

Microsoft Dynamics CRM 2013 and Microsoft Dynamics CRM 2015 Service Provider Planning and Deployment Guide

Updated June 2015

Copyright

This document is provided "as-is". Information and views expressed in this document, including URL and other Internet website references, may change without notice. You bear the risk of using it.

Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

© 2015 Microsoft Corporation. All rights reserved.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

For more information, see **Use of Microsoft Copyrighted Content** at <http://www.microsoft.com/about/legal/permissions/>.

Microsoft, Active Directory, IntelliSense, Internet Explorer, Microsoft Dynamics, the Microsoft Dynamics logo, Outlook, SQL Server, Visual Studio, Windows, Windows PowerShell, Windows Server, Windows Server System, and Windows Vista are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Microsoft products mentioned herein may be either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks are property of their respective owners.

Table of Contents

1	Introduction	1
1.1	Who Should Read This Document	1
1.2	Scope and Assumptions	1
2	Deployment Overview.....	2
2.1	Summary of Deployment Process	2
2.2	Deploying CRM as a Hosted Service	4
2.2.1	Infrastructure for CRM Dynamics Hosting	4
2.2.2	High Availability in Infrastructure.....	5
2.2.3	Installation Privileges Requirements.....	7
2.3	Architectural Planning and Considerations.....	7
2.3.1	CRM Services	7
2.3.2	Network Segmentation.....	10
2.3.3	Internet-facing Deployment of CRM.....	11
2.3.4	Deployment Groups	11
2.3.5	Architectural Tiers	12
2.3.6	Backup and Restore Considerations	19
2.3.7	Planning for Email Processing using Server-Side Synchronization	20
3	Deployment Installation.....	21
3.1	Example Names.....	21
3.1.1	Server Names, Roles, and Associated Software	21
3.1.2	Claims-based Authentication Considerations	23
3.1.3	Example Domain Names	23
3.2	Deploy the Hosted Microsoft Dynamics CRM Infrastructure	24
3.2.1	Prepare the Active Directory Forest Domain Infrastructure	24
3.2.2	Build and Deploy the External DNS Server	24
3.2.3	Determine the Multi-tenancy Design.....	25
3.2.4	Build and Deploy the Messaging Platform	25
3.2.5	Deploy Federation and Claims-based Authentication Platform	25
3.3	Deploy Hosted Microsoft Dynamics CRM Deployment Group Components.....	26
3.3.1	Deploy Hosted Microsoft Dynamics CRM Database Server.....	27
3.3.2	Deploy the CRM Front-end Servers.....	28
3.3.3	Install the Back-end Servers.....	31
3.3.4	Deploy Deployment Administration Servers	33
3.4	Email Processing through Server-Side Synchronization	34
3.4.1	Create the CRM Exchange Administrator Account.....	34
3.4.2	Create email server profiles	35
3.5	Deploy SharePoint Grid	38
3.6	Scripting Deployment Installations with Configuration Files	38
3.7	Deploy CRM for Outlook.....	39
4	Post-Installation Configuration and Management.....	40
4.1	Microsoft Dynamics CRM 2013 Best Practices Analyzer	40
4.1.1	Best Practices Analyzer requirements	40
4.1.2	Installation instructions.....	40
4.1.3	Run a scan using the Best Practices Analyzer	41

4.1.4	Add Deployment Administrators	41
4.1.5	Creating a New CRM Deployment Administrator Account	41
4.1.6	Creating a New CRM Deployment Administrators Group	41
4.1.7	Adding Deployment Administrator Group to CRM Server Local Administrators Group	42
4.1.8	Granting CRM Deployment Administrator Permissions to the CRM Active Directory Groups	42
4.1.9	Granting CRM Deployment Administrators Permissions to CRM SQL Objects	43
4.1.10	Adding Domain User Account to CRM Deployment Administrators Group	44
4.1.11	Adding User as a CRM Deployment Administrator in CRM Deployment Manager	44
4.1.12	Adding a Deployment Administrator	45
4.2	Configure Claims and IFD	45
4.2.1	Configuring the Microsoft Dynamics CRM Server Websites for SSL/HTTPS	45
4.2.2	Configuring Fault Tolerance and Firewall	46
4.2.3	Configuring Microsoft Dynamics CRM Server for Claims-based Authentication	47
4.2.4	Configuring the AD FS Server for Claims-based Authentication	48
4.2.5	Configuring Microsoft Dynamics CRM for Internet-facing Deployment	49
5	Upgrade Guidance.....	51
5.1	Design Hosted Microsoft Dynamics CRM	53
5.2	Deploy Hosted Microsoft Dynamics CRM	53
5.3	Upgrade a CRM Organization	53
5.3.1	Upgrade Options	53
5.3.2	Software Prerequisites	54
5.3.3	Migrate by using a new instance of SQL Server	54
5.3.4	Backing up CRM Organization Database	55
5.3.5	Restoring CRM 2011 Organization Database into CRM SQL	55
5.3.6	Importing a CRM Organization Database	55
5.3.7	Modifying DNS Records for CRM Organization.....	57
5.3.8	Migrate settings from the Email Router to server-side synchronization	57
5.3.9	Enabling Anonymous Authentication for the Discovery Web Service	58
5.3.10	Refreshing the CRM Organization Identifiers in AD FS.....	58
5.4	Verify Access Using Web Client and Outlook.....	58
5.4.1	Verify the Web Client	59
5.4.2	Verify the CRM for Outlook Client.....	59
5.5	Upgrade the CRM for Outlook Client.....	59
6	Provisioning.....	61
6.1	Manual Provisioning.....	61
6.1.1	Creating, Importing, Editing Organizations	61
6.1.2	Business Unit Provisioning	67
6.1.3	User Provisioning	67
6.1.4	Enabling CRM Organization and Users for Email Routing	68
6.1.5	Security Role Provisioning	72
6.1.6	Field Security Profile Provisioning	73
6.1.7	Language Provisioning	74
6.1.8	Troubleshooting Options	74
6.2	Automated Provisioning.....	75
6.2.1	Prerequisites	75
6.2.2	Using the CRM Dynamics Deployment Web Service to Provision Tenant Organizations.....	75
6.2.3	Using the CRM Dynamics Web Services to Provision Tenant Organization Objects .	82

1 Introduction

Welcome to the *Microsoft Dynamics CRM 2015 and Microsoft Dynamics CRM 2013 Service Provider Planning and Deployment Guide*. This document provides instructions and steps for deploying and running hosted Microsoft Dynamics® CRM in a Microsoft® Windows Server System™ hosting environment.

The hosted Microsoft Dynamics CRM service is built around Microsoft Dynamics CRM 2015 or Microsoft Dynamics CRM 2013. By deploying a hosted Microsoft Dynamics CRM environment, service providers can offer advanced customer relationship management (CRM) functionality to business customers over the Internet.

Because deploying hosted Microsoft Dynamics is based on the Microsoft Dynamics CRM 2015 or Microsoft Dynamics CRM 2013 products, the details in this document build on the information discussed in the main [Microsoft Dynamics CRM 2015 Implementation Guide](#) or [Microsoft Dynamics CRM 2013 Implementation Guide](#), and should be considered a supplement to the main product documentation.

1.1 Who Should Read This Document

This document is intended for service provider IT personnel, system integrators, and technical consultants who may assist in the planning and deployment of hosted Microsoft Dynamics CRM.

The technical nature of a Microsoft Dynamics CRM deployment assumes Microsoft Certified Systems Engineer (MCSE)-level skills, particularly with Microsoft Exchange Server, Microsoft SQL Server®, Microsoft Windows Server®, and Microsoft Active Directory®. If you need assistance with your implementation, you may consider hiring a systems integrator that specializes in Microsoft Dynamics CRM deployments.

Upon completion of the deployment walkthrough, you should be able to confirm that you have a fully functioning hosted Microsoft Dynamics CRM environment, and are able to provision customers and users either manually or automatically (by integrating these concepts with internally developed provisioning scripts or third-party automation solutions).

1.2 Scope and Assumptions

Readers of this document should first familiarize themselves with the documentation for Microsoft Dynamics CRM. This document focuses on the special considerations and installation procedures required to deploy a hosted Microsoft Dynamics CRM environment; information that is common to an enterprise deployment of Microsoft Dynamics CRM in general is not duplicated.

For more information about the Microsoft Dynamics CRM documentation, go to the [CRM Setup & Administration Help](#) page.

This document provides guidance on how to prepare your environment and how to properly install and configure hosted Microsoft Dynamics CRM. Information about supporting components and systems is also provided.

Chapter 2

2 Deployment Overview

This deployment guide details the hosted Microsoft Dynamics CRM installation starting with the server operating system installation. Even if you have pre-existing servers, you should read this chapter carefully to ensure your current infrastructure meets the prerequisites for each server.

2.1 Summary of Deployment Process

The following flowchart helps direct you to the appropriate sections of this document.

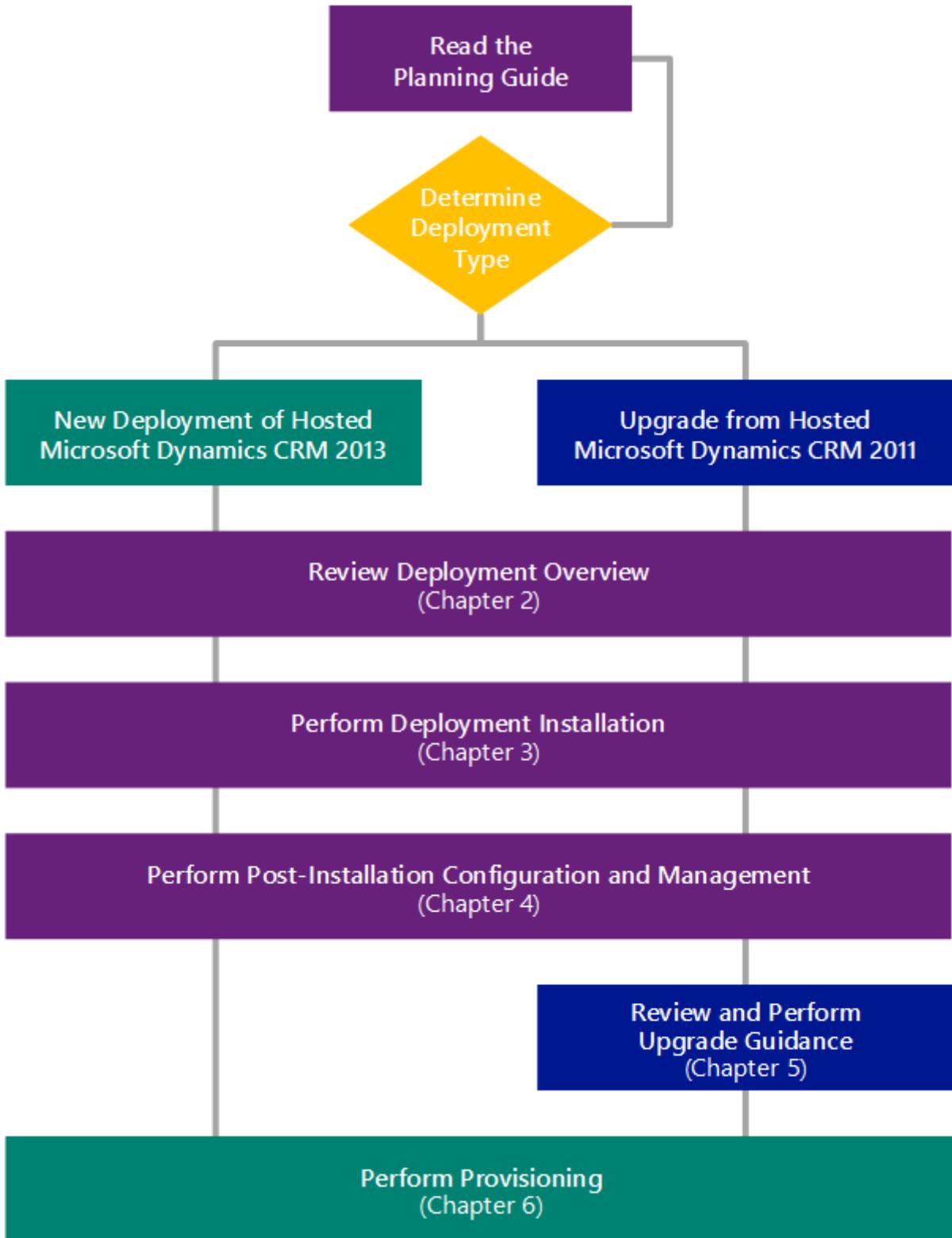


Figure 1: Flowchart indicates the appropriate sections to read in this document

The following sections provide summary descriptions of the multi-tenant deployment and upgrade process for Microsoft Dynamics CRM.

2.2 Deploying CRM as a Hosted Service

The primary focus of this document is to provide complete deployment instructions for Microsoft Dynamics CRM in a multi-tenant (hosted) environment. Because hosted Microsoft Dynamics CRM requires a variety of supporting infrastructure to be in place before the actual CRM deployment process begins, the deployment instructions reference the installation and configuration of Microsoft Active Directory, Microsoft Exchange Server, and other required servers. Only after these supporting technologies have been properly installed will you be directed to deploy the CRM-specific components.

2.2.1 Infrastructure for CRM Dynamics Hosting

Microsoft Dynamics CRM requires several software applications and components that work together to create an effective system. The majority of the system requirements for a hosted Microsoft Dynamics CRM environment are similar to the on-premises deployment of Microsoft Dynamics CRM.

Before you install hosted Microsoft Dynamics CRM, use this chapter as a guide to verify that system requirements are met and the necessary software components are available. See the pages referenced in the following list for the most current information available on supported software components, and the minimum recommendations for hardware:

Tip Throughout this document there are links to Microsoft TechNet online topics that use a version mechanism that allow you to easily switch between different versions of the same topic. To do this, select **Other Versions** near the top of the topic to select the Microsoft Dynamics CRM version that you want.

- **Microsoft Dynamics CRM 2015 and Microsoft Dynamics CRM Server 2013**
 - [Microsoft Dynamics CRM Server hardware requirements](#)
 - [Microsoft Dynamics CRM system requirements and required technologies](#)
- **Microsoft SQL Server for Microsoft Dynamics CRM Server**
 - [Microsoft SQL Server hardware requirements for Microsoft Dynamics CRM Server](#)
- **Microsoft Dynamics CRM Server-side Synchronization**
 - [Supported scenarios for server-side synchronization](#)

2.2.1.1 Active Directory Details

Microsoft Dynamics CRM uses Microsoft Active Directory to store user and group information, and application security associations. Depending on the multi-tenant Active Directory design, how organizations and users are stored and secured varies. However, there are common requirements and considerations for the Active Directory infrastructure for Dynamics CRM, which can be found at [Active Directory and network requirements for Microsoft Dynamics CRM](#).

Active Directory Federated Services (AD FS) is one of the components involved in providing claims-based authentication for Microsoft Dynamics CRM Server. You need to deploy a Security Token Service to prepare for later deploying claims-based authentication for your internet-facing deployment. You can use the Federation Service role as a security token service. To learn more about this, see:

- [Understanding the Federation Service Role Service](#)
- [Active Directory Federation Services](#)

Read more about the prerequisites for deploying claims-based authentication in "About Claims Authentication" in the *Microsoft Dynamics CRM Configuring Claims-based Authentication* white paper, available for download at <http://go.microsoft.com/fwlink/?LinkId=386527>.

2.2.1.2 SQL Server Details

How you choose to deploy SQL Server as part of your hosting infrastructure will depend on a number of factors, many of which are discussed in more detail below. Before starting to think through issues of availability and scalability, you should familiarize yourself with these SQL Server topics:

- [SQL Server requirements and recommendations for Microsoft Dynamics CRM](#)
- [SQL Server installation and configuration](#)
- [SQL Server Deployment](#)
- [Planning a SQL Server Installation](#)
- [Additional resources for SQL Server](#)

2.2.1.3 Email processing through server-side synchronization

The Server-side synchronization can be configured to connect to one or more email servers running Microsoft Exchange Server. Server-side synchronization can also connect to POP3-compliant servers to provide incoming email routing. For outgoing email, you can use SMTP and Exchange Web Services (EWS). For more information about the email server versions and protocols that Microsoft Dynamics CRM supports, see [Server-side synchronization software requirements](#).

Exchange Server is an enterprise messaging system with the versatility to support various organizations. As with Active Directory Service and Microsoft Dynamics CRM, Exchange Server requires planning before it is deployed. Many documents are available from Microsoft that explain how to plan, deploy, and operate Exchange Server. For more information, see [Additional resources for Exchange Server](#).

To begin the default deployment process for hosted Microsoft Dynamics CRM, see [Deploy the Hosted Microsoft Dynamics CRM Infrastructure](#), later in this guide.

2.2.2 High Availability in Infrastructure

In many ways, Hosted Microsoft Dynamics CRM Server deployments are similar to on-premises deployments. They can include multiple servers, which provide additional performance and scaling benefits.

Note

The Microsoft Dynamics CRM Workgroup Server does not support more than one tenant organization, and is limited to five active users. This limitation means that this edition is not a reasonable choice for a service provider implementing a multi-tenant hosting environment for Microsoft Dynamics CRM.

2.2.2.1 Front-end and Authentication Fault Tolerance

Consider how to provide fault tolerance for your front-end servers. In Microsoft Dynamics CRM Server, you can install specific server functionality, components, and services on different computers. These components and services correspond to specific server roles. For a hosting implementation, the number of front-end servers and the associated configuration details will vary depending on the total number of organizations and totally number of users the deployment needs to support. As expected in a hosted environment, the CRM deployment will serve many users across multiple tenant organizations. In addition, Service Level Agreements (SLAs) are likely in place between the service provider and customers that demand high availability from the platform.

To support SLA requirements, consider carefully your requirements for high availability and performance. Knowing how you intend to reduce the chance of a single point of failure in your architecture design will help you balance the processing load across multiple servers. With Microsoft Dynamics CRM Server, you can take advantage of Network Load Balancing to direct requests coming in from the front-end servers.

It is also possible to use hardware load balancing to offload SSL encryption. Consult with your hardware vendor about how to configure fault tolerance on your existing hardware and network infrastructure.

For information, see [Install Microsoft Dynamics CRM Server on multiple computers](#). If you plan on using Network Load Balancing, be sure the NLB has been enabled as described in [Step 4: Configure NLB for the deployment](#).

Federation provided through Active Directory Federation Services provides identity delegation so that authorized applications can impersonate their users when they access infrastructure services, even when the original users do not have local accounts. For a service provider considering a multi-forest implementation, deploying AD FS to front-end servers facilitates a single sign-on experience for users. For examples of multi-forest configurations, see [Support for Microsoft Dynamics CRM multiple-server topologies](#).

If you will use Active Directory Federation Services to operate an AD FS server farm, you could use Network Load Balancing as described in [When to Create a Federation Server Farm](#).

2.2.2.2 Fault Tolerance for SQL Server

The following SQL Server configurations are supported for use with Microsoft Dynamics CRM:

- Local
- Remote
- Mirrored
- Clustered

However, when implementing a hosted Microsoft Dynamics CRM solution, you should consider providing the benefit of high availability to customers and users through use of a fault tolerant configuration.

Although both mirrored and clustered SQL high availability configurations are supported, this guide describes use of an active/passive SQL Server cluster serving the Microsoft Dynamics CRM databases.

When working with SQL Server clusters, see the following documentation:

- [Creating a Windows Server 2012 Failover Cluster](#)
- [Understanding Requirements for Failover Clusters](#)
- [High Availability Solutions Overview](#)
- [Selecting a High Availability Solution](#)
- [SQL Server 2012 Failover Cluster Installation](#)
- [Install Microsoft Dynamics CRM Server 2013 to use a Microsoft SQL Server 2008 cluster environment](#)
- [Set configuration and organization databases for SQL Server 2012 AlwaysOn failover](#)

2.2.2.3 Fault Tolerance for Server-side synchronization

The Dynamics CRM server-side synchronization is a component in Microsoft Dynamics CRM that is used to integrate Microsoft Dynamics CRM with Exchange and POP3- or SMTP-based email servers. You can use server-side synchronization to:

- Enable email synchronization for users and queues with external email systems.
- Enable synchronization of email, appointments, contacts, and tasks from Exchange.

Server-side synchronization offers new features like efficient resource utilization, connection throttling, data migration, service isolation, error reporting, and new counters.

Server-side synchronization configuration is accomplished through CRM administration web pages hosted on the CRM Front-end Server. Fault tolerance can be achieved by one or more individual servers, a Windows cluster for high availability and failover, or multiple Windows Clusters for scaled-out highly available solution. In a hosted CRM environment, it is recommended to deploy the Front-end servers in a high availability and failover configuration using Microsoft Windows Clustering.

