Access Teams with Microsoft Dynamics CRM 2013

Version: 1.0

Author: Roger Gilchrist
Company: Microsoft Corporation

Released: November 2013

Microsoft Dynamics CRM

Copyright

Feedback

To send comments or suggestions about this document, please click the following link and type your feedback in the message body: http://go.microsoft.com/fwlink/?LinkID=335812

Important: The subject-line information is used to route your feedback. If you remove or modify the subject line, we may be unable to process your feedback.

Table of Contents

Team selling

With the release of Microsoft Dynamics CRM 2013, we've added capabilities to enhance the management of teams who support particular customers or sets of records.

This capability can be used for managing teams of users supporting other types of records but this article will describe the use in the context of sales.

The key capabilities we're introducing are:

- Access teams: Lightweight teams aimed at high-volume sharing scenarios
- Automated creation and management of access teams

In this article, we'll describe:

- The key challenges faced by customers that these capabilities are intended to address
- How the capabilities can be enabled and used
- How the capabilities work
- Design considerations for using these capabilities

Key challenges

While Microsoft Dynamics CRM offers a rich set of capabilities for modelling security, we're aware of some challenges that can be faced in certain scenarios.

Managing teams that interact with customers

When deciding who can access information, there are scenarios where the users who can view or act on a customer record have to be specified. This can be the case, for example, when managing high worth individuals where a specific team is set up to manage that customer. Due to the sensitivity of the interaction, such as when interacting with a celebrity, it's important that only the necessary people can view that customer's information.

Microsoft Dynamics CRM allows setting up of teams and granting access directly to particular records. There are cases when this is done so rapidly or in large volumes that setting up and managing this sort of team access can be onerous in terms of administration, or require complex customization to automate.

Large volumes of teams and team memberships

Another challenge with this model can occur with larger implementations, such as where the number of customers grows into the millions, with each customer needing individual security and a team set up for each.

Experience from implementations supporting these scenarios has indicated that this can reach into millions of teams and individuals being members of tens of thousands of teams.

With Microsoft Dynamics CRM 2011, the introduction of team ownership and team security roles has provided a significant increase in the flexibility of usage, but it also comes with the additional processing required to check access based on each of these teams and their security roles.

To avoid this affecting performance and scalability, the security privileges a user is granted either directly, or through the security roles of a team they are a member of, are cached on first access. With very large, for example, greater than 1000, numbers of teams, the performance impact of retrieving the details of all of these teams when a user first accesses the system can become a challenge, particularly when an implementation doesn’t use team security roles.

Where there are regular changes to the teams that a user is a member of, the constant updating of their privileges in the cache can also have both a performance and scalability impact as the cached privileges are reloaded and calculated.

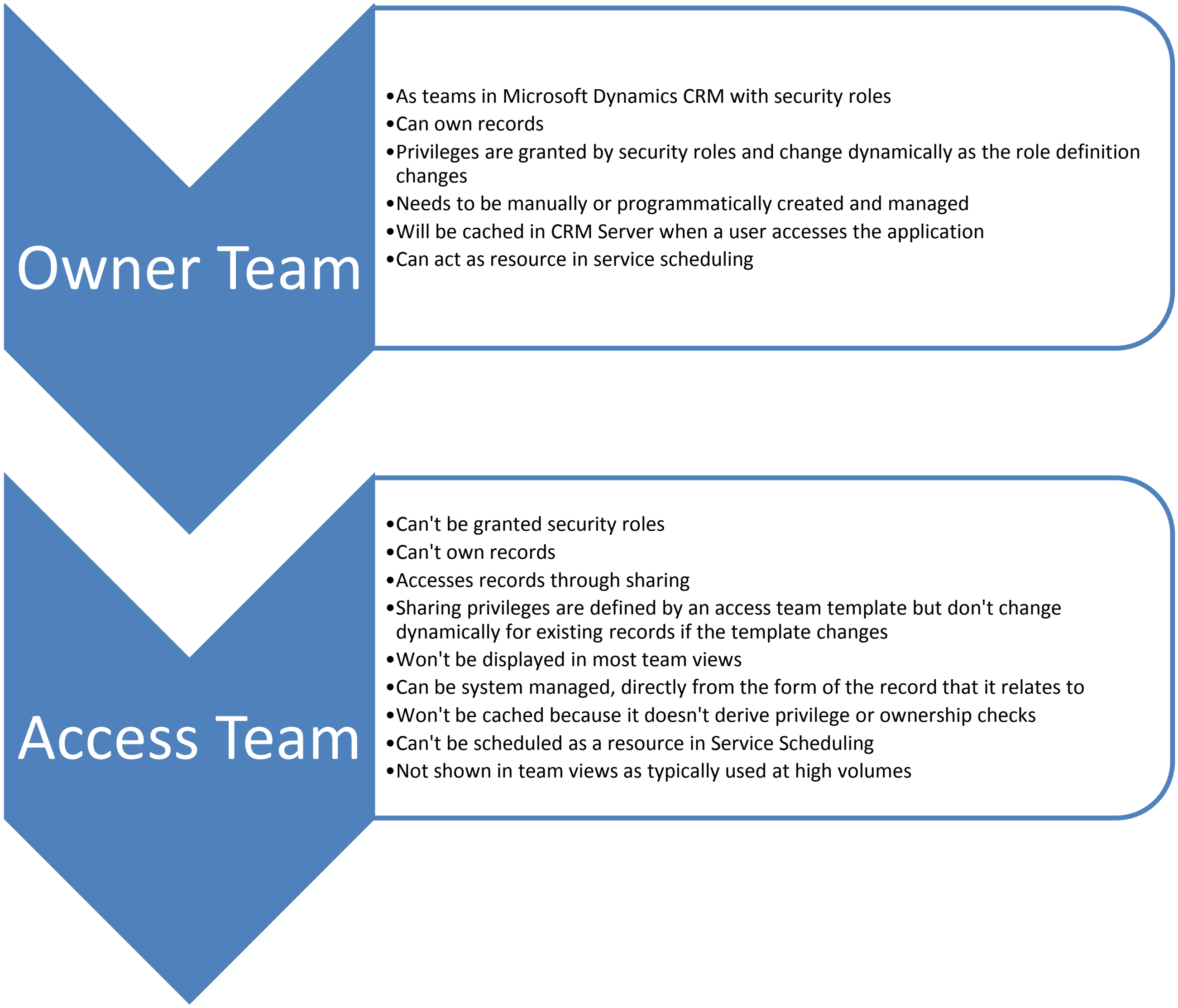
Key features

In Microsoft Dynamics CRM 2013, we have introduced new features to address these challenges.

Access teams

To address the concerns of the overhead of calculating security roles when they are needed, we will be splitting out different types of teams to cater for these different access scenarios.

With Microsoft Dynamics CRM 2013 we will allow you to create two different types of teams:



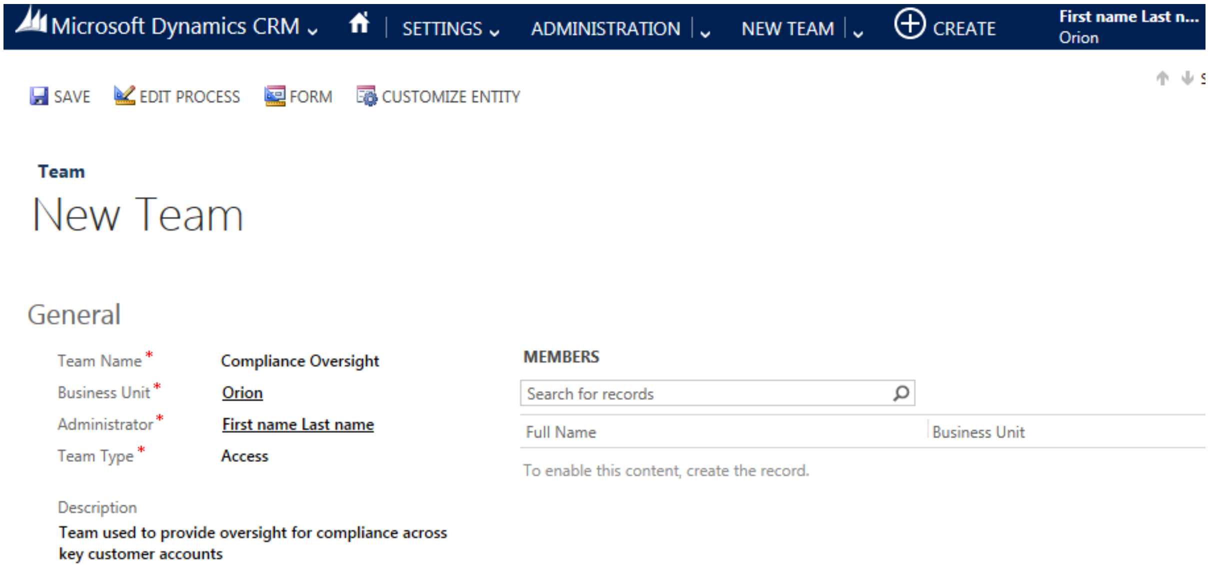
Owner Team

- As teams in Microsoft Dynamics CRM with security roles
- Can own records
- Privileges are granted by security roles and change dynamically as the role definition changes
- Needs to be manually or programmatically created and managed
- Will be cached in CRM Server when a user accesses the application
- Can act as resource in service scheduling

