

## Allgemein

Durch das Section Handling wird festgelegt, welcher User bestimmte Teile (sog. Sections) innerhalb der Smartform sehen kann/darf. Diese Beschränkungen können entweder innerhalb eines ganzen Prozesses gelten oder bei einem gewissen Stand innerhalb des Prozessverlaufs. Die folgenden Rechte können vergeben, bzw. entzogen werden:

- Schreibrecht
- Leserecht

## Vorbereitung der Smartform

In der Smartform wird der Bereich, der durch das Section Handling beeinflusst werden soll, beispielsweise mit einem **DIV** umschlossen und markiert somit einen abgeschlossen Teil. Diesem **DIV** muss zwingend das Attribut class=„section“ mitgegeben werden. Die ID des **DIVs** kann dabei beliebig gewählt werden.

In dem folgenden Beispiel sind zwei DIV-Bereiche zu sehen:

- DIV mit der ID=„section1“, beinhaltet:
  - Label „Section1“
  - Textfeld
- DIV mit der ID=„section2“, beinhaltet:
  - Label „Section2“
  - Textfeld

Wobei der Inhalt innerhalb eines **DIVs** keine Rolle spielt und beliebig sein kann.

```
<div class="section" id="section1">
  <label for="section1Input">Section1</label>
  <input id="section1Input" name="section1Input" type="text"></input>
</div>
<div class="section" id="section2">
  <label for="section2Input">Section2</label>
  <input id="section2Input" name="section2Input" type="text"></input>
</div>
```

Befindet sich die Section innerhalb einer Tabelle (<table>), so muss ein tbody anstatt eines DIVs benutzt werden, um die Section abzugrenzen.

## Section Handling Parameter

Die Parameter für das Section Handling, mit welchen die Rechte zugewiesen werden, werden in [Signavio](#) für den Prozess definiert :



## Section Handling Parameter Allgemein

```
<section-node-mapping>
  <node-mapping name="Name_einer_Node_aus_dem_Prozess">
    <section name="ID_der_Section_aus_der_Smartform">
      <read assignment="*" />
      <write assignment="*" />
    </section>
  </node-mapping>
</section-node-mapping>
```

- **<section-node-mapping>** : das section-node-mapping Element ist das Hauptstück des Section Handlings, innerhalb welchem sich die Regeln für die Beschränkungen befinden
- **<node-mapping>** : hier kann optional das Attribut name mitgegeben werden. Damit ist es möglich die Beschränkungen so zu konfigurieren, dass diese nur an einem bestimmten Stand innerhalb des Prozesses gelten, z.B. User A darf das Textfeld sehen, wenn die Smartform während der ersten Node geöffnet wird, aber nicht mehr sobald der Prozess sich auf Node B befindet.  
Wird das Attribut name an dieser Stelle nicht mitgegeben erstrecken sich die Beschränkungen auf den gesamten Prozessverlauf. Zudem können innerhalb des **<section-node-mapping>** Elements mehrere **<node-mapping>** s angegeben werden.
- **<section>** diesem Element muss das Attribut name mitgegeben werden, welches die ID der gewünschten Section aus der Smartform angibt. Innerhalb des Section Elements können nun die Rechte verteilt werden, die für die angegebene Section gelten sollen.
- **<read>** über das read Element wird nun bestimmt, wer den Inhalt innerhalb einer Section sehen darf. Die Personen werden über das Assignment Attribut bestimmt.
- **<write>** über das write Element wird nun bestimmt, wer den Inhalt innerhalb einer Section bearbeiten darf (z.B. in ein Textfeld schreiben). Die Personen werden über das Assignment Attribut bestimmt.

