

# Anhang Download

Diese Dokumentation bietet umfassende Anleitung für externe AL-Entwickler zur effektiven Nutzung der DXP Freeze Anhang-Download-Funktionalität in ihren Business Central Erweiterungen.

## Überblick

Die DXP Freeze Result Management Codeunit bietet zwei Hauptmethoden zum Herunterladen archivierter Anhänge als ZIP-Dateien:

1. `DownloadAttachmentsAsZipWithPagination` - Lädt Anhänge aus Suchanfrage-Ergebnissen mit Paginierung herunter
2. `DownloadAttachmentsAsZipFromRecord` - Lädt Anhänge von spezifischen Business Central Datensätzen herunter

Beide Methoden organisieren Anhänge in strukturierte ZIP-Archive mit umfassenden Metadaten für Audit- und Nachverfolgungszwecke.

## Methode 1:

### DownloadAttachmentsAsZipWithPa

## Zweck

Lädt alle Anhänge aus einem Suchanfrage-Ergebnissatz herunter und verarbeitet Ergebnisse seitenweise, um große Datensätze effizient zu handhaben. Die Anhänge jedes Datensatzes werden in individuelle ZIP-Dateien innerhalb eines Haupt-ZIP-Archivs organisiert.

## Verfügbare Überladungen

### 1. Grundlegende Verwendung

```
procedure DownloadAttachmentsAsZipWithPaging(SearchQuery: Text; var ZipArchive: Codeunit
>Data Compression"; var AttachmentCount: Integer): Boolean
```

## 2. Mit Filtern

```
procedure DownloadAttachmentsAsZipWithPaging(SearchQuery: Text; var ZipArchive: Codeunit
>Data Compression"; var AttachmentCount: Integer; FilenameFilter: Text; FileExtensionFilter:
Text): Boolean
```

## 3. Vollständige Kontrolle

```
procedure DownloadAttachmentsAsZipWithPaging(SearchQuery: Text; var ZipArchive: Codeunit
>Data Compression"; var AttachmentCount: Integer; FilenameFilter: Text; FileExtensionFilter:
Text; StoreApiLink: Text; RecordsPerPage: Integer; SuppressDialog: Boolean): Boolean
```

## 4. Vorab gefüllte Datensätze (Erweitert)

```
procedure DownloadAttachmentsAsZipWithPaging(var TempFrzResultQueryHeader: Record "DXP
FRZ Query Result Header" temporary; var TempFrzResultRecordHeader: Record "DXP FRZ Record
Result Header" temporary; var TempFrzResultRecordField: Record "DXP FRZ Result Record-Field"
temporary; var TempFrzAttachmentResult: Record "DXP FRZ Attachment Result" temporary;
SearchQuery: Text; var ZipArchive: Codeunit "Data Compression"; var AttachmentCount: Integer;
FilenameFilter: Text; FileExtensionFilter: Text; SuppressDialog: Boolean): Boolean
```

## Parameter

Parameter	Typ	Beschreibung
<code>SearchQuery</code>	Text	Freeze-Suchanfrage-String. Wenn leer in vorab gefüllter Überladung, verwendet vorhandene Anfrage oder führt Suche erneut aus
<code>ZipArchive</code>	Codeunit "Data Compression"	ZIP-Archiv-Objekt, das die heruntergeladenen Dateien enthalten wird
<code>AttachmentCount</code>	Integer (var)	Gibt die Gesamtanzahl der heruntergeladenen Anhänge zurück

Parameter	Typ	Beschreibung
<code>FilenameFilter</code>	Text	Filter für Anhangsdateinamen (z.B. <code>'pdf'</code> , <code>'rechnung'</code> )
<code>FileExtensionFilter</code>	Text	Filter für Dateierweiterungen (z.B. <code>'pdf'</code> , <code>'docx'</code> )
<code>StoreApiLink</code>	Text	Optional spezifischer Store-API-Link
<code>RecordsPerPage</code>	Integer	Anzahl Datensätze pro Seite (Standard: 100)
<code>SuppressDialog</code>	Boolean	Ob Fortschrittsdialog unterdrückt werden soll