Access Team

- Can't be granted security roles
- Can't own records
- Accesses records through sharing
- Sharing privileges are defined by an access team template but don't change dynamically for existing records if the template changes
- Won't be displayed in most team views
- Can be system managed, directly from the form of the record that it relates to
- Won't be cached because it doesn't derive privilege or ownership checks
- Can't be scheduled as a resource in Service Scheduling
- Not shown in team views as typically used at high volumes

Teams, both owner and access types, can be created manually through the teams user interface. The team type field is editable only on creation and used to define the type of team created:



Team

New Team

General

Team Name * Compliance Oversight

Business Unit * Orion

Administrator * First name Last name

Team Type * Access

Description

Team used to provide oversight for compliance across key customer accounts

MEMBERS

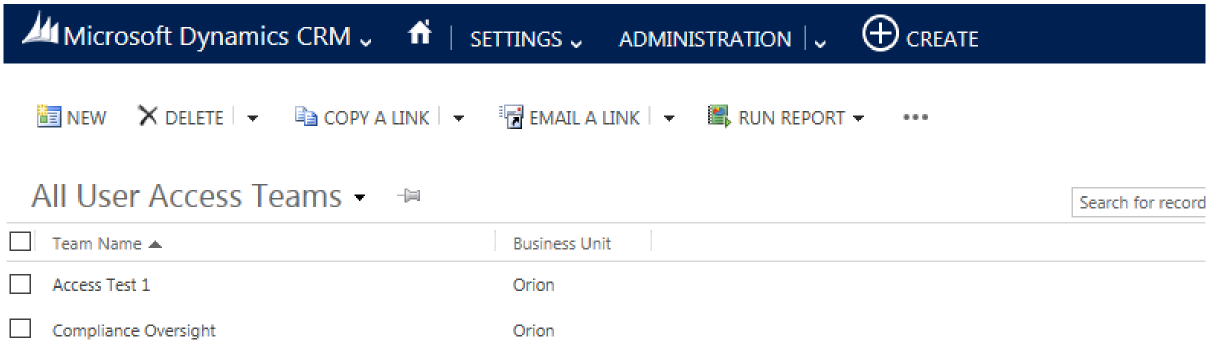
Search for records

Full Name	Business Unit

To enable this content, create the record.

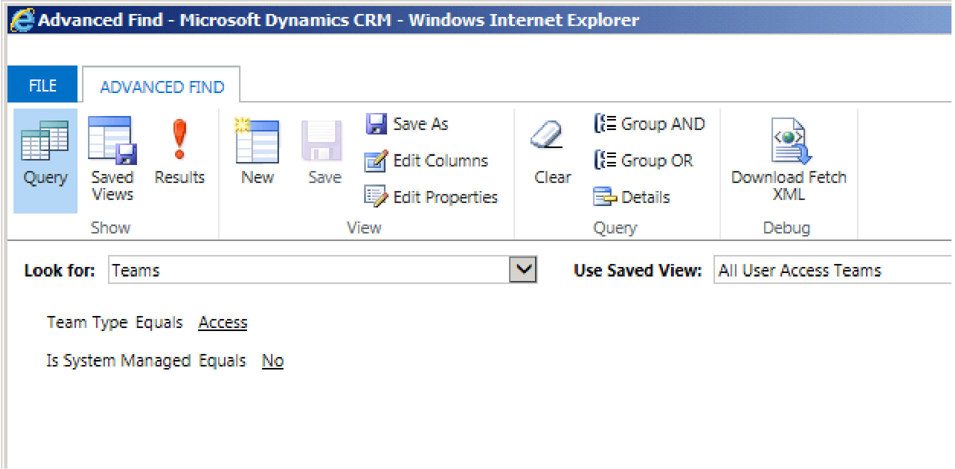
To view access teams that have been created, there are two key team attributes to be considered in views or advanced find:

- Team Type: Owner or Access
- Is System Managed: defines whether the teams are automatically generated by the system or manually created



All User Access Teams

Team Name	Business Unit
Access Test 1	Orion
Compliance Oversight	Orion



Look for: Teams

Use Saved View: All User Access Teams

Team Type Equals Access

Is System Managed Equals No

By default, Access Teams are only viewable from the “All Access Teams” view and are excluded from the “All Teams” view as they are generally expected to be for internal system visibility and management rather than something a user would directly view or interact with.

Automatic creation of access teams

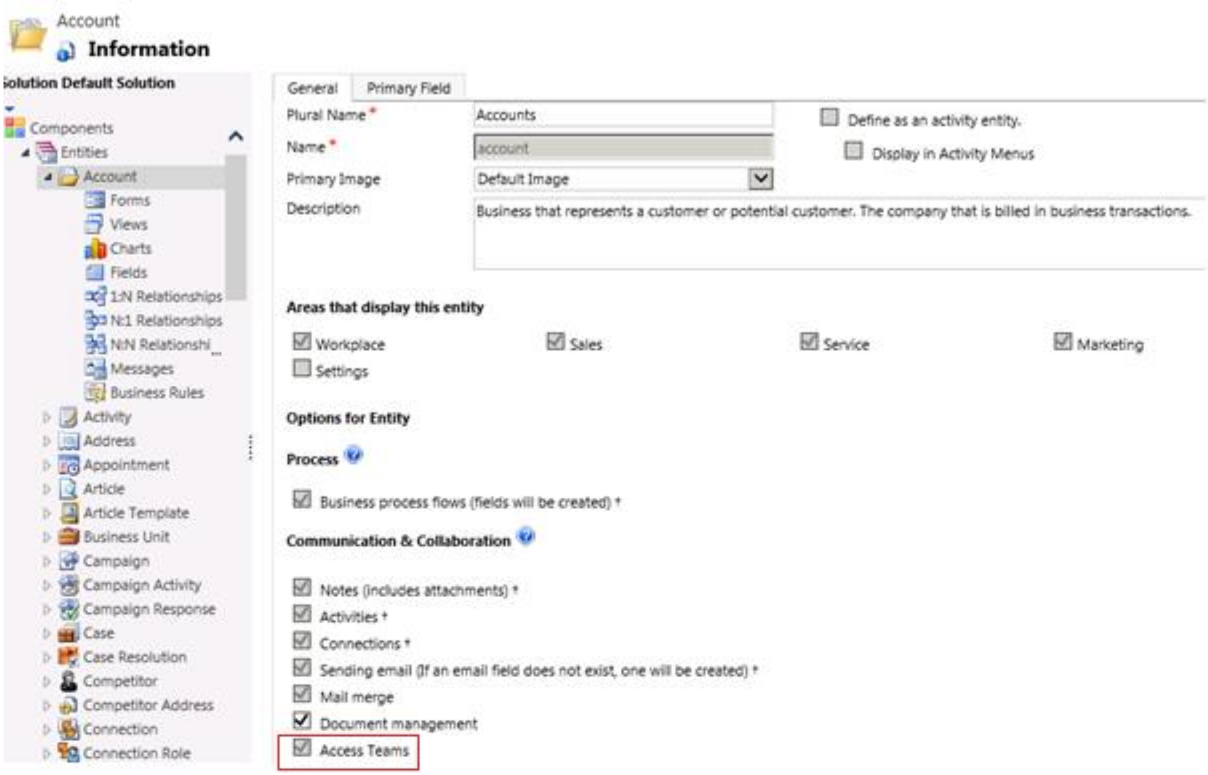
When choosing to take advantage of access teams for a particular entity, there are a few steps to perform:

- Enable the entity for access teams
- Set up an access team template
- Add a subgrid to the entity form to view the team members
- Add members to the subgrid

Enable the entity for access teams

The first step is to enable the access teams for that entity.

This is done through the entity configuration. On the entity definition, there is a check box for access teams, which enables access teams for that entity, as shown here.



