

```
// Importiert die Datei (BLOB) im Feld F123 des DV1 mit Hilfe des Schnittstellen Applets
"MeinDatenschnittstellenApplet"
DV1.F123.ImportInto("MeinDatenschnittstellenApplet?Param1=2345");
```

Des Weiteren können Sie auf beiden Plattformen den `ImportData`-Parameter verwenden. Er gibt an, aus welcher Spalte ein zu importierender BLOB aus einer Datenbanktabelle geladen werden soll. Genau wie beim `Template`-Parameter gehört dazu die Angabe `ImportDataUK` mit dem Primärschlüssel des BLOB-Datensatzes:

[SL-Script]

```
// BLOB in Feld F1039 der Tabelle T1009 mit Schlüsselwert aus DV11.F1038 importieren:
open://MeinDatenschnittstellenApplet?ImportData=T1009.F1039&ImportDataUK=${DV11.F1038}
```

10.3 Dokumentvorlagen

Dokumentvorlagen werden in SCOPELAND benutzt, um die Struktur und das Erscheinungsbild von exportierten Dokumenten (→ 10.1) und um die Datenanordnung von zu importierenden Dokumenten (→ 10.2) zu beschreiben.

Bitte beachten Sie, dass mit dem Begriff **Dokumentvorlage** nicht die Vorlagentypen der Office-Anwendungen (u. a. *.dotx, *.ott, ...) gemeint sind, sondern dass es sich hier um normale Dokumente handelt, die mit SCOPELAND-Kennzeichnern und Platzhaltern versehen wurden.

Damit Sie bei der Gestaltung von Dokumentvorlagen alle Möglichkeiten haben, benutzen Sie dazu in der Regel die zum Dokumenttyp gehörigen marktüblichen Standard-Anwendungen. Für die gängigen Office-Dokumenttypen beispielsweise Microsoft Office, OpenOffice.org oder LibreOffice.

Allgemeine Struktur der Kennzeichner

Die Kennzeichner in Dokumentvorlagen sind selbst Bestandteile des normalen Dokumentinhalts und können genau wie dieser bearbeitet und formatiert werden. Damit sie von SCOPELAND gefunden werden, unterscheiden sie sich von allgemein gebräuchlichem Inhalt, indem sie in eckige Klammern eingeschlossen werden und ein Dollar-Zeichen (\$) vorangestellt haben.

Wenn Kennzeichner im Dokument nicht richtig ersetzt werden (aus z. B. `${DV1.F123}` wird nur `DV1.F123`), dann wurde der Kennzeichner nicht erkannt. Prüfen Sie, ob der Kennzeichner richtig geschrieben ist (auch Groß- und Kleinschreibung beachten), dass keine Leerzeichen enthalten sind (z. B. **nicht** `${DV1. F123}`) und dass der Kennzeichner möglichst nicht durch Formatierungen beschädigt wurde.

Es wird zwischen zwei Arten von Kennzeichnern unterschieden:

Einzelkennzeichner

Einzelkennzeichner (→ 10.3.1) bestehen aus nur einem Kennzeichner. Sie werden beim Export entweder durch einen vom SCOPELAND-Applet ermittelten Wert ersetzt oder aus der Dokumentvorlage entfernt – je nach Funktionalität. Wenn sie ersetzt werden, erhält der ersetzende Wert im Allgemeinen die gleiche Formatierung wie sein platzhaltender Kennzeichner. Ausnahmen hängen von der jeweiligen Kennzeichner-Aufgabe ab. Beim vorlagenbasierten Import beschreiben sie, welche Daten an dieser Stelle in der zu importierenden Datei zu erwarten sind.