## Rückgabewert

- `Boolean`: `true` wenn Anhänge gefunden und heruntergeladen wurden; `false` andernfalls

## Verwendungsbeispiele

### Grundlegender Download

```

procedure DownloadSearchResults()
var
  ResultMgt: Codeunit "DXP FRZ Result Mgt.";
  ZipArchive: Codeunit "Data Compression";
  FileMgt: Codeunit "File Management";
  TempBlob: Codeunit "Temp Blob";
  AttachmentCount: Integer;
  InStr: InStream;
  OutStr: OutStream;
  SearchQuery: Text;
begin
  SearchQuery := 'rechnung AND 2024';

  if ResultMgt.DownloadAttachmentsAsZipWithPagination(SearchQuery, ZipArchive,
AttachmentCount) then begin
    // ZIP in Datei speichern
    TempBlob.CreateOutputStream(OutStr);

```

```

ZipArchive. SaveZipArchive( OutStr );
TempBlob. CreateInStream( InStr );

FileMgt. DownloadFromStreamHandler( InStr, '', '', '', 'Suchergebnisse.zip' );
Message( 'Erfolgreich %1 Anhänge heruntergeladen.', AttachmentCount );
end else
    Message( 'Keine Anhänge für die Suchanfrage gefunden.' );
end;

```

## Mit Filtern

```

procedure DownloadPDFInvoices()
var
    ResultMgt: Codeunit "DXP FRZ Result Mgt.";
    ZipArchive: Codeunit "Data Compression";
    AttachmentCount: Integer;
    SearchQuery: Text;
begin
    SearchQuery := ' typ:rechnung';

    if ResultMgt. DownloadAttachmentsAsZipWithPagination(
        SearchQuery,
        ZipArchive,
        AttachmentCount,
        '*.pdf', // Nur PDF-Dateien
        '.pdf' // Dateierweiterungsfilter
    ) then begin
        // ZIP-Archiv verarbeiten
        ProcessDownloadedFiles( ZipArchive, AttachmentCount );
    end;
end;

```

## Verwendung vorab gefüllter Datensätze

```

procedure DownloadFromExistingResults()
var
    ResultMgt: Codeunit "DXP FRZ Result Mgt.";

```

```

TempFrzResultQueryHeader: Record "DXP FRZ Query Result Header" temporary;
TempFrzResultRecordHeader: Record "DXP FRZ Record Result Header" temporary;
TempFrzResultRecordField: Record "DXP FRZ Result Record-Field" temporary;
TempFrzAttachmentResult: Record "DXP FRZ Attachment Result" temporary;
ZipArchive: Codeunit "Data Compression";
AttachmentCount: Integer;
begin
    // Angenommen, diese Datensätze sind bereits aus einer vorherigen Suche gefüllt
    PopulateSearchResults(TempFrzResultQueryHeader, TempFrzResultRecordHeader,
TempFrzResultRecordField, TempFrzAttachmentResult);

    // Download mit vorhandenen Ergebnissen ohne erneute Suchausführung
    if ResultMgt.DownloadAttachmentsAsZipWithPagination(
        TempFrzResultQueryHeader,
        TempFrzResultRecordHeader,
        TempFrzResultRecordField,
        TempFrzAttachmentResult,
        'rechnungssuche', // SearchQuery - wenn leer, wird Suche erneut ausgeführt
        ZipArchive,
        AttachmentCount,
        '', // Kein Dateinamenfilter
        '', // Kein Erweiterungsfilter
        true // Dialog unterdrücken
    ) then begin
        ProcessDownloadedFiles(ZipArchive, AttachmentCount);
    end;
end;
end;

