Understanding the “Team” concept

There are two types of teams available in Dynamics CRM 2015/2016: **O*wner*** and ***Access.*** Both enable you to easily share business objects and collaborate with other users across business units. Which one you choose to use depends on your business need.

Each team belongs to one business unit, but teams can include users from other business units. And individual users can be associated with more than one team.

An Owner team owns records and is assigned security roles which define the team’s privileges. Team members have additional privileges, which are defined by their individual security roles for that team, and by roles from other teams of which they are members. A team has full access rights on the records that the team owns.

An Access team owns no records and has no security roles assigned to it. Team members have privileges defined by their individual security roles, and by roles from other teams of which they are members. Records are shared with an Access team, and the team is granted access rights on records such as Read, Write and Append.

Team functionality is supported by the **Team** entity and the **Team Template** entity. Only the **Team** entity is used for Owner teams and user-created Access teams. Both the **Team** entity and the **Team Template** entity are used for auto-created Access teams.

Access teams vs. Owner teams

The type of team you choose (Owner or Access) depends on your business goals, the nature of your project, or even the size of your organization. Here are a few guidelines that you can follow to help your decision.

Use Owner teams when:

* Your business policies require entities other than users to own records
* You know the number of teams at the time your CRM system is designed
* Daily reporting on progress by owning teams is required

Use Access teams when:

* Teams are dynamically formed and dissolved  
  *This typically happens when clear criteria, such as established territory, product or volume, aren’t used to define teams.*
* You don’t know the number of teams at the time your CRM system is designed
* Team members require different access rights on records  
  *You can share a record with several Access teams, with each team providing different access rights on the record. For example, one team is granted only the Read access right on the account, while another team is granted the Read, Write and Share access rights on the same account.*
* A unique set of users requires access to a single record without having ownership of the record

Access teams come in 2 major types

User-created Access teams

To create a user-created Access team, use the team entity, and set the **Team. TeamType** attribute to **Access**. For a list of **TeamType** values, refer to the **Team** entity metadata, which can be found in the your organization’s metadata. See preceding metadata browser information.

You can share multiple records with a user-created Access team. To share a record, use the **GrantAccessRequest**message. For user-created teams, the **Team.SystemManaged** attribute is **false**. For a list of the **Team.SystemManaged** values, refer to the **Team** entity metadata in your organization’s metadata. See preceding metadata browser information.

To add or remove team members, use **AddMembersTeamRequest** or **RemoveMembersTeamRequest**messages.

To provide team members with different access rights on the records, create several teams and grant each team a different set of access rights.

Auto-created Access teams

An auto-created, or system-managed, Access team is created for a specific record and can’t be shared with other records. For system-managed teams, you have to provide a team template. To create a template, use the team template entity. In the template, you have to specify the entity type and the access rights on the entity record, such as Read or Write, that are granted to team users when the team is created.

The entity you specify in the template must be enabled for auto-created Access teams. To provide team members with different access rights on the record, create several team templates.

Example: For the account entity, provide one template with the Read access right for the team that only needs to view the record. For the team that requires more access to the same record, provide a second template with Read, Write and Share access rights.

To enable an entity for auto-created Access teams, set the **[AutoCreateAccessTeams](https://msdn.microsoft.com/en-us/library/microsoft.xrm.sdk.metadata.entitymetadata.autocreateaccessteams.aspx)** attribute to **true**. When you add or remove users in a particular record using **AddUserToRecordTeamRequest** or  **RemoveUserFromRecordTeamRequest** messages, those users are automatically added or removed from the team.

An auto-created team is created when you add the first user to the record and the team ID is returned in **AccessTeamId**. The **Team.SystemManaged** attribute for an auto-created team is automatically set to **true**.

For a list of **Team.SystemManaged** values, refer to the **Team** entity metadata in the metadata for your organization. See the preceding metadata browser information.

The caller of the message must have Share privileges on the entity and access rights on the record that match the access rights provided in the template.

Example: If the template specifies Read access rights, the calling user must have Read access rights on the record. To be added to the team, users must have a minimum of Basic (User) Read access on the entity specified in the template.

Because of the parental relation between the team template and system-managed Access teams, when you delete a template, all teams associated with the template are deleted according to the cascading rules.

A few things worth noting:

* The maximum number of team templates you can create for an entity is specified in the **MaxAutoCreatedAccessTeamsPerEntity** deployment setting. The default value is 2.
* The maximum number of entities you can enable for auto-created Access teams is specified in the **MaxEntitiesEnabledForAutoCreatedAccessTeams** deployment setting. The default value is 5.
* If you change access rights for the team template, the changes are only applied to new auto-created Access teams. Existing teams are not affected.