## Assignment

Für die Assignments innerhalb der read und write Elemente stehen nun folgende Möglichkeiten zur Auswahl:

Zielgruppe	Schreibweise
einzelne Benutzer	„user(Benutzername)“

Zielgruppe	Schreibweise
Swimlanes	„swimlane(Swimlanename)“
Gruppe	„group(Gruppenname)“
Wildcard (für keinen Benutzer)	„“
Wildcard (keine Beschränkung)	„*“
Negation(dieser User darf nicht...)	„! user(...)„

Mehrere Angaben können getrennt durch Semikolons realisiert werden, z. B.:  
 „group(Vertrieb);group(Controlling);user(Max.Mustermann)“

## Locking

Locking kann zusätzlich zu den Assignments angegeben werden und dient als Bearbeitungssperre. Um das Locking zu implementieren muss der Parameter lockable=„true“ angegeben werden. Locking kann auf das node-mapping oder die section selbst gelegt werden. Das locking Attribut der section überwiegt immer das Attribut des node-mappings.

```
<section-node-mapping>
  <node-mapping lockable="true">
    <section name="section1">
      <read assignment="*" />
      <write assignment="*" />
    </section>
    <section name="section2" lockable="false">
      <read assignment="*" />
      <write assignment="*" />
    </section>
  </node-mapping>
</section-node-mapping>
```

Wird nun die Smartform geöffnet, ist der Bereich standardmäßig gesperrt, jedoch befindet sich hinter den gelockten Elementen ein Stiftsymbol :



Über einen Klick auf das Symbol kann das gelockte/gesperrte Feld entsperrt werden und somit auch bearbeitet werden. Das Feld ist nach dem Klick für alle anderen User gesperrt, d.h. öffnet ein anderer Benutzer die Smartform ist es für ihn nicht möglich das für ihn gesperrte Feld ebenfalls zu entsperren und zu bearbeiten während das Feld von dem ersten User entsperrt ist. Damit wird verhindert dass eingegebene Daten versehentlich von einem anderen Nutzer überschrieben werden. Das Feld kann erst dann wieder von anderen Nutzern entsperrt werden, wenn der Benutzer der das derzeitige Bearbeitungsrecht besitzt die Smartform speichert und schließt.

## Stiftsymbole ausblenden

Ist lockable auf false gesetzt, wird vor der einer Section ein Stiftsymbol angezeigt, solange sich der Prozess nicht an der Node befindet die in dem zugehörigen NodeMapping angegeben wurde. Befindet

sich der Prozess dann an der Stelle, die angegeben wurde, so wird das Stiftsymbol ausgeblendet. Sollen nun alle Stiftsymbole ausgeblendet werden, so muss ein globales NodeMapping angegeben werden, in welchem alle Sections einmal definiert werden und ein Assignment bekommen. Soll sich an der Funktionsweise der Smartform nichts ändern, so reicht es in dem globalen NodeMapping jeder Section eine Wildcard zu geben.

## NodeMapping bereits vor Instanzstart

Sollen bereits vor Start der Instanz Bereiche eingegrenzt werden, so muss ein NodeMapping eingefügt werden, welches den Namen des Starts trägt. Ist kein Name für den Start angegeben, so bekommt der Start standardmäßig den Namen „**StartEvent\_1**“ und das Ende den Namen „**EndEvent\_1**“, dieser kann dann im NodeMapping benutzt werden

---

## Section Handling Parameter Beispiele anhand des Smartform Beispiels

Das folgende Bild zeigt den Ausgangszustand der Smartform ohne Section Handling (alles ist sichtbar und bearbeitbar):



Der obere Teil befindet sich in der ersten Section und der untere Teil befindet sich in der zweiten Section. Bei den Beispielen werden alle auf einen User als Beschränkter ausgelegt, es ist jedoch genauso möglich alle folgenden Beispiele mit den oben genannten Assignments durchzuführen. Es können nun die folgenden Szenarien mit Hilfe des Section Handlings erzeugt werden:

---

### Section nur für bestimmten User sichtbar und bearbeitbar

Nun soll der obere Teil nur für einen bestimmten User sichtbar und bearbeitbar sein. Die dafür benutzten Section Handling Parameter könnten wie folgt aussehen:

```
<section-node-mapping>
  <node-mapping>
    <section name="section1">
      <read assignment="user(TIM)"/>
      <write assignment="user(TIM)"/>
    </section>
    <section name="section2">
      <read assignment="*"/>
      <write assignment="*"/>
    </section>
  </node-mapping>
```

```
</section-node-mapping>
```

Die Smartform würde für den User TIM nach wie vor aussehen wie der Ausgangszustand für alle anderen jedoch würde sich das folgende Bild ergeben:



Wie man sieht ist nur der untere Teil der Smartform sichtbar.