Visit these pages to become familiar with or to refresh your understanding of planning for high availability with Windows Server 2012:

- [Creating a Windows Server 2012 Failover Cluster](#)
- [Understanding Requirements for Failover Clusters](#)

2.2.3 Installation Privileges Requirements

Review the requirements in "Microsoft Dynamics CRM Server Setup" at [Minimum permissions required for Microsoft Dynamics CRM Setup, services, and components](#) to make sure the user account used to run Microsoft Dynamics CRM Server Setup has the necessary permissions.

2.3 Architectural Planning and Considerations

When deciding to offer Hosted Microsoft Dynamics CRM, you need to consider several questions, which will determine the architecture and size of the deployment or migration. Some of the considerations are:

- How many customers and users do you anticipate hosting?
- How much of the platform will you virtualize?
- What activities will you register as asynchronous activities in Microsoft Dynamics CRM? For example, will you set bulk email delivery or bulk imports to occur asynchronously?
- What level of support will you provide for platform and organization customizations?
- Will you deploy to a single datacenter or to multiple datacenters?
- If deploying to multiple datacenters, how will customers be allocated and provisioned?

Each of these factors will impact the overall size of the deployment. As each business needs may vary, this document will address sizing of the deployment based on tiers (Entry, Middle, and Upper), and guidance for virtualizing servers for service providers.

2.3.1 CRM Services

Microsoft Dynamics CRM consists of a number of service roles that can be run on separate servers to provide better performance and to offer improved fault tolerance. The following table introduces these roles, giving a description of the service's role and listing the server group to which the role belongs.

Table 1: CRM Service Roles

Server Role	Description	Server Group
Discovery Web Service	Finds the organization that a user belongs to in a multi-tenant deployment.	Front-end Server

Server Role	Description	Server Group
Organization Web Service	Supports running applications that use the methods described in the Microsoft Dynamics CRM Software Development Kit.	Front-end Server
Web Application Server	Runs the Web Application Server that is used to connect users to Microsoft Dynamics CRM data. The Web Application Server role requires the Organization Web Service role.	Front-end Server
Help Server	Makes Microsoft Dynamics CRM Help available to users.	Front-end Server
Asynchronous Processing Service	Processes queued asynchronous events, such as workflows, bulk email, or data import.	Back-end Server
Sandbox Processing Service	Enables an isolated environment to allow for the execution of custom code, such as plug-ins. This isolated environment reduces the possibility of custom code affecting the operation of the organizations.	Back-end Server
Email Integration Service	Sends and receives email by connecting to an external email server	Back-end Server
Monitoring Service	<p>With Microsoft Dynamics CRM 2013, the service is used to detect expired digital certificates that may affect Microsoft Dynamics CRM 2013 services that are running in the deployment. With Microsoft Dynamics CRM 2015, the Monitoring Service functionality has been increased to monitor all server roles running on the local computer and validates the health of each service. More information: Microsoft Dynamics CRM Monitoring Service.</p> <p>The Monitoring Service does not perform any other monitoring tasks and does not transmit information outside the computer where the service is running.</p> <p>The Monitoring Service is installed with the installation of any Microsoft Dynamics CRM Server role and records events under the MSCRMMonitoringServerRole source in the Event log.</p>	All server roles
Deployment Web Service	Manages the deployment by using the methods described in the Microsoft Dynamics CRM Deployment Software Development Kit.	Deployment Administration Server

Server Role	Description	Server Group
Deployment Tools	Includes Deployment Manager and Windows PowerShell cmdlets.	Deployment Administration Server
VSS Writer Service	Provides the interface to backup and restore Microsoft Dynamics CRM Server data by using the Windows Server Volume Shadow Copy Service (VSS) infrastructure	Deployment Administration Server

Service providers intending to offer hosted Microsoft Dynamics CRM services may opt to deploy the services through use of the Server Groups. However, separating the services across an architecture designed for high availability may entail further separation of the roles. Consider providing redundancy for these service roles in particular as you design your implementation:

- Web Application Server
- Asynchronous Service
- Sandbox Processing Service
- Email Integration Service
- VSS Writer Service

The CRM service accounts should have limited access in the domain, restricting them to only the necessary resources in the related CRM deployment group. If you plan to have more than one deployment group, consider establishing an account-naming scheme that is helpful in identifying relationships.

2.3.1.1 Service Principal Name Management

The Service Principal Name (SPN) attribute is a multivalued, non-linked attribute that is built from the DNS host name. The SPN is used during mutual authentication between the client and the server hosting a particular service. The client finds a computer account based on the SPN of the service to which it is trying to connect.

The Microsoft Dynamics CRM Server installer deploys role-specific services and web application pools that operate under user credentials specified during setup. To review the complete list of these roles and their permission requirements, see [Minimum permissions required for Microsoft Dynamics CRM Setup, services, and components](#).

When deploying a hosted Microsoft Dynamics CRM infrastructure, two of these roles may require additional consideration:

- Deployment Web Service
- Application Service

In web farm scenarios, as is the case for a hosted offering, the recommendation is to leave kernel-mode authentication enabled. In addition, you should closely consider using separate domain user accounts to run these services because:

- Having separate service accounts for these server roles facilitates being able to implement hardware load balancing.
- The CRM Deployment Web Service server role requires elevated permissions to provision organizations in the CRM database. If you want to adhere to a least-privileged model, the safest

approach for implementing SPNs in a hosted Microsoft Dynamics CRM infrastructure involves having the Deployment web service run under a different domain user account than the Application Service.

If you follow this suggestion to use separate domain accounts for these server roles, you should check to make sure that the SPN is correct for each account before you start Microsoft Dynamics CRM Server Setup. This will make it easier for you to set the correct SPN when necessary.

If Kernel Mode Authentication is enabled, the SPNs will be defined for the machine account, regardless of the specified service account. When implementing a web farm, Kernel Mode Authentication should be enabled and the local **ApplicationHost.config** file should be modified accordingly.

If application and deployment web services are running on the same system, and Kernel-mode authentication is disabled, you could configure both services to run under the same domain user account to prevent duplicate SPN issues. If Kernel-mode authentication cannot be enabled, install the Application and Deployment web services on separate systems. The SPNs may still need to be created manually since Kernel-mode authentication is disabled.

For more information about SPNs and how to set them, see [Service Principal Name \(SPN\) checklist for Kerberos authentication with IIS 7.0/7.5](#).

2.3.2 Network Segmentation

The reference architecture for hosted Microsoft Dynamics CRM is based on a three-tiered, four-zone approach, where the tiers define various levels of scale, and the zones illustrate the use of network segmentation to reduce the attack surface and secure data access.

The zones referenced in Microsoft Dynamics CRM are as follows:

Zone 0 - "Boundary"

- The area of the network that is closest to the Internet. Generally, this security zone contains the boundary routers, intrusion detection, first layer of denial of service (DoS) blocking, and boundary firewalls.
- Secure Sockets Layer (SSL) and initial access/certificate validation may be located at this layer. Network Operation Center (NOC) services may be logically housed in this zone.
- For Microsoft Dynamics CRM, none of its servers resides in this zone.

Zone 1 - "Edge"

- This zone contains those servers and services that provide first-level authentication, application proxy services, and load balancing across Zone 1 servers and services.
- No domain membership with the Zone 3 Active Directory Domain Services and no direct connection to servers in Zone 3 for security purposes. This reduces the attack surface.
- A "Secure by Default" approach. Locked down servers in this zone.
- Communication via secure protocols between servers in Zone 1 and Zone 2.

Zone 2 - "Proxy"

- Servers in this zone have domain membership with Active Directory in Zone 3.
- Relays or "proxies" authentication requests between Zone 1 and Zone 3.
- Two-tier services or applications make use of firewall or gateway in Zone 1 to publish secure application access in lieu of a dedicated Zone 1 or edge server.
- CRM Front-end Application Server roles reside in this zone.

- Though included in Zone 2 for the example deployment in this guide, these servers could be deployed in either Zone 2 or 3 based on your security requirements because they are not accessed by remote end users:
 - CRM Back-end Asynchronous and Sandbox Server roles reside in this zone.
 - CRM Deployment Service role server resides in this zone.
 - SQL Reporting Servers for CRM reside in this zone.

Zone 3 - "Data center"

- Most secure area of the network.
- Data repository servers reside in this zone.
- No direct access to these servers. Access is via proxies in Zone 2 or published services via firewall or gateway in Zone 1.
- CRM databases reside in this zone.

2.3.2.1 CRM Port Usage

Hosted Microsoft Dynamics CRM uses the same ports as the on-premises version. For a complete listing of which default ports are used by each CRM role, see [Network ports for Microsoft Dynamics CRM](#).

2.3.3 Internet-facing Deployment of CRM

In Microsoft Dynamics Server, configuring an internet-facing deployment depends on claims-based authentication. This means that a security token service (such as Active Directory Federation Services 2.1) must be installed. Certificate management is also important for service providers to understand.

Using federation identity technology such as Active Directory Federation Services (AD FS), Microsoft Dynamics CRM supports claims-based authentication. This technology helps simplify access to applications and other systems by using an open and interoperable claims-based model that provides simplified user access and single sign-on to applications on-premises, cloud-based, and even across organizations.

Configuring claims-based authentication and settings for an internet-facing deployment now take place as post-installation tasks. The steps to accomplish both tasks have been built into the Deployment Manager. Administrators that would prefer to script IFD configuration can do so using Dynamics CRM Windows PowerShell™ cmdlets.

Use of a wildcard certificate is recommended for Microsoft Dynamics CRM Server for hosting because each organization will be accessed using a unique host name in a common domain for the deployment. This should be a certificate provided by a known and trusted third-party certificate authority (CA). Although not required, you may simplify the certificate management by reusing the CRM wildcard certificate as the encryption certificate for the AD FS platform. However, this may not be appropriate when authenticating users from partner domains.

For more information, see "Certificate selection and requirements" in the [Microsoft Dynamics CRM Configuring Claims-based Authentication](#) white paper, available for download.

2.3.4 Deployment Groups

To assist service providers in planning to deploy a multi-tenanted hosted CRM environment, we recommend the use of a *deployment group*. A deployment group is a specific set of servers, which along with the associated security groups and service accounts, are associated with a single instance of a CRM configuration database. The hosted CRM platform for a given service provider may consist of a collection of CRM deployment groups. The number of deployment groups needed for a given hosting platform will depend on the

number of hosted organizations and on the expected number of concurrent users. Therefore, service providers can scale the CRM infrastructure by adding resources to an existing deployment group or by bringing additional deployment groups online to satisfy increasing demand.

For hosted CRM implementations, the capacity of a deployment group depends on usage scenarios like number of organizations. In general, an application server can support approximately 200 organizations with 10 users each, or 2,000 users total. Because the scalability and performance of your hosted CRM environment depends on the type of hardware, you may experience different capacity limits in terms of the number of customer organizations that can be hosted in a deployment group.

Support for email server profiles and server-side synchronization can reside in each deployment group; alternatively, a single instance can be configured as a shared service across deployment groups depending on the workload for routing email.

2.3.5 Architectural Tiers

The reference architecture is designed to support a tiered approach to implementation of hosted Microsoft Dynamics CRM services. The architecture is designed to support those hosters entering the market with plans to grow their services offers on pace with the growth of the business.

The architecture targets three design points, where the primary scale considerations are the size and number of organizations, and the number of users:

- **Entry Tier** – Based on a single deployment group that supports up to 20 organizations and 200 users with minor provisions for asynchronous workloads and customizations.
- **Middle Tier** – Based on a single deployment groups that supports up to 200 organizations and 2,000 users with moderate provisions for asynchronous workloads and customizations.
- **Upper Tier** – Based on two deployment groups that support up to 400 organizations and 20,000 users with moderate provisions for asynchronous workloads and customizations.

The hosted Dynamics CRM design defines the number of servers required for each design point. Service providers can use these examples as a starting point for planning how to grow their CRM service from one design point to the next.

Three different reference deployment architectures, using the concept of deployment groups, gives service providers a way to choose an appropriate model based on knowledge of business plans and support factors. These reference tiers assume a concurrency rate of 60%. Given those assumptions, this table compares the number of deployment groups, the estimated organizations and users, and the required hardware for each tier.

Table 2: Architectural Tier Details

Tier	Additional details	Server types, including number of processors and RAM
Entry	One deployment group Up to 20 organizations with an average of 10 users in each Total of approximately 200 users	CRM front-end servers <ul style="list-style-type: none"> ▪ Two servers ▪ Two processors with 16 GB RAM CRM Backend role / Reporting servers <ul style="list-style-type: none"> ▪ Two servers ▪ Two processors with 8 GB RAM CRM deployment service role servers <ul style="list-style-type: none"> ▪ One server ▪ Two processors with 8 GB RAM CRM database server <ul style="list-style-type: none"> ▪ Two servers ▪ Two processors with 16 GB RAM
Middle	One deployment group Up to 200 organizations with an average of 10 users in each Total of approximately 2,000 users	CRM front-end servers <ul style="list-style-type: none"> ▪ Two or more servers ▪ Two processors with 32 GB RAM CRM Asynchronous service role servers <ul style="list-style-type: none"> ▪ Two or more servers ▪ Two processors with 8 GB RAM CRM Sandbox service role servers <ul style="list-style-type: none"> ▪ Two or more servers ▪ Two processors with 8 GB RAM CRM Deployment service role servers <ul style="list-style-type: none"> ▪ One or more servers ▪ Two processors with 16 GB RAM CRM SQL Reporting servers <ul style="list-style-type: none"> ▪ Two or more servers ▪ Two processors with 8 GB RAM CRM database server <ul style="list-style-type: none"> ▪ Two servers ▪ Four processors with 32 GB RAM

Tier	Additional details	Server types, including number of processors and RAM
Upper	Two deployment groups Up to 400 organizations with an average of 50 users in each Total of approximately 20,000 users	CRM front-end servers <ul style="list-style-type: none">▪ Five or more servers per DG▪ Four processors with 16 GB RAM CRM Asynchronous service role servers <ul style="list-style-type: none">▪ Three or more servers per DG▪ Two processors with 8 GB RAM CRM Sandbox service role servers <ul style="list-style-type: none">▪ Three or more servers per DG▪ Two processors with 8 GB RAM CRM deployment service role servers <ul style="list-style-type: none">▪ Two or more servers per DG▪ Two processors with 16 GB RAM CRM SQL Reporting servers <ul style="list-style-type: none">▪ Three or more servers per DG▪ Two processors with 8 GB RAM CRM database server <ul style="list-style-type: none">▪ Two high-capacity servers per DG▪ 16 processors with 64 GB RAM

2.3.5.1 Entry Tier Architecture

The Entry Tier architecture has a single deployment group that supports up to 20 organizations or approximately 200 total users at an average 60 percent concurrency rate. It includes servers dedicated to CRM processes as well as servers running supporting infrastructure. The following figure shows the architecture for an Entry Tier deployment, including supporting infrastructure.

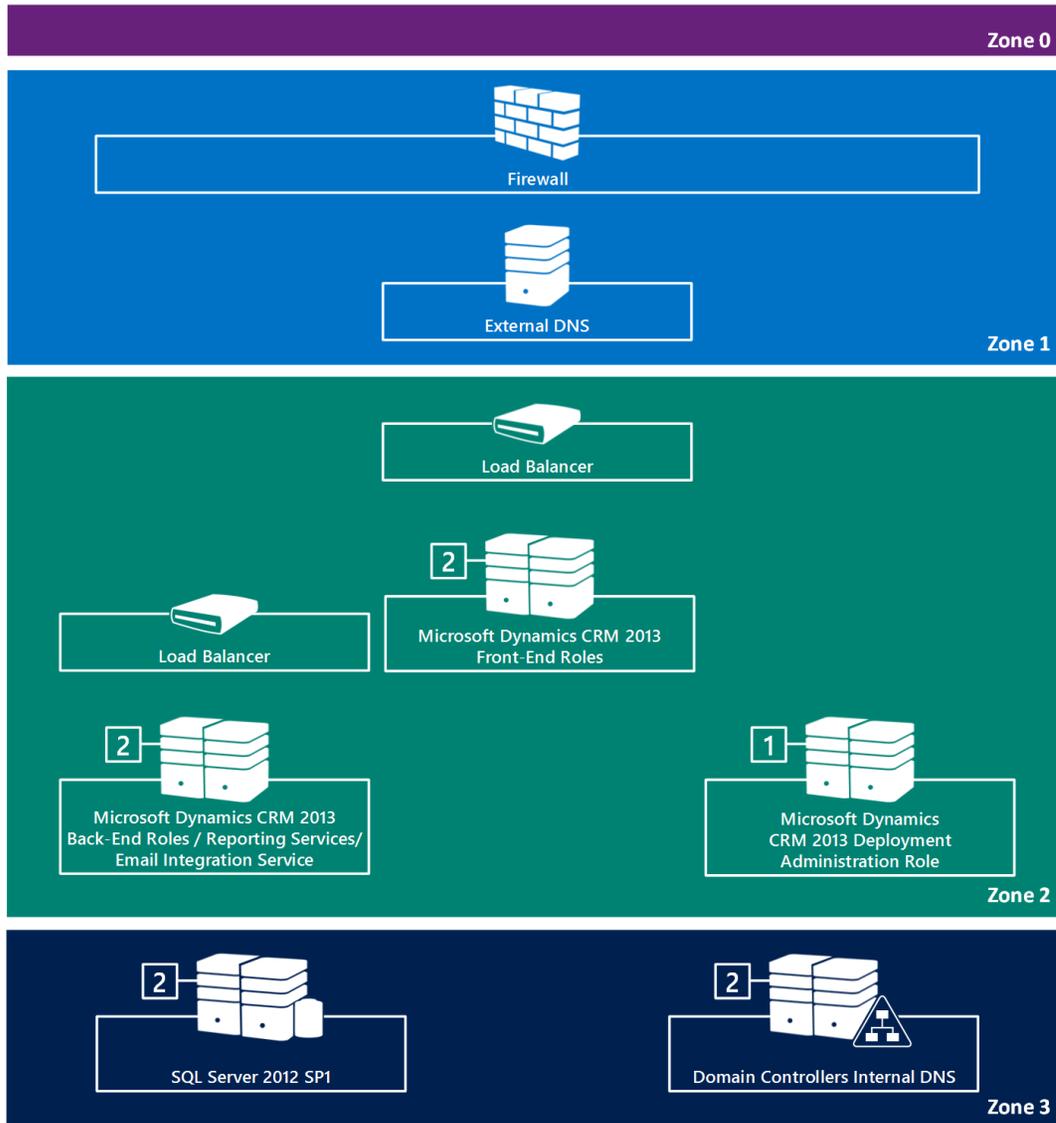


Figure 2: CRM Architecture for Entry Tier Deployment

2.3.5.2 Middle Tier Architecture

The Middle Tier architecture is designed to support an environment with a large number of organizations with relatively fewer users per organization: up to 200 organizations with an average of 10 users per organization at an average 60 percent concurrency rate. In this design, a single CRM deployment group may be used to support the expected load.

Similar to the Entry Tier model, the Middle Tier architecture includes servers dedicated to CRM processes as well as servers running supporting infrastructure.

The following figure shows the architecture for a Middle Tier deployment, including supporting infrastructure.

