

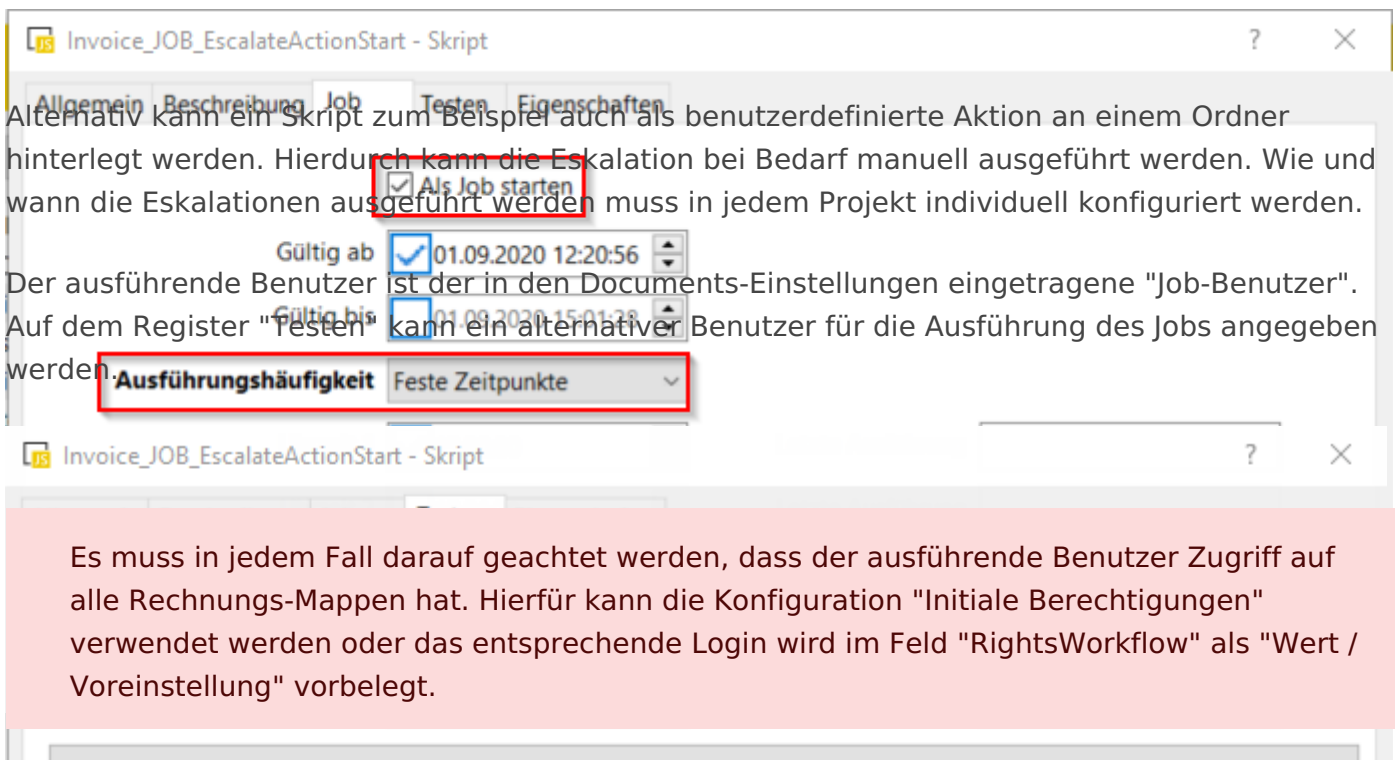
# Job-Skript

Für jede Eskalation kann ein separates Portal-Skript als Job eingeplant werden. Hierdurch lassen sich die Ausführungszeitpunkte beliebig definieren. Es können auch projektspezifische Jobs erstellt werden.

- [Portal-Skript als Job einrichten](#)
- [Aufbau Job-Skript](#)
- [Arbeits-Ordner](#)
- [Logging](#)

# Portal-Skript als Job einrichten

Dieser Abschnitt soll eine grobe Übersicht über den Aufbau und den Ablauf der Standard-Eskalation darstellen. Die Eskalations-Jobs werden in Documents als Portal-Skripte eingerichtet. Die in der Standard-Auslieferung enthaltenen Eskalations-Jobs führen jeweils nur einen Eskalations-Typen aus. Die Job-Einstellungen für das Portal-Skript werden auf dem Register "Job" konfiguriert. die Checkbox "Als Job starten" muss zwingend gesetzt sein. Für gewöhnlich soll eine Eskalationen einmal täglich zu festen Zeitpunkten ausgeführt werden.



Alternativ kann ein Skript zum Beispiel auch als benutzerdefinierte Aktion an einem Ordner hinterlegt werden. Hierdurch kann die Eskalation bei Bedarf manuell ausgeführt werden. Wie und wann die Eskalationen ausgeführt werden muss in jedem Projekt individuell konfiguriert werden.

Der ausführende Benutzer ist der in den Documents-Einstellungen eingetragene "Job-Benutzer". Auf dem Register "Testen" kann ein alternativer Benutzer für die Ausführung des Jobs angegeben werden.

