**Create and edit entity relationships:**

[**Types of entity relationships**](javascript:void(0))

When you look at the solution explorer you might think that there are three types of entity relationships. Actually there are only two, as shown in the following table.

|  |  |
| --- | --- |
| **Relationship Type** | **Description** |
| **1:N (One-to-Many)** | An entity relationship where one entity record for the **Primary Entity** can be associated to many other **Related Entity** records because of a lookup field on the related entity.  When viewing a primary entity record you can see a list of the related entity records that are associated with it. |
| **N:N (Many-to-Many)** | An entity relationship that depends on a special **Relationship Entity**, sometimes called an Intersect entity, so that many records of one entity can be related to many records of another entity.  When viewing records of either entity in a N:N relationship you can see a list of any records of the other entity that are related to it. |

The **N:1 (many-to-one)** relationship type exists in the solution explorer user interface because the solution explorer shows you a view grouped by entities. 1:N relationships actually exist between entities and refer to each entity as either a **Primary Entity** or **Related Entity**. The related entity, sometimes called the child entity, has a lookup field that allows storing a reference to a record from the primary entity, sometimes called the parent entity. A N:1 relationship is just a 1:N relationship viewed from the related entity.

### [Create or edit 1:N relationships between entities](javascript:void(0))

1. Go to **Settings**>**Customizations**.
2. Click **Customize the System**.
3. Under **Components**, expand **Entities**, and then expand the entity you want to work with.
4. Click **1: N Relationships**.
5. To edit a relationship or view the details for a relationship, select the relationship, and on the Actions toolbar, click **More Actions**, and then click **Edit**.

- OR -

To add a new relationship, click **new 1-to-Many Relationship**.

|  |
| --- |
| **System_CAPS_importantImportant** |
| If **New 1-to-Many Relationship** does not appear on the Actions toolbar, you cannot create a 1: N relationship for that entity. |

1. For a new relationship, in the **Relationship Definition** section, in the **Related Entity** list, select the entity to be related.

|  |
| --- |
| **System_CAPS_noteNote** |
| Specifying the related entity sets a default value for the **Name** field. If you change the related entity before you save, the value of the **Name** changes accordingly. |

1. Select whether this will be searchable to not.
2. In the **Lookup Field** section, specify a value for the **Display Name** field.

|  |
| --- |
| **System_CAPS_importantImportant** |
| Specifying the **Display Name** sets a default value for the **Name** field. If you change the **Display Name** of the lookup field before you save, the value in the **Name** field will not change. As a result, be sure the **Name** is meaningful before saving. |

1. In the **Field Requirement** list, choose an option to specify data requirements for the field prior to saving a record.
2. In the **Navigation Pane Item for Primary Entity** section, in the **Display Option** list, choose an option for displaying associated views or a custom label.
3. In the **Relationship Behavior** section, in the **Type of Behavior** list, choose one of the following options:
   * **Parental**. In a parental relationship between two entities, any action taken on a record of the parent entity is also taken on any child entity records that are related to the primary (or parent) entity record.
   * **Referential**. In a referential relationship between two entities, you can navigate to any related records, but actions taken on one will not affect the other.
   * **Referential, Restrict Delete**. In a referential, restrict delete relationship between two entities, you can navigate to any related records. Actions taken on the parent record will not be applied to the child record, but the parent record cannot be deleted while the child record exists. Note that you cannot delete a record when related records exist.
   * **Configurable Cascading**. In a configurable cascading relationship between two entities, you select the behavior associated with each of a set of possible actions.

|  |
| --- |
| **System_CAPS_importantImportant** |
| If you set the behaviors for the actions so that they match the behaviors for the actions associated with another **Type of Behavior**, when you save the relationship, the **Type of Behavior** is automatically set to the matching type. |

1. More information: [MSDN: Configure entity relationship behavior](https://msdn.microsoft.com/library/gg309412.aspx#BKMK_ConfigureEntityRelationshipBehavior)
2. Click **Save and Close** to close the **Relationship** form.
3. When your customizations are complete, publish them:
   * To publish customizations for only the component that you are currently editing, on the Actions toolbar, click **Publish**.
   * To publish customizations for all unpublished components at one time, on the nav bar or in the Navigation Pane, click **Entities**, and then on the Actions toolbar, click **Publish All Customizations**.