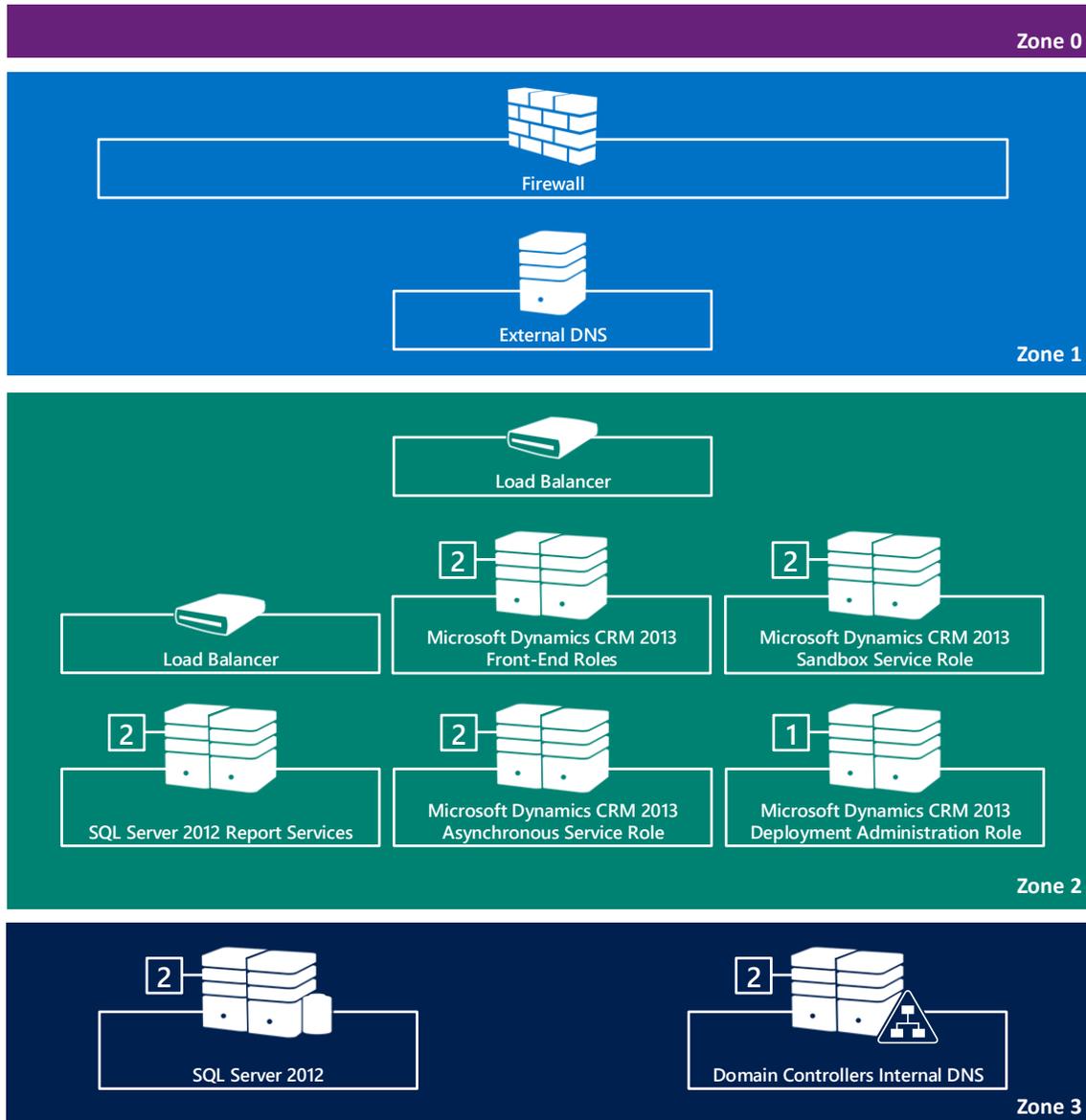


Figure 2: CRM Architecture for Middle Tier Deployment

2.3.5.3 Upper Tier Architecture

The Upper Tier architecture is designed around a multi deployment groups to support an environment with a large number of users across many organizations: up to 400 organizations at an average of 50 users per organization for a total of approximately 20,000 users with an average 60 percent concurrency rate.

Similar to the other architectural models, the Upper Tier architecture includes servers dedicated to CRM processes as well as servers running supporting infrastructure.

The following figure shows the architecture for an Upper Tier deployment, including supporting infrastructure.

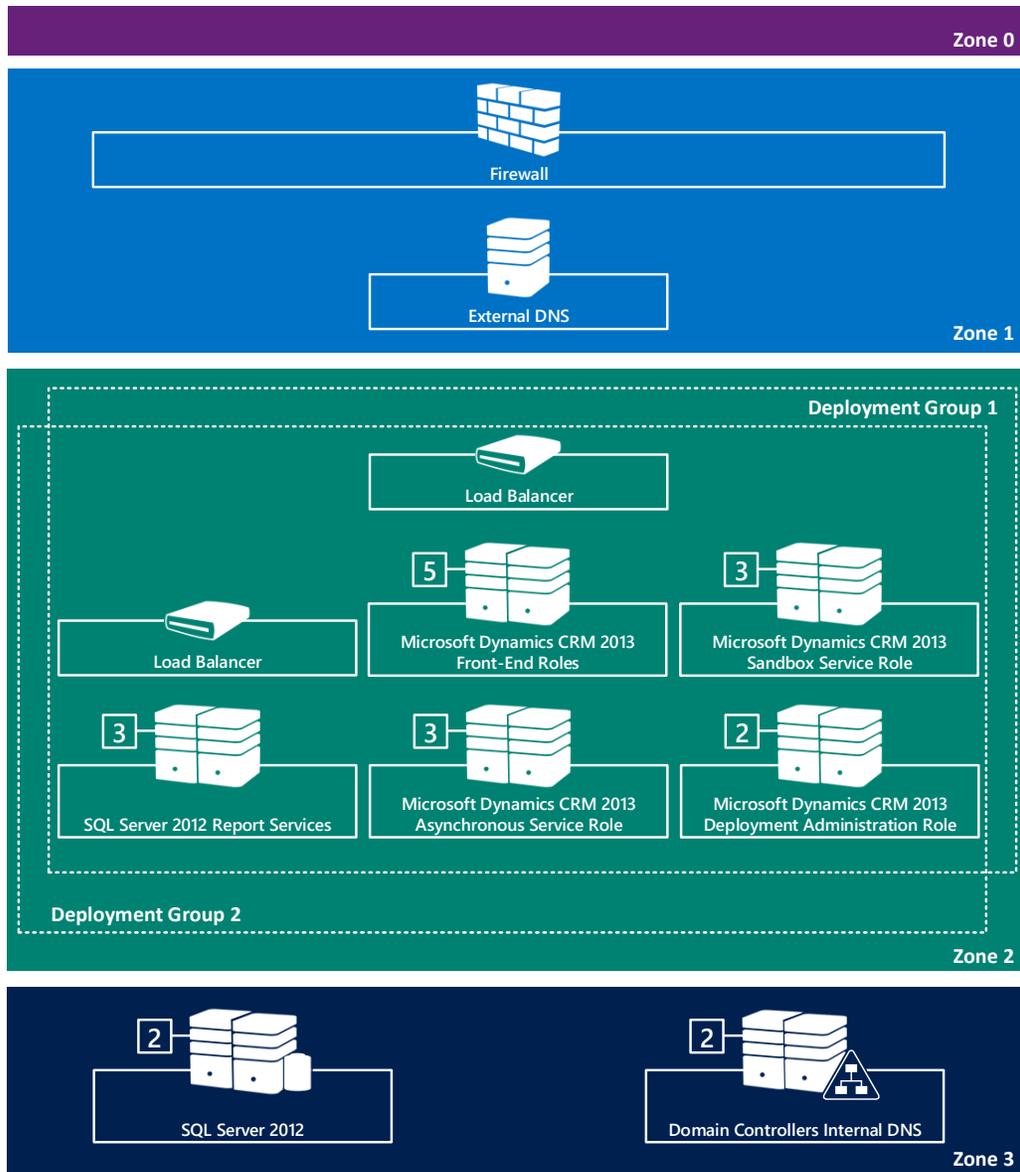


Figure 3: CRM Architecture for Upper Tier Deployment

Note

For environments with even higher numbers of concurrent users, hosted Microsoft Dynamics CRM can scale beyond what is displayed in the upper tier diagram by adding more deployment groups and/or more front-end servers in similar proportion to the increase in organizations and users.

2.3.5.4 Designing an Architecture

Always start by defining what you need from your system (such as how much data will you have? What entities will you use the most? What features will you use the most?). This is the only way to find out if performance will meet your expectations.

Use the architectural tiers as guidelines. Think of them as starting points to help you design a reference architecture that meets your specific business requirements. Once you have a deployment group design that meets your initial service offering goals, you can use it to scale out your hosted CRM service to host more organizations and customers as demand grows.

Because every business has unique needs, it is impossible to provide specific hardware recommendations for every company. However, the following list can help you understand which types of CRM activities impact the various parts of the CRM environment:

- Due to their significant boost in performance, 64-bit servers should be used throughout the environment.
- Hard disk drives on all the servers should be RAID 0 or RAID 1 (Striping and Mirroring).
- If the workflow usage is high, we recommend that you install the CRM Back-end server group on separate box(s) instead of keeping it on the same server as the CRM Front-end server group.
- If you expect your reporting usage will be high, you should consider installing SQL Reporting Services on dedicated servers in a SRS web farm configuration with clustered SRS databases.
- For high availability, consider installing duplicate CRM servers.
- A restriction on the IIS cache results in Garbage Collection starting the cleanup process on memory when this cache reaches 10 GB. This process is expensive and takes all CPU time on dual core machines until it is completed. Though there is a theoretical limit of 16GB on Front-end servers, you need to carefully consider how to balance the number of organizations and the size of the customer database because of this IIS cache constraint:
 - The more organizations you add, the greater your memory requirements will be.
 - The larger the customer database, the greater your memory requirements will be.
 - An increasing number of concurrent users is also likely to increase your memory requirements.
- The larger the customer database, the faster disk I/O system you will need on your CRM database server.

- The more users you add, the more CPUs you will need on the CRM database server. However, one large organization may require more CPU time than several small organizations with the same total number of users.

Regardless of the particular set of hardware you specify for your reference architecture, performance tuning will be required to obtain the maximum performance from your CRM environment.

2.3.6 Backup and Restore Considerations

Service providers need to plan for how to back up and restore infrastructure, services, and customer data. Such plans need to account for all server software, configurations, and customizations deployed into the CRM hosting platform. Any such plans should include all aspects of the infrastructure and platform serving the hosted customers. This includes but is not limited to Windows Server, Active Directory, Exchange, SQL Server, Dynamics CRM, AD FS, provisioning system, firewall, and load balancers.

The Microsoft Dynamics CRM VSS Writer Service provides added functionality for backup and restore of Microsoft Dynamics CRM databases through the Volume Shadow Copy Service framework. The Microsoft Dynamics CRM VSS Writer supports:

- Backup and restore of the configuration (MSCRM_CONFIG) and multiple organization (*organizationName_MSCRM*) databases.
- Databases backed up without needing to take the Microsoft Dynamics CRM application offline
- During a database restore, the application is automatically taken offline, and after successful restoration, brought back online again.

The Microsoft Dynamics CRM VSS Writer doesn't support:

- Backup and restore of Microsoft SharePoint databases that are integrated with Microsoft Dynamics CRM. For these databases, use the [SharePoint VSS Writer](#).
- Backup and restore of Microsoft SQL Server Reporting Services databases that are used for Microsoft Dynamics CRM reporting. For these databases, use the [SQL Server VSS Writer](#).

For detailed guidance and considerations on the CRM components, and configuration to include in the backup plan, see the [Backing Up the Microsoft Dynamics CRM System](#).

2.3.6.1 General Tenant Backup Requirements

While the overall recovery strategy should include plans for the entire CRM deployment, you should also consider plans and processes for recovering specific tenant organizations, their users, and their CRM organization content and customizations. The specific requirements for the plan will also depend on whether the hosted organization was deployed to shared hosting infrastructure, or is on servers dedicated only to that organization.

As a service provider, you can establish tools and templates to help you assess a tenant's backup and recovery requirements based on your service offerings. These might include:

- Checklists to review with customers before provisioning their organization into your shared or dedicated hosting platform.
- Script templates designed to automate creating and maintaining backups on a daily, weekly, or monthly basis.
- Service level agreements to communicate how quickly customer data can be made available in the case of unexpected system failure.

- If you have integrated with an automated provisioning system that stores stateful information relevant to the organization, users, or CRM site, that information should be included in a per tenant recovery plan.

2.3.6.2 Tenant Backup and Business Cycles

When developing the plan, consider the tenant on-boarding process, and how it may leverage the same processes as restoring a CRM site data and customizations. Conversely, consider how the cancellation of service by a tenant could leverage the backup process, as they will likely want a copy of all the data and customizations relevant to their organization CRM site(s).

Server-side synchronization in Microsoft Dynamics CRM enables you to centrally manage mailboxes and profiles, configure email for users and queues, and track email processing errors. If you've been using the Email Router but now want to use server-side synchronization, you can use the migration wizard to quickly move the configuration settings into Microsoft Dynamics CRM.

In addition to setting up email, you can use server-side synchronization to synchronize appointments, contacts, and tasks from Microsoft Exchange Server.

Server-side synchronization connects Microsoft Dynamics CRM with one or more Exchange servers (or POP3 servers) to set up incoming email and one or more SMTP or Exchange servers to set up outgoing email.

2.3.7 Planning for Email Processing using Server-Side Synchronization

2.3.7.1 What server-side synchronization provides

If you're an administrator, you can use server-side synchronization to:

- Enable incoming and outgoing email for users and queues (through personal mailboxes).
- Monitor and report errors regarding email to users and email server profile owners.
- Enable contact, task, and appointment synchronization from Exchange.

2.3.7.2 How server-side synchronization works

When Microsoft Dynamics CRM is set up, a mailbox record is created for each user and for each queue in the organization. These mailboxes are associated with an email server profile that defines the settings for connecting to the email server. Server-side synchronization uses this information to process email for all mailboxes for a specific email server profile.

If your organization has a large number of mailboxes to monitor, you can consider using a forward mailbox to reduce the administrative effort.

After you create the email server profile and associate the mailboxes, you must test incoming and outgoing email, and appointment, task, and contact synchronization, and enable the mailboxes for email processing. With server-side synchronization, you can do all this from within Microsoft Dynamics CRM. Once configured, incoming email messages are tracked in Microsoft Dynamics CRM and stored as activity records, and email messages created in Microsoft Dynamics CRM are also sent. The email activity records in Microsoft Dynamics CRM include the contents of the email message—such as the text of the message and its subject line—and also relevant associations with other Microsoft Dynamics CRM records. For example, when a salesperson replies to a customer about a case, an email activity record is created that includes the text of the message, as well as the information associating the email activity record with the correct case record.

Chapter 3

3 Deployment Installation

This section first introduces the server roles and associated software along with the fictitious names used later in the deployment procedures. The remainder of this section takes you through an example of a greenfield deployment, an installation and configuration of a network where none existed before, for a hosted CRM platform using a middle tier architecture design.

3.1 Example Names

This section provides an overview of the server and customer organization names used throughout the remainder of this guide.

3.1.1 Server Names, Roles, and Associated Software

The guidance for deploying hosted Microsoft Dynamics CRM requires the use of a consistent set of server and domain names. The following table shows the default names for servers used in the documentation and the required software for each server. For more information about the roles used in this documentation, see [Microsoft Dynamics CRM server roles](#).

This document will outline the deployment of the systems, role groups, and individual roles per system as noted in the following table. Depending upon the requirements of your hosted CRM solution, you may choose to combine some roles or further separate out individual roles. However, all server roles must be installed and running in the CRM deployment to provide a fully functioning system.

Table 3: CRM Hosting Solution Servers

Server name	Role	Installed software
AD01	Domain controller for the service provider domain Global catalog server Internal DNS server	Microsoft Windows Server 2012, Standard Edition
DNS01	External DNS Server	Microsoft Windows Server 2012, Standard Edition
CRMFE01	CRM Front-end Server, including these individual server roles: <ul style="list-style-type: none"> ▪ Discovery Web Service ▪ Organization Web Service ▪ Web Application Server ▪ Help Server 	Microsoft Windows Server 2012, Standard Edition

Server name	Role	Installed software
CRMFE02	CRM Front-end Server, including these individual server roles: <ul style="list-style-type: none"> ▪ Discovery Web Service ▪ Organization Web Service ▪ Web Application Server ▪ Help Server 	Microsoft Windows Server 2012, Standard Edition
CRMAS01	CRM Asynchronous Processing Service	Microsoft Windows Server 2012, Standard Edition
CRMAS02	CRM Asynchronous Processing Service	Microsoft Windows Server 2012, Standard Edition
CRMSP01	CRM Sandbox Processing Service	Microsoft Windows Server 2012, Standard Edition
CRMSP02	CRM Sandbox Processing Service	Microsoft Windows Server 2012, Standard Edition
CRMDEP01	CRM Deployment Administration Server	Microsoft Windows Server 2012, Standard Edition
CRMSQL	CRM SQL Server Cluster Name	
CRMSQL01	CRM SQL Server	Microsoft Windows Server 2012, Standard Edition SQL Server 2012 with SP1
CRMSQL02	CRM SQL Server	Microsoft Windows Server 2012, Standard Edition SQL Server 2012 with SP1
CRMREP01	SQL Reporting Server	Microsoft Windows Server 2012, Standard Edition
CRMREP02	SQL Reporting Server	Microsoft Windows Server 2012, Standard Edition
Client01	Client computer	Microsoft Windows 7 or Windows 8 Microsoft Outlook® 2013
PKIROOT	Root Certificate Authority Certificate issuing server [See note below for more details.]	Microsoft Windows Server 2012, Standard Edition
ADFSWEB01	AD FS Web Front-end	Microsoft Windows Server 2012, Standard Edition Active Directory Federation Services 2.1

Server name	Role	Installed software
ADFSWEB02	AD FS Web Front-end	Microsoft Windows Server 2012, Standard Edition Active Directory Federation Services 2.1 You might also consider using the AD FS Proxy server role.

Note

PKIROOT is used only when internal domain certificates are needed to protect internal web interfaces. All public-facing external interfaces should be protected using certificates provided by a known and trusted third-party certificate authority (CA) to simplify access by end users, and reduce client-side system modifications.

You cannot explicitly select the SQL Server "role" for installation during Microsoft Dynamics CRM Server Setup. CRM sets this logical role when you specify a particular instance of SQL Server, either local or on another computer (recommended) for use in the Microsoft Dynamics CRM deployment.

Use one of the following options to install server roles:

- Run the **Microsoft Dynamics CRM Server Setup Wizard** to select one or more server role groups or one or more individual server roles. Certain prerequisites are installed based on the server role selected. For example, if **CRM Application Front End Server** is selected, IIS will be installed. However, if you later remove that role from a server, IIS will be left in place because other applications may be using it.
- If Microsoft Dynamics CRM Server is already installed, you can use **Programs and Features** in Control Panel to add or remove server roles.
- Create an XML Setup configuration file specifying one or more individual server roles or a server role group and run **SetupServer.exe** at the command prompt. For more details, see [Scripting Deployment Installations with Configuration Files](#).

This document will instruct the use of the Microsoft Dynamics CRM Server Setup Wizard utility for installing the CRM services. Although not part of the procedures, you may choose to install via command line and the XML Setup configuration files.

3.1.2 Claims-based Authentication Considerations

Active Directory Federation Services is a highly secure, highly extensible, and Internet-scalable identity access solution that allows organizations to authenticate users from partner organizations. Using Active Directory Federation Services in Windows Server, you can simply and very securely grant external users access to your organization's domain resources. AD FS can also simplify integration between untrusted resources and domain resources within your own organization.

3.1.3 Example Domain Names

The following table lists the domain names of fictitious companies that are used as examples in the documentation. During your deployment, you will want to use the appropriate DNS name for your environment.

Table 4: DNS Domain Names

DNS name	Description
contoso.com	Active Directory domain for the service provider
consolidatedmessenger.com	Reseller domain
alpineskihouse.com	Customer domain

3.2 Deploy the Hosted Microsoft Dynamics CRM Infrastructure

This section provides guidance on how to prepare an infrastructure required for a hosted Microsoft Dynamics CRM fault-tolerant deployment. As your environment and requirements may differ, make the appropriate adjustments for the numbers of servers and where certain components are installed.

3.2.1 Prepare the Active Directory Forest Domain Infrastructure

You must build and deploy the first domain controller and establish the Active Directory forest and domain before you add the other infrastructure components.

To prepare your environment, build and deploy the first domain controller to establish the domain and internal domain name services. Then, add one or more additional domain controllers as necessary for the environment.

This guidance for hosted Microsoft Dynamics CRM references two domain controllers, AD01 and AD02, providing directory services for an internal contoso.com domain.

For more details on the Active Directory considerations, see the following documentation:

- [Active Directory Requirements](#)
- [Active Directory modes](#)

3.2.2 Build and Deploy the External DNS Server

One of the core infrastructure components for hosted Microsoft Dynamics CRM is a DNS server, either a Microsoft DNS server or a compatible version; it does not need to be a Microsoft DNS server.

The Internet-facing deployment (IFD) of CRM in this solution requires publicly resolvable DNS domain and host entries for the following systems/sites. A risk for namespace conflict exists if this domain is used for other shared services. To mitigate potential naming conflicts, the example deployment in this guide makes use of subdomains to create unique names for hosted Dynamic DRM services, such as:

- CRM IFD domain/subdomain (for example, crm.consolidatedmessenger.com)
- CRM web server host (for example, host1.crm.consolidatedmessenger.com)
- CRM SDK/Platform server host (for example, sdk.crm.consolidatedmessenger.com)
- CRM Report server host (for example, reports.crm.consolidatedmessenger.com)
- Per hosted organization CRM site host (for example, alpineskihouse.crm.consolidatedmessenger.com)

As you may notice from the preceding list, all sites in a Hosted CRM deployment must share a common external domain name (that is, crm.consolidatedmessenger.com). Every hosted CRM site will be accessed using a unique fully qualified domain name consisting of the CRM site name and the common external domain name. For example, if the CRM deployment is configured for an external domain of crm.consolidatedmessenger.com, and a CRM site is provisioned with the name of AlpineSkiHouse for the

Alpine Ski House customer organization, their users would access the CRM site via the following URL:
<https://alpineskihouse.crm.consolidatedmessenger.com>.

Important

There are several names that cannot be used to name an organization. To view a list of reserved names, open the **dbo.ReservedNames** table in the MSCRM_CONFIG database, and review the names in the **ReservedName** column.