Übersicht der Einzelkennzeichner

| Kennzeichner | Schreibweise (minimal) |
|--------------------------------------|---------------------------------------|
| Feldwert (→ 10.3.1) | <code>\${Feld-ID}</code> |
| SCOPELAND-Expression (→ 10.3.1) | <code>\${Scopeland-Expression}</code> |
| Datensatzanzahl (→ 10.3.1) | <code>\${RowCount}</code> |
| Datensatznummer (→ 10.3.1) | <code>\${RowNo}</code> |
| Datensatzindex (→ 10.3.1) | <code>\${RowIndex}</code> |
| Dokument-Applet einbetten (→ 10.3.1) | <code>\${ContentApplet}</code> |

Blockkennzeichner

Blockkennzeichner umfassen einen Dokumentausschnitt und bestehen aus einem Anfangskennzeichner und einem Endkennzeichner. Beim Export legen sie fest, was mit dem Block, den sie umfassen, passieren soll. In Anlehnung an übliche Auszeichnungssprachen haben Endkennzeichner den gleichen Namen wie ihre zugehörigen Anfangskennzeichner und werden durch einen einleitenden Schrägstrich (/) markiert.

Bitte beachten Sie, dass auch Leerzeichen, Tabstopps und Absatzmarken durch Blockkennzeichner eingerahmt und genau wie normaler Text behandelt werden.

In Spreadsheet-Dokumenten und Tabellen können Anfangs- und Endkennzeichner auch in unterschiedlichen Zellen stehen. Auf diese Weise ist es möglich rechteckige Ausschnitte aus einer Tabelle zu markieren. Der Anfangskennzeichner steht dabei in der linken oberen Ecke des zu markierenden Rechtecks und der Endkennzeichner in der unteren rechten Ecke. *Es werden allerdings immer komplette Zeilen markiert.*

Da Dokumente – insbesondere textbasierte Dokumente – unterschiedliche Strukturelemente (Text, Tabellen, Textfelder, Listen, usw.) enthalten können, dürfen Anfangs- und Endkennzeichner nicht an allen beliebigen Stellen eines Dokuments stehen, sondern sind voneinander abhängig. Es gilt immer, dass der Anfangskennzeichner vor dem Endkennzeichner auftauchen muss.

Allgemein sind nur folgende Konstellationen zulässig:

- Anfangs- und Endkennzeichner stehen beide im selben Absatz.
- Anfangs- und Endkennzeichner stehen in unterschiedlichen Absätzen, aber auf derselben textstrukturellen Ebene. Dazwischen dürfen andere Strukturelemente vorkommen (Tabellen, Textfelder, Listen usw.) solange sie vollständig von den Kennzeichnern eingeschlossen werden.
- Anfangs- und Endkennzeichner stehen zwar in unterschiedlichen Zellen, befinden sich aber in derselben Tabelle auf derselben textstrukturellen Ebene. Das heißt zum Beispiel, dass der eine Kennzeichner nicht tiefer in einer geschachtelten Tabelle stehen darf.
- In Spreadsheet-Dokumenten können nur Zellbereiche ausgezeichnet werden. Das markieren von einzelnen Werten oder Textelementen ist nicht möglich. Das bedeutet, dass sich ein Anfangskennzeichner immer auf die ganze Zelle bezieht, in der er steht, egal wo genau er sich in der Zelle befindet. Gleiches gilt für den Endkennzeichner.

Tipp: Sie können die DV-ID vor Kennzeichnern weglassen, wenn durch einen übergeordneten Kennzeichner klar ist, welcher DV gemeint ist. Zum Beispiel können Sie bei `$(DV1)$(DV1.Repeat) ... $[/DV1.Repeat]$[/DV1]` das DV1 vor Repeat weglassen und nur `$(DV1)$[Repeat] ... $[/Repeat]$[/DV1]` oder sogar gleich `$(DV1.Repeat) ... $[/DV1.Repeat]` schreiben.

Es gibt DirectView-bezogene (→ 10.3.2), Datensatz-bezogene (→ 10.3.3) und Feld-bezogene (→ 10.3.4) Blockkennzeichner.

