Field mapping in relationships:

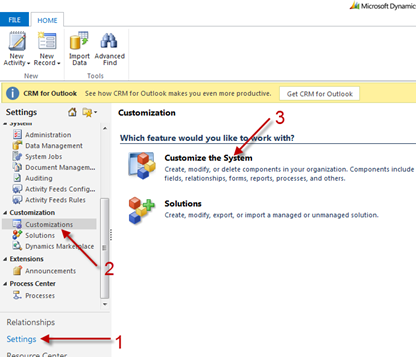
Save Time Entering Data into CRM by Mapping Fields between Records

Been said that a database is only as good at the data in it. A good way to insure good data is to reduce the amount of data entry by your users and to insure the correct data is entered. A way to accomplish this is through the mapping of fields from parent records to related, child records.

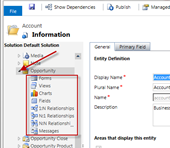
The key to making the mapping work is that the users need to create the new, child record from the parent record. For example, if you have an Opportunity record open, you would then select Quote from the left navigation and then click the ‘Add New Quote’ button from the ribbon when it changes.

To setup the mapping, do the following:

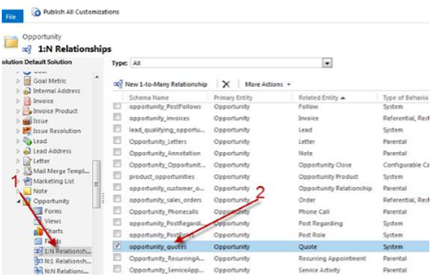
1. Get to ‘Customizations’.  One way to do this is to go to ‘Settings’, ‘Customizations’, and then choose ‘Customize the System’.



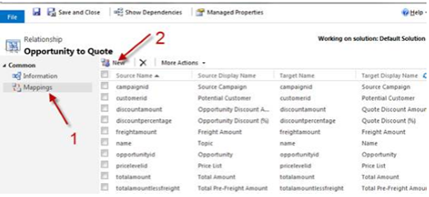
1. Select arrow to the left of the parent entity to expose the sub-menu items. In our example, that is the Opportunity.



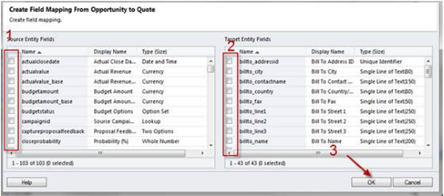
1. Once there, go to the Opportunity entity and click on the ‘1:N Relationships’.



1. Next click on ‘Mappings’ and then the ‘New’ button.



1. Lastly, choose the source and target fields, click ‘OK’ and you’re done.



An important note here is that the field types must be same on both entities if you want to map fields!

As long as the entities are related then it is fine.

For example there are 2 entities:

Account and Project

Then, you have two identical fields:

Industry in both entities with same global option set.

And also

Which you want to mapping those fields. If you create a project from relates Account Classification or Account level field and address field that you want to inherit

Then once you are in the account form then you go to Project sub grid or associated view then you create a new Project record from that Account

Sample data:

Account

Name = Bank of England

Industry = Banking

Level = Platinum Class

Address = St Louis 23, Newcastle

Then you have a new Project MS Dynamics CRM implementation in that bank you want to make sure that once you or users create the new project will be by default those fields are auto populated.

So you need to define those fields in the mapping in your 1 to many Account and Project relationship.

The things that you need to make note are:

- You only can do mapping from the related entities, this is the main concept

- Mapping is only works from Parent to Child, cannot from child to parent

- Mapping only works for the fields that are having same types and if you want to map the option set field you must define as global option set, you cannot inherit lookup to option set and vice versa

- it will work if only you create the child record from the parent, through sub grid or associated view, it won’t work if you create the project outside the parent from blank form then you later key in account field, it won’t inherit the fields mapping define before.

As explained before, if you create the record and then choose the lookup to get related parent record, the mapping won’t work because it does mean the record was created from outside not from parent context.

Basically, you can only create record from sub grid or associated view in the Parent form to get the mapping works.

## [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_note**Note |
| 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_note**Note |
| 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_note**Note |
| * 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_important**Important |
| 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_note**Note |
| 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_important**Important |
| 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_note**Note |
| 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_note**Note |
| 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. |