verändern.

Attribut	Werte	Funktion
splitterdist	<ZAHL>	Abstand in Pixel vom linken, rechten, oberen oder unteren ⁹⁾ Rand der dokumentenliste
mode	0	Inaktiv
	1	Namen
	2	Dateien
	3	Vorschau
showbehinddatalist	0	Die Anzeige erfolgt links der Dokumentenliste
	1	Die Anzeige erfolgt rechts der Dokumentenliste
splitterorient		Hiermit wird bestimmt ob der Trenner horizontal oder vertikal angeordnet werden soll
	Ver	für vertikal
	Hor	für horizontal

Defaultwerte in **Fett**

Beispiel

```
<ListViewSettings splitterdist="376" mode="0" showbehinddatalist="1"
splitterorient="Ver" />
```

[gui](#)

⁹⁾

abhängig von showbehinddatalist und splitterorient

From:

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

Permanent link:

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

Last update: **2023/02/08 12:01**



Menübefehle

Für den Aufbau der [Menüs](#) stellt das Primus Framework Befehle zur Verfügung. Die folgende Auflistung ist Alphabetisch sortiert.

Allgemeine Attribute

Für die Menübefehle können / müssen folgende [Attribute](#) angegeben werden.

Attribut	Pflicht	Wert	Funktion
addin	X ¹⁰⁾	Addin	Name des Addin das den Befehl zur Verfügung stellt
enabled		TRUE/FALSE	Gibt an ob der Befehl aktiv ist
icon		Dateipfad	Gibt den Pfad zum Icon ohne die Dateiendung <code>.ico</code> an
parameter	X ¹¹⁾		Übergibt Parameter an den Befehl die seine Funktion beeinflussen
text		Text	Der der für diesen Befehl im Menü angezeigt wird, die Verwendung von Elementtexten wird empfohlen
tooltip			Ein beschreibender Text, der angezeigt wird wenn die Maus auf dem Menüeintrag platziert wird, die Verwendung von Elementtexten wird empfohlen
visible		TRUE/FALSE	Gibt an ob der Befehl im Menü sichtbar ist

Allgemeine Menüfunktionen

Für den Aufbau und die Strukturierung von Menüs stehen folgende Befehle zur Verfügung.

Befehl	Funktion
SEPARATOR	Erzeugt eine Trennlinie innerhalb des Kontextmenüs
SUBMENU	Bindet ein separat definiertes Untermenü ein
INCLUDE	Bindet die separat definierten Einträge direkt in das Kontextmenü ein

Befehle des Primus Framework

Funktion	Beschreibung
prmallwithbrokenlinks	Listet alle Dateien mit unterbrochenen Verknüpfungen auf
prmallwithreiteranname	Listet alle Dateien mit wiederholenden Namen auf
prmattachementcopy	Anhang kopieren
prmattachementcut	Anhang ausschneiden
prmattachementdelete	Anhang löschen
prmattachementnew	Erzeugt einen neuen Anhang, dazu wird das angegebene DataSheet angezeigt.
prmattachementnewcopy	Kopiert einen Anhang, dazu wird das angegebene DataSheet angezeigt.
prmattachementopen	Öffnet den gewählten Anhang

Funktion	Beschreibung
prmattachmentpaste	Fügt einen Anhang ein
prmattachmentrename	Anhang umbenennen
prmElementsChangeValueByDlg	Datenfeld über einen Dialog ändern
prmClearRecyclebin	Papierkorb leeren
prmCopyDocData	Kopiert alle Primus Eigenschaften in die Zwischenablage
prmcopyfieldvalues	Kopiert die Primus Datenfelder der gewählten Datei in die Zwischenablage
prmcreateallnotexistingxmls	Fehlende XMLs anlegen oder Ungenutzte / verwaiste XML-Dateien entfernen
prmDeleteAllPrimusXMLs	Löscht alle von Primus erzeugten XML Dateien
prmDeleteUnusedXMLs	Löscht alle Primus XML zu denen es kein Dokument mehr gibt
prmelementsactivatevirtlink	Kläarungsbedarf
prmelementsattachexternfile	Datei ersetzen durch Externe
prmelementsattachmentstoclipboard	Kopiert alle oder bestimmte Anhänge der gewählten Datei.
prmelementschange filetype	FileType ändern
prmelementscheckinlocal	Lokal ausgecheckte Dateien einchecken
prmelementscheckoutlocal	Check Elemente lokal aus, kann im Kontext des lokalen Zielordners oder des Quellordners auf dem Server aufgerufen werden.
prmelementscopy	Element kopieren (Zwischenablage)
prmelementscopylocalto server	Einchecken von lokalen Dokumenten
prmelementscopypath	Dateipfad kopieren
prmelementscut	Element ausschneiden (Zwischenablage)
prmelementsdelete	Element löschen
prmelementsdeletedoclink	Löschen fehlerhafter Dokumentverknüpfungen Klärungsbedarf
prmelementsdeletewirtlink	Verknüpfung Löschen
prmElementsEditReadOnlyDataSheet	Datenblatt zum Ändern von Eigenschaften aufrufen
prmelementsexportfile	Element exportieren
prmelementsfilestoclipboard	Kopiert Elemente in die Zwischenablage
prmElementsGenerateNewDocumentName	Generiert einen neuen Dateinamen und legt ihn in die Zwischenablage
prmelementsjobserverstart	Startet den Jobserver Prozess
prmelementsjobserverstop	Stoppt den Jobserver Prozess
prmelementsnew	Neues Element anlegen
prmelementsnewcopy	Erstellt eine neue (?) Kopie eines Elements
prmElementsNewVersion	Neue Version erstellen
prmelementsopen	Element öffnen
prmelementspaste	Element einfügen (Zwischenablage)
prmelementspastewirtlink	ElementLink in einen Virtuellen Ordner einfügen (Zwischenablage).
prmelementsp rintattachments	Druckt Anhänge
prmElementsRemoveNewestVersion	Löscht das gewählte Dokument und setzt die Vorversion zurück auf freigegeben
prmElementsRename	Element umbenennen

Funktion	Beschreibung
prmElementsRenameMsgByParam	Ändert die Endung einer Messagedatei und markiert sie so als erledigt
prmElementsReplaceByNewestVersion	Durch neueste Version ersetzen
prmelementsrestartjob	Neustart des Jobserver / Job
prmelementsrestore	Element€ aus dem Papierkorb wiederherstellen.
prmelementsshowjoblogs	Zeigt die Jobserver Logs an.
prmelementsstatus	Status ändern
prmelementssynchroniselocal	Synchronisieren
prmExecuteNamedCalls	Ruft einen Named Call auf
prmFindCorruptXMLDataFiles	Findet beschädigte Daten-XML
prmfindextended	Erweiterte Suche
prmfindfilebypath	Dateipfad suchen
prmfindinallfields	Suchen
prmfindnextinallfields	Weiter suchen F3
prmfoldersettings	Ordner Einstellungen
prmlocalworkmode	Schaltet zwischen Online, Offline und Serverarbeitsweise um.
prmopenexplorer	Öffnet den Explorer für das gewählte Element
prmpartsliststructpositions	Stückliste Autopositionen
prmpartsliststructpositionscheck	Stückliste auf doppelte Positionen prüfen
prmPasteDocData	Schreibt die zuvor mit prmCopyDocData kopierten Daten in die ausgewählten Datenfelder.
prmpastefieldvalues	(Feld-)Daten einfügen (Zwischenablage)
prmprojectassignallchildvaluesfromparent	Aktualisiere Unterordnern
prmprojectcheckinlocal	Einchecken und bereinigen
prmprojectcheckoutlocal	Lokal auschecken
prmprojectscreatebytemplate	Struktur anlegen mit Vorlage
prmprojectsdelete	Löscht in Element aus dem Projektbaum
prmprojectsedit	Projekt bearbeiten
prmProjectsFindByData	Suchen nach Verzeichnissen anhand ihrer Eigenschaften
prmprojectsfindbyname	Suchen nach Projekt-/Ordernamen
prmprojectslistallsubprojects	Alle Unterprojekte auflisten
prmprojectsnew	Neuer Projektknoten
prmprojectremoveallunusedparentrefs	Entfernt alle unbenutzten Parent Referenzen
prmprojectsresetnode	Liest die Ordner Struktur unterhalb des gewählten Ordners neu ein
prmProjectsUploadXmIsToDB	Suchindex aktualisieren
prmprojectsynchroniselocal	Projekt synchronisieren
prmProjectSynchroniseAllDataFields	Synchronisiert alle Datenfelder der XMLs untereinander
prmProjectUpdateAllRefs	Aktualisiert alle CAD-Referenzen
prmprojectvirtfolderactivate	Referenzelement aktivieren
prmprojectvirtfolderdelete	Virtuelles Element löschen
prmprojectvirtfolderrename	Virtuelles Element umbenennen
prmrefelementssetactive	Element aktivieren, Gehe zu Ordner
prmRefElementsSetRef1	Gewähltes Element als Ref1 setzen

Funktion	Beschreibung
prmreloadcfg	PrimusSE.CFG.XML neu einlesen, lädt die aktuelle Primus Konfiguration, Strings werden nicht neu eingelesen
prmRemoveWriteAccessCall	Klärungsbedarf
prmSetReservation	Reservierung setzen / aufheben
prmshowdocumentswithsamename	Elemente Dokumenten mit gleichen Namen auflisten
prmshowpartslist	Stückliste Model
prmshowpartsliststruct	Stückliste Summe
prmShowWhereUses	Verwendungsnachweis
prmshowwhereuses_allcomponentsbyext	Listet alle zugehörigen Komponenten eines bestimmten Dateityps auf.
prmshowwhereuses_alldrawings	Alle Zeichnungen der Komponenten
prmshowwhereuses_allwithoutdrawings	Alle Komponenten ohne Zeichnung
prmshowwhereuses_componentsdrawings	Alle Komponenten und Zeichnungen
prmStartProcess	Aufruf eines externen Programms über einen Menüpunkt
prmTestFAMServiceConnection	Klärungsbedarf
prmTestWriteAccessCall	
prmUpdateParentRef1Data	Bei Änderungen eines Eingabefelds wird das Feld des zugehörigen Parentelements mit aktualisiert. Automatisches Aktualisieren bei Änderungen in einem Eingabefeld
prmUpdateXMLFieldsAndParentRefs	Alle XML Dateien für den gewählten Projektknoten aktualisieren.
prmUseDBDataActive	Wechselt zwischen DB- und XML-Modus (ab Primus2025)

Addins

Zu den vom Primus Framework bereitgestellten Befehlen und Funktionen, kommen noch Befehle dazu die durch [Addins](#) bereitgestellt werden. Um einen solchen Befehl aufzurufen muss der [Tag-Name](#) der Name des Befehls sein, das [Attribut addin](#) ist erforderlich und muss den Namen mit dem das Addins von Primus geladen wird haben.

Beispiel

```
<seCopySEMaterialsToCFG addin="SEInterface" visible="TRUE" enabled="TRUE"
parameter=".\\Resources\\SE_MATERIAL.LIST.xml|Material"
text="$Primus.MNU_COPY_SE_MATERIAL_TOCFG"
icon="Resources\\AppInterfaces\\SolidEdge\\Icons\\segetproperties" />
```

Menu, GUI, Befehlsliste, Liste

10)

Für Befehle aus Addins ein Pflichtfeld

11)

abhängig vom Befehl

From:

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

Permanent link:

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

Last update: **2025/12/04 07:36**



method

Bei **method** handelt es sich um ein [Attribut](#) der Tags mit dem [Tag-Namen](#) `call`. Damit werden Funktionen die in [Workflows](#) genutzt werden können, zur Verfügung gestellt. Die von einer Funktion benötigten Parameter werden dem Attribut [parameter](#) zugewiesen.

Liste der Methoden des Primus Framework

Die aufgelisteten Parameter müssen durch | getrennt werden.

Name	Parameter	Funktion
CheckChildrenStatus	Dokument, XML-Feld, Wert(e)	Prüft ob ein bestimmtes Dokument oder eine Liste von Dokumenten, in dem XML-Feld den gesuchten Inhalt haben
CheckChildrenWriteable	<DOKUMENT>, <FELDNAME>, <WERT>, ?, ?	Überprüft ob die zu einem Dokument zugehörigen Dokumente, beschreibbar sind
CheckParentsStatus	<DOKUMENT>, <FELDNAME>, <WERT>, <BEDINGUNG>, <MELDUNG>	Überprüft ob die auf ein Dokument verlinkenden Dokumente, einen bestimmten Status haben
CheckParentsWriteable	<DOKUMENT>, <FELDNAME>, <WERT>, <BEDINGUNG>, <MELDUNGSTEXT>	Überprüft ob die übergeordneten Dokumente eines Dokument, beschreibbar sind
CheckXMLValue	<DOKUMENT>, <BEDINGUNG>, <MELDUNG>	Prüft zu einem Dokument ob eine Bedingung erfüllt ist und gibt eine Meldung aus
CopyFile	<QUELLE>, <ZIEL>	Kopiert eine Datei
CreateJob_StatusPostCalls	JobServer Pfad	Erzeugt einen Job der mehrere PostCall Vorgänge zusammenfasst und dann zentral auf dem Jobserver ausgeführt werden kann
IgnoreStatus	>DOKUMENT>, <FELDNAME>, <WERT>	Überprüft ob das Dokument einen bestimmten Status hat
SetChildrenToStatus	Dokument, XML-Feld, Wert, Parameter	Setzt den Status aller Children-Elemente auf den gesuchten Status wenn sie diesen noch nicht haben
SetParentToStatus	Dokument, XML-Feld, Wert(e)	Dadurch wird für alle SE-Modelle die Parent „.dft“-Datei, die nicht auf einem der WorkflowStatus .003.004.005. sind, automatisch Status 003 gesetzt
SetPrevVersionToStatus	Dokument, XML-Feld, Wert	Weist der vorhergehenden Version einen Status zu
SetSubFilesToStatus	Dokument, XML-Feld, zu prüfende Status, Dateieindungen	Setzt den Status von zusätzlichen Dokumenten, die im Rahmen eines Workflows erzeugt werden
SetXMLValue	<DOKUMENT>, <DATENFELD>, <WERT>	Weist einem Datenfeld einen Wert zu

Externe Funktionen

Mit dem Attribut **method** können auch externe Funktionen, z.B. von [Addins](#) bereitgestellte, aufgerufen werden. Dazu wird dem Attribut ein Wert nach dem Schema **AddInName.Funktion** zugewiesen.

```
<Call method="SEInterface.seUpdatePropertiesByPath" parameter="%FILEPATH%|1"
condition="$(Condition.isSEModel)" />
```

[Workflow](#), [Attribut](#), [Methode](#), [Liste](#)

From:

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

Permanent link:

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

Last update: **2024/03/12 12:15**



NamedCalls

Im [Abschnitt NamedCalls](#), der in [Options](#) definiert sein muss, werden die benutzerdefinierten Funktionsaufrufe (Named Call) definiert. Bei einem Named Call handelt es sich um eine Zusammenfassung mehrerer [Calls](#) unter einem Namen, die dann über einen [Menübefehl](#) oder [Button](#) aufgerufen und in der definierten Reihenfolge abgearbeitet werden. Der Name z.B. `<OldVerCall>` ist frei definierbar. Es können beliebig viele Call-Listen mit verschiedenen Namen definiert werden. Diese können mit der Funktion [prmExecuteNamedCalls](#) aufgerufen werden. Auch aus dem Menü heraus. Die Funktion `prmExecuteNamedCalls` kann auch innerhalb eines Named Calls verwendet werden um einen anderen Named Call aufzurufen.

Beispiel

Im Beispiel wird der Named Call `OldVerCall` definiert. Es wird zunächst der Status des Documents geändert und anschließend falls es eine Solid Edge Draft ist, ein PDF davon erzeugt.

```
<NamedCalls>
  <OldVerCall>
    <Call method="StatusChangeByFilePath"
parameter="%FILEPATH%|StatusWorkflow_Settings|004"
condition="$(Condition.isSEDocument)" />
    <Call method="SEJob_Export.seExportJobByPath"
parameter="%FILEPATH%|C300|PDF|.\\Attachments\\%FILENAME%%EXTENSION%\\%FILENAME
%|XML_DocumentAttachments" condition="$(Condition.isSEDraft)" />
  </OldVerCall>
</NamedCalls>
```

Beispiel eines Menüaufrufs

```
<prmExecuteNamedCalls visible="TRUE" enabled="TRUE" parameter="|OldVerCall"
text="ExecuteNamedCalls: OldVerCall" icon="" />
```

Siehe auch

- [prmExecuteNamedCalls](#)
- [StatusWorkflow_Settings](#)
- [ProcessCalls](#)

[Workflow](#), [Menübefehl](#)

From:

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

Permanent link:

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

Last update: **2024/03/12 13:25**



Notepad++

Notepad++ ist ein Texteditor der unter der GPL 2 lizenziert ist. Notepad++ hat sich als Editor für alle Konfigurationsdateien von Primus bewährt. Notepad++ kann durch sog. Plugins erweitert werden. Für die Arbeit mit Primus empfiehlt es sich folgende Plugins zu installieren.

- **XML-Tools:** Ein nützliches Werkzeug beim editieren von [xml](#) Dateien. Es wird u.a. auch eine Syntaxprüfung der XML Struktur beim Speichern durchgeführt.
- **Compare:** Hilft beim Vergleichen von Dateien. Es ist hilfreich wenn man Änderungen zwischen verschiedenen Konfigurationsentwürfen vergleichen will.

Weblinks

- <https://de.wikipedia.org/wiki/Notepad%2B%2B> - Wikipedia
- <https://notepad-plus-plus.org/> - Download

From:

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

Permanent link:

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

Last update: **2023/02/08 12:01**



Options

Der **Abschnitt Options**, definiert globale Einstellungen die von verschiedenen Primus Funktionen genutzt werden. Er enthält wiederum verschiedene Abschnitte.

Abschnitt	Funktion
Public	Diverse Globale Parameter wie z.B. Bedingungen
DirectoryTemplates	Vorlage für die Neuanlage von Projektordnern
MandatoryFields	Definition von Pflichtfeldern
SynchroniseDocumentDataFields	Datenfelder die zwischen verknüpften Dokumenten synchronisiert werden, z.B. vom 3D Modell zur Zeichnung
NamedCalls	Liste von Benutzerdefinierten Callaufrufen
PreAssignData	Fasst die nachfolgenden Assign Abschnitte zusammen (optional)
AssignProjectDataFields	Klärungsbedarf
AssignDocumentDataFields	Klärungsbedarf
AssignDocumentDataFields_ADDIN	Steuert die Wertzuweisung für Aktionen die über das PrimusSEAddin ausgeführt werden
AssignDocumentDataFields_COPY	Datenfelder die beim COPY-Befehl ausgefüllt werden
AssignDocumentDataFields_NewDrawing	Datenfelder die beim Anlegen einer neuen Zeichnung gefüllt werden sollen
AssignDocumentDataFields_SMARTCOPY	Datenfelder die beim SMARTCOPY-Befehl ausgefüllt werden
AssignDocumentDataFields_SMARTCOPY_RENAME	Datenfelder die beim SMARTCOPY_RENAME-Befehl ausgefüllt werden
AssignDocumentDataFields_SMARTCOPY_MOVE	Datenfelder die beim SMARTCOPY_MOVE-Befehl ausgefüllt werden
AssignDocumentDataFields_RENAME	Klärungsbedarf
AssignNewVersionDataFields	Datenfelder die einer neuen Version eines Dokuments zugewiesen werden
AssignAttachmentsDataFields	Datenfelder die einem Anhang zugewiesen werden
AssignExportJobDocumentDataFields	Klärungsbedarf
AssignExportJobAttachmentsDataFields	Klärungsbedarf

Options

From:

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

Permanent link:

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

Last update:

2023/06/16 09:17



parentsrefscheck

Bei **parentsrefscheck** handelt es um ein [Attribut](#) des [Tags Folder_Paths](#) der Primus [Konfigurationsdatei](#). Dieses Attribut legt fest wie Primus beim Aktualisieren von Parentreferenzen verfahren soll. Drückt man bei aktiven [Komponenten](#)fenster die Taste F5 dann wird in alle Bauteile dieser Baugruppe der Link auf diese Baugruppe unter <RefParents> überprüft bzw. eingetragen. Gleichzeitig werden alle anderen <RefParents> dieses Teil ebenfalls geprüft, in dem in den jeweiligen XML kontrolliert wird, ob unter <RefChildren> dieses Teil auch aufgelistet ist. Dieses Vorgehen kostet Zeit hält aber die Datenbank konsistent und bereinigt Fehler selbsttätig.

Werte

Das Attribut kann folgende Werte annehmen.

Wert	Funktion
0	Keine Aktualisierung aller <RefParents> Einträge
1	Aktualisiert alle <RefParents> automatisch

Wenn das Attribut nicht angegeben ist, dann ist der Defaultwert 1.

Siehe auch

- [checkiswriteable_directory](#)
- [checkwriteaccess](#)
- [watchactivedir](#)

[Attribut](#), [Primus Datenbank](#)

From:

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

Permanent link:

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

Last update: **2023/02/08 12:01**



Path

Mit einem [Leer-Tag](#) im [Abschnitt Folder_Paths](#), werden Pfade zu Verzeichnissen, die im [Ordnerfenster](#) angezeigt werden sollen, definiert. Es können verschiedene [Attribute](#) übergeben werden.

Attribute

Folgende Attribute sind möglich:

Attribut	Wert ¹²⁾	Funktion
datacontext		Definiert abweichende GUI-Elemente für diese Datenablage
db		Gibt den Namen der Indexdatenbank in dem dieses Verzeichnis indiziert ist.
icon		Definiert ein statisches nicht typabhängiges Icon
iconbytype		Gibt ein Symbol oder eine Symbolliste für die Anzeige im Ordnerfenster an
id	Klärungsbedarf	
limit		Definiert die maximale Anzahl von Suchergebnissen, die der PrimusSearchService (nur SQLite) zurück liefert, beim Erreichen dieser Anzahl wird eine Meldung in der Statuszeile ausgegeben.
localpath		Definiert den Pfad in dem die Daten für den localmode abgelegt werden sollen.
name		Gibt den Namen an der im Ordnerfenster angezeigt werden soll.
path		Pflichtangabe. Definiert den Ordnerpfad
readtopxml		Steuert ob Primus für das Verzeichnis selbst nach einer pro.xml suchen soll. Dies kann u.U. den Zugriff verzögern
	0	Es wid nicht geprüft ob zu dem in path eingetragenen Verzeichnis eine .PRO.XML existiert
	1	Es wird geprüft ob zu dem in path eingetragenen Verzeichnis eine .PRO.XML existiert und ggf. auch angelegt.
type		Gibt den Verzeichnistyp an
	virtfolders	virtueller_ordner
	recyclebin	papierkorb
	jobserver	Jobserver Verzeichnis
visiblefilescond		Definiert welche Dateien in der Dokumentenliste angezeigt werden sollen oder welche nicht
visibleprojectscond		Definiert ob der Ordner im Ordnerfenster angezeigt werden soll

limit

Wenn der [PrimusSearchService](#) genau die in `limit` angegebene Anzahl an Suchergebnissen zurück liefert, meldet Primus in der [Statuszeile](#), dass die Suche mehr Ergebnisse liefert als definiert sind und dass der Anwender seine Suchkriterien verfeinern soll. Dazu muss im Primus Search Service ein Limit in gleicher Höhe eingetragen sein.

Beispiele

```
<PATH path="//primus\Data$Messages" name="Messages" datacontext="PRIMSG"
icon="Resources\Icons\messages" visiblefilescond="+;.primsg;.prim_"
watchfiles=".primsg" />
<PATH path="//primus\CAD-Daten" localpath=".LocalFiles"
name="$Primus.FRM_PROJECTS" id="FileServer" db="PDM" limit="250"
readtopxml="0" iconbytype="TYPEICON:List_ProjectTypeIcon" />
<PATH path="//primus\Data$VirtualFolders" name="$Primus.FRM_VIRTFOLDER"
type="virtfolders" icon="Resources\Icons\vrt" />
<PATH path="//primus\Data$JobServer" name="$Primus.FRM_JOBSEVER"
type="jobserver" datacontext="jobserver" icon="Resources\Icons\job"
visiblefilescond="+;.job;._o_" />
<PATH path="//primus\Data$Papierkorb" name="$Primus.FRM_RECYCLEBIN"
type="recyclebin" icon="Resources\Icons\recyclebin" />
<PATH path="//primus\Data$Standard Parts Ger" name="$Primus.FRM_STDPARTS"
type="stdparts" icon="Resources\Icons\stdpart" visiblefilescond="+;.isl" />
<PATH path="//primus\Data$Help" name="$Primus.FRM_HELP" type="info"
icon="Resources\Icons\help" visiblefilescond="+;.pdf;.lnk" />
```

12)

Defaultwerte in **Fettschrift**

From:

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

Permanent link:

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

Last update: **2024/04/15 10:20**



Primus.xml

Die **Primus.xml** wird beim Start von Primus als erstes gelesen. Der grundsätzliche Aufbau ist wie bei einer **CFG.XML**, bestehend aus **XML-Header**, **Stammelement** und mehreren Abschnitten im Stammelement.

Update

Wird ein Primus Client aktualisiert, dann wird der Inhalt der *Primus.xml*, mit Ausnahme des Attributs **version**, nicht verändert. Änderungen an den anderen Einträgen, wird nur von Primus Client selbst vorgenommen. Dadurch bleiben diese von einem Update unbeeinflusst.

Stammelement

Das Stammelement heißt **PrimusSettings** und hat mehrere Attribute.

Attribut	Wert ¹³⁾	Funktion
iconspath	Resources\Icons	Pfad aus dem Primus die Icons laden soll
licenseactivation	LOCAL	Die Lizenzaktivierung erfolgt nur lokal.
licensecheck_message_users		Durch getrennte Liste von Benutzernamen, die bei Aktivierung einen Hinweis erhalten dürfen.
licensename	CAD, OFFICE, VIEWER	Hiermit kann optional (Pflicht ab Primus2021) erzwungen werden das Primus mit einer bestimmten Lizenz gestartet wird Umstellung P7 auf P2021
licenseuncpath	<PFAD> oder .	Gibt den Pfad zur Lizenzdatei an. Wird nur benötigt wenn dieser von primusserver abweicht. Die Angabe eines Punkt (.) bedeutet, dass die Lizenz lokal auf dem Client liegt.
primusserver	<PFAD>	Pfad zum Server, mit dessen Primus.xml die version verglichen wird und ggf. ein Update erfolgt.
stringspath	\Resources\Strings\de	Definiert aus welchem Verzeichnis die sprachabhängigen Dateien geladen werden
version	<ZAHL>	Dieser Wert dient zum Vergleich mit dem Attribut version des Setup Tag auf dem Server um ggf. ein Update auszuführen. Beim Update wird der Wert mit dem des Servers gleichgesetzt.

Attribute

licensecheck_message_users

Die Primuslizenzdatei muss einmal pro Monat online geprüft und reaktiviert werden. Oft sind in den Firmen die Firewallregeln sehr streng und Programmen oder Diensten können online keine Daten austauschen. Problem ist die Reaktivierung macht der erste Primus-Client der an dem Tag gestartet wird und der bekommt eine Fehlermeldung. Mit `licensecheck_message_users` können Windowsuser definiert werden, die eine Fehlermeldung erhalten. Z.B. der Admin oder Keyuser. Alle anderen bekommen einfach eine Viewer-Lizenz wenn innerhalb von 10 Tagen die Reaktivierung nicht erfolgreich ist.

[CADAdaptApproved](#)

Abschnitte

Der **Primus.xml** ist in mehrere Abschnitte unterteilt. Der Abschnitt *Setup* ist nur auf dem Server vorhanden und wird vom Primus Setup auch nicht mit kopiert. Er würde auf dem Client jedoch ignoriert werden.

MainWindow

Im Abschnitt **MainWindow** wird die Größe und die Aufteilung des Primusfenster definiert. Es gibt verschiedene [Tag-Namen](#), die als Befehl dienen.

Tag-Name	Funktion
Position	Definiert die Größe und Position des Primusfensters
Splitter	Legt die Unterteilung des Fensters fest
Preview	Legt mit dem Attribut active (0 / 1) fest ob die Vorschau aktiv ist
Addressbar	Legt mit dem Attribut active (0 / 1) fest ob die Adressleiste aktiv ist
Toolbar	Legt mit dem Attribut iconsize (z.B. 16)) die Größe der Icons in den Kontextmenüs fest
TreeView	Legt mit dem Attribut iconsize (z.B. 18)) die Größe der Icons in Ordnerliste fest

Setup

Der Abschnitt **Setup** definiert welche Aktionen zum Aktualisieren eines Clients erforderlich sind und ob eine aktualisiert werden muss. Dazu wird der Wert des [Attributs version](#) des *Stammelements* auf dem Client, mit der Version von *Setup* auf dem Server verglichen. Ist der Wert auf dem Server höher als auf dem Client, dann wird der Anwender zu einer Aktualisierung aufgefordert. Sie ist nur in der *Primus.xml* auf dem Server, bzw. im gemeinsamen Setup Verzeichnis vorhanden. Die Anweisungen werden in der Reihenfolge, in der sie im Abschnitt *Setup* stehen, ausgeführt. Es gibt verschiedene [Tag-Namen](#), die als Befehl dienen.