Plural Name * Accounts

Name * account

Primary Image Default Image

Description Business that represents a customer or potential customer. The company that is billed in business transactions.

☐ Define as an activity entity.

☐ Display in Activity Menus

Areas that display this entity

☑ Workplace ☑ Sales ☑ Service ☑ Marketing

☐ Settings

Options for Entity

Process

☑ Business process flows (fields will be created) †

Communication & Collaboration

☑ Notes (includes attachments) †

☑ Activities †

☑ Connections †

☑ Sending email (If an email field does not exist, one will be created) †

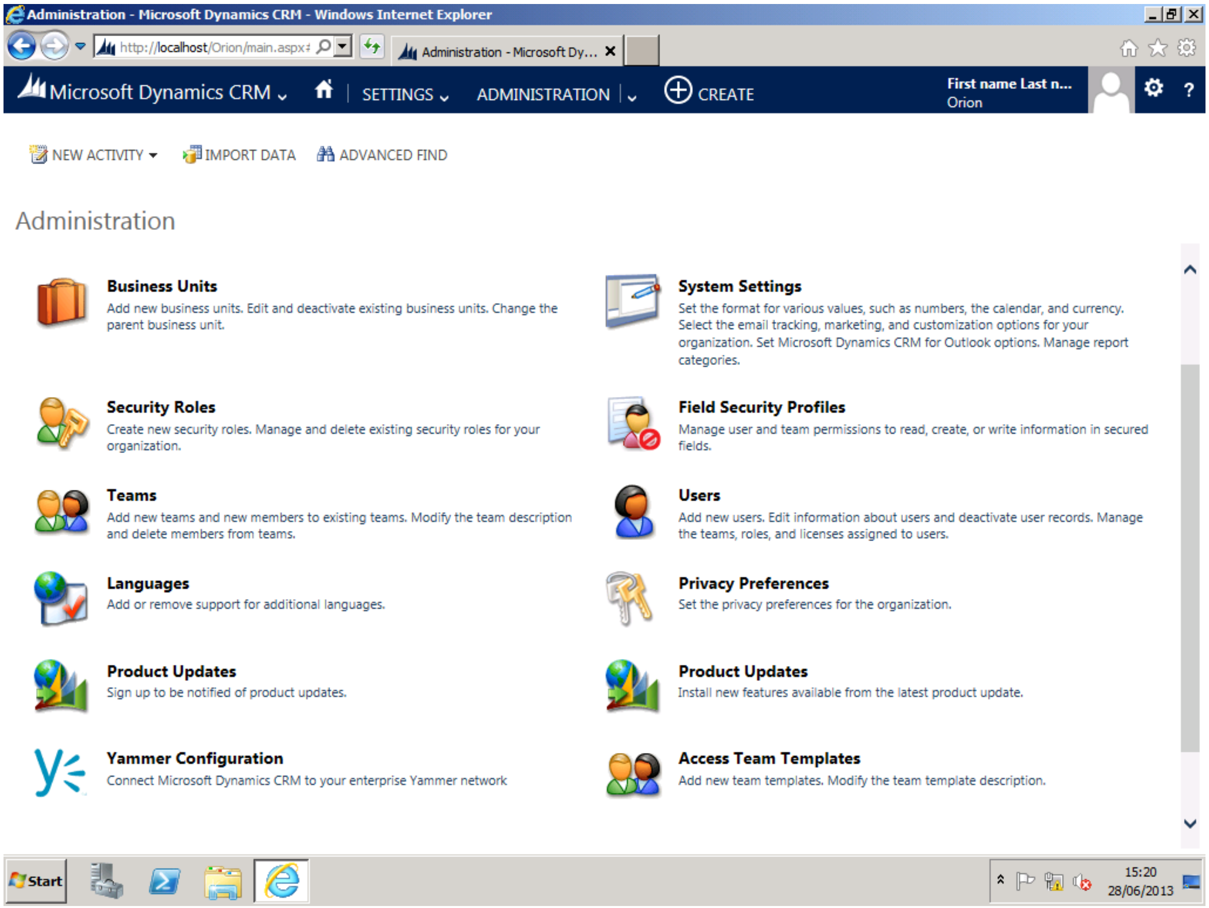
☑ Mail merge

☑ Document management

☑ Access Teams

Set up team template for that entity type

The next piece to set up is the template for the access team for the entity type. This is done from the Administration section in Settings.


Administration - Microsoft Dynamics CRM - Windows Internet Explorer
http://localhost/Orion/main.aspx
Administration - Microsoft Dy...
Microsoft Dynamics CRM
SETTINGS
ADMINISTRATION
CREATE
First name Last n...
Orion
NEW ACTIVITY
IMPORT DATA
ADVANCED FIND
Start
15:20
28/06/2013

Administration

Business Units
Add new business units. Edit and deactivate existing business units. Change the parent business unit.

System Settings
Set the format for various values, such as numbers, the calendar, and currency. Select the email tracking, marketing, and customization options for your organization. Set Microsoft Dynamics CRM for Outlook options. Manage report categories.

Security Roles
Create new security roles. Manage and delete existing security roles for your organization.

Field Security Profiles
Manage user and team permissions to read, create, or write information in secured fields.

Teams
Add new teams and new members to existing teams. Modify the team description and delete members from teams.

Users
Add new users. Edit information about users and deactivate user records. Manage the teams, roles, and licenses assigned to users.

Languages
Add or remove support for additional languages.

Privacy Preferences
Set the privacy preferences for the organization.

Product Updates
Sign up to be notified of product updates.

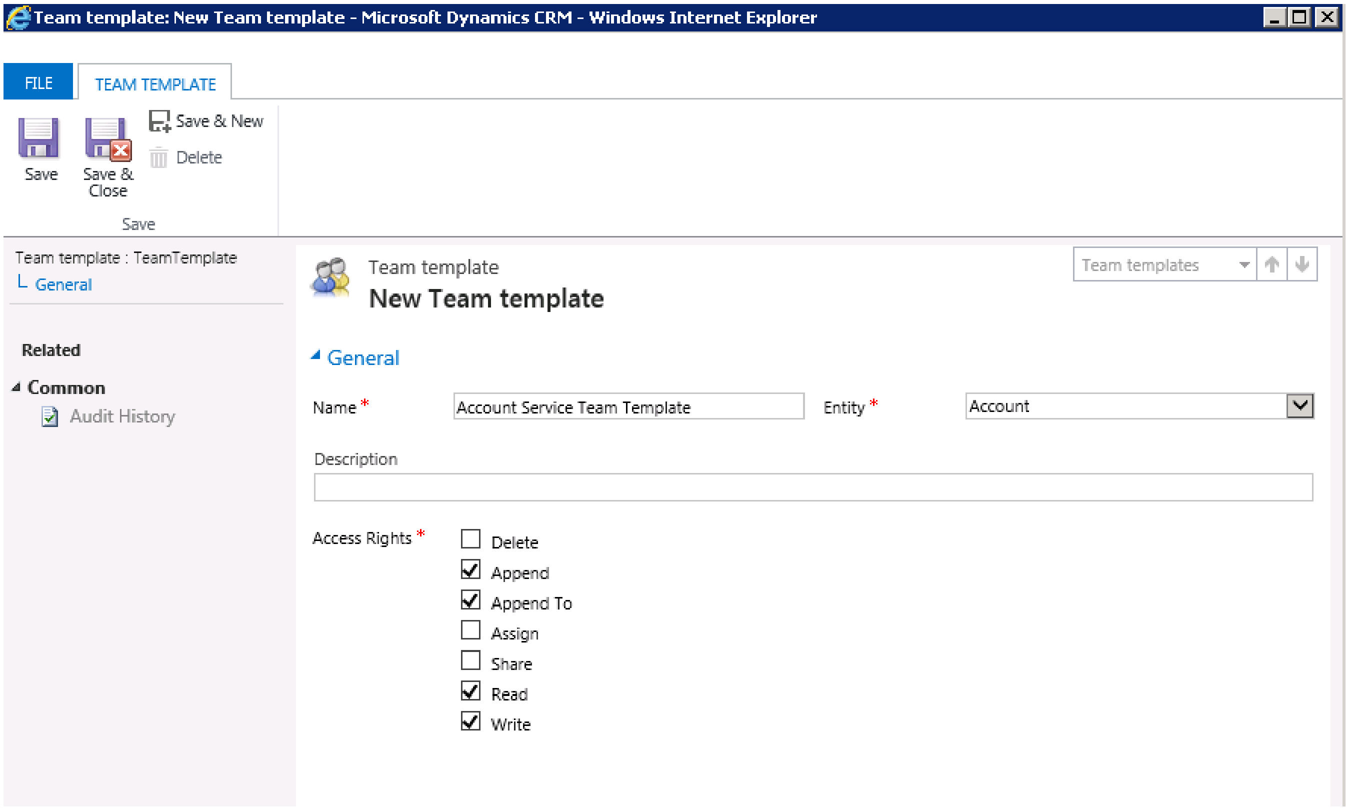
Product Updates
Install new features available from the latest product update.