|  |
| --- |
| **System_CAPS_noteNote** |
| * A custom entity cannot be the primary entity in a relationship with a related system entity that cascades. This means you cannot have a relationship with any action set to Cascade All, Cascade Active, or Cascade User-Owned between a primary custom entity and a related system entity. * No new relationship can have any action set to **Cascade All**, **Cascade Active**, or **Cascade User-Owned** if the related entity in that relationship already exists as a related entity in another relationship that has any action set to **Cascade All**, **Cascade Active**, or **Cascade User-Owned**. This prevents relationships that create a multi-parent relationship. * Any time you change user-interface elements or implement form scripts for an entity, you need to publish changes to apply them. Any customizations that change the data schema of Microsoft Dynamics CRM, such as custom entities, relationships, or fields are applied immediately. * If a relationship is part of a managed solution, the developer of the managed solution can restrict you from customizing the relationship. * Installing a solution or publishing customizations can interfere with normal system operation. We recommend that you schedule a solution import when it’s least disruptive to users. |

### [Relationship definition](javascript:void(0))

Depending on whether you chose to create a **new 1-to-Many Relationship** or a **New Many-to-1 Relationship** from the solution explorer, either the **Primary Entity** or **Related Entity** fields will be pre-populated. You only need to choose the other one. A default value for the **Name** field is pre-populated based on the solution publisher’s customization prefix and the names of the entities you choose to participate in the relationship. You can edit this if you want. If you create more than one custom relationship between two entities and use the same customization prefix for both, the auto-generated name value will not be unique and you will not be able to save the new relationship. You must edit the name to differentiate it from any existing name before you can save it. Once saved, you cannot change it.

If you don’t want to have this entity relationship visible in **Advanced Find**, set the **Searchable** value to **No**.

### [Lookup fields](javascript:void(0))

These fields are the common properties all fields have except **Searchable**, **Field Security**, and **Auditing**. To edit these values for the lookup field that is created with the entity relationship, you must locate and edit the lookup field separately after you create the entity relationship. More information: [Create and edit fields](https://technet.microsoft.com/en-us/library/dn531187.aspx#BKMK_CreatingAndEditngFields)

As a general rule, the **Display Name** should correspond to the primary entity display name.

### [Navigation pane item for primary entity](javascript:void(0))

The primary entity can reveal lists of related entities if you expand the navigation pane. The options in this group control how or whether to display this list. These navigation items can also be edited using the form editor and, by using JavaScript, a developer can apply changes to these items when the form is displayed.

|  |  |
| --- | --- |
| **Field** | **Description** |
| **Display Option** | * **Do Not Display**: Choose this if you do not want to allow people to be able to navigate to a list of related entity records. * **Use Custom Label**: Choose this if you want to specify a custom label to use. * **Use Plural Name**: Choose this if you want to use the plural name of the related entity as the label. |
| **Custom Label** | When you select **Use Custom Label** as the display option, enter the custom label you want to use instead of the related entity plural name. |
| **Display Area** | * **Details**: Choose this to include the navigation item in the **Common** group. * **Marketing**: Choose this to include the navigation item in the **Marketing** group. * **Sales**: Choose this to include the navigation item in the **Sales** group. * **Service**: Choose this to include the navigation item in the **Service** group. |
| **Display Order** | This number controls where the navigation item will be included within the selected display area. The range of allowed numbers begins with 10,000. Navigation pane items with a lower value appear above other relationships with a higher value. |

### [Relationship behavior](javascript:void(0))

In a 1: N relationship, you can control how the relationship behaves to support business rules for your organization. Why would you want to do this? Let’s look at an example.