When selecting a DNS solution, consider the ability to provision the DNS host records during the CRM site provisioning actions. An automated provisioning solution that is capable of connecting to and provisioning external DNS resources is recommended for hosted CRM.

If an external DNS solution is not available for the hosted CRM platform, it should be built now. For those who are deploying to a completely new environment and want further details on DNS services, see [Domain Name System \(DNS\) Overview](#). At this point, you should decide on the shared external domain name for the CRM deployment. In addition, this DNS zone should be defined in the external name servers.

This guide references an external domain name for the Hosted CRM deployment as `crm.consolidatedmessenger.com`. It also references an external domain name server with a machine name of `DNS01`.

3.2.3 Determine the Multi-tenancy Design

Hosted Microsoft Dynamics CRM must be deployed within an Active Directory platform that has been configured for multi-tenancy. However, this guide does not provide directions for implementing multi-tenancy within Active Directory.

Before proceeding with the hosted CRM deployment, you should determine the multi-tenancy design and implementation plans. The Active Directory and multi-tenant infrastructure must be built before proceeding with the hosted CRM deployment.

3.2.4 Build and Deploy the Messaging Platform

One of core supporting infrastructure components for hosted Microsoft Dynamics CRM is an email server that performs automated CRM email routing and tracking tasks. Although the POP protocol is supported by CRM, this documentation assumes that the email integration will be performed through a Microsoft Exchange Server infrastructure. It also assumes the Exchange Server has been built using the same forest and domain as that planned for the hosted Dynamics CRM solution. For more information, see [Set up email through server-side synchronization](#).

Before proceeding with deploying Microsoft Dynamics CRM and integrating with your existing Exchange Server platform, verify that:

- Your messaging platform is functioning properly.
- Email is routing in and out of the platform.
- Users are able to connect successfully and authenticate via the Outlook Web Access client, as well as use Exchange AutoDiscover to connect their desktop Outlook client.

3.2.5 Deploy Federation and Claims-based Authentication Platform

When you configure Microsoft Dynamics CRM for Internet-facing access, Microsoft Dynamics CRM requires federated services that support claims-based authentication. If you do not already have an existing Secure Token Services (STS) solution for federation and claims-based authentication in the environment, one must to be deployed prior to configuring Microsoft Dynamics CRM as an Internet-facing deployment.

Active Directory Federation Services is a recommended STS solution, and the one used in this guide. This guide for hosted Microsoft Dynamics CRM references two AD FS 2.1 systems, ADFSWEB01 and ADFSWEB02, which serve as the federation web front-ends.

3.2.5.1 Prepare the AD FS 2.1 Platform

Perform the following steps to prepare the AD FS 2.1 platform for integration with Microsoft Dynamics CRM. For detailed instructions, see [Implementing Claims-based Authentication - Internal Access](#).

1. Create a Federation Service domain user account (such as **CONTOSO\ADFSServiceAcct**) that will be configured to run the Windows services on all servers in the farm.
2. Identify the Federation Service name or URL (that is, sts.crm.consolidatedmessenger.com) that will be used by the Federation Service website. Users will be redirected to this URL, and displayed a forms-based authentication page when signing into their CRM site.

Important

If you are sharing the same DNS namespace for the Federation Service name and the CRM deployment common domain name (that is, *.crm.consolidatedmessenger.com), consider a host name that will not conflict with potential tenant CRM organization names.

3. Install the Federation Service certificate into IIS. This should be a certificate provided by a known and trusted third-party certificate authority (CA). If you are sharing the same DNS namespace for this service, you may use the [same wildcard certificate](#) planned for use on the CRM front-end web servers. This guide assumes the use of the same wildcard certificate.
4. Deploy the AD FS on the first front-end server (ADFSWEB01) to create the new Federation Service farm. In Windows Server 2012, you install the AD FS server role using Server Manager. Server Manager provides improved AD FS configuration wizard pages that perform server validation checks before you continue with the AD FS server role installation. It will automatically list and install all the services that AD FS depends on during the AD FS server role installation. For more information, see [AD FS Federation Server Configuration Wizard](#).
5. Deploy additional front-ends to the farm for fault tolerance.
6. Configure load balancing of the AD FS web services.
7. Configure the Internet firewall to allow inbound traffic on the ports used for the AD FS web services. By default, the configuration would enable HTTPS (SSL) over port 443 to the load-balanced interface.
8. Add a DNS host record for the Federation Service name in the external DNS zone, pointing to the firewall listener for the federation web services.
9. Verify the AD FS 2.1 installation by browsing to the federation metadata URL from internal and Internet-based client.

For more details, see [Active Directory Federation Services Overview](#).

3.3 Deploy Hosted Microsoft Dynamics CRM Deployment Group Components

This section of the documentation provides detailed instructions for deploying the CRM-specific components of the solution, such as the CRM database, and front-end and back-end servers. These systems will be deployed in a deployment group for scale purposes. As the platform scales up, add additional servers to the deployment group or additional deployment groups.

Note

As noted earlier, a deployment group is a specific set of servers that function as a single unit to support a defined hosted CRM workload. By bringing additional deployment groups online to satisfy demand, service providers can scale their CRM infrastructure as their customer base increases. For more information about CRM deployment groups, see [Deployment Groups](#).

3.3.1 Deploy Hosted Microsoft Dynamics CRM Database Server

This section provides summary descriptions of procedures and links to detailed procedures, as well as step-by-step guidance for where we deviate from the on-premises Dynamics CRM deployment.

3.3.1.1 Prepare the CRM Database Server

1. Prepare the fault tolerant SQL Database environment as desired for SQL Clustering (or Mirroring), CRMSQL01 and CRMSQL02.
2. Prepare the hardware.
3. Deploy the base OS and configure networking.
4. Join the Active Directory Domain.
5. Install Windows Cluster Services, if desired, and verify that the shared disk resources are available.

3.3.1.2 Create the CRM SQL Service account

1. Create a domain user account for the SQL services, such as **CRMSQLService**.
2. Consider creating unique accounts for each CRM deployment group to limit the scope of rights for the account across the domain systems.
3. Ensure this account has a secure (non-blank) password.
4. Ensure the password for this account is not set to expire or a process in place to manage the password changes if you have a password expiration policy.
5. Add the CRM SQL Service account as a local administrator on the CRM SQL database servers.

3.3.1.3 Install SQL Server 2012

1. Identify the [SQL Server 64-bit version and edition to be used for Dynamics CRM](#). For specific versions supported by Dynamics CRM, see [SQL Server editions](#) and [SQL Server Reporting Services](#).
2. Install the following SQL Server services on the database servers:
 - ▶ SQL Server Database services
 - ▶ SQL Agent Service
 - ▶ SQL Full-Text Search Service
 - ▶ Management Tools
3. Configure the \Program Files\ and \Data Files\ locations as desired.

Note

For performance reasons, you should store the SQL Server program files on a different hard disk than the data. For example, for the program files specify drive **C:** and for the data files specify drive **D:**. You should use high-performance drives; using RAID is recommended.

4. Configure the SQL services to run under the domain account previously created, **CRMSQLService**.
5. Configure the Authentication Mode for Windows Authentication only.

3.3.1.4 Configure SQL Server Service Startup

- Verify that the **SQL Server Agent** and **SQL Server Full-Text Search Service** are configured to start up automatically. Reconfigure to **Automatic startup** if necessary.

3.3.1.5 Prepare the CRM Reporting Server

1. Prepare the CRM reporting servers CRMREP01 and CRMREP02.
2. Prepare the hardware.
3. Deploy the base OS and configure networking.
4. Join the Active Directory Domain.

3.3.1.6 Install SQL Server 2012 Reporting Services

1. Identify the SQL Server Reporting Services 64-bit version and edition to be used for CRM. For specific versions supported, see [SQL Server Reporting Services](#).
2. To scale-out the Reporting Services deployment on a network load balanced (NLB) cluster, you should configure the NLB cluster before you configure the scale-out deployment. For more information, see [Configure a Report Server on a Network Load Balancing Cluster](#).
3. Install the SQL Server Reporting Services server, making sure to configure the SQL Reporting Services to run under the domain account previously created (such as **CRMSQLService**).
4. Choose the **Install but do not configure server** option on the **Report Server Installation Options** page.
5. For more details on deploying a scaled-out Report Server farm, see [Configure a Native Mode Report Server Scale-Out Deployment](#).

3.3.1.7 Configure SQL Server Reporting Services

1. Configure **SQL Server Reporting services** for the CRM SQL database instance previously created. If you created a SQL Server cluster, the database name is the SQL Cluster virtual server name.
2. Unless preferred otherwise, the default options can be selected throughout the configuration wizard.
3. Verify connectivity to the SQL Reporting Services database through each reporting server as well as the load balanced IP address or fully qualified domain name.
4. Verify that the SQL Reporting Service is also configured to start up automatically. Reconfigure it to **Automatic startup** if necessary.

3.3.2 Deploy the CRM Front-end Servers

This section describes installing servers in the front-end server group.

3.3.2.1 Prepare the CRM Front-end Servers

1. Prepare the fault-tolerant front-end servers, CRMFE01 and CRMFE02.
2. Prepare the hardware according to the instructions in the [Install Microsoft Dynamics CRM Server on multiple computers](#).
3. Deploy a [supported version of Windows Server 2012](#) as the base operating system.
4. Configure networking.
5. Join the Active Directory domain.

3.3.2.2 Prepare Active Directory for Microsoft Dynamics CRM

As part of the installation of Microsoft Dynamics CRM, the setup program requires the input of a domain organization unit in which the CRM security objects will be created. Although this organization unit may be any container in the domain hierarchy, it is recommended that you define a dedicated container for these objects for manageability. Also, you should consider locating this container in the domain hierarchy to limit access to other domain resources for CRM services and functionality enabled under these credentials.

For example, you may create an organization unit named “CRM Security Groups,” and select that container during the CRM setup procedure. Upon install, CRM will create four security groups for the CRM deployment in that organizational unit. Each security group will contain the GUID of the CRM deployment as part of the name.

If you plan to install multiple CRM deployment groups in the domain, mapping the security groups to the CRM deployment group may be challenging. Consider performing the following:

- Prior to installing CRM, create unique organization units for each is CRM deployment group, to isolate the security objects during the install. Each container will include only the security groups related to that CRM deployment group.
- After installing CRM, update the description of the security groups with an easily identifiable value to create the mapping. This value is then displayed in **Active Directory Users and Computers** for easier identification.

3.3.2.3 Create the CRM Application Service Account

When deploying multiple Microsoft Dynamics CRM servers in a load balanced configuration, the CRM Application service (CRMAppPool) must run as a domain user account. Use the following procedure to create a domain user account for the CRM Application Processing service:

1. Create a domain user account for the CRM Application service, such as **CRMAppSvc**.
 - a. Consider creating unique accounts for each CRM deployment group to limit the scope of rights for the account across the domain systems.
 - b. Ensure this account has a secure (non-blank) password.
 - c. Ensure the password for this account is not set to expire or a process in place to manage the password changes if you have a password expiration policy.
2. Add the CRM Application service account to the **Performance Log Users** group on the CRM Application servers.
3. Add the CRM Application Service account to the local **Administrators** group on the:
 - a. CRM Application server(s)
 - b. CRM SQL Servers

For more details on the permissions required for the Microsoft Dynamics CRM Application Service, see [Minimum permissions required for Microsoft Dynamics CRM Setup, services, and components](#).

3.3.2.4 Install the First CRM Front-end Server

When installing the first CRM Front-end server, the setup utility installs all the local system software and components, and creates the configuration database on the specified SQL database instance and the related database components. When additional CRM Front-end systems are installed, the installation option to connect to an existing deployment should be used, leveraging the same central configuration database of the deployment group.

For guidance on installing the first CRM front-end server, refer to [Install Microsoft Dynamics CRM Server on a server without Microsoft Dynamics CRM installed](#).

1. Login to CRMFE01 as a domain administrator.
2. Locate the Microsoft Dynamics CRM Server installation media, and execute **SetupServer.exe** and proceed through the wizard, making note of the following sections.
 - a. On the **Specify Server Roles** page, ensure that only the **Front End Server** role is selected for install. You may optionally select the **Deployment Tools** role, if you want to be able to manage the deployment locally.
 - b. On the **Specify Deployment Options** page, in the **Enter or select the name of the computer that is running SQL Server to use with the deployment**, enter the SQL database (virtual) server name and select **Create a new deployment**.
 - c. On the **Select the Organizational Unit**, browse **Active Directory** and select the container where the Microsoft Dynamics CRM security groups should be created.
 - d. On the **Specify Security Accounts** page, for the **Application Service**, choose the **CRMAppSvc** account.
 - e. On the **Select a Web Site** page, choose **Select a Web Site** and select the default website listed running on port 80.
 - f. On the **Specify E-mail Router Settings** page, the following options are available:
 - If the Email Router will be used for CRM email processing, in the **Email router server name** field, type the name of the computer where the email router will be installed. If the server is not yet joined to the domain, you can leave the field blank at this time.
 - If Server-side Synchronization will be used for email processing, this step can be skipped.
 - g. On the **Specify Reporting Services Server** page, enter the **Report Server URL**. If you have deployed Reporting Services in a scale-out deployment and using load balancing, the hostname of the URL should point to the load balanced IP.
 - h. On the **Select Microsoft Update Preference** page, choose **I don't want to use Microsoft Update**.
3. Complete the installation wizard.

Note

This setting is recommend because for most service providers updates to the production hosted platform are strictly controlled and only applied after testing the patch to verify installing it will not introduce service issues.

3.3.2.5 Deploy Additional Front-end Servers

When deploying additional CRM servers to the deployment group front-end farm, the option to connect to an existing deployment must be used. Perform the following steps on each additional CRM Front-end server in the deployment.

1. Login to CRMFE02 as a domain administrator.
2. Locate the Microsoft Dynamics CRM Server installation media, execute **SetupServer.exe**, and then proceed through the wizard, making note of the following sections.
 - a. On the **Specify Server Roles** page, ensure that only the **Front End Server** role is selected for install. You can optionally select the **Deployment Tools** role, if you want to be able to manage the deployment locally.

- b. On the **Specify Deployment Options** page, in the **Enter or select the name of the computer that is running SQL Server to use with the deployment**, enter the SQL database (virtual) server name and select **Connect to, and if necessary, upgrade an existing deployment**.
 - c. On the **Specify Security Accounts** page, for the **Application Service**, choose the **Network Service** account.
 - d. On the **Select a Web Site** page, choose **Select a Web Site** and then select the default website listed running on port 80.
 - e. On the **Specify E-mail Router Settings** page, the following options are available:
 - If the Email Router will be used for CRM email processing, in the **Email router server name** field, type the name of the computer where the email router will be installed. If the server is not yet joined to the domain, you can leave the field blank at this time.
 - If Server-side Synchronization will be used for email processing, this step can be skipped.
 - f. On the **Select Microsoft Update Preference** page, choose **I don't want to use Microsoft Update**. Updates to the production-hosted platform should be strictly controlled and only applied after (1) approved by Microsoft for deployment in a hosting environment, and (2) validated in a test environment.
 - g. Complete the installation wizard.
3. Repeat the preceding steps on all additional CRM Front-end servers in the deployment group.

3.3.2.6 Install Microsoft CRM Reporting Extensions on SRS Servers

After you install Microsoft Dynamics CRM Server, you must install the Microsoft Dynamics CRM Reporting Extensions to create, run, and schedule reports in Microsoft Dynamics CRM.

Note

Only one instance of the Microsoft Dynamics CRM Reporting Extensions can be deployed on a server, which means an SRS server can only be bound to a single CRM deployment. However, a CRM deployment may use multiple SRS instances or farms.

For instruction details, see the [Install Microsoft Dynamics CRM Reporting Extensions](#) section.

3.3.3 Install the Back-end Servers

This section describes installing servers in the back-end server group.

3.3.3.1 Create the CRM Asynchronous Processing Service Account

Use the following procedure to create a domain user account for the CRM Asynchronous Processing service

1. Create a domain user account for the CRM Asynchronous service, such as **CRMAsyncSvc**.
 - a. Consider creating unique accounts for each CRM deployment group to limit the scope of rights for the account across the domain systems.
 - b. Ensure this account has a secure (non-blank) password.
 - c. Ensure the password for this account is not set to expire, or that a process is in place to manage the password changes if you have a password expiration policy.
2. Add the CRM Asynchronous service account to the **Performance Log Users** group on the CRM Asynchronous servers.

For more details on the permissions required for the Microsoft Dynamics CRM Asynchronous Processing Service and Microsoft Dynamics CRM Asynchronous Processing Service (maintenance) services, see [Minimum permissions required for Microsoft Dynamics CRM Setup, services, and components](#).

3.3.3.2 Deploy Asynchronous and Email Integration Service

The Asynchronous Service role processes queued asynchronous events such as workflows, bulk email, or data import. The Email Integration Service role handles sending and receiving of email messages by connecting to an external email server using Server-side Synchronization. By deploying a separate instance of the asynchronous and email integration service, hosters can improve system performance and isolate the asynchronous activities from the client real-time activities. Furthermore, adding multiple Asynchronous Service servers can significantly improve fault tolerance and further boost performance.

The Asynchronous Service role can be deployed by itself. These servers become fault tolerant by simply having the service running on multiple hosts. This is also helpful in improving performance as asynchronous processes and workflows may consume more resources.

1. Login to CRMAS01 as a domain administrator.
2. Locate the Microsoft Dynamics CRM Server installation media, and execute **SetupServer.exe**, and then proceed through the wizard, making note of the following sections.
 - a. On the **Specify Server Roles** page, select both the **Asynchronous Processing Service** and **Email Integration Service** roles for installation.
 - b. On the **Specify Deployment Options** page, in the **Enter or select the name of the computer that is running SQL Server to use with the deployment**, enter the SQL database (virtual) server name and select **Connect to, and if necessary, upgrade an existing deployment**.
 - c. On the **Specify Security Accounts** page, for the **Asynchronous Service**, choose the **CRMAsyncSvc** account.
 - d. On the **Select Microsoft Update Preference** page, choose **I don't want to use Microsoft Update**. Updates to the production-hosted platform should be strictly controlled and only applied after (1) approved by Microsoft for deployment in a hosting environment, and (2) validated in a test environment.
 - e. Complete the installation wizard.
3. Repeat the preceding on all additional CRM Asynchronous Processing Back-end servers in the deployment group.

3.3.3.3 Create the CRM Sandbox Processing Service Account

Use the following procedure to create a domain user account for the CRM Sandbox Processing service.

1. Create a domain user account for the CRM Sandbox service, such as **CRMSandboxSvc**.
 - a. Consider creating unique accounts for each CRM deployment group to limit the scope of rights for the account across the domain systems.
 - b. Ensure this account has a secure (non-blank) password.
 - c. Ensure the password for this account is not set to expire, or that a process is in place to manage the password changes if you have a password expiration policy.
2. Add the CRM Sandbox service account to the **Performance Log Users** group on the CRM Sandbox servers.

For more details on the permissions required, see the "Microsoft Dynamics CRM Sandbox Processing Service" section at [Minimum permissions required for Microsoft Dynamics CRM Setup, services, and components](#).

3.3.3.4 Deploy Sandbox Processing Servers

The Sandbox Processing Service enables an isolated environment to allow for the execution of custom code, such as plug-ins. This isolated environment reduces the possibility of custom code affecting the operation of the organizations in the production Microsoft Dynamics CRM deployment.

Consider separating the Sandbox role from the Async role. This security best practice helps to ensure that custom code does not impact workflows or other asynchronous processes. Because you are deploying a multi-tenant environment, it is critical to prevent one customer's custom code bringing down other customer accounts--or hanging the entire platform.

We recommend that the Sandbox Processing Service role be installed onto a dedicated server on a separate virtual LAN (VLAN) from other computers that are running Microsoft Dynamics CRM roles. This network isolation strategy can help protect other Microsoft Dynamics CRM resources from being compromised if there is a malicious plug-in running in the sandbox.

3.3.3.4.1 Installing a Sandbox Processing Server

1. Log in to CRMSP01 as a domain administrator.
2. Locate the Microsoft Dynamics CRM Server installation media, and then execute **SetupServer.exe** and proceed through the wizard, making note of the following sections.
 - a. On the **Specify Server Roles** page, ensure only the **Sandbox Processing Service** role is selected for install.
 - b. On the **Specify Deployment Options** page, in the **Enter or select the name of the computer that is running SQL Server to use with the deployment**, enter the SQL database (virtual) server name and select **Connect to, and if necessary, upgrade an existing deployment**.
 - c. On the **Specify Security Accounts** page, for the **Sandbox Service**, choose the **CRMSandboxSvc** account.
 - d. On the **Select Microsoft Update Preference** page, choose **I don't want to use Microsoft Update**. Updates to the production-hosted platform should be strictly controlled and only applied after (1) approved by Microsoft for deployment in a hosting environment, and (2) validated in a test environment.
 - e. Complete the installation wizard.
3. Configure Service Principal Name (SPN) for the CRM Sandbox Processing Service Account if necessary. For more details on how to define an SPN for the Microsoft Dynamics CRM Sandbox Processing Service, see [Minimum permissions required for Microsoft Dynamics CRM Setup, services, and components](#).
4. Repeat the preceding steps for all additional CRM Sandbox Processing Back-end servers in the deployment group.