Yammer Configuration
Connect Microsoft Dynamics CRM to your enterprise Yammer network

Access Team Templates
Add new team templates. Modify the team template description.

Select the Access Team Templates area, and you can add a new template for the entity type you are enabling.

At this point, you can define the access rights that should be granted to the team when it is created for records of this entity type.


Team template: New Team template - Microsoft Dynamics CRM - Windows Internet Explorer
FILE
TEAM TEMPLATE
Save
Save & Close
Save & New
Delete
Save
Team template : TeamTemplate
General
Related
Common
Audit History
Team templates
General

Team template

New Team template

Name * Account Service Team Template

Entity * Account

Description

Access Rights *

- ☐ Delete
- ☑ Append
- ☑ Append To
- ☐ Assign
- ☐ Share
- ☑ Read
- ☑ Write

You can create multiple templates for each entity type, and these different template types can be selected when the subgrid is added to the entity form.

Add a subgrid to the entity form to view the team members

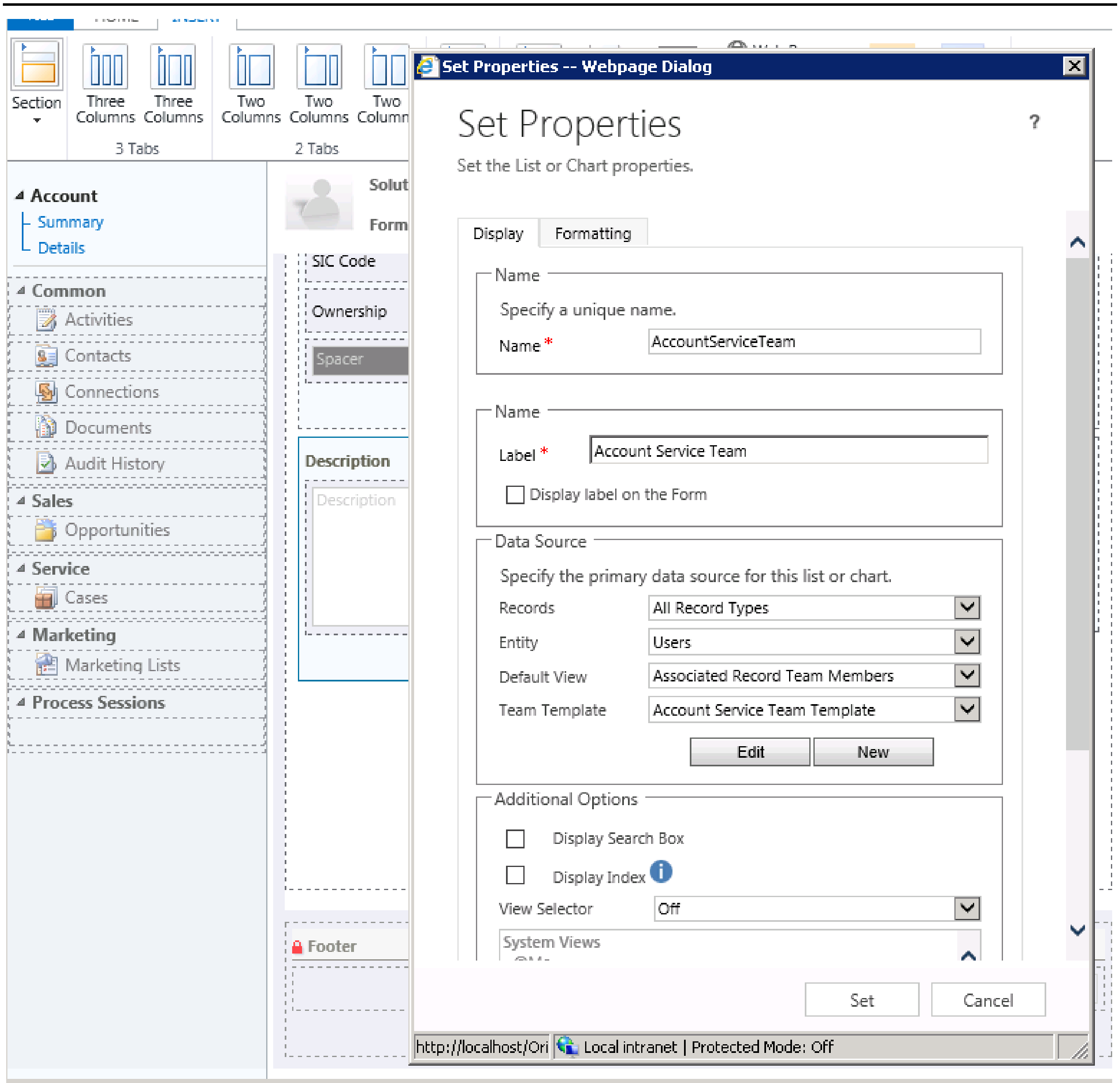
Once you enable an entity for access teams and you define the access team template, you need to add the user interface to view and manage the team members.

This is performed by adding a subgrid to the form, and showing the related access team members. As will be shown in the next section, the link between the team members in the access team and the record is managed automatically. Providing the mechanism to view existing team members and to change the team membership is done by adding a subgrid with a lookup to:

- Records: All Record Types

  Note: The instinct may be to select ”Only Related Records” but, because there is no direct relationship created between the entity enabled for access teams and the users, this won’t work and the indirect relationship is managed differently.
- Entity Type: Users
- Default View: Associate Record Team Members
- Team Template: The template you’d like to use for this type of team

This is shown in the properties dialog box for the subgrid in the next screenshot.



Set Properties

Set the List or Chart properties.

Name

Specify a unique name.

Name * AccountServiceTeam

Name

Label * Account Service Team

□ Display label on the Form

Data Source

Specify the primary data source for this list or chart.

Records	All Record Types
Entity	Users
Default View	Associated Record Team Members
Team Template	Account Service Team Template