Let’s say that you have a new salesperson and you want to assign them a number of existing opportunities currently assigned to another salesperson. Each opportunity record may have a number of task activities associated with it. You can easily locate the active opportunities you want to reassign and assign them to the new salesperson. But what should happen for any of the task activities that are associated with the opportunities? Do you want to open each task and decide whether they should also be assigned to the new salesperson? Probably not. Instead, you can let the relationship apply some standard rules for you automatically. These rules only apply to task records associated to the opportunities you are reassigning. The entity relationship is named **Opportunity\_Tasks**. Your options are:

* Reassign all active tasks.
* Reassign all tasks. This is the default behavior.
* Reassign none of the tasks.
* Reassign all tasks currently assigned to the former owner of the opportunity.

The relationship can control how actions performed on a record for the primary entity record cascade down to any related entity records. The actions and possible behaviors are shown in the following table.

|  |  |  |
| --- | --- | --- |
| **Action** | **Description** | **Possible behaviors** |
| **Assign** | What should happen when the primary entity record changes ownership? | * Cascade Active * Cascade All * Cascade None * Cascade User Owned |
| **Share** | What should happen when the primary entity record is shared? | * Cascade Active * Cascade All * Cascade None * Cascade User Owned |
| **Unshare** | What should happen when sharing of the primary entity record stops? | * Cascade Active * Cascade All * Cascade None * Cascade User Owned |
| **Reparent** | What should happen when a lookup field value for a parental type relationship in the primary entity record is changed?  A parental type relationship is one that uses **Cascade All** for all actions. [Customizable parental entity relationships](https://technet.microsoft.com/en-us/library/dn531061.aspx) lists the customizable parental system relationships. | * Cascade Active * Cascade All * Cascade None * Cascade User Owned |
| **Delete** | What should happen when the primary entity record is deleted? | * Cascade All * Remove Link * Restrict Delete |
| **Merge** | What should happen when the primary entity record is merged with another record? | * Cascade All * Cascade None |

Each of these actions can be configured to control how actions cascade down to records related to the primary entity record through the 1:N entity relationship. The behavior options are in the following table.

|  |  |
| --- | --- |
| **Behavior** | **Description** |
| **Cascade Active** | Perform the action on all active related entity records. |
| **Cascade All** | Perform the action on all related entity records. |
| **Cascade None** | Do nothing. |
| **Remove Link** | Remove the value of the lookup field for all related entity records. |
| **Restrict Delete** | Prevent the primary entity record from being deleted when related records exist. |
| **Cascade User Owned** | Perform the action on all related entity records owned by the same user as the primary entity record. |

How these actions are applied within a relationship can be categorized or applied using the **Type of Behavior** field values described in the following table.

|  |  |
| --- | --- |
| **Field value** | **Description** |
| **Parental** | All actions use the **Cascade All** behavior.  [Customizable parental entity relationships](https://technet.microsoft.com/en-us/library/dn531061.aspx) lists all of the customizable system entity relationships that use the parental behavior. |
| **Referential** | **Assign**, **Share**, **Unshare**, and **Reparent** use the **Cascade None** behavior.  **Delete** uses the **Remove Link** behavior.  **Merge** uses the **Cascade All** behavior. |
| **Referential, Restrict Delete** | The same as **Referential**, except that **Delete** uses the **Restrict Delete** behavior. |
| **Configurable Cascading** | Individual behaviors can be assigned for each action. If the choices match any of the other **Type of Behavior** categories, the value will change to that **Type of Behavior** value. |

### [Limitations on behaviors you can set](javascript:void(0))

There are some limitations you should keep in mind when you define entity relationships.

* A custom entity can’t be the primary entity in a relationship with a related system entity that cascades. This means you can’t have a relationship with any action set to **Cascade All**, **Cascade Active**, or **Cascade User-Owned** between a primary custom entity and a related system entity.
* No new relationship can have any action set to **Cascade All**, **Cascade Active**, or **Cascade User-Owned** if the related entity in that relationship already exists as a related entity in another relationship that has any action set to **Cascade All**, **Cascade Active**, or **Cascade User-Owned**. This prevents relationships that create a multi-parent relationship.

## [Map entity fields](javascript:void(0))

You can map attributes between entities that have an entity relationship. This lets you set default values for a record that is created in the context of another record. Let’s say that you want to add a new contact record for a person who is an employee for a specific account. You can do this in two different ways:

You could just navigate to **Sales**>**Contacts** and create a new contact record from scratch. But then you need to set the parent account and enter several items of information (such as address and phone information) which are probably the same as the parent account. This can be time consuming and introduces opportunities for errors.