Tag-Name	Funktion
CheckRunningProcesses	Überprüft ob ein bestimmter Process läuft und gibt ggf. einen Hinweis aus diesen zu beenden

Tag-Name	Funktion
CopyDir	Kopiert ein Verzeichnis
CopyFile	Kopiert eine Datei
Execute	Führt ein Programm oder einen Batch aus
RegAsmNET	Registriert eine DLL in Windows

CheckRunningProcesses

Mit CheckRunningProcesses kann geprüft werden ob bestimmte Prozesse noch aktiv sind, um den Anwender mit einer Meldung zu informieren diese vor dem Update zu beenden. Der Updatevorgang wird angehalten bis der Anwender auf **Weiter** klickt. Der Befehl wird mit folgenden Attributen gesteuert:

Attribut	Funktion
description ¹⁴⁾	Meldung die dem Anwender angezeigt wird
name	Name des zu prüfenden Prozesses

CopyDir / CopyFile

Mit **CopyDir** und **CopyFile** können komplette Verzeichnisse samt Inhalt oder einzelne Dateien kopiert werden. Der Befehl wird mit folgenden Attributen gesteuert:

Attribut	Funktion	Beispiel
from	Verzeichnis das kopiert werden soll	%SERVER%\Resources\AddIns
to	Zielverzeichnis	to=„%CLIENT%\Resources\AddIns“
replace	Mit replace wird definiert ob das to -Element ggf. vorher gelöscht werden soll. Beim Kopieren von Verzeichnissen (CopyDir) werden, die Dateien jedoch überschrieben, es wird nur vorher nicht das gesamte Verzeichnis gelöscht.	replace=„1“

Execute

Mit **Execute** können während des Installationsprozesses, Programme oder Skripte ausgeführt werden. Der Befehl wird mit folgenden Attributen gesteuert:

Attribut	Funktion	Beispiel
path	Pfad zum Programm / Skript	path=„%CLIENT%\RegisterAddIns.bat“
parameters	Parameter die übergeben werden	

Es können in beiden Attributen die Platzhalter %CLIENT% und %SERVER% genutzt werden.

RegAsmNET

Mit **RegAsmNET** können DLLs von [AddIns](#) in Windows registriert werden. Dies ist z.B. für das Primus Solid Edge AddIn erforderlich, damit es von der Anwendung beim Start mit geladen wird. Der Befehl wird mit folgenden Attributen gesteuert:

Attribut	Funktion	Beispiel
param	Übergibt Parameter an RegAsm.exe ¹⁵⁾	param=„/codebase“
path	Vollständiger Pfad zur DLL	path=„%CLIENT%\Resources\AppInterfaces\SolidEdge\AddIn\PrimusSEAddIn.dll“

SavedSelections

Beispiel

Hier ein komplettes Beispiel einer *Primus.xml*, wie sie auf dem Server liegt.

```
<?xml version="1.0" encoding="UTF-16"?>
<PrimusSettings primusserver="\primus\Setup" version="25"
stringspath="\Resources\Strings\de" iconspath="Resources\Icons"
licenseuncpath="." licensename="CAD" licenseactivation_="LOCAL"
licensecheck_message_users_="herac|salmin">
  <MainWindow>
    <Position left="-3417" top="51" width="1013" height="945" maximized="0"
  />
    <Splitter dist1="263" dist2="381" dist3="430" dist4="462" dist5="40"
model="Hor" mode2="Ver" mode3="Hor" />
    <Preview active="1" />
    <Addressbar active="1" />
    <Toolbar iconssize="16" />
    <TreeView_ iconssize="18" />
  </MainWindow>
  <Setup version="21">
    <CheckRunningProcesses name="Edge" description="Close all Sold Edge
processes to be able to install SEInterface AddIn and press OK" />
    <CopyDir from="%SERVER%\Resources\AddIns" to="%CLIENT%\Resources\AddIns"
replace="1" />
    <CopyDir from="%SERVER%\Resources\AppInterfaces"
to="%CLIENT%\Resources\AppInterfaces" replace="1" />
    <CopyDir from="%SERVER%\Resources\Icons" to="%CLIENT%\Resources\Icons"
replace="1" />
    <CopyDir from="%SERVER%\Resources\Images" to="%CLIENT%\Resources\Images"
replace="1" />
    <CopyDir from="%SERVER%\Resources\Strings"
to="%CLIENT%\Resources\Strings" replace="1" />
    <CopyFile from="%SERVER%\Resources\LISTS.xml"
to="%CLIENT%\Resources\LISTS.xml" replace="1" />
    <CopyFile from="%SERVER%\Resources\SE_MATERIAL.LIST.xml"
to="%CLIENT%\Resources\SE_MATERIAL.LIST.xml" replace="1" />
    <CopyFile from="%SERVER%\RegisterAddIns.bat"
to="%CLIENT%\RegisterAddIns.bat" overwrite="1" />
    <CopyFile from="%SERVER%\PrimusSE.CFG.xml"
to="%CLIENT%\PrimusSE.CFG.xml" overwrite="1" />
    <CopyFile from="%SERVER%\Newtonsoft.Json.dll"
```

```
to="%CLIENT%\Newtonsoft.Json.dll" overwrite="1" />
  <CopyFile from="%SERVER%\Newtonsoft.Json.xml"
to="%CLIENT%\Newtonsoft.Json.xml" overwrite="1" />
  <RegAsmNET
path="%CLIENT%\Resources\AppInterfaces\SolidEdge\AddIn\PrimusSEAddIn.dll"
param="/codebase" />
  <CopyDir from="%SERVER%\Data\LocalFiles" to="%CLIENT%\LocalFiles"
replace="0" />
</Setup>
<SavedSelections>
</SavedSelections>
</PrimusSettings>
```

Lizenzierung, Konfigurations-XML

¹³⁾

Defaultwerte in Fett

¹⁴⁾

Hier ist die Verwendung eines [Elementtextes](#) nicht möglich

¹⁵⁾

Details siehe Hilfe zur RegAsm.exe

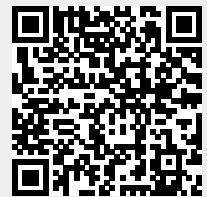
From:

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

Permanent link:

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

Last update: **2026/02/11 10:06**



PRIMUS Framework

Das Primus Framework stellt einen Rahmen mit Funktionen zur Verfügung mit denen der Anwender sich eine auf seine Bedürfnisse zugeschnittene Datenverwaltungslösung aufbauen und konfigurieren kann. Die Konfiguration erfolgt hauptsächlich durch [XML](#)-Dateien. Die zentrale Konfigurationsdatei ist die [PrimusSE.CFG.XML](#). Wobei der eigentliche Name frei wählbar ist, wichtig ist die doppelte Dateierweiterung **.CFG.xml**.

Primus Konfigurationsdateien

Dateiname	Pflicht	Funktion
Primus.xml	Ja	Enthält Informationen zu Fenstergröße und Fenstereinteilung, sowie abgespeicherte Auswahlen und Sichten
PrimusSE.CFG.xml	Ja ¹⁶⁾	Definiert die Oberfläche und die Funktionen von Primus. Es werden alle .CFG.XML im lokalen Ordner geladen
PrimusSE.CFGUSER.xml	Nein	In dieser Datei werden benutzerspezifische Änderungen zu Abschnitten aus der .CFG.XML abgespeichert

Primuskonfigurationsordner

Verzeichnis ¹⁷⁾	Pflicht	Funktion
\Ressources	Ja	
AddIns	Nein	Addins die von Primus genutzt werden sollen
appinterfaces	Nein	Primus Integrationen in Anwendungsprogramme (z.B. MS-Office, Solid Edge usw.)
icons	Ja	Enthält alle in Primus anzuzeigenden Icons .
images	Unitec spezifisch	Enthält alle Bilder z.B. Firmenlogos
strings	Ja	Enthält in Unterordnern die sprachabhängigen Texte für die Primus GUI
utils	Server, Ja	Enthält Primus Serverkomponenten (primussearchservice)
\LocalFiles	Client, Ja	Ablage von Daten die für lokales Arbeiten ausgecheckt wurden
\Data	Server, Ja	
help	Nein	Ablage der Hilfedateien zur Primus Installation
JobServer	Nein	Hier werden Jobs zum Abarbeiten abgelegt
papierkorb	Nein	Ablage für in Primus gelöschte Dateien
standard_parts_ger	Nein	Pfad für die ISL Normteile
virtualfolders	Nein	Hier werden virtuelle Ordner definiert
\CFGLayers	Nein	Ablagepfad für zu verlinkende .CFGLAYER.XML Dateien

¹⁶⁾

Es muß mind. eine CFG.XML im Verzeichnis sein

¹⁷⁾

Die Namen der Verzeichnisse sind konfigurierbar

From:

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

Permanent link:

https://dokuwiki.unitec.de/doku.php?id=primus:primus_framework

Last update: **2023/02/08 12:01**



PrimusReplicationService

Beim **PrimusReplicationService** handelt es sich um einen [Dienst](#), der den Abgleich des Datenbestands zwischen verschiedenen, gleichberechtigten Servern, über eine Netzwerkverbindung ermöglicht. Die Installation des Dienstes erfolgt lokal auf den jeweiligen Servern, die miteinander repliziert werden sollen. Konfiguriert wird der Dienst über die [PrimusReplicationService.xml](#). Er benötigt darüber hinaus ein Verzeichnis, welches im Netzwerk freigegeben ist und auf das alle anderen *PrimusReplicationService* Schreibzugriff haben.

Funktion der Replikation

In der [CFG.XML](#) wird mit dem Tag [ReplicationService](#), der jeweilige lokale Dienst angesprochen. Wird von Primus eine Änderung an den Daten vorgenommen, dann wird in die jeweiligen unter `task_path` angegebenen Verzeichnissen eine XML-Datei mit den gerade vorgenommenen Änderungen hinterlegt. Folgende Aufträge gibt es:

Auftragsname	Beschreibung
Supply	Bietet dem anderen Server eine Änderung an
RequestLocation	Fordert das Bearbeitungsrecht von einem anderen Server an
Order	Fordert die mit Supply angebotene Änderung vom anderen Server an

[Klärungsbedarf](#)

PrimusReplicationService.xml

Bei der **PrimusReplicationService.xml** handelt es sich um eine [XML](#) Konfigurationsdatei mit dem [Stammelement](#) `<Configuration>`. Es gibt folgende [Tags](#) mit den jeweiligen Attributen.

Tag	Attribut	Wert	Funktion
Server	location	<Name des eigenen Standorts	Der Standort muß in den nachfolgenden Tags aufgelistet sein
	tcp_port	<TCP/IP Port>	Netzwerkport über den der PrimusReplicationService auf dem jeweiligen Server erreichbar ist
	timer	<Zeit in Millisekunden>	Klärungsbedarf
<STANDORT>	Standortkürzel	Kürzel für den Standort, wird in location verwendet	
	task_path	<UNC-Pfad>	UNC-Pfad auf den alle anderen <i>PrimusReplicationService</i> Schreibzugriff benötigen
PATH	id	FileServer	Klärungsbedarf
	path	<Pfadangabe>	Pfad für den lokalen Dienst zur Datenablage

Die *PrimusReplicationService.xml* ist im Grunde für alle Server gleich, es ändert sich jeweils nur das `location` Attribut.

Beispiel

```
<Configuration>
  <Server location="HG" tcp_port="9091" timer="1000">
    <HG task_path="\nbuh50\task_path_HG$">
      <PATH id="FileServer" path="E:\PDM"/>
    </HG>
    <OF task_path="\nbuh52\task_path_OF$">
      <PATH id="FileServer" path="D:\PrimusPLM_SE_ReplikationDemo"/>
    </OF>
    <TR task_path="\nbuh56\task_path_TR$">
      <PATH id="FileServer" path="D:\PDM"/>
    </TR>
    <HU task_path="\sv-insight01\task_path_HU$">
      <PATH id="FileServer" path="E:\SECloud"/>
    </HU>
  </Server>
</Configuration>
```

Siehe auch

- [ReplicationService](#)

[PrimusReplicationService](#), [Primusservice](#), [Replikation](#)

From:

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

Permanent link:

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

Last update: **2024/09/12 13:27**



PrimusSE.CFG.xml

Primus wird über eine Konfigurationsdatei im [EXtensible Markup Language](#) (XML) Format konfiguriert. Der Name der Datei ist frei wählbar, er hat jedoch die doppelte Dateiendung **.CFG.XML**. Beispiel: **PrimusSE.CFG.xml** Die Kodierung ist **UCS-2 LE-BOM** und darf nicht geändert werden. Bei der Verwendung eines Texteditors ist darauf zu achten dass dieser die Kodierung unterstützt. Für die tägliche Arbeit hat sich hier [Notepad++](#) mit der Erweiterung **XML-Tools** bewährt. Wenn es zu Problemen beim Verarbeiten der Einstellungen durch Primus kommt, dann liegt dies häufig daran dass die Kodierung nicht richtig ist. Das betrifft u.a. auch die [primus_lizenzdatei](#) (PrimusLicenses.lic), diese wird dann nicht als gültig erkannt.

Grundaufbau

Der Grundsätzliche Aufbau der Konfiguration hat zwei Pflichtelemente, den **XML-Header** und das **Stammelement**.

XML-Header

Der Header der XML Dateien sieht folgendermaßen aus.

```
<?xml version="1.0" encoding="UTF-16"?>
```

Das führt allerdings beim Speichern mit [Notepad++](#) immer zu der Fehlermeldung **Document labeld UTF-16 but has UTF-8 content**.

Stammelement

Jede **.CFG.XML** muss ein und nur ein **Stammelement** enthalten. Dabei handelt es um ein [Tag](#), das alle [Abschnitte](#) der Konfiguration enthält. Der Tag-Name lautet **PrimusFolderConfig**, und darf das Attribut **name** enthalten. Ist kein Name angegeben, dann wird in Primus am obersten Projektknoten auch kein Name angezeigt.

Beispiel

```
<PrimusFolderConfig name="Unitec">
```

Abschnitte

Die Konfigurationsdatei gliedert sich in verschiedene [Abschnitte](#), von denen einige von Primus benötigt werden, andere wiederum können vom Anwender bei Bedarf erzeugt werden. Einige Abschnitte konfigurieren die [Benutzeroberfläche](#) (GUI). Alle Abschnitte **müssen** innerhalb des

[Stammelements](#) liegen.

Primus Framework

In [PRIMUS Framework](#) ist Struktur der Primus Installation mit allen Dateien und Verzeichnissen beschrieben.

[Konfigurations-XML](#)

From:

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

Permanent link:

<https://dokuwiki.unitec.de/doku.php?id=primus:primusse.cfg.xml>

Last update: **2023/02/08 12:01**



PrimusSE.CFGUSER.xml

In der **PrimusSE.CFGUSER.xml** werden Einstellungen die der Benutzer z.B. an Listen vornimmt abgespeichert, anstatt diese in der [PrimusSE.CFG.xml](#) zu speichern. Der Sinn ist, das bei einem Konfigurationsupdate, die Benutzereinstellungen nicht verloren gehen.

Aufbau

Die Datei enthält ein [Stammelement](#) wie die PrimusSE.CFG.xml auch. Darunter werden dann bei Änderungen durch den Benutzer in Primus, beim schließen von Primus bzw. beim Speichern vom Listeneinstellungen, diese in der Datei abgelegt. Beim Start von Primus werden diese Einstellungen gelesen und überlagern die Einstellungen aus der PrimusSE.CFG.xml. Es werden allerdings nur vereinfachte [Tags](#) abgespeichert die z.B. im Fall von Listen nur die [Attribute](#) `visible`, `position` und `width` haben. Da lediglich diese Attribute vom Anwender selbst in der Benutzeroberfläche verändert werden können.

Liste der Abschnitte deren Einstellungen gespeichert werden

Die Liste ist nicht auf Vollständigkeit geprüft.

Äbschnitt	Funktion
GUI_Document_List	Dateiliste
gui_document_list_bom	Stückliste im relationshipstab
relationshipstab	Die letzte aktive Ansicht (displaylastcall)

[Klärungsbedarf](#), [Konfigurations-XML](#)

From:

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

Permanent link:

<https://dokuwiki.unitec.de/doku.php?id=primus:primusse.cfguser.xml>

Last update: **2023/02/08 12:01**



PrimusSEAddin

Das **PrimusSEAddin** ist ein Addin für [Solid Edge](#). Damit stehen in der Solid Edge Oberfläche verschiedene Funktionen zur Verfügung, das Addin führt aber auch bestimmte Hintergrundoperationen aus, wenn der Anwender z.B. das aktive Dokument öffnet oder eine Datei schließt. Das Addin Verzeichnis lautet `\Resources\AppInterfaces\SolidEdge\AddIn`. Zur Konfiguration des Addins wird die [XML](#)-Datei `PrimusSEAddIn.xml` verwendet.

Konfiguration

Das [Stammelement](#) ist `PrimusSEAddIn_Settings` und hat folgende Attribute.

Attribut	Wert	Beschreibung
checkwriteaccess	0/1	Gibt an ob beim Öffnen einer Datei auf Schreibschutz geprüft wird. Ist nur in Verbindung mit dem FileAccessManager nötig.
saveoccurrences_onsave	0/1	Klärungsbedarf
Nur für die Replikation von Bedeutung		
location	<STANDORT>	Gibt den Namen des Standortes an, mit dem der Client verbunden ist. Muß mit einem der Standorte des PrimusReplicationService übereinstimmen
reserv_onopen_cond	<BEDINGUNG>	Eine Bedingung die Angibt ob eine Datei beim öffnen automatisch reserviert werden soll
reserv_sestatus	0 - Verfügbar	Gibt an welcher Status in der Solid Edge Datei durch die Reservierung gesetzt werden soll
	1 - In Arbeit	
	2 - Wird geprüft	
		Weitere in Solid Edge mögliche Status sind hier nicht sinnvoll
removereserv_onclose	0/1	Gibt an ob die beim öffnen automatisch gesetzte Reservierung beim schließen auch zurück gesetzt werden soll. Eine vorher manuell gesetzte Reservierung würde nicht zurück gesetzt werden

Beispiel

```
<PrimusSEAddIn_Settings checkwriteaccess="0" saveoccurrences_onsave="1"
location="HG" reserv_onopen_cond="'#(WorkflowStatus)==001'"
reserv_sestatus="1" removereserv_onclose="1" >
```

AssignDocumentDataFields_ADDIN

Die Zuweisung von Werten in Primus durch Aktionen des Addins werden im [Abschnitt AssignDocumentDataFields_ADDIN](#) definiert.

BOM

[Klärungsbedarf](#)

Einfluss auf die Replikation

Beim SEAddIn Save-Event wird [Replikation](#) ausgelöst.

Automatische Reservierung

Bei einer automatischen Reservierung wird in dem [Datenfeld](#) Reservation der Name des Benutzers geschrieben, der das Dokument gerade bearbeitet. In das Datenfeld ResByAddIn wird eingetragen, dass es sich um eine automatische Reservierung handelt. Nur automatische Reservierungen werden auch automatisch zurückgesetzt.

Siehe auch

- [PrimusReplicationService](#)
- [AssignDocumentDataFields_ADDIN](#)

[Klärungsbedarf](#), [Replikation](#), [Addin](#), [Konfigurations-XML](#)

From:

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

Permanent link:

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

Last update: **2025/03/17 09:57**



PrimusSearchService

Der **PrimusSearchService** indiziert die zu verwaltenden Daten und bietet eine Möglichkeit für eine schnelle Suche.

Installationsverzeichnis

Der Primus Search Service liegt i.d.R. im Setup Verzeichnis von Primus unter `|Resources\Utils\PrimusSearchService`. Es gibt dort folgende Unterverzeichnisse:

Verzeichnis	Beschreibung
db	Verzeichnis in dem die Indexdatenbank nebst Log abliegt
offline_calls	Wenn der Suchdienst gestoppt ist, können Clients hier geänderte Dateien hinterlegen die der Dienst nachträglich verarbeitet wenn er wieder gestartet wird.
x64	Binary Verzeichnis von SQLite
x86	Binary Verzeichnis von SQLite

offline_calls

Wenn der Suchdienst gestoppt ist, können Clients hier geänderte Dateien hinterlegen die der Dienst nachträglich verarbeitet wenn er wieder gestartet wird. Dazu muss das Verzeichnis für die Clients erreichbar- und beschreibbar sein. Das geschieht idealerweise über eine verdeckte Freigabe (`offline_calls$`). Das Verzeichnis muss sowohl unter [SearchServiceDB](#) auf den Clients als auch in der [PrimusSearchService.xml](#) eingetragen sein.

PrimusSearchService.xml

In der `PrimusSearchService.xml` erfolgt die Konfiguration des Suchdienstes.

Aufbau der XML

Tag			Attribute	Beschreibung
<Configuration>	<Settings>			Stammelement
				Den Dienst betreffende Einstellungen
			db_path	Ablageverzeichnis für die Datenbankdatei
			tcp_port	TCP/IP Port über den die Clients mit dem Suchdienst kommunizieren
			offline_calls	Ablageverzeichnis für offline_calls
			culture	Definiert die Datumsformatierung für die SQLite Datenbank, culture=„de-DE“
	<Tables>			Definition der Suchbereiche
		<XXX_Projects>		Definiert einen Suchbereich mit dem Namen XXX für Verzeichnisse
		<XXX_Documents>		Definiert einen Suchbereich mit dem Namen XXX für Dokumente
			path	Ablageverzeichnis für die zu indizierenden Daten
			prodata	Definiert ob es sich um eine Projekt- (1) oder einen Dokumentenindex (0) handelt
			sqlite	Definiert ob SQLite genutzt (1) werden soll oder nicht (0)
			limit	Maximale Anzahl der Ergebnisse die eine Suche zurück liefern soll (nur SQLite)
			map	Ordnet einen Laufwerksbuchstaben einem UNC-Pfad zu

Indizierte Datenfelder

In den Suchbereichen werden die zu indizierenden Datenfelder als [Leer-Tag](#) mit dem Namen des Datenfelds und dem Attribut length für die Anzahl der Zeichen des Datenfelds die indiziert werden sollen. Das Attribut length ist bei Verwendung von [SQLite](#) nicht mehr erforderlich, für Datumsangaben sollte allerdings das Attribut type mit dem Wert date angegeben werden um korrekte Suchergebnisse zu erhalten.

```
<ChangeName length="8" />
<ChangeDate length="" type="date" />
```

map Attribut

Der PrimusSearchService läuft als Dienst, evtl. mit einem speziellen [Dienstkonto](#). Diesem Dienst stehen zur Laufzeit keine Laufwerksbuchstaben zur Verfügung, da diese erst im Kontext einer Benutzersitzung erstellt werden. Damit nun der PrimusSearchService auch diese Dokumente indizieren kann, erfolgt mit map eine Zuordnung eines Pfads mit Laufwerksbuchstaben zu einem UNC-Pfad.

Wenn auf dem Client

```
<PATH path="K:\FileServer"
```

dann wird der dies so konfiguriert

```
<FileServerLocal_Documents path="K:\FileServer" prodata="0" sqlite="1"  
map="K:\FileServer=\\Cadadaptdev1\d\CADAdapt\PrimusProjekte\DemoPLM\PrimusPL  
M_SE\Data\FileServer" limit="1000" >
```

Dienstkonto

Der PrimusSearchService muss mit als Windowsdienst mit einem Dienstkonto gestartet werden, das folgende Anforderungen erfüllen muss:

- Lesender Zugriff auf die verwaltenden Dateien
- Auf Ablage der Suchdatenbank Datenbank muss er schreiben können

Firewalleinstellungen

Der im Attribut tcp_port konfigurierte Port muss in der Firewall ein- und ausgehend, sowohl auf dem Client als auch dem Server freigegeben sein.

Siehe auch

- [SearchServiceDB](#)
- [Path](#)

[Primusservice](#)

From:

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

Permanent link:

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

Last update: **2023/12/08 16:01**



PrimusSmartCopy

Bei **PrimusSmartCopy** handelt es sich um ein [Addin](#) für Primus, das dem Anwender die [SmartCopy](#)-Funktionen zur Verfügung stellt. Die Funktionen sind in der **PrimusSmartCopy.dll** enthalten, die standardmäßig unter `.\Resources\AddIns` zu finden ist.

Allgemeine Einstellungen

Im Abschnitt [Public](#) der `Primus.CFG.XML` gibt es einen Eintrag [SmartCopy](#) in dem einige allgemeine Parameter des Addins mit Attributen konfiguriert werden können.

Liste der Menübefehle

ACHTUNG, die Liste ist derzeit nicht vollständig!

Funktion	Beschreibung
smartcopyalldocs	Markiert alle Dokumente eines Ordner für die aktuelle SmartCopy Operation
smartcopybreaksession	Bricht die aktuelle SmartCopy Session ab. Ein Fortsetzung ist mit smartLoadLastSession möglich
smartcopyignore	Ignoriert das gewählte Dokument für die aktuelle SmartCopy Session
smartcopyignorerec	Ignoriert das gewählte und alle darin enthaltenen Dokumente für die aktuelle SmartCopy Session
smartsmartcopymanager	Ruft den SmartCopy Manager auf
smartCopyMarkTarget	Definiert den Zielpfad für die aktuelle SmartCopy Session. Darf nicht bei Rename benutzt werden!
smartCopyNewSession	Startet eine neue SmartCopy Session
smartcopyremovemarktarget	Macht die Definition des Zielpfads rückgängig
smartCopyRenameDir	Umbenennen von Verzeichnissen
smartCopySet	
smartCopySetRec	
smartcopysettarget	
smartignorealldocs	Ignoriert alle Dokumente des gewählten Ordners für die SmartCopy Session
smartloadlastsession	Letzte Session Laden
smartpaste	Fügt die für SmartCopy markierten Dateien in das Zielverzeichnis ein
smartreplacereferences	Referenzen ersetzen
smartrestoremovesession	Umbenennen von Verzeichnissen rückgängig machen

Move und Rename Sessions

Bei SmartCopy Move oder Rename werden abschließend alle nicht benutzten Parent Einträge aus der [Daten-XML](#) entfernt.

Besonderheiten bei Move

Damit beim Verschieben die Anhänge ebenfalls mitgenommen werden muß dies in [AttachmentFiles](#) entsprechend konfiguriert werden.

Besonderheiten bei Rename

[smartCopyMarkTarget](#) darf bei SmartCopy Rename nicht angewendet werden, da sonst die Dokumente während des Umbenennens verschoben werden.

[Addin](#), [Klärungsbedarf](#), [SmartCopy](#)

From:

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

Permanent link:

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

Last update: **2024/09/05 07:57**



prmCopyDocData

Der [Menübefehl](#) **prmCopyDocData** kopiert alle Felder aus XML eines Dokuments in die Zwischenablage. Diese können dann mit [prmPasteDocData](#) in die Datenfelder des gewählten Dokuments eingefügt werden.

Attribute

Attribut	Wert	Funktion
enabled	Bedingung	Steuert ob der Befehl aktiv ist
	TRUE	Aktiv
	FALSE	Inaktiv
icon	copy_attri ¹⁸⁾	Pfad zum Icon (optional)
parameter		Kann leer bleiben
text	\$Primus.MNU_ELEM_COPY_ATTRIB	Elementtext für den Menüeintrag
visible	Bedingung	Steuert ob der Befehl im Menü sichtbar ist
	TRUE	Sichtbar
	FALSE	Unsichtbar

Beispiel

```
<prmCopyDocData visible="TRUE" enabled="TRUE" parameter=""
text="$Primus.MNU_ELEM_COPY_ATTRIB" icon="Resources\Icons\copy_attri" />
```

Siehe auch

- [prmPasteDocData](#)

[menu](#), [gui](#), [kontext](#), [copy](#)

¹⁸⁾

Standardicon in Primus

From:

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

Permanent link:

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

Last update:

2024/08/14 08:07



prmElemenRequestLocationWithComponents

Mit dem [Menübefehl](#) **prmElemenRequestLocationWithComponents** wird der Bearbeitungsstandort der [Replikation](#) für ein Element nebst Verknüpfungen und ggf. übergeordneten Elementen auf den aktuellen Standort festgelegt.

Parameter

Mit dem [Attribut](#) parameter können folgende Werte in der angegebenen Reihenfolge übergeben werden.

Nr.	Parameter	Funktion
1	%FILEPATH% ¹⁹⁾ (optional)	Pfad auf das zu verarbeitende Dokument, wenn der Parameter leer ist werden alle gewählten Dokumente verarbeitet
2	0	Standort nur für das gewählte Dokument übernehmen
	1	Das gewählte Dokument inkl. Komponenten
3	.dft	Wenn Parameter 2 = 1 dann wird auch der Standort für die Zeichnungen der Komponenten geändert

Beispiel

```
<prmElemenRequestLocationWithComponents visible="CALL(IsProjectActive)"
enabled="%XML_Project:Replicate%==1" parameter="|0" text="Standort-InArbeit
übernehmen" icon="" />
<prmElemenRequestLocationWithComponents visible="CALL(IsProjectActive)"
enabled="%XML_Project:Replicate%==1" parameter="|1" text="Standort-InArbeit
übernehmen inkl. Komponenten" icon="" />
<prmElemenRequestLocationWithComponents visible="CALL(IsProjectActive)"
enabled="%XML_Project:Replicate%==1" parameter="|1|.dft" text="Standort-
InArbeit übernehmen inkl. Komponenten und Zeichnungen" icon="" />
<!-- Aufruf mittels Call in einem Workflow -->
<Call method="prmElemenRequestLocationWithComponents"
parameter="%FILEPATH%|0" condition="" />
```

Siehe auch

- [prmProjectRequestLocationForDocuments](#)

[Menübefehl](#), [Replikation](#)

¹⁹⁾

Ist nur zulässig wenn Parameter 2 = 0

From:

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

Permanent link:

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

Last update: **2023/08/08 15:23**



prmElementsChangeValueByDlg

Mit dem [Menübefehl](#) **prmElementsChangeValueByDlg** können einzelne Datenfelder über einen Dialog geändert werden. Dies ist auch bei schreibgeschützten Dokumenten möglich.

Parameter

Mit dem [Attribut](#) parameter können folgende Werte in der angegebenen Reihenfolge übergeben werden.

Nr.	Parameter	Funktion
1	<FELDDNAME>	Datenfeld das editiert werden soll
2	TEXT	Bezeichnung im Datenblatt
3	<LISTNAME>	Auswahlliste für die Dateneingabe (optional)
4	1	Ändern schreibgeschützter Dokumente (optional)

Beispiele

```
<prmElementsChangeValueByDlg visible="TRUE" enabled="TRUE"
parameter="CreatedName|Ersteller|List_User:text|0" text="Ersteller (nur wenn
schreibbar)" icon="" />
<prmElementsChangeValueByDlg visible="TRUE" enabled="TRUE"
parameter="CreatedDate|ErstellDatum" text="ErstellDatum (nur wenn
schreibbar)" icon="" />
<prmElementsChangeValueByDlg visible="$(Condition.isAdminUser)"
enabled="TRUE" parameter="WorkflowStatus|Status|List_WorkflowStatus|1"
text="Workflow-Status" icon="" />
<prmElementsChangeValueByDlg visible="$(Condition.isAdminUser)"
enabled="TRUE" parameter="WriteStatus|Status|List_WriteStatus|1"
text="Schreib-Status" icon="" />
```

Siehe auch

- [prmElementsEditReadOnlyDataSheet](#)
- [SetXMLValue](#)

Menübefehl

From:

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

Permanent link:

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

Last update: **2023/07/25 12:16**



prmElementsEditReadOnlyDataSheet

Mit dem [Menübefehl](#) **prmElementsEditReadOnlyDataSheet** kann ein Datenblatt zum ändern von Datenfeldern aufgerufen werden. Es können dazu eigene [Datenblätter](#) definiert werden. Das Ändern der Felder schreibgeschützter Dokumente ist möglich. Es werden nur die Daten in der [XML](#) geändert, es erfolgt kein Abgleich mit dem Dokument selbst.

