

Events vor dem Abholen und Anlegen von SQUEEZE-Belegen

Zielgruppe

Diese Dokumentation richtet sich an Drittanbieter-Entwickler, die steuern möchten, **ob** Business Central Belege aus SQUEEZE abholt – typischerweise, um das Abholen in einem Testsystem zu unterbinden.

Im Fokus stehen die beiden **Integration Events** in der Codeunit `DXP SQZ API Mgt.`, die *vor* dem Abholen ausgelöst werden.

Die beiden Events sind ab SQUEEZE Version 2.54.0.0 verfügbar.

Hintergrund: Warum ein Testsystem keine Belege abholen sollte

Ist ein Testsystem mit derselben SQUEEZE-Instanz verbunden wie das Produktivsystem – etwa weil es als Kopie der Produktion angelegt wurde –, holt es dort ebenfalls Belege ab.

Beim Anlegen wird ein Beleg in SQUEEZE auf `FINISHED` gesetzt. Ein Beleg, den ein Testsystem abgeholt hat, steht dem Produktivsystem danach nicht mehr zur Verfügung.

Soll das Testsystem gar nicht mit SQUEEZE kommunizieren, geht das auch ohne Erweiterung, siehe [Testsystem holt Belege aus Squeeze ab](#). Die Events sind für die Fälle gedacht, in denen das Testsystem das Abholen selbst erkennen und unterbinden soll oder nur bestimmte Belege zurückgehalten werden sollen.

Einordnung: Ablauf des Abholens

Das Abholen läuft in zwei Schritten, beide in der Codeunit `DXP SQZ API Mgt.`:

1. `DownloadDocumentData` ruft die IDs der Belege ab, die in SQUEEZE zum Export bereitstehen, und trägt sie in die [Importwarteschlange](#) ein. Außerdem ruft es Belege im Status „Neu erstellen“ (Resynchronize) erneut ab.
2. `CreateCoreAndSQUEEZEDocument` ruft für jeden offenen Eintrag der Importwarteschlange den Beleg ab, legt den Core-Beleg (`DXP Document`) und den SQUEEZE-Beleg an, markiert den Eintrag als verarbeitet und setzt den Beleg in SQUEEZE auf `FINISHED`.

Beide Schritte laufen in der Aufgabenwarteschlange und bei der Aktion **Belege jetzt herunterladen** auf den Seiten Squeeze Belege, Squeeze Importwarteschlange und Squeeze Warteschlange. Vor jedem der beiden Schritte wird ein Event mit `IsHandled` ausgelöst.

Event-Matrix (Kurzüberblick)

Event	Zeitpunkt	Steuerung	Typische Use-Cases
<code>OnBeforeDownloadDocumentData</code>	vor dem Abruf der Beleg-IDs aus SQUEEZE	<code>IsHandled</code>	Abholen im Testsystem vollständig unterbinden
<code>OnBeforeCreateCoreAndSQUEEZEDocumentV2</code>	vor dem Abruf und der Anlage eines einzelnen Belegs	<code>IsHandled</code>	bereits eingereihte Belege nicht anlegen; einzelne Dokumentenklassen oder Belege zurückhalten

Events in `DXP SQZ API Mgt.`

`OnBeforeDownloadDocumentData`

Wird ganz am Anfang von `DownloadDocumentData` ausgelöst, bevor eine Verbindung zu SQUEEZE aufgebaut wird.

Signatur:

```
[IntegrationEvent(false, false)]
local procedure OnBeforeDownloadDocumentData(var IsHandled: Boolean)
```

Mit `IsHandled := true` wird der gesamte Schritt übersprungen: Es werden keine Beleg-IDs abgerufen, keine Einträge in die Importwarteschlange geschrieben und keine Belege im Status „Neu erstellen“ erneut abgerufen.

OnBeforeCreateCoreAndSQUEEZEDocumentV2

Wird in `CreateCoreAndSQUEEZEDocument` für jeden Beleg ausgelöst, bevor er aus SQUEEZE abgerufen wird.