The easier way is to start with the account entity and, using the **Contacts** sub grid on the form, click **+** to add a contact. It will first guide you to look up any existing related contacts so you don’t accidentally create a duplicate record. If you don’t find an existing record, you can click **New** and create a new contact record. The difference is that certain items of data from the account record will be copied into the new contact form to set certain default values that you can edit before saving. This can save a lot of time when you are entering data, and help reduce errors.

[Default entity and attribute mappings](https://technet.microsoft.com/en-us/library/dn531076.aspx) shows all the default mappings set for Microsoft Dynamics CRM.

|  |
| --- |
| **System_CAPS_noteNote** |
| These mappings aren’t applied to related records created using a workflow or dialog process. They aren’t automatically applied to new records created using code, although developers can use a special message called [MSDN: InitializeFromRequest](https://msdn.microsoft.com/library/microsoft.crm.sdk.messages.initializefromrequest.aspx) to create a new record using available mappings.  These mappings only set default values to a record before it is saved. People can edit the values before saving. The data that is transferred is the data at that point in time. It isn’t synchronized. If the information in the primary entity record changes, the related entity record data that was transferred when it was created won’t change. |

The default values set when you create a new record from a list aren’t actually defined within the entity relationships, but they are exposed in the relationship user interface. Not every 1:N entity relationship has them. When you view a list of 1: N (or N: 1) entity relationships for an entity, you can filter the relationships shown by type. You can select either **All**, **Custom**, **Customizable**, or **Mappable**. Mappable entity relationships provide access to allow mapping entity fields.

The following rules show what kinds of data can be mapped.

* Both fields must be of the same type and the same format.
* The length of the target field must be equal to or greater than the length of the source field.
* The target field can’t be mapped to another field already.
* The source field must be visible on the form.
* The target field must be a field that a user can enter data into.
* If the fields are option sets, the integer values for each option should be identical.
* Address ID values can’t be mapped.

|  |
| --- |
| **System_CAPS_noteNote** |
| If you need to map option set fields, we recommend you configure both fields to use the same global option set. Otherwise, it can be difficult to keep two separate sets of options synchronized manually. If the integer values for each option aren’t mapped correctly you can introduce problems in your data. More information: [Create and edit global option sets](https://technet.microsoft.com/en-us/library/dn531201.aspx) |

### Create or edit mapping between fields

1. Go to **Settings**>**Customizations**.
2. Click **Customize the System**.
3. Under **Components**, expand **Entities**, and then expand the entity you want.
4. Click either **1: N Relationships** or **N: 1 Relationships**.
5. In the main pane, in the **Type** list, select **Mappable**.
6. Select a mappable relationship. Then, on the Actions toolbar, click **Actions**, and then click **Edit**.
7. Under **Related**, click **Mappings**.
8. For each new mapping, on the **Actions** toolbar, click **New**.
9. In the **Create Field Mapping** dialog box, select the source field from **Source Entity Fields**. Select the target field from **Target Entity Fields**.
10. Click **OK**.
11. Click **Save and Close** to close the **Relationship** form.
12. When your customizations are complete, publish them

|  |
| --- |
| **System_CAPS_noteNote** |
| * After publishing customizations, these mappings are available for all users. If you reset Internet Information Services (IIS) before you publish customizations, these mappings are available for all users, even though other customizations won’t be available. * If you map to or from a field that isn’t displayed on a form, the mapping won't be done until the field is added to a form. |

### [Automatically generate field mappings](javascript:void(0))

You can also generate mappings automatically but you should use care when doing this with system entities. Use this when you create custom entities and want to leverage mapping. When viewing the list of mappings, in the **More Actions** menu select **Generate Mappings**. This removes any existing mappings and replaces them with suggested mappings that are based only on the fields that have similar names and data types. If you use this on a system entity, you could lose some expected mappings. For custom entities, it helps save time because you can more easily delete any mappings you don’t want and add any others that the generate mappings action didn’t create.