Parameter

Dem Befehl werden folgende Parameter übergeben.

Parameter	Funktion
1 <DATENBLATTNAME>	Name des Datenblatts das verwendet werden soll

Beispiel

```
<prmElementsEditReadOnlyDataSheet visible="TRUE" enabled="TRUE"
parameter="GUI_Document_DataSheet_Einkauf" text="Artikelkategorie ändern"
icon="" />
```

Siehe auch

- [prmElementsChangeValueByDlg](#)

[Menübefehl](#)

From:

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

Permanent link:

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

Last update: **2023/02/08 12:01**



prmElementsGenerateNewDocumentName

Mit dem Menübefehl **prmElementsGenerateNewDocumentName** wird ein neuer Dateiname generiert und in der Zwischenablage abgelegt. Dieser kann dann in einer Anwendung genutzt werden um von dieser aus einen Primuskonformen Dateinamen zu generieren.

Attribute

Attribut	Wert	Funktion
enabled	Bedingung	Steuert ob der Befehl aktiv ist
	TRUE	Aktiv
	FALSE	Inaktiv
icon	counter	Pfad zum Icon (optional)
parameter	0	Klärungsbedarf
	1	
	2	
	3	%FILENAME% Klärungsbedarf
text	Dateiname generieren	Elementtext für den Menüeintrag
visible	Bedingung	Steuert ob der Befehl im Menü sichtbar ist
	TRUE	Sichtbar
	FALSE	Unsichtbar

(Primus Standardwerte in **Fett**.)

Bedingungen

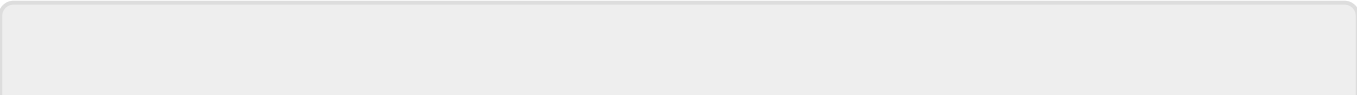
Damit der Befehl genutzt werden kann, sollten folgende Bedingungen geprüft werden. Ob der Befehl sichtbar oder aktiv ist, ist dabei unerheblich.

Bedingung	Funktion
CALL(IsProjectActive)	Ist der aktiv gewählte Knoten „Project“

Beispiel

```
<prmElementsGenerateNewDocumentName visible="TRUE"
enabled="'CALL(IsProjectActive)'" parameter="||%FILENAME%" text="Dateiname
generieren" icon="Resources\Icons\counter" />
```

GUI, Menübefehl, Nummerngenerator, Dateiname



From:

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

Permanent link:

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

Last update: **2024/08/14 08:11**



prmElementsNewVersion

Mit dem [Menübefehl](#) **prmElementsNewVersion** werden neue Revisionen von Dokumenten erstellt.

Attribute

Attribut	Wert	Funktion
enabled	Bedingung	Steuert ob der Befehl aktiv ist
	TRUE	Aktiv
	FALSE	Inaktiv
icon	newversion	Pfad zum icon (optional)
parameter	0	Ohne Komponenten
	1	Mit Komponenten
	2	Mit Zeichnungen (und Komponenten?) Klärungsbedarf
text	\$Primus.MNU_ELEM_NEW_VERSION	Elementtext für den Menüeintrag
visible	Bedingung	Steuert ob der Befehl im Menü sichtbar ist
	TRUE	Sichtbar
	FALSE	Unsichtbar

(Primus Standardwerte in **Fett**.)

Bedingungen

Damit der Befehl genutzt werden kann, sollten folgende [Bedingungen](#) geprüft werden. Ob der Befehl sichtbar oder aktiv ist, ist dabei unerheblich.

Bedingung	Funktion
\$(Workflow.newversioncond)	Prüft ob die Bedingung für eine neue Version erfüllt ist
CALL(IsProjectActive)	Ist der aktiv gewählte Knoten „Project“
\$(Condition.isSEDraft)	Bei neuer Version ohne Komponenten zum Beispiel

Siehe auch

- [prmElementsRemoveNewestVersion](#)

[GUI](#), [Menübefehl](#), [Workflow](#)

From:

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

Permanent link:

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

Last update:

2024/08/14 08:14



<https://dokuwiki.unitec.de/>

Printed on 2026/03/29 08:39

prmElementsRemoveNewestVersion

Mit dem Menübefehl **prmElementsRemoveNewestVersion** kann die aktuell in Arbeit befindliche Revision, die den **WorkflowStatus** 001 haben muss, wieder verworfen werden. Dabei wird die Datei gelöscht und die Vorversion wird von Veraltet (004) auf Freigegeben (003) zurückgesetzt.

Attribute

Folgende Attribute können angegeben werden:

Attribut	Wert	Funktion
enabled		Steuert ob der Befehl aktiv ist
	TRUE	Aktiv
	FALSE	Inaktiv
icon	optional	Pfad zum Icon
parameter	003 Read	Status und Schreibschutzflag für die Vorversion
text	\$Primus.MNU_ELEM_REMOVELASTVERSION	Elementtext für den Menüeintrag
visible	Bedingung	Steuert ob der Befehl im Menü sichtbar ist
	\$(WorkflowStatus)==001	Nur bei Dokumenten mit dem Status 001 sichtbar
	FALSE	Unsichtbar

Beispiel

```
<prmElementsRemoveNewestVersion visible="$(WorkflowStatus)==001"
enabled="TRUE" parameter="003|Read"
text="$Primus.MNU_ELEM_REMOVELASTVERSION" icon="Resources\Icons\delete" />
```

GUI, Menübefehl, Workflow, Kontext

From:

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

Permanent link:

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

Last update:

2024/08/14 08:15



prmFindCorruptXMLDataFiles

Mit dem Menübefehl **prmFindCorruptXMLDataFiles** kann der Datenbestand oder auch nur ein Verzeichnis mit Unterverzeichnissen, nach beschädigten **Daten-XML** durchsucht werden. Es wird eine *FindCorruptXMLDataFiles.log* angelegt.

Beispiel

```
<prmFindCorruptXMLDataFiles visible="TRUE" enabled="TRUE" parameter=""  
text="FindCorruptXMLDataFiles" icon="" />
```

Menübefehl

From:

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

Permanent link:

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

Last update: **2025/06/03 10:48**



prmPasteDocData

Der [Menübefehl](#) **prmPasteDocData** schreibt zuvor mit [prmCopyDocData](#) in die Zwischenablage kopierte Datenfelder, in das gewählte Dokument.

Attribute

Attribut	Wert	Funktion
enabled	CALL(IsDataForPaste)	Steuert ob der Befehl aktiv ist
	TRUE	Aktiv
	FALSE	Inaktiv
icon	optional	Pfad zum Icon
parameter	<FELDNAME>	Liste von Feldnamen die geschrieben werden sollen. Mehrere Felder werden mit getrennt
text	\$Primus.MNU_ELEM_PASTE_ATTRIB	Elementtext für den Menüeintrag
visible	Bedingung	Steuert ob der Befehl im Menü sichtbar ist
	TRUE	Sichtbar
	FALSE	Unsichtbar

Bedingungen

Die [IsDataForPaste](#) wird benötigt um festzustellen ob es überhaupt Daten zum Einfügen in der Zwischenablage gibt. Falls dies nicht der Fall ist, dann ist die Funktion inaktiv. Dieses Verhalten ist der Anwender von Windowsanwendungen her so gewohnt.

Beispiel

```
<prmPasteDocData visible="TRUE" enabled="CALL(IsDataForPaste)"
parameter="Title|Number" text="$Primus.MNU_ELEM_PASTE_ATTRIB" icon="" />
```

Siehe auch

- [prmCopyDocData](#)
- [menu](#), [gui](#), [kontext](#), [copy](#)

From:

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

Permanent link:

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

Last update: **2024/08/14 08:19**



prmProjectsFindByData

Der [Menübefehl](#) **prmProjectsFindByData** ermöglicht die Suche nach Verzeichnissen mit bestimmten Eigenschaften.

Attribute

Attribut	Wert	Funktion
enabled	Bedingung	Steuert ob der Befehl aktiv ist
	TRUE	Aktiv
	FALSE	Inaktiv
icon		Pfad zum icon (optional)
parameter		Siehe Parameter
text		Elementtext für den Menüeintrag
visible	Bedingung	Steuert ob der Befehl im Menü sichtbar ist
	TRUE	Sichtbar
	FALSE	Unsichtbar

Parameter

Mit dem Parameter wird angegeben nach welcher Eigenschaft gesucht werden soll, und die Feldbeschreibung in der Eingabemaske. Als Trennzeichen fungiert hier der Doppelpunkt (:).

Parameter	Wert
1	Eigenschaftsfeld des Projekts
2	Anzeigename im Eingabedialog für den Suchbegriff

Die Verwendung eines [Elementtextes](#) ist an dieser Stelle z.Zt. nicht möglich.

Beispiel

```
<prmProjectsFindByData visible="TRUE" enabled="TRUE"
parameter="Number:Nummer" text="Dateipfad suchen nach Nummer" icon=""/>
```

Siehe auch

- [prmprojectsfindbyname](#)
- [prmfindfilebypath](#)

[menu](#), [gui](#), [kontext](#), [Suche](#)

From:

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

Permanent link:

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

Last update: **2023/02/08 12:01**



prmProjectsUploadXmlsToDB

Mit dem Menübefehl **prmProjectsUploadXmlsToDB** wird der [PrimusSearchService](#) angewiesen die Indexdatenbank neu aufzubauen. Es handelt sich dabei um einen Befehl für [Administratoren](#), daher wird er normalerweise nur mit gedrückter STRG-Taste sichtbar.

Attribute

Der Befehl kennt folgende Attribute:

Attribut	Wert	Funktion
enabled	Bedingung	Steuert ob der Befehl aktiv ist
	TRUE	Aktiv
	FALSE	Inaktiv
icon		Pfad zum Icon (optional)
parameter		Ohne Parameterangabe wird nur der Suchindex für Dokumente aktualisiert
1		Bleibt leer
2	Documents:Projects	Aktualisiert den Index für Dokumente und Verzeichnisse. Der Doppelpunkt (:) fungiert hier als Trennzeichen zwischen den Begriffen, der Trennstrich () trennt die Parameter voneinander.
text		Elementtext für den Menüeintrag
visible	CALL(IsCtrlKey)	Steuert ob der Befehl im Menü sichtbar ist, normalerweise nur bei gedrückter STRG-Taste
	TRUE	Sichtbar
	FALSE	Unsichtbar

Beispiel

```
<prmProjectsUploadXmlsToDB
visible="'CALL(IsProjectActive)'AND'CALL(IsCtrlKey)'" enabled="TRUE"
parameter="|Documents:Projects" text="Suchindex aktualisieren" icon="" />
```

[GUI](#), [Menübefehl](#), [Administrator](#), [Suche](#)

From:

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

Permanent link:

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

Last update:

2024/08/14 08:25



prmProjectSynchroniseAllDataFields

Mit dem Menübefehl **prmProjectSynchroniseAllDataFields** können die Datenfelder aller Dokumente eines Verzeichnisses und seiner Unterverzeichnisse synchronisiert werden. Dabei werden lediglich die Inhalte der **Daten-XML** abgeglichen, nicht jedoch die Dateien selbst verändert. Es handelt sich dabei um einen Befehl für **Administratoren**, daher wird er normalerweise nur mit gedrückter STRG-Taste sichtbar. Die Synchronisation findet anhand eines **Abschnitts** wie z.B. **SynchroniseDocumentDataFields** statt.

Attribute

Der Befehl kennt folgende Attribute:

Attribut	Wert	Funktion
enabled	Bedingung	Steuert ob der Befehl aktiv ist
	TRUE	Aktiv
	FALSE	Inaktiv
icon		Pfad zum Icon (optional)
parameter	SynchroniseDocumentDataFields	Names des Abschnitts der die zu synchronisierenden Felder definiert
text		Elementtext für den Menüeintrag
visible	CALL(IsCtrlKey)	Steuert ob der Befehl im Menü sichtbar ist, normalerweise nur bei gedrückter STRG-Taste
	TRUE	Sichtbar
	FALSE	Unsichtbar

Beispiel

```
<prmProjectSynchroniseAllDataFields visible="'CALL(IsCtrlKey)'"
enabled="TRUE" parameter="SynchroniseDocumentDataFields"
text="SynchroniseAllDataFields-DFT" icon="" />
```

GUI, Menübefehl, Administrator

From:

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

Permanent link:

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

Last update:

2024/08/14 08:37



prmProjectUpdateAllRefs

Der [Menübefehl](#) **prmProjectUpdateAllRefs** aktualisiert die Komponentenliste für alle Dokumente im gewählten Ordner und dessen Unterordnern. Dabei werden jedoch die im [Abschnitt Public](#) unter [DocumentRefListCall](#) definierten [Attribute](#) ausgewertet. Wichtig ist hier die [Bedingung](#) die für gewöhnlich die CAD Lizenz erfordert, auf Servern ist jedoch häufig eine JOBSERVER Lizenz eingetragen.

Attribute

Der Befehl kennt folgende Attribute:

Attribut	Wert	Funktion
enabled	Bedingung	Steuert ob der Befehl aktiv ist
	TRUE	Aktiv
	FALSE	Inaktiv
icon		Pfad zum Icon (optional)
parameter		Parameterbeschreibung
text		Elementtext für den Menüeintrag
visible	CALL(IsCtrlKey)	Steuert ob der Befehl im Menü sichtbar ist, normalerweise nur bei gedrückter STRG-Taste
	TRUE	Sichtbar
	FALSE	Unsichtbar

Parameter

Mit dem [Attribut](#) parameter können folgende Werte in der angegebenen [Reihenfolge](#) übergeben werden. Die einzelnen Parameter werden durch ein Pipezeichen (|) getrennt.

Nr.	Wert	Parameter	Funktion
1	Liste der Dateiendungen	<DATEITYPENLISTE>	Mit Semikolon getrennte Liste der Dateitypen die berücksichtigt werden sollen. Der Dateityp wird mit führendem Punkt angegeben
2	Modus	ONLYCHANGED	Optionaler Parameter der veranlasst das nur Dokumente bei denen das Dokument geändert, die XML aber nicht geändert wurde, berücksichtigt werden

Beispiel

```
<<!-- Alle Solid Edge Dateitypen berücksichtigen -->>
<prmProjectUpdateAllRefs visible="" enabled="TRUE"
parameter=".par;.psm;.asm;.dft" text="Alle CAD-Referenzen aktualisieren"
icon="" />
<<!-- Alle Solid Edge Dateitypen aber nur bei geänderten Dokumenten
```

```
berücksichtigen -->>  
<prmProjectUpdateAllRefs visible="" enabled="TRUE"  
parameter=".par;.psm;.asm;.dft|ONLYCHANGED" text="Nur geänderte CAD-  
Referenzen aktualisieren" icon="" />
```

Siehe auch

- [prmUpdateXMLFieldsAndParentRefs](#)
- [DocumentRefListCall](#)

[GUI](#), [Menübefehl](#), [Administrator](#)

From:

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

Permanent link:

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

Last update: **2024/08/14 08:39**



prmRefElementsSetRef1

Mit dem Menübefehl **prmRefElementsSetRef1** kann eine bestimmte Komponente einer Zeichnung als Ref1 gesetzt werden. Der Befehl lässt sich nur sinnvoll im Menü des Komponentenfensters einer Zeichnungsdatei nutzen. Das zuerst in einer Zeichnung verwendete Modell, ist standardmäßig die Ref1. Im Komponentenfenster wird die Ref1 durch ein } in der linken Spalte gekennzeichnet. Mit diesem Befehl wird nur die Ref1 in Primus geändert. Eine Änderung der Indexreferenz in Solid Edge ist derzeit technisch möglich und wird von diesem Befehl auch nicht durchgeführt.

Attribute

Der Befehl kennt folgende Attribute:

Attribut	Wert	Funktion
enabled	Bedingung	Steuert ob der Befehl aktiv ist
	TRUE	Aktiv
	FALSE	Inaktiv
icon		Pfad zum Icon (optional)
parameter		
text	\$Primus.MNU_REF_ELEMENTS_SET_REF1	Elementtext für den Menüeintrag
visible	CALL(IsActiveRegTab tabChildRefs)	Steuert ob der Befehl im Menü sichtbar ist, normalerweise nur im Komponentenfenster
	TRUE	Sichtbar
	FALSE	Unsichtbar

Beispiel

```
<prmRefElementsSetRef1 visible="'CALL(IsActiveRegTab tabChildRefs)'  
enabled="TRUE" parameter="" text="$Primus.MNU_REF_ELEMENTS_SET_REF1"  
icon=""/>
```

GUI, Menübefehl, Registerspezifisch

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

Permanent link:
<https://dokuwiki.unitec.de/doku.php?id=primus:prmrefelementssetref1>

Last update: 2024/08/14 09:18



prmsetreservation

Mit dem [Menübefehl](#) **prmsetreservation** können Dokumente für einen bestimmten Benutzer reserviert oder die [reservierung](#)(en) aufgehoben werden.

Attribute

Der Befehl kennt folgende Attribute:

Attribut	Wert	Funktion
enabled	Bedingung	Steuert ob der Befehl aktiv ist
	TRUE	Aktiv
	FALSE	Inaktiv
icon		Pfad zum Icon (optional)
parameter		Parameterbeschreibung
text		Elementtext für den Menüeintrag
visible		Steuert ob der Befehl im Menü sichtbar ist
	TRUE	Sichtbar
	FALSE	Unsichtbar

Parameter

Mit dem [Attribut](#) parameter können folgende Werte in der angegebenen [Reihenfolge](#) übergeben werden. Die einzelnen Parameter werden durch ein Pipezeichen (|) getrennt.

Nr.	Wert	Parameter	Funktion
1	Zugehörige Datei	<DATEITYP>	Datei der zugehörigen Datei die mit berücksichtigt werden soll
2	Modus	0 / 1	Setzt bzw. hebt die Reservierung auf
3	Adminuser	.adminuser.	Optionale Angabe der / des Adminusers. Die Liste beginnt, endet und ist getrennt mit einem Punkt .. Wenn der Parameter leer ist, dann kann nur der Benutzer der die Reservierung hat, diese wieder aufheben.

Beispiel

```
<<!-- Nur der Reservierungsbenutzer kann zurücksetzen -->>
<prmSetReservation visible="'CALL(IsProjectActive)'" enabled="TRUE"
parameter=".dft|0"
text="Reservierung aufheben (Admin)" icon="" />
<<!-- Auch Adminbenutzer können zurücksetzen -->>
<prmSetReservation visible="'CALL(IsProjectActive)'" enabled="TRUE"
parameter=".dft|0|.user1.user2." text="Reservierung aufheben (Admin)"
icon="" />
```

[GUI, Menübefehl](#)

From:

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

Permanent link:

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

Last update: **2024/08/14 07:56**



prmShowWhereUses

Der Menübefehl **prmShowWhereUses** zeigt im [GUI_Menu_Relationships](#) den Verwendungsnachweis für das ausgewählte Dokument an. Der Befehl steht in der Listbox oberhalb der Liste zur Auswahl zur Verfügung.

Parameter

Der Befehl hat folgende Parameter.

Parameter	Wert	Funktion
1		Klärungsbedarf
2		
3	<DATEITYPEN>	Listet nur die angegebenen Dateitypen auf

Beispiel

```
<prmShowWhereUses visible="'$(Condition.isActiveDocEditableNode)'"
enabled="TRUE" parameter="||.par.psm.asm.dft.pwd"
text="$Primus.MNU_SHOW_WHERE_USES" icon="" />
```

Siehe auch

- [prmShowWhereUses_AllComponentsByExt](#)
- [prmShowWhereUses_AllDrawings](#)
- [prmShowWhereUses_AllWithoutDrawings](#)
- [prmShowWhereUses_ComponentsDrawings](#)

[Menübefehl](#), [Klärungsbedarf](#)

From:

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

Permanent link:

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

Last update: **2023/07/28 09:26**



prmShowWhereUses_AllDrawings

Der Menübefehl **prmShowWhereUses_AllDrawings** zeigt im [GUI_Menu_Relationships](#) die Liste der Zeichnungen aller Komponenten für das ausgewählte Dokument an. Der Befehl steht in der Listbox oberhalb der Liste zur Auswahl zur Verfügung.

Parameter

Der Befehl hat folgende Parameter.

Parameter	Wert	Funktion
1	<DATEITYPEN>	Listet nur die angegebenen Dateitypen auf
2		Klärungsbedarf
3		
4		
5 (optional)		Alle Zeichnungen der Komponenten
	OccurrList.Ref1	Listet nur die Zeichnungen auf in der die Komponenten die Ref1 sind (optional)

Beispiel

```
<!-- Alle Zeichnungen der Komponenten -->
<prmShowWhereUses_AllDrawings
visible="'$(Condition.isActiveDocEditableNode)'" enabled="TRUE"
parameter=".dft" text="$Primus.MNU_SHOW_WHERE_USES_ALL_DRAWINGS" icon="" />
<!--Für alle 3D-Komponenten werden mit der Option Ref1 nicht mehr alle
Zeichnungen aufgelistet, sondern nur die, wo diese Komponenten die erste
Referenz (Ref1) sind. -->
<prmShowWhereUses_AllDrawings
visible="'$(Condition.isActiveDocEditableNode)'" enabled="TRUE"
parameter=".dft|||OccurrList.Ref1" text="Zeichnungsliste" icon="" />
```

Siehe auch

- [prmShowWhereUses](#)
- [prmShowWhereUses_AllComponentsByExt](#)
- [prmShowWhereUses_AllWithoutDrawings](#)
- [prmShowWhereUses_ComponentsDrawings](#)

[Menübefehl](#), [Klärungsbedarf](#)

From:

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

Permanent link:

https://dokuwiki.unitec.de/doku.php?id=primus:prmshowwhereuses_alldrawings

Last update: **2023/07/28 09:31**



prmStartProcess

Mit dem [Menübefehl](#) **prmStartProcess** können externe Programme über einen Menüeintrag aufgerufen werden. Sind mehrere Dateien ausgewählt dann wird der Prozess für jede gewählte Datei aufgerufen.

Attribute

Attribut	Wert	Funktion
enabled	Bedingung	Steuert ob der Befehl aktiv ist
	TRUE	Aktiv
	FALSE	Inaktiv
icon		Pfad zum Icon (optional)
parameter	STARTH()	Programm wird ohne sichtbares (hidden) Fenster aufgerufen
	STARTV()	Programm wird mit sichtbarem (visible) Fenster aufgerufen
text		Elementtext für den Menüeintrag
visible	Bedingung	Steuert ob der Befehl im Menü sichtbar ist
	TRUE	Sichtbar
	FALSE	Unsichtbar

Beispiel

```
<prmStartProcess
parameter="STARTV(. \Resources\AddIns\DemoProcess.exe| %FILEPATH%| %DIRPATH%\attachments\%FILENAME%%EXTENSION%)" visible="" enabled="TRUE" text="Start
DemoProcess" icon="" />
```

Parameterübergabe an das Programm

An das aufzurufende Programm können Parameter übergeben werden. Diese müssen mit ein | direkt an den Dateipfad angehängt werden.

Prozessaufrufe testen

Um den Aufruf von externen Programmen durch Primus zu testen gibt es ein Programm **DemoProcess.exe**. Das Programm listet alle übergebenen Aufrufparameter auf. Man verwendet dieses Programm anstelle der eigenen Exe.

Siehe auch

- [prmStartProcessByPath](#)

[menu](#), [gui](#), [kontext](#)

From:

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

Permanent link:

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

Last update: **2024/08/14 08:47**



prmStartProcessByPath

Mit dem [Menübefehl](#) **prmStartProcessByPath** können externe Programme über einen Menüeintrag aufgerufen werden. Der Prozess wird nur mit der zu Beginn von **parameter** angegebenen Datei durchgeführt und nicht wie bei [prmStartProcess](#) für alle selektierten. Dies ist sinnvoll wenn der Prozessaufruf innerhalb eines [Workflows](#) stattfindet.

Attribute

Attribut	Wert	Funktion
enabled	Bedingung	Steuert ob der Befehl aktiv ist
	TRUE	Aktiv
	FALSE	Inaktiv
icon		Pfad zum Icon (optional)
parameter	1 %FILEPATH%	Gibt den Pfad der zu verarbeitenden Datei an
		Abtrennung der beiden Parameter
	2 STARTH()	Programm wird ohne sichtbares (hidden) Fenster aufgerufen
	2 STARTV()	Programm wird mit sichtbarem (visible) Fenster aufgerufen
text		Elementtext für den Menüeintrag
visible	Bedingung	Steuert ob der Befehl im Menü sichtbar ist
	TRUE	Sichtbar
	FALSE	Unsichtbar

Beispiel

```
<Call method="prmStartProcessByPath"
parameter="%FILEPATH%|STARTH(C:\Unitec\SEAdvancedSave\SEAdvancedSave.exe|input|%FILEPATH%|/export|pdf|/OpenDraftInactive)"
condition="$(Condition.isSEDraft)" />
```

Parameterübergabe an das Programm

An das aufzurufende Programm können Parameter übergeben werden. Diese müssen mit ein | direkt an den Dateipfad angehängt werden.

Prozessaufrufe testen

Um den Aufruf von externen Programmen durch Primus zu testen gibt es ein Programm **DemoProcess.exe**. Das Programm listet alle übergebenen Aufrufparameter auf. Man verwendet dieses Programm anstelle der eigenen Exe.

Siehe auch

- [prmStartProcess](#)

[Workflow](#)

From:

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

Permanent link:

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

Last update: **2024/08/14 08:49**



prmUpdateXMLFieldsAndParentRefs

Der Menübefehl **prmUpdateXMLFieldsAndParentRefs** erweitert fehlende Datenfelder in den XMLs laut `<XML_DataTemplates>` `<XML_Document>` `<Data>` in der `CFG.xml` und wenn nicht `parameter=„0“` gesetzt ist, wird auch die Komponentenliste und ParentRefsliste aktualisiert (das dauert natürlich etwas länger bei vielen Verwendungen z.B. Normteile).

Attribute

Der Befehl kennt folgende Attribute:

Attribut	Wert	Funktion
enabled	Bedingung	Steuert ob der Befehl aktiv ist
	TRUE	Aktiv
	FALSE	Inaktiv
icon		Pfad zum icon (optional)
parameter		Parameterbeschreibung
text	<code>\$Primus.MNU_UPDATE_XMLS</code>	Interfacetext für den Menüeintrag
visible	<code>CALL(IsCtrlKey)</code>	Steuert ob der Befehl im Menü sichtbar ist, normalerweise nur bei gedrückter STRG-Taste
	TRUE	Sichtbar
	FALSE	Unsichtbar

Parameter

Mit dem [Attribut](#) `parameter` können folgende Werte in der angegebenen [Reihenfolge](#) übergeben werden. Die einzelnen Parameter werden durch ein Pipezeichen (|) getrennt.

Nr.	Wert	Parameter	Funktion
1	mit Komponenten- und ParentRefsliste	1	Der Parameter muss ungleich 0 (Null) sein

Beispiel

```
<<!-- Alle XML Dateien werden entsprechend <XML_DataTemplates> erweitert-->
<prmUpdateXMLFieldsAndParentRefs visible="'CALL(IsProjectActive)'"
enabled="TRUE" parameter="" text="$Primus.MNU_UPDATE_XMLS" icon="" />
<<!-- dito. und die Komponentenliste sowie die ParentRefsliste werden
aktualisiert-->
<prmUpdateXMLFieldsAndParentRefs visible="'CALL(IsProjectActive)'"
enabled="TRUE" parameter="1" text="$Primus.MNU_UPDATE_XMLS" icon="" />
```


Siehe auch

- [prmProjectRemoveAllUnusedParentRefs](#)
- [prmProjectUpdateAllRefs](#)

[GUI](#), [Menübefehl](#), [Administrator](#)

From:

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

Permanent link:

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

Last update: **2023/08/11 15:22**



ProcessCalls

Die Definition von **ProcessCalls** ermöglicht **Calls**-Listen nach bzw. vor bestimmten Prozessen auszuführen. Dabei werden beim Auftreten eines bestimmten Ereignisses, automatisch die definierten Listen abgearbeitet. Die ProcessCalls werden im Tag <ProcessCalls> im Abschnitt **Public** definiert. Darunter gibt es zwei Subtags <BeforeProcessCalls> für den Aufruf vor einem und <AfterProcessCalls> nach einem Ereignis. Für jedes Ereignis gibt es dann jeweils ein Tag unter dem alle Calls zusammengefasst werden.