Übersicht der Blockkennzeichner

| | Kennzeichner | Schreibweise (minimal) |
|---------------------|-------------------------------------|---|
| Bezug auf DV | DV-Kontext (→ 10.3.2) | <code>\$(DV-ID) ... \$[/DV-ID]</code> |
| | Aufzählung (→ 10.3.2) | <code>\$(Repeat) ... \$[/Repeat]</code> |
| | Leerer DV (→ 10.3.2) | <code>\$(Empty) ... \$[/Empty]</code> |
| | Nicht-leerer DV (→ 10.3.2) | <code>\$(NotEmpty) ... \$[/NotEmpty]</code> |
| | Einzeiliger DV (→ 10.3.2) | <code>\$(OneRow) ... \$[/OneRow]</code> |
| | Mehrzeiliger DV (→ 10.3.2) | <code>\$(ManyRows) ... \$[/ManyRows]</code> |
| Bezug auf Datensatz | Erster Datensatz (→ 10.3.3) | <code>\$(FirstRow) ... \$[/FirstRow]</code> |
| | Innerer Datensatz (→ 10.3.3) | <code>\$(MidRow) ... \$[/MidRow]</code> |
| | Letzter Datensatz (→ 10.3.3) | <code>\$(LastRow) ... \$[/LastRow]</code> |
| | Ungerader Datensatz (→ 10.3.3) | <code>\$(OddRow) ... \$[/OddRow]</code> |
| | Gerader Datensatz (→ 10.3.3) | <code>\$(EvenRow) ... \$[/EvenRow]</code> |
| | Datensatz hat Vorgänger (→ 10.3.3) | <code>\$(HasPrevRow) ... \$[/HasPrevRow]</code> |
| | Datensatz hat Nachfolger (→ 10.3.3) | <code>\$(HasNextRow) ... \$[/HasNextRow]</code> |

| | Kennzeichner | Schreibweise (minimal) |
|----------------|-------------------------------------|--|
| Bezug auf Feld | Leerer Feldwert (→ 10.3.4) | <code>[\$[Feld-ID.Empty] ... \$[/Feld-ID.Empty]</code> |
| | Nicht-leerer Feldwert (→ 10.3.4) | <code>[\$[Feld-ID.NotEmpty] ... \$[/Feld-ID.NotEmpty]</code> |
| | Feldwert ergibt "wahr" (→ 10.3.4) | <code>[\$[Feld-ID.IsTrue] ... \$[/Feld-ID.IsTrue]</code> |
| | Feldwert ergibt "falsch" (→ 10.3.4) | <code>[\$[Feld-ID.IsFalse] ... \$[/Feld-ID.IsFalse]</code> |

10.3.1 Einzel-Dokumentkennzeichner

Auf dieser Seite finden Sie eine detaillierte Beschreibung aller in SCOPELAND verfügbaren Dokumentkennzeichner zum Einfügen von Daten und Werten in Dokumentvorlagen (→ 10.3).

Feldwert

SYNTAX

- `[$[Feld-ID]`
- `[$[DV-ID.Feld-ID]`
- `[$[Feld-ID?Length=Anzahl]`
- `[$[DV-ID.Feld-ID?Length=Anzahl]`

FUNKTION BEIM EXPORT

Um den Wert eines bestimmten Datensatzfeldes ins Dokument exportieren (→ 10.1) zu lassen, werden Feldwert-Kennzeichner verwendet. Das sind Einzelkennzeichner, die einfach durch den aktuellen Wert des von ihnen spezifizierten Feldes ersetzt werden.

Ein Feldwert-Kennzeichner fügt den Feldwert mit der ID **Feld-ID** des aktuell aufgezählten Datensatzes im DirectView an der Stelle des Kennzeichners ein. Für den Feldwert wird die Formatierung des Kennzeichners übernommen und danach ggf. bedingte Formatierung des Feldwerts angewendet. Wenn kein Kontext-DirectView gegeben ist oder wenn ein anderer benutzt werden soll, kann der Feld-ID eine DirectView-ID vorangestellt werden. Mit dem Parameter **Length** ist es möglich, den ausgegebenen Wert auf eine bestimmte Zeichenanzahl zu beschränken oder – falls er kürzer ist – am Ende mit Leerzeichen aufzufüllen.

Nicht auflösbare Feldwert-Kennzeichner (z. B. ungültige Feld-ID oder fehlender DV) werden aus dem Dokument entfernt.

