# Bearbeitungssperre: per Event eingrenzen oder verschärfen

Solange ein Workflow läuft, schränkt Advanced Workflow ein, wer den zu genehmigenden Datensatz und seine Zeilen bearbeiten darf: Der Genehmiger, der die Genehmigung übernommen hat, darf in einem Schritt mit **Datensatzbearbeitung erlauben** bearbeiten, alle anderen werden abgewiesen. Partner und Kunden können diese Sperre über ein Integration Event **eingrenzen** – eine Änderung erlauben, die Advanced Workflow ablehnt – oder **verschärfen** – eine Änderung ablehnen, die Advanced Workflow erlaubt.

Diese Seite beschreibt den Vertrag für Entwickler: das Event, seine Parameter, die Hilfsfunktionen und einen Beispiel-Subscriber.

# Typischer Anwendungsfall

Ein Schritt im Genehmigungsmodus **Nur Zugehörige** erstellt Genehmigungen für die Zeilen, aber keine für den Kopf. Niemand hält eine Kopfgenehmigung, also kann sie auch niemand übernehmen – der Kopf ist während des Schritts für alle gesperrt. Häufig sollen die Genehmiger der Zeilen trotzdem einzelne Kopffelder korrigieren dürfen, etwa **Ihre Referenz** einer Einkaufsbestellung, aber keine anderen. Genau das kann ein Subscriber erlauben.

# Das Event

Herausgeber: Codeunit `DXP AWF Access Control Mgmt.`, Integration Event `OnAfterCheckRecordChange`. Es wird bei jeder Änderung ausgelöst, die ein Benutzer im Client an einem vom Workflow gesperrten Datensatz vornimmt – am zu genehmigenden Datensatz, an einer Zeile mit eigenen zugehörigen Genehmigungen oder an einer Zeile, die die Sperre des Kopfes über **Für Zugriffskontrolle verwenden** erbt –, und zwar nachdem Advanced Workflow entschieden hat.

| Parameter | Bedeutung |
| --- | --- |

| | |
|---|---|
| `RecRef: RecordRef` | Der geänderte Datensatz mit seinen neuen Werten. |
| `xRecRef: RecordRef` | Die Werte des Datensatzes vor der Änderung. Beim Einfügen und Löschen enthält er dieselben Werte wie `RecRef`. |
| `ChangeType: Enum "DXP AWF Record Change Type"` | `Insert`, `Modify`, `Delete` oder `Rename`. |
| `ApprovalRecordID: RecordId` | Der zu genehmigende Datensatz, dessen Sperre gilt: der geänderte Datensatz selbst oder der Kopf, von dem eine Zeile sie erbt. |
| `WorkflowInstance: Record "DXP AWF Workflow Instance"` | Der Workflow, gegen den die Änderung geprüft wird – Vorlage, aktueller Schritt, Quelldatensatz. Leer, wenn keiner gefunden wurde. |
| `UserID: Code[50]` | Der Benutzer, der die Änderung vornimmt. |
| `var IsAllowed: Boolean` | Die Entscheidung von Advanced Workflow. Setzen Sie den Wert, um das Ergebnis zu ändern. |
| `var DeniedMessage: Text` | Der Fehler, den der Benutzer bei einer abgelehnten Änderung sieht. Advanced Workflow füllt ihn, wenn es ablehnt. Setzen Sie ihn, wenn Sie eine erlaubte Änderung ablehnen; bleibt er leer, erhält der Benutzer die Meldung **Sie können diesen Datensatz nicht ändern. Er befindet sich in der Genehmigung im Schritt %1.** |

# Hilfsfunktionen

Alle in der Codeunit `DXP AWF Access Control Mgmt.`:

- **`HasOnlyFieldsChanged(RecRef, xRecRef, AllowedFieldNos: List of [Integer]): Boolean`** – wahr, wenn sich kein anderes als die erlaubten Felder geändert hat.
- **`GetChangedFields(RecRef, xRecRef, var ChangedFieldNos: List of [Integer])`** – die Nummern der geänderten Felder. Nur normale Felder: FlowFields, FlowFilter und die Systemfelder werden nicht verglichen.
- **`CanUserModifyInWorkflow(InstanceID: Guid; UserID: Code[50]): Boolean`** – wahr, wenn der Benutzer eine offene Genehmigung des Workflows hält, die ihn bearbeiten lässt: eine eigene, von ihm übernommene, wo übernommen werden muss, in einem Schritt mit **Datensatzbearbeitung erlauben**. Kopf- und zugehörige Genehmigungen zählen gleichermaßen.
- **`CanUserChangeRecord(RecRef, xRecRef, ChangeType, ApprovalRecordID, UserID, var DeniedMessage): Boolean`** – die vollständige Prüfung einschließlich des Events. Verwenden Sie sie, wenn Ihr eigener Code dieselbe Frage stellen soll.

> Eine Tabelle, deren `OnModify`-Trigger eigene Felder setzt – zum Beispiel **Korrigiert am** bei Debitoren oder Kreditoren –, ändert diese Felder mit. Nehmen Sie sie in die erlaubten Felder

auf, sonst liefert `HasOnlyFieldsChanged` falsch.