3.3.4 Deploy Deployment Administration Servers

3.3.4.1 Create the CRM Deployment Web Service Account

Use the following procedure to create a domain user account for the CRM Deployment web service.

1. Create a domain user account for the CRM Deployment service, such as **CRMDeploySvc**.
 - a. Consider creating unique accounts for each CRM deployment group to limit the scope of rights for the account across the domain systems.
 - b. Ensure this account has a secure (non-blank) password.
 - c. Ensure the password for this account is not set to expire or a process in place to manage the password changes if you have a password expiration policy.

2. Add the CRM Deployment Service account to the local **Administrators** group on the:
 - a. CRM Deployment server(s)
 - b. CRM SQL Servers
3. Grant the CRM Deployment service account the **sysadmin** role in the CRM SQL Server.

For more details on the permissions required for the Deployment web service, see [Minimum permissions required for Microsoft Dynamics CRM Setup, services, and components](#).

3.3.4.2 Create a CRM Deployment Administration Server

1. Login to CRMDEP01 as a domain administrator.
2. Locate the Microsoft Dynamics CRM Server installation media, execute **SetupServer.exe**, and then proceed through the wizard, making note of the following sections.
 - a. On the **Specify Server Roles** page, ensure that only the **Deployment Administration Server** role is selected for install.
 - b. On the **Specify Deployment Options** page, in the **Enter or select the name of the computer that is running SQL Server to use with the deployment**, enter the SQL database (virtual) server name and select **Connect to, and if necessary, upgrade an existing deployment**.
 - c. On the **Specify Security Accounts** page, for the **Deployment Web Service**, choose the **CRMDeploySvc** account.
 - d. On the **Select a Web Site** page, choose **Select a Web Site**, and then select the default website listed running on port 80.
 - e. On the **Select Microsoft Update Preference** page, choose **I don't want to use Microsoft Update**. Updates to the production-hosted platform should be strictly controlled and only applied after (1) approved by Microsoft for deployment in a hosting environment, and (2) validated in a test environment.
 - f. Complete the installation wizard.
3. Repeat the preceding steps on all additional CRM Deployment Administration servers in the deployment group.

3.4 Email Processing through Server-Side Synchronization

3.4.1 Create the CRM Exchange Administrator Account

CRM Server-side Synchronization will need to connect to the Exchange mail servers as an Exchange Administrator, to manage the forward mailboxes for the hosted CRM organizations. To create and configure the CRM Server-side Synchronization administrator account, using the following procedures.

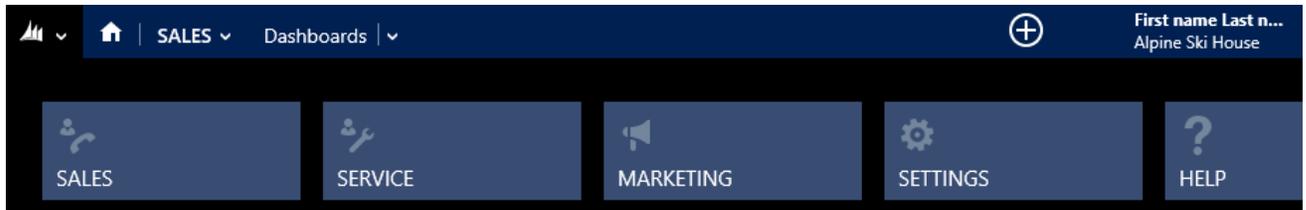
1. Create a new domain user account with a username of CRMEmailAdmin, with a secure password. Ensure the password is set to never expire.
2. Using the Exchange Management interface, add the CRMEmailAdmin account as an Exchange Organization Administrator.

The Dynamic CRM Administration - Email Configuration page provides an interface for configuring email server profiles for managing incoming and outgoing email configurations for users and queues, creating and associating a forward mailbox with an email server profile, creating user mailbox records, configuring system-wide email configuration settings, and migrating Email Router data from a previous installation of CRM.

3.4.2 Create email server profiles

Email server profiles in Microsoft Dynamics CRM store settings that are used by server-side synchronization to connect to an email server and process email for the associated mailboxes. You must configure at least one email server profile to process email for your CRM organization. Depending on the complexity of your organization's email system, you may have to create multiple email server profiles.

1. In the upper left corner of the Microsoft Dynamics CRM web page, select the down arrow next to the Dynamics icon () and select **Settings** from the menu bar.



2. At the **Business Management** page, select the down arrow next to the **Settings** and select **Email Configuration** from the menu bar.
3. On the Email Configuration page, select **Email Server Profiles**.
4. On the Email Server Profiles page, click **New > Exchange Profile**, and specify the following details:

Fields	Description
General	
Name	Specify a meaningful name for the profile.
Description	Type a short description about the objective of the email server profile.
Auto Discover Server Location	Click Yes if you want to use the automatically discover service to determine the server location. This applies only if you are creating a profile for an Exchange Server. If you set this to No , you must specify the email server location manually. If the server location doesn't change for mailboxes, we recommended that you don't use auto discover because it may affect performance.
Incoming Server Location and Outgoing Server Location	If you have selected No in Auto Discover Server Location , type the location of the incoming and outgoing email server. For Exchange Server, use the Exchange Web Services (EWS) URL format <EWS/exchange.asmx>, as illustrated in the following examples: https://autodiscover01.crm.consolidatedmessenger.com/ews/exchange.asmx https://autodiscover02.crm.consolidatedmessenger.com/ews/exchange.asmx
Credentials	

Authenticate Using	<p>Select the method to authenticate while connecting to the specified email server. You can select one of the following:</p> <ul style="list-style-type: none"> ○ Credentials Specified by a User or Queue. If you select this option, the credentials specified in the mailbox record of a user or queue are used for sending or receiving email for the respective user or queue. <p>To ensure the credentials are secured in Microsoft Dynamics CRM, SQL encryption is used to encrypt the credentials stored in the mailbox.</p> <ul style="list-style-type: none"> ○ Credentials Specified in Email Server Profile. If you select this option, the credentials specified in the email server profile are used for sending or receiving email for the mailboxes of all users and queues associated to this profile. The credentials must have impersonation or delegation permissions on the mailboxes associated with profile. This option requires some configuration on the email server, for example, configuring impersonation rights on Exchange for the mailboxes associated with the profile. <p>To ensure the credentials are secured in Microsoft Dynamics CRM, SQL encryption is used to encrypt the credentials stored in the email server profile if you are processing email by using server-side synchronization.</p> <ul style="list-style-type: none"> ○ Windows Integrated Authentication. This option applies only to Exchange and SMTP email server types. If you select this option, the credentials with which the Microsoft Dynamics CRM Asynchronous Service has been configured will be used. <p>The other options in the list aren't applicable for Exchange.</p>
User Name	<p>Type the user name that is used to connect to the email server for sending or receiving email for the mailboxes of all users and queues associated to this profile. This field is enabled and valid only if Authenticate Using is set to Credentials Specified in Email Server Profile. The user name that you specify must have permission to send and receive email from the mailboxes of users and queues associated with this profile.</p> <p>If you are using HTTP for Microsoft Dynamics CRM, the User Name and Password fields will be disabled. To enable the option, change the value of the deployment property AllowCredentialsEntryViaNonSecureChannels to 1. Read about the configuration settings</p>
Password	<p>Specify the password of the user that will be used together with the user name to connect to the email server for sending or receiving email for the mailboxes of users and queues associated with this profile. The password is stored securely.</p> <p>If you are using HTTP for Microsoft Dynamics CRM, the User Name and Password fields will be disabled. To enable the option, change the value of the deployment property AllowCredentialsEntryViaNonSecureChannels to 1. Read about the configuration settings</p>

Use Impersonation	Click Yes if you have configured Exchange impersonation rights to send or receive email on behalf of the other mailboxes. Click No if you have configured Exchange delegation rights to send or receive email on behalf of other mailboxes. Read how to configure Exchange impersonation.
Use same settings for Outgoing	If you want to use the same credential settings for the incoming and outgoing connections, click or tap Yes .
Advanced	
Incoming Port	This field shows the port on the email server for accessing the incoming email. This field is automatically populated when you save the record.
Outgoing Port	This field shows the port on the email server for accessing the outgoing email. This field is automatically populated when you save the record.
Use SSL for Outgoing Connection	Click Yes if the email channel is on a secure channel and SSL must be used for sending email. In Microsoft Dynamics CRM (on-premises), this field is set to Yes by default, but the configuration database setting “AllowNonSSLEmail” allows you to override this and set the value to False. Read about the configuration settings.
Use SSL for Incoming Connection	Click Yes if the email channel is on a secure channel and SSL must be used for receiving email. In Microsoft Dynamics CRM (on-premises), this field is set to Yes by default, but the configuration database setting “AllowNonSSLEmail” allows you to override this and set the value to False.
Incoming Authentication Protocol and Outgoing Authentication Protocol	Select a protocol that will be used for authentication for incoming and outgoing email.
Move Failed Emails to Undeliverable Folder	To move the undelivered email to the Undeliverable folder, click or tap Yes . If there's an error in tracking email messages in CRM as email activities, and if this option is set to Yes , the email message will be moved to the Undeliverable folder. This option is available only for an Exchange email server profile.
Additional Settings	
Process Email From	Select a date and time. Email received after the date and time will be processed by server-side synchronization for all mailboxes associated to this profile. If you set a value less than the current date, the change will be applied to all newly

	associated mailboxes and their earlier processed emails will be pulled.
Maximum Concurrent Connections	Type the maximum number of simultaneous connections that can be made by CRM to the corresponding email server per mailbox. Increase the value to allow more parallel calls to Exchange to improve performance or reduce the value if there are errors on Exchange due to large number of calls from Microsoft Dynamics CRM. The default value of this field is 10. The maximum number is considered per mailbox or per email server profile depending on whether the credentials are specified in a mailbox or email server profile.
Minimum Polling Intervals in Minutes	Type the minimum polling interval, in minutes, for mailboxes that are associated with this email server profile. The polling interval determines how often server-side synchronization polls your mailboxes for new email messages.

5. Click **Save**.

3.5 Deploy SharePoint Grid

The Microsoft Dynamics CRM List component for SharePoint makes your Microsoft Dynamics CRM documents that are stored on SharePoint available to you in a format that has the look and feel of Microsoft Dynamics CRM. This component also enables Microsoft Dynamics CRM to create automatically folders that will be used to store documents related to Microsoft Dynamics CRM records on SharePoint.

For more details, see the [SharePoint Document Software Requirements](#).

- Download and install the [Microsoft Dynamics CRM List Component for Microsoft SharePoint Server 2010 and Microsoft SharePoint Server 2013 \(for multiple browsers\)](#).

3.6 Scripting Deployment Installations with Configuration Files

You can install Microsoft Dynamics CRM, Microsoft Dynamics CRM Reporting Extensions, Microsoft Dynamics CRM for Microsoft Office Outlook, and the Microsoft Dynamics CRM E-mail Router from their respective installation disks or file download location by using the command prompt. The required setup information is provided to the Setup program both as command-line parameters and as an XML configuration file that the Setup program references.

One advantage of using the command prompt to install Microsoft Dynamics CRM is that you do not have to attend the installation. Attended installation requires you to make decisions and provide information so that installation can run successfully. Unattended installation, by using the command prompt, requires you to provide the installation information as command-line parameters and an XML configuration file. No other action is required until the Setup program is finished. Warnings and installation progress can be logged to a file that you can view and analyze later.

For details on how to install the Microsoft Dynamics CRM components using the command line and XML configuration files, see [Use the Command Prompt to Install Microsoft Dynamics CRM](#).

3.7 Deploy CRM for Outlook

Visit these pages to view requirements and installation instructions:

- [Microsoft Dynamics CRM for Outlook hardware requirements](#)
- [Microsoft Dynamics CRM for Outlook software requirements](#)
- [Microsoft Dynamics CRM for Outlook Installation Instructions](#)

For guidance on how to deploy Outlook to a larger base of end users, see the following pages:

- [Deploy Microsoft Dynamics CRM for Outlook by using deployment management software](#)
- [Deploy Microsoft Dynamics CRM for Outlook by using Group Policy](#)

Chapter 4

4 Post-Installation Configuration and Management

This chapter provides details about how to accomplish post-installation tasks. You need to add deployment managers and to configure the platform to be Internet-facing before provisioning organizations and users.

4.1 Microsoft Dynamics CRM 2013 Best Practices Analyzer

After the Microsoft Dynamic CRM 2013 deployment has been completed, we strongly recommend that the Microsoft Dynamics CRM 2013 Best Practices Analyzer be run to perform the following functions:

- Gather information about the Microsoft Dynamics CRM 2013 server roles that are installed on that server.
- Determine if the configurations are set according to the recommended best practices.
- Generate a report on all configurations, indicating settings that differ from recommendations.
- Identify potential problems in the Microsoft Dynamics CRM 2013 features installed.
- Review recommended solutions to potential problems.

Note A Best Practices Analyzer for Microsoft Dynamics CRM 2015 is not available.

4.1.1 Best Practices Analyzer requirements

The Microsoft Dynamics CRM 2013 Best Practices Analyzer has the following requirements:

- At least one CRM 2013 Server role on the computer where Dynamics CRM 2013 BPA is running.
- Microsoft Baseline Configuration Analyzer 2.0 installed where the Best Practices Analyzer is running. Download [Microsoft Baseline Configuration Analyzer 2.0](#).
- The user who is running the Best Practices Analyzer must be a member of the administrators group on the local computer that is scanned.

The Best Practices Analyzer doesn't support the following features:

- Remote capability
- Best practices analytics for CRM for Outlook

4.1.2 Installation instructions

1. Download the [Microsoft Dynamics CRM 2013 Best Practices Analyzer from the Microsoft Download Center](#).
2. Double-click **MicrosoftDynamicsCRMBPA.msi**.
3. On the Welcome screen, click **Next**.
4. If you accept the terms, click **I accept the terms in the License Agreement**, and then click **Next**.

5. If you allow Dynamics CRM BPA to modify the Windows PowerShell settings, click **Select to continue, or cancel to exit Setup**, and then click **Next**.
6. Click **Install**.

4.1.3 Run a scan using the Best Practices Analyzer

1. Click All Programs, right-click **Microsoft Baseline Configuration Analyzer 2.0**, and then click **Run as Administrator**.
2. In the **Select a product list**, click **Dynamics CRM 2013 BPA**, and then click **Start Scan**.
3. If the Enter Parameters screen appears, enter the name of the computer where the CRM server roles are installed. For the local computer, you can enter `*`.

4.1.3.1 View the scan results

On the **Noncompliant** tab, expand **Error** or **Warnings** to view any errors or warnings that may have been detected. For detailed information, click each error. To view all results of the scan, including **Compliant** results, click the **All** tab.

4.1.4 Add Deployment Administrators

To run the CRM Deployment Manager and provision tenant CRM sites, you must be assigned the Deployment Administrator role. Deployment Administrators have complete and unrestricted access to perform Deployment Manager tasks on all organizations and servers in a Microsoft Dynamics CRM deployment.

- Deployment administrators must be defined in each CRM deployment group.
- While you could use the same account, it is recommended that you use separate accounts per CRM deployment group to minimize risk to a security breach.
- Although you should limit the number of Deployment Administrators, the best practice is to enable at least two or three trusted user accounts with this role. This will help prevent potential lockouts that could occur when only using a single account.

4.1.5 Creating a New CRM Deployment Administrator Account

1. Create a new CRM Deployment Administrators Group.
2. Add Deployment Administrator Group to CRM server Local Administrators Group.
3. Grant CRM Deployment Administrator permissions to the CRM Active Directory groups.
4. Grant CRM Deployment Administrators permissions to CRM SQL Objects.
5. Add domain user account to CRM Deployment Administrators group.
6. Add user as a CRM Deployment Administrator in CRM Deployment Manager.

4.1.6 Creating a New CRM Deployment Administrators Group

- Create a new active directory user group for the CRM Deployment Administrator(s). This group will be used to assign permissions to the systems and security groups necessary to administer fully the CRM organizations in a CRM deployment. Consider naming this group unique to the CRM Deployment, such as **CRMDG01Admins**.

4.1.7 Adding Deployment Administrator Group to CRM Server Local Administrators Group

When you add a Deployment Administrator role to a user, Deployment Manager does not grant the user local administrative rights to the CRM Deployment Administration and database servers. This is required to provision resources properly within the deployment.

1. Log on to **CRMDEP01** using an account that is a member of Domain Administrators group.
2. Add the **Contoso\CRMDG01Admins** group to the local Administrators group.
3. Repeat this procedure on **CRMSQL01**.

4.1.8 Granting CRM Deployment Administrator Permissions to the CRM Active Directory Groups

The user who creates, modifies, edits, and imports organizations in Microsoft Dynamics CRM must have permissions in the following Microsoft Dynamics CRM security groups in Active Directory:

- PrivReportingGroup {*guid*}
- PrivUserGroup {*guid*}
- ReportingGroup {*guid*}
- SQLAccessGroup {*guid*}

The CRM Deployment Administrator must have permissions to all four Microsoft Dynamics CRM security groups. The specific permissions a deployment administrator must have on the CRM security groups are:

Permissions

- Read
- Write
- Add/Remove self as member

Advanced permissions

- List Contents
- Read All Properties
- Write All Properties
- Read Permissions
- Modify Permissions
- All Validated Writes
- Add/Remove self as member

The group will be used to grant the necessary permissions to the Microsoft Dynamics CRM security groups. To do so, use the following steps:

1. On a domain controller, start the Active Directory Users and Computers management console.
2. On the **View** menu, click **Advanced Features**.
3. Expand **contoso.com**.
4. Select the organization unit containing the CRM Security groups (as defined during the installation of the first CRM server), The listing pane should display the following CRM security groups:
 - PrivReportingGroup {...}

- PrivUserGroup {...}
- ReportingGroup {...}
- SQLAccessGroup {...}

Note

In the previous list, the {...} represents the globally unique identifier (GUID) following the group name. The GUID will be unique in every deployment. A representative example group name could be **ReportingGroup {4efba72a-232f-44ec-9d95-155eb6ffb1be}**.

5. Right-click the **PrivReportingGroup** security group and then click **Properties**.
6. In the **Properties** dialog box, select the **Security** tab, and in the **Group or user names** list, click **Add**.
7. In the **Enter the object name to select** text box, type **CRMDG01Admins**, click the **Check Names** button, and then click **OK**.
8. With the **CRMDG01Admins** group selected, click to select the **Allow** check box for the **Write** permission. This action causes the system to select automatically the **Add/Remove self as member** check box.

Note

By default, the **Allow** check box is selected for the **Read** permission.

9. Click **Advanced**.
10. In the **Permission** list, select the **CRMDG01Admins** group, and then click **Edit**.
11. Click to select the **Allow** check box for the **Modify Permissions** permission.

Note

By default, the **Allow** check box is selected for the following permissions:

- List Contents
- List Object
- Read All Properties
- Write All Properties
- Read Permissions
- All Validated Writes
- Add/Remove self as member

12. Click **OK** three times.
13. Repeat the steps in this procedure to grant the **CRMDG01Admins** permissions to modify the **PrivUserGroup**, **ReportingGroup**, and **SQLAccessGroup** security groups.

4.1.9 Granting CRM Deployment Administrators Permissions to CRM SQL Objects

When you add a Deployment Administrator role to a user, Deployment Manager does not add the required permissions on the instance of SQL Server where the Microsoft Dynamics CRM databases are stored. When the user tries to start Deployment Manager, the user might receive an error message that says, "Unable to access the MSCRM_CONFIG database. SQL Server does not exist or access denied." To resolve this issue, you must add the user to SQL log-ins by using SQL Server Management Studio. For the new deployment administrator to manage CRM organizations created by other deployment administrators, he or she must be

granted **db_owner** permissions to those databases, or be assigned the sysadmin server role to manage all databases.

1. Log on to **CRMSQL** using an account that is a member of **Domain Administrators** group.
2. Launch the SQL Server Management Studio.
3. On the **Connect to Server** dialog box, click **Connect**.
4. Expand **Security**.
5. Right-click **Logins** and select **New Login**.
6. Click the **Search** button.
7. In the **Select User or Group** dialog box, do the following:
 - a. Click **Object Types**, and then enable the **Groups** type.
 - b. Click **Locations**, and then select **Entire Directory**.
8. Click **OK**.
9. In the **Enter the object name to select** text box, type domain group name (that is, **CRMDG01Admins**), click the **Check Names** button, and then click **OK**.
10. In the **Default database** drop-down box, select **MSCRM_CONFIG**.
11. From the page list on the left, select **Server Roles**, enable the sysadmin role for the user, and then click **OK**.
12. Expand **Databases**.
13. Expand the **MSCRM_CONFIG** database.
14. Expand **Security**.
15. Right-click **Users** and then select **New User**.
16. In the **User name** field, type the domain user login name (that is, **CRMDG01Admins**).
17. In the **Login name** field, type **Contoso\CRMDG01Admins**.
18. In the **Database role membership** section, select the **db_owner** check box, and then click **OK**.
19. Close the SQL Server Management Studio.

