Edit restriction: narrowing or tightening it with an event

While a workflow runs, Advanced Workflow restricts who may edit the record under approval and its lines: the claimed approver may edit at a stage with **Allow Record Edit**, everybody else is refused. Partners and customers can **narrow** that restriction — allow a change Advanced Workflow refuses — or **tighten** it — refuse a change it allows — with one integration event.

This page is the developer contract: the event, its parameters, the helper procedures, and an example subscriber.

Typical use

A stage in approval mode **Related Only** creates approvals for the lines and none for the header. Nobody holds a header approval, so nobody can claim one, and the header is locked for everybody while the stage runs. Often the approvers of the lines should still be able to correct a few header fields — for example **Your Reference** on a purchase order — but no others. That is exactly what a subscriber can allow.

The event

Publisher: codeunit `DXP AWF Access Control Mgmt.`, integration event `OnAfterCheckRecordChange`. It is raised for every change a user makes in the client to a record the workflow restricts — the record under approval, a line with Related Approvals of its own, or a line that inherits the header's restriction through **Use for Access Control** — after Advanced Workflow has decided.

Parameter	Meaning
`RecRef: RecordRef`	The record being changed, with its new values.
`xRecRef: RecordRef`	The record's values before the change. For an insert or a delete it holds the same values as `RecRef`.
`ChangeType: Enum "DXP AWF Record Change Type"`	`Insert`, `Modify`, `Delete` or `Rename`.
`ApprovalRecordID: RecordId`	The record under approval whose restriction applies: the changed record itself, or the header a line inherits it from.

`WorkflowInstance: Record "DXP AWF Workflow Instance"`	The workflow the change is checked against — template, current stage, source record. Empty if none could be found.
`UserID: Code[50]`	The user making the change.
`var IsAllowed: Boolean`	Advanced Workflow's decision. Set it to change the outcome.
`var DeniedMessage: Text`	The error the user sees when the change is refused. Advanced Workflow fills it when it refuses. Set it when you refuse a change it allowed; if you leave it empty, the user is told **You cannot modify this record. It is under approval at stage %1.**

Helper procedures

All in codeunit `DXP AWF Access Control Mgmt.`:

- **`HasOnlyFieldsChanged(RecRef, xRecRef, AllowedFieldNos: List of [Integer]): Boolean`** — true if no field but the allowed ones changed.
- **`GetChangedFields(RecRef, xRecRef, var ChangedFieldNos: List of [Integer])`** — the numbers of the changed fields. Normal fields only: FlowFields, FlowFilters and the system fields are not compared.
- **`CanUserModifyInWorkflow(InstanceID: Guid; UserID: Code[50]): Boolean`** — true if the user holds an open approval of the workflow that lets them edit: their own, claimed by them where it has to be claimed, at a stage that allows record edit. Header approvals and Related Approvals count alike.
- **`CanUserChangeRecord(RecRef, xRecRef, ChangeType, ApprovalRecordID, UserID, var DeniedMessage): Boolean`** — the complete check, including the event. Use it if your own code wants to ask the same question.

> A table whose `OnModify` trigger stamps fields of its own — **Last Date Modified** on a customer or vendor, for example — changes those fields as well. Add them to the allowed fields, or `HasOnlyFieldsChanged` returns false.

Example: line approvers may change "Your Reference"

```
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;
        // Only a change to the purchase header itself, and only a modification.
        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;

        // Only somebody who works on this workflow - e.g. the claimed approver of a line.
        if AccessControlMgmt.CanUserModifyInWorkflow(WorkflowInstance."Instance ID", UserID)
then
            IsAllowed := true;
    end;
}
```

Example: tightening

To refuse a change Advanced Workflow allows — for instance, nobody may change the vendor of a purchase order while it is under approval, the approver included:

```
    [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;
```

What the event does not change

- **Posting, releasing and deleting** a document under approval stay blocked by Business Central's own record restriction, whatever a subscriber decides.
- **Who may approve, reject, claim or delegate** is not decided here. A subscriber cannot let anybody act on an approval.
- The event fires only where Advanced Workflow restricts at all: the table belongs to the rule set of an active template, and the record under approval has an open approval or an active **Related Only** workflow. It cannot lock records that are not under approval.
- It is raised for changes users make on pages. If code of your own should respect the restriction before it writes, call `CanUserChangeRecord`.
- Without a subscriber, nothing changes.

Related pages

- **Define steps** — approval mode and **Allow Record Edit**

- **Configure related tables (optional)** — **Use for Access Control** and **Use for Related Approval**
- **Related approvals**

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