Liste der Ereignisse

BeforeProcessCalls	
New	Bevor ein neues Dokument erzeugt wird
Copy	Bevor ein Dokument kopiert wird
Move	Bevor ein Dokument verschoben wird
Rename	Bevor ein Dokument umbenannt wird
Delete_recyclebin	Bevor der ein Dokument gelöscht und in den Papierkorb verschoben wird Klärungsbedarf
AfterProcessCalls	
New	Nachdem ein neues Dokument erzeugt wurde
Copy	Nachdem ein Dokument kopiert wurde
Copy_SMARTCOPY	Nachdem ein Dokument mit SmartCopy kopiert wurde
Rename_SMARTCOPY	Nachdem ein Dokument mit SmartCopy umbenannt wurde
Move_SMARTCOPY	Nachdem ein Dokument mit SmartCopy verschoben wurde
Delete_recyclebin	Nachdem ein Dokument gelöscht und in den Papierkorb verschoben wurde
Restore_recyclebin	Nachdem ein Dokument aus dem Papierkorb wiederhergestellt wurde
Delete	Nachdem ein Dokument gelöscht wurde Klärungsbedarf

Beispiele

```
<ProcessCalls>
  <AfterProcessCalls>
    <New>
      <Call method="CustomerAddIn.custWorkerLinkDocToVirtTPs"
parameter="%FILEPATH%|LinkDocPhToVirtTPs" condition="TRUE"/>
    </New>
    <Copy>
      <Call method="CustomerAddIn.custWorkerLinkDocToVirtTPs"
parameter="%FILEPATH%|LinkDocVrToVirtTPs" condition="TRUE"/>
      <Call method="CustomerAddIn.custWorkerLinkDocToVirtTPs"
parameter="%FILEPATH%|LinkDocPhToVirtTPs" condition="TRUE"/>
    </Copy>
    <Rename>
      <Call method="CustomerAddIn.custWorkerLinkDocToVirtTPs"
parameter="%FILEPATH%|RelinkDocVrInVirtTPs|%OLDFILEPATH%" condition="TRUE"/>
    </Rename>
  </AfterProcessCalls>
</ProcessCalls>
```

```
</Rename>
<Move>
  <Call method="CustomerAddIn.custWorkerLinkDocToVirtTPs"
parameter="%FILEPATH%|RelinkDocVrInVirtTPs|%OLDFILEPATH%" condition="TRUE"/>
  <Call method="CustomerAddIn.custWorkerLinkDocToVirtTPs"
parameter="%FILEPATH%|LinkDocPhToVirtTPs" condition="TRUE"/>
</Move>
<Copy_SMARTCOPY>
  <Call method="CustomerAddIn.custWorkerLinkDocToVirtTPs"
parameter="%FILEPATH%|LinkDocVrToVirtTPs" condition="TRUE"/>
  <Call method="CustomerAddIn.custWorkerLinkDocToVirtTPs"
parameter="%FILEPATH%|LinkDocPhToVirtTPs" condition="TRUE"/>
</Copy_SMARTCOPY>
<Rename_SMARTCOPY>
  <Call method="CustomerAddIn.custWorkerLinkDocToVirtTPs"
parameter="%FILEPATH%|RelinkDocVrInVirtTPs|%OLDFILEPATH%" condition="TRUE"/>
</Rename_SMARTCOPY>
<Move_SMARTCOPY>
  <Call method="CustomerAddIn.custWorkerLinkDocToVirtTPs"
parameter="%FILEPATH%|RelinkDocVrInVirtTPs|%OLDFILEPATH%" condition="TRUE"/>
  <Call method="CustomerAddIn.custWorkerLinkDocToVirtTPs"
parameter="%FILEPATH%|LinkDocPhToVirtTPs" condition="TRUE"/>
</Move_SMARTCOPY>
<Delete_recyclebin>
  <Call method="CustomerAddIn.custWorkerLinkDocToVirtTPs"
parameter="%FILEPATH%|DeleteDocVrInVirtTPs" condition="TRUE"/>
</Delete_recyclebin>
<Restore_recyclebin>
  <Call method="CustomerAddIn.custWorkerLinkDocToVirtTPs"
parameter="%FILEPATH%|LinkDocVrToVirtTPs" condition="TRUE"/>
</Restore_recyclebin>
</AfterProcessCalls>
</ProcessCalls>
```

Siehe auch

- [NamedCalls](#)
- [StatusWorkflow_Settings](#)

From:

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

Permanent link:

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

Last update: **2023/06/20 10:06**



Public

Im **Abschnitt Public** werden globale Definitionen abgelegt, die von verschiedenen Primus Funktionen genutzt und benötigt werden. Es stehen verschiedene **Tags** zur Verfügung. Jedes Tag hat verschiedene **Attribute**, diese werden der Übersichtlichkeit wegen in verschiedenen Zeilen desselben Tags abgelegt.

Liste der Tag-Namen

Tag	Pflicht	Eindeutig	Funktion
NewFileNameDialog	Ja	Nein	Definiert welches Datenblatt beim Neuanlegen verwendet werden soll
NewFileNameFormat	?	?	klaerungsbedarf
Workflow	Ja	Nein	Definiert was beim Anlegen einer Neuen Revision geschehen soll
ReservationCall	?	?	klaerungsbedarf
RelationshipsTab	Ja	Ja	Definiert wie das Beziehungsfenster standardmäßig angezeigt werden soll
History	Ja	Ja	Relativer Ablagepfad für ältere Versionen (Aktueller Pfad = path=" . ")
Attachments	Ja	Ja	relativer Pfad für Dateianhänge
AttachmentFiles	Ja	Ja	Definiert das Pattern für die Auflistung von Anhängen
JobServer	Ja	Ja	Ablagepfad für Jobs (relativ oder absolut)
ReplaceTitleBlockTemplate	?	?	klaerungsbedarf
Condition	Nein	Nein	Definiert eine Bedingung , die in der Konfiguration an anderer Stelle genutzt werden kann
DocumentOpen	Nein	?	Steuert diverse Optionen beim Öffnen von Dokumenten
DocumentOpenCall	Nein	Nein	Definiert wie ein bestimmter Dokumenttyp mit einem Primus Addin geöffnet werden soll
DocumentOpenIgnoreReadOnlyCond	Nein	Ja	Definiert die Dateitypen und die Bedingungen, bei denen keine Schreibschutzmeldung angezeigt wird
DocumentRefListCall	Ja	Nein	Definiert den Addin Aufruf für verknüpfte Dokumente
Reservation	Nein	Nein	Steuert die automatische Reservierung beim Öffnen mit Solid Edge
SmartCopy	Nein	Nein	Hier werden allgemeine Einstellungen zu SmartCopy in diversen Attributen hinterlegt
SmartCopyFileName	Nein	Nein	Definiert wie der Dateiname bei SmartCopy -Operationen aufgebaut ist
SEInterface	Nein	Nein	Hier werden allgemeine Einstellungen für das Solid Edge Interface in diversen Attributen hinterlegt
ISLFilename	klaerungsbedarf		

Tag	Pflicht	Eindeutig	Funktion
ISLPrimusaddin	Klaerungsbedarf		
MSOInterface	Nein	Nein	Hier werden allgemeine Einstellungen für das MS-Office Interface in diversen Attributen hinterlegt
SendEMAIL_FILES	Nein	Nein	Definiert die Attribute für den Email Versand aus Primus
SendEMAIL_FORCHECKING	Nein	Nein	Definiert die Attribute für den Email Versand in einem Workflow

Liste

From:

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

Permanent link:

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

Last update: **2023/04/25 10:09**



ReplicationService

Im [Abschnitt ReplicationService](#) wird definiert wie der [PrimusReplicationService](#) (PRS) zu erreichen ist und an welchem Standort sich dieser Client befindet.

Folgende Attribute sind erforderlich:

Attribut	Funktion
server	Netzwerkname des Servers auf dem der Replikationsdienst läuft
port	TCP-IP Port über den der Replikationsdienst auf dem Server erreichbar ist
location	Standortkürzel des Clients, dieses muß mit dem Eintrag in der PrimusReplicationService.xml übereinstimmen. Der Wert kann mit dem Platzhalter %LOCATION% zb. zur Generierung der ReplD abgerufen werden.

Beispiel

Der Server ist unter dem [DNS_Namen](#) Primus erreichbar, der PRS ist mit Port 9091 verbunden. Der Standort des Clients ist **HU**.

```
<ReplicationService server="primus" port="9091" location="HU"/>
```

Menübefehle

Die Replikation kann über folgende [Menübefehle](#) gesteuert werden:

Befehl	Funktion
prmReplicationServiceTestConnection	Testet die Verbindung zum PRS
prmReplicationServiceStartTimer	Klärungsbedarf
prmReplicationServiceStopTimer	
prmReplicationServiceReloadSettings	Lädt die Replikationseinstellungen neu
prmReplicationServiceGetStatus	Fragt den Status des PRS ab
prmProjectSetDirectoriesToReplicate	Ermöglicht es ein Verzeichnis oder eine Struktur als replizierbar zu definieren
prmProjectGenerateReplIDs	Generiert die für die Replikation notwendigen eindeutigen IDs für die gewählten Verzeichnisse
prmProjectReplicateDirectories	Repliziert eine leere Verzeichnisstruktur
prmProjectReplicateDocuments	Repliziert ausgewählte Dokumente
prmProjectRequestLocationForDocuments	Fordert den aktuellen Standort als Bearbeitungsstandort für die gewählten Dokumente an
prmElementReplicateDocumentWithComponents	Repliziert ein Element nebst Verknüpfungen und ggf. übergeordneten Elementen

Befehl	Funktion
prmElementRequestLocationWithComponents	Legt den Bearbeitungsstandort für ein Element nebst Verknüpfungen und ggf. übergeordneten Elementen auf den aktuellen Standort fest
prmElementGenerateReplIDs	Generiert die für die Replikation notwendigen eindeutigen IDs für die gewählten Elemente

Tipp

Da es bei einer Replication auch immer mehrere Server gibt, die jeweils einen anderen **ReplicationService** Eintrag in der [Konfiguration](#) haben müssen, kann man bei Konfigurationsänderungen diese nicht einfach auf einen anderen Server kopieren. Zu diesem Zweck empfiehlt es sich mit [IncludeConfigLayers](#) zu arbeiten. Dabei steht in der Konfiguration nur das IncludeConfigLayers Tag und der eigentliche Eintrag liegt in einer separaten Datei. Diese ist für jeden Server eine andere und wird jeweils nicht mitkopiert, muß aber i.d.R. auch nicht geändert werden.

Beispiel

```
<!-- Da für die Replication jeder Server einen eigenen Eintrag für seinen Standort braucht, ist dieser nicht direkt in dieser CFG.XML abgelegt, sondern in einer CFG LAYER.XML die den jeweiligen Standort konfiguriert. Dadurch kann bei Anpassungen an dieser Konfiguration diese einfach auf den anderen Server übertragen werden. -->
<IncludeConfigLayers>
  <Layer path=".\\CFGLayers\\ReplicationService.CFG LAYER.xml" />
</IncludeConfigLayers>
```

[Replikation](#), [PrimusService](#), [Tipps&Tricks](#)

From:

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

Permanent link:

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

Last update: **2024/07/09 10:57**



Replikations ID

Bei der **Replikations ID** kurz **RepID**, handelt es sich um eine über alle Standorte eindeutige und sich nicht mehr verändernde Kennzeichnung für jede zur Replikation vorgesehene Datei. Der Name des Datenfelds in [XML_DataTemplates](#) lautet RepID. Der Replikationsservice überprüft anhand der RepID in der [Replikationsanforderung](#) ob diese mit der zu replizierenden Datei übereinstimmt. Kommt es hier zu Abweichungen, dann wird die Replikationsanforderung nicht verarbeitet und bleibt im Übergabeverzeichnis stehen.

Wertzuweisung

Die RepID muß für jede Datei eindeutig sein, das wird erreicht in dem man eine [GUID](#) als Wert verwendet. Ein Dokument erhält nur dann eine RepID wenn das Verzeichnis ([Project](#)) in dem es abgespeichert ist auf replizierbar gesetzt wurde. Das geschieht mit der [Bedingung](#) %XML_Project:Replicate%==1. In Primus ist es üblich, zusätzlich den Standort an dem die Datei erstellt wurde anzugeben. Primus bietet dazu folgende Platzhalter an:

Platzhalter	Beschreibung
%NEWGUID%	Erzeugt eine neue GUID
%LOCATION%	Liefert den unter ReplicationService im Attribut location eingetragenen Standort

Die RepID setzt sich dann aus der LOCATION einem # und der NEWGUID zusammen.

Beispiel

```
<RepID value="%LOCATION%#%NEWGUID%" condition="'%XML_Project:Replicate%==1' " />/>
```

Besonderheiten

Da die RepID eindeutig sein muß ist es wichtig bei der Zuweisung darauf zu achten das diese z.B. bei Kopieroperationen nicht mit kopiert sondern geleert bzw. neu erzeugt wird. Auch ist sicherzustellen das sich die RepID niemals ändern darf, da sonst die Repliken an anderen Standorten nicht mehr aktualisiert werden. An den folgenden Stellen werden Zuweisungen der RepID vorgenommen, bzw. dürfen sie nicht vorgenommen werden:

Abschnitt	Wertzuweisung	Beschreibung
XML_DataTemplates	Nein	Keine Zuweisung wenn eine XML zu einem Dokument angelegt wird. Ob ein Dokument replizierbar ist kann später festgelegt werden.
XML_Document		
XML_DocumentAttachments		
PreAssignData		

Abschnitt		Wertzuweisung	Beschreibung
	AssignDocumentDataFields	Nein	Hier darf <u>keinesfalls</u> ein Wert zugewiesen werden, da Primus diese Einstellung nutzt um eine XML zu jedem Dokument zu erzeugen das keine hat. Arbeiten zwei Clients in ein und demselben Verzeichnis kann es sein das der eine Client eine Datei erzeugt und bevor er die dazugehörige XML anlegen kann erzeugt diese ein anderer Client. Dabei kann zweimal hintereinander eine RepID erzeugt werden, wobei dann möglicherweise die falsche RepID im Replikationsauftrag steht.
	AssignDocumentDataFields_NewDrawing	Ja	Wird genutzt wenn eine neue Zeichnung mittels Primus angelegt wird
	AssignDocumentDataFields_ADDIN	Ja	Wird vom Solid Edge Addin genutzt
	AssignDocumentDataFields_COPY	Ja	Wenn von einem Dokument eine Kopie erzeugt wird
	AssignDocumentDataFields_SMARTCOPY	Ja	Wenn von einem Dokument eine Kopie mit SmartCopy erzeugt wird
	AssignNewVersionDataFields	Ja	Beim erzeugen einer neuen Version
	AssignAttachmentsDataFields	Ja	Für Attachments
	AssignExportJobDocumentDataFields	Ja	Klärungsbedarf
	AssignExportJobAttachmentsDataFields	Ja	

Alle nicht explizit aufgeführten Umbenenn- und Verschiebeoperationen haben keinen Einfluss auf die RepID, diese ändert sich in diesen Fällen nicht. Da weder in XML_Document noch in AssignDocumentDataFields eine RepID zugewiesen wird, hätte dies zur Folge das direkt mit Primus erzeugte Dokumente grundsätzlich nicht repliziert würden. Daher wird zusätzlich ein [Process Call](#) angelegt, der bei neu über die Primusoberfläche angelegten Dokumenten anschließend eine RepID generiert.

Beispiel

```
<ProcessCalls>
  <AfterProcessCalls>
    <New>
      <!-- Erzeugt eine RepID nachdem über die Primusoberfläche ein neues
Dokument erstellt wurde -->
      <Call method="SetXMLValue"
parameter="%FILEPATH%|RepID|%LOCATION%##NEWGUID%"
condition="'%XML_Project:Replicate%==1'" />
```

```
</New>
</AfterProcessCalls>
</ProcessCalls>
```

Siehe auch

- [ReplicationService](#)
- [prmProjectSetDirectoriesToReplicate](#)
- [prmProjectGenerateRepIDs](#)

[Replikation](#)

From:

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

Permanent link:

https://dokuwiki.unitec.de/doku.php?id=primus:replikations_id

Last update: **2024/07/09 12:42**



Reservation

Mit **Reservation** wird im [Abschnitt Public](#), definiert wie mit Reservierungen beim Öffnen eines Dokuments verfahren werden soll.

Attribute

Attribut	Wert	Bedeutung
user_list	<LISTENNAME>	Gibt eine Liste mit Benutzernamen an
remove_condition	<BEDINGUNG>	Gibt eine Bedingung an, wann eine Reservierung zurückgenommen wird
set_condition		Bedingung wann eine Reservierung gesetzt werden darf

Beispiel

```
<Reservation set_condition="'#(WorkflowStatus)==001'" user_list="List_User" />
```

[options](#), [public](#), [Klärungsbedarf](#), [Replikation](#)

From:

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

Permanent link:

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

Last update: **2023/02/08 12:01**



ReservationCall

Im Eintrag **ReservationCall** im Abschnitt [Public](#) wird verwendet um zu steuern wie der Status einer [Solid Edge](#) Datei, bei manueller Reservierung in Primus, gesetzt werden soll.

Beispiel

```
<ReservationCall  
condition="($(Condition.isSEDocument))AND'#(WorkflowStatus)==001'"  
set="SEInterface.seSetSolidEdgeFileStatusByPath 1"  
remove="SEInterface.seSetSolidEdgeFileStatusByPath 0" />
```

Reservierung in Primus ohne Solid Edge Statusänderung

Es ist möglich den Status in Solid Edge nicht zu setzen aber trotzdem die Primus Reservierung zu setzen.

```
<PrimusSEAddIn_Settings reserv_sestatus="0" <ReservationCall condition="0"  
set="" remove=""
```

Dabei wird einfach beim Versuch eine reservierte Datei von einem anderen Benutzer zu öffnen der Windows-Schreibschutz gesetzt.

[CADAdaptApproved](#)

Siehe auch

- [PrimusSEAddIn](#)

[Public](#)

From:

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

Permanent link:

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

Last update: **2025/03/14 16:39**



SearchServiceDB

Im [Abschnitt SearchServiceDB](#) wird definiert wie die Primus Clients mit dem [PrimusSearchService](#) (PSS) kommunizieren können. Die Verbindung zu einer bestimmten Indexdatenbank wird als [Tag](#) definiert. Der [Tag-Name](#) ist der Name der Datenbank. Die Parameter werden als Attribute übergeben. Folgende Attribute sind möglich:

Attribut	Funktion
server	Netzwerkname des Servers mit dem PSS
port	TCP-IP Port über den der PSS auf dem Server erreichbar ist
offline_calls	Verzeichnis in dem Clients Indexänderungen ablegen können wenn der Dienst offline ist

Beispiel

Der Name der Indexdatenbank ist PDM. Der Server ist unter dem [DNS_Namen](#) Primus erreichbar, der PSS ist mit Port 9090 verbunden. Für die Offline Calls gibt es eine Freigabe [\\primus\offline_calls](#).

```
<SearchServiceDB>  
  <PDM server="primus" port="9090" offline_calls="\\primus\offline_calls" />  
</SearchServiceDB>
```

[PrimusService](#)

From:

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

Permanent link:

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

Last update: **2025/06/02 07:54**



seCheckForStatusChangeIfDocumentOpen

Mit dem Befehl **seCheckForStatusChangeIfDocumentOpen** des [SEInterface Addin](#), wird im PreCall eines [Workflows](#) geprüft ob das Dokument im [Solid Edge](#) offen ist. Falls das der Fall ist wird folgende Meldung gezeigt: „*Opened document will be closed after status change:*“. Dazu wird der fest vorgegebene Interfacetext **\$Primus.MSG_CHECKSTATUSCHANGE_FOR_OPENEDDOC** verwendet. Wenn der Benutzer OK klickt dann geht er weiter mit Status-Wechsel sonst bricht er ab. Hier wird davon ausgegangen, dass die offene Datei nach der Freigabe mit PostCall Close geschlossen wird.

Liste der Parameter

Der Funktion müssen folgende Parameter, durch | getrennt, in der angegebenen Reihenfolge übergeben werden.

Parameter	Funktion
1 Dokumentpfad	Pfad des Dokuments, z.B. %FILEPATH%

[SEInterface](#), [Addin](#), [Befehl](#), [Workflow](#), [Status](#), [PreCalls](#)

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

Permanent link:
<https://dokuwiki.unitec.de/doku.php?id=primus:secheckforstatuschangeifdocumentopen>

Last update: **2026/01/13 09:13**



seCheckSolidEdgeApplication

Die **funktion seCheckSolidEdgeApplication** des **SEInterface Addin**, versucht **solid_edge** zu verbinden und wenn es nicht geht dann zu starten. Wenn Starten auch nicht geht, dann gibt es eine Fehlermeldung und FALSE zurück. Es ist jedoch zu berücksichtigen, dass die Rückmeldung von Solid Edge, wenn keine Lizenz vorhanden ist ziemlich lange dauert (ca. 1-3 Minuten).

Liste der Parameter

Der Funktion müssen folgende Parameter, durch | getrennt, in der angegebenen Reihenfolge übergeben werden. derzeit sind keine Parameter vorgesehen.

Parameter	Funktion
1	z.Zt. nicht definiert

Rückgabewert

Wert	Bedeutung
TRUE	Solid Edge ist gestartet
FALSE	Solid Edge ist nicht gestartet

Beispiel

```
<PreCalls>
  <Call method="SEInterface.seCheckSolidEdgeApplication" parameter=""
condition="'$(Condition.isSEDocument)'" />
```

Zweck

Die Funktion kann als Bedingung in PreCalls eines **Workflow**s verwendet werden um zu prüfen um sicher zu stellen das nachfolgende Prozesse, die Solid Edge benötigen, durchgeführt werden können.

[SEInterface](#), [Addin](#), [Funktion](#), [Workflow](#), [Status](#), [Bedingung](#)

From:

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

Permanent link:

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

Last update: **2023/02/08 12:01**



seCloseAndSave

Mit dem Befehl **seCloseAndSave** des [SEInterface Addin](#), kann eine zuvor geöffnete [Solid Edge](#) Datei gespeichert und geschlossen werden. Der Aufruf kann mittels [Call](#) im Rahmen eines [Workflows](#) erfolgen, nachdem die Datei zuvor z.B. mit [seUpdatePropertiesByPath](#) geöffnet wurde.

Liste der Parameter

Der Funktion müssen folgende Parameter, durch | getrennt, in der angegebenen Reihenfolge übergeben werden.

	Parameter	Funktion
1	Dokumentpfad	Pfad des Dokuments, z.B. %FILEPATH%. Damit ist sichergestellt das nur dieses Dokument und nicht auch andere geöffnete Dokumente geschlossen werden.

Beispiel

Im gezeigten Beispiel wird eine zuvor geöffnete Solid Edge Zeichnungsdatei gespeichert und geschlossen.

```
<Call method="SEInterface.seCloseAndSave" parameter="%FILEPATH%"
condition="$(Condition.isSEDraft)" />
```

Siehe auch

- [seSave](#)
- [seClose](#)
- [seCloseSEAfter](#)

[SEInterface](#), [Addin](#), [Befehl](#), [Workflow](#), [Status](#)

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

Permanent link:
<https://dokuwiki.unitec.de/doku.php?id=primus:secloseandsave>

Last update: **2023/07/28 12:34**



seCopySEMaterialsToCFG

Die Methode **seCopySEMaterialsToCFG** des **SEInterface Addin**, liest die **solid_edge** Materialtabelle aus und erstellt daraus Einträge in einer Primus **Liste**. Der Ablagepfad der Materialtabelle ist der in Solid Edge, unter Dateiablagen, hinterlegte Pfad.

Liste der Parameter

Der Funktion müssen folgende Parameter, durch | getrennt, in der angegebenen Reihenfolge übergeben werden. derzeit sind keine Parameter vorgesehen.

	Parameter	Funktion
1	<DATEIPFAD>	Pfad zur Primus Listendatei inkl. vollständigem Dateiname
2	<SE MATERIALTABELLE>	Name der Solid Edge Materialtabelle ohne Dateiendung

Beispiel

```
<seCopySEMaterialsToCFG addin="SEInterface" visible="TRUE" enabled="TRUE"
parameter="\\primus\setup\Resources\List_Material.xml|Materials"
text="$Primus.MNU_COPY_SE_MATERIAL_TOCFG"
icon="Resources\AppInterfaces\SolidEdge\Icons\segetproperties"/>
```

Zweck

Mit dieser Methode kann ein **Menübefehl** eingerichtet werden, mit dem auf Anforderung des Anwenders, die Materialtabelle aus Solid Edge an Primus übertragen werden. Dadurch wird vermieden, dass Änderungen sowohl in Solid Edge als auch in Primus vorgenommen werden müssen. Solid Edge ist hierbei das führende System.

[SEInterface](#), [Addin](#), [Methode](#), [Menübefehl](#)

From:

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

Permanent link:

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

Last update: **2023/02/08 12:01**



seCreateExportJob

Bei **seCreateExportJob** handelt es sich um einen [Menübefehl](#) des Primus [Addins SEJob_Export](#).
Damit können Jobs zum exportieren von Exportformaten durch [Solid Edge](#) erzeugt werden.

Parameter

Die Parameter müssen in folgender Reihenfolge und durch | getrennt übergeben werden.

Parameter	Werte	Bemerkung
0	JobServerPath	Pfad zum JobServer
1	DPI	Auflösung bei Bildformaten
2	TargetFileFormat	Ausgabeformat (TIF, DXF, STP, ...)
3	TargetPath	Ablagepfad der Exportdatei
4	TargetXMLData	Zu verwendendes XML-DataTemplate (optional)
5	SINGLESHEETNAME	Es wird nur bei mehr als einem Blatt, der Blattname angehängt
6	Ext	Bestimmt welche Dateien bei einer Mehrfachauswahl verarbeitet werden sollen z.B.,,.dft"

XML DataTemplate

Wenn XML DataTemplate nicht angegeben ist, dann werden keine AssignExportJobAttachmentsDataFields ausgeführt. Es wird nur die Exportdatei angelegt und die XML wird später beim ersten Auflisten in einem Primus-Client ohne [PreAssignMappings](#) erstellt.

[CADAdaptApproved](#)

Erstellen mehrere Formate in einem Job

Es können mehrere Formate in einem Job erstellt werden, dazu werden die einzelnen Formate (Parameter 2) und optional die Ablagepfade (Parameter 3), durch ein Paragraph (§) getrennt.²⁰⁾

Beispiel

```
<seCreateExportJob addin="SEJob_Export" visible="$(Condition.isSEDraft)"
enabled="TRUE"
parameter="$(JobServer.path)|300|TIF|\\primus\DataConnect\TiffOut\#(UD-PPS-
ID:replace#.|_)-#(Revision)|0||.dft"
text="$Primus.MNU_CREATE_EXPORT_JOB_TIF" icon=""/>
<!-- Mehrfachverarbeitung mittels Button ab Primus_Export_SE_Job.dll vom
14.06.2024 -->
<seCreateExportJob ctrltype="button" data="SEJob_Export.seCreateExportJob
$(JobServer.path)|300|SE-&gt;TIF$DXF|\\primus\DataConnect\TiffOut\#(UD-PPS-
```

```
ID:replace#.!_) -#(Revision)$\\primus\DataConnect\DXF\#(UD-PPS-  
ID:replace#.!_) -#(Revision)|0" label="ERP - Tiff+DXF Export" type="Button"  
mode="W" left="163" top="399" width="150" height="22"/>
```

Siehe auch

- [seExportJobByPath](#)

Menübefehl

20)

Diese Funktion steht erst ab der *Primus_Export_SE_Job.dll* vom 14.06.2024 zur Verfügung

From:

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

Permanent link:

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

Last update: **2025/06/05 07:31**



seExportJobByPath

Bei **seExportJobByPath** handelt es sich um eine [Methode](#) des Primus [Addins SEJob_Export](#), die mittels [Call](#) aufgerufen wird. Damit können Jobs zum exportieren von Exportformaten durch [Solid Edge](#) erzeugt werden.

Parameter

Die Parameter müssen in folgender Reihenfolge und durch | getrennt übergeben werden.

Parameter	Werte	Bemerkung
1	<Dokument>	Vollständiger Pfad auf das zu exportierende Dokument
2	<Auflösung> / <Modus>	Die Auflösung und ggf. Farbe bei Pixelformaten, bzw. Modus bei Vektorformaten
3	<Exportformat>	Dateiformat in das exportiert werden soll (Liste siehe unten)
4	<Ziel>	Ablagepfad und Name der Exportdatei
5	<XML DataTemplate>	Zu verwendendes XML-DataTemplate (optional)
6	<Dateinamensergänzung>	Mögliche Ergänzung des Dateinamens (typabhängig, optional)
	<NoMSG>	Mit der Angabe NOMSG kann die Ausgabe einer MessageBox unterdrückt werden, es wird dann nur ein Eintrag im Log gemacht.
		Beide Parameter dürfen auch kombiniert werden

Liste der Exportformate

Folgende Exportformate können erzeugt werden.

Format	Auflösung / Modus	Dateinamensergänzung (optional)	Beschreibung
DXF / DWG	0		Erzeugt ein einzelnes Dokument mit allen Blättern, ohne den Dateinamen zu ergänzen
	1		Erzeugt für jedes Blatt ein eigenes Dokument und ergänzt es um den Blattnamen
	0	SINGLESHEETNAME	Es wird nur bei mehr als einem Blatt, der Blattname angehängt
DXF/PSM			Erzeugt eine DXF Abwicklung eines Blechteils

Format	Auflösung / Modus	Dateinamensergänzung (optional)	Beschreibung
SE→JPG	Auflösung;Breite;Höhe		Erzeugt ein JPG mit den angegebenen Werten. Die tatsächliche Bildgröße errechnet sich aus der Multiplikation der Abmessung mit der Auflösung.
SE→TIF			Erzeugt ein TIF mit den angegebenen Werten. Die tatsächliche Bildgröße errechnet sich aus der Multiplikation der Abmessung mit der Auflösung.
SE→PDF	C300		Erzeugt ein farbiges PDF in 300dpi mit „Speichern unter“ aus Solid Edge heraus
	300		Erzeugt ein Schwarz-weiß PDF in 300 dpi mit „Speichern unter“ aus Solid Edge heraus
STP	0		Erzeugt eine STEP-Datei

Hinweis: Das > Zeichen ist innerhalb von [Attribut](#)werten verboten und muss im XML als > eingetragen werden, z.B. **SE->PDF**.

XML DataTemplate

Wenn XML DataTemplate nicht angegeben ist, dann werden keine AssignExportJobAttachmentsDataFields ausgeführt. Es wird nur die Exportdatei angelegt und die XML wird später beim ersten auflisten in einem Primus-Client ohne [PreAssignMappings](#) erstellt.

[CADAdaptApproved](#)

Kombinationen bei Parameter 6

Die Argumente von Parameter 6 können auch kombiniert werden, Reihenfolge und Trennzeichen sind egal. z.B.

SINGLESHEETNAMENOMSG
NOMSGSINGLESHEETNAME
NOMSG_SINGLESHEETNAME
NOMSG+SINGLESHEETNAME
...