4.1.10 Adding Domain User Account to CRM Deployment Administrators Group

1. Verify that a domain user account exists for the new deployment administrator. If it does not, create a new account.
2. Add the new user to the previously created **CRMDG01Admins** group. Also, ensure this account is also member of the **Domain Users** group.

4.1.11 Adding User as a CRM Deployment Administrator in CRM Deployment Manager

The CRM Deployment Administrator role may be granted to a domain user account from within the Deployment Administrator or using a CRM PowerShell cmdlet. The following procedures outline how to add a deployment administrator using the CRM PowerShell tools.

4.1.11.1 Use Windows PowerShell cmdlet to Add a New Deployment Administrator

The **New-CrmDeploymentAdministrator** cmdlet adds a new Deployment Administrator to the deployment.

Syntax:

```
New-CrmDeploymentAdministrator -Name username
```

where:

username is the name of the user being given the Deployment Administrator role. It must be in the form **domain\username**. The user must exist in Active Directory.

4.1.12 Adding a Deployment Administrator

Use the following procedures to commands to add the CRM PowerShell Snap-in, and create a new Deployment Administrator.

1. Logon to the CRM server with the Deployment Administrator role, such as **CRMDEP01**, using the account used to install CRM services.
2. Launch an administrative Windows PowerShell command window from the quick launch bar, or from the **Start** menu under **Program Files, Accessories, Windows PowerShell**, and then **Windows PowerShell**.
3. In the Windows PowerShell command window, execute the following commands:

```
Add-PSSnapin Microsoft.Crm.PowerShell
```

```
New-CrmDeploymentAdministrator -Name contoso\<<username>
```

Note

No data will be returned upon successful completion, as the call is asynchronously processed.

4. To verify that the account was properly created, either open the Deployment Manager and confirm the account is displayed in the **Deployment Administrators** list, or run the following CRM PowerShell cmdlet and confirm the account specified is found in the **Name** field.

```
Get-CrmDeploymentAdministrator
```

4.2 Configure Claims and IFD

The following post-installation activities are required to configure Microsoft Dynamics CRM for an Internet-facing deployment (IFD). Visit [Configure a CRM Internet-facing deployment](#) to learn more about this topic.

4.2.1 Configuring the Microsoft Dynamics CRM Server Websites for SSL/HTTPS

Enabling CRM in an Internet-facing deployment configuration requires the web services to be configured for SSL, and bound to HTTPS.

Configuring a site for HTTPS will cause a disruption in the Microsoft Dynamics CRM application, so plan the configuration when it will result in minimal disruption to users. The high-level steps for configuring Microsoft Dynamics CRM for HTTPS are as follows:

1. Disable the server where the **Web Application Server, Organization Web Service, Discovery Web Service**, and **Deployment Web Service** roles are running. Use the Microsoft Dynamics CRM Deployment Manager to view the **Servers** list and disable the front-end systems.

2. Configure the website where the **Web Application Server** role is installed to use HTTPS.
 - a. Create a certificate request for the Internet-facing CRM websites using one system running IIS. A wildcard certificate is recommended to simplify management and costs, as all CRM sites will be accessed using the shared domain name. The certificate should be provided by a known and trusted third-party certificate authority (CA) to simplify access by end users, and reduce client-side system modifications.
 - b. After the certificate has been obtained from the public certificate authority (CA), install the certificate on the same server to complete the certificate request process. The certificate is now complete and ready for use.
 - c. Export the completed certificate and secure it with a password. The exported certificate will be imported to the other CRM front-ends, as well as provide a backup of the completed certificate.
 - d. Import the wildcard certificate using IIS on the remaining CRM web servers to complete the configuration across all front-end systems.
 - e. Add a binding to the Microsoft Dynamics CRM website for HTTPS, and specify the wildcard certificate. Perform this on each front-end server.
 - f. Restart IIS on all web servers to complete the configuration.

For more information about how to do this, see Internet Information Services (IIS) Help on online resource, such as [How to Set Up SSL on IIS 7](#) and [Configuring Server Certificates in IIS 7](#).
3. Configure the website where the **Deployment Administration Server** role is installed to use HTTPS.
 - a. Create and install a certificate for the internal-facing website running the CRM deployment web service using **CRMDEP01**. A domain-signed certificate is recommended, such that all hosts in the active directory domain will trust the certificate. If you only plan to manage the deployment from **CRMDEP01**, you can use a self-signed certificate.
 - b. Add a binding to the Microsoft Dynamics CRM website for HTTPS, and specify the certificate in the previous step.
 - c. Restart IIS to complete the configuration.
4. Enable the previously disabled front-end systems. Use the Microsoft Dynamics CRM Deployment Manager to view the **Servers** list and enable the disabled front-end systems.

4.2.2 Configuring Fault Tolerance and Firewall

4.2.2.1 Configure Fault Tolerance on CRM Front-end Servers

The CRM front-end servers should be configured for fault tolerance in a production environment. You can use a hardware load balancer or network load balancing (NLB). This guide assumes the use of NLB for fault tolerance of the CRM services.

Define the following load balanced prerequisites:

- Internal CRM Front-end array virtual IP address (VIP)
- Internal CRM Front-end array alias, such as **CRMDG01WEB.contoso.com**
- To enable network load balancing on the CRM Front-end servers, see [Install Microsoft Dynamics CRM Server on multiple computers](#). Special focus should be given to the following sections:
 - ▶ Step 1: Enable network load balancing

- ▶ Step 2: Configure Active Directory
- ▶ Step 4: Configure NLB for the deployment

4.2.2.2 Configure Fault Tolerance for AD FS Web Front-end Servers

Similar to the CRM front-end servers, the AD FS web front-ends should also be configured for fault tolerance in a production environment to ensure availability of the authentication platform. You can use a hardware load balancer or network load balancing (NLB).

Define the following load balanced prerequisites:

- Internal AD FS Front-end array virtual IP address (VIP)
- Internal AD FS Front-end array alias, such as **sts.contoso.com**

See [When to create an AD FS-enabled Web server farm](#) for details about configuring network load-balancing.

4.2.2.3 Configure Internal DNS for Load Balanced CRM and AD FS

Configure the following Internal DNS records to ensure resolution of the load balanced CRM front-end systems using the alias and virtual IP.

On the internal DNS servers, such as **AD01**, perform the following:

- Create a Host (A) record for the CRM front-end array alias to the load balanced virtual IP address.
- Create a Host (A) record for the AD FS web server array alias to the load balanced virtual IP address.
- Verify that you can access the CRM and AD FS front-ends via alias. Open a web browser from any domain member system and navigate to the DNS alias value, such as **https://crmdg01web.contoso.com**.

4.2.2.4 Verify Firewall Configuration

The CRM front-end and AD FS authentication websites must be exposed through the Internet firewall for end users to connect remotely to the hosted environment. Consult your firewall documentation on how to publish the following services:

- The Internal CRM Front-end array URL for CRM organization access (for example, **https://*.crm.consolidatedmessenger.com:443**)
- The Internal AD FS Front-end array URL for user authentication (for example, **https://sts.crm.consolidatedmessenger.com:443**)
- The Internal CRM Front-end array URL and 'external domain' of the IFD configuration (for example, **https://auth.crm.consolidatedmessenger.com:443**)

4.2.3 Configuring Microsoft Dynamics CRM Server for Claims-based Authentication

Use following procedure references to configure Microsoft Dynamics CRM for claims-based authentication.

Except where noted below, execute the procedures as described in [Implement claims-based authentication: internal access](#).

4.2.3.1 Set Microsoft Dynamics CRM Server Binding to HTTPS and Configure the Root Domain Web Addresses

Perform this same titled procedure as described in [Implementing Claims-based Authentication - Internal Access](#) with the following considerations.

- In step 5, configure the **Web Address** endpoints to use the load balanced URLs, such as:

Web Application Server	CRMDG01WEB:443
Organization Web Service	CRMDG01WEB:443
Discovery Web Service	CRMDG01WEB:443
Deployment Web Service	CRMDEP01:443

4.2.3.2 The CRMAppPool Account and the Microsoft Dynamics CRM Encryption Certificate

Perform this same titled procedure as described in [Implementing Claims-based Authentication - Internal Access](#) with the following considerations:

- In step 1, perform this procedure on all CRM Web Application servers.

4.2.3.3 Configure Claims-based Authentication Using the Configure Claims-Based Authentication Wizard

Perform this same titled procedure as described in [Implementing Claims-based Authentication - Internal Access](#) with the following considerations:

- In step 4, enter the Federation metadata URL for your environment, such as:
https://sts1.crm.consolidatedmessenger.com/federationmetadata/2007-06/federationmetadata.xml
- In step 6, select the AD FS encryption certificate. If you set up AD FS to use the same wildcard certificate as CRM, select that certificate.
- In step 10, be sure to capture the URL returned, as it will be required to set up the relaying party trust.

4.2.3.4 Configuring Claims-based Authentication Using Windows PowerShell

You may [skip this procedure](#) as it performs the same actions using PowerShell as previously performed using the wizard.

4.2.4 Configuring the AD FS Server for Claims-based Authentication

Use following procedure references to configure Active Directory Federation Services claims provider trusts and relying party trusts for Microsoft Dynamics CRM.

4.2.4.1 Configure Claims Provider Trusts

Perform this same titled procedure as described in [Implementing Claims-based Authentication - Internal Access](#).

4.2.4.2 Configure Relying Party Trusts

Skip this procedure as it will be performed later when Implementing Claims-based Authentication for External Access.

4.2.4.3 Test Internal Claims-based Authentication

Skip this procedure as it will be performed later when Implementing Claims-based Authentication for External Access.

4.2.5 Configuring Microsoft Dynamics CRM for Internet-facing Deployment

Use following procedure references to configure Microsoft Dynamics CRM for an Internet-facing deployment.

Except where noted below, execute the procedures as described in [Implement claims-based authentication: external access](#).

4.2.5.1 Configure an Internet-facing Deployment using the Configure Internet-Facing Deployment Wizard

Perform this same titled procedure as described in [Implement claims-based authentication: external access](#) with the following considerations:

- In step 4, configure the domain values, such as:

Web Application Server Domain	crm.consolidatedmessenger.com:443
Organization Web Service Domain	crm.consolidatedmessenger.com:443
Discovery Web Service Domain	dev.crm.consolidatedmessenger.com:443

- In step 5, configure the Internet-facing server location to a subdomain of the Web Application Server domain, such as **auth.crm.consolidatedmessenger.com:443**
- In step 9, restart IIS on all CRM web front-end servers.
- Verify that the DNS records exist, and point to the public IP for **dev.crm.consolidatedmessenger.com** and **auth.crm.consolidatedmessenger.com**.

4.2.5.2 Configuring Claims-based Authentication Using Windows PowerShell

You may skip this procedure as it performs the same actions using PowerShell as previously performed using the wizard.

4.2.5.3 Configure Relying Party Trusts

Perform this same titled procedure as described in [Implement claims-based authentication: external access](#) with the following considerations:

- In step 4, use the federation metadata URL created during the IFD, such as:
https://auth.crm.consolidatedmessenger.com/FederationMetadata/2007-06/FederationMetadata.xml
- Or, simply **https://auth.crm.consolidatedmessenger.com/**

4.2.5.4 Test External Claims-based Authentication

If you have a CRM organization, you can perform this procedure; otherwise, you can test this later.

Chapter 5

5 Upgrade Guidance

This section of the documentation provides detailed instructions for Service Providers currently running hosted Microsoft Dynamics CRM 2011, and wanting to upgrade their environment to a hosted Microsoft Dynamics CRM 2013.

Note For Microsoft Dynamics CRM 2013 to Microsoft Dynamics CRM 2015 upgrade guidance, see [Upgrade a Microsoft Dynamics CRM Server](#).

The recommended strategy is to install CRM 2013 using a new instance of SQL Server and new hardware for the CRM 2013 roles. An in-place upgrade of CRM is not recommended, as the hosted platform is serving a high number of users across many tenant organizations. Managing the customer expectations along with meeting service level agreements could be at risk during an in-place upgrade.

The CRM 2013 environment should be installed into the domain hosting the CRM 2011 environment. Tenant organizations may then be upgraded from the CRM 2011 deployment to the CRM 2013 deployment in a scheduled manner, to minimize disruption to the customer as well as the support desk.

The CRM 2013 solution should be designed to meet your current and future needs. Take into consideration the new roles, design points, and changes to the IFD implementation.

The new CRM 2013 environment should be designed and deployed using the guidance from the earlier chapters in this document. Once built, the new CRM 2013 deployment should be thoroughly tested.

When the platform is fully functional and ready for production users, you should begin to upgrade (or migrate) them from the CRM 2011 deployment to the CRM 2013 deployment. This upgrade will consist of backing up their current CRM organization database, restoring it in the CRM 2013 deployment, and then importing the database using the CRM 2013 tools. During this import, the database will be upgraded for CRM 2013.

Objectives:

- Minimize disruption of production users during the upgrade period.
- Tenant organizations can be upgraded individually or in bulk.
- Tenant organizations can be rolled back to CRM 2011 if upgrade fails.
- Minimize end-user configuration changes.
- Minimize impact to the operations team.

Considerations:

The following considerations should be made prior to upgrading the environment and tenant organizations to Microsoft Dynamics CRM 2013.

- New hardware will be required for the CRM 2013 systems.
- CRM 2013 includes ability to separate further deployment roles. Given this, you should reevaluate your existing solution and usage, and design the CRM 2013 environment to suit the current and future needs.

- Automated provisioning solutions may not integrate with CRM 2013, and should be fully tested in a lab prior to upgrading to a production environment.
- Existing CRM 2011 systems will not be modified during the upgrade.
- The public-facing domain of the CRM 2013 deployment should match the domain used in the CRM 2011 deployment, to minimize end user client configuration changes.
- A separate public IP should be used for the new CRM 2013 environment.
- Tenant organization DNS records should be updated after the CRM organization has been upgraded to the CRM 2013 deployment. Lowering the **Time To Live (TTL)** value for the host records (and allowing the new value to propagate) will help minimize client connectivity failures.
- CRM 2011 and CRM 2013 cannot be deployed on the same instance of SQL Server. If you intend to leverage the same hardware for the database, you must install a new instance of SQL Server.
- Before you upgrade from Microsoft Dynamics CRM 2011 for Outlook to Microsoft Dynamics CRM 2013 for Microsoft Office Outlook, consider the following
 - **Base languages must match.** To upgrade CRM for Outlook, the base language of the version of CRM 2013 for Outlook you want to install must match the base language of Microsoft Dynamics CRM 2011 for Outlook.
 - **Upgrade is required to continue offline access after server upgrade.** For an on-premises deployment of Microsoft Dynamics CRM, after the Microsoft Dynamics CRM Server 2011 has been upgraded to Microsoft Dynamics CRM Server 2013, users running CRM 2013 for Outlook can continue to connect and use Microsoft Dynamics CRM. However, the clients must upgrade to CRM 2013 for Outlook to be able to go offline.
 - **Changing from 32-bit to 64-bit.** If you want to change to a different processor architecture (move from 32-bit to 64-bit) while upgrading, note the following:
 - **In-place cross-architecture upgrade isn't supported.** If you're running the 32-bit version of Microsoft Dynamics CRM 2011 for Outlook, you can't in-place upgrade to 64-bit CRM 2013 for Outlook. This also applies to Microsoft Office: If you're running Microsoft Office and want to retain 32-bit, you can upgrade only to the 32-bit version of CRM 2013 for Outlook.
 - **Cross-architecture upgrade requires uninstalling and reinstalling.** To change to a different architecture, perform the following steps in the order listed:
 1. Uninstall Microsoft Dynamics CRM 2011 for Outlook 32-bit.
 2. Make sure that you have a 64-bit edition of Microsoft Office installed. If not, install one. To determine the version, in Outlook click or tap **File**, click or tap **Help**, and in the right-pane next to **Version** the bit-version is displayed.
 3. Install the 64-bit edition of CRM 2013 for Outlook.
- Tenant users should upgrade their client application prior to their CRM organization upgrade.
- Review the [Tips for a successful upgrade](#), and evaluate existing CRM organizations for any issues.
- Review the [Upgrade issues and considerations](#) to review known issues that may occur following the upgrade.
- Only deployments with Microsoft Dynamics CRM 2011 Update Rollup 14 or later are supported for upgrade. For more information, see Microsoft Dynamics CRM 2011 versions supported for upgrade.

Prerequisites:

A Hosted Microsoft Dynamics CRM 2011 platform must be fully deployed and functioning properly.

Important

It is highly recommended that the upgrade solution and scenarios be fully tested in a lab environment prior to implementing it in production. The lab environment should mirror the production environment as closely as possible with respect to the hardware, software, topology, network implementation, and organization data and configuration.

For additional information about upgrading from Microsoft Dynamics CRM 2011, see [The server upgrade process](#).

5.1 Design Hosted Microsoft Dynamics CRM

As you design your Hosted Microsoft Dynamics CRM deployment, keep in mind your current environment as requirement inputs:

- How many organizations are you hosting?
- How many users are there total? Per organization?
- What are the usage patterns?
- Do you have asynchronous workloads?
- Are you hosting customizations?

Each of these issues will help guide your decision-making during the design process. Use the planning information and example deployment in this guide, along with the CRM server product documentation, as an aid to this process.

5.2 Deploy Hosted Microsoft Dynamics CRM

Build the designed solution using the deployment guidance from this document, adjusting as necessary for your solution.

Deploy Active Directory Federation Services (AD FS) for Claims-based authentication, or leverage an existing Secure Token Solution.

Configure and expose the CRM deployment publicly for SSL on a static IP address.

Configure CRM for the Claims-based Authentication and an Internet-facing deployment using the same root domain as the previous CRM deployment.

Verify access from the web and CRM Outlook client after creating a test organization and DNS record.

5.3 Upgrade a CRM Organization

5.3.1 Upgrade Options

When installing Microsoft Dynamics CRM Server in an organization that already has a previous version of Microsoft Dynamics CRM Server installed, the following three different upgrade options should be considered:

- **Migrate by using a new instance of Microsoft SQL Server.** We recommend this option for upgrading Microsoft Dynamics CRM Server. Although this option requires a different computer for the new version of Microsoft Dynamics CRM Server and a different instance of SQL Server, it provides the least amount of potential downtime for Microsoft Dynamics CRM because the previous Microsoft Dynamics CRM deployment can remain functioning until the upgrade is completed and verified.
- **Migrate by using the same instance of SQL Server.** This option requires a different computer for the new version of Microsoft Dynamics CRM Server, but will upgrade in-place the configuration and default organization databases using the same instance of SQL Server. If issues occur during the

upgrade, you must roll back to the previous version of Microsoft Dynamics CRM Server to avoid significant downtime.

- **In-place upgrade.** Although this option doesn't require a different computer for the new version of Microsoft Dynamics CRM Server or a different instance of SQL Server, it poses the greatest risk if upgrade issues occur because a rollback and reinstall of the previous version of Microsoft Dynamics CRM Server will be required to avoid potential downtime.

5.3.2 Software Prerequisites

Before you run the upgrade, we recommend that you install the following components on the computer where you will install the new version of Microsoft Dynamics CRM or on the computer where the new version of Microsoft Dynamics CRM Server will be migrated to:

- Windows Identity Foundation Framework
- Microsoft .NET Framework 4
- Microsoft SQL Server Native Client
- Microsoft Application Error Reporting Tool
- Microsoft URL Rewrite Module for IIS

If these components aren't present when you run the upgrade, Setup will install them. However, to reduce the upgrade processing time, consider installing these components in advance. These components can be found in the **redist** folder of the Microsoft Dynamics CRM installation media.

5.3.3 Migrate by using a new instance of SQL Server

The **Migrate by using a new instance of SQL Server** option provides the least amount of potential downtime in the event of an issue arising from the upgraded deployment. The following steps should be completed in order:

1. Establish a new instance of SQL Server. You can use an existing instance, but it *must not* be the same instance where the Microsoft Dynamics CRM configuration database is located.
2. Run the new version of Microsoft Dynamics CRM Server Setup on a new 64-bit computer that does not already have Microsoft Dynamics CRM Server installed.
3. Back up the previous version's production Microsoft Dynamics CRM configuration and organization databases and restore them to the new instance of SQL Server.
4. Run the **Import Organization Wizard** to import one or more Microsoft Dynamics CRM organizations to the newly installed Microsoft Dynamics CRM system. During the import, the previous version's Microsoft Dynamics CRM organization database will be upgraded.
5. If you have additional organizations or if you are using a new SQL Server for the migration, you must import the organization databases to the new system. To do this, on the computer where Microsoft Dynamics CRM Server is installed and running, start Microsoft Dynamics CRM Deployment Manager, right-click Organizations, click Import Organization, and then select the newly restored Microsoft Dynamics CRM OrganizationName_MSCRM database.
6. If customizations were made to .NET assemblies or configuration files, you must copy those customized files to the new system. By default, these files are located under the <drive>:\Program Files\Microsoft Dynamics CRM\Server\bin\assembly\ folder on the previous version of Microsoft Dynamics CRM Server.