Additional Options

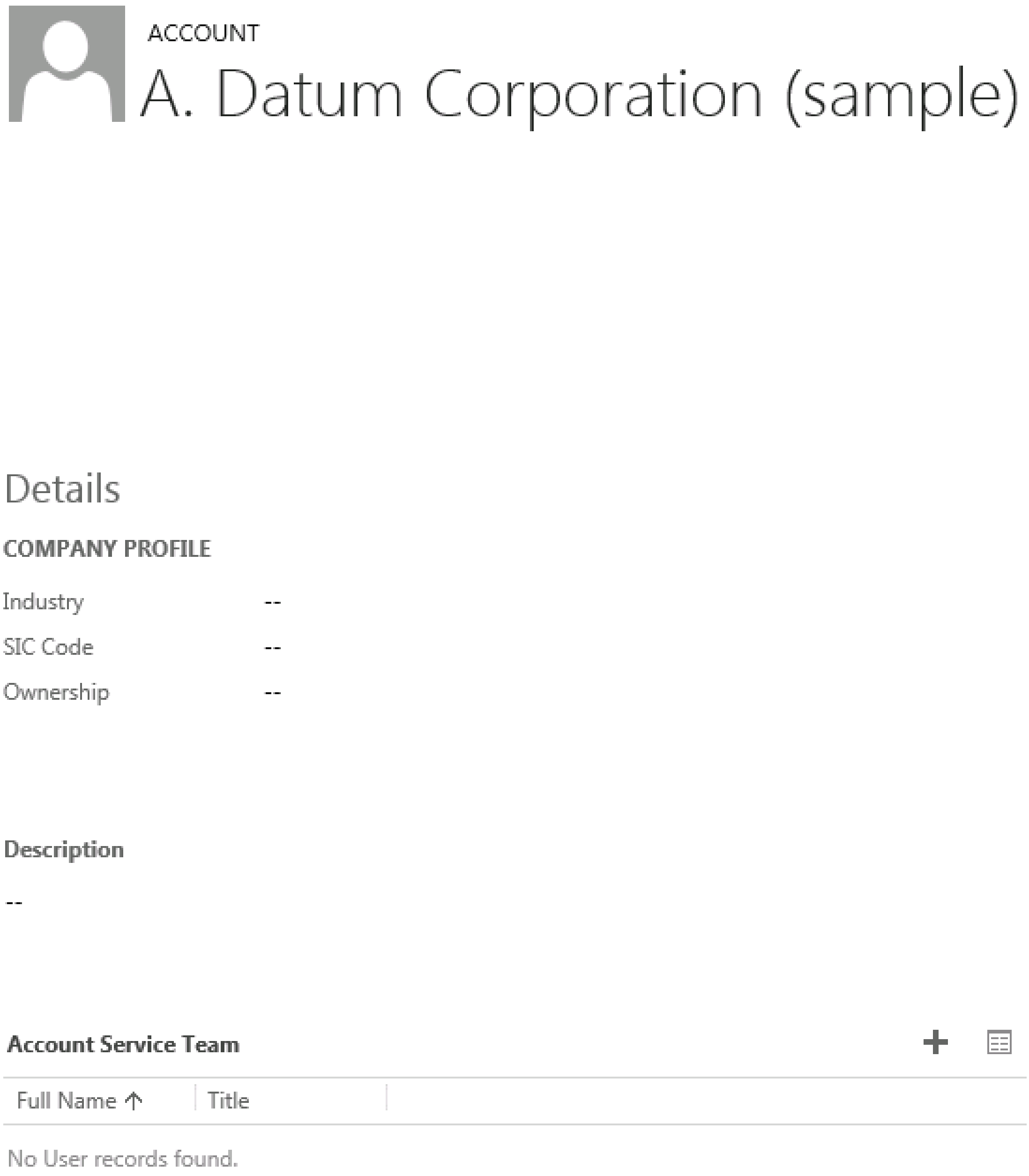
□ Display Search Box

□ Display Index

View Selector Off

System Views

Once this is saved and published, this is visualized on the form as shown in the following screenshot.



ACCOUNT

A. Datum Corporation (sample)

Details

COMPANY PROFILE

Industry	--
SIC Code	--
Ownership	--

Description

--

Account Service Team

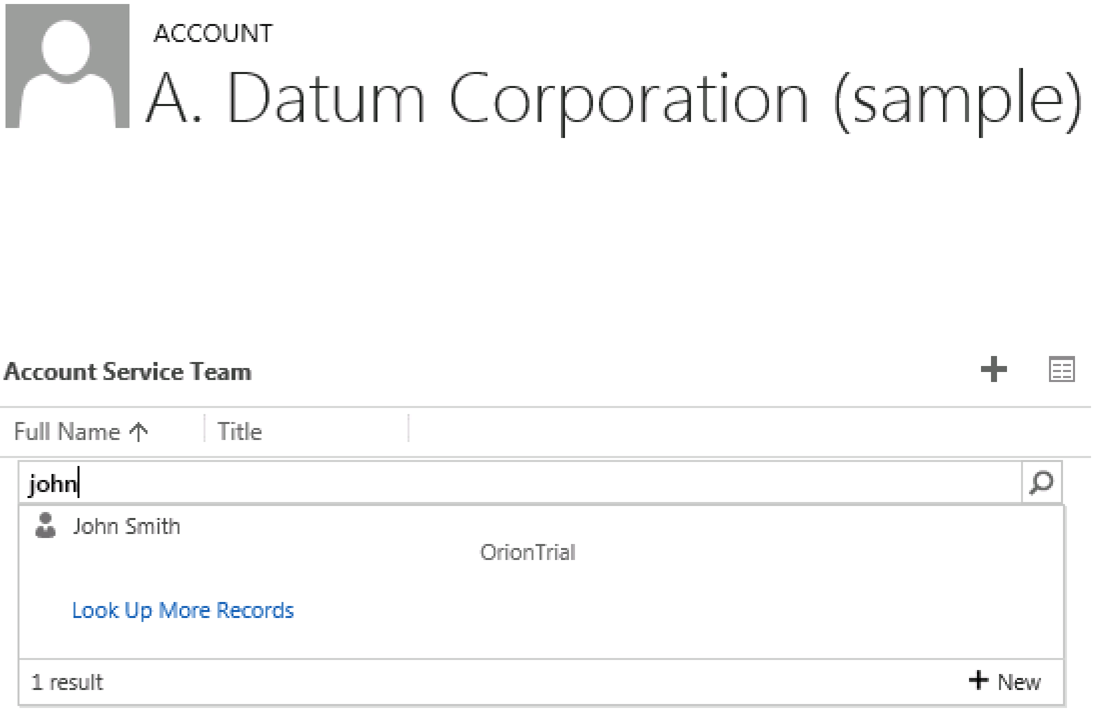
Full Name ↑	Title

No User records found.

It’s possible to add multiple subgrids linked to different access team templates for the same entity enabling multiple access teams to be created against the same record but with different permissions. It is important in this scenario to make sure that the naming of the team template and the label of the subgrid surfacing this are clear and descriptive to keep the uses separate and clear.

Add members to the sub grid

From here an end user can then add a user to the sales team via the subgrid, by selecting the plus sign (+) on the sub grid and performing a lookup against the user who is then added to the team. This is possible for any user with 'Sharing' privileges for the current record.



ACCOUNT

A. Datum Corporation (sample)

Account Service Team

Full Name ↑	Title

john

John Smith

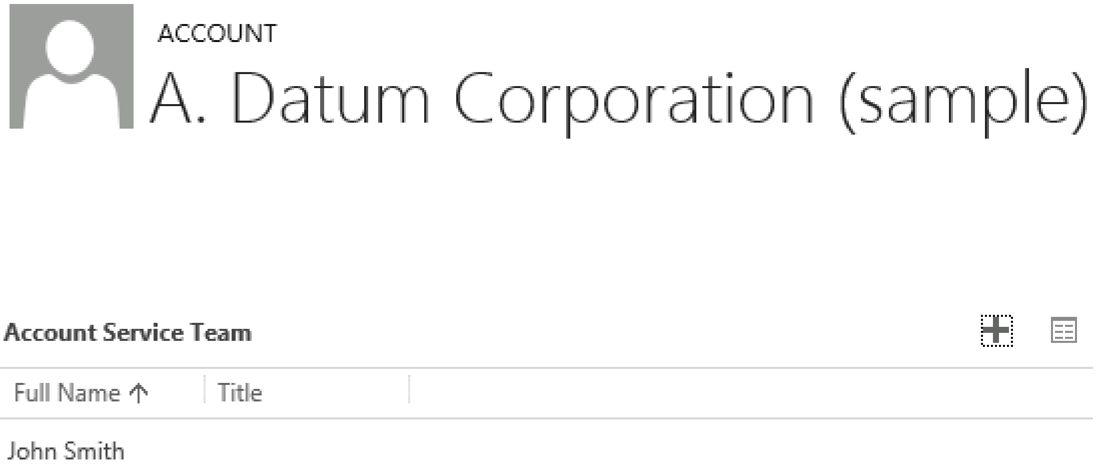
OrionTrial

Look Up More Records

1 result

+ New

The user is now added to the access team.



ACCOUNT

A. Datum Corporation (sample)

Account Service Team

Full Name ↑	Title
John Smith	

And at this point the user is able to access the record with the privileges defined in the team template for this entity type.

How it works

So, we've now enabled this functionality for access teams. But if we're going to do this at scale, we want to understand what's actually happening so we can plan for the implications.

Access teams

Access Teams are created as any other team, just without the ability to have security roles or own records, so in many ways as teams were in Microsoft Dynamics CRM 4.0.

These can then have users added to them as members and can have records shared with them.

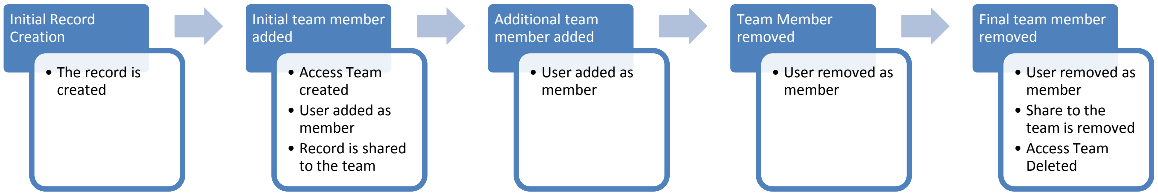
The advantage of splitting out these type of teams from owner-based teams is that the overhead of adding security roles to a team can be avoided where it is not required. When security roles are added, these can affect performance and scalability in some key ways:

- Team security roles act in a cumulative way, and as a result for each user who is a team member, the privileges for that user need to take into account all the roles of the teams that they are a member of.
- The more teams a user is a member of with security roles, the more complex the calculation that has to be performed.
- To reduce the impact of doing this on each request, the system caches the cumulative permissions for the user as they first connect to CRM.
    - When a user has a large number of teams with security roles, this can cause a delay on initial connection after a restart or when the user's record has been flushed from the cache after 20 minutes of inactivity.
- Whenever the user is added or removed from a team, or the team has its security roles changed, the cache for each user affected needs to be flushed and recalculated on the next connection.
- For rapidly changing teams or team memberships, this can introduce a significant performance and scalability impact to the system
- In these cases, access teams can avoid this impact for team memberships where this is not necessary, that is, where a combination of ownership and sharing is used, access teams can be used for the sharing cases, and avoiding the cache and calculation impact when they change.