FUNKTION BEIM IMPORT

Beim vorlagenbasierten Import (→ 10.2) gibt der Feldwert-Kennzeichner an, dass an dieser Stelle im zu importierenden Dokument ein Wert für ein Datensatzfeld steht.

Hier gibt der optionale Parameter **Length** – analog zum Export – an, wie viele Zeichen für den Wert importiert werden sollen.

BEISPIELE

[Code]

```

$[F1000032]
$[DV1.F120034]
$[DV1.F120034?Length=20]
$[DV1.F3500029.T3500008.F3500041]

```

SCOPELAND-Expression

SYNTAX

- `[$[Scopeland-Expression]`

FUNKTION BEIM EXPORT

Wertet die **Scopeland-Expression** aus und fügt den errechneten Wert an der Stelle des Kennzeichners ein. Für das Ergebnis wird die Formatierung des Kennzeichners übernommen.

Achtung: Im Vergleich zum DirectDesk ist in generierten Dokumenten nur eine sehr eingeschränkte Auswahl an Expressions möglich.

FUNKTION BEIM IMPORT

(keine)

BEISPIELE

[Code]

```
Heute ist der ${Today}.
Sie sind ${UserName}.
Wert des Applet-Parameters X: ${?X}
```

Datensatzanzahl

SYNTAX

- `${RowCount}`
- `${ DV-ID .RowCount }`

FUNKTION BEIM EXPORT

Fügt die Anzahl (0 bis n) der Datensätze des Kontext-DVs an der Stelle des Kennzeichners ein. Für die numerische Anzahl wird die Formatierung des Kennzeichners übernommen.

FUNKTION BEIM IMPORT

(keine)

Datensatznummer

SYNTAX

- `${RowNo}`
- `${ DV-ID .RowNo }`

FUNKTION BEIM EXPORT

Fügt die Nummer (0 bis n-1) des aktuell aufgezählten Datensatzes im Kontext-DV an der Stelle des Kennzeichners ein. Für die Nummer wird die Formatierung des Kennzeichners übernommen.

FUNKTION BEIM IMPORT

(keine)

Datensatzindex

SYNTAX

- `${RowIndex}`
- `${ DV-ID .RowIndex }`

FUNKTION BEIM EXPORT

Fügt den Index (1 bis n) des aktuell aufgezählten Datensatzes im Kontext-DV an der Stelle des Kennzeichners ein. Für den Index wird die Formatierung des Kennzeichners übernommen. Im Gegensatz zu `${RowNo}` beginnt die Nummerierung hier bei 1 und nicht bei 0.

Report-Applet einbetten

SYNTAX

- `${ContentApplet}`

FUNKTION BEIM EXPORT

An der Stelle des Kennzeichners wird das exportierte Dokument eines anderen Applets eingebettet (siehe Zusammengesetzte Dokumente (→ 10.1). Der Name des Applets wird per Applet-Parameter (→ 11.14) (...?ContentApplet=...&...) übergeben und beide Applets haben Zugriff auf alle Parameter.

FUNKTION BEIM IMPORT

(keine)

10.3.2 DV-Dokumentkennzeichner

Auf dieser Seite finden Sie eine detaillierte Beschreibung der DirectView-bezogenen Dokumentkennzeichner in Dokumentvorlagen (→ 10.3).

DirectView-Blockkennzeichner beziehen sich immer auf die Eigenschaften eines DirectViews (DV). Die Kennzeichner *DV-Kontext* und *Aufzählung* nehmen dabei eine besondere Rolle ein – die anderen sind lediglich Schalter, die abhängig von ihrer geprüften Eigenschaft beim Export Dokumentinhalt entfernen oder stehen lassen:

| Anzahl Datensätze im DV | \$_[Empty] | \$_[NotEmpty] | \$_[OneRow] | \$_[ManyRows] |
|-------------------------|------------|---------------|-------------|---------------|
| 0 | + | – | – | – |
| 1 | – | + | + | – |
| > 1 | – | + | – | + |

Die Tabelle gibt an, bei welcher Datensatzanzahl der jeweilige Kennzeichner den Dokumentabschnitt erhält (+) oder aus dem Dokument entfernt (–).