Es darf nur nicht Leerzeichen oder Splitter | sein.

NOMSG SINGLESHEETNAME
NOMSG SINGLESHEETNAME

Beispiel

```
<Call method="SEJob_Export.seExportJobByPath"
parameter="%FILEPATH%|0|DXF|.Attachments\%FILENAME%%EXTENSION%\%FILENAME%
|XML_DocumentAttachments| SINGLESHEETNAME"
condition="$ (Condition.isSEDraft)" />
<!-- Erzeugt ein JPG mit einer Auflösung von 900x900 Pixeln -->
<Call method="SEJob_Export.seExportJobByPath"
parameter="%FILEPATH%|100;9;9|SE-
>JPG|.Attachments\%FILENAME%%EXTENSION%\%FILENAME%|NOMSG"
condition="$ (Condition.isSEModel)" />
```

Siehe auch

- [seCreateExportJob](#)

[Workflow](#), [Methode](#)

From:

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

Permanent link:

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

Last update: **2025/05/23 07:50**



SEInterface

Das **SEInterface** ist ein Primus [Addin](#), das Befehle zur Einbindung von [Solid Edge](#) in Primus zur Verfügung stellt.

Konfigurationsordner

Das Addin liegt bei einer [Primus Installation](#) unter [Appinterfaces\SolidEdge](#). Dort gibt es folgende Verzeichnisse:

Verzeichnis	Funktion
AddIn	Verzeichnis mit dem PrimusSEAddin , das beim Start von Solid Edge mit geladen wird
AppAttributes	Hier liegen die Konfigurationsdateien mit denen der Abgleich der Datenfelder von Primus und Solid Edge gesteuert wird
Icons	Enthält die Solid Edge spezifischen Icons
Templates	Enthält die Solid Edge Vorlagen die in GUI_NewDocument_DataSheet aufgelistet werden. Der Ablagepfad wird im Abschnitt Templates definiert.

Allgemeine Einstellungen

Im Abschnitt [Public](#) der [Primus.CFG.XML](#) gibt es einen Eintrag **SEInterface** in dem einige allgemeine Parameter des Addins mit Attributen konfiguriert werden können.

Liste der Attribute		
Attribut	Wert	Funktion
AppAttributesPath	.\AppAttributes	Verzeichnispfad zu den Ini-Dateien für den Eigenschaftsabgleich zwischen Solid Edge und Primus
DesMgr_Path	C:\Program Files\Siemens\Solid Edge 20XX\Program\	Pfad zur DesMgr.exe (Designmanager), er muss mit einem \ enden
OccurrenceTitleOFF	[%FILENAME%%EXTENSION%] # (Title)	Klärungsbedarf
OnSavePreCalls		Vorgänge die vor dem Speichern in Solid Edge von Primus durchgeführt werden sollen. Ähnlich SEAdvancedSave Klärungsbedarf
OnSavePostCalls		Wie OnSavePreCalls , jedoch für Prozesse die nach dem Speichern durchgeführt werden sollen.
Reservation_Status	0/1	Klärungsbedarf

Liste der Attribute		
Attribut	Wert	Funktion
PropertiesToPrimusOnOpen	0/1	Steuert ob die Dateieigenschaften beim Öffnen eines Solid Edge Dokuments an Primus übertragen werden sollen. Da dazu bei Baugruppen auch das Gewicht berechnet werden muss, kann dies zu einer niedrigen Performance beim Öffnen von Dateien führen.
UpdateWeightField	<DATENFELD>	Definiert das Datenfeld für die Masse aus dem 3D Modell
UpdateWeightFormat	kg (#,###0.000)	Definiert das Darstellungsformat der Gewichtsangabe
writexmlcond	0/1	Steuert ob bei jedem Speichern Gewicht übertragen werden soll, für manuelles Aktualisieren siehe seUpdateWeight

Liste der Befehle für den Aufruf aus einem Menü

Funktion	Beschreibung
seCopySEMaterialsToCFG	Auslesen der SE Materialien
seCreateDraft	Einfügen im Draft
seCreateRenameJob	Umbenennen
seCreateReplaceRefJob	Vorgemerkte ersetzen durch
seCreateUpdateSEDocJob	Dateieigenschaften und Ansichten aktualisieren
seGetAllProperties	SE Datei-Eigenschaften auslesen
	SE Datei-Eigenschaften neu auslesen
seGetProperties	Dateieigenschaften auslesen
seInsert	Einfügen im Solid Edge
seJobPreSelectDocuments	Vormerken zum Ersetzen
seOpenWithRevMan	Mit Design- bzw. Revision Manager öffnen
seOpen	Öffnet eine Dokument mit Solid Edge
seReplace	Ersetzen im Solid Edge
seSetSolidEdgeFileStatus	Ändert den Status der gewählten Solid Edge Datei auf einen bestimmten Wert
seUpdate3DDocument	Aktualisiert vom gewählten Dokument ausgehend, rekursiv die gesamten 3D Daten in der Struktur
seUpdateProperties	Dateieigenschaften aktualisieren
	Dateieigenschaften und Draftansichten aktualisieren
seUpdateSolidEdgeFileStatus	Ändert den Status in der Solid Edge Datei abhängig vom Status in Primus
seUpdateWeight	Aktualisiert für alle selektierten Dateien das Gewicht
Funktionen für den Aufruf aus einer SmartCopy Session heraus	
seReplaceDocRef	Ersetzt die Dateiverknüpfungen in der Solid Edge Datei
smartupdatepropsandrefsforlastsession	Aktualisiert die Dateieigenschaften im Solid Edge Dokument <u>und</u> die Referenzen in der Dokument-XML

Funktion	Beschreibung
smartupdatepropsforlastsession	Aktualisiert <u>nur</u> die Dateieigenschaften im Solid Edge Dokument
smartupdaterefsforlastsession	Aktualisiert <u>nur</u> die Referenzen in der Dokument-XML
Funktionen für den Aufruf mittels Call aus einem Workflow heraus	
CheckChildrenStatus	Prüft ob die Children Dokumente einen bestimmten Status haben
CheckChildrenWriteable	Prüft ob die Children Dokumente des Dokuments beschreibbar sind
CheckParentsWriteable	Prüft ob die Parent Dokumente des Dokuments beschreibbar sind
CheckXMLValue	Prüft ob ein Feld einen bestimmten Wert hat
CopyFile	Kopiert eine bestimmtes Dokument unter Angabe von Quell- und Zielpfad
IgnoreStatus	Gibt an welche Dokumente mit welchem Status ignoriert werden sollen
seCheckForStatusChangelfDocumentOpen	Prüft im PreCall ob das Dokument im SE offen ist und zeigt eine Meldung
seCheckSolidEdgeApplication	Prüft ob Solid Edge läuft und startet es ggf.
seClose	Schließt die angegebene Datei
seCloseSEAfter	Schließt Solid Edge nach einer vorgegebenen Anzahl von Aufrufen durch SEInterface
seSave	Speichert die angegebene Datei, passt ggf. vorher die Ansicht ein, schließt die Datei aber nicht
seSetSolidEdgeFileStatusByPath	Setzt den Status in einer Solid Edge Datei während diese <u>nicht</u> in Solid Edge geöffnet ist
SetChildrenToStatus	Setzt den Status für der Children Dokumente
SetXMLValue	Setzt ein bestimmtes Datenfeld eines Dokuments auf einen Wert
seUpdatePropertiesByPath	Öffnet eine Solid Edge Datei und aktualisiert die Eigenschaften
seCloseAndSave	Speichert eine mit seUpdatePropertiesByPath geöffnete Solid Edge Datei und schließt sie anschließend

Siehe auch

- [SEBOM2XLSAddIn](#)
- [SEJob_Export](#)
- [PrimusSmartCopy](#)
- [PrimusSEAddin](#)

[SolidEdge](#), [Addin](#), [Menu](#), [GUI](#)

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

Permanent link:
<https://dokuwiki.unitec.de/doku.php?id=primus:seinterface>

Last update: **2023/07/28 13:09**



SEJob_Export

Bei **SEJob_Export** handelt es sich um ein [Addin](#) für Primus das die Funktion des [JobServer](#) für [Solid Edge](#) zur Verfügung stellt. Die Funktionen des Addin werden in der [Primus_Export_SE_Job.xml](#) konfiguriert.

Primus_Export_SE_Job.xml

In der **Primus_Export_SE_Job.xml** wird der JobServer konfiguriert. Der Name des [Stammelements](#) lautet `Primus_Export_SE_Job_Settings`.

Attribut	Funktion
<code>reloadaftercalls</code>	Definiert nach wieviel Jobs Solid Edge beendet werden soll. Bei 1 wird Solid Edge nach jedem Job beendet.

SEToPrinter_Format_Mappings

Attribut	Funktion
<code>lastprinter</code>	Klärungsbedarf
<code>lastformat</code>	

SEToPrinter

Attribut	Funktion
<code>name</code>	Name des Druckers

Papierformate

Unter **Papierformate** werden Tags mit den einzelnen Papierformaten angelegt. Der [Tag-Name](#) bezieht sich auf einen Eintrag im Abschnitt [SolidEdge_PaperFormats](#).

Attribut	Funktion
<code>printer_format</code>	Papiergrößenbezeichnung des Druckers

```
<IsoA4Tall printer_format="ISO A4" />
```

PaperFormatToPrinter

[Klärungsbedarf](#)

Papierformate

Unter **Papierformate** werden Tags mit den einzelnen Papierformaten angelegt. Der [Tag-Name](#) bezieht sich auf einen Eintrag im Abschnitt [SolidEdge_PaperFormats](#).

Attribut	Funktion
printer_name	Name des Druckers
printer_format	Papiergrößenbezeichnung des Druckers

```
<IsoA4Tall printer_name="HP Officejet Pro 8620 (Netzwerk)"
printer_format="A4" />
```

SolidEdge_PaperFormats

In **SolidEdge_PaperFormats** werden den Konstanten mit denen Solid Edge die Papiergrößen definiert, Namen zugeordnet. Die einzelnen Einträge haben den Tag-Namen Format.

Attribut	Funktion
const	Solid Edge interne Konstante für die Papiergröße ²¹⁾
name	Name der Papiergröße. In den Abschnitten SEToPrinter und PaperFormatToPrinter wird auf diesen Namen Bezug genommen

Beispiel

```
<Format const="28" name="IsoA4Tall" />
```

Liste der Methoden

Die von diesem Addin zur Verfügung gestellten [Methoden](#) können über [Call](#) z.B. aus [Workflows](#) heraus aufgerufen werden.

Methode	Funktion
seCreateReplaceTitleblockJob	Erstellt einen Job zum Ersetzen des Schriftfelds in einer Zeichnung
seExecuteReplaceTitleblockJob	Führt den mit seCreateReplaceTitleBlockJob erstellten Job aus
OpenAndReplaceTitleBlock	Öffnet eine Zeichnung direkt und ersetzt das Schriftfeld
seCreatePrintJob	Erzeugt einen Druckjob für einen Papierausdruck
seExecutePrintJob	
OpenAndPrint	Öffnet eine Zeichnung und druckt sie direkt
seCreatePrintPDFJob	Erstellt einen Job zum Erstellen eines PDF
seExecutePrintPDFJob	Führt den mit seCreatePrintPDFJob erstellten Job aus
OpenAndPrintPDF	Öffnet eine Zeichnung und erstellt direkt ein PDF
seExportJobByPath	Erstellt ein Exportformat von einer über Pfadangabe angegebenen Datei
seCreateExportJob	Erstellt einen Exportjob
seExecuteExportJob	Führt den mit seCreateExportJob erstellten Job aus
CloseSEDoc	Schließt alle Dokumente und beendet Solid Edge

Siehe auch

- [CreateJob_StatusPostCalls](#)

[Addin](#), [Workflow](#), [Methode](#), [Konfigurations-XML](#)

²¹⁾

In der **Solid Edge Programmers Guide** unter **DraftPrintPaperSizeConstants** zu finden

From:

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

Permanent link:

https://dokuwiki.unitec.de/doku.php?id=primus:sejob_export

Last update: **2023/02/08 12:01**



seOpen

Mit dem Befehl **seOpen** des [SEInterface Addin](#), können [solid_edge](#) Dokumente geöffnet werden. Der Befehl wird z.B. in [DocumentOpenCall](#) verwendet.

Parameter

Dem Befehl können durch ein Leerzeichen getrennt Parameter mit übergeben werden. Folgende Parameter sind möglich:

Parameter	Funktion
a	Öffnet eine Draft aktiv (Default)
b	Aktualisiert die Stückliste beim Öffnen
i	Öffnet eine Draft inaktiv
o	Verwendet den Solid Edge Öffnen Dialog und bietet somit Zugriff auf die Optionen.
r	Reserviert das Dokument automatisch beim Öffnen. Wird vor allem für die Replikation benötigt.
v	Aktualisiert die Ansichten beim Öffnen

Die Parameter können auch miteinander kombiniert werden, in dem Fall werden sie untereinander nicht getrennt.

Beispiel

```
<DocumentOpenCall PAR="SEInterface.seOpen ro" />
```

Siehe auch

- [DocumentOpenCall](#)

[SEInterface](#), [Addin](#), [Befehl](#), [Replikation](#)

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

Permanent link:
<https://dokuwiki.unitec.de/doku.php?id=primus:seopen>

Last update: **2024/07/23 13:50**



seSave

Mit dem Befehl **seSave** des [SEInterface Addin](#), kann eine zuvor geöffnete [Solid Edge](#) Datei gespeichert und ggf. vorher noch die Ansicht eingepasst werden. Der Aufruf kann mittels [Call](#) im Rahmen eines [Workflow](#)s erfolgen, nachdem die Datei zuvor z.B. mit [seUpdatePropertiesByPath](#) geöffnet wurde.

Liste der Parameter

Der Funktion müssen folgende Parameter, durch | getrennt, in der angegebenen Reihenfolge übergeben werden.

Parameter	Funktion
1 Dokumentpfad	Pfad des Dokuments, z.B. %FILEPATH%. Damit ist sichergestellt das nur dieses Dokument und nicht auch andere geöffnete Dokumente gespeichert werden.
2 FIT	Optionaler Parameter der angibt das vor dem Speicher die Ansicht eingepasst werden soll
ISO	Bringt die 3D Ansicht in die Isometrie, bei Zeichnungen wird nur eingepasst (FIT) ²²⁾

Beispiel

Im gezeigten Beispiel wird in einer zuvor geöffneten Solid Edge Zeichnungsdatei zuerst die Ansicht eingepasst (FIT) und dann gespeichert aber nicht geschlossen.

```
<Call method="SEInterface.seSave" parameter="%FILEPATH%|FIT"
condition="$(Condition.isSEDraft)" />
```

Siehe auch

- [seCloseAndSave](#)
- [seClose](#)

[SEInterface](#), [Addin](#), [Befehl](#), [Workflow](#), [Status](#)

²²⁾

Ist derzeit nur ein Verbesserungswunsch an CADAdapt

From:

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

Permanent link:

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

Last update: **2023/07/13 11:26**



seSetSolidEdgeFileStatus

mit dem Menübefehl **seSetSolidEdgeFileStatus** des **SEInterface Addins**, kann der Status einer oder mehrerer ausgewählter **Solid Edge** Dateien auf einen bestimmten Status gesetzt werden. Die Solid Edge Datei darf dabei nicht geöffnet bzw. im Zugriff sein. Es erfolgt eine Prüfung auf IsFileinUse 20 x alle 0,2 s.

Liste der Parameter

Der Funktion muss folgender Parameter übergeben werden.

	Parameter	Funktion
1	SEStatus	Status von Solid Edge, mögliche Werte sind:
	0	Verfügbar
	1	In Arbeit
	2	Wird geprüft
	3	Freigegeben
	4	Festgeschrieben
	5	Ungültig

Beispiel

```
<seSetSolidEdgeFileStatus addin="SEInterface" visible="'CALL(IsCtrlKey)'"
enabled="TRUE" parameter="0" text="SE-Status 0" icon="" />
```

Siehe auch

- [seSetSolidEdgeFileStatusByPath](#)
- [seUpdateSolidEdgeFileStatus](#)

[SEInterface](#), [Addin](#), [Befehl](#), [Workflow](#), [Status](#)

From:

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

Permanent link:

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

Last update:

2023/06/20 11:24



seSetSolidEdgeFileStatusByPath

Mit dem Befehl **seSetSolidEdgeFileStatusByPath** des [SEInterface Addin](#), kann der Status in einer [Solid Edge](#) Datei geändert werden. Die Solid Edge Datei darf dabei nicht geöffnet bzw. im Zugriff sein. Es erfolgt eine Prüfung auf IsFileinUse 20 x alle 0,2 s. Der Aufruf kann mittels [Call](#) z.B. im Rahmen eines [Workflows](#) oder in [AssignDocumentDataFields_SMARTCOPY](#) um den gewünschten Status zu setzen.

Liste der Parameter

Der Funktion müssen folgende Parameter, durch | getrennt, in der angegebenen Reihenfolge übergeben werden.

	Parameter	Funktion
1	Dokumentpfad	Pfad des Dokuments, z.B. %FILEPATH%
2	SEStatus	Status von Solid Edge, mögliche Werte sind:
	0	Verfügbar
	1	In Arbeit
	2	Wird geprüft
	3	Freigegeben
	4	Festgeschrieben
	5	Ungültig
3	Primus-Status	Zugehöriger Status in Primus

Beispiel

Im gezeigten Beispiel wird im Rahmen der Wertzuweisung an das Datenfeld WorkflowStatus, mittels dieser Funktion auch der Status in der Solid Edge Datei geändert.

```
<WorkflowStatus value="CALL(SEInterface.seSetSolidEdgeFileStatusByPath  
%FILEPATH%|0)|001" />
```

Siehe auch

- [seSetSolidEdgeFileStatus](#)
- [seUpdateSolidEdgeFileStatus](#)

[SEInterface](#), [Addin](#), [Befehl](#), [Workflow](#), [Status](#)

From:

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

Permanent link:

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



Last update: **2023/06/20 11:24**

SetChildrenToStatus

Bei **SetChildrenToStatus** handelt es sich um eine [Methode](#), die mit [Call](#) aufgerufen werden kann. Damit können z.B. in einem [Workflow](#) die untergeordneten Elemente zu einem Dokument, wie Unterbaugruppen und Einzelteile, auf einen bestimmten Status gesetzt werden. Es wird dabei nicht nur der Wert des Datenfelds geändert, sondern die komplette Workflowdefinition für das Childrendokument durchlaufen. Es ist sinnvoll SetChildrenToStatus in den [PreCalls](#) einzutragen, damit die Childrendokumente vor dem Parentdokument verarbeitet werden.

Rückgabewert

Der Rückgabewert der Methode ist:

Wert	Bedeutung
TRUE	Alle verknüpfenden Dokumente haben den angegebenen Status
FALSE	Mindestens eines der verknüpfenden Dokumente hat nicht den gesuchten Status

Aufrufparameter

Die Aufrufparameter werden mit dem [Attribut](#) parameter, durch | getrennt übergeben. Die Parameter sind in folgender Reihenfolge zu übergeben:

	Parameter	Funktion
1	<Dokument>	Dokument zu dem die Children gesucht werden sollen (%FILEPATH%)
2	<Datenfeld>	Names des Datenfelds in das der Status eingetragen werden soll (WorkflowStatus)
3	<Ausgeschlossene Status>	Eine mit . getrennte und abgeschlossene Liste der Status die nicht berücksichtigt werden sollen
4	all	Alle Referenzen
	dmb	Baugruppenfamilien
	ref1	Nur die erste Indexreferenz
	refwithname	Nur die gleichnamige Referenz
5	<BEDINGUNG>	Der optionale Parameter ist eine Condition für Komponenten

Beispiel

Im Beispiel wird innerhalb des Freigabeworkflows (003), die Children die nicht Status 002 oder 003 haben ebenfalls auf den Status 003 gesetzt.

```
<Call method="SetChildrenToStatus"
parameter="%FILEPATH%|WorkflowStatus|.002.003.|all"
condition="$(Condition.isSEDocument)" />
<-- Weitere Beispiele von CADadapt -->
<Call method="SetChildrenToStatus"
```

```
parameter="%FILEPATH%|WorkflowStatus|.005.|all|'%FILEPATH:cn|peripherie%=='A
ND'%FILEPATH:cn|\08 cad\produkt\%==' "
condition="'%PATHID%==HBib'AND'$(Condition.isSEDocument)'OR'%PATHID%==HBib'A
ND'$(Condition.isSWDocument)'" />
<Call method="SetChildrenToStatus"
parameter="%FILEPATH%|WorkflowStatus|.003.004.005.006.|all|'%CHILD_FILEPATH:
cn|peripherie%==' "
condition="'%PATHID%!=HBib'AND'$(Condition.isSEDocument)'OR'%PATHID%!=HBib'A
ND'$(Condition.isSWDocument)'" />
<Call method="SetChildrenToStatus"
parameter="%FILEPATH%|WorkflowStatus|.001.004.005.|dmb"
condition="$(Condition.isSWDocument)" />
```

Zusammenspiel mit SetParentToStatus

[SetParentToStatus](#) bezieht sich nur auf Zeichnungen (siehe Parameter .dft). Dadurch geben die Teile nur eigene Zeichnungen frei. Sollte in einer Zeichnung ein weiteres Teil abgeleitet sein, dann wird der durch SetChildrenToStatus freigegeben. Das führt aber nicht zu einer Rekursion. Falls in der Rekursion die gleiche Zeichnung wieder auftaucht, wird diese ignoriert und nicht ein zweites mal verarbeitet.

[CADAdaptApproved](#)

Siehe auch

- [CheckChildrenStatus](#)
- [CheckChildrenWriteable](#)
- [IgnoreStatus](#)
- [CheckParentsStatus](#)
- [SetParentToStatus](#)
- [CheckParentsWriteable](#)
- [SetSubFilesToStatus](#)
- [SetPrevVersionToStatus](#)
- [SetSubFilesToStatus](#)

[Workflow](#), [Methode](#)

From:

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

Permanent link:

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

Last update: **2023/07/19 11:02**



SetParentToStatus

Bei **SetParentToStatus** handelt es sich um eine [Methode](#), die mit [Call](#) aufgerufen werden kann. Damit können z.B. in einem [Workflow](#) die übergeordneten Elemente zu einem Dokument, wie Zeichnungen zum 3D-Modell, auf einen bestimmten Status gesetzt werden. Es wird dabei nicht nur der Wert des Datenfelds geändert, sondern die komplette Workflowdefinition für das Parentdokument durchlaufen. Wenn SetParentToStatus verwendet werden soll, **muss** es in den [PostCalls](#) eingetragen werden, damit die Parentdokumente **nach** dem Childrendokument verarbeitet werden. Ein Aufruf in den [PreCalls](#) führt zu einer Endlosschleife durch die Primus abgebrochen wird.

Aufrufparameter

Die Aufrufparameter werden mit dem [Attribut](#) parameter, durch | getrennt übergeben. Die Parameter sind in folgender Reihenfolge zu übergeben:

	Parameter	Funktion
1	Dokument	Dokument zu dem die Parents gesucht werden sollen (%FILEPATH%)
2	Datenfeld	Names des Datenfelds in das der Status eingetragen werden soll (WorkflowStatus)
3	Ausgeschlossene Status	Eine mit . getrennte und abgeschlossene Liste der Status die nicht berücksichtigt werden sollen
4	Dateityp des Parentdokuments	Dateityp (Liste?) des zu berücksichtigenden Parentelements Klärungsbedarf

Beispiel

Im Beispiel wird innerhalb des Freigabeworkflows (003), die zum 3D-Modell zugehörige Zeichnungsdatei (.dft) mit freigegeben, wenn diese nicht schon einen der Status (003, 004, 005) hat. Mit dem Attribut condition wird sichergestellt das diese Aktion nur bei 3D-Modellen ausgeführt wird.

```
<Call method="SetParentToStatus"
parameter="%FILEPATH%|WorkflowStatus|.003.004.005.|.dft"
condition="'$(Condition.isSEModel)'" />
```

Zusammenspiel mit SetChildrenToStatus

SetParentToStatus bezieht sich nur auf Zeichnungen (siehe Parameter .dft). Dadurch geben die Teile nur eigene Zeichnungen frei. Sollte in einer Zeichnung ein weiteres Teil abgeleitet sein, dann wird der durch [SetChildrenToStatus](#) freigegeben. Das führt aber nicht zu einer Rekursion. Falls in der Rekursion die gleiche Zeichnung wieder auftaucht, wird diese ignoriert und nicht ein zweites mal verarbeitet.

[CADAdaptApproved](#)

Siehe auch

- [CheckChildrenStatus](#)
- [CheckChildrenWriteable](#)
- [CheckParentsStatus](#)
- [CheckParentsWriteable](#)
- [IgnoreStatus](#)
- [SetChildrenToStatus](#)
- [SetPrevVersionToStatus](#)
- [SetSubFilesToStatus](#)

Workflow

From:

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

Permanent link:

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

Last update: **2023/07/20 11:55**



SetPrevVersionToStatus

Bei **SetPrevVersionToStatus** handelt es sich um eine **Methode**, die mit **call** aufgerufen werden kann. Damit kann z.B. in einem **Workflow** die vorhergehende Version zu einem Dokument, auf einen bestimmten Status gesetzt werden. Dabei wird nur der Wert des Datenfelds geändert, aber keine Workflowdefinition durchlaufen.

Aufrufparameter

Die Aufrufparameter werden mit dem **Attribut** parameter, durch | getrennt übergeben. Die Parameter sind in folgender Reihenfolge zu übergeben:

	Parameter	Funktion
1	Dokument	Dokument zu dem die vorhergehende Version gesucht werden soll (%FILEPATH%)
2	Datenfeld	Names des Datenfelds in das der Status eingetragen werden soll (WorkflowStatus)
3	Status	Status der zugewiesen werden soll

Beispiel

Im Beispiel wird innerhalb des Freigabeworkflows (003), die vorhergehende Version auf Status 005 gesetzt.

```
<Call method="SetPrevVersionToStatus"
parameter="%FILEPATH%|WorkflowStatus|005" condition="TRUE" />
```

Siehe auch

- [CheckChildrenStatus](#)
- [SetChildrenToStatus](#)
- [SetParentToStatus](#)
- [SetSubFilesToStatus](#)

[Workflow](#)

From:

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

Permanent link:

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

Last update: **2023/02/08 12:01**



SetSubFilesToStatus

Bei **SetSubFilesToStatus** handelt es sich um eine [Methode](#), die mit [Call](#) aufgerufen werden kann. Damit können z.B. in einem [Workflow](#) die abgeleiteten Elemente zu einem Dokument, wie z.B. [PDF](#) oder [DXF](#) von Zeichnungen, auf einen bestimmten Status gesetzt werden. Es wird dabei nicht nur der Wert des Datenfelds geändert, sondern die komplette Workflowdefinition für das Subdokument durchlaufen.

Aufrufparameter

Die Aufrufparameter werden mit dem [Attribut](#) parameter, durch | getrennt übergeben. Die Parameter sind in folgender Reihenfolge zu übergeben:

	Parameter	Funktion
1	Dokument	Dokument zu dem die abgeleiteten Dokumente gesucht werden sollen (%FILEPATH%)
2	Datenfeld	Names des Datenfelds in das der Status eingetragen werden soll (WorkflowStatus)
3	Ausgeschlossene Status	Eine mit . getrennte und abgeschlossene Liste der Status die nicht berücksichtigt werden sollen Klaerungsbedarf
4	Dateityp(en) des abgeleiteten Dokuments	Dateitypen der zu berücksichtigenden abgeleiteten Dokumente. Die Liste beginnt mit einem Punkt (.), die Listeneinträge sind durch Punkte getrennt.

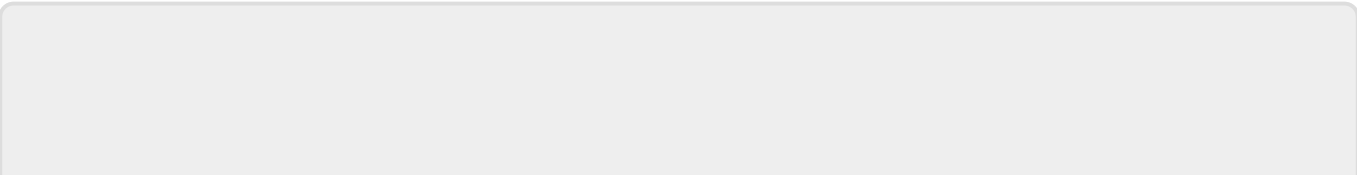
Beispiel

```
<Call method="SetSubFilesToStatus"
parameter="%FILEPATH%|WorkflowStatus|.003.004.005.|.pdf.txt"
condition="$(Condition.isSEDocument)" />
```

Siehe auch

- [SetParentToStatus](#)
- [SetPrevVersionToStatus](#)
- [SetChildrenToStatus](#)
- [CheckChildrenStatus](#)

[Method](#), [Workflow](#), [Status](#), [CADAdaptApproved](#)



From:

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

Permanent link:

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

Last update: **2023/06/01 15:00**



SetXMLValue

Bei **SetXMLValue** handelt es sich um eine [Methode](#), die mit [call](#) aufgerufen werden kann. Damit können z.B. in einem [Workflow](#), die Inhalte von Datenfeldern auf einen bestimmten Wert gesetzt werden.