Service Scheduling uses caching extensively to optimize its calculations. Therefore, where teams are used as resources in service scheduling, we need to load the teams into the cache. Access teams, therefore, can't be used as resources in service scheduling to avoid the need to cache them and to avoid the impact of recalculating the resource groups when a user's team memberships change.

Team selling

When access teams are enabled for an entity, the lifecycle of the team is managed automatically.

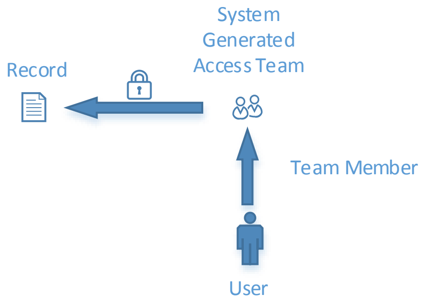


As shown in the preceding diagram, the access team is generated automatically on demand as the first team member is added.

When the final team member is removed, the team is deleted.

This brings advantages where you have rapid changing of large numbers of teams, such as removing any old and redundant access that may no longer be needed.

When a user is added to the team connected to a record, a membership is set up for that team and the user is linked to the team in the **SystemUserPrincipals** table.



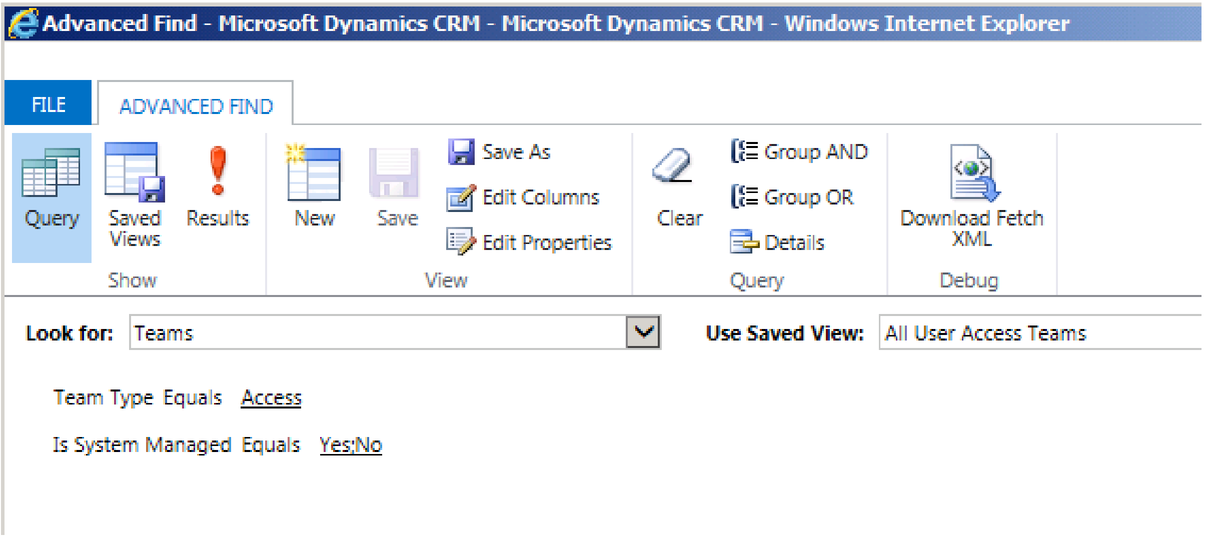
Access to the record is established through automatically sharing the record with the team, with the privileges defined in the access team template for that entity type.

If the record is deactivated, this won’t affect the related access team, only removing all the team members will cause deletion of the automatically generated access team

Viewing the access teams

In normal operation, the access team created automatically, system generated, is hidden from use as it is managed behind the scenes by the Microsoft Dynamics CRM application rather than needing manual administration.

In order to view system generated teams, as these are deliberately withheld from standard views, you can use advanced find to query for teams of type ”Access” and that are ”Is System Managed” as shown here or alternatively create or edit a personal or system view to more regularly view these:



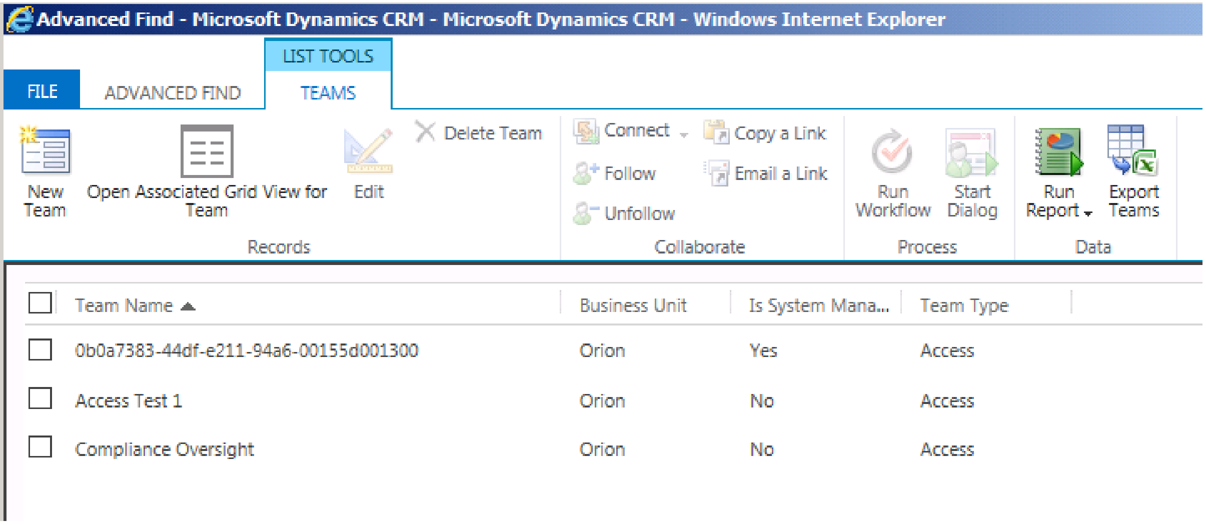
Look for: Teams

Use Saved View: All User Access Teams

Team Type Equals Access

Is System Managed Equals Yes;No

This view shows both manually created and system managed access teams, and shows that system managed teams are named with the GUID of the record to which they are linked, in this case of the account to which it is linked (although if multiple access teams are linked to the same record, the second team will have a different name).



Team Name	Business Unit	Is System Mana...	Team Type
0b0a7383-44df-e211-94a6-00155d001300	Orion	Yes	Access
Access Test 1	Orion	No	Access
Compliance Oversight	Orion	No	Access

Sharing

As part of team selling, the access privileges are based on use of the sharing model. As part of the Microsoft Dynamics CRM 2013 release, a number of optimizations are performed on the sharing model.

At very high scale this has however been known in the past to lead to large record sets and performance implications so these implications do still need to be allowed for in the design and use of this feature.

One characteristic to note, is that shares granted to system-managed access teams aren't shown to the end user. As these sharing permissions are automatically set up and managed, they aren't available for an end user to overrule and are therefore not shown in the normal sharing dialog.

As access teams use sharing to control record privileges, being granted membership of an access team for a record will trigger normal cascading behavior to any related records. If there is a relationship between the record with the access team and any related records that have cascading sharing behaviors enabled, the access team will also be shared with the child records.

Programmatic control of access teams

Access teams can be controlled through the SDK as normal for programmatic control of teams.

One scenario that needs special consideration is when you add members to an access team for a record. Because a system-managed access team is only created on addition of the first member, this team may not yet exist to add the member to. The normal SDK messages require a team to exist before a member can be added to it.

To handle this situation, a new SDK message is introduced, **AddUserToRecordTeamRequest**, which allows you to specify the record and team template that the user should be added to. This will resolve the request to the access team that is relevant and, if necessary, create the team before adding the user as a member.

When working with system-generated access teams, this message should be used for adding users as members.