```

## ZIP-Struktur (Paginierung)

```

Suchergebnisse.zip
├─ export-metadata.json
├─ Rechnung_001_V1_20241201_1430.zip
│   └─ {GUID}_rechnung.pdf
│       └─ {GUID}_begleitdokument.docx
├─ Bestellung_002_V2_20241202_0900.zip
│   └─ {GUID}_bestelldokument.pdf

```

```
└─ Vertrag_003_V1_20241203_1200.zip
  └─ {GUID}_vertrag.pdf
    └─ {GUID}_nachtrag.pdf
```

## Methode 2:

# DownloadAttachmentsAsZipFromRe

## Zweck

Lädt Anhänge von spezifischen Business Central Datensätzen herunter. Die Anhänge jedes ausgewählten Datensatzes werden in individuelle ZIP-Dateien innerhalb eines Haupt-ZIP-Archivs organisiert.

## Verfügbare Überladungen

### 1. Grundlegende Verwendung

```
procedure DownloadAttachmentsAsZipFromRecord(var SelectedRecord: RecordRef; var ZipArchive:
Codeunit "Data Compression"): Boolean
```

### 2. Mit Filtern

```
procedure DownloadAttachmentsAsZipFromRecord(var SelectedRecord: RecordRef; var ZipArchive:
Codeunit "Data Compression"; FilenameFilter: Text; FileExtensionFilter: Text): Boolean
```

## Parameter

Parameter	Typ	Beschreibung
SelectedRecord	RecordRef (var)	RecordRef mit den ausgewählten Business Central Datensätzen

Parameter	Typ	Beschreibung
<code>ZipArchive</code>	Codeunit "Data Compression"	ZIP-Archiv-Objekt, das die heruntergeladenen Dateien enthalten wird
<code>FilenameFilter</code>	Text	Filter für Anhangsdateinamen
<code>FileExtensionFilter</code>	Text	Filter für Dateierweiterungen

## Rückgabewert

- `Boolean`: `true` wenn Anhänge gefunden und heruntergeladen wurden; `false` andernfalls

## Verwendungsbeispiele

### Download von Verkaufsrechnungen

```

procedure DownloadInvoiceAttachments()
var
    SalesInvoiceHeader: Record "Sales Invoice Header";
    ResultMgt: Codeunit "DXP FRZ Result Mgt.";
    ZipArchive: Codeunit "Data Compression";
    RecordRef: RecordRef;
    HasAttachments: Boolean;
begin
    // Spezifische Rechnungen auswählen
    SalesInvoiceHeader.SetRange("Posting Date", DMY2Date(1, 1, 2024), DMY2Date(31, 12,
2024));
    SalesInvoiceHeader.SetFilter("Sell-to Customer No.", '10000|20000');

    if SalesInvoiceHeader.FindSet() then begin
        RecordRef.GetTable(SalesInvoiceHeader);

        HasAttachments := ResultMgt.DownloadAttachmentsAsZipFromRecord(RecordRef,
ZipArchive);

        if HasAttachments then
            SaveZipFile(ZipArchive, 'Rechnungsanhänge.zip')

```

```
        else
            Message('Keine Anhänge für die ausgewählten Rechnungen gefunden. ');
        end;
    end;
end;
```

## Download mit Filtern von Seite

```
// In einer Seitenerweiterung
action(DownloadAttachmentsFiltered)
{
    Caption = 'Gefilterte Anhänge herunterladen';
    Image = ExportFile;

    trigger OnAction()
    var
        ResultMgt: Codeunit "DXP FRZ Result Mgt.";
        ZipArchive: Codeunit "Data Compression";
        RecordRef: RecordRef;
        FilenameFilter: Text;
        FileExtensionFilter: Text;
    begin
        // Filterdialog anzeigen
        if ShowFilterDialog(FilenameFilter, FileExtensionFilter) then begin
            CurrPage.SetSelectionFilter(Rec);
            RecordRef.GetTable(Rec);

            if ResultMgt.DownloadAttachmentsAsZipFromRecord(
                RecordRef,
                ZipArchive,
                FilenameFilter,
                FileExtensionFilter
            ) then
                DownloadZipFile(ZipArchive, 'GefilterteAnhänge.zip');
        end;
    end;
end;
```

