

Workflow-Aktion

"Buchungsschnittstelle"

Die technische Workflow-Aktion für die Buchungsschnittstelle stellt die Rechnungsdaten im ersten Schritt in SQL-Tabellen zur Verfügung und wartet im zweiten Schritt auf die Rückmeldung vom Buchungssystem.

The screenshot shows the DEXPRO Solutions Gm... interface. On the left is a sidebar with navigation options: '+ Neue Mappe', 'Favoriten', 'Eingang' (644), 'Aufgaben', 'Zuletzt benutzt', 'Wiedervorlage', 'Gelöscht', 'Workflow', 'Workflow Regeln', 'Workflows', 'Workflow Aktionen' (selected), 'Workflow Konfiguration', 'Feldkonfigurationen', 'Parameter', 'Initiale Berechtigungen', 'Mandanten', and 'Übersetzungen'. The main area displays a list of actions: 'Aktion', 'Posting (Buchungsschnittstelle)', and 'PrePosting (Vorgang)'. A 'NEUER EINTRAG' button is visible. A modal window titled 'Aktionstyp' is open, showing the configuration for the 'Posting' action. The modal contains three sections: 'Technischer Name und Anzeigewert', 'Aufgabentexte', and 'Aktionsdefinition'. The 'Technischer Name und Anzeigewert' section shows 'Posting' with a value of 7 / 50. The 'Aufgabentexte' section shows 'Buchungsschnittstelle' with a value of 21 / 500 and 'Posting-Interface' with a value of 17 / 500. The 'Aktionsdefinition' section shows 'Technische Aktion' as the action type and 'Job' as the executing action. At the bottom of the modal are buttons for 'SCHLIESSEN' and 'SPEICHERN'.

Technischer Name und Anzeigewert	
Aktion	Posting
Deutsch	Buchungsschnittstelle
Englisch	Posting Interface

Aufgabentexte	
Deutsch	Buchungs-Schnittstelle
Englisch	Posting-Interface

Aktionsdefinition	
Aktionstyp	Technische Aktion
Ausführende Aktion	Job

Technisch wird in dem Skript "DEXPRO_WF_TechAction1_Send" die UserExit-Funktion "**ue_post()**" aufgerufen. Die Funktion befindet sich im Skript "**DEXPRO_UserExit_TechActionLib**" und kann somit projektspezifisch angepasst und erweitert werden. Die Funktion erstellt ein PostObject().