DV-Kontext

SYNTAX

- `$_[DV-ID] ... $_[/ DV-ID]`

FUNKTION BEI EXPORT UND IMPORT

Um unterscheiden zu können, auf welchen DirectView in der DirectView-Hierarchie (→ 11.8) sich ein Kennzeichner bezieht, muss dieser von einem DirectView-Kennzeichner umgeben werden. Dessen Anfangskennzeichner besteht nur aus dem Namen des zu betrachtenden DVs und einem ebenso benannten Endkennzeichner. Natürlich lassen sich DirectView-Kennzeichner auch schachteln, um so die DV-Hierarchie abzubilden.

BEISPIELE

[Code]

```
$_[DV1]
Kennzeichner hier beziehen sich auf DV1
$_[DV11]
Und hier auf DV11
$_[ /DV11 ]
$_[ /DV1 ]
```

Aufzählung

SYNTAX

- `$_[Repeat] ... $_[/Repeat]`
- `$_[DV-ID .Repeat] ... $_[/ DV-ID .Repeat]`

FUNKTION BEIM EXPORT

Das Durchlaufen eines DirectViews wird mit dem Aufzählungskennzeichner gesteuert, welcher beim Dokumentexport (→ 10.1) den von sich eingeklammerten Dokumentabschnitt für jeden Datensatz des gerade aktiven DVs wiederholt. Die in diesem Dokumentabschnitt befindlichen Kennzeichner beziehen sich dann auf den Datensatz, für den sie kopiert werden.

Der Repeat-Kennzeichner muss sich in einem DV-Kontext befinden, um zu ermitteln, welcher DV durchlaufen werden soll. Enthält der zu durchlaufende DV keine Datensätze, so wird der zu wiederholende Dokumentausschnitt aus dem Dokument entfernt.

Spezialfall: In Tabellen, deren zu wiederholende Zeile nur eine Zelle enthält, ist nicht eindeutig spezifizierbar, ob mit dem Repeat-Tag die Zeile oder der Zelleninhalt eingeklammert werden soll. Um dieses Problem zu lösen gilt: Wenn sowohl Anfangs- als auch Endkennzeichner des Repeats vorhanden sind, dann wird der Zelleninhalt wiederholt. Wenn der Endkennzeichner weggelassen wird, dann wird die Zeile wiederholt. In der strikten Syntaxvariante ist dies der einzige Fall, wo ein Endkennzeichner weggelassen werden darf.

Technische Anmerkung: Das Repeat geht alle Zeilen im DV durch und ruft in jedem Schritt `SetCurrentRow(i)` und `Refresh(DirectView.RefreshFlags.RefreshData | DirectView.RefreshFlags.RecurChildren)` am DV auf, wobei `i` die Nummer (0 bis `n-1`) der jeweiligen Zeile ist. Auf diese Weise werden untergeordnete Kind-DVs ebenfalls refresht, sofern eine echte Master-Detail-Beziehung zu einem übergeordneten DV besteht. Ist dies nicht der Fall, muss im Applet für das `AfterRefresh`-Event eines Masters eine eigene Refresh-Logik eingebaut werden.

FUNKTION BEIM IMPORT

Beim vorlagenbasierten Dokumentimport (→ 10.2) arbeitet der Repeat-Kennzeichner umgekehrt: Er gibt an, dass an dieser Stelle im zu importierenden Dokument nun die Datensätze seines DVs folgen. In seinem Rumpf befinden sich dann Feldwert-Kennzeichner (→ 10.3.1), die die Anordnung der Felder innerhalb des Importdatensatzes beschreiben.

Achten Sie darauf, dass falls Ihre zu importierenden Dokumente mehrere Repeat-Bereiche enthalten, diese eindeutig voneinander abgegrenzt sind. Beispielsweise durch Leerzeilen in Spreadsheets oder statischem Text.