Rückgabewert

Der Rückgabewert der Methode ist:

Wert	Bedeutung
TRUE	Die Wertzuweisung war erfolgreich
FALSE	Die Wertzuweisung war nicht erfolgreich

Aufrufparameter

Die Aufrufparameter werden mit dem [Attribut](#) parameter, durch | getrennt übergeben. Die Parameter sind in folgender Reihenfolge zu übergeben:

	Parameter	Funktion
1	<DOKUMENT>	Dokument zu dem z.B. ein Feldwert geprüft werden soll (%FILEPATH%)
2	<DATENFELD>	Datenfeld dem ein Wert zugewiesen werden soll
3	<WERT>	Wert der dem Datenfeld zugewiesen werden soll

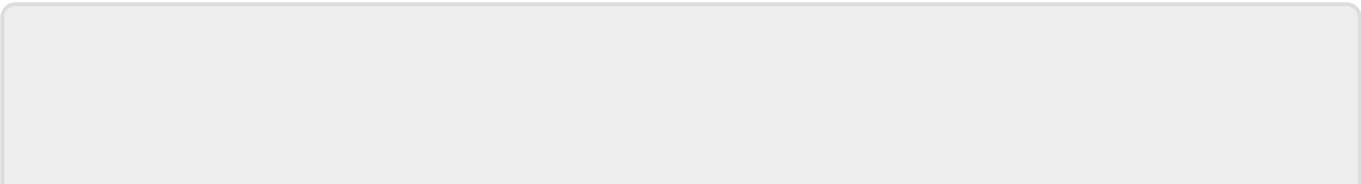
Beispiel

Im Beispiel wird dem Feld Workflowstatus der Wert 003 zugewiesen, wenn es sich um ein SE Dokument handelt.

```
<Call method="SetXMLValue" parameter="%FILEPATH%|WorkflowStatus|003"
condition="$(Condition.isSEDocument)" />
```

Siehe auch

- [prmElementsChangeValueByDlg](#)
- [Liste der Methoden des Primus Framework](#)



From:

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

Permanent link:

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

Last update: **2023/07/25 12:14**



seUpdate3DDocument

Mit dem Befehl **seUpdate3DDocument** des [SEInterface Addin](#), kann rekursiv ein komplette Baugruppenstruktur aktualisiert werden. Dadurch können z.B. die bekannten „*Dicken Ecken*“ in [solid_edge](#) Zeichnungen beseitigt werden. Die Funktion kann ausgehend von der Zeichnung oder Baugruppe bzw. Bauteil ausgeführt werden. Die Funktion geht immer in der Rekursion ganz nach unten und aktualisiert Dokumente von unten nach oben. Ausgehend von der Draft-Datei, werden auch die Zeichnungsansichten aktualisiert. Es ist auch möglich dies mit bereits freigegebenen Dokumenten durchzuführen, es wird aber davon abgeraten bzw. sollte diese Funktion nur für Administratoren zur Verfügung gestellt werden.

Parameter

Der Funktion können folgende Parameter, durch | getrennt, in der angegebenen Reihenfolge übergeben werden.

Parameter	Wert	Funktion
1	<LEER>	Alle Dokumente ohne Einschränkung
	<BEDINGUNG>	Nur Dokumente die diese Bedingung erfüllen
2	<STRING>	Die optionale Fehlermeldung, falls Parameter 1 FALSE ist, ansonsten kommt die Standardfehlermeldung

Beispiel

```
<--Benutzerfunktion: Nur nicht freigegebene Dokumente->
<seUpdate3DDocument addin="SEInterface"
visible="'$(Condition.isSEDocument)'" enabled="TRUE"
parameter="'#(WorkflowStatus)==001'|NICHT IN ARBEIT" text="Komponenten und
Ansichten aktualisieren" icon="" />
<--Adminfunktion: Alle Dokumente auch bereits freigegebene-->
<seUpdate3DDocument addin="SEInterface"
visible="'$(Condition.isSEDocument)'AND'CALL(IsCtrlKey)'" enabled="TRUE"
parameter="" text="Komponenten und Ansichten aktualisieren (inkl.
Freigegebene)" icon="" />
```

Siehe auch

- [seupdateproperties](#)
- [SEInterface](#), [Addin](#), [Befehl](#)

From:

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

Permanent link:

<https://dokuwiki.unitec.de/doku.php?id=primus:seupdate3ddocument>

Last update: **2023/02/08 12:01**



seUpdatePropertiesByPath

Mit dem Befehl **seUpdatePropertiesByPath** des [SEInterface Addin](#), kann eine [solid_edge](#) Datei mittels Angabe eines Pfad und Dateinamens geöffnet und aktualisiert werden. Der Aufruf kann mittels [Call](#) z.B. im Rahmen eines [Workflows](#) erfolgen.

Liste der Parameter

Der Funktion müssen folgende Parameter, durch | getrennt, in der angegebenen Reihenfolge übergeben werden.

Parameter	Funktion
1 Dokumentpfad	Pfad des Dokuments, z.B. %FILEPATH%
2 Parameter 1	Sollen mehrere Versuche beim Öffnen gemacht werden
	0 = 1 Versuch
	1 = 10 Versuche Klärungsbedarf
3 Parameter 2	Optionaler Parameter, wenn mit 0 angegeben, wird die Datei nicht geschlossen. Dies muß dann mittels seCloseAndSave erfolgen

Beispiel

Im gezeigten Beispiel wird im Rahmen der [PostCalls](#) eines Workflows eine Solid Edge Zeichnungsdatei (Draft) geöffnet aktualisiert und nicht geschlossen. Dadurch ist es möglich mit weiteren PostCalls z.B. [seExportJobByPath](#) Exportformate auszugeben ohne die Datei erneut öffnen zu müssen.

```
<Call method="SEInterface.seUpdatePropertiesByPath"
parameter="%FILEPATH%|1|0" condition="$(Condition.isSEDraft)" />
```

[SEInterface](#), [Addin](#), [Befehl](#), [Workflow](#), [Status](#)

From:

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

Permanent link:

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

Last update:

2025/08/26 13:43



seUpdateSolidEdgeFileStatus

Mit dem Menübefehl **seUpdateSolidEdgeFileStatus** des [SEInterface Addin](#), kann der Status in einer [Solid Edge](#) Datei geändert werden.

Parameter

Der Befehl benötigt folgende Parameter, die durch ein | Symbol getrennt werden.

Parameter	Funktion
1	Status Mapping Tabelle
2	Primus Datenfeld

Beispiel

```
<seUpdateSolidEdgeFileStatus addin="SEInterface" visible="'CALL(IsCtrlKey)'"
enabled="TRUE" parameter="Status2SEStatus|WorkflowStatus" text="SE-Status
update" icon="" />
```

Status Mapping Tabelle

Für die Zuordnung der Primus Workflow Status zu den Status der Solid Edge Datei wird eine **Status Mapping Tabelle** benötigt. Diese ist im [Abschnitt StatusWorkflow_Settings](#) definiert.

Beispiel

```
<Status2SEStatus>
  <Status Primus="001" SE="0" />
  <Status Primus="002" SE="2" />
  <Status Primus="003" SE="3" />
  <Status Primus="004" SE="4" />
  <Status Primus="005" SE="5" />
</Status2SEStatus>
```

Siehe auch

- [StatusWorkflow_Settings](#)

From:

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

Permanent link:

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

Last update: **2023/06/20 12:14**



seUpdateWeight

Der Menübefehl **seUpdateWeight** des **SEInterface Addins** aktualisiert das Gewicht der selektierten Dokumente. Falls das Gewicht einer **Solid Edge** Datei nicht bei jedem Speichern aktualisiert werden soll, kann dies manuell aus dem Menü heraus aufgerufen werden. Der Befehl kann aber auch aus einem **Workflow** mittels **Call** aufgerufen werden, benötigt dann aber einen Parameter für das zu verarbeitende Dokument.

Attribute

Der Befehl benötigt dazu die Definition folgender Attribute im **Abschnitt Public**.

Attribut	Wert	Funktion
writexmlcond	0/1	Definiert ob das Aktualisieren des Gewichts automatisch ²³⁾ beim Speichern erfolgen soll
UpdateWeightField	<DATENFELD>	Definiert das Datenfeld in dem das Gewicht gespeichert werden soll
UpdateWeightFormat	kg (#,###0.000)	Definiert die Zahlendarstellung des Gewichts

Beispiel

Aufruf als Menübefehl:

```
<seUpdateWeight addin="SEInterface" visible="'$(Condition.isSEDocument)'"
enabled="TRUE" parameter="" text="Gewicht aktualisieren" icon="" />
```

Aufruf während eines **Workflows** mittels **Call**

```
<Call method="SEInterface.seUpdateWeight" parameter="%FILEPATH%"
condition="$(Condition.isSEDocument)" />
```

Siehe auch

- [\[PropertiesToPrimus\]](#)

²³⁾

SEAddIn Save-Event

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

Permanent link:
<https://dokuwiki.unitec.de/doku.php?id=primus:seupdateweight>

Last update: **2023/07/18 11:27**



SmartCopy-Ordner per Default ausnehmen

Im [Abschnitt Public](#) wird mit dem Tag [SmartCopy](#) definiert welche Dokumente in die Aktion einbezogen werden sollen.

Informationen von Salmin Herac:

- Variante mit „Starts With“ (sw) „Contains“ (cn) nicht *Inventornormteile*:
- sw Beispiel: wenn der FILEPATH mit %SOURCETOPDIRPATH% startet dann gibt es %SOURCETOPDIRPATH% zurück sonst leer
- cn Beispiel: wenn der FILEPATH Inventornormteile nicht enthält dann wird leer zurückgegeben, daher Vergleich mit leer =='

```
<SmartCopy  
componentfilter=" '%FILEPATH:sw|%SOURCETOPDIRPATH%==%SOURCETOPDIRPATH%' AND '%  
FILEPATH:cn|Inventornormteile%==' ">
```

- Variante mit [enthält nicht](#) (!~)\Standard\ (~ Ist nur eine kürzere Form für cn)

```
<SmartCopy componentfilter=" '%FILEPATH%!~\Standard\ ' ">
```

- Variante mit Kundenfunktion für komplexere Überprüfungen:

```
<SmartCopy componentfilter=" 'CALL(AATAddIn.SmartCopyComponentFilter  
%FILEPATH%|%TARGETDIRPATH%) ' ">
```

- Variante mit [ContainsListValue](#) und einer [Liste](#), der ausgenommenen Verzeichnisse.

```
<SmartCopy componentfilter=" 'CALL(ContainsListValue  
%FILEPATH%|List_ExcludeFromSmartCopy) '!=TRUE" />
```

[SmartCopy](#), [Beispiel](#), [Überarbeiten](#), [Bedingung](#)

From:

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

Permanent link:

https://dokuwiki.unitec.de/doku.php?id=primus:smartcopy-ordner_per_default_ausnehmen

Last update: **2023/02/08 12:01**



SmartCopy-Status kopierter Dateien ändern

Beim Kopieren von Dokumenten mit [SmartCopy](#) soll den kopierten Dokumenten ein bestimmter [Status](#) zugewiesen werden.

Die Zuweisung von Informationen beim Erstellen von Kopien mit SmartCopy erfolgt im [Abschnitt AssignDocumentDataFields_SMARTCOPY](#). Die Zuweisung eines Status erfolgt mit dem [SEInterface](#)-Befehl `seSetSolidEdgeFileStatusByPath`.

Beispiel

```
<AssignDocumentDataFields_SMARTCOPY load="XML_Project">
  <WorkflowStatus value="CALL(SEInterface.seSetSolidEdgeFileStatusByPath
%FILEPATH%|0)|001" />
</AssignDocumentDataFields_SMARTCOPY>
```

Siehe auch

- [SEInterface](#)
- [Addins](#)

[SmartCopy](#), [Beispiel](#), [Überarbeiten](#)

From:

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

Permanent link:

https://dokuwiki.unitec.de/doku.php?id=primus:smartcopy-status_kopierter_dateien_aendern

Last update: **2023/02/08 12:01**



SmartCopy-Zeichnungsname gleich Modellname

Der Zeichnungsname bei [SmartCopy](#) soll gleich dem Name des Models sein. Im [Abschnitt Public](#) wird mit dem Tag [SmartCopyFileName](#) der Dateiname für die mit [smartcopy](#) zu bearbeiteten Dateien definiert. Standardmäßig gibt es einen Eintrag der im [Attribut condition](#) auf alle CAD-Dateitypen prüft, im Fall von [solid_edge](#) sind das .ASM, .DFT, .PAR, .PSM, .PWD, und diesen einen Dateinamen nach dem gleichen Schema zuweist.

```
<SmartCopyFileName
condition="'%EXTENSION%==.par'OR'%EXTENSION%==.psm'OR'%EXTENSION%==.pwd'OR'%
EXTENSION%==.asm'OR'%EXTENSION%==.dft'"
reuse_old_counter="%FILENAME:between|two|_%" counter_preview="XXXX"
prefix="%XML_Project:FileNameFormat%" suffix="%FILENAME:between|two|_#%"
filename_format="%NewNamePrefix%_%COUNTER%_%NewNameSuffix%#%REVISION%"
counter="0001|\\primus\setup\Data\filenamecounter.cnt" />
```

Lösung

Den Dateityp für Zeichnungen (.DFT) aus dem einen Tag entfernen und ein eigenes Tag nur für Zeichnungen anlegen. Dem Attribut [filename_format](#) wird dann als Wert %ref1% zugewiesen.

```
<SmartCopyFileName condition="'%EXTENSION%==.dft'" counter_preview="XXXX"
filename_format="%ref1%" />
```

[SmartCopy](#), Beispiel

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

Permanent link:
https://dokuwiki.unitec.de/doku.php?id=primus:smartcopy-zeichnungsname_gleich_modellname

Last update: **2023/02/08 12:01**



SmartCopy (Tag)

Mit dem Tag **SmartCopy** im [Abschnitt Public](#) wird definiert, welche Dokumente bei **SmartCopy**-Operationen in die Auswahl aufgenommen werden sollen. Das Tag wird nur benötigt wenn der Kunde eine SmartCopy Lizenz hat, es können mehrere²⁴⁾ **SmartCopy**-Tags in *Public* definiert werden.

Liste der Attribute

Folgende Attribute können in dem Tag verwendet werden:

Name	Wert	Funktion
componentfilter	TRUE/FALSE	Definiert eine Bedingung für welche Dateitypen diese Definition verwendet werden soll. Es können Stringvergleichsoperatoren verwendet werden.
NameFormat	<BEZEICHNUNG>	Definiert den Defaultwert für NameFormat Klärungsbedarf
outercomponents_target	RELATIVE/SOURCE/TARGET	Position des Verzeichnisses relativ zum Ausgangsordner Klärungsbedarf

Beispiel

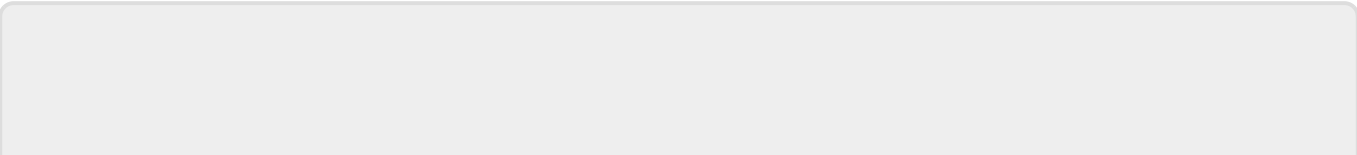
```
<SmartCopy
componentfilter="'%FILEPATH:sw|%SOURCETOPDIRPATH%=='%SOURCETOPDIRPATH%' "
outercomponents_target="RELATIVE" />
```

NameFormat

Mit **NameFormat** kann beim Aufruf einer SmartCopy Session gesteuert werden welches optionale Dateinamensformat für diese Session verwendet werden soll. Dazu wird beim Start der Session mit dem 4. Parameter `SetPublicOptionTemporary(SmartCopy.NameFormat=MYNAME)` ein Wert für NameFormat festgelegt, der im Tag `<SmartCopyFileName>` unter [Public](#) mittels einer [Bedingung](#) `'$(SmartCopy.NameFormat)==MYNAME'` ausgewertet werden kann.

[SmartCopy](#), [Klärungsbedarf](#)
²⁴⁾

Können mehrere **SmartCopy**-Tags in *Public* definiert werden?
[Klärungsbedarf](#)



From:

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

Permanent link:

https://dokuwiki.unitec.de/doku.php?id=primus:smartcopy_tag

Last update: **2023/06/20 14:42**



SmartCopyFileName

Mit dem Tag **SmartCopyFileName** im Abschnitt **Public** wird definiert, wie ein Dateiname bei **smartcopy**-Operationen aufgebaut sein soll. Das Tag wird nur benötigt wenn der Kunde eine SmartCopy Lizenz hat, es können mehrere **SmartCopyFileName**-Tags in *Public* definiert werden. Dabei muß über die Auswahl der **Bedingungen** sichergestellt sein, dass die Definition für jeden Dateityp eindeutig ist.

Liste der Attribute

Name	Wert	Funktion
condition	TRUE/FALSE	Definiert für welche Dateitypen diese Definition verwendet werden soll
counter	000001 ..\Setup\Data\filenamecounter.cnt	Name der Counterdatei oder Aufruf eines externen Programms zur Nummernerzeugung
counter_preview	XXXX	Wert der im Counterfeld angezeigt wird um vom Anwender festgelegt oder durch eine Nummer aus dem nummerngenerator ersetzt zu werden
filename_format	%NewNamePrefix%%COUNTER%_%NewNameSuffix%#%REVISION%	Aufbau des Dateinamens
prefix	%XML_Project:FileNameFormat%	Zeichenfolge die im Dateinamen voran gestellt wird
reuse_old_counter	%FILENAME:befor first -%	Klärungsbedarf
suffix	%FILENAME:between two _#%	Zeichenfolge die an den Dateinamen aber vor der dateiendung angehängt wird

Beispiel

Im Beispiel definiert der erste Eintrag den Dateinamen für alle **solid_edge** 3D-Modelle, während der zweite Eintrag nur für die Zeichnungen (.dft) gilt und diesen den Namen des zur Zeichnung zugehörigen Modells (%ref1%) zuweist. Jeder Dateityp wird in diesem Beispiel nur einmal berücksichtigt und ist damit eindeutig definiert.

```
<SmartCopyFileName
condition="'%EXTENSION%==.par'OR'%EXTENSION%==.psm'OR'%EXTENSION%==.pwd'OR'%EXTENSION%==.asm'"
counter_preview="XXXX"
```



```
prefix="%XML_Project:FileNameFormat%" suffix="%FILENAME:between|two|_#%"  
filename_format="%NewNamePrefix%%COUNTER%_NewNameSuffix%%REVISION%"  
counter="000001|\\primus\Setup\Data\filenamecounter.cnt" />  
<SmartCopyFileName condition="'%EXTENSION%==.dft'" counter_preview="XXXX"  
filename_format="%ref1%" />
```

Public, SmartCopy

From:

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

Permanent link:

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

Last update: **2023/02/08 12:01**



smartCopyMarkTarget

Der Menübefehl **smartCopyMarkTarget** gehört zum [PrimusSmartCopy Addin](#). Mit dem Befehl kann der [Zielordner](#) für die aktive SmartCopy Session festgelegt werden. Die Festlegung kann mit [smartCopyRemoveMarkTarget](#) rückgängig gemacht werden.

Aufrufparameter

Die Aufrufparameter werden mit dem [Attribut](#) parameter, durch | getrennt übergeben. Die Parameter sind in folgender Reihenfolge zu übergeben:

	Parameter	Funktion
1	0	Mit „Einfügeordner vormerken“ wird ein Einfügeordner vorgemerkt, der anschließend an eine einzelne Datei in einer Session zugewiesen werden kann
	1	Mit „Zielordner vormerken“ wird der Zielordner für die ganze Session definiert

Besonderheit bei Rename

Dieser Befehl darf bei SmartCopy Rename nicht angewendet werden, da sonst die Dokumente während des Umbenennens verschoben werden. Dazu wird im Attribut `visible` eine [Bedingung](#) definiert (siehe [Beispiel](#)), die verhindert das der Menüpunkt bei aktiver Rename-Session angezeigt wird.

Beispiel

```
<smartCopyMarkTarget addin="PrimusSmartCopy"
visible="'CALL(PrimusSmartCopy.IsSmartCopySessionActive)'OR'CALL(PrimusSmartCopy.IsSmartMoveSessionActive)'" enabled="TRUE" parameter="1"
text="$Primus.MNU_PROJECT_SMARTCOPY_SET_TARGETDIR"
icon="Resources\Icons\sctest"/>
```

[SmartCopy](#), [Menübefehl](#), [GUI](#), [Addin](#)

From:

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

Permanent link:

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

Last update:

2024/03/22 09:56



smartCopyNewSession

Der [Menübefehl smartCopyNewSession](#) gehört zum [PrimusSmartCopy Addin](#).

Aufrufparameter

Die Aufrufparameter werden mit dem [Attribut](#) parameter, durch | getrennt übergeben. Die Parameter sind in folgender Reihenfolge zu übergeben:

Parameter	Funktion
COPY	Mit <i>COPY</i> wird eine Kopiersession gestartet
1 MOVE	Mit <i>MOVE</i> wird eine Verschiebesession gestartet
RENAME	Mit <i>RENAME</i> wird eine Umbenennensession gestartet
2 Startordner ²⁵⁾	Mit diesem optionalen Parameter wird der Startordner für die Kopier- oder Verschiebesession festgelegt. Ist der Parameter nicht angegeben, verwendet Primus den Ordner den der Anwender beim Befehlsaufruf im Ordnerfenster angewählt hat.
3 Zielordner ²⁶⁾	Mit diesem optionalen Parameter wird der Zielordner für die Kopier- oder Verschiebesession festgelegt. Ist der Parameter nicht angegeben, dann muß der Anwender mit smartCopyMarkTarget ein Verzeichnis festlegen oder die Session durch auswählen im Ordnerfenster mit dem Befehl smartPaste abschließen.

Beispiel

```
<smartCopyNewSession addin="PrimusSmartCopy"
visible="CALL(PrimusSmartCopy.IsSessionActive)!=TRUE" enabled="TRUE"
parameter="COPY|\\primus\Daten|\\primus\Daten" text="Session Kopieren
starten (Projekte)" icon="Resources\Icons\sccopy" />
```

SmartCopy, Menübefehl, GUI, Addin

²⁵⁾ , ²⁶⁾

nicht bei RENAME verwendbar

From:

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

Permanent link:

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

Last update: **2023/02/08 12:01**



smartCopyRenameDir

Der [Menübefehl smartCopyRenameDir](#) gehört zum [PrimusSmartCopy Addin](#). Mit dem Befehl werden Verzeichnisse umbenannt und die entsprechenden Verknüpfungen in den Dokumenten aktualisiert.

Aufrufparameter

Die Aufrufparameter werden mit dem [Attribut](#) parameter, durch | getrennt übergeben. Die Parameter sind in folgender Reihenfolge zu übergeben:

Parameter	Funktion
1 <LEER>	
2 SEInterface.seReplaceDocRef;smartUpdateRefsForLastSession(.par.psm.asm.dft.)	Gibt die Funktion an die zum Aktualisieren der Dokumentverknüpfungen verwendet werden soll, sowie die zugehörigen Dateitypen

Sichtbarkeit im Menü

Da es sich um einen Administrativen Befehl handelt, sollte dieser nur sichtbar sein, wenn die STRG-Taste gedrückt wird, außerdem darf keine SmartCopy Session aktiv sein. Dafür stellt das PrimusSmartCopy Addin folgende interne [Bedingung](#) zur Verfügung.

Bedingung	Bedeutung
IsSessionActive	Prüft ob eine SmartCopy Sitzung aktiv ist

Diese Bedingungen wird über CALL(PrimusSmartCopy.<BEDINGUNG>) angesprochen.

Beispiel

```
<smartCopyRenameDir addin="PrimusSmartCopy"
visible="'CALL(PrimusSmartCopy.IsSessionActive)!=TRUE'AND'CALL(IsCtrlKey)'"
enabled="TRUE"
parameter="|SEInterface.seReplaceDocRef;smartUpdateRefsForLastSession( .par.psm.asm.dft. )" text="Ordner Umbenennen" icon="Resources\Icons\rename" />
```

Siehe auch

- [smartCopySetRec](#)
- [smartCopyIgnore](#)
- [smartCopyIgnoreRec](#)
- [smartCopyBreakSession](#)

- [smartRestoreMoveSession](#)

[Menübefehl](#), [GUI](#), [Addin](#), [SmartCopy](#)

From:

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

Permanent link:

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

Last update: **2023/04/14 12:39**



smartCopySet

Der **Menübefehl smartCopySet** gehört zum **PrimusSmartCopy Addin**. Mit dem Befehl werden Dokumente für die Verarbeitung mit der aktiven SmartCopy Session markiert.

Aufrufparameter

Die Aufrufparameter werden mit dem **Attribut** parameter, durch | getrennt übergeben. Die Parameter sind in folgender Reihenfolge zu übergeben:

	Parameter	Funktion
1	<LEER>	Wenn nichts angegeben wird, wird das Dokument zum Kopieren markiert
1	MOVE	Markiert das Dokument zum verschieben
1	RENAME	Markiert das Dokument zum Verschieben
2	INCLREADONLY	Nur bei MOVE und RENAME, berücksichtigt auch schreibgeschützte Dokumente
3	%PARAMOPTION%	Legt eine benutzerdefinierteOption fest, die mittels Bedingung bei anderen SmartCopy Einstellungen ausgewertet werden kann. Siehe Beispiel

Sichtbarkeit im Menü

Damit der jeweils zur aktiven SmartCopy Sitzung passende **smartCopySet** Menüeintrag angezeigt sind, müssen diese mit dem Attribut **visible** und der entsprechenden **Bedingung** sichtbar gemacht werden. Dafür stellt das PrimusSmartCopy Addin folgende interne **Bedingungen** zur Verfügung.

Bedingung	Bedeutung
IsSmartCopySessionActive	Prüft ob eine Kopiersitzung aktiv ist
IsSmartMoveSessionActive	Prüft ob eine Verschiebesitzung aktiv ist
IsSmartRenameSessionActive	Prüft ob eine Umbenennensitzung aktiv ist

Alle Bedingungen werden über `CALL (PrimusSmartCopy.<BEDINGUNG>)` angesprochen.

Beispiel

```
<smartCopySet addin="PrimusSmartCopy"
visible="CALL(PrimusSmartCopy.IsSmartMoveSessionActive)" enabled="TRUE"
parameter="MOVE|INCLREADONLY" text="$Primus.MNU_STATUS_MOVE"
icon="Resources\Icons\move" />
```

Beispiel unter Verwendung von %PARAMOPTION%

```
<smartCopySet addin="PrimusSmartCopy"
visible="CALL(PrimusSmartCopy.IsSmartCopySessionActive)" enabled="TRUE"
parameter="|INCLREADONLY" text="$Primus.MNU_ELEM_SMARTCOPY_STAT_COPY"
```

```
icon="Resources\Icons\copy" />
<smartCopySet addin="PrimusSmartCopy"
visible="CALL(PrimusSmartCopy.IsSmartCopySessionActive)" enabled="TRUE"
parameter="|INCLREADONLY|NewNr" text="$Primus.MNU_ELEM_SMARTCOPY_STAT_COPY
(immer mit Nummer)" icon="Resources\Icons\copy" />

<SmartCopyFileName condition="('%EXTENSION%==.par' OR '%EXTENSION%==.psm' OR
'%EXTENSION%==.pwd' OR '%EXTENSION%==.asm')AND'%PARAMOPTION%==NewNr'"
prefix="%FILENAME:bevor|last|-%" counter_preview="XXX"
filename_format="%NewNamePrefix%-%COUNTER%_%REVISION%"
counter="000|%TARGETDIRPATH%\_%NewNamePrefix%.cnt" />
<SmartCopyFileName condition="('%EXTENSION%==.par' OR '%EXTENSION%==.psm' OR
'%EXTENSION%==.pwd' OR '%EXTENSION%==.asm')AND'%PARAMOPTION%=='"
prefix="%FILENAME%" counter_preview="" filename_format="%NewNamePrefix%" />
```

Siehe auch

- [smartCopySetRec](#)
- [smartcopyignore](#)
- [smartcopyignorerec](#)

[Menübefehl](#), [GUI](#), [Addin](#), [SmartCopy](#)

From:

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

Permanent link:

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

Last update: **2023/06/20 15:05**



smartCopySetRec

Der **Menübefehl smartCopySetRec** gehört zum **PrimusSmartCopy Addin**. Mit dem Befehl werden Dokumente rekursiv für die Verarbeitung mit der aktiven SmartCopy Session markiert.

Aufrufparameter

Die Aufrufparameter werden mit dem **Attribut** parameter, durch | getrennt übergeben. Die Parameter sind in folgender Reihenfolge zu übergeben:

	Parameter	Funktion
	<LEER>	Wenn nichts angegeben wird, wird das Dokument zum Kopieren markiert
1	MOVE	Markiert das Dokument zum verschieben
	RENAME	Markiert das Dokument zum Verschieben
2	<DATEITYP>	Gibt an welche Parent Referenzen berücksichtigt werden sollen
3	INCLREADONLY	Nur bei MOVE und RENAME, berücksichtigt auch schreibgeschützte Dokumente
	ONLYPARENTS	Sucht nur Parentdokumente aber keine Komponenten
	REV1	Nimmt nur die Indexreferenz 1
	WITHNAME	Nur gleichnamige Referenzen werden berücksichtigt

Sichtbarkeit im Menü

Damit der jeweils zur aktiven SmartCopy Sitzung passende **smartCopySet** Menüeintrag angezeigt sind, müssen diese mit dem Attribut **visible** und der entsprechenden **Bedingung** sichtbar gemacht werden. Dafür stellt das PrimusSmartCopy Addin folgende interne **Bedingungen** zur Verfügung.

Bedingung	Bedeutung
IsSmartCopySessionActive	Prüft ob eine Kopiersitzung aktiv ist
IsSmartMoveSessionActive	Prüft ob eine Verschiebesitzung aktiv ist
IsSmartRenameSessionActive	Prüft ob eine Umbenennensitzung aktiv ist

Alle Bedingungen werden über CALL (PrimusSmartCopy.<BEDINGUNG>) angesprochen.