## [Create and edit N:N (many-to-many) relationships](javascript:void(0))

1:N entity relationships establish a hierarchy between records. With N:N (many-to-many) relationships there is no explicit hierarchy. There are no lookup fields or behaviors to configure. Records created using N: N relationships can be considered peers and the relationship is reciprocal.

With N: N relationships a special entity is created called a Relationship (or Intersect) entity. This entity has a relationship with each of the related entities and only stores the necessary values to define the relationship. You can’t add custom fields to a relationship entity.

The procedure to create a N: N relationship is essentially choosing the two entities that you want to participate in the relationship, and then for each entity defining how you want the respective lists to be available within the navigation pane of the form for each entity. These are the same options used for the primary entity in 1: N entity relationships. More information: [Navigation pane item for primary entity](https://technet.microsoft.com/library/dn531171.aspx#BKMK_NavigationPaneOptions)

Not all entities can be used with N: N relationships. If the **New Many-to-Many Relationship** button isn’t present, you can’t create a new N: N relationship with this entity. If you use the metadata browser, you can filter on entities that have the **CanBeInManyToMany** value set to **true**. More information: [Use the metadata browser](https://technet.microsoft.com/en-us/library/dn531137.aspx#BKMK_MetadataBrowser)

### [Create or edit N-N relationships between entities](javascript:void(0))

1. Go to **Settings**>**Customizations**.
2. Click **Customize the System**.
3. Under **Components**, expand **Entities**, and then expand the entity you want to work with.
4. Click **N: N Relationships**.
5. To edit or view the details for an existing relationship, select the relationship, on the Actions toolbar, click **Actions**, and then click **Edit**.

- OR -

To add a new relationship, click **New Many-to-Many Relationship**.

|  |
| --- |
| **System_CAPS_importantImportant** |
| If **New Many-to-Many Relationship** does not appear on the Actions toolbar, you cannot create a N: N relationship for that entity. |

1. For a new relationship, in the **Current Entity** section, in the **Display Option** list, choose one of the following options:
   * **Do Not Display**: The other entity will not display an associated view for the current entity.
   * **Use Custom Label**: This label will be used for the associated view created for the other entity. Be sure to enter a corresponding value in the **Custom Label** field.
   * **Use Plural Name**: This will use the plural name of the current entity for the associated view.

|  |
| --- |
| **System_CAPS_noteNote** |
| When the **Use Plural Name** or **Use Custom Label** options are selected, you can choose from the **Display Area** option list to specify the display area (for example Marketing or Sales) on the form where the relationship label will be displayed. You can also specify the **Display Order** to control where the label will be included within the selected display area. |
| **System_CAPS_importantImportant** |
| The navigation paradigm for the forms associated with updated entities is significantly different than that for entities that have not been updated. While the mechanics for defining the Display Area and Display Order are common, be sure you understand the various navigation paradigms as you establish entity relationships. |

1. In the **Other Entity** section, select the other entity from the **Entity Name** list.

|  |
| --- |
| **System_CAPS_noteNote** |
| When you specify the entity name, default values are set for the **Name** and **Relationship Entity Name** fields in the **Relationship Definition** section. If you change the **Entity Name** value before you save, these names will not change, so be sure these names are meaningful before saving. |

1. In the **Relationship Definition** section, confirm the **Name** and the **Relationship Entity Name**.

These values must be unique among N: N relationships.

1. Click **Save and Close** to close the N: N Relationship form.
2. When your customizations are complete, publish them:
   * To publish customizations for only the component that you’re currently editing, on the **Home** tab, in the **Save** group, click **Publish**.
   * To publish customizations for all unpublished components at one time, click **Publish All Customizations**.

|  |
| --- |
| **System_CAPS_noteNote** |
| Any time you change user-interface elements or implement form scripts for an entity, you must publish changes to apply them. Any customizations that change the data schema of Microsoft Dynamics CRM such as custom entities, relationships, or fields, are applied immediately.  Installing a solution or publishing customizations can interfere with normal system operation. We recommend that you schedule a solution import when it’s least disruptive to users. |