BEISPIELE

[Code]

```

$[DV1]
$[Repeat]Land: $[F123] in $[F234]
$[/Repeat]
$[/DV1]

$[DV1.Repeat]Land: $[F123] in $[F234]
$[/DV1.Repeat]

```

Leerer DV

SYNTAX

- `[$[Empty] ... $[/Empty]`
- `[$[DV-ID .Empty] ... $[/ DV-ID .Empty]`

FUNKTION BEIM EXPORT

Der von diesem Kennzeichner markierte Dokumentabschnitt wird nur ausgegeben, wenn der Kontext-DV keine Datensätze enthält und somit leer ist.

FUNKTION BEIM IMPORT

(keine)

BEISPIELE

[Code]

```

$[DV1]
$[Repeat]
Land: $[F123]
$[DV11]${[Empty]Keine Städte$[/Empty]}$[Repeat]${[F234]} $[/Repeat]${[DV11]}
$[/Repeat]
$[/DV1]

```

Nicht-leerer DV

SYNTAX

- `[$[NotEmpty] ... $[/NotEmpty]`
- `[$[DV-ID .NotEmpty] ... $[/ DV-ID .NotEmpty]`

FUNKTION BEIM EXPORT

Der von diesem Kennzeichner markierte Dokumentabschnitt wird nur ausgegeben, wenn der Kontext-DV mindestens einen Datensatz enthält und somit nicht leer ist.

FUNKTION BEIM IMPORT

(keine)

Einzeiliger DV

SYNTAX

- `#[OneRow] ... #[/OneRow]`
- `#[DV-ID .OneRow] ... #[/ DV-ID .OneRow]`

FUNKTION BEIM EXPORT

Der von diesem Kennzeichner markierte Dokumentabschnitt wird nur ausgegeben, wenn der Kontext-DV genau einen Datensatz enthält.

FUNKTION BEIM IMPORT

(keine)

Mehrzeiliger DV

SYNTAX

- `#[ManyRows] ... #[/ManyRows]`
- `#[DV-ID .ManyRows] ... #[/ DV-ID .ManyRows]`

FUNKTION BEIM EXPORT

Der von diesem Kennzeichner markierte Dokumentabschnitt wird nur ausgegeben, wenn der Kontext-DV mehr als einen Datensatz enthält.

FUNKTION BEIM IMPORT

(keine)

10.3.3 Datensatz-Dokumentkennzeichner

Auf dieser Seite finden Sie eine detaillierte Beschreibung der Datensatz-bezogenen Dokumentkennzeichner in Dokumentvorlagen (→ 10.3).

Datensatz-Blockkennzeichner beziehen sich immer auf die Eigenschaft eines bestimmten Datensatzes (engl. "Row"). So kann beispielsweise ein Datensatz abhängig von seiner Position im DirectView beim Export anders im Dokument dargestellt werden (Erster/letzter anders, ungerade/gerade Zeile anders, usw.).

Die folgenden Wahrheitstabellen geben an, wann welcher Kennzeichner zu logisch WAHR bzw. FALSCH ausgewertet wird und dann Dokumentinhalt erhält (+) oder entfernt (-). Dies hängt nämlich von der Position des betroffenen Datensatzes und der Anzahl Datensätze im DirectView ab. Danach folgen genauere Beschreibungen der möglichen Kennzeichner.

Wahrheitstabelle für einen DirectView mit genau einem Datensatz (Spezialfall)

| Datensatznummer | <code>#[FirstRow]</code> | <code>#[MidRow]</code> | <code>#[LastRow]</code> | <code>#[OddRow]</code> | <code>#[EvenRow]</code> | <code>#[HasPrevRow]</code> | <code>#[HasNextRow]</code> |
|-----------------|--------------------------|------------------------|-------------------------|------------------------|-------------------------|----------------------------|----------------------------|
| 1 | + | - | + | + | - | - | - |

Wahrheitstabelle für einen DirectView mit genau zwei Datensätzen (Spezialfall)

| Datensatz Nummer | `\${FirstRow}` | `\${MidRow}` | `\${LastRow}` | `\${OddRow}` | `\${EvenRow}` | `\${HasPrevRow}` | `\${HasNextRow}` |
|------------------|----------------|--------------|---------------|--------------|---------------|------------------|------------------|
| 1 | + | - | - | + | - | - | + |
| 2 | - | - | + | - | + | + | - |