Beispiel

```
<!-- Nur gleichnamige Zeichnung, keine Komponenten -->
<smartCopySetRec addin="PrimusSmartCopy"
parameter="|.dft|.ONLYPARENTS.WITHNAME."
visible="CALL(PrimusSmartCopy.IsSmartCopySessionActive)" enabled="TRUE"
text="Status: Kopieren mit gleichn. Zeichnung"
icon="Resources\Icons\newcopy" />
<!-- Nur die Zeichnung bei der das 3D Modell die Indexrreferenz 1 ist, keine
```



```
Komponenten -->
<smartCopySetRec addin="PrimusSmartCopy"
parameter="|.dft|.ONLYPARENTS.REV1."
visible="CALL(PrimusSmartCopy.IsSmartCopySessionActive)" enabled="FALSE"
text="Status: Kopieren mit Ref1 Zeichnung" icon="Resources\Icons\newcopy" />
<!-- Alle Zeichnungen in denen dieses 3D Modell enthalten ist, keine
Komponenten -->
<smartCopySetRec addin="PrimusSmartCopy" parameter="|.dft|.ONLYPARENTS."
visible="CALL(PrimusSmartCopy.IsSmartCopySessionActive)" enabled="FALSE"
text="Status: Kopieren mit allen Zeichnungen" icon="Resources\Icons\newcopy"
/>
<!-- Nur gleichnamige Zeichnung und alle Komponenten -->
<smartCopySetRec addin="PrimusSmartCopy" parameter="|.dft|.WITHNAME."
visible="CALL(PrimusSmartCopy.IsSmartCopySessionActive)" enabled="TRUE"
text="Status: Kopieren mit Komponenten und Zeichnungen"
icon="Resources\Icons\newcopy" />
<!-- Umbenennen mit zugehörigen Zeichnung und aller Komponenten auch der
schreibgeschützten -->
<smartCopySetRec addin="PrimusSmartCopy"
parameter="RENAME|.dft|.INCLREADONLY"
visible="CALL(PrimusSmartCopy.IsSmartRenameSessionActive)" enabled="TRUE"
text="Status: Umbenennen mit Komponenten und Zeichnungen"
icon="Resources\Icons\move" />
```

Siehe auch

- [smartCopySet](#)
- [smartCopySetRec](#)
- [smartcopyignore](#)
- [smartcopyignorerec](#)

[Menübefehl](#), [GUI](#), [Addin](#), [SmartCopy](#)

From:

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

Permanent link:

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

Last update: **2023/02/08 12:01**



smartPaste

Der **Menübefehl smartPaste** gehört zum **PrimusSmartCopy Addin**. Der Befehl fügt die für SmartCopy gewählten Dokumente in das Zielverzeichnis ein.

Aufrufparameter

Die Aufrufparameter werden mit dem **Attribut** parameter, durch | getrennt übergeben. Die Parameter sind in folgender Reihenfolge zu übergeben:

	Parameter	Funktion
1	Einfügemodus	Definiert wie die Dokumente im Zielverzeichnis abgelegt werden sollen
2	PreCondition	Ruft eine Benutzerdefinierte Funktion auf die Vorbedingungen prüft
3	Additional Files	Dateien die zusätzlich mitverarbeitet werden sollen
4	Postaction	Aufruf einer Funktion nach der Dateioperation
5	INCLREADONLY	Es werden auch schreibgeschützte Dateien aktualisiert (Optional)

Einfügemodus

Einfügemodus	Funktion
0	break if subdirectory not exist
1	create not existing subdirectory
2	copy all to subdirectory, if not exist copy to sPasteTargetPath
3	copy all to sPasteTargetPath
4	call checktargetdir method

PreCondition

Ruft eine Funktion eines **Addins** auf, welche etwas benutzerdefiniert prüft, und als Rückgabewert TRUE oder FALSE liefert.

Additional Files

Listet alle **Dateiendungen** auf die ebenfalls mitverarbeitet werden sollen. Die Dateiendungen müssen mit Punkt (.) angegeben werden, dieser ist auch die Trennung zwischen den aufgelisteten Dateiendungen.

Postaction

In der Postaction können nach Abschluss der Dateioperation, noch nachfolgende Funktionen aufgerufen werden. Dazu kann auch auf Funktionen von anderen Addins zugegriffen werden. Es kann

allerdings nicht jede Funktion verwendet werden, da die jeweiligen Funktionen für diesen Zweck vorgesehen sein müssen. Es können mehrere Funktionen eines Addins mit ; getrennt angegeben werden.

Addin	Funktion	Beschreibung
SEInterface	sereplacedocref	Ersetzt die Dateiverknüpfungen in der Solid Edge Datei
	smartupdatepropsandrefsforlastsession(.par.psm.asm.dft.)	Aktualisiert die Dateieigenschaften im Solid Edge Dokument <u>und</u> die Referenzen in der dokument.xml
	smartupdatepropsforlastsession(.par.psm.asm.dft.)	Aktualisiert <u>nur</u> die Dateieigenschaften im Solid Edge Dokument
	smartupdaterefsforlastsession(.par.psm.asm.dft.)	Aktualisiert <u>nur</u> die Referenzen in der dokument.xml

Beispiel

```
<smartPaste addin="PrimusSmartCopy"
parameter="1|Unitec.checkTargetDir|.dft|SEInterface.seReplaceDocRef;smartUpdatePropsAndRefsForLastSession(.par.psm.asm.dft.)|INCLREADONLY"
visible="CALL(PrimusSmartCopy.IsSmartCopySessionActive)AND'CALL(PrimusSmartCopy.IsMarkTarget 1)!=TRUE'" enabled="TRUE"
text="$Primus.MNU_SMCOPY_INSERT_HERE" icon="Resources\Icons\paste" />
```

Menübefehl, GUI, Addin, SmartCopy

From:

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

Permanent link:

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

Last update: **2023/02/08 12:01**



smartRestoreMoveSession

Der [Menübefehl smartRestoreMoveSession](#) ist ein Befehl des [PrimusSmartCopy Addins](#) um eine komplette Verschiebesession rückgängig zu machen. Die Verschiebesession muss die letzte SmartCopy Session sein die auf diesem Rechner ausgeführt wurde. Es besteht keine Möglichkeit eine bestimmte Verschiebesession auszuwählen. Wegen möglicher Veränderung an den Daten nach dem Verschieben, ist es nicht empfehlenswert eine ältere Verschiebesession zu einem späteren Zeitpunkt einfach rückgängig zu machen.

Aufrufparameter

Die Aufrufparameter werden mit dem [Attribut](#) parameter, durch | getrennt übergeben. Die Parameter sind in folgender Reihenfolge zu übergeben:

	Parameter	Funktion
1	SEInterface.seReplaceDocRef	Statischer Parameter

Sichtbarkeit im Menü

Damit der Menüeintrag nur angezeigt wird wenn keine SmartCopy Session aktiv ist, muss dieser mit dem Attribut `visible` und der entsprechenden [Bedingung](#) sichtbar gemacht werden. Dafür stellt das PrimusSmartCopy Addin folgende interne [Bedingungen](#) zur Verfügung. Im Beispiel muss zusätzlich noch die STRG-Taste gedrückt werden damit der Befehl sichtbar wird.

Beispiel

```
<smartRestoreMoveSession addin="PrimusSmartCopy"
visible=" 'CALL(PrimusSmartCopy.IsSessionActive)==FALSE' AND 'CALL(IsCtrlKey) ' "
enabled="TRUE" parameter="SEInterface.seReplaceDocRef" text="Verschieben
rückgängig machen" icon="" />
```

Siehe auch

- [smartCopySetRec](#)
- [smartCopyIgnore](#)
- [smartCopyIgnoreRec](#)
- [smartCopyBreakSession](#)
- [smartCopyRenameDir](#)

[Menübefehl](#), [GUI](#), [Addin](#), [SmartCopy](#)

From:

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

Permanent link:

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

Last update: **2023/04/14 11:57**



Den Start von Primus beschleunigen

Um den Start von [Primus](#) zu beschleunigen gibt es mehrere Faktoren die man beeinflussen kann.

Lesen der .PRO.XML des obersten Ordners

Primus versucht stets zu dem obersten Ordner der Pathdefinition die zugehörige [PRO.xml](#) zu lesen. Dies ist aber nicht immer möglich, z.B. wenn der oberste Pfad die Freigabe selbst ist ([\\primus\Daten](#)). In diesem Fall versucht Primus diese nicht nur zu lesen sondern ggf. auch anzulegen, was naturgemäß bei einer [Freigabe](#) nicht möglich ist. Mit dem [Attribut](#) readtopxml von [Path](#) kann man dies steuern.

Wert	Bedeutung
0	Primus versucht nicht die .PRO.XML des in Path definierten Verzeichnisses zu lesen und diese auch nicht anzulegen
1	Die .PRO.XML wird gelesen und ggf. neu angelegt. Dies ist der Defaultwert wenn das Attribut nicht angegeben ist

Siehe auch

- [Tipps & Tricks](#)

[Tipps&Tricks](#), [Performance](#)

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

Permanent link:
https://dokuwiki.unitec.de/doku.php?id=primus:start_von_primus_beschleunigen

Last update: 2025/03/18 15:40



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	PreCalls	Funktionen die vor dem zuweisen des Status aufgerufen werden und z.B. prüfen ob alle Vorbedingungen erfüllt sind
2	SetDataValues	Enthält die Felder und die Werte die einem Dokument mit diesem Status zugewiesen werden
3	PostCalls	Funktionen die nach dem Zuweisen dieses Status aufgerufen werden
4	TargetStatus	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
CopyFile	Kopiert eine bestimmtes Dokument unter Angabe von Quell- und Zielpfad
SetParentToStatus	Bezieht auch übergeordnete Elemente in den Statuswechsel mit ein und darf nur in den PostCalls verwendet werden
SetXMLValue	Setzt ein bestimmtes Datenfeld eines Dokuments auf einen Wert
seUpdatePropertiesByPath	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
CheckChildrenStatus	Prüft ob verlinkte untergeordnete Dokumente einen bestimmten Status haben
CheckParentsStatus	Prüft ob verlinkte übergeordnete Dokumente einen bestimmten Status haben
CheckParentsWriteable	Prüft ob übergeordnete Elemente beschreibbar sind
CheckXMLValue	Prüft ob ein Feld einen bestimmten Wert hat
IgnoreStatus	Dokumente mit bestimmten Status werden für den Workflow ignoriert
SetChildrenToStatus	Bezieht auch untergeordnete Elemente in den Statuswechsel mit ein
SetPrevVersionToStatus	Setzt das Dokument der vorherigen Revision auf einen bestimmten Status
SetSubFilesToStatus	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**

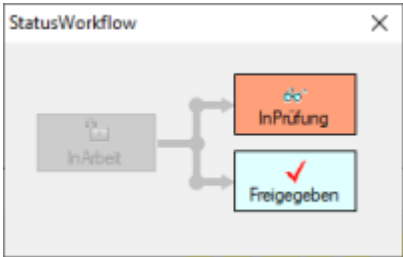


StatusWorkflow_Settings

In **StatusWorkflow_Settings** werden die verschiedenen Status definiert die ein Dokument in Primus während seines [Lebenszyklus](#) annehmen kann.

Buttons

Mit dem [Leer-Tag Buttons](#) wird über [Attribute](#) definiert wie die einzelnen Schaltflächen angeordnet werden und welche Größe und Abstände sie haben sollen.



Attribut	Wert	Funktion
orientation		Ausrichtung der Schaltflächen
	H	Horizontal
	V	Vertikal
button_height	<PIXELANZAHL>	Höhe einer Schaltfläche
button_width	<PIXELANZAHL>	Breite einer Schaltfläche
topbottom_margin	<PIXELANZAHL>	Abstand zum oberen und unteren Rand
leftright_margin	<PIXELANZAHL>	Abstand zum rechten und linken Rand
ver_dist	<PIXELANZAHL>	Senkrechter Abstand von Oberkant zu Oberkante
hor_dist	<PIXELANZAHL>	Waagrechter Abstand von linker Kante zu linker Kante

Beispiel

```
<Buttons orientation="H" button_height="40" button_width="80"
topbottom_margin="20" leftright_margin="20" ver_dist="10" hor_dist="50" />
```

Hinweis

Bleibt dieser Dialog leer, es werden also keine Schaltflächen angezeigt, dann existiert keine Wertzuweisung für das Datenfeld [WorkflowStatus](#).

Workflow

Im Abschnitt **Workflow** werden die verschiedenen [Status](#) jeweils als eigene Abschnitte definiert. Der Abschnitt hat zwei Attribute die Pflichtfelder sind. In dem Abschnitt ist **ein einziger Kommentar**

erlaubt.

Kommentierbar

Attribut	Wert	Funktion
field	WorkflowStatus ²⁷⁾	Datenfeld in dem der Workflowstatus gespeichert ist
list	List_WorkflowStatus ²⁸⁾	Liste in der Texte, Symbole und Farben für den Status definiert sind

Beispiel

```
<Workflow field="WorkflowStatus" list="List_WorkflowStatus">
```

gui, workflow

²⁷⁾ ²⁸⁾
,
Defaultwert

From:

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

Permanent link:

https://dokuwiki.unitec.de/doku.php?id=primus:statusworkflow_settings

Last update:

2023/02/08 12:01



Stringoperatoren

Mit **Stringoperatoren** ist es möglich in Teile von Zeichenfolgen (Strings) abzutrennen und zu neuen Strings zusammenzusetzen. Der Stringoperator wird mit ein Doppelpunkt (:) direkt hinter die betreffende Stringvariable geschrieben. Der Vergleichsstring wird direkt an den Operator, durch ein | getrennt, angehängt. Der Aufbau ist *Operator|Position|Trennzeichen*. Ausnahme Between, da werden zwei Positionen übergeben.

Liste der Stringoperatoren

Folgende Operatoren stehen zur Verfügung:

Operator		Funktion
after	first	Liefert den Teilstring rechts des ersten Trennzeichens
	last	Liefert den Teilstring rechts des zuletzt (ganz rechts) vorkommenden Trennzeichens
befor	first	Liefert den Teilstring links des ersten Trennzeichens
	last	Liefert den Teilstring links des zuletzt (ganz rechts) vorkommenden Trennzeichens
between	first	Liefert den Teilstring zwischen den beiden genannten Trennzeichen
	second	
	last	Liefert den Teilstring zwischen den angegeben Zeichen. Ist nur eines angegeben, wird zwischen zwei Vorkommen dieses Zeichen gesucht
	two	
FillToLen	L	Füllt einen String links oder rechts mit einem bestimmten Zeichen auf die angegebene Länge auf ²⁹⁾
	R	
replace		Ersetzt ein Zeichen in einem String, durch ein anderes
r	ZAHL	Entfernt (remove) hinter der n. Stelle von links, Ergebnis ist der linke Teil des Strings
s	ZAHL	Teilstring (substring) ab der n. Stelle von links, Ergebnis ist der Teil rechts davon

Beispiele

Hier wird eine Eigenschaft mit der Benennung ArtNr mit dem Dateinamen beschrieben, der allerdings nach dem letzten „-“ abgekürzt wird.

```
<ArtNr value="%FILENAME:bevor|last|-%" />
<!-- Liefert den Teilstring zwischen einem _ und einem - -->
<DocLang value="%FILENAME:between|two|_-%" />
<!-- Teilstrings -->
<!-- Schneidet die Nummer hinter dem zweiten Zeichen ab -->
<ArtNr value="#(Number:r|2)" />
<!-- Nimmt den Teilstring ab dem zweiten Zeichen und schneidet davon hinter dem dritten Zeichen weg -->
<ArtNr value="#(Number:s|2:r|3)" />
```

Abweichende Syntax bei FillToLen

Im Beispiel wird die *Number* Linksbündig mit 0 auf 7 Stellen aufgefüllt und anschließend um die Revision ergänzt.

```
<Sample filename="CALL(FillToLen #(Number)|L|0|7)#(Revision)" />
```

Alternatives Pipe Zeichen bei Verwendung in Parameter

An einigen Stellen, z.B. in [Jobserver Parametern](#), darf das Pipe Zeichen jedoch nicht verwendet werden da es die übergeordneten Parameter unterbricht. In dem Fall wird das erste Pipe durch ein „#“ und folgende durch die unterbrochene Pipe (| ALT+0166) ersetzt. Beispiel:

```
. \#(Number:replace#.|-)
```

Teilstrings bei %FILENAME%

Soll vom Dateiname nur ein Teilstring verwendet werden, dann darf kein Pipesymbol (|) verwendet werden.

```
<!-- Substring ab Stelle 7 -->  
<ArtNr value="%FILENAME:s7%" />  
  
<!-- Remove ab Stelle 7: -->  
<ArtNr value="%FILENAME:r7%" />
```

Siehe auch

- [Stringvergleichsoperatoren](#)

[Stringfunktion](#)

²⁹⁾

Abweichende Syntax beachten

From:

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

Permanent link:

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

Last update: **2023/12/15 16:27**



Stringvergleichsoperatoren

Mit **Stringvergleichsoperatoren** ist es möglich in **Bedingungen** nur Teile von Zeichenfolgen (Strings) zu vergleichen. Der Vergleichsoperator wird mit ein Doppelpunkt (:) direkt hinter die betreffende Stringvariable geschrieben. Der Vergleichsstring wird direkt an den Operator, durch ein | getrennt, angehängt.

Beispiel

```
<SmartCopy  
componentfilter="'%FILEPATH:sw|%SOURCETOPDIRPATH%%'=='%SOURCETOPDIRPATH%' ">
```

Erklärung

ZEICHENFOLGE ist z.B. der aktuelle FILEPATH und **VERGLEICHSTRING** ist der gesuchte SOURCETOPDIRPATH aus obigem Beispiel.

Dieser Term **%ZEICHENFOLGE:sw|%VERGLEICHSTRING%%** gibt den **%VERGLEICHSTRING%** zurück wenn die **%ZEICHENFOLGE** damit beginnt, sonst wird **leer** zurückgegeben.

Daher: **%ZEICHENFOLGE:sw|%VERGLEICHSTRING%%'=='%VERGLEICHSTRING%**

Liste der Vergleichsoperatoren

Folgende Operatoren stehen zur Verfügung:

Operator	Bedeutung	Funktion
cn	Contains	Prüft ob der Vergleichsstring in der Zeichenfolge enthalten ist
~		Verkürzte Schreibweise. Der Ausdruck !~ bedeutet <i>Contains not</i>
CN		Prüft unter Berücksichtigung der Groß- / Kleinschreibung
ew	Ends with	Prüft ob die Zeichenfolge mit dem Vergleichsstring endet
sw	Starts with	Prüft ob die Zeichenfolge mit dem Vergleichsstring beginnt
il	in List	Prüft ob der Zeichenfolge in einer Liste aufgeführt ist. Hier wird anstelle des Vergleichsstrings die Liste angegeben

Siehe auch

- [Stringoperatoren](#)

[Stringfunktion](#), [Bedingung](#), [SmartCopy](#)

From:

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

Permanent link:

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

Last update: **2023/02/08 12:01**



SynchroniseDocumentDataFields

Bei **SynchroniseDocumentDataFields** handelt es sich um einen [Abschnitt](#) in dem definiert wird welche [datenfelder](#) der [daten-xml](#) eines Quelldokuments mit welchen Datenfelder eines zugehörigen Zieldokuments verknüpft werden sollen. Der Abschnittsname ist vorgegeben und wird von Primus intern zum Synchronisieren bei Änderungen genutzt. Für den [Menübefehl](#) [prmProjectSynchroniseAllDataFields](#) kann er im Attribut parameter angegeben werden.

Attribute

Im [Start-Tag](#) können folgende Attribute angegeben werden:

Attribut	Wert	Funktion
load	XML_Document.Ref1	Gibt an aus welcher Daten-XML die Werte entnommen werden sollen
postcondition	'XML_Document.RefCount>=1'	Klärungsbedarf
precondition	'#(FILE_TYPE)==DFT'	Vorbedingung die für ein Zieldokument erfüllt sein muß, damit synchronisiert wird. Im Beispiel muß es eine solid_edge Draftdatei sein.

Beispiel

```
<SynchroniseDocumentDataFields precondition="'#(FILE_TYPE)==DFT' "  
load="XML_Document.Ref1" postcondition="'XML_Document.RefCount>=1' ">
```

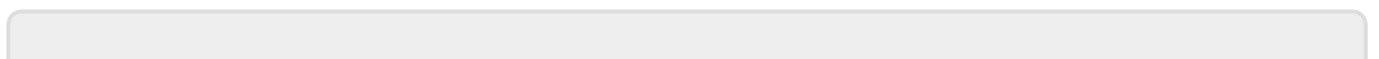
Datenfeldeinträge

Es können beliebig viele Datenfeldeinträge in den Abschnitt eingetragen werden. Alle Einträge sind als [Leer-Tag](#) nach folgendem Schema aufgebaut: [Tag-Name](#) ist der Name des XML-Datenfelds und einziges Attribut ist value.

```
<Title value="%XML_Document.Ref1:Title%" />
```

Wertzuweisungen

Folgende Wertzuweisungen sind möglich:



From:

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

Permanent link:

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

Last update: **2023/02/08 12:01**



Templates

Im [Abschnitt Templates](#) werden die Vorlagen für Dokumente definiert, die in Primus neu erzeugt werden können.

Templates ausblenden

Die mitgelieferten Standardeinstellungen sollen nicht entfernt werden. Ist es trotzdem gewünscht einzelne Templates auszublenden, dann kann dazu das Attribut **visible** verwendet werden.

From:

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

Permanent link:

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

Last update: **2023/02/08 12:01**



textbox

Eine **textbox** ist ein [GUI Element](#) für [DataSheets](#), das dem Anwender die freie Eingabe von Werten ermöglicht. Der Eintrag erfolgt im Attribut **ctrltype**.

Zusätzliche Attribute

Eine Textbox hat folgende spezifische Attribute.

Attribut	Wert	Pflicht	Funktion
maxlength	<ZAHL>	Nein	Begrenzt die Zeichenanzahl die eingegeben werden kann

Beispiel

```
<Number ctrltype="textbox" data="Number" label="$Primus.SHEET_DOC_NUMBER"
tooltip="Number Format?" type="Text" maxlength="50" />
```

Siehe auch

- [listbox](#)
- [combobox](#)

[GUI](#), [DataSheet](#), [ctrltype](#)

From:

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

Permanent link:

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

Last update:

2024/03/26 09:06



Tipps & Tricks

Hier werden verschiedene **Tipps & Tricks** aufgelistet mit denen man [primus](#) optimieren kann.

Liste der Tipps & Tricks

- [Start von Primus beschleunigen](#)
- [Wechsel von Primus 7 auf Primus 2021](#)
- [Bedingte Werte von Primus an Solid Edge übertragen](#)

Siehe auch

- [Beispiele](#)

[Tipps&Tricks](#)

From:

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

Permanent link:

https://dokuwiki.unitec.de/doku.php?id=primus:tipps_tricks

Last update: **2023/12/19 14:52**



type

Bei **type** handelt es sich um **Attribut** von **GUI_Data_Sheets** und **GUI_Data_Lists** Elementen. Der type definiert in welcher Form der Inhalt eines **Datenfelds** darzustellen ist. Die Namen der types sind grundsätzlich wie aufgelistet zu schreiben.

Name	Funktion	Zusätzliches Attribut
Button ³⁰⁾	Definiert eine Schaltfläche	ctrltype
filetypeicon	Klärungsbedarf	list
iconlist	Zeigt ein abhängig vom in data gespeicherten Inhalt ein Symbol an	list
list	Zeigt ein abhängig vom in data gespeicherten Inhalt eine Text aus der Liste an	list, listchildren
listchild	Listet abhängig von der Auswahl im übergeordneten Listfeld , Listeneinträge auf	listchildren
rwicon	Klärungsbedarf	list
text	Zeigt den in data gespeicherten Inhalt an, oder optional einen Wert aus einer Liste	list (optional)

Beispiele

```
<FileName ctrltype="textbox" type="Text" />
<FileTypeIcon ctrltype="icon" data="FILE_TYPE" type="FileTypeIcon"
list="List_FileTypes" />
```

Siehe auch

- [ctrltype](#)

[gui](#), [DataSheet](#),, [DataList](#), [type](#)
³⁰⁾

Nur in DataSheets

From:

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

Permanent link:

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

Last update:

2024/09/05 07:55



Wechsel von Primus 7 auf Primus 2021

Beim **Wechsel von Primus 7 auf Primus 2021** ändern sich folgende Einträge in der [CFG.XML](#):

Primus 7	Primus 2021
Abschnitte	
- GUI	
- GUI_Data_Lists	
GUI_DocumentBOM_ListModel	GUI_Document_List_BOM
GUI_DocumentBOM_ListStruct	GUI_Document_List_BOMStruct
- GUI_Data_Sheets	
GUI_DocumentBOM_DataSheet	GUI_Document_DataSheet_BOM
Options	
- Public	
CheckActiveLicense	CheckActiveLicenseType

Siehe auch

- [Wechsel von Primus 2021 auf Primus 2025](#)

[Umstellung P7 auf P2021](#)

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

Permanent link:
https://dokuwiki.unitec.de/doku.php?id=primus:wechsel_von_primus_7_auf_primus_2021

Last update: **2025/11/25 09:20**



Wechsel von Primus 2021 auf Primus 2025

In diesem Thema wird beschrieben was beim Wechsel von Primus 2021 auf Primus 2025 zu beachten ist.

PrimusSearchService

Der [PrimusSearchService](#) wird durch den [PrimusDBService](#) ersetzt.

Erweiterung der Primus.xml

Attribute

Folgende [Attribute](#) sind in den einzelnen Abschnitten hinzugekommen:

Abschnitt	Attribut	Funktion
PrimusSettings		
	licenseactivation	Steuert wo die Aktualisierung erfolgt Klärungsbedarf
	licensecheck_message_users_OFF	Steuert welche Benutzer, bei einer fehlgeschlagenen Online-Aktivierung der Lizenzen, eine Fehlermeldung erhalten
MainWindow		
	big_icons	Steuert ob große Icons angezeigt werden sollen

Setup

In der [Primus.xml](#) müssen folgende Zeilen eingefügt werden:

```
<CopyFile from="%SERVER%\Primus.exe.config" to="%CLIENT%\Primus.exe.config"
overwrite="1" />
<CopyFile from="%SERVER%\Primus.exe.manifest"
to="%CLIENT%\Primus.exe.manifest" overwrite="1" />
```

Icons

Die neue Version unterstützt Icons in verschiedenen Größen. Bisher war nur eine Auflösung von 16×16 Pixeln vorgesehen. Es werden entsprechende Icon von CADAdapt mitgeliefert, Kundenspezifische Icons müssen angepasst werden.

Benutzerspezifische Addins

Für ältere Primusversionen erstellte [Addins](#) müssen für Primus 2025 unter Verwendung von .NET 4.8 neu kompiliert werden.

Siehe auch

- [Wechsel von Primus 7 auf Primus 2021](#)

[Umstellung P2021 auf P2025](#)

From:

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

Permanent link:

https://dokuwiki.unitec.de/doku.php?id=primus:wechsel_von_primus_2021_auf_primus_2025

Last update: **2026/02/11 10:15**



Weiteres Datenfeld anlegen

In diesem Artikel wird beschrieben wie ein zusätzliches Datenfeld in eine bestehende [Primuskonfiguration](#) eingefügt wird und an welchen Stellen Änderungen vorgenommen werden müssen. Dabei ist zu beachten das abhängig vom Zweck des Felds verschiedene Anpassungen gemacht werden.

Beispielsituation

Der Kunde möchte ein zusätzliches Datenfeld z.B. für die ERP-Nummer haben, die bisherige Zeichnungsnummer bleibt davon jedoch unberührt.

Änderungen in der CFG.XML

In folgenden [Abschnitten](#) müssen/können Einträge hinzugefügt werden.

Abschnitt	Pflicht	Beschreibung
XML_DataTemplates		Definition der Felder die beim anlegen einer XML direkt mit erzeugt werden, Primus würde das Feld aber auch anlegen, wenn das Feld das erste mal mit einem Wert gefüllt wird.
GUI_Document_List		Der Eintrag ist notwendig damit nach dem Feld gesucht und gefiltert werden kann, außerdem wird eine eventuelle Listendefinition für die Übertragung mittels AppAttributes hier definiert.
GUI_Document_List_BOM		Klärungsbedarf
GUI_Document_List_BOMStruct		Klärungsbedarf
GUI_Document_DataSheet		Hier kann der Inhalt des Datenfelds geändert werden
GUI_NewDocument_DataSheet		Wenn das Feld beim neu Anlegen einer Datei vom Anwender ausgefüllt werden muss
GUI_Document_DataSheet_BOM		Klärungsbedarf
FastFilter		Stellt das Feld im Filter für die Suche dar
Options/PreAssignData		Hier muss das Datenfeld in die verschiedenen Abschnitte, die relevant sind eingetragen werden.
SynchroniseDocumentDataFields		Übertragung des Felds vom 3D Modell zur 2D Zeichnung
StatusWorkflow_Settings		Ggf. Eintragung in PreCalls zum Prüfen des Felds oder in den PostCalls falls das Feld als Name für Exportformate in Betracht kommt.

Elementtext

Für das Feld wird in den Datenblättern ein Label und in den Listen eine Überschrift benötigt, diese muß als [Elementtext](#) definiert werden.

PrimusSearchService

Damit nach dem Feld gesucht werden kann muss es im [PrimusSearchService](#) eingetragen und dann von diesem indiziert werden.

Abgleich zwischen Primus und CAD

Damit das Feld zwischen Primus und dem CAD System abgeglichen wird, muss es in die Konfigurationsdateien für [AppAttributes](#) eingetragen werden.

[HowTo](#), [Beispiel](#)

From:

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

Permanent link:

https://dokuwiki.unitec.de/doku.php?id=primus:weiteres_datenfeld_anlegen

Last update: **2024/07/31 14:43**



Workflow

Mit **Workflow** werden im [Abschnitt Public](#), verschiedene Optionen für die Versionierung von Dokumenten definiert.

Attribute

Über folgende Attribute kann festgelegt werden was bei Versionsänderungen passieren soll. Die Attribute können alle in einem einzigen Workflow-[Tag](#), oder aber auch auf mehrere verteilt werden, was die Übersichtlichkeit erhöht.