Here’s a handy team reference guide

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Team type** | **When to use it** | **What entity to use** | **User team template?** | **SDK message for adding or removing team members** | **Owns records?** | **Records owned or has access to** | **Are security roles assigned?** |
| **Owner-created** | Record ownership by the team is required  The number of teams is known at design time | Team | No | AddMembersTeamRequest  RemoveMembersTeamRequest | Yes | Can own multiple records | Yes |
| **Access:**  **User-created** | Multiple records have to be shared with the team  The number of teams isn’t known at design time  Team members require different access rights on the records | Team | No | AddMembersTeamRequest  RemoveMembersTeamRequest | No | Can access multiple records | No  Provides access rights on the records |
| **Access:**  **Auto- created** | Unique set of users works on a single record  Team members require different access rights on the records  Creating teams automatically per record is desirable | Team Template  Team | Yes | AddUserToRecordTeamRequest  RemoveUserFromRecordTeamRequest | No | Can access only one record | No  Provides access rights on the records |

How to create an Access team

Enable an entity

1. Make sure you have the System Administrator security role or equivalent permissions in Dynamics CRM
2. Go to **Settings** > **Customizations**
3. In the **Customization** window, choose **Customize the System**
4. In the navigation pane, expand **Entities**, and then choose the entity you want to use in the team template
5. On the **Entity Definition** form, in the **Communication & Collaboration** section, select **Access Teams**
6. On the **Actions** toolbar, choose **Save**

Create a team template

1. Go to **Settings** > **Security** and choose **Access Team Templates**
2. On the **Actions** toolbar, choose **New**, complete the required fields, and then **Save**

Add a team template to the entity form

1. Make sure you have the System Administrator security role or equivalent permissions in Dynamics CRM
2. Go to **Settings** > **Customizations**
3. In the **Customization** window, choose **Customize the System**
4. In the navigation pane, expand **Entities**, expand the entity you want to use in the team template, and then choose **Forms**
5. In **System Forms**, select **Active Forms** > **Main** form
6. On the **Main** form, open the **Insert** tab
7. On the ribbon, choose **Sub-Grid** (**Set Properties** dialog box appears)
8. In **Set Properties**, complete the required fields, and then select **Display label on the Form**
9. In the **Records** drop-down list, select **All Record Types**
10. In the **Entity** drop-down list, select **Users**
11. In the **Default View** drop-down list, select **Associated Record Team Members**
12. In the **Team Template** drop-down list, select the desired template and choose **Set t**he team template you selected now appears on the **Main** form.
13. On the **Actions** toolbar, click or tap **Save**, and then choose **Publish**

Adding/removing users to/from system-managed Access teams

In order to add/remove users to/from system-managed access team, you need:

* Team template name to get a templateID
* User ID in order to add to team
* Record ID on which to add the users

To **Add** users to a system-managed Access team, use **AddUserToRecordTeamRequest**sdk message.  
For more information about this sdk message, please visit **[AddUserToRecordTeamRequest](https://msdn.microsoft.com/en-us/library/microsoft.crm.sdk.messages.addusertorecordteamrequest.aspx)**

To **Remove** users from a system-managed Access team, use **RemoveUserFromRecordTeamRequest**sdk message. For more information about this sdk message, please visit **[RemoveUserFromRecordTeamRequest](https://msdn.microsoft.com/en-us/library/microsoft.crm.sdk.messages.removeuserfromrecordteamrequest.aspx)**

Here’s the code snippet for adding users to an Access team

static void Main(string[] args)

{

xRMDataConnector connector = new xRMDataConnector();

IOrganizationService service = connector.Connect();

//check connection is established or not...

WhoAmIRequest whoAmIrequest = new WhoAmIRequest();

WhoAmIResponse whoAmIresponse = (WhoAmIResponse)service.Execute(whoAmIrequest);

//Assume that Access team is enable on contact entity and respective team template created for contact entity.

//First Create Contact Record

Entity contact = new Entity("contact");

contact["firstname"] = "John";

contact["lastname"] = "Doe";

contact["emailaddress1"] = "john.doe@gmail.com";

//creating contact record and storing GUID to contact Entity object.

contact.Id = service.Create(contact);

//Time to retrieve Team Teamplate Entity Reference by Name.

EntityReference accessTeamEntityReference = GetTeamTemplateEntityReferencebyName(service, "Contact Team Template with R-W-A-AT Permission");

if (accessTeamEntityReference is EntityReference)

{

//Time to add current user record to team...

AddUserToRecordTeamRequest teamAddRequest = new AddUserToRecordTeamRequest();

teamAddRequest.Record = contact.ToEntityReference();

teamAddRequest.SystemUserId = whoAmIresponse.UserId;

teamAddRequest.TeamTemplateId = accessTeamEntityReference.Id;

AddUserToRecordTeamResponse teamAddResponse = (AddUserToRecordTeamResponse)service.Execute(teamAddRequest);

}

}

///

/// Retrieve Team Template EntityReference by Name

///

/// Object of IOrganization Service /// Name of Team Teamplate /// Returns EntityReference object of Team Template private static EntityReference GetTeamTemplateEntityReferencebyName(IOrganizationService service, string teamTemplateName) { // Query using ConditionExpression and FilterExpression ConditionExpression condition = new ConditionExpression(); //attribute name add to condition condition.AttributeName = “teamtemplatename”; //operator add to condition condition.Operator = ConditionOperator.Equal; //values added to condition condition.Values.Add(teamTemplateName); // filter creation FilterExpression filter = new FilterExpression(); //condition added filter.Conditions.Add(condition); //create query expression QueryExpression query = new QueryExpression(“teamtemplate”); //filter added to query query.Criteria.AddFilter(filter); //retrieve all columns query.ColumnSet = new ColumnSet(“teamtemplatename”); // execute query which will retrieve the Access team teamplate EntityCollection accessTeamEntityCollection = service.RetrieveMultiple(query); if (accessTeamEntityCollection is EntityCollection && accessTeamEntityCollection.Entities.Count > 0) return accessTeamEntityCollection.Entities[0].ToEntityReference(); else return null; }

https://www.neudesic.com/blog/access-team-in-crm-20152016/