Wahrheitstabelle für einen DirectView mit mehr als zwei Datensätzen (allgemeiner Fall)

| Datensatz Nummer | `\${FirstRow}` | `\${MidRow}` | `\${LastRow}` | `\${OddRow}` | `\${EvenRow}` | `\${HasPrevRow}` | `\${HasNextRow}` |
|------------------|----------------|--------------|---------------|--------------|---------------|------------------|------------------|
| 1 | + | - | - | + | - | - | + |
| 2 | - | + | - | - | + | + | + |
| 3 | - | + | - | + | - | + | + |
| 4 | - | - | + | - | + | + | - |

Erster Datensatz

SYNTAX

- `\${FirstRow}` ... `\${/FirstRow}`
- `\${ DV-ID .FirstRow}` ... `\${/ DV-ID .FirstRow}`

FUNKTION

Der von diesem Kennzeichner markierte Dokumentabschnitt wird nur ausgegeben, wenn der aktuell ausgewertete Datensatz der erste Datensatz im Kontext-DV ist.

Innerer Datensatz

SYNTAX

- `\${MidRow}` ... `\${/MidRow}`
- `\${ DV-ID .MidRow}` ... `\${/ DV-ID .MidRow}`

FUNKTION

Der von diesem Kennzeichner markierte Dokumentabschnitt wird nur ausgegeben, wenn der aktuell ausgewertete Datensatz nicht erster und nicht letzter Datensatz im Kontext-DV ist.

Letzter Datensatz

SYNTAX

- `\${LastRow}` ... `\${/LastRow}`
- `\${ DV-ID .LastRow}` ... `\${/ DV-ID .LastRow}`

FUNKTION

Der von diesem Kennzeichner markierte Dokumentabschnitt wird nur ausgegeben, wenn der aktuell ausgewertete Datensatz der letzte Datensatz im Kontext-DV ist.

Ungerader Datensatz

SYNTAX

- `\${OddRow}` ... `\${/OddRow}`
- `\${ DV-ID .OddRow}` ... `\${/ DV-ID .OddRow}`

FUNKTION

Der von diesem Kennzeichner markierte Dokumentabschnitt wird nur ausgegeben, wenn der aktuell ausgewertete Datensatz eine ungerade Nummer im Kontext-DV hat (es gilt also $\$[\text{RowIndex}] \bmod 2 = 1$).

Gerader Datensatz

SYNTAX

- `#[EvenRow] ... #[/EvenRow]`
- `#[DV-ID .EvenRow] ... #[/ DV-ID .EvenRow]`

FUNKTION

Der von diesem Kennzeichner markierte Dokumentabschnitt wird nur ausgegeben, wenn der aktuell ausgewertete Datensatz eine gerade Nummer im Kontext-DV hat (es gilt also $\$[\text{RowIndex}] \bmod 2 = 0$).

Datensatz hat Vorgänger

SYNTAX

- `#[HasPrevRow] ... #[/HasPrevRow]`
- `#[DV-ID .HasPrevRow] ... #[/ DV-ID .HasPrevRow]`

FUNKTION

Der von diesem Kennzeichner markierte Dokumentabschnitt wird nur ausgegeben, wenn vor dem aktuell ausgewerteten Datensatz bereits ein vorheriger Datensatz ausgewertet wurde.

Datensatz hat Nachfolger

SYNTAX

- `#[HasNextRow] ... #[/HasNextRow]`
- `#[DV-ID .HasNextRow] ... #[/ DV-ID .HasNextRow]`

FUNKTION

Der von diesem Kennzeichner markierte Dokumentabschnitt wird nur ausgegeben, wenn auf den aktuell ausgewerteten Datensatz ein weiterer Datensatz folgt.

BEISPIELE

Tipp: Wie das folgende Beispiel zeigt, eignet sich dieser Kennzeichner besonders gut, um kommagetrennte Aufzählungen umzusetzen.

