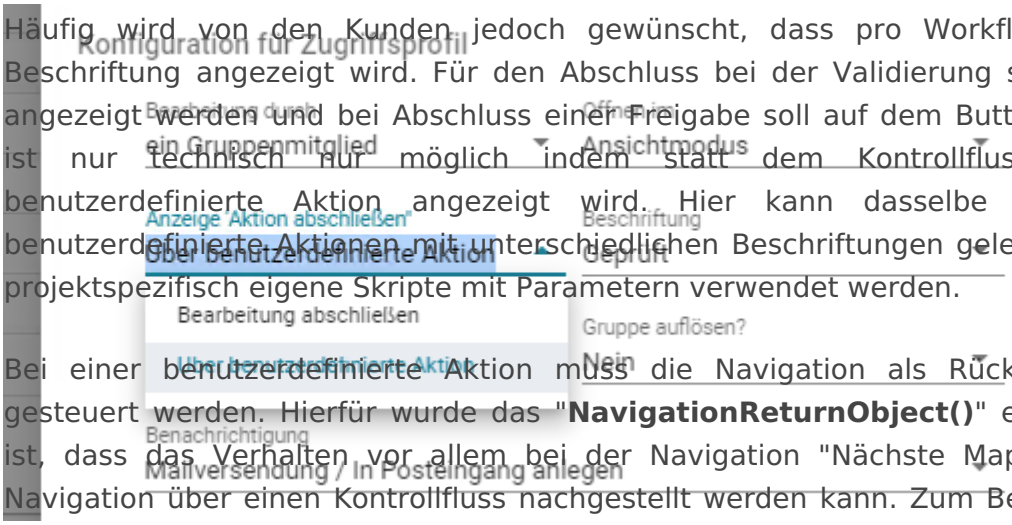
Versteckte User-Exits: Navigation nach Abschluss der Aktion

Bei Abschluss der aktuellen Workflow-Aktion über einen ausgehenden Kontrollfluss gesteuert werden. Der Nachteil dieser Umsetzung ist, dass der Kontrollfluss nur eine Beschriftung möglich ist. Da sich alle Workflow-Aktionen technisch im Hintergrund dieselbe Aktion teilen kann nur eine Beschriftung für den Button (pf:**WfButton_ActionEnd**) gesetzt werden und diese Anzeige gilt dann für alle Workflow-Aktionen wo "Bearbeitung abschließen" ausgewählt wurde. Dafür greift die ausgewählte Navigation ("Mappe beibehalten", "Nächste Mappe", "Zum Eingangsordner") exakt so funktioniert wie man es vom Documents-Workflow kennt.



Häufig wird von den Kunden jedoch gewünscht, dass pro Workflow-Aktion eine spezifische Beschriftung angezeigt wird. Für den Abschluss bei der Validierung soll zum Beispiel "Validiert" angezeigt werden und bei Abschluss einer Freigabe soll auf dem Button "Freigeben" stehen. Das ist nur technisch nur möglich indem statt dem Kontrollfluss-Button ein Button als benutzerdefinierte Aktion angezeigt wird. Hier kann dasselbe Skript auf beliebig viele benutzerdefinierte Aktionen mit unterschiedlichen Beschriftungen gelegt werden. Es können auch projektspezifisch eigene Skripte mit Parametern verwendet werden.

Bei einer benutzerdefinierte Aktion muss die Navigation als Rückgabewert über das Skript gesteuert werden. Hierfür wurde das "**NavigationReturnObject()**" erstellt. Ein großer Nachteil ist, dass das Verhalten vor allem bei der Navigation "Nächste Mappe" nicht 1:1 wie bei der Navigation über einen Kontrollfluss nachgestellt werden kann. Zum Beispiel kann bei selektierten Belegen in einem Ordner nicht automatisch einer der nächsten selektierten Belege angezeigt werden, da "context.selectedFiles" auch bei selektierten Belegen ein leeres FileResultset zurückliefert. Es kann auch nicht irgendein Beleg aus dem aktuellen Ordner angezeigt werden, da "context.folder", "context.folderName" und "context.folderFiles" keine Werte liefern.

Für die Anzeige des nächsten Belegs wird der Aufgaben-Ordner des aktuellen Benutzers ermittelt und es wird aus diesem Ordner ein möglichst passender Beleg ermittelt. Diese Ermittlung kann unter Umständen lange dauern, wenn der Anwender sehr viele Belege in seinen Aufgaben hat. Projektspezifisch könnte der neu anzuzeigende Beleg zum Beispiel über einen spezifischen Filterordner ermittelt werden und der Filterordner kann anhand einiger Daten wie die Aktion-ID und ggf. weiteren Informationen wie dem Mandanten abgeleitet werden.

