

Since we have defined Apex Sharing Reason on Custom Object sharing, it will keep Share table records updated whenever record owner is changed. So, still granted user can access the records without any issue. This scenario is totally different for standard object because Standard objects don't support Apex Sharing Reason. So, while sharing standard object records, by default you must define RowCause is "Manual".

It will impact the sharing table when record owner is changed. To solve this issue, we need to call apex sharing logic for both Insert as well as update function.

Example Apex Sharing Code for Standard Objects:

```
trigger Account_AT on Account(after insert,After Update) {
    // We only execute the trigger after a Account record has been inserted
    // because we need the Id of the AccountShare record to already exist.
    if(trigger.isInsert){
        // AccountShare is the "Share" table that was created when the
        // Organization Wide Default sharing setting was set to "Private".
        // Holding list of AccountShare records.
        List<AccountShare> accountList = new List<AccountShare>();

        // Sharing records to Chapter Staff.
        for(Account currRec : trigger.new)
        {
            if(trigger.isInsert || (trigger.isUpdate && currRec.OwnerId != trigger.oldMap.get(currRec.Id).OwnerId))
            {
                // Create a new AccountShare record to be inserted in to the AccountShare table.
                AccountShare accShare = new AccountShare();

                // Populate the AccountShare record with the ID of the record to be shared.
                accShare.ParentId = currRec.Id;

                // Specify the Chapter_Staff user id for sharing records.
                accShare.UserOrGroupId = currRec.Chapter_Staff__c;

                // Granting Edit Permission for the Chapter Staff.
                accShare.AccessLevel = 'edit';

                // Specify the mode of sharing.
                accShare.RowCause = Schema.AccountShare.RowCause.Manual;

                // Add the new Share record to the list of new Share records.
                accountList.add(accShare);
            }
        }
        // Insert all of the newly created Share records and capture save result
        Database.SaveResult[] accShareInsertResult = Database.insert(accountList,false);
    }
}
```

Record Sharing via Workflow Rule.

When you are sharing records to external system via Outbound Message, you should add condition like "OwnerId <> PriorValue(OwnerId)". So, whenever record owner is going to change, it will share the records to external system again.