[Code]

```
Eine Liste mit Früchten: #[DV1.Repeat]#[F123]#[HasNextRow], #[/HasNextRow]#[/DV1.Repeat]
```

10.3.4 Feld-Dokumentkennzeichner

Auf dieser Seite finden Sie eine detaillierte Beschreibung der Feld-bezogenen Dokumentkennzeichner in Dokumentvorlagen (→ 10.3).

Feldwert-Blockkennzeichner beziehen sich immer auf die Eigenschaften eines einzelnen Feldwerts. Abhängig von ihrer geprüften Eigenschaft entfernen sie Dokumentinhalt (–) oder lassen ihn stehen (+).

Die folgende Tabelle gibt an, bei welchen Feldwerten dies für den jeweiligen Kennzeichner der Fall ist:

| Feldart | Feldwert | <code>#[IsTrue]</code> | <code>#[IsFalse]</code> | <code>#[Empty]</code> | <code>#[NotEmpty]</code> |
|---------|----------|------------------------|-------------------------|-----------------------|--------------------------|
| Logisch | WAHR | + | – | – | + |
| Logisch | FALSCH | – | + | – | + |
| Zahl | 0 | – | + | – | + |
| Zahl | Sonstige | + | – | – | + |

| Feldart | Feldwert | \$_[IsTrue] | \$_[IsFalse] | \$_[Empty] | \$_[NotEmpty] |
|-----------------|-----------------|-------------|--------------|------------|---------------|
| Text | "false" | - | + | - | + |
| Text | "0" | - | + | - | + |
| Text | "" | + | - | + | - |
| Text | <i>Sonstige</i> | + | - | - | + |
| Datum | <i>Beliebig</i> | - | + | - | + |
| Zeit | <i>Beliebig</i> | - | + | - | + |
| Datum+Zeit | <i>Beliebig</i> | - | + | - | + |
| Blob | <i>Beliebig</i> | - | + | - | + |
| <i>Beliebig</i> | NULL | - | - | + | - |

Leerer Feldwert

SYNTAX

- `$_[Feld-ID .Empty] ... $[/Feld-ID .Empty]`
- `$_[DV-ID . Feld-ID .Empty] ... $[/DV-ID . Feld-ID .Empty]`

FUNKTION

Der von diesem Kennzeichner markierte Dokumentabschnitt wird nur ausgegeben, wenn der Feldwert mit der ID **Feld-ID** des aktuell aufgezählten Datensatzes des Kontext-DVs leer ist.

Nicht-leerer Feldwert

SYNTAX

- `$_[Feld-ID .NotEmpty] ... $[/Feld-ID .NotEmpty]`
- `$_[DV-ID . Feld-ID .NotEmpty] ... $[/DV-ID . Feld-ID .NotEmpty]`

FUNKTION

Der von diesem Kennzeichner markierte Dokumentabschnitt wird nur ausgegeben, wenn der Feldwert mit der ID **Feld-ID** des aktuell aufgezählten Datensatzes des Kontext-DVs nicht leer ist.

Feldwert ergibt "wahr"

SYNTAX

- `$_[Feld-ID .IsTrue] ... $[/Feld-ID .IsTrue]`
- `$_[DV-ID . Feld-ID .IsTrue] ... $[/DV-ID . Feld-ID .IsTrue]`

FUNKTION

Der von diesem Kennzeichner markierte Dokumentabschnitt wird nur ausgegeben, wenn der Feldwert mit der ID **Feld-ID** des aktuell aufgezählten Datensatzes des Kontext-DVs nach wahr ausgewertet werden kann. Die Auswertung erfolgt dabei gemäß der oben angegebenen Tabelle.

Feldwert ergibt "falsch"

SYNTAX

- `$_[Feld-ID .IsFalse] ... $[/Feld-ID .IsFalse]`
- `$_[DV-ID . Feld-ID .IsFalse] ... $[/DV-ID . Feld-ID .IsFalse]`

FUNKTION

Der von diesem Kennzeichner markierte Dokumentabschnitt wird nur ausgegeben, wenn der Feldwert mit der ID **Feld-ID** des aktuell aufgezählten Datensatzes des Kontext-DVs nach falsch ausgewertet werden kann. Die Auswertung erfolgt dabei gemäß der oben angegebenen Tabelle.