Design considerations of using access teams

Access teams can't own a record, they don't have any security roles, and therefore they can't be granted the privileges to own a record. Nor can they access records through the security role privileges of Owner, Business Unit or Organization level scopes.

For very large volumes, the implications of sharing still needs to be considered carefully. In particular, managing the lifecycle of the access team's existence for records that no longer need to be directly viewed should be considered.

When a record is deactivated, this doesn't change the related access team membership as this may be needed if the record is reactivated, or for compliance purposes. Removing all the team members, either through the user interface or programmatically, for a record can be used to remove the related access team.

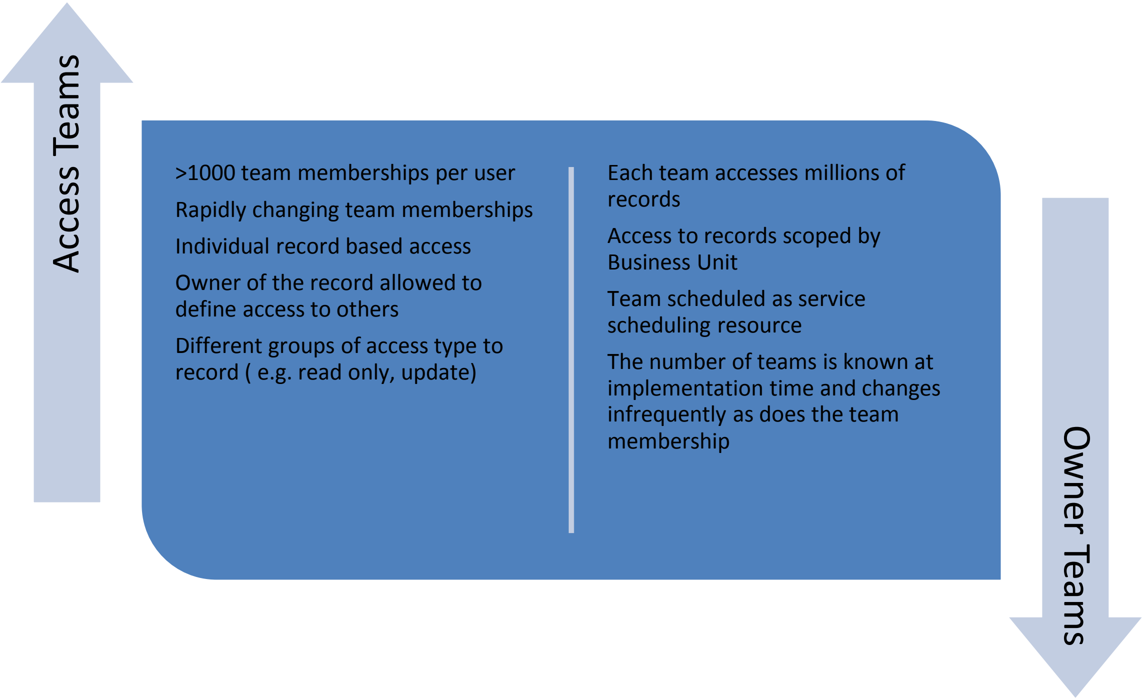
Access teams can't be used as resources in service scheduling.

Multiple access teams can be linked to a single record, allowing different access types to be defined for the same record, such as defining a read-only access team and an update team.

It's not possible to selectively choose the type of team per individual record, as the definition of the number of access teams and the team templates to be used are defined within the form for that entity, so will apply to all records of that particular entity type. For example, it wouldn't be possible to have an attribute of account defining whether they are a prospect or existing customer and therefore applying different team templates based on whether they are a prospect or existing customer. In that example, all account records would be linked to the same types of access teams. Whether an actual team instance is created would depend on whether individual users are added to the subgrid in the form itself though.

One option to consider is that different role-based forms can be used, each linking to different access team templates which does provide some ability to offer variations, although the likelihood if users of each role type access the record would be that teams of all the different types would be created even though users would only see some of them when they access the form.

With the separation of owner teams and access teams, there will be scenarios where each is the more appropriate choice. The following diagram highlights some key factors to consider when deciding which to use:



>1000 team memberships per user

Rapidly changing team memberships

Individual record based access

Owner of the record allowed to define access to others

Different groups of access type to record (e.g. read only, update)

Each team accesses millions of records

Access to records scoped by Business Unit

Team scheduled as service scheduling resource

The number of teams is known at implementation time and changes infrequently as does the team membership

One option to note, it is possible to combine:

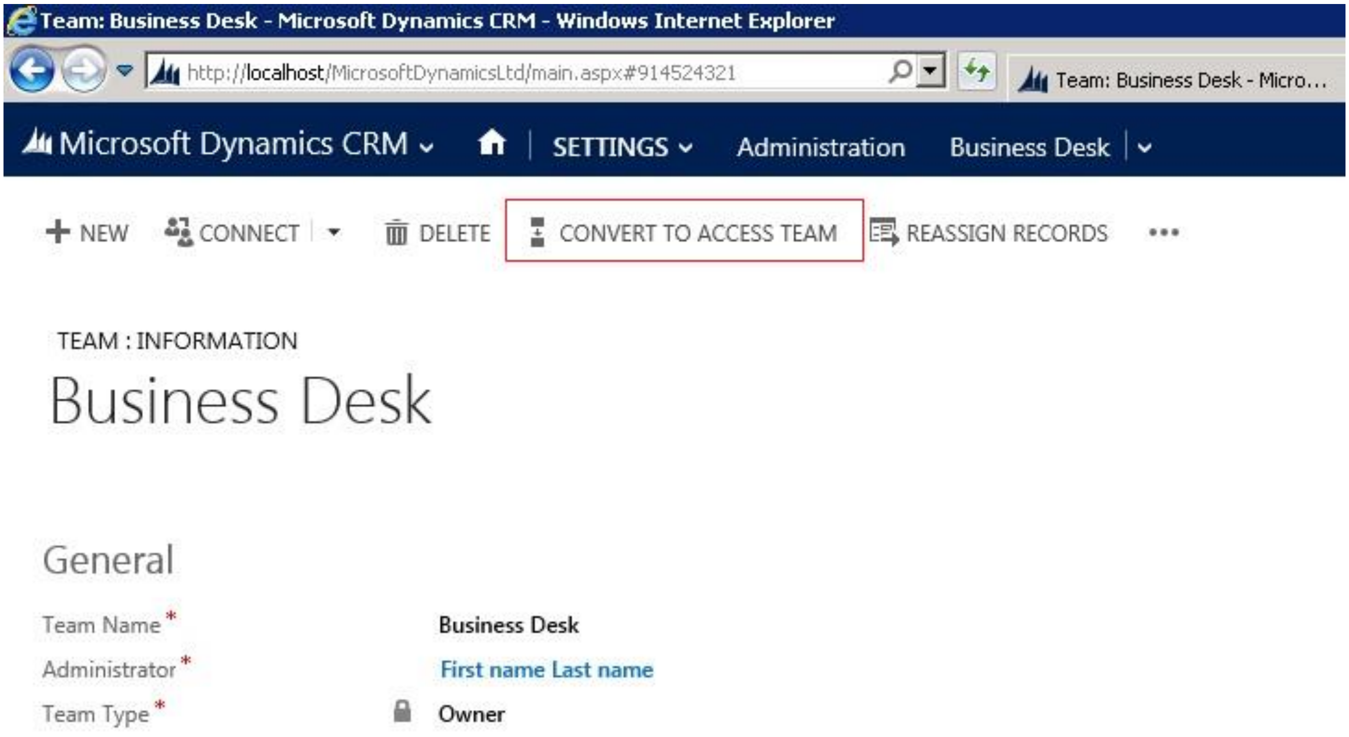
- Using an owner team to “own” a record, perhaps giving broader access to a wider range of records to a particular team
- And then separately to use access teams to provide more granular access for other users

Migration path

Where teams have been migrated from a previous version, they will be migrated into Microsoft Dynamics CRM 2013 as owner teams. It’s possible to migrate owner teams to access teams as long as:

- The owner team doesn’t have any security roles
- The owner team isn’t the owner of any records

This will enable scenarios where customer’s teams are good candidates for access teams to be migrated without having to be recreated and set up. The conversion is done through an action on the command bar, as shown in the following illustration.