## Section nur für bestimmten User bearbeitbar

Nun soll der obere Teil zwar sichtbar sein, jedoch nur für den User TIM bearbeitbar sein. Die dafür benutzten Section Handling Parameter könnten wie folgt aussehen:

```
<section-node-mapping>
  <node-mapping>
    <section name="section1">
      <read assignment="*" />
      <write assignment="user(TIM)" />
    </section>
    <section name="section2">
      <read assignment="*" />
      <write assignment="*" />
    </section>
  </node-mapping>
</section-node-mapping>
```

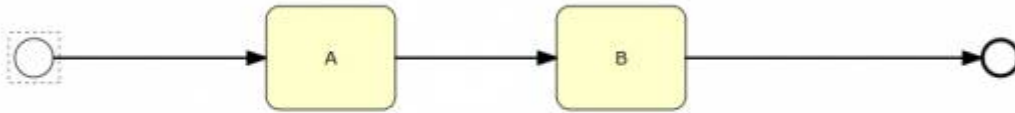
Daraus ergibt sich wiederum das folgende Bild für alle anderen User, die die Smartform öffnen wollen:



Die obere Section ist nun zwar für alle sichtbar, jedoch kann in das Textfeld nichts eingegeben werden, da der read Parameter nicht gesetzt ist.

## Section Beschränkung auf begrenzten Bereich innerhalb des Prozesses

Die Beschränkungen in den bisherigen Beispielen waren global und gelten für den gesamten Prozess. Nun kann man die Beschränkungen aber auch nur an bestimmten Stellen in einem Prozess geltend machen. Nun soll das erste Beispiel, in dem nur die zweite Section sichtbar ist so erweitert werden, dass die Beschränkung nur dann gilt, wenn sich der Prozessfortschritt auf einer bestimmten Node befindet. Dies soll anhand des folgenden Prozesses verdeutlicht werden:



Das gewünschte Ziel ist, dass solange sich der Prozess auf der Node A befindet nur der User TIM den oberen Teil der Smartform sehen darf. Sobald sich der Prozess auf Node B befindet darf zwar jeder den oberen Teil sehen, aber nicht bearbeiten. Hierfür müssen für jede Node die section Parameter angegeben werden. Die Parameter um dieses Ziel zu erreichen sind die folgenden:

```
<section-node-mapping>
  <node-mapping name="A">
    <section name="section1">
      <read assignment="user(TIM)"/>
      <write assignment="user(TIM)"/>
    </section>
    <section name="section2">
      <read assignment="*"/>
      <write assignment="*"/>
    </section>
  </node-mapping>
  <node-mapping name="B">
    <section name="section1">
      <read assignment="*"/>
      <write assignment="user(TIM)"/>
    </section>
    <section name="section2">
      <read assignment="*"/>
      <write assignment="*"/>
    </section>
  </node-mapping>
</section-node-mapping>
```

Während des ganzen Prozesses kann der User TIM die gesamte Smartform einsehen und bearbeiten da für ihn keine Beschränkungen angegeben sind. Befindet sich der Prozess auf der Node A ergibt sich für alle anderen das folgende Bild wenn die Smartform geöffnet wird: Nur der untere Teil ist sichtbar.

Section2

Wird die Aufgabe nun erledigt und der Prozess schreitet auf Node b voran, kann der User TIM weiterhin alles sehen. Für alle anderen sieht die Smartform so aus: Der obere Teil ist nicht bearbeitbar, aber sichtbar.

Section1   
Section2

## Wie verhält sich das Sectionhandling in der Smartform Suche?

From:

<https://wiki.tim-solutions.de/> - **TIM Wiki** / [NEW TIM 6 Documentation](#)

Permanent link:

[https://wiki.tim-solutions.de/doku.php?id=software:tim:section\\_handling&rev=1375170181](https://wiki.tim-solutions.de/doku.php?id=software:tim:section_handling&rev=1375170181)

Last update: **2021/07/01 09:58**

