Mass Update Access to Objects And Fields For Profiles And Permission Sets

If you need to update Object-level permissions (CRED) or Field level Permissions (FLS) for a large number of Objects, Fields, Profiles, or Permission Sets, rather than manually clicking dozens of checkboxes on multiple pages, it is sometimes easier and faster to make those updates using tools like Data Loader.

This article describes how to make those updates, as well as relevant information about the data model regarding FLS and CRED. It is intended for declarative developers.

- Object Permissions - Basic Functionality
- Field Permissions - Basic Functionality
- Query CRED And FLS Permissions - Examples
- Updating, Deleting, and Adding Permissions
- Important Notes

Object Permissions - Basic Functionality

When dealing with Profiles and CRED there are three objects involved:

- **Profile** object
- **PermissionSet** object
- **ObjectPermissions** object

Note: *Every Profile has a corresponding child PermissionSet record, as indicated by the ProfileId field on the PermissionSet record. When dealing with Permission Sets, the Profile object doesn't factor in.*

For every combination of Profile and Object, there is a corresponding ObjectPermissions record with six boolean fields that control the access level for that Profile to that object. The same goes for Permission Sets. Those six fields are:

- `PermissionsCreate`
- `PermissionsDelete`
- `PermissionsEdit`
- `PermissionsRead`
- `PermissionsViewAllRecords`
- `PermissionsModifyAllRecords`

Note: *If a Profile or Permission Set has no access to an object, then there is no ObjectPermissions record for that object/profile combination. You cannot have an ObjectPermissions record where all "permissions" fields are FALSE.*

In addition to these boolean fields, there are two other uneditable fields which indicate which object the record is related to (`sObjectType`), as well as the related Permission Set (`ParentId`). Remember, even if the ObjectPermissions record is controlling access for a Profile, it will be related to a Permission Set. That Permission Set will have the Id of the corresponding Profile in the `ProfileId` field.

When a Profile or Permission Set is granted access to an Object, Salesforce automatically creates a new ObjectPermissions record. When access to that Object is removed, Salesforce deletes that record.

Field Permissions - Basic Functionality

Field-Level Security works very similarly to Object-Level Permissions. When dealing with Profiles and FLS, there are three objects involved:

- **Profile** object
- **PermissionSet** object
- **FieldPermissions** object

Note: *Every Profile has a corresponding child PermissionSet record, as indicated by the ProfileId field on the PermissionSet record. When dealing with Permission Sets, the Profile object doesn't factor in.*

For every combination of Profile and Field, there is a corresponding FieldPermissions record. Each record has two boolean fields that control the access level for that Profile to that field. The same goes for Permission Sets. Those two fields are:

- `PermissionsEdit`
- `PermissionsRead`

Note: *If a Profile or Permission Set has no access to a Field, then there is no FieldPermissions record for that Field/Profile combination. You cannot have a FieldPermissions record where all "permissions" fields are FALSE.*

In addition to these boolean fields, there are three other uneditable fields which indicate which Object the record is related to (`sObjectType`), which specific Field this record controls access to (`Field`), and the related Permission Set (`ParentId`). Remember, even if the FieldPermissions record is controlling access for a Profile, it will be related to a Permission Set. That Permission Set will have the Id of the corresponding Profile in the `ProfileId` field.

When a Profile or Permission Set is granted access to a Field, Salesforce automatically creates a new FieldPermissions record. When access to that Field is removed, Salesforce deletes that record.

Query CRED And FLS Permissions - Examples

Query All Permissions

To get a list of every CRED setting for every Profile and Permission Set in Salesforce run the following query, or use Data Loader to export all ObjectPermissions records with the following fields:

```
SELECT Id, ParentId, Parent.ProfileId, Parent.Profile.Name, SobjectType, PermissionsCreate,
PermissionsDelete, PermissionsEdit, PermissionsRead, PermissionsViewAllRecords,
PermissionsModifyAllRecords
FROM ObjectPermissions
```

To query all Field permissions use a similar query:

```
SELECT Id, ParentId, Parent.ProfileId, Parent.Profile.Name, SobjectType, Field,
PermissionsEdit, PermissionsRead
FROM FieldPermissions
```

Query Permissions For Specific Profiles

In order to limit your search to specific profiles, add a filter to the end using the `Parent.ProfileId` field . Example:

```
SELECT Id, (...)
FROM (...)
WHERE Parent.Profile.Name = 'Sales Manager'
```

Or if you have a list of profiles:

```
SELECT Id, (...)
FROM (...)
WHERE Parent.Profile.Name IN ('Sales Manager', 'Sales', 'Marketing')
```

Query Permissions For Profiles Only (Or Permission Sets Only)

To limit your query to only see permissions related to Profiles and not Permission Sets, add a filter to the end using the `Parent.ProfileId` field to make sure it’s not empty:

```
SELECT Id, (...)
FROM (...)
WHERE Parent.ProfileId != null
```