# ZIP-Struktur (Datensätze)

```
Datensatzanhänge.zip
├─ export-metadata.json
├─ Sales_Invoice_Header_Unternehmen_SI-001.zip
│   ├─ {GUID}_rechnung.pdf
│   └─ {GUID}_bedingungen.pdf
├─ Sales_Invoice_Header_Unternehmen_SI-002.zip
│   └─ {GUID}_rechnung.pdf
└─ Sales_Invoice_Header_Unternehmen_SI-003.zip
    ├─ {GUID}_rechnung.pdf
    ├─ {GUID}_lieferschein.pdf
    └─ {GUID}_quittung.jpg
```

# Metadaten-Struktur

Beide Methoden generieren umfassende Metadaten in `export-metadata.json`:

# Paginierungs-Export-Metadaten

```
{
  "exportInfo": {
    "exportTimestamp": "2024-12-19T10:13:52.248Z",
    "exportedBy": "USER001",
    "searchQuery": "typ:rechnung AND jahr:2024",
    "totalRecordsFound": 150,
    "totalPages": 15,
    "exportType": "paginated-search",
    "description": "Freeze Suchanfrage Export"
  },
  "appliedFilters": {
    "filenameFilter": "*.pdf",
    "fileExtensionFilter": "pdf"
  }
}
```

```
},
  "statistics": {
    "pagesProcessed": 15,
    "totalRecordsProcessed": 150,
    "recordsWithAttachments": 120,
    "recordsWithoutAttachments": 30,
    "totalAttachments": 245,
    "exportCompletedAt": "2024-12-19T10:15:33.021Z"
  },
  "records": [
    {
      "recordId": "{GUID}",
      "title": "Rechnung REG-2024-001",
      "version": 1,
      "archivedAt": "2024-12-01T09:30:00Z",
      "archivedBy": "SYSTEM",
      "type": "Verkaufsrechnung",
      "masterId": "{GUID}",
      "attachmentCount": 3,
      "hasAttachments": true,
      "zipFile": "Rechnung_REG-2024-001_V1_20241201_0930.zip"
    }
  ]
}
```

## Datensatz-Export-Metadaten

```
{
  "exportTimestamp": "2024-12-19T14:30:00Z",
  "exportedBy": "USER001",
  "totalRecordsProcessed": 25,
  "description": "DXP Freeze Anhänge Export",
  "sourceTable": {
    "tableNumber": 112,
    "tableName": "Sales Invoice Header",
    "tableCaption": "Geb. Verkaufsrechnung"
  },
}
```

```
"appliedFilters": {
  "filenameFilter": "*.pdf",
  "fileExtensionFilter": "pdf"
},
"statistics": {
  "totalAttachments": 45,
  "recordsWithAttachments": 20,
  "recordsWithoutAttachments": 5,
  "totalZipFiles": 20
},
"records": [
  {
    "recordId": "Sales Invoice Header: Unternehmen, SI-001",
    "systemId": "{GUID}",
    "primaryKey": {
      "fields": [
        {
          "fieldName": "Nr.",
          "fieldValue": "SI-001",
          "fieldType": "Code"
        }
      ]
    },
    "hasAttachments": true,
    "attachmentCount": 2,
    "zipFile": "Sales_Invoice_Header_Unternehmen_SI-001.zip"
  }
]
}
```

# Leistungsüberlegungen

## Paginierungsmethode

- **Große Ergebnismengen:** Automatische Handhabung der Paginierung zur effizienten Verarbeitung großer Datensätze

- **Speicherverwaltung:** Verarbeitet eine Seite nach der anderen, bereinigt Speicher zwischen Seiten
- **Fortschrittsverfolgung:** Zeigt Echtzeit-Fortschritt für länger dauernde Operationen
- **Empfohlen für:** Suchanfragen, die Hunderte oder Tausende von Datensätzen zurückgeben können