Für solche Umsetzungen wird das NavigationReturnObject() ab der Invoice Version 1.1. um optionale UserExit-Funktion pro Navigation erweitert. Sobald eine der Funktionen definiert ist und

diese Funktion den Wert true zurückliefert ersetzt diese das Navigations-Standardverhalten! Folgende UserExit-Funktionen wurden ergänzt:

- **ue_Keep()**
- **ue_Next()**
- **ue_Overview()**
- **ue_Folder()**
- **ue_Inbox()**

Alle Funktionen haben denselben Aufbau - daher wird nur eine der Funktionen ein Beispiel-Skript angezeigt.

> Wenn eine UserExit-Funktion verwendet wird, dann wird der Standard-Code nicht ausgeführt! Die Rückgabe-Werte müssen in jedem Fall durch die UserExit-Funktion gesetzt werden!

```
/** Rückgabewert für die Navigation 'Mappe beibehalten'.
 * Diese Funktion zeigt ein Beispiel wie man nach der Validierung den nächsten Validierungs-Beleg anzeigen kann.
 * @returns {boolean} true (verwende UserExit Navigation) / false (verwerfe Rückgabewerte und ermittle nach Standard)
 **/
NavigationReturnObject.prototype.ue_Next(){
  if( this.ActionID==="ValidationInvoice" ){
    var folderIter = context.getFoldersByName("ValidationInvoice", "dynamicpublic");
      if( folderIter && folderIter.size()===1 ){
          var folder = folderIter.first();
          var files  = folder.getFiles();
            /* ... nächste Mappe ermitteln. */
          var showFile = ...
    this.ReturnType = "multipleAction";
this.ReturnVal  = JSON.stringify([
            { returnType : 'showFolder', returnValue : folder.id },
                { returnType : 'showFile', returnValue : showFile.getid() }
            ]);
            return true;
        }
    }
  return false;
}
```

Das NavigationReturnObject() hat Folgende Attribute:

Attribut	Beschreibung	Voreinstellung
ReturnType	Rückgabetyp (context.returnType)	"stay"
ReturnVal	Rückgabewert	
ForwardAction	Weiterleitungs-Art	"FinishAction"/"Forward"/"AnswerQuesti
Navigation	Navigation	"keepfile"/"next"/"overview"/"inbox"/"fol
DocFile	Aktuelle Mappe	context.file
DocFileID	Documents ID der aktuellen Mappe	context.file.getid()
Filetype	Mappentyp der aktuellen Mappe	context.file.getAttribute("FromTemplate
ActionID	Aktion ID der aktuellen Mappe	context.file.ActionID
WorkflowID	Workflow ID der aktuellen Mappe	context.file.WorkflowID
SystemUser	Aktueller Benutzer als SystemUser Objekt	context.getSystemUser()
MonoView	Angabe, ob beim aktuellen Benutzer der MonoView aktiv ist (ab Invoice 1.1.015).	""/"true"

Am Ende des Aufrufs gibt es ein zusätzliches User-Exit, über welches generell jegliche Navigation nochmals manipuliert werden kann. Das gilt auch für die oben genannten UserExits. Dieses UserExit bietet sich zum Beispiel an, wenn bestimmte Benutzer immer dieselbe Navigation haben möchten.

```
/** Manipulation des Objekts am Funktions-Ende.
 * @returns Kein Rückgabewert!
**/
NavigationReturnObject.prototype.ue_NavigationManipulation = function(){
	this.Log += "[INFO] function ue_NavigationManipulation()\n";
	if( this.SystemUser instanceof SystemUser ){
		this.Log += "[INFO] valid system user\n";
		var lastUsedFolder = this.SystemUser.getPrivateFolder("lastused");
		if( lastUsedFolder ){
			/* Dieses Beispiel öffnet den privaten Ordner 'Zuletzt benutzt' und zeigt die
aktuelle Mappe an. */
```

```
□□□this.ReturnType = "multipleAction";
□□□this.ReturnVal  = JSON.stringify( [
□□□□{ returnType : 'showFolder', returnValue :  lastUsedFolder.id },
□□□□{ returnType : 'showFile', returnValue :  this.DocFile.getid() }
            ] );
        }
□□else{
□□□this.Log += "[ERROR] invalid private folder(lastused)\n";
□□}
□}
□else{
□□this.Log += "[ERROR] invalid system user\n";
    }
□util.log(this.Log);
}
```

Revision #15
Created 8 October 2021 08:25:32 by Markus Meisner
Updated 5 August 2022 11:53:05 by Markus Meisner