Signatur:

```
[IntegrationEvent(false, false)]  
local procedure OnBeforeCreateCoreAndSQUEEZEDocumentV2(DocClass: Enum "DXP Document Class";  
SQUEEZEDocId: Text[50]; var IsHandled: Boolean)
```

Parameter:

- `DocClass` – Dokumentenklasse des Belegs, z. B. `DXP Invoice / Credit Memo` oder `DXP Order Confirmation`
- `SQUEEZEDocId` – ID des Belegs in SQUEEZE
- `IsHandled` – auf `true` setzen, um den Beleg zu überspringen

Mit `IsHandled := true`:

- wird weder ein Core-Beleg noch ein SQUEEZE-Beleg angelegt,
- bleibt der Beleg in SQUEEZE unverändert – er wird nicht auf `FINISHED` gesetzt,
- bleibt der Eintrag in der Importwarteschlange unverarbeitet und wird beim nächsten Lauf erneut angeboten.

Um ein Testsystem vollständig abzuschirmen, braucht ihr dieses Event zusätzlich zu `OnBeforeDownloadDocumentData`: Eine Kopie des Produktivsystems bringt auch die Einträge mit, die dort bereits in der Importwarteschlange standen. Ohne dieses Event würden sie im Testsystem angelegt.

Übersprungene Einträge werden bei jedem Lauf erneut versucht und zählen dabei zu den bis zu 10 Einträgen, die ein Lauf verarbeitet. Haltet ihr nur einzelne Belege zurück, können diese Einträge die Verarbeitung der übrigen aufhalten.

Das ältere Event `OnBeforeCreateCoreAndSQUEEZEDocument` ist obsolet und wird nicht mehr ausgelöst. Verwendet stattdessen `OnBeforeCreateCoreAndSQUEEZEDocumentV2`.

Minimalbeispiel: Abholen in einer Sandbox unterbinden

Das Beispiel behandelt jede Sandbox als Testsystem. Passt Objekt-IDs/Namen an euren Namespace und die Prüfung in `IsTestSystem` an eure Regeln an.

```
codeunit 50102 "My SQZ Test System Guard"
{
    [EventSubscriber(ObjectType::Codeunit, Codeunit::"DXP SQZ API Mgt.",
    'OnBeforeDownloadDocumentData', '', false, false)]

    local procedure SkipDownloadInTestSystem(var IsHandled: Boolean)
    begin
        if IsTestSystem() then
            IsHandled := true;
    end;

    [EventSubscriber(ObjectType::Codeunit, Codeunit::"DXP SQZ API Mgt.",
    'OnBeforeCreateCoreAndSQUEEZEDocumentV2', '', false, false)]

    local procedure SkipCreationInTestSystem(DocClass: Enum "DXP Document Class";
    SQUEEZEDocId: Text[50]; var IsHandled: Boolean)
    begin
        if IsTestSystem() then
            IsHandled := true;
    end;

    local procedure IsTestSystem(): Boolean
    var
        EnvironmentInformation: Codeunit "Environment Information";
    begin
        // Beispiel: jede Sandbox gilt als Testsystem.
        // On-Premises oder bei anderen Regeln eine eigene Prüfung einsetzen,
        // z. B. ein Kennzeichen in einer eigenen Einrichtungstabelle.
        exit(EnvironmentInformation.IsSandbox());
    end;
}
```

Bei der Aktion **Belege jetzt herunterladen** erhält der Benutzer keine Rückmeldung, dass das Abholen übersprungen wurde. Wenn ihr das möchtet, zeigt im Subscriber eine Meldung an – nur wenn `GuiAllowed()` zutrifft, damit die Aufgabenwarteschlange davon unberührt bleibt.

Revision #1

Created 2026-10-27 14:39:35 UTC by Bernd Feddersen

Updated 2026-10-27 14:39:35 UTC by Bernd Feddersen