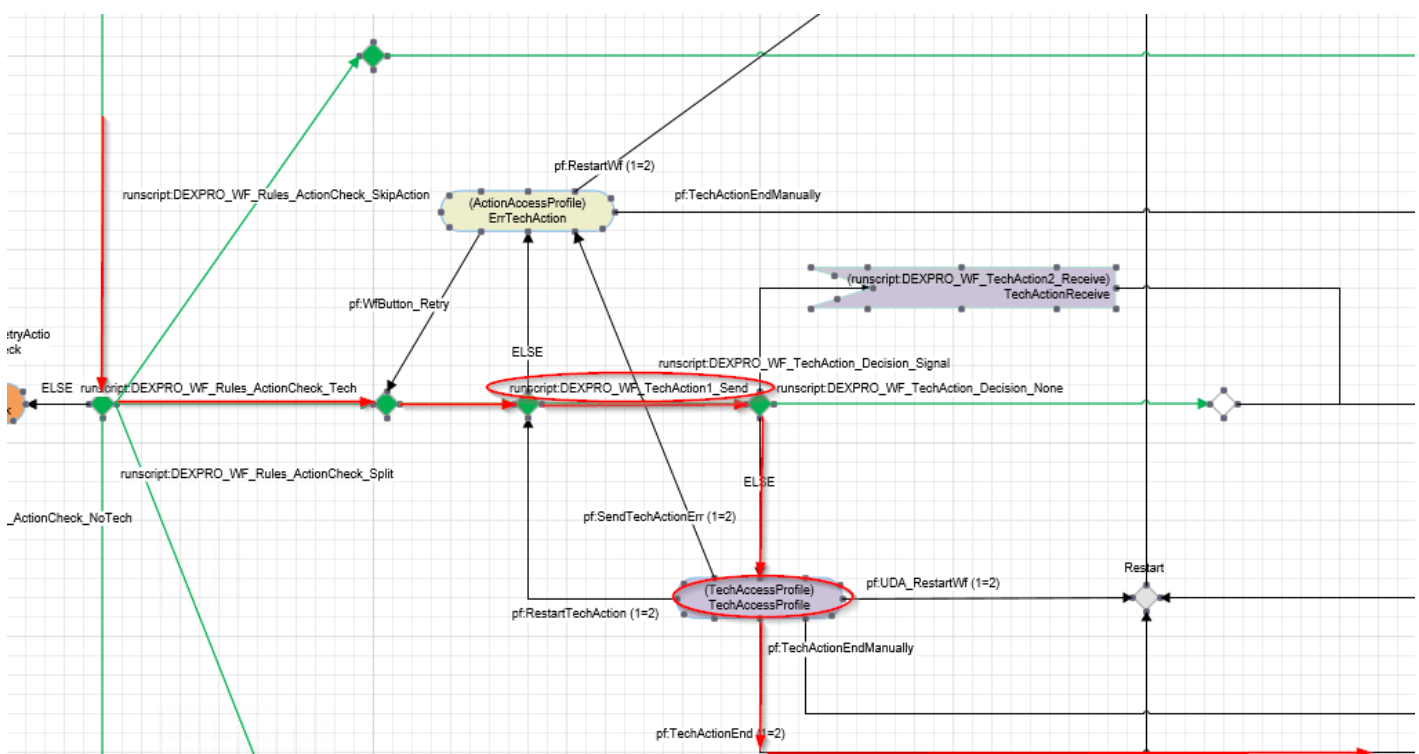
```
var postObj = new PostObject();
```

Das Skript ermittelt den aktuellen Status in der Tabelle über die Funktion "**postObj.getStatusValues()**"

". Wenn der ermittelte Status ok ist werden die Rechnungsdaten erneut in die Tabelle geschrieben. Sollte der Status bereits auf "ready" (cPostingStatusReady) stehen könnte es sein, dass die Daten bereits an das Zielsystem übertragen wurden. Ein erneutes Senden könnte zu zwei Buchungssätzen zu einer Rechnung führen. Dasselbe gilt für die Status "transfer" (cPostingStatusTransfer), "transferred" (cPostingStatusTansferred) und erst recht für bereits gebuchte Rechnungen mit dem Status "posted" (cPostingStatusPosted). Gültige Status bei denen die Rechnungsdaten geschrieben werden sind "workflow" (cPostingStatusWF) und "error" (cPostingStatusError).

Der Status darf erst auf "ready" umgestellt werden, wenn alle projektspezifischen Anpassungen erfolgreich durchgelaufen sind!

Nach dem Schreiben in die Datenbank kann eine weitere projektspezifische Verarbeitung angefügt werden. Die Rechnungswerte können zum Beispiel zusätzlich in einem definierten Dateiformat in ein Übergabe-Verzeichnis abgelegt werden oder können zusätzlich in eine weitere Tabelle geschrieben werden oder können via WEB-Service direkt das Zielsystem gesendet werden.



Im Anschluss wird die Rechnungs-Mappe an die im Feld "TechAccessProfile" hinterlegte Gruppe weitergeleitet. Im Standard steht die gleichnamige Gruppe im Feld. Das Feld "ActionStatus" wird auf den Wert "TechActionJob" gesetzt. Über diesen Wert zusammen mit dem Wert "Posting" aus dem Feld "ActionID" können die Mappen leicht gefiltert werden.

```
var filter = "ActionID='Posting' AND ActionStatus='TechActionJob'";
var frs = new FileResultset("Invoice", filter, "");
```

An dieser Stelle wird bewusst auf den Signaleingang verzichtet. Bei einem Signaleingang wird das hinterlegte Prüfskript alle 5 Minuten pro wartende Mappe ausgeführt. Das kann unter Umständen zu Performance-Problemen führen. Häufig werden alle bereitgestellten Daten nur einmal am Tag in einem Zug an das Zielsystem übertragen. Die Rückmeldung kann in vielen Fällen effizienter durch ein Job-Skript abgebildet werden.

Das Job-Skript "**Invoice_JOB_CheckPostingStatus**" überprüft den Status in der Tabelle und leitet die Mappen beim Status "posted" weiter im Workflow. Beim Status "error" wird an die im Feld "ActionAccessProfile" hinterlegte Gruppe weitergeleitet und das Feld "ActionStatus" wird bei Eingang in die Aktion auf "TechActionError" gesetzt. Das Feld "ActionAccessProfile" wird übrigens in der UserExit-Funktion "ue_Post()" gesetzt und kann auf eine beliebige Gruppe geändert werden.

Das Job-Skript ist in der Standard-Auslieferung nicht aktiv und kann auf beliebige Ausführungszeitpunkte konfiguriert werden. Hierfür muss die Checkbox "Als Job starten" auf dem Register "Job" aktiviert werden. Job-Skripte können entweder zu fest definierten Zeitpunkten oder im Intervall ausgeführt werden. Falls das kürzeste Intervall "Viertelstündlich" nicht ausreichend sein sollte kann über die bereits gesetzte Eigenschaft "frequency" ein noch häufigeres Intervall konfiguriert werden (Angabe in Minuten).

Die voreingestellte "frequency" von einer Minute ist für Demo-Systeme und für die Testphase vor der Produktivsetzung hilfreich. Beim Go-Live sollte die Ausführungshäufigkeit sinnvoll neu konfiguriert werden.

Gültig ab ☒ 21.05.2019 16:38:38

Invoice_JOB_CheckPostingStatus - Skript *

?

X

Allgemein Beschreibung Job Testen Eigenschaften

Revision #3

Created 19 October 2020 06:40:46 by Markus Meisner

Updated 19 October 2020 06:47:13 by Markus Meisner

Bezeichnung

Wert