# Beispiel: Zeilengenehmiger dürfen „Ihre Referenz“ ändern

```
codeunit 50110 "MY AWF Header Edit"
{
    [EventSubscriber(ObjectType::Codeunit, Codeunit::"DXP AWF Access Control Mgmt.",
'OnAfterCheckRecordChange', '', false, false)]
    local procedure AllowYourReferenceForLineApprovers(RecRef: RecordRef; xRecRef: RecordRef;
ChangeType: Enum "DXP AWF Record Change Type"; ApprovalRecordID: RecordId; WorkflowInstance:
Record "DXP AWF Workflow Instance"; UserID: Code[50]; var IsAllowed: Boolean; var
DeniedMessage: Text)
    var
        PurchaseHeader: Record "Purchase Header";
        AccessControlMgmt: Codeunit "DXP AWF Access Control Mgmt.";
        AllowedFieldNos: List of [Integer];
    begin
        if IsAllowed then
            exit;
        // Nur eine Änderung am Einkaufskopf selbst, und nur ein Ändern.
        if (RecRef.Number() <> Database::"Purchase Header") or (RecRef.RecordId() <>
ApprovalRecordID) then
            exit;
        if ChangeType <> ChangeType::Modify then
            exit;

        AllowedFieldNos.Add(PurchaseHeader.FieldNo("Your Reference"));
        if not AccessControlMgmt.HasOnlyFieldsChanged(RecRef, xRecRef, AllowedFieldNos) then
            exit;

        // Nur wer an diesem Workflow arbeitet - z. B. der Genehmiger einer Zeile, der sie
übernommen hat.
        if AccessControlMgmt.CanUserModifyInWorkflow(WorkflowInstance."Instance ID", UserID)
then
```

```
            IsAllowed := true;
    end;
}
```

# Beispiel: Sperre verschärfen

Eine Änderung ablehnen, die Advanced Workflow erlaubt – etwa: Den Kreditor einer Einkaufsbestellung darf während der Genehmigung niemand ändern, auch nicht der Genehmiger:

```
    [EventSubscriber(ObjectType::Codeunit, Codeunit::"DXP AWF Access Control Mgmt.",
'OnAfterCheckRecordChange', '', false, false)]
    local procedure KeepVendorDuringApproval(RecRef: RecordRef; xRecRef: RecordRef;
ChangeType: Enum "DXP AWF Record Change Type"; ApprovalRecordID: RecordId; WorkflowInstance:
Record "DXP AWF Workflow Instance"; UserID: Code[50]; var IsAllowed: Boolean; var
DeniedMessage: Text)
    var
        PurchaseHeader: Record "Purchase Header";
        AccessControlMgmt: Codeunit "DXP AWF Access Control Mgmt.";
        ChangedFieldNos: List of [Integer];
        VendorLockedErr: Label 'The vendor cannot be changed while the order is under
approval.';
    begin
        if (not IsAllowed) or (RecRef.Number() <> Database::"Purchase Header") then
            exit;
        AccessControlMgmt.GetChangedFields(RecRef, xRecRef, ChangedFieldNos);
        if ChangedFieldNos.Contains(PurchaseHeader.FieldNo("Buy-from Vendor No.")) then begin
            IsAllowed := false;
            DeniedMessage := VendorLockedErr;
        end;
    end;
```

# Was das Event nicht ändert

- **Buchen, Freigeben und Löschen** eines Belegs in der Genehmigung bleiben durch die Datensatzsperre von Business Central blockiert, unabhängig von der Entscheidung eines Subscribers.

- **Wer genehmigen, ablehnen, übernehmen oder delegieren darf**, wird hier nicht entschieden. Ein Subscriber kann niemanden zu einer Genehmigungsaktion berechtigen.
- Das Event wird nur ausgelöst, wo Advanced Workflow überhaupt sperrt: Die Tabelle gehört zum Regelsatz einer aktiven Vorlage, und der zu genehmigende Datensatz hat eine offene Genehmigung oder einen aktiven Workflow im Modus **Nur Zugehörige**. Datensätze außerhalb einer Genehmigung lassen sich damit nicht sperren.
- Es wird für Änderungen ausgelöst, die Benutzer auf Seiten vornehmen. Soll eigener Code die Sperre vor dem Schreiben beachten, rufen Sie `CanUserChangeRecord` auf.
- Ohne Subscriber ändert sich nichts.

# Verwandte Seiten

- **Schritte definieren** – Genehmigungsmodus und **Datensatzbearbeitung erlauben**
- **Zugehörige Tabellen konfigurieren (optional)** – **Für Zugriffskontrolle verwenden** und **Für zugehörige Genehmigungen verwenden**
- **Zugehörige Genehmigungen**

---

Revision #1
Created 2026-10-29 13:44:07 UTC by Bernd Feddersen
Updated 2026-10-29 13:44:07 UTC by Bernd Feddersen