Es muss in jedem Fall darauf geachtet werden, dass der ausführende Benutzer Zugriff auf alle Rechnungs-Mappen hat. Hierfür kann die Konfiguration "Initiale Berechtigungen" verwendet werden oder das entsprechende Login wird im Feld "RightsWorkflow" als "Wert / Voreinstellung" vorbelegt.

# Aufbau Job-Skript

Zu Beginn des Skripts werden alle Standard-Skript-Bibliotheken importiert. Hierdurch kann das Skript auf alle Standard-Objekte und User-Exits zugreifen. Im unteren Beispiel wird von einem Invoice-Modul ausgegangen.

Im Anschluss wird ein Eskalations-Objekt erzeugt. Dem Objekt werden der Mappentyp und der Eskalations-Typ als Parameter mitgegeben. Über das Objekt wird automatisch eine Log-Datei unter "...\\Documents6\\DEXPRO\\Escalation\\" erstellt.

Pro Modul und Eskalationstyp werden automatisch Unterordner erzeugt und für jeden Benutzer, an dem eine Eskalationsmail versendet werden soll, wird eine html-Datei angelegt. Damit die Anwender möglichst wenige Emails erhalten, werden die Eskalations-Informationen zunächst gesammelt. Via "FileResultSet()" werden alle Rechnungen auf eine Eskalation geprüft.

Wie erwähnt wird pro ermittelten Anwender eine HTML-Datei auf dem Server erstellt. In jeder Datei wird initial der Kopf-Text geschrieben - gefolgt von einer tabellarischen Übersicht der eskalierten Belege. Am Ende wird nur eine Mail pro Benutzer und Eskalations-Typ versendet. Die Dateien bleiben nach der Verarbeitung liegen, damit die Verarbeitung für den Support nachvollziehbar bleibt. Die alten Dateien werden zu Beginn eines neuen Job-Laufs gelöscht.

Erst wenn alle Vorgänge im FileResultSet geprüft wurden, werden die Eskalations-Mails generiert und versendet.

## Beispiel:

```
/* Job-Beispiel für Eskalationen */
// #import "Invoice__ImportLib";

var filetype = "Invoice";
var esc = new Escalation(filetype, "MeinEskalationsTyp");
    esc.cleanupFilePath(); // alte Dateien löschen

context.setSuperMode(true);
var frs = new FileResultSet(filetype, "", "");
for( var docFile=frs.first(); docFile; docFile=frs.next() ){
    /* Die escalate()-Funktion ruft die folgenden UserExits auf:
        * - Escalation.invoice_CheckEscalation()
        * - Escalation.invoice_GetEscalationUserArray()
        * - Escalation.invoice_CreateTableHead()
```

```
    * - Escalation.invoice_CreateTableEntry()
    */
    esc.escalate( docFile ); // Eskalation prüfen und Email-Html generieren
}

/* Beim Abschluss der Funktion wird das UserExit Escalation.invoice_CloseTable() aufgerufen.
 * Im Anschluss werden die Mailsversendet.
 */
esc.closeFilesAndSendMails(); // HTML abschließen und Mails versenden
esc.Log.stop(); // Logging beenden
context.setSuperMode( false);
context.returnValue = esc.Error;
```

# Arbeits-Ordner

Die Eskalations-Informationen werden zunächst pro Anwender in Text-Dateien gesammelt. Der Eskalations-Ordner für Rechnungs-Belege befindet sich im Ordner "DEXPRO" der Documents-Installation unter "Escalation" und "Invoice".

**"...\\Documents5\\DEXPRO\\Escalation\\Invoice\\..."**

Der Documents-Dienste-Benutzer muss Schreibrechte auf den Verarbeitungs-Ordner haben. Im Standard befindet sich der Ordner unterhalb "..\\Documents5\\DEXPRO\\Escalation\\".

Pro Eskalations-Typ wird ein separater Unter-Ordner erstellt. Zu Beginn einer Job-Ausführung wird der bestehende Arbeitsordner geleert. Für jeden Anwender der eine Eskalations-Mail erhalten soll wird eine neue Text-Datei erstellt. Durch das Sammeln der Informationen wird erreicht, dass jeder Anwender nur eine Mail pro Eskalation erhält.

Erst nachdem alle Rechnungen kontrolliert wurden werden die Mails versendet. Die Dateien bleiben nach der Verarbeitung für den Support zur Nachvollziehbarkeit liegen und werden erst bei der nächsten Job-Ausführung gelöscht.

# Logging

Das Log wird automatisch durch das Objekt "Escalation" erstellt. Pro Eskalations-Typ wird ein Log mit Tages-Stempel geschrieben. Alle Eskalations-Logs werden in der Standardauslieferung im Ordner "Escalation" unterhalb des "DEXPRO"-Ordners abgelegt.

**"...\\Documents5\\DEXPRO\\Escalation\\..."**

Der Ablage-Ordner für die Log-Dateien kann über den Parameter "Escalation\_Paths" ebenso angepasst werden wie das Intervall, wann neue Log-Dateien erstellt werden (pro Ausführung bzw. Zeitstempel / täglich (Standard) / monatlich / jährlich). Die Log-Ausgaben innerhalb der User-Exits können beliebig angepasst werden.