Team: Business Desk - Microsoft Dynamics CRM - Windows Internet Explorer
http://localhost/MicrosoftDynamicsLtd/main.aspx#914524321
Team: Business Desk - Micro...
Microsoft Dynamics CRM
SETTINGS
Administration
Business Desk
NEW
CONNECT
DELETE
CONVERT TO ACCESS TEAM
REASSIGN RECORDS
•••

TEAM : INFORMATION

Business Desk

General

Team Name * Business Desk

Administrator * First name Last name

Team Type * Owner

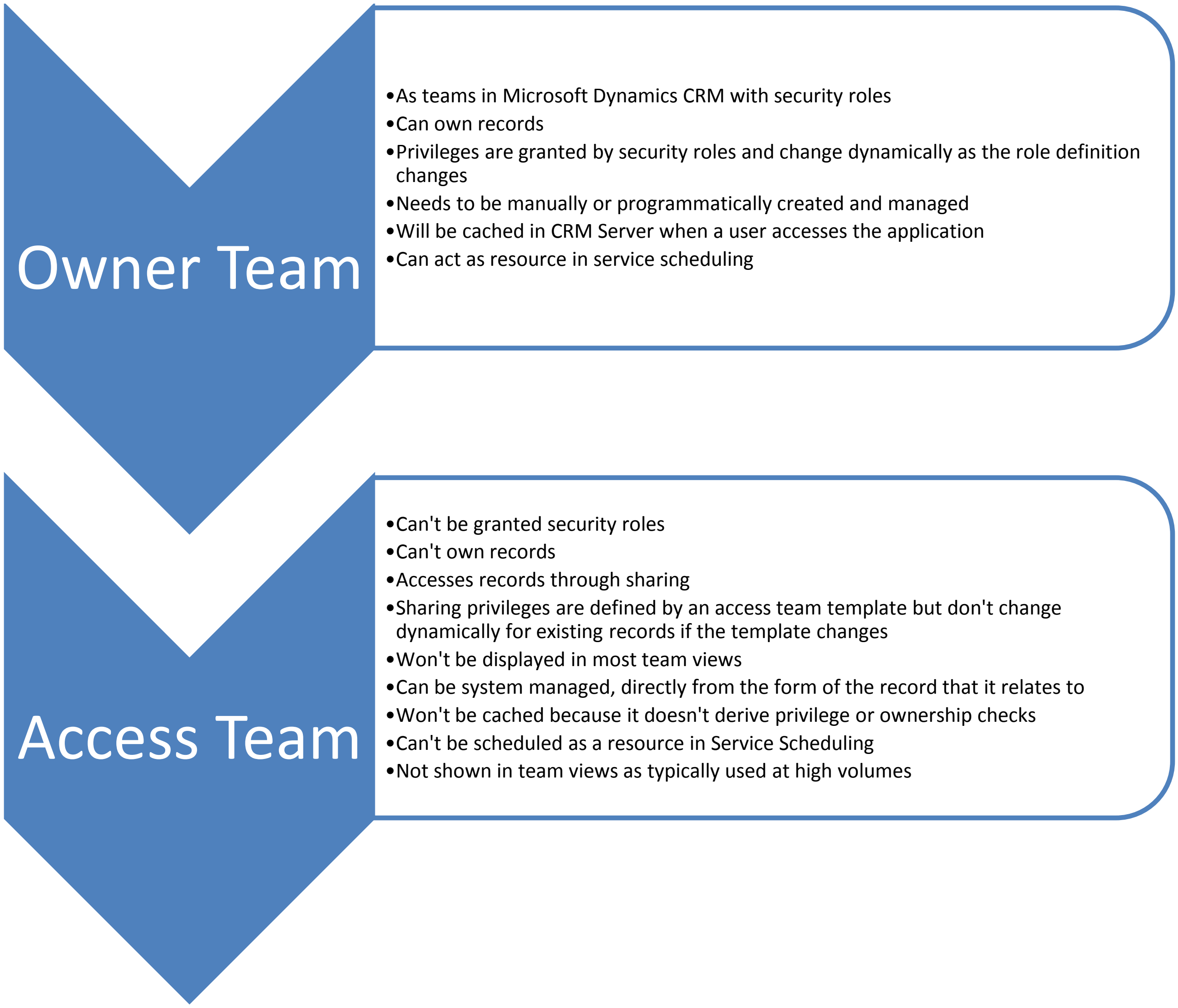
While the introduction of team ownership and team security roles add great flexibility, for customers who do not need them this overhead can be avoided by using access teams. This is a good path for customers migrating from Microsoft Dynamics CRM 4.0 who have implemented solutions with large volumes of teams using sharing and can benefit from the simplicity of Access Teams in Microsoft Dynamics CRM 2013 as access teams more closely reflect the teams approach from version 4.0.

It isn’t possible to migrate access teams to owner teams.

Summary

With the introduction of access teams in Microsoft Dynamics CRM 2013, we simplify the process of managing granular access to records where individual access needs to be defined and managed. We also optimize the way this access can be managed.

Key features of owner teams and access teams are summarized as follows:



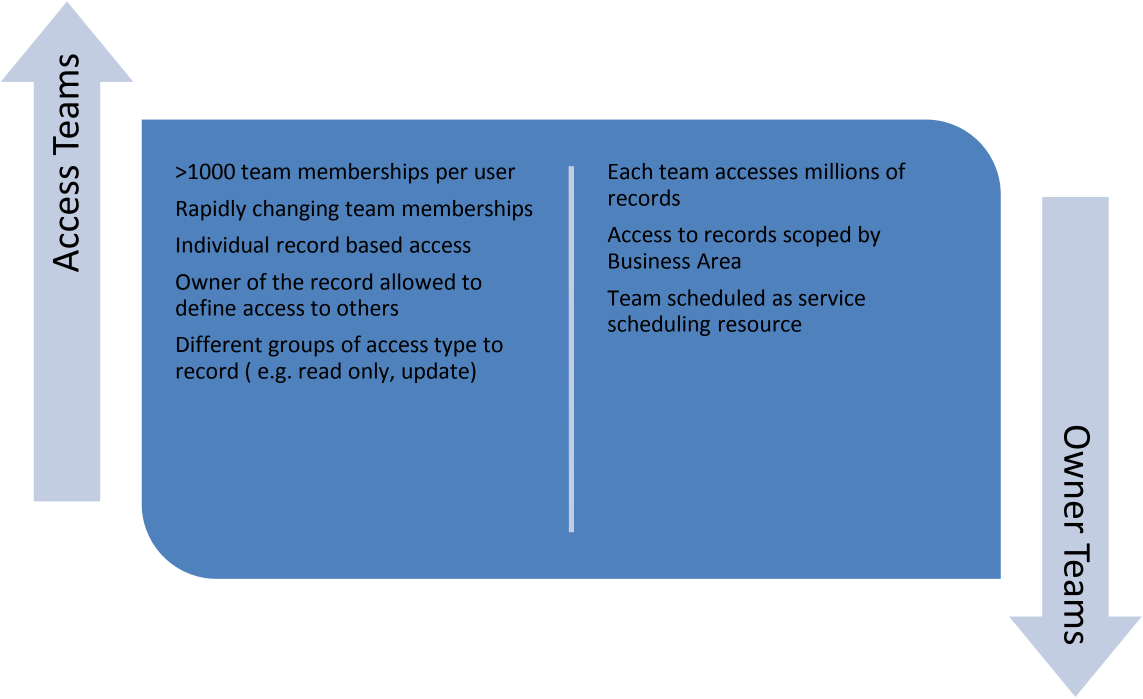
Owner Team

- As teams in Microsoft Dynamics CRM with security roles
- Can own records
- Privileges are granted by security roles and change dynamically as the role definition changes
- Needs to be manually or programmatically created and managed
- Will be cached in CRM Server when a user accesses the application
- Can act as resource in service scheduling

Access Team

- Can't be granted security roles
- Can't own records
- Accesses records through sharing
- Sharing privileges are defined by an access team template but don't change dynamically for existing records if the template changes
- Won't be displayed in most team views
- Can be system managed, directly from the form of the record that it relates to
- Won't be cached because it doesn't derive privilege or ownership checks
- Can't be scheduled as a resource in Service Scheduling
- Not shown in team views as typically used at high volumes

In the context of a particular implementation design, use of teams needs to be carefully analyzed as each type is intended to target specific access scenarios and other capabilities in the CRM security model should also still be considered and used as appropriate. In particular comparison between when to use access teams and owner teams is summarized as shown in the following illustration.



- >1000 team memberships per user
- Rapidly changing team memberships
- Individual record based access
- Owner of the record allowed to define access to others
- Different groups of access type to record (e.g. read only, update)

- Each team accesses millions of records
- Access to records scoped by Business Area
- Team scheduled as service scheduling resource

For further information on when to use different security modeling features of Microsoft Dynamics CRM, see the white paper, Scalable Security Modeling with Microsoft Dynamics CRM 2013.