5.3.4 Backing up CRM Organization Database

Prior to starting the upgrade, perform the following:

1. Ensure that the existing CRM organization database is healthy and CRM is operational.
2. Capture the following information about the CRM organization, which can be seen on the properties of the organization in the CRM Deployment Manager.
 - Unique Name
 - Display Name
3. Ensure a domain account is defined with the system administrator role in the CRM organization, and that account credential is available. This account will be the 'setup account' used to perform the import of the organization into the upgraded CRM deployment. If the credential is not known, or another account is preferred for this purpose, add the preferred account as a system administrator of the CRM organization before proceeding.

Note

This account should have the sysadmin role in SQL Server in the upgraded CRM deployment.

4. Consider disabling the language packs, prior to backing up the database, if the new CRM deployment does not have the same language packs installed.
5. Disable the existing **CRM Organization** to prevent users from accessing the site.
6. Perform a full database backup of the existing CRM organization database (**OrganizationName_MSCRM**) using SQL management tools.

For more information about how to back up databases, see [Backing Up and Restoring Databases in SQL Server](#).

5.3.5 Restoring CRM 2011 Organization Database into CRM SQL

Restore the existing CRM organization database backup on to the SQL server containing the upgraded CRM organization databases, using the following steps.

1. Copy the existing CRM organization database backup file to the SQL server in the upgraded CRM deployment.
2. Restore the customer CRM database using the same database name. Ensure that you change the restore file paths as necessary for the new deployment.

For more information about how to restore databases, see [Backing Up and Restoring Databases in SQL Server](#).

5.3.6 Importing a CRM Organization Database

You can import an existing Microsoft Dynamics CRM organization by using the Import Organization Wizard in the Deployment Manager, or by using the CRM PowerShell tools. You can use this wizard when you move the CRM organization to another database server. The Import Organization procedure supports mapping users from any domain in the Active Directory service forest.

Important

- Before you import, upgrade, or update a Microsoft Dynamics CRM organization, you should back up the configuration and organization databases.

- You cannot import a database that is already deployed in the target Microsoft Dynamics CRM deployment.
- Only Microsoft Dynamics CRM Server editions support multiple organizations in the deployment.
- If the Import Organization procedure detects that the organization you are importing is from an earlier version or patch level of Microsoft Dynamics CRM, the organization database will be upgraded to the new version during the import.
- We recommend that for each organization that you upgrade, the volume have free space that is at least three times the size of the organization database file and four times the size of the log file. Therefore, if a single organization database and log file are located on the same volume and are one gigabyte in total, you should have at least seven gigabytes of available disk space before you perform the upgrade.
- Before you can import an organization by using the Import Organization procedure, the organization database must be restored and attached to SQL Server. For more information about how to restore databases, see SQL Server Books Online.

5.3.6.1 Windows PowerShell cmdlet to Import an Organization

The **Import-CrmOrganization** command imports the specified organization database into the Microsoft Dynamics CRM deployment.

Syntax:

```
Import-CrmOrganization -SqlServerName sql_server_name -DatabaseName database_name -SrsUrl srs_url [-DisplayName display_name -Name name -UserMap usermap_file_path]
```

where:

- *sql_server_name* is the name of SQL Server database server on which to look for the organization database to import. (mandatory)
 - *database_name* is the name of the organization database that will be imported. (mandatory)
 - *srs_url* is the URL of the SQL Server Reporting Services reporting server for the organization that will be imported. (mandatory)
 - *display_name* is the display name of the organization that will be imported. If no display name is provided, the default behavior is to use the existing display name in the organization's database. (optional)
 - *name* is the unique name of the organization that will be imported. If no unique name is provided, the default behavior is to use a unique version of the display name. (optional)
 - *usermap_file_path* is the file path to a user mapping file. If no file path is provided, the default behavior is to auto-map based on the Active Directory account name.
1. Logon to the CRM server with the **Deployment Administrator** role, as a CRM Deployment Administrator. This account should be the CRM system administrator defined as the 'setup account' earlier.
 2. Launch an administrative Windows PowerShell window from the quick launch bar, or from the Start menu under **Program Files, Accessories, Windows PowerShell**, and then **Windows PowerShell**.
 3. In the Windows PowerShell command window, execute the following command.

```
Add-PSSnapin Microsoft.Crm.PowerShell
```
 4. In the Windows PowerShell command window, execute the following command.

```
Import-CrmOrganization -SqlServerName CRMSQL -DatabaseName alpineskihouse_MSCRM -SrsUrl
http://crmrep.contoso.com/ReportServer" -DisplayName "Alpine Ski House Inc" -Name alpineskihouse
```

Notes

- You will not need to supply a user-mapping file, as the users (and user names) are not changing. The CRM organization users and role memberships will be maintained during the import.
 - Although the GUID of job will be returned upon successful submission of the request, the actual import process may take a significant amount of time depending upon the size of the database.
5. Using the CRM Deployment Manager or the **get-CrmOrganization** cmdlet, verify that the CRM organization successfully imported and is enabled.

For more details on the **Import-CRMOrganization** cmdlet parameters and other cmdlets, see the [CRM 2015 PowerShell Reference](#) and [Administer the deployment using Windows Powershell](#).

5.3.7 Modifying DNS Records for CRM Organization

After the CRM organization has been imported into the CRM deployment, and the deployment has been enabled for IFD and publicly exposed through the firewall, access to the site should be tested. This may be done by using a local hosts file entry for the organization name to the public IP. Once access to the CRM organization has been confirmed using the fully qualified domain name using a hosts file, the public DNS records should be changed and the local hosts file records should be removed.

1. Test connectivity the CRM fully qualified domain name using a local hosts file entry on a client system. Verify that the web client and Microsoft Dynamics CRM for Outlook client connect and function as expected.
2. Modify the public DNS host record for the organization to resolve to the public IP of the CRM deployment.

5.3.8 Migrate settings from the Email Router to server-side synchronization

To switch from Microsoft Office Outlook synchronization to server-side synchronization, simply change the synchronization method in mailbox records to server-side synchronization. That's all you have to do to make the change from Outlook synchronization to server-side synchronization.

During migration, the old incoming and outgoing profiles for the user and queue mailboxes are merged to create a new email server profile that will be used by server-side synchronization.

1. Before you start, make sure you have Read, Write, and Create privileges on the Mailbox entity and Read, Write, and Create privileges on the Email Server Profile entity.
2. Click **Migrate Email Router Data**.
3. In the Email Router Data Migration wizard, in the three text boxes, click **Browse**, and select the three files specified at the top of the page in the same order. If you must migrate data from multiple email routers, click **More Email Routers** and again select the three files. You can migrate data from up to four email routers at once.

The maximum combined size of all the files from all the email routers that you can upload at a time is 32 MB.

4. Click **Next**.
5. On the **Select Email Server Profiles to Migrate** page, the incoming and outgoing email server profiles of the Email Router are listed and the details about the new email server profile for server-side synchronization is also listed. If you want to migrate the profile, in **Migrate Server Profile**, click **Yes**.

In the Email Router, incoming and outgoing email server profiles are different and each user or queue is associated with both incoming and outgoing profiles. However, with server-side synchronization, the incoming and outgoing settings are defined in a single profile and a user or queue is associated with this profile. Thus, when you migrate the data, the data from two server profiles is combined into one. The **Select Email Server Profiles to Migrate** page shows details about the new email server profile that will be created. Read more about [merging email server profiles for migration](#).

6. Click **Next**.
7. The **Migration Review Summary** page shows what data will be migrated. Click **Start**.

After the migration is complete, you'll see the summary of the migrated data. You must test the email configuration for the mailboxes after the migration is complete. To be able to start email processing through server-side synchronization, in the **Process Email From** field in the **System Settings** dialog box, select **Server-Side Synchronization**. Read how to configure [System Settings dialog box - Email tab](#).

5.3.9 Enabling Anonymous Authentication for the Discovery Web Service

To enable anonymous authentication for the Microsoft Dynamics CRM Discovery web service, see [Anonymous authentication](#).

5.3.10 Refreshing the CRM Organization Identifiers in AD FS

To authenticate successfully to a CRM organization secured using AD FS for claims-based authentication, the Identifiers for the relying party trust must include the organization specific identifier. By default, AD FS updates the CRM identifiers once daily. To verify access to the upgraded site, the identifiers must be manually updated.

Important

Consider changing the automatic update interval for AD FS for your environment. This will depend upon when and how often CRM organizations are provisioned, the expected delay between provisioning and customer sign-on, and the expense of performing the update against the CRM front-end systems.

1. On the computer that is running the AD FS 2.1 federation server, start **AD FS 2.0 Management**.
2. Click the **AD FS 2.0 Trust Relationships** and select **Relying Party Trusts**.
3. In the details pane, click the relying party trust for the CRM deployment.
4. In the **Action** pane, click **Update from Federation Metadata**.
5. In the properties dialog box for the trust, click **Update** to perform the update.

5.4 Verify Access Using Web Client and Outlook

Using a client system configured to the previous version of CRM organization public URL, attempt to authenticate as a CRM user to the upgraded CRM organization.

Note

If you change the public root domain for CRM and/or the unique name for the CRM organization, the URL will no longer match the address of the CRM site previously configured on the client. The client will need to be reconfigured for the new URL, as discussed in [Task 2: Configure Microsoft Dynamics CRM for Outlook](#).

5.4.1 Verify the Web Client

1. Launch a web browser and access the CRM fully qualified domain name, such as **<https://alpineskihouse.crm.consolidatedmessenger.com>**.
2. When prompted with the new claims-based authentication form, enter the credentials of a CRM user.
3. Verify that the CRM web client is displayed and operating properly.

5.4.2 Verify the CRM for Outlook Client

1. Notice that you may need to update Microsoft Dynamics CRM for Outlook before you can connect to an upgraded Microsoft Dynamics CRM Server. More information: [Microsoft Dynamics CRM for Outlook compatibility with Microsoft Dynamics CRM Server](#)
2. Launch and login to Outlook.
3. Login to CRM if prompted.
4. Verify that the CRM features are displayed and operating properly.

5.5 Upgrade the CRM for Outlook Client

For details on how to upgrade the Microsoft Dynamics CRM 4.0 for Outlook to the CRM 2013 client, see the procedures outlined in [Upgrade Microsoft Dynamics CRM 2011 for Outlook to Microsoft Dynamics CRM 2013 for Outlook](#).

For details about how to upgrade Microsoft Dynamics CRM 2013 for Outlook to CRM 2015 for Outlook, see [Upgrade to Microsoft Dynamics CRM 2015 for Outlook](#).

Before you upgrade from Microsoft Dynamics CRM 2011 for Outlook to Microsoft Dynamics CRM 2013 for Microsoft Office Outlook, consider the following:

- **CRM 2013 for Outlook is incompatible with earlier versions of Microsoft Dynamics CRM Server**, such as Microsoft Dynamics CRM Server 2011 (on-premises). However, Microsoft Dynamics CRM 2011 for Outlook is compatible with Microsoft Dynamics CRM Server 2013. More information: [Upgrade Microsoft Dynamics CRM 2011 for Outlook to Microsoft Dynamics CRM 2013 for Outlook](#).
- **Verify that your computer and software supports running CRM 2013 for Outlook**. More information: [Microsoft Dynamics CRM for Outlook hardware requirements](#), [Microsoft Dynamics CRM for Outlook software requirements](#)
- **Keep your Microsoft Dynamics CRM deployment current by installing the latest updates and hotfixes**. You can automatically apply the latest update by using [Microsoft Update](#). You can also search for updates on the [Microsoft Download Center](#). Choosing Microsoft Update lets you install recommended updates automatically and without administrator permissions.
- **Base languages must match**. To upgrade CRM for Outlook, the base language of the version of CRM 2013 for Outlook you want to install must match the base language of Microsoft Dynamics CRM 2011 for Outlook.
- **Verify Microsoft Dynamics CRM 2011 for Outlook is in online mode**. If you have an earlier version of CRM for Outlook, you cannot upgrade that version of CRM for Outlook when it is in **Go offline** mode. You must bring CRM for Outlook online before you can upgrade to CRM 2013 for Outlook.

- **Upgrade is required to continue offline access after server upgrade.** For an on-premises deployment of Microsoft Dynamics CRM, after the Microsoft Dynamics CRM Server 2011 has been upgraded to Microsoft Dynamics CRM Server 2013, users running CRM 2013 for Outlook can continue to connect and use Microsoft Dynamics CRM. However, the clients must upgrade to CRM 2013 for Outlook to be able to go offline.
- **Changing from 32-bit to 64-bit.** If you want to change to a different processor architecture (move from 32-bit to 64-bit) while upgrading, note the following:
 - In-place cross-architecture upgrade isn't supported. If you're running the 32-bit version of Microsoft Dynamics CRM 2011 for Outlook you can't in-place upgrade to 64-bit CRM 2013 for Outlook. This also applies to Microsoft Office: If you're running Microsoft Office and want to retain 32-bit, you can upgrade only to the 32-bit version of CRM 2013 for Outlook.
 - Cross-architecture upgrade requires uninstalling and reinstalling. To change to a different architecture, perform the following steps in the order listed:
 1. Uninstall Microsoft Dynamics CRM 2011 for Outlook 32-bit.
 2. Make sure that you have a 64-bit edition of Microsoft Office installed. If not, install one. To determine the version, in Outlook click or tap **File**, click or tap **Help**, and in the right-pane next to **Version** the bit-version is displayed.
 3. Install the 64-bit edition of CRM 2013 for Outlook.

Chapter 6

6 Provisioning

This chapter includes two sections that describe how to provision organizations and users. The first section details the steps involved in manual provisioning. The second describes an approach to automating provisioning including code samples in C#.

6.1 Manual Provisioning

These provisioning examples show how you can use Windows PowerShell to accomplish many provisioning tasks. In cases where it takes less time to simply use the administrative interfaces available, this section provides those steps instead.

6.1.1 Creating, Importing, Editing Organizations

CRM is designed for multi-tenancy, and with Microsoft Dynamics CRM Server, you can provision multiple organizations within a single CRM deployment.

This may be done via the Deployment Manager, or via Windows PowerShell. We will use Windows PowerShell in the procedures outlined below.

Prior to provisioning the CRM Organization for a hosted tenant organization, the organization container and user accounts should already exist in active directory. It is assumed that these accounts will be created in the corresponding tenant organization container, based upon your multi-tenant design in Active Directory.

6.1.1.1 Create a CRM Organization

You can create a new CRM organization in the Microsoft Dynamics CRM deployment, using the Deployment Manager or the CRM PowerShell tools. Use the following procedure to create a new tenant organization in CRM using the CRM **PowerShell** cmdlet.

1. Logon to the CRM server with the **Deployment Administrator** role, such as **CRMDEP01**, as a CRM Deployment Administrator.
2. Launch an administrative Windows PowerShell window from the quick launch bar, or from the **Start** menu under **Program Files, Accessories, Windows PowerShell**, and then **Windows PowerShell**.
3. In the Windows PowerShell command window, execute the following commands:

```
Add-PSSnapin Microsoft.Crm.PowerShell
```

4. In the Windows PowerShell command window, execute the following command to create a CRM organization for the hosted tenant organization.

```
New-CrmOrganization -DisplayName "Alpine Ski House" -SQLServerName "CRMSQL" -SrsUrl
"http://CRMSQL/ReportServer" -Name "alpineskihouse" -BaseCurrencyCode "USD" -BaseCurrencyName
"US Dollar" -BaseCurrencySymbol "$" -BaseCurrencyPrecision "2" -BaseLanguageCode 1033 -SqlCollation
"Latin1_General_CI_AI" -SQMOptIn false
```

Note

Although the GUID of job will be returned upon successful submission of the request, the actual creation process may take a significant amount of time.

For more details on the **New-CRMOrganization** cmdlet parameters and other cmdlets, see [CRM 2015 PowerShell Reference](#) and [Administer the deployment using Windows Powershell](#).

Syntax:

```
New-CrmOrganization -DisplayName display_name -SQLServerName sql_server_name -SrsUrl srs_url -
Name name -BaseCurrencyCode base_currency_code -BaseCurrencyName base_currency_name -
BaseCurrencySymbol base_currency_symbol -BaseCurrencyPrecision base_currency_precision -
BaseLanguageCode base_language_code -SqlCollation sql_collation {-SQMOptIn true|false}
```

where:

- *display_name* is the display name of the organization that will be imported. If no display name is provided, the default behavior is to use the existing display name in the organization's database. (mandatory)
- *sql_server_name* is the name of the SQL Server database server on which to look for the organization database to import. (mandatory)
- *srs_url* is the URL of the SRS reporting server for the organization that will be imported. (mandatory)
- *name* is the unique name of the organization that will be imported. If no unique name is provided, the default behavior is to use a unique version of the display name. (Optional)
- *base_currency_code* is the base currency code for the organization to be created, for example, "USD" for US dollar.
- *base_currency_name* is the name of the base currency, for example, "US Dollar" for USD. If no base currency name is provided, the default behavior is to use a currency name based on the **base_currency_code** specified.
- *base_currency_symbol* is the symbol of the base currency, for example, "\$" for the US Dollar. If no base currency symbol is provided, the default behavior is to use a value based on the **base_currency_code** specified.
- *base_currency_precision* is the decimal precision of the base currency, for example, two decimal places for US Dollar. If no base currency precision is provided, the default behavior is to use a value based on the **base_currency_code** specified.
- *base_language_code* is the code for the base language for the organization being created, for example, "1033" for English.
- *sql_collation* is an enumerated (enum) value that specifies the preferred SQL collation that the organization database will use to sort and compare data characters.
- *SQMOptIn* specifies whether to have the organization participate in the Customer Experience Improvement Program. The default value is false.

6.1.1.2 Disable or Enable a CRM Organization.

You use the **Organizations** page to disable or enable an organization. It is a best practice to disable an organization when you perform database maintenance.

Important

When you disable an organization, users will no longer be able to access the Microsoft Dynamics CRM application for the organization. To make it available to users again, you must enable it.

6.1.1.2.1 Windows PowerShell cmdlet to Disable a Microsoft Dynamics CRM Organization

The **Disable-CrmOrganization** cmdlet disables the specified organization.

Syntax:

```
Disable-CrmOrganization -Name organization_name
```

where:

- *organization_name* is the name of the organization to disable. (mandatory)
1. Logon to the CRM server with the Deployment Administrator role, such as CRMDEP01, as a CRM Deployment Administrator.
 2. Launch an administrative Windows PowerShell window from the quick launch bar, or from the **Start** menu under **Program Files, Accessories, Windows PowerShell**, and then **Windows PowerShell**.
 3. In the Windows PowerShell command window, execute the following commands.

```
Add-PSSnapin Microsoft.Crm.PowerShell
```

4. In the Windows PowerShell command window, execute the following command.

```
Disable-CrmOrganization -Name alpineskihouse
```

For more details on the **Disable-CRMOrganization** cmdlet parameters and other cmdlets, see the [CRM 2015 PowerShell Reference](#) and [Administer the deployment using Windows Powershell](#).

6.1.1.2.2 Windows PowerShell cmdlet to Enable a Microsoft Dynamics CRM Organization

The **Enable-CrmOrganization** cmdlet enables the specified organization.

Syntax:

```
Enable-CrmOrganization -Name organization_name
```

where:

- *organization_name* is the name of the organization to enable. (mandatory)
1. Logon to the CRM server with the **Deployment Administrator** role, such as **CRMDEP01**, as a CRM Deployment Administrator.
 2. Launch an administrative Windows PowerShell window from the quick launch bar, or from the **Start** menu under **Program Files, Accessories, Windows PowerShell**, and then **Windows PowerShell**.
 3. In the Windows PowerShell command window, execute the following commands.

```
Add-PSSnapin Microsoft.Crm.PowerShell
```

4. In the Windows PowerShell command window, execute the following command.

```
Enable-CrmOrganization -Name alpineskihouse
```

For more details on the **Enable-CRMOrganization** cmdlet parameters and other cmdlets, see the [CRM 2015 PowerShell Reference](#) and [Administer the deployment using Windows Powershell](#).

6.1.1.3 Edit a CRM Organization's Attributes

Before you can edit an organization, you must disable it. After it is disabled, make the changes that you want, and then enable the organization so that it is available to users.

You can change the display name and the SQL Server computer where the organization database is located. Also, you can change the Microsoft Dynamics CRM Connector for SQL Server Reporting Services URL, which specifies where the Microsoft Dynamics CRM Connector for SQL Server Reporting Services is installed.

