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
`OnBeforeDownloadDocumentData`	before the document ids are fetched from SQUEEZE	`IsHandled`	prevent all retrieval in a test system; switch a test system to a SQUEEZE test instance
`OnBeforeCreateCoreAndSQUEEZEDocumentV2`	before a single document is fetched and created	`IsHandled`	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.

The Squeeze Setup is only read after this event. A subscriber can therefore also switch the connection to a SQUEEZE test instance here instead of preventing retrieval – see the section **Connecting a test system to a SQUEEZE test instance** below.

`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. the database name from the "Active Session" table.
        // A flag in a table of your own does not work when the
        // test system is created as a copy of production.
        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.

Connecting a test system to a SQUEEZE test instance

Instead of preventing retrieval, a test system can also work with a SQUEEZE test instance of its own. This pays off especially when the test system is rebuilt regularly from a backup of production – for example every sprint – and should then run against the test instance without manual work.

You do not need a dedicated event or a parameter for the connection. SQUEEZE reads the Squeeze Setup from the database, and during retrieval only after `OnBeforeDownloadDocumentData`. Once you switch the setup to the test instance permanently, all subsequent calls go there: retrieving and creating documents, reporting `FINISHED`, attachments, viewer, training and master data synchronization.

Switching the connection

1. Assign the `URL` field of the setup **directly** – without `Validate` and without a trailing slash. `Validate` on this field opens the SQUEEZE login as a dialog. The job queue and `Invoke-NAVCodeunit` have no dialog, so the call fails there with an error.
2. Then call `GetAPIData(SqzSetup, ApiKey, false)` from `DXP SQZ API Mgt.` with the API key of the test instance. This is the same path as the login on the setup page, only without the dialog: the key is checked against the test instance and stored, the version and tenant of the test instance are taken over, the master data tables are determined again and the setup is saved.

> Do not put the API key of the test instance into the source code. Store it, for example, in the isolated storage of your own extension. If you store it in the production system, it is available in the test system after every copy.

What else you need to switch

- **Document classes:** The **Squeeze Document Class ID** and **Squeeze Export Interface ID** fields in the Document Classes Setup belong to a specific SQUEEZE instance. If they differ in the test instance, switch them in the same step.
- **Open import queue entries:** The copy brings along the entries that were not yet processed in production. Processing does not filter by tenant, so these documents are looked up in the test instance. If the test instance does not know them, every attempt fails, is repeated on every run and takes up slots of the up to 10 entries per run. Delete them after switching with `DeleteUnprocessedImportQueueEntries` from `DXP SQUEEZE Setup Mgt.`, passing the previous tenant. Note it from the `Tenant` field of the setup before you

switch. In a session with a user interface, the function asks for confirmation before deleting.

When to switch

- **Right after the rebuild (recommended):** On-premises, call your codeunit with `Invoke-NAVCodeunit` for each company in the script that restores the backup – before the job queue starts. Restoring a backup does not put job queue entries on hold. Without this step the SQUEEZE tasks, e.g. retrieval and master data synchronization, start right away with the production connection.
- **In addition, in `OnBeforeDownloadDocumentData` as a safeguard:** The same switch in the event, with `IsHandled` left at `false`. The current retrieval then already runs against the test instance. The event only applies to retrieval, though. Anything that runs before it, e.g. master data synchronization, still uses the production connection.

Recognizing the test system

Do not recognize the test system by a flag in a table. Tables are overwritten with every copy of production, so the test system then holds the production value again. On-premises, use the database or server instance name from the `Active Session` table; in Business Central online, `IsSandbox()` or `GetEnvironmentName()` from codeunit `Environment Information`.

Example

The example switches an on-premises test system that is recognized by its database name. Adapt object ids, names, the check in `IsTestSystem` and where URL and API key come from to your environment.

```
codeunit 50103 "My SQZ Test Connection"
{
    // On-premises: run for each company after restoring the backup, e.g.
    // Invoke-NAVCodeunit -ServerInstance BC -CompanyName 'CRONUS International Ltd.' -
CodeunitId 50103
    trigger OnRun()
    begin
        ApplyTestConnection();
    end;

    [EventSubscriber(ObjectType::Codeunit, Codeunit::"DXP SQZ API Mgt.",
'OnBeforeDownloadDocumentData', '', false, false)]
    local procedure SwitchToTestConnection(var IsHandled: Boolean)
```

```
begin
    // Safeguard in case the call after the restore was missed.
    // IsHandled stays false: retrieval then runs against the test instance.
    ApplyTestConnection();
end;

local procedure ApplyTestConnection()
var
    SqzSetup: Record "DXP SQUEEZE Setup";
    SqzApiMgt: Codeunit "DXP SQZ API Mgt.";
    SqzSetupMgt: Codeunit "DXP SQUEEZE Setup Mgt.";
    PreviousTenant: Text[250];
begin
    if not IsTestSystem() then
        exit;
    if not SqzSetup.Get() then
        exit;
    if SqzSetup.URL = GetTestUrl() then
        exit;

    PreviousTenant := SqzSetup.Tenant;
    SqzSetup.URL := GetTestUrl(); // no Validate: it opens the login dialog
    SqzApiMgt.GetAPIData(SqzSetup, GetTestApiKey(), false);

    // Remove open import queue entries that came from production
    SqzSetupMgt.DeleteUnprocessedImportQueueEntries(PreviousTenant);

    // If the document class id or export interface id differs in the
    // test instance, adjust "DXP Document Class Setup" here as well.
end;

local procedure IsTestSystem(): Boolean
var
    ActiveSession: Record "Active Session";
begin
    // Do not use a table of your own: it is copied along.
    ActiveSession.Get(ServiceInstanceId(), SessionId());
    exit(ActiveSession."Database Name" = 'BC_TEST');
end;
```

```
    local procedure GetTestUrl(): Text[1024]
    begin
        // URL of the test instance without a trailing slash,
        // e.g. from a setup of your own
    end;

    local procedure GetTestApiKey(): Text
    begin
        // API key of the test instance, e.g. from the isolated storage
        // of your own extension - not in the source code
    end;
}
```

> In the event, the safeguard only costs reading the session and the setup on every retrieval: once the URL has been switched, `ApplyTestConnection` exits right away.

Revision #2
Created 2026-10-27 14:39:38 UTC by Bernd Feddersen
Updated 2026-11-17 13:40:45 UTC by Bernd Feddersen