Eskalation Funktionsbibliothek und User-Exits

In diesem Kapitel wird der Aufbau und die generelle technische Abarbeitung der Eskalationen beschrieben.

- "Escalation()"-Objekt

"Escalation()"-Objekt

Für die Eskalation werden unter anderem die folgenden Portal-Skript-Bibliotheken benötigt:

- **DEXPRO__EscalationLib**
- **DEXPRO__UserExit_Escalation // Invoice**
- **DEXPRO__UserExit_Escalation_Mailroom**
- **DEXPRO__UserExit_Escalation_Procurement**

Für den eigentlichen Eskalations-Aufruf wird ein Portal-Skript als Job eingeplant. Im Job-Skript wird ein "Escalation()" Objekt erstellt. Dem Objekt muss der Mappentyp-Name als erster Parameter mitgegeben werden. Als zweiter Parameter wird der Eskalations-Typ mitgegeben. Im Standard sind die Typen "Deadline", "Action" und "PPC" definiert - es können aber beliebige Typen projektspezifisch definiert werden. Das folgende Skript zeigt ein Beispiel für einen "Deadline"-Eskalations-Job.

```
// #import "Invoice__ImportLib"
var esc = new Escalation("Invoice", "Deadline");
if( esc.Result===true ){
    // Arbeits-Ordner "aufräumen"
    esc.cleanUpFilePath();
    // Alle Rechnungs-Mappen iterieren und Zugriffsberechtigungen ignorieren
    context.setSuperMode(true);
    var frs = new FileResultset("Invoice", "", "");
    for( var docFile=frs.first(); docFile; docFile=frs.next() ){
        // Eskalations-Dateien pro Benutzer erstellen
        // -> esc.invoice_checkEscalation(docFile)
        // -> esc.invoice_getEscalationUserArray(docFile)
        // -> esc.invoice_addBodyContent
        //      -> esc.invoice_createTableHead();
        //      -> esc.invoice_createTableEntry();
        esc.escalate(docFile);
    }
    // Versenden der Eskalations-Mails
    // -> esc.invoice_closeTable(su)
    // -> esc.invoice_sendMail(su, html)
    esc.closeFilesAndSendMails();
    esc.Log.stop();
    context.setSuperMode(false);
```

```
    if( esc.Result===true ){
        return 1;
    }
}
return esc.Error;
```

Das "**Escalation**"-Objekt erhält 2 Parameter und diverse Property.

@param	**{string}**	**template**	**Name des aufrufenden Mappentypen ("Invoice", "Mailroom")**
@param	**{string}**	**escType**	**Eskalations-Typ (Standard: "Deadline", "Action", "PPC")**
@property	**{string}**	**Error**	**Fehlermeldung**
@property	**{boolean}**	**Result**	**Ergebnis (true/false)**
@property	**{string}**	**Template**	**Name des aufrufenden Mappentypen ("Invoice", "Mailroom")**
@property	**{string}**	**EscalationType**	**Eskalations-Typ (Standard: "Deadline", "Action", "PPC")**
@property	**{string}**	**LogPath**	**Pfad für die Ablage der Logdatei**
@property	**{string}**	**Path**	**Arbeits-Pfad, in dem pro Benutzer eine Datei mit den Informationen der eMail abgelegt wird.**
@property	**{string}**	**LogInt**	**Intervall für den Zeitpunkt, zu dem eine neue Logdatei erstellt wird ("timestamp", "date" (Standard), "month", "year").**
@property	**{Log}**	**Log**	**new Log(context.scriptName, this.LogPath, this.LogInt)**

Funktionsliste (DEXPRO__EscalationLib):

boolean **setEscalationType(String newEscalationType)**
Setzt einen neuen Eskalations-Typen.
Return: true / false

boolean **cleanUpFilePath()**
Löscht alle Dateien aus dem Verarbeitungs-Ordner zum aktuellen Eskalations-Typen.
Return: true / false

boolean **escalate(DocFile docFile)**
Prüft die Eskalation für die aktuelle Mappe. Ggf. wird ein Eskalations-Eintrag pro Anwender hinzugefügt. Bei Gruppen erhält jeder Benutzer aus der Gruppe eine Eskalations-Mail. Die Funktion ruft weitere interne Funktionen auf.
- this.checkEscalation(DocFile docFile)
- this.getEscalationUserArray(DocFile docFile)
- this.addBodyContent(DocFile docFile)
Return: true / false

boolean **addBodyContent(DocFile docFile, SystemUser userobj)**
Wenn zum aktuellen Benutzer noch keine Info-Datei existiert wird eine neue Datei erstellt. Andernfalls wird die bestehende Datei verwendet. Die Funktion fügt einen Tabellen-Eintrag für die aktuelle Rechnung hinzu und ruft hierfür weitere interne User-Exit Funktionen auf.
- this.createTableHead(DocFile docFile, SystemUser userobj, String lang)
- this.createTableEntry(DocFile docFile, SystemUser userobj, String lang)
Return: true / false

boolean **closeFilesAndSendMails(DocFile docFile, SystemUser userobj)**
Die Funktion iteriert alle erstellten Dateien aus dem Arbeits-Ordner. Die Funktion schließt die Html-Tabelle und versendet die HTML als Mail. Hierfür werden die folgenden User-Exit-Funktionen aufgerufen:
- this.closeTable(SystemUser su)
- this.sendMail(SystemUser su, String html)
Return: true / false

boolean **parseDateToLocaleString(Date dateObj, String language)**
Die Funktion wandelt ein Datums-Objekt in einen String um. Das Datumsformat wird automatisch über die Sprache ermittelt. Bei einem ungültigen Datumsobjekt wird ein Leerstring zurückgegeben. Wenn der Parameter bereits vom Typ "String" ist wird der String zurückgegeben.