## Datensatzmethode

- **Ausgewählte Datensätze:** Verarbeitet nur die spezifisch ausgewählten Datensätze
- **Direkte Verarbeitung:** Kein Paginierungs-Overhead für kleinere Datensätze
- **Stapelverarbeitung:** Effizient für die Verarbeitung spezifischer Datensatzmengen
- **Empfohlen für:** Gezielte Downloads von spezifischen Business Central Datensätzen

## Fehlerbehandlung

Beide Methoden beinhalten umfassende Fehlerbehandlung:

## Häufige Szenarien

- **Keine Ergebnisse gefunden:** Gibt `false` zurück, wenn keine Datensätze oder Anhänge gefunden werden
- **Berechtigungsprobleme:** Schließt automatisch Datensätze aus, auf die der Benutzer nicht zugreifen kann
- **API-Fehler:** Zuverlässiger Umgang mit API-Kommunikationsfehlern
- **Leere Filter:** Behandelt leere oder ungültige Filterparameter

## Best Practices

```
// Rückgabewert immer prüfen
if not ResultMgt.DownloadAttachmentsAsZipWithPagination(SearchQuery, ZipArchive,
AttachmentCount) then begin
    Message('Keine Anhänge gefunden oder Download fehlgeschlagen. ');
    exit;
end;
```

```

// Anhanganzahl validieren
if AttachmentCount = 0 then begin
    Message('Suche abgeschlossen, aber keine Anhänge entsprechen den Kriterien.');
```

exit;

```
end;

// Große Downloads handhaben
if AttachmentCount > 1000 then
    if not Confirm('Dies wird %1 Anhänge herunterladen. Fortfahren?', false, AttachmentCount)
then
    exit;
```

# Integrationsereignisse

Beide Methoden unterstützen Integrationsereignisse zur Anpassung:

## Verfügbare Ereignisse

- `OnBeforeDownloadAttachmentsAsZip`: Verhalten vor Download-Start modifizieren
- `OnBeforeProcessAttachmentForZip`: Einzelne Anhänge überspringen oder modifizieren
- `OnAfterGetAttachmentBase64`: Anhanginhalt nach Abruf modifizieren
- `OnAfterAddAttachmentToZip`: Aktionen nach Hinzufügung zum ZIP durchführen
- `OnNoAttachmentsFound`: Szenario ohne Anhänge behandeln

## Integrationsbeispiel

```

[EventSubscriber(ObjectType::Codeunit, Codeunit::"DXP FRZ Result Mgt.",
'OnBeforeProcessAttachmentForZip', '', false, false)]
local procedure OnBeforeProcessAttachmentForZip(var TempFrzAttachmentResult: Record "DXP FRZ
Attachment Result" temporary; var IsHandled: Boolean)
begin
    // Anhänge größer als 10MB überspringen
    if TempFrzAttachmentResult.FileSize > 10485760 then
```

```
IsHandled := true;  
end;
```

# Dateinamen-Konventionen

## Automatische Bereinigung

Alle Dateinamen werden automatisch mit der `SanitizeFileName`-Methode bereinigt:

- Ungültige Zeichen (`< > : " / \ | ? *`) werden durch Unterstriche ersetzt
- Leerzeichen werden durch Unterstriche ersetzt
- Maximale Dateinamenlängen werden erzwungen

## Eindeutige Benennung

- **Einzelne Dateien:** Enthalten Anhang-GUID-Präfix zur Eindeutigkeit
- **ZIP-Dateien:** Enthalten Datensatzinformationen und Zeitstempel
- **Keine Konflikte:** Garantiert eindeutige Namen innerhalb jedes ZIP-Archivs

---

Revision #1

Created 25 June 2025 05:41:00 by Bernd Feddersen

Updated 25 June 2025 05:48:30 by Bernd Feddersen