Conversely, to limit your query to only show permissions related to Permission Sets, adjust the filter:

```
WHERE Parent.ProfileId = null
```

Query Permissions For Specific Objects

In order to limit the Objects you want permissions for, add a filter to the end using the `SobjectType` field. Example:

```
SELECT Id, (...)
FROM (...)
WHERE SobjectType IN ('Account','Opportunity','Contact')
```

Query Permissions For Specific Fields

In order to limit the Fields you want permissions for, add a filter to the end using the `Field` field. Note that the values in the `Field` field include API name of the Object, followed by a period, then the API name of the Field. Example:

```
SELECT Id, (...)
FROM (...)
WHERE Field IN ('Account.Customer__c','Opportunity.Total__c','Contact.LastName')
```

Updating, Deleting, and Adding Permissions

After running your query you will have a table describing access for all objects/fields **where at least one profile or permission set has some kind of access**. This is an important concept to understand. If no Profiles or Permission Sets have access to an Object or Field, there will not be a record for that object/field.

Updating Permissions

For existing ObjectPermissions/FieldPermissions records, you can make updates to the TRUE and FALSE values in each column, then use Data Loader to upload the changes using the **Update** feature.

Removing Permissions

To remove all access to an Object/Field, you will need to use the **Delete** feature in Data Loader to delete the appropriate ObjectPermissions/FieldPermissions records, using a list of Ids.

Adding Permissions

To add access where there is none, you will need to use the Insert feature in Data Loader to create new ObjectPermissions/FieldPermissions records.

Necessary Fields

ObjectPermissions

To data load ObjectPermissions records, include the following fields:

- `sObjectType`
- `ParentId`
- `PermissionsCreate`
- `PermissionsDelete`
- `PermissionsEdit`
- `PermissionsRead`
- `PermissionsViewAllRecords`
- `PermissionsModifyAllRecords`

FieldPermissions

To data load FieldPermissions records, include the following fields:

- `sObjectType`
- `Field`
- `ParentId`

- `PermissionsEdit`
- `PermissionsRead`

Important Notes

General

- Upserts are generally not recommended due to the extremely slow speed. It will most likely take much longer to make the upsert than it would to split the records into separate Insert and Update files.
- As stated above, you cannot have an ObjectPermissions or FieldPermissions record where all “permissions” fields are FALSE. If you try to update or insert one, you will get an error. Instead, to remove all access to an object, you have to delete the ObjectPermissions record.
- Custom Settings and Custom Metadata Types don’t have ObjectPermissions records related to them. Trying to insert or update them will just return an error.

Dependencies

- Watch out for permissions dependencies. When updating permission using the Profile edit page for example, Salesforce will automatically enable dependent permissions when needed. When data loading permissions, Salesforce will not automatically update user or system permissions on the profile if you try to update an object permission that has a dependency. Instead the update or insert will fail and you will get an error on that row. Accounts in particular have a large number of dependencies. Example:

```
FIELD_INTEGRITY_EXCEPTION: Permission Convert Leads depends on permission(s): Create Account; Permission Read All Asset depends on permission(s): Read All Account; Permission Read All Contract depends on permission(s): Read All Account; Permission Read All Dsx_Invoice__c depends on permission(s): Read All Account; Permission Read All Orders__c depends on permission(s): Read All Account; Permission Read All OrgChartPlus__ADP_OrgChartEntityCommon__c depends on permission(s): Read All Account; Permission Read All OrgChartPlus__ADP_OrgChart__c depends on permission(s): Read All Account; Permission Read All Partner_Keyword_Mapping__c depends on permission(s): Read All Account; Permission Read All Zuora__CustomerAccount__c depends on permission(s): Read All Account
```

- Additionally, keep in mind what is required at the Object level when setting certain permissions. For example, all levels of access (Edit, Create, etc..) require Read access. Delete access requires Read as well as Edit. Modify All requires all levels of access except Create. Salesforce will not allow you to data load permissions with illegal combinations of CRED access.

Modify All Data

- When using SOQL to query object permissions, be aware that some object permissions are enabled because a user permission requires them. The exception to this rule is when

“Modify All Data” is enabled on the Profile or Permission Set (*note: not to be confused with the "Modify All" CRED permission*). While it enables all object permissions, it doesn’t physically store any object permission records in the database. As a result, unlike object permissions that are required by a user permission - such as “View All Data” or “Import Leads” - the query still returns permission sets with “Modify All Data,” but the object permission record will contain an invalid ID that begins with “000”. This ID indicates that the profile has full access due to “Modify All Data” and the object permission record can’t be updated or deleted.

- **To remove full access from these objects, disable “Modify All Data” at the Profile level, and then delete the resulting object permission record.**

Resources

Object Permissions:

https://developer.salesforce.com/docs/atlas.en-us.api.meta/api/sforce_api_objects_objectpermissions.htm

Field Permissions:

https://developer.salesforce.com/docs/atlas.en-us.api.meta/api/sforce_api_objects_fieldpermissions.htm