Return: Date as String

Neben den Standard-Funktion werden einige Funktionen als User-Exit-Funktionen ausgeliefert. Die Funktionen sind in der Bibliothek DEXPRO__EscalationLib enthalten und können projektspezifisch angepasst werden.

Ab der Invoice-Version 1.1.110 werden eigene Funktionen pro Modul (Invoice/Mailroom/Procurement) aufgerufen und pro Modul wird eine separate UserExit-Lib ausgeliefert. Wenn bei einem Invoice-Update weiterhin die alten Funktionen verwendet werden sollen, dann kann einfach weiterhin die alte Lib "DEXPRO__UserExit_EscalationLib" verwendet werden. Wenn die neuen Funktionen nicht existieren, dann werden weiterhin die alten Funktionen verwendet.

boolean **checkEscalation(DocFile df) // Alt**
boolean **invoice_CheckEscalation(DocFile df)**
boolean **mailroom_CheckEscalation(DocFile df)**
boolean **procurement_CheckEscalation(DocFile df)**
Die Funktion wird pro DocFile-Objekt aufgerufen und gibt durch die Rückgabe zurück ob eine Eskalation stattfinden soll oder nicht. Im ausgelieferten Beispiel werden bereits die Eskalations-Typen und die Mappentypen unterschiedlich behandelt.
Return: true / false

boolean **getEscalationUserArray(DocFile df) // Alt**
boolean **invoice_GetEscalationUserArray(DocFile df)**
boolean **mailroom_GetEscalationUserArray(DocFile df)**
boolean **procurement_GetEscalationUserArray(DocFile df)**
Die Funktion ermittelt pro DocFile-Objekt die zu informierenden Benutzer. Für jeden ermittelten Benutzer wird ein eMail-Eintrag in die Datei zum Benutzer erzeugt.
Return: Array mit SystemUser-Objekten

boolean **createTableHead(DocFile df, SystemUser su, String lang)**
boolean **invoice_CreateTableHead(DocFile df,SystemUser su,String lang)**
boolean **mailroom_CreateTableHead(DocFile df,SystemUser su,String lang)**
boolean **procurement_CreateTableHead(DocFile df,SystemUser su,String lang)**
Wenn zu einem Benutzer noch keine Eskalations-Text-Datei existiert, dann wird zunächst eine neue Datei angelegt. Die Mail wird als HTML-Mail erzeugt und soll in der Regel einen einleitenden Info-Text zur Eskalation enthalten. Die ermittelten Rechnungen werden anschließend tabellarisch gelistet und hierfür muss zunächst die Zeile mit den Tabellen-Überschriften generiert werden. Die Tabelle kann je nach Mappentyp und Eskalations-Typ unterschiedliche Spalten mit unterschiedlich

aufbereiteten Informationen enthalten.
Return: true / false

boolean **createTableEntry(DocFile df, SystemUser su, String lang)**
boolean **invoice_CreateTableEntry(DocFile df, SystemUser su, String lang)**
boolean **mailroom_CreateTableEntry(DocFile df, SystemUser su, String lang)**
boolean **procurement_CreateTableEntry(DocFile df, SystemUser su, String lang)**
Die Funktion fügt einen Tabellen-Eintrag zu einer existierenden Eskalations-Text-Datei hinzu. Die Tabelle kann je nach Mappentyp und Eskalations-Typ unterschiedliche Spalten mit unterschiedlich aufbereiteten Informationen enthalten.
Return: true / false

boolean **closeTable(SystemUser su)**
boolean **invoice_CloseTable(SystemUser su)**
boolean **mailroom_CloseTable(SystemUser su)**
boolean **procurement_CloseTable(SystemUser su)**
Nachdem alle Mappen auf Eskalation geprüft wurden muss bei allen Dateien sowohl die HTML-Tabelle als auch der "body" und der "html" Tag geschlossen werden. Ggf. kann noch ein abschließender Text angefügt werden.
Return: true / false

boolean **sendMail(SystemUser su, String body)**
boolean **invoice_SendMail(SystemUser su, String body)**
boolean **mailroom_SendMail(SystemUser su, String body)**
boolean **procurement_SendMail(SystemUser su, String body)**
Zum Schluss muss die Eskalation als Email versendet werden. Die Funktion erhält den Benutzer als SystemUser-Objekt und den gesamten eMail-Text im HTML-Format aus der Datei. Absender und Betreff können über die properties-Sprachdateien definiert werden.
Return: true / false