Attribut	Wert	Funktion
newvercall	<FUNKTION>	Interne oder externe Primusfunktion die bei der Versionierung aufgerufen werden soll um die neue Version zu bearbeiten
newvercall_cond	<BEDINGUNG>	Bedingung unter der eine neue Version zulässig ist Klärungsbedarf
newversioncond	<BEDINGUNG>	Bedingung unter der eine neue Version zulässig ist Klärungsbedarf
oldvercall	<FUNKTION>	Interne oder externe Primusfunktion die bei der Versionierung aufgerufen werden soll um die bestehende (alte) Version zu bearbeiten
oldvercall_cond	<BEDINGUNG>	Bedingung unter der oldvercall ausgeführt wird
oldverstatus	<STATUS>	Nummer des Status den die bestehende Version im Falle einer Versionierung bekommen soll
revision_firstnewver	<WERT>	Wert den die erste Revision eines Dokuments erhält
revision_newdoc	<WERT>	Numerischer oder alphanumerischer Wert, den ein neu angelegtes Dokument bekommt
revision_split	<WERT>	Zeichen oder Zeichenfolge mit der die Revision vom Dateinamen getrennt ist
synchronize_ref	<PARAMETER>	Definiert welcher Referenzen mit versioniert werden

newvercall

Mit dem Attribut **newvercall** wird definiert welche Funktion aufgerufen werden soll, um die neue Version des Dokuments zu bearbeiten, z.B. um den Status in der Datei zusetzen. Für [solid_edge](#) kann beispielsweise die Funktion [seSetSolidEdgeFileStatusByPath](#) des [Addins SEInterface](#) genutzt werden.

```
<Workflow> newvercall="SEInterface.seSetSolidEdgeFileStatusByPath  
%FILEPATH%|0" />
```

oldvercall

Mit dem Attribut **oldvercall** wird definiert welche Funktion aufgerufen werden soll, um die bestehende (alte) Version des Dokuments zu bearbeiten, z.B. um den Status in der Datei zusetzen.

seSetSolidEdgeFileStatusByPath

Für Solid Edge kann beispielsweise die Funktion [seSetSolidEdgeFileStatusByPath](#) des [Addins SEInterface](#) genutzt werden. Dabei wird der Status der Solid Edge Datei direkt gesetzt.

```
<Workflow> newvercall="SEInterface.seSetSolidEdgeFileStatusByPath  
%FILEPATH%|0" />
```

StatusChangeByFilePath

Das oldverstatus wird nicht mehr direkt gesetzt. Es wird nur die Methode StatusChangeByFilePath aufgerufen, die einen echten Statusübergang mit PreCalls und PostCalls ausführt. Dazu muss aber im Gegensatz zu [seSetSolidEdgeFileStatusByPath](#) eine entsprechende Workflowdefinition in [StatusWorkflow_Settings](#) existieren.

```
<Workflow oldverstatus="" oldvercall="StatusChangeByFilePath  
%FILEPATH%|StatusWorkflow_Settings|004" />
```

prmExecuteNamedCalls

Es ist auch möglich einen [NamedCall](#) zu definieren und hier aufzurufen. Der Unterschied zu einer Workflowdefinition ist, dass z.B. keine PreCalls überprüft werden.

```
<Workflow oldverstatus="" oldvercall="prmExecuteNamedCalls  
%FILEPATH%|OldVerCall" />
```

oldvercall_cond

Die Angabe des Attributs `oldvercall_cond` ist optional. Diese Bedingung definiert ob `oldvercall` für das Dokument ausgeführt wird.

Wichtig

Wenn dieses Attribut definiert ist, wird nicht der Statusworkflow der unter [StatusWorkflow_Settings](#) definiert ist aufgerufen. Es kann nur entweder der Workflowstatus oder direkt die hier definierte [funktion](#) aufgerufen werden.

synchronize_ref

Das Tag **synchronize_ref** hat folgende Parameter, welche in der angegebenen Reihenfolge, getrennt durch `|`, angegeben werden müssen.

Parameter	Wert	Funktion
1	ref1	Revisioniert alle Dokumente in denen dieses Dokument als erste Indexreferenz eingetragen ist
	refwithname	Revisioniert nur die Dokumente die auf dieses Dokument referenzieren und den gleichen Namen haben
2	<DATEITYP>	Definiert welchen Dateityp die referenzierenden Dokumente haben müssen, z.B. nur .dft

```
<Workflow synchronize_ref="refwithname|.dft" />
```

Workflow

From:

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

Permanent link:

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

Last update:

2023/06/16 07:24



Benutzerhandbuch



Hier wird beschrieben wie die Benutzerschnittstelle von Primus aufgebaut ist und vom Anwender angepasst werden kann. Dazu zählen u.a. das [Menü](#), die [Listenansichten](#), die [Datenblattansichten](#), die [Suche](#), und die [Benutzeroberfläche](#). Es wird aber auch die Handhabung von Funktionen wie [SmartCopy](#) beschrieben. Hier geht es zur vorläufigen [Liste der Hilfethemen](#).

Hilfe zum Wiki

Es gibt eine [deutsche Beschreibung](#) zu diesem Wiki, dort wird u.a. beschrieben wie Texte [formatiert](#) werden.

Glossar

Für die unvermeidlichen Fachbegriffe gibt es hier ein [Glossar](#). Für das bessere Verständnis sollten die Begriffe durchgängig auch in diesem Wiki verwendet werden.

From:

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

Permanent link:

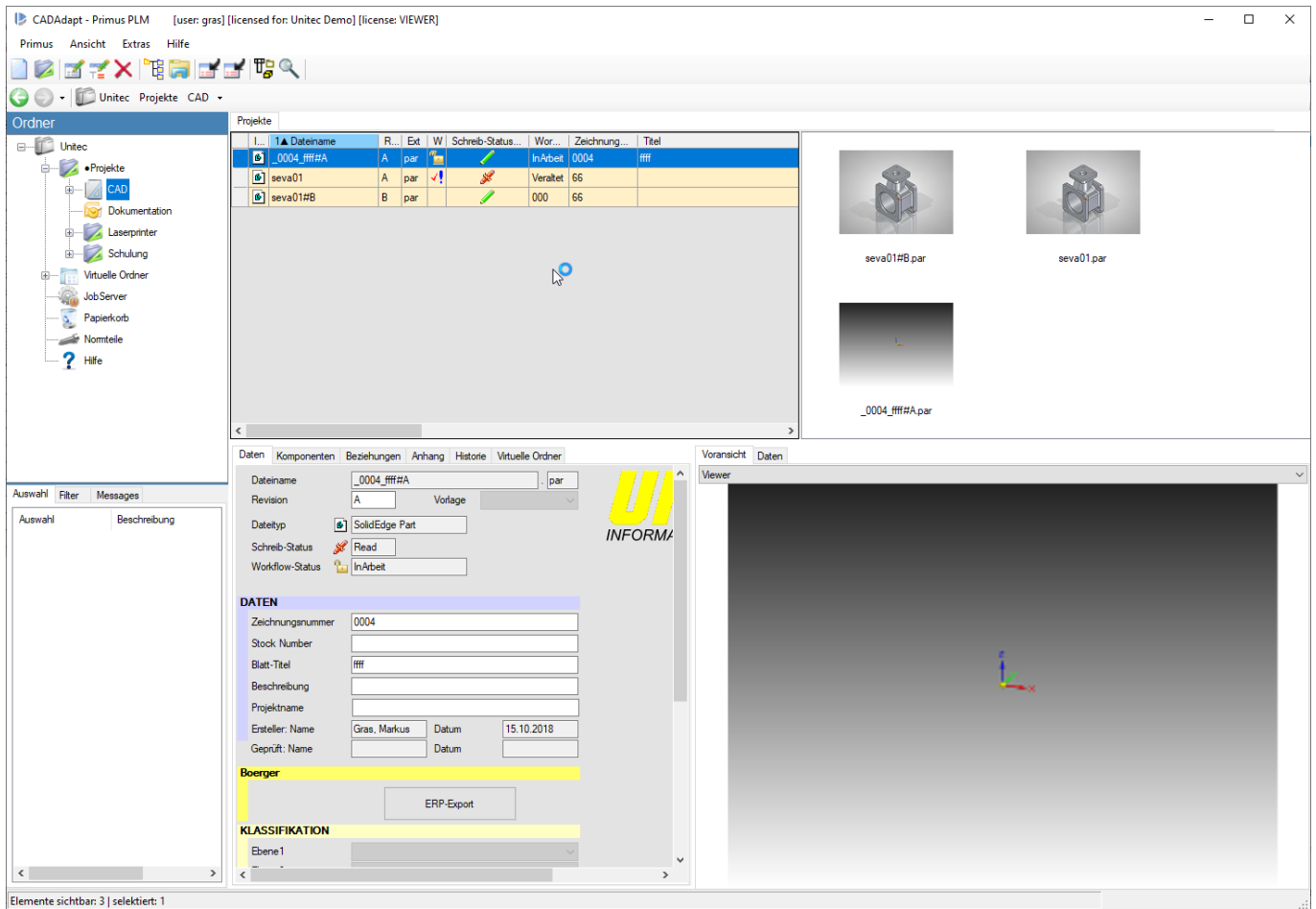
<https://dokuwiki.unitec.de/doku.php?id=userman:start>

Last update: **2023/02/08 12:01**



Benutzeroberfläche

Die Benutzeroberfläche von Primus ist folgendermaßen aufgebaut. Zum Anzeigen auf die Grafik deuten, für eine ausführliche Hilfe bitte klicken. Weitere Befehle erreichen Sie über das [Kontextmenü](#).



- Ordnerfenster
- Dokumentenliste
- Vorschauliste
- Filter & Selektion
- Registerkartenfenster
- Vorschauenfenster

From:

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

Permanent link:

<https://dokuwiki.unitec.de/doku.php?id=userman:benutzeroberflaeche>

Last update: **2023/02/08 12:01**



Dokumentenliste

Die Dokumentenliste ist eine Liste aller Dokumente im ausgewählten Ordner. Oberhalb der Dokumentenliste ist die Dateinamenliste zu sehen, wo alle aufgelisteten Dateien mit verschiedenen Endungen aber gleichem Dateinamen nur einmal angezeigt werden und so eine Vorselektion aller gleichnamigen Dokumente möglich ist.

Projekte

I...	1▲Dateiname	R...	Ext	W	Schreib-Status...	Wor...	Zeichnung...	Titel
	0004_mff#A	A	par			InArbeit	0004	mff
	seva01	A	par			Veraltet	66	
	seva01#B	B	par			000	66	

From:

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

Permanent link:

<https://dokuwiki.unitec.de/doku.php?id=userman:dokumentenliste>

Last update: **2023/02/08 12:01**

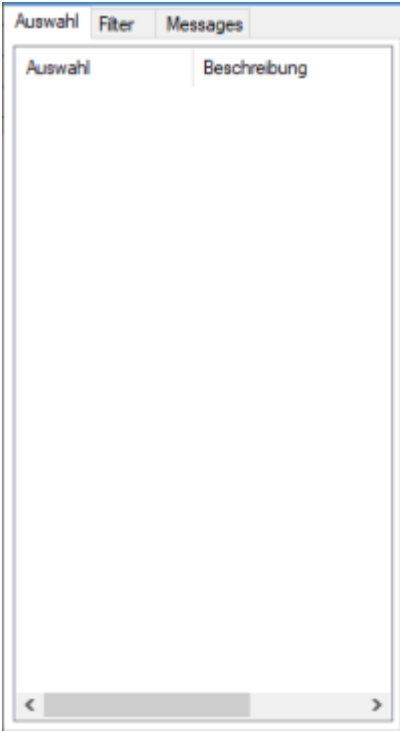


Filterfenster

Im **Filterfenster** der Primus [Benutzeroberfläche](#) hat der Anwender Zugriff auf gespeicherte [Suchabfragen](#), die [Filterfunktion](#) und [Primus Meldungen](#). Dazu ist hat das Fenster drei [Register](#) mit denen zwischen den verschiedenen Bereichen umgeschaltet werden kann.

Auswahl

In diesem Register kann die aktuelle Auswahl des [Ordnerfensters](#) und der [Dokumentenliste](#) abgespeichert werden. Dazu muß mit der rechten Maustaste in den freien Bereich des Registers geklickt werden und mit **Aktuelle Auswahl und Filter speichern** abgespeichert werden. Dabei kann ein Name eingegeben werden.



Filter

Im Register **Filter** ist es möglich die Anzeige in der Dokumentenliste nach bestimmten Suchkriterien zu filtern. Das Fenster enthält oben zwei [Auswahllisten](#), die obere für vorkonfigurierte Filter und die untere für den Suchbereich. Folgende Suchbereiche stehen zur Verfügung:

Suchbereich	Beschreibung
Nur aktiven Ordner	Listet nur Dokumente auf, die in dem im Ordnerfenster gewählten Verzeichnis direkt enthalten sind.
Mit Unterordner	Listet alle Dokumente auf, die in dem im Ordnerfenster gewählten Verzeichnis und dessen Unterverzeichnissen enthalten sind.
Vom obersten Ordner	Listet alle Dokumente auf, beginnend mit dem obersten Ordner und allen Unterverzeichnissen. Quasi eine Suche über alles.

In die jeweiligen Suchfelder können Suchbegriffe eingegeben werden, wobei die Begriffe nicht vollständig ausgeschrieben werden müssen, da bei der Suche alle Dokumente aufgelistet werden, bei denen das jeweilige Feld den gesuchten Begriff **enthält**. Die eingabe eines Platzhalters (* oder ?) am Anfang oder Ende des Begriffs ist nicht erforderlich.

Dateiname	
Titel	
Beschreibung	
Zeichnungsnum...	

Suche mit Platzhalter

Es ist jedoch möglich im Filterfeld auch Platzhalter zu verwenden.

- z.B. T*A

Dadurch werden alle Werte wie

- T1A
- T2A
- TestA

Messages

In diesem Register werden Meldungen von Primus angezeigt, welche den Indexdienst betreffen. Hier wird angezeigt welcher Suchindex beim Start von Primus geladen wurde bzw. ob dies erfolgreich war. Bei Schwierigkeiten mit der Suche ist es empfehlenswert zunächst hier zu prüfen ob der Index aktiv und verbunden ist.

```

Auswahl  Filter  Messages
Search db connected: PDM
Connecting search db: PDM
  
```

Siehe auch

- [Spaltenfilter](#)
- [Erweiterte Suche](#)

[Hilfethema](#)

From:

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

Permanent link:

<https://dokuwiki.unitec.de/doku.php?id=userman:filterfenster>

Last update: **2023/02/08 12:01**



Kontextmenü

Die Befehle werden im Kontextmenü angezeigt und von dort aus gestartet. Das Kontextmenü kann nach der Auswahl eines Ordners oder Dokuments mit der rechten Maustaste (RMT) aufgerufen werden.

Befehle bei ausgewähltem Ordner



From:

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

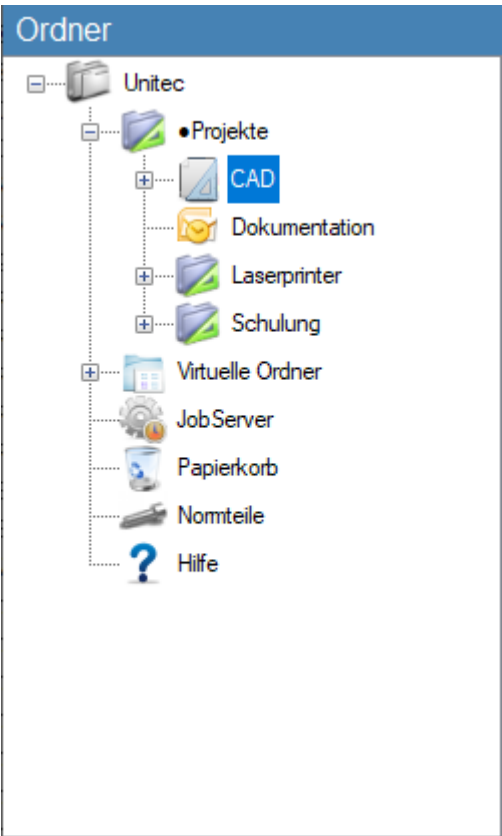
Permanent link:

<https://dokuwiki.unitec.de/doku.php?id=userman:kontextmenue>

Last update: **2023/02/08 12:01**



Ordnerfenster



- [Primus Konfigurationsordner](#)
- [Projekte](#)
- [Virtuelle Ordner](#)
- [JobServer](#)
- [Papierkorb](#)
- [Normteile](#)
- [Hilfe](#)

Im **Ordnerfenster** werden die Datenverzeichnisse von Primus angezeigt. Es gibt verschiedene Datenverzeichnisse die sich von Konfiguration zu Konfiguration unterscheiden können.

Ordner	Bedeutung
Unitec	Oberster Primus Konfigurationsordner
Projekte	Hier werden die Dokumente abgelegt
Virtuelle Ordner	Liste der Virtuellen Ordner
JobServer	Zeigt offene und bereits abgearbeitete Jobs
Papierkorb	Hier liegen Dokumente die in Primus gelöscht werden
Normteile	Katalog der ISL Normteile (optional)
Hilfe	Hilfedokumentation zu Primus

From:

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

Permanent link:

<https://dokuwiki.unitec.de/doku.php?id=userman:ordnerfenster>

Last update: **2023/02/08 12:01**



Registerkartenfenster

In der **Registerkartenfenster** können zusätzliche Informationen zu dem angewählten Dokument in der [Dokumentenliste](#) eingesehen und ggf. verändert werden. Im Registerkartenfenster stehen, verschiedene Registerkarten zur Verfügung:

- [Datenblatt](#)
- [Komponenten](#)
- [Beziehungen](#)
- [Anhang](#)
- [Historie](#)
- [Virtuelle Ordner](#)

The screenshot displays the 'Daten' (Data) tab of the 'Registerkartenfenster' (Tabbed Window) in the Informa software. The window is divided into several sections. At the top, there are tabs for 'Daten', 'Komponenten', 'Beziehungen', 'Anhang', 'Historie', and 'Virtuelle Ordner'. The 'Daten' tab is active, showing a form with various fields. The 'Dateiname' field contains '_0004_ffff#A' and the 'Revision' field contains 'A'. The 'Datentyp' field is set to 'SolidEdge Part'. The 'Schreibe-Status' field is 'Read' and the 'Workflow-Status' field is 'InArbeit'. Below these are sections for 'DATEN' (Data) and 'KLASSIFIKATION' (Classification). The 'DATEN' section includes fields for 'Zeichnungsnummer' (0004), 'Stock Number', 'Blatt-Titel' (ffff), 'Beschreibung', 'Projektname', and 'Ersteller: Name' (Gras, Markus) and 'Datum' (15.10.2018). The 'KLASSIFIKATION' section includes a dropdown for 'Ebene1'. A yellow bar at the bottom of the 'DATEN' section contains the text 'Boerger' and an 'ERP-Export' button. The Informa logo is visible in the top right corner.

Datenblatt

Im Datenblatt werden alle [Metadaten](#) als Felder übersichtlich angezeigt

Komponenten

Im Register **Komponenten**, werden alle zu einem in der [Dokumentenliste](#) markierten Dokument zugehörigen Dokumente angezeigt. Bei CAD Baugruppen sind dies alle darin enthaltenen Dokumentverknüpfungen ungeachtet ihrer Relevanz für Stücklisten. Wie Stücklisten mit Primus erstellt werden steht im Artikel [Stückliste erstellen](#).

GUI-Element

From:

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

Permanent link:

<https://dokuwiki.unitec.de/doku.php?id=userman:registerkartenfenster>

Last update: **2023/02/08 12:01**



Smart Copy

Der Befehl **Smart Copy** ermöglicht es Ihnen Ihre Dokumente strukturiert und basierend auf den bestehenden Verknüpfungen innerhalb der Projekte zu [verschieben](#), [umzubenennen](#) und zu [kopieren](#). Bei diesen Operationen werden sämtliche Verweise der Dokumente beibehalten. In jeder Session können Sie neu entscheiden, welche Dokumente übernommen, ignoriert und kopiert werden sollen. Bei Smart Copy handelt es sich um ein kostenpflichtiges Primus-Modul, das zusätzlich lizenziert werden muss.

Arbeiten mit SmartCopy

Um **SmartCopy** nutzen zu können, muß zunächst eine Session gestartet werden.

1. Rechtsklick auf den Startordner und starten einer Session. Der Startordner soll so gewählt werden, dass er alle [mitzukopierenden](#) Unterordner enthält. Bei der Auswahl der zu kopierenden Dokumente ist es möglich, diese erst in einem tieferen Unterordner auszuwählen.
2. [Ordner zum Kopieren vormerken](#). Hier werden die Dokumente hinein kopiert / verschoben.
3. Dokumente zur Verarbeitung [wählen](#)
4. [Session abschließen](#)

Die Vorgehensweise ist für Kopier- und Verschiebesessions grundsätzlich gleich.

Beibehalten der Ordnerstruktur

Ob beim Kopieren oder Verschieben der Dokumente, diese flach in den Zielordner kopiert oder wieder in Unterordnern strukturiert abgelegt werden, wird vom Administrator beim Einrichten von Primus konfiguriert. Für welche Variante Ihr Primus konfiguriert ist, müssen Sie bei Ihrem Systembetreuer erfragen.

Ordner zum Kopieren vormerken



Damit die Daten nach dem Kopier- oder Verschiebevorgang gleich im richtigen Ordner abgelegt werden, wird vor dem Beginn der Session der Zielordner vorgemerkt.



Um einen Befehl oder Workflow zu starten, wird im Ordnerfenster der Ordner markiert, in dem die zu kopierenden Dateien liegen, und über die rechte Maustaste (RMT) das [Kontextmenü](#) aufgerufen. Innerhalb dieses Menüs kann der Zielordner unter dem [Smart Copy](#) Befehl der Zielordner vorgemerkt werden. Wenn Sie keinen Ordner vormerken, können Sie nach der Session die Dateien über das [Kontextmenü](#) in einem gewünschten Ordner ablegen.



Session starten



In jeder Session von SmartCopy können Sie entscheiden welche Dateien kopiert, verschoben, umbenannt oder ignoriert werden sollen. Wenn eine Session nicht durchgeführt werden soll, kann diese jederzeit über das [Kontextmenü](#) abgebrochen werden. Ihre Daten werden dabei nicht verändert. Es können 3 verschiedene Sessionarten gestartet werden.

- Session : [kopieren](#)
- Session : [verschieben](#)
- Session : [umbenennen](#)

Dokumente kopieren



Es gibt verschiedene Arten Dateien zu kopieren. Die Auswahl der zu kopierenden Dokumente werden innerhalb der [Dokumentenliste](#) angewählt und über das [Kontextmenü](#) in die Session eingebunden.

- Status : [kopieren](#)
- Status : [kopieren_mit_komponenten](#)
- Status : [kopieren_mit_komponenten_und_zeichnungen](#)

Nachdem die Dokumente zum Kopieren ausgewählt wurden, werden diese in der [Dokumentenliste](#) mit einem SC-Status „COPY“ angezeigt. Hier kann entschieden werden, ob Dateien ggf. beim Kopieren ignoriert werden sollen.



[Hilfethema](#)

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

Permanent link:
<https://dokuwiki.unitec.de/doku.php?id=userman:smartcopy>

Last update: **2023/02/08 12:01**



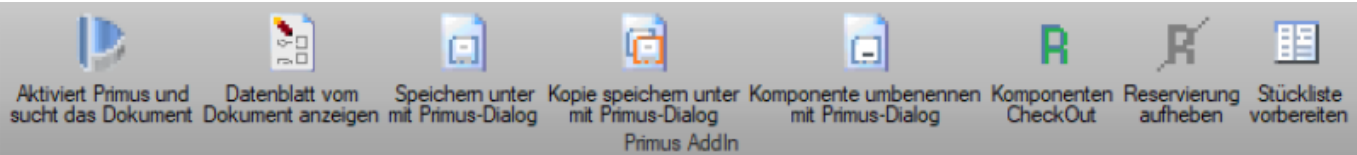
Stückliste erstellen

In diesem Artikel wird beschrieben welche Möglichkeiten es gibt um in Primus eine Stückliste zu erhalten. Durch die Auswertung der Dokumentenverknüpfungen aus dem CAD-System, stehen in Primus die Rohdaten zur Verfügung. Der Anwender hat dann aber die Möglichkeit den Aufbau der Stückliste detailliert zu steuern. Für die Vorbereitungen ist die Art der benötigten Stückliste unerheblich. Die Art der Stückliste kann der Anwender wählen, wenn später eine [Stückliste erzeugt](#) werden soll.

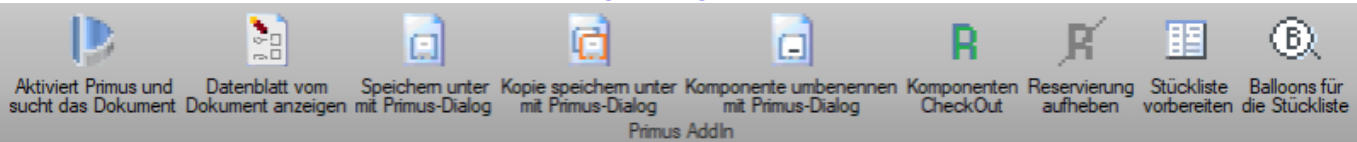
Stückliste vorbereiten

Zunächst muß jedoch für das betreffende Dokument über das [Primus Addin](#) die Stückliste vorbereitet werden. Dazu ist auf die Schaltfläche **Stückliste vorbereiten** zu klicken. Dadurch legt Primus erstmalig die Datenstruktur für die Stückliste zum Dokument an. Ist die Datenstruktur erstmal vorhanden wird sie bei zukünftigen Speichervorgängen automatisch aktualisiert.

Menüleiste des SEAddins in der Assemblyumgebung³¹⁾



Menüleiste des SEAddins in der Draftumgebung



Konfiguration der Stückliste

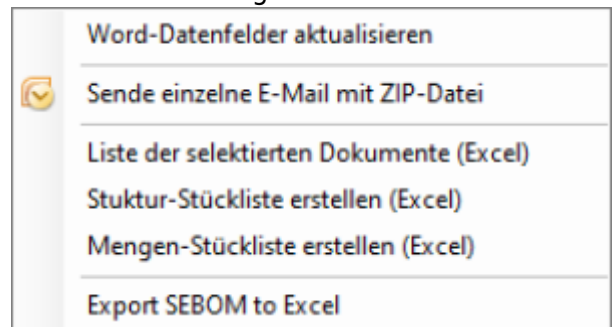
Nachdem die Stückliste vorbereitet wurde hat der Anwender die Möglichkeit den Aufbau der Stückliste zu konfigurieren. Dazu listet Primus zunächst die Verknüpften Dokumente und die zugehörige Menge auf. In der Spalte **Status** hat der Anwender nun die Wahl zwischen mehreren Möglichkeiten.

Stückliste				
Reload ASM		AutoPosNr.		ClearPosNr.
FileName	State	Quantity	Position	
seva01.par	Normal	1		
seva02.par	Ignorier...	1		
seva03.par	Aufgeteilt	2		
seva04.asm	Als Bauteil	2		
seva05.asm	Normal	1		
seva07.par	Normal	1		
seva08.par	Normal	1		
seva09.par	Normal	1		

Status	Bedeutung
Normal	Das Dokument wird als normales Element in der Stückliste aufgeführt
Ignorieren	Das Dokument wird für die Stückliste ignoriert
Aufgeteilt	Das Dokument selbst erscheint nicht in der Stückliste, aber alle darin enthaltenen Elemente. Nur sinnvoll bei Baugruppen
Als Bauteil	Das Dokument erscheint in der Stückliste, es werden aber keine weiteren darin enthaltenen Elemente aufgelistet. Sinnvoll bei Baugruppen in Strukturstücklisten

Stückliste erzeugen

Das Erzeugen der Stückliste findet direkt in Primus statt. Dazu wird mit der rechten Maustaste das **Kontextmenü** des Dokuments aufgerufen. Dort gibt es einen Menüpunkt **Office**. Hier hat der Anwender die Wahl zwischen verschiedenen Arten der Stückliste. Die erzeugten Stücklisten werden als Exceltabellen, im **Register Anhänge** abgelegt.



Hilfethema

³¹⁾

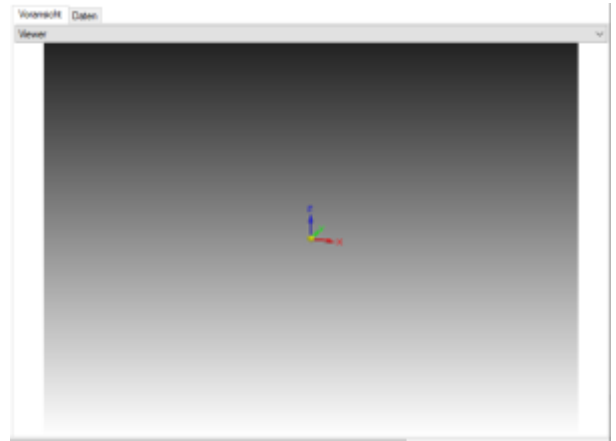
Die Benutzeroberfläche von Solid Edge ist vom Anwender anpassbar, daher kann die Darstellung von der hier gezeigten abweichen

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

Permanent link:
https://dokuwiki.unitec.de/doku.php?id=userman:stueckliste_erstellen

Last update: **2023/02/08 12:01**





Vorschaufenster

Im **Vorschaufenster** wird der Inhalt des in der [Dokumentenliste](#) gewählten Dokuments angezeigt. In der Auswahlliste am oberen Rand kann die Art der Anzeige ausgewählt werden. Es stehen [Explorer](#), [Thumbnail](#) und [Viewer](#) zur Verfügung. Alternativ kann statt der grafischen Vorschau auch das [Dokumentendatenblatt](#) angezeigt werden. Es wird dann das Datenblatt des im [Registerkartenfenster](#) markierten Elements angezeigt.

Explorer

Zeigt eine Vorschau an wie sie auch der Dateexplorer anzeigt.

[Klärungsbedarf](#)

Thumbnail

[Klärungsbedarf](#)

Viewer

Das Dokument wird mit dem für diesen Dateityp vorgesehenen Viewer angezeigt. Die Viewer Software wird normalerweise vom Hersteller einer Software die diese Daten erzeugt zur Verfügung gestellt.

PDF

Zum Anzeigen von PDF Dateien ist der Acrobat Reader von Adobe erforderlich.

Solid Edge ASM, DFT, PAR und PSM

Zur Anzeige von Solid Edge Daten muß der Solid Edge Viewer installiert sein, ein installiertes Solid Edge reicht hierzu nicht. Ist der Viewer nicht installiert, dann kann keine interaktive Vorschau angezeigt werden und es wird nur eine statische Vorschau angezeigt.

From:

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

Permanent link:

<https://dokuwiki.unitec.de/doku.php?id=userman:vorschaufenster>

Last update: **2023/02/08 12:01**

