

Events before retrieving and creating SQUEEZE documents

Audience

This documentation is for third-party developers who want to control **whether** Business Central retrieves documents from SQUEEZE – typically to prevent a test system from retrieving them.

It focuses on the two **integration events** in codeunit `DXP SQZ API Mgt.` that are raised *before* documents are retrieved.

Both events are available from SQUEEZE version 2.54.0.0.

Background: why a test system should not retrieve documents

If a test system is connected to the same SQUEEZE instance as the production system – for example because it was created as a copy of production – it retrieves documents from there as well.

When a document is created, it is set to `FINISHED` in SQUEEZE. A document retrieved by a test system is no longer available to the production system.

If the test system should not communicate with SQUEEZE at all, no extension is needed, see [Test system retrieves documents from Squeeze](#). The events are meant for cases where the test system should detect and prevent the retrieval by itself, or where only certain documents should be held back.

Context: how documents are retrieved

Retrieval runs in two steps, both in codeunit `DXP SQZ API Mgt.`:

1. `DownloadDocumentData` fetches the ids of the documents waiting for export in SQUEEZE and adds them to the [import queue](#). It also fetches documents with status Resynchronize again.
2. `CreateCoreAndSQUEEZEDocument` fetches the document for each open import queue entry, creates the Core document (`DXP Document`) and the SQUEEZE document, marks the entry as processed and sets the document to `FINISHED` in SQUEEZE.

Both steps run in the job queue and with the **Download Documents now** action on the Squeeze Documents, Squeeze Import Queue Entries and Squeeze Queue pages. Each step is preceded by an event with `IsHandled`.

Event matrix (overview)

Event	When	Control	Typical use cases
<code>OnBeforeDownloadDocumentData</code>	before the document ids are fetched from SQUEEZE	<code>IsHandled</code>	prevent all retrieval in a test system
<code>OnBeforeCreateCoreAndSQUEEZEDocumentV2</code>	before a single document is fetched and created	<code>IsHandled</code>	do not create documents that are already queued; hold back certain document classes or documents

Events in `DXP SQZ API Mgt.`

`OnBeforeDownloadDocumentData`

Raised at the very start of `DownloadDocumentData`, before any connection to SQUEEZE is made.

Signature:

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

With `IsHandled := true` the whole step is skipped: no document ids are fetched, no import queue entries are written and no documents with status Resynchronize are fetched again.

`OnBeforeCreateCoreAndSQUEEZEDocumentV2`

Raised in `CreateCoreAndSQUEEZEDocument` for each document, before it is fetched from SQUEEZE.

Signature:

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

Parameters:

- `DocClass` – document class of the document, e.g. `DXP Invoice / Credit Memo` or `DXP Order Confirmation`
- `SQUEEZEDocId` – id of the document in SQUEEZE
- `IsHandled` – set to `true` to skip the document

With `IsHandled := true`:

- neither a Core document nor a SQUEEZE document is created,
- the document in SQUEEZE stays unchanged – it is not set to `FINISHED`,
- the import queue entry stays unprocessed and is offered again on the next run.

To shield a test system completely, you need this event in addition to `OnBeforeDownloadDocumentData`: a copy of the production system also brings along the entries that were already in its import queue. Without this event they would be created in the test system.

Skipped entries are retried on every run and count towards the up to 10 entries a run processes. If you hold back only some documents, these entries can delay the processing of the others.

The older event `OnBeforeCreateCoreAndSQUEEZEDocument` is obsolete and no longer raised. Use `OnBeforeCreateCoreAndSQUEEZEDocumentV2` instead.

Minimal example: prevent retrieval in a sandbox

The example treats every sandbox as a test system. Adapt object ids/names to your namespace and the check in `IsTestSystem` to your rules.

```
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
                // Example: every sandbox counts as a test system.
                // On-premises or for other rules, use a check of your own,
                // e.g. a flag in a setup table of your own.
                exit(EnvironmentInformation.IsSandbox());
            end;
        }
    }

```

With the **Download Documents now** action the user gets no feedback that retrieval was skipped. If you want that, show a message in the subscriber – only when `GuiAllowed()` is true, so the job queue is not affected.

Revision #1

Created 2026-10-27 14:39:38 UTC by Bernd Feddersen

Updated 2026-10-27 14:39:38 UTC by Bernd Feddersen