You cannot change the name of the organization.

6.1.1.3.1 Windows PowerShell cmdlet to Edit a Microsoft Dynamics CRM Organization

The **Edit-CrmOrganization** command edits various properties of an organization.

Syntax:

```
Edit-CrmOrganization -Name organization_name [-DisplayName display_name -SqlServerName sql_server_name -SrsUrl srs_url]
```

where:

- *organization_name* is the name of the organization to edit. (mandatory)
- *display_name* is the new display name. (optional)
- *sql_server_name* is the name of computer that is running the SQL Server database for the organization. (optional)
- *srs_url* is the URL to SRS instance. (optional)

6.1.1.3.2 Windows PowerShell cmdlet to Retrieve a List of Organizations

The **Get-CrmOrganization** cmdlet retrieves a list of the organizations in the deployment.

Syntax:

```
Get-CrmOrganization [-Name organization_name]
```

where:

- *organization_name* is the name of the organization to edit. (optional)
1. Logon to the CRM server with the **Deployment Administrator** role, such as **CRMDEP01**, as a CRM Deployment Administrator.
 2. Launch an administrative Windows PowerShell window from the quick launch bar, or from the **Start** menu under **Program Files, Accessories, Windows PowerShell**, and then **Windows PowerShell**.
 3. In the Windows PowerShell command window, execute the following commands.

```
Add-PSSnapin Microsoft.Crm.PowerShell
```

4. In the Windows PowerShell command window, execute the following command to disable the CRM organization prior to editing.

```
Disable-CrmOrganization -Name alpineskihouse
```

5. To edit the desired properties for the CRM organization, use the following command and include the additional properties being changed:

```
Edit-CrmOrganization -Name alpineskihouse -DisplayName "Alpine Ski House Inc"
```

6. Display the desired properties for the CRM organization by executing the following command:

```
Get-CrmOrganization -Name alpineskihouse
```

The organization details will be returned. The updated display name will be found in the **FriendlyName** property. Also, notice that the disabled status of the organization is displayed in the **State** property.

7. The organization must be enabled to restore end user access. Enable the site using the following command:

```
Enable-CrmOrganization -Name alpineskihouse
```

8. Verify that the site has been enabled and is accessible.

For more details on the **Edit-CRMOrganization** cmdlet parameters and other cmdlets, see the [CRM 2015 PowerShell Reference](#) and [Administer the deployment using Windows Powershell](#).

6.1.1.4 Create DNS record for CRM Organization

After the CRM organization has been created in the CRM deployment, and the deployment has been enabled for IFD and publicly exposed through the firewall, the CRM organization can be made available for access.

To enable this access, the organization-specific DNS host record must be configured to direct traffic to the public IP of the CRM deployment:

- Create a DNS host record for alpineskihouse.crm.consolidatedmessenger.com to the destination IP of the CRM front-end servers.

6.1.1.5 Refresh the CRM Organization Identifiers in AD FS

To authenticate successfully to a CRM organization secured using AD FS for claims-based authentication, the Identifiers for the relying party trust must include the organization specific identifier. By default, AD FS updates the CRM identifiers once daily. To verify access to the upgraded site, the identifiers must be manually updated.

Important

Consider changing the automatic update interval for AD FS for your environment. This will depend upon when and how often CRM organizations are provisioned, the expected delay between provisioning and customer sign-on, and the expense of performing the update against the CRM front-end systems.

1. On the computer that is running the AD FS 2.1 federation server, start **AD FS 2.1 Management**.
2. Click **AD FS \Trust Relationships** and select **Relying Party Trusts**.
3. In the details pane, click the relying party trust for the CRM deployment.
4. In the **Action** pane, click **Update from Federation Metadata**.
5. In the properties dialog box for the trust, click **Update** to perform the update.

6.1.1.6 CRM Organization Administrator Provisioning

This example shows provisioning an administrator within the context of a single organization.

To enable an Alpine Ski House Customer Administrator Account for hosted Microsoft Dynamics CRM:

Note

Any users you add must already be in the Active Directory Domain Services. After the users are set up, the users use their password from Active Directory to access Microsoft Dynamics CRM. Users cannot be deleted, but can be deactivated.

1. Log on to the client computer.

Note

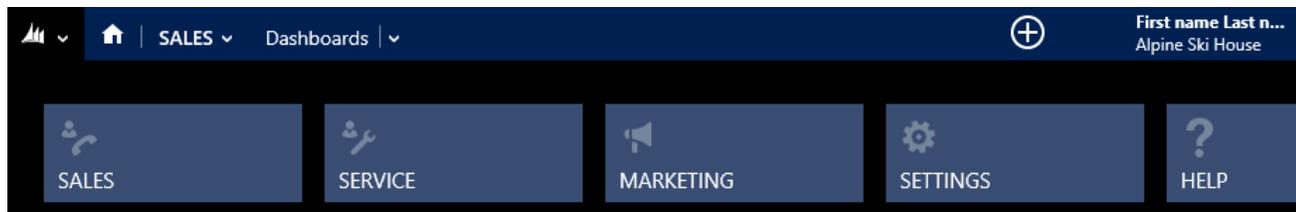
Ensure that you disable the Pop-Up Blocker in Windows Internet Explorer® before continuing.

2. Start Microsoft Internet Explorer and browse to <https://alpineskihouse.crm.consolidatedmessenger.com>.
3. When prompted to login to **Microsoft Dynamics CRM**, enter the credentials of the CRM Deployment Service account (that is, **CRMDeploySvc**). Click **Sign In**.

Note

When CRM organizations are provisioned in an environment with the CRM Deployment web services configured to use a service account. CRM assigns that service account as the default administrator of the CRM organization. Provisioning of the organization resources must be done using this account, until an alternate administrative user has been created.

4. In the upper left corner of the **Microsoft Dynamics CRM** web page, select the down arrow next to the Dynamics icon () and select **Settings** from the menu bar.



5. At the **Business Management** page, select the down arrow next to the **Settings** and select **Administration** from the menu bar.



6. On the **Administration** page, click **Users**.
7. On the **Enabled Users** page, click **New**.
8. In the **User Name** field, type new organization administrator user name (that is, **contoso\johnc_alpineskihouse**).
9. In the **First Name** field, verify or enter the user's first name.
10. In the **Last Name** field, verify or enter the user's last name.
11. Verify or enter additional user information in the fields provided as desired.

12. From the **New User** toolbar, click **Save** to save the new user.

Note: After you save this user record, you must assign the user's security roles.

13. On the toolbar below the **Settings** menu bar, click the ellipse (...) symbol and select **Manage Roles**.
14. In the **Manage User Roles** dialog box, select **System Administrator**, and then click **OK**.
15. Click **Save & Close** to save the user settings.

To verify that the new user has administrative access to the CRM organization, close the browser and reconnect to the organization specific URL. When prompted to login, authenticate as the new Administrative user for the organization and verify access to the Administration options in the Settings application.

6.1.2 Business Unit Provisioning

Depending upon the corporate structure of the hosted tenant organization, business units may be necessary for the CRM organization. To provision a business unit for the organization, use the CRM web interface as demonstrated below.

6.1.2.1 Provision a Business Unit for hosted Microsoft Dynamics CRM Organization

1. Log on to a client computer.

Note: Ensure that you disable the pop-up blocker in Internet Explorer before continuing.
2. Start **Microsoft Internet Explorer** and browse to <https://alpineskihouse.crm.consolidatedmessenger.com>.
3. When prompted to login to **Microsoft Dynamics CRM**, enter the credentials of a System Administrator for the organization. Click **Sign In**.
4. In the upper left corner of the **Microsoft Dynamics CRM** web page, select the down arrow next to the Dynamics icon () and select **Settings** from the menu bar.
5. At the **Business Management** page, select the down arrow next to the **Settings** and select **Administration** from the menu bar.
6. On the **Administration** page, click **Business Units**.
7. On the **Business Units** page, click **New**.
8. In the **Information** page, in the **Name** field, enter the name of the new business unit.
9. Enter additional business unit information in the fields provided as desired.
10. From the toolbar, click **Save & Close**.

6.1.3 User Provisioning

You can add one or more tenant organization users by using the **Administration** area in the CRM web client.

6.1.3.1 Enable a Customer User Account for CRM

Note: Any users you add must already be in the Active Directory directory service. After the users are set up, the users use their password from Active Directory to access Microsoft Dynamics CRM. Users cannot be deleted, but can be deactivated.

1. Log on to the client computer.

Note: Ensure that you disable the Pop-Up Blocker in Internet Explorer before continuing.

2. Start Microsoft Internet Explorer and browse to **https://alpineskihouse.crm.consolidatedmessenger.com**.
3. When prompted to login to Microsoft Dynamics CRM, enter the credentials of a System Administrator for the organization, such as **johnc@alpineskihouse.com**. Click **Sign In**.
4. In the upper left corner of the **Microsoft Dynamics CRM** web page, select the down arrow next to the Dynamics icon () and select **Settings** from the menu bar.
5. At the **Business Management** page, select the down arrow next to **Settings** and select **Administration** from the menu bar.
6. On the **Administration** page, click **Users**.
7. On the **Enabled Users** page, click **New**.
8. In the **User Name** field, type new organization administrator user name (that is, **contoso\kima_alpineskihouse**).
9. In the **First Name** field, verify or enter the user's first name.
10. In the **Last Name** field, verify or enter the user's last name.
11. Verify that the **Business Unit** is set to the appropriate value for this user.
12. Verify or enter additional user information in the fields provided as desired.

Important

The values on the user information form are not restricted, and everyone in the system can view the information about the user.

13. From the **New User** toolbar, click **Save** to save the new user.

Note

After you save this user record, you must assign the user's security roles.

14. On the toolbar below the **Settings** menu bar, click the ellipse (...) symbol and select **Manage Roles**.
15. In the **Manage User Roles** dialog box, select **Sales Manager** and then click **OK**.
16. Click **Save & Close** to save the user settings..

6.1.4 Enabling CRM Organization and Users for Email Routing

You have now successfully configured Microsoft Dynamics CRM to use the email router for incoming and outgoing email.

6.1.4.1 Team Provisioning

To provision a Team for the CRM organization or Business Unit, use the CRM web interface as demonstrated below.

6.1.4.1.1 Provisioning a Team for Hosted Microsoft Dynamics CRM Organization

1. Log on to a client computer.
2. Start Microsoft Internet Explorer and browse to **https://alpineskihouse.crm.consolidatedmessenger.com**.

3. When prompted to log in to Microsoft Dynamics CRM, enter the credentials of a System Administrator for the organization. Click **Sign In**.
4. In the upper left corner of the **Microsoft Dynamics CRM** web page, select the down arrow next to the Dynamics icon () and select **Settings** from the menu bar.
5. At the **Business Management** page, select the down arrow next to the **Settings** and select **Administration** from the menu bar.
6. On the **Administration** page, click **Teams**.
7. In the Local Business Teams page, click **New**.
8. In the **New** team page, in the **Name** field, enter the name of the new team.
9. Verify that the **Business Unit** is set to the appropriate value for this team.
10. In the **Administrator** field, enter the name of the team administrator or use the **Find** icon to locate the user name.
11. Complete the rest of the form as desired.
12. From the toolbar, click **Save**.

6.1.4.2 Create forward mailboxes or edit mailboxes

By default, when users and queues are created in Microsoft Dynamics CRM, their respective mailbox records are also created. These mailbox records contain information that is specific to an individual mailbox on the email server, like email address, mailbox credentials, and email synchronization method. To process email messages using server-side synchronization for users and queues, their respective mailbox records should be associated to an email server profile record in Microsoft Dynamics CRM.

If your organization wants to configure server-side synchronization using a forward mailbox, you can create a new forward mailbox record. A forward mailbox is used as a collection box for email messages that are transferred from each user's mailbox on the email system by a server-side rule. The forward mailbox must be dedicated to server-side synchronization, and must not be used as a working mailbox by an individual user. This can be used to process email messages for users and queues whose mailboxes have **Incoming Email Synchronization Method** set to **Forward Mailbox**. You must associate the forward mailbox record to an email server profile record to process email using server-side synchronization.

1. In the upper left corner of the Microsoft Dynamics CRM web page, select the down arrow next to the Dynamics icon () and select **Settings** from the menu bar.
2. At the **Business Management** page, select the down arrow next to the **Settings** and select **Email Configuration** from the menu bar.
3. On the Email Configuration page, select **Mailboxes**.
4. In the Mailboxes page, click **New Forward Mailbox**, or to edit an existing mailbox record, open the mailbox record and specify the following details:

Fields	Description
General	
Name	Type a meaningful name for the mailbox.

Owner	Shows the owner of the mailbox. For a user mailbox that is automatically populated, the owner of the mailbox is the user itself. For a queue mailbox that is automatically populated, the owner of the mailbox is the owner of the queue record.
Email address	Type the email address for the forward mailbox, such as forwardmailbox@contoso.com. For a user or a queue mailbox, the email address is the same as that specified in the corresponding user or queue record form. If you edit the email address here, the email address in the user or queue record is updated automatically.
Delete Emails After Processing	Specify if you want to delete email from the mailbox after processing. This field is available and can be set to Yes only for a forward mailbox and a queue mailbox.
Regarding	Select the user or queue that the mailbox is associated with. This field is empty and cannot be set for a forward mailbox.
Is Forward Mailbox	This field indicates whether the mailbox record is a forward mailbox. When set to No , it indicates that the mailbox record is associated to an individual user or queue in Microsoft Dynamics CRM.
Credentials	
Allow to Use Credentials for Email Processing	Click Yes if the email server profile associated to this mailbox has Authenticate Using set to Credentials Specified by a User or Queue . You must provide the username and password when this field is set to Yes . These credentials will be used to send and receive email from the mailbox on the email server. To ensure the credentials are secured in CRM, SQL encryption is used to encrypt the credentials stored in the mailbox if you're processing email by using server-side synchronization.
Synchronization Method	
Server Profile	Select the email server profile that is used for email processing for this mailbox.
Incoming Email	Select the delivery method for incoming email. This will determine how incoming email will be accessed for this mailbox. <ul style="list-style-type: none"> ○ None. Email won't be received. ○ Forward Mailbox. Email will be received using a forward mailbox.

	<ul style="list-style-type: none"> ○ Microsoft Dynamics CRM for Outlook. Email is received by using CRM for Outlook. ○ Server-Side Synchronization or Email Router. Email is received by using server-side synchronization or the Email Router.
Outgoing Email	<p>Select the delivery method for outgoing email. This determines how outgoing email will be sent for this mailbox.</p> <ul style="list-style-type: none"> ○ None. Email won't be sent. ○ Microsoft Dynamics CRM for Outlook. Email is received by using CRM for Outlook. ○ Server-Side Synchronization or Email Router. Email is sent by using server-side synchronization or Email Router. <p>For a forward mailbox, only None is allowed.</p>
Appointments, Contacts, and Tasks	<p>Select whether you want to use CRM for Outlook or server-side synchronization to synchronize appointments, contacts, and tasks in CRM. If you select None, appointments, contacts, and tasks won't be synchronized.</p>
Configuration Test Results	
Incoming Email Status	<p>Show the result of the email configuration test for incoming email. The various statuses can be:</p> <ul style="list-style-type: none"> ○ Not Run. The email configuration test has not been run for this mailbox. ○ Success. The incoming email has been configured and email can be received for this mailbox. ○ Failure. The incoming email has been configured but it is not possible to pull email from the corresponding configured mailbox.
Outgoing Email Status	<p>Show the result of the email configuration test for outgoing email. The various statuses can be:</p> <ul style="list-style-type: none"> ○ Not Run. The email configuration test hasn't been run for this mailbox. ○ Success. The outgoing email has been configured and email can be sent from this mailbox. ○ Failure. The outgoing email has been configured but it's not possible to send email from the corresponding configured mailbox.
Appointments,	Show the result of the synchronization of appointments, contacts, and tasks.

Contacts, and Tasks Status	The various statuses can be: <ul style="list-style-type: none">○ Not Run. The synchronization has not been tested for this mailbox.○ Success. Appointments, contacts, and tasks can be synchronized for this mailbox.○ Failure. Appointments, contacts, and tasks can't be synchronized for this mailbox.
Mailbox Test Completed On	This field shows the date and time when the email configuration was tested for this mailbox record.

5. Click **Save** or **Save & Close**.

6.1.4.3 Set the delivery method for incoming and outgoing email

When a user or queue is created in Microsoft Dynamics CRM, a mailbox for each user and queue is also created. To process email for these mailboxes, the incoming and outgoing delivery method must be set.

1. Before you start, make sure you have the administrator permissions. You must also have read and write privileges on the mailbox entity to edit mailboxes to set the delivery method.
2. On the Navigation bar, click **Microsoft Dynamics CRM > Settings**.
3. On the Navigation bar, click or tap **Settings > Email Configuration**.
4. Click **Mailboxes**.
5. For each user or queue whose email must be processed through server-side synchronization, set **Incoming Email to Email Router or Server-Side Synchronization** or **Forward Mailbox**, and **Outgoing Email to Email Router or Server-Side Synchronization**.

To set the default delivery methods for multiple mailboxes, in the list of mailboxes, select the mailboxes, and then click or tap the **More Commands (...)** icon > **Apply Default Email Settings**. [System Settings dialog box - Email tab](#).

To open a mailbox associated to a specific queue or user, open the queue or user record, and in the **Mailbox** lookup, click or tap the mailbox name.

6.1.5 Security Role Provisioning

Due to the complexity in defining security roles in CRM, the recommended approach to creating a new security role is to copy an existing role and modify the permissions. The following procedure is an example of how to copy and modify an existing security role.

6.1.5.1 Copying a Security Role

1. Log on to a client computer.
2. Start Microsoft Internet Explorer and browse to **https://alpineskihouse.crm.consolidatedmessenger.com**.
3. When prompted to login to Microsoft Dynamics CRM, enter the credentials of a System Administrator for the organization. Click **Sign In**.

4. In the upper left corner of the **Microsoft Dynamics CRM** web page, select the down arrow next to the Dynamics icon () and select **Settings** from the menu bar.
5. At the **Business Management** page, select the down arrow next to the **Settings** and select **Administration** from the menu bar.
6. On the **Administration** page, click **Security Roles**.
7. Select an existing role that closely matches the new role permissions. On the **Actions** menu, select **Copy Role**.
8. In the **Copy Security Role** window, in the **New Role Name** field, enter the new security role name and then click **OK**.
9. In the **Security Role** window, use the tabs to modify the permissions to CRM objects as desired. Click the **Save** icon to save changes on each tab.
10. When finished, click **Save & Close**.

6.1.6 Field Security Profile Provisioning

In Microsoft Dynamics CRM, you can use field-level security to restrict access on certain fields to specific users and teams. This may be used to enable for only certain users the ability to update a field, while others have read-only access. At this time, field security can only be applied to custom fields.

Note

Before applying field permissions in Microsoft Dynamics CRM 2013, you must have a custom field defined in Dynamics CRM. For Microsoft Dynamics CRM 2015, most fields on default and custom forms can be secured.

6.1.6.1 Creating a Field Security Profile

1. Log on to a client computer.
2. Start Microsoft Internet Explorer and browse to **<https://alpineskihouse.crm.consolidatedmessenger.com>**.
3. When prompted to login to Microsoft Dynamics CRM, enter the credentials of a System Administrator for the organization. Click **Sign In**.
4. In the upper left corner of the **Microsoft Dynamics CRM** web page, select the down arrow next to the Dynamics icon () and select **Settings** from the menu bar.
5. At the **Business Management** page, select the down arrow next to the **Settings** and select **Administration** from the menu bar.
6. On the **Administration** page, click **Field Security Profiles**.
7. In the **Field Security Profiles** page, click **New**.
8. In the **Field Security Profile** page, enter the **Name** and **Description** of the new profile and click the **Save** icon.
9. Under **Common**, click **Field Permissions**.
10. Select a field and then click **Edit**. Select the types of permissions the users and teams assigned to the field security profile will have for the secured field and click **OK**.
11. Under **Related**, select **Teams** or **Users** depending upon how you want to add users. From the **More Actions** toolbar, click **Add**.

12. In the **Look Up Records** page, select a team or user from the list or search for the team or user and select it, then click **Add**.
13. When finished, click **OK**.

6.1.7 Language Provisioning

Before users can display the CRM client interfaces and help in a different language, the Language Pack must be installed and provisioned in the Microsoft Dynamics CRM deployment. For users running Microsoft Dynamics CRM for Outlook, they must also install the preferred Language Pack on their computer.

1. To install a Language Pack on the computer running Microsoft Dynamics CRM Front-end server roles or the Microsoft Dynamics CRM for Outlook client, download the Language Pack(s) and perform the procedure described in [Step 1: Install the Language Pack](#).
2. To provision an installed Language Pack for an organization, login to the CRM organization with a CRM system administrator user and perform the procedure described in [Step 2: Provision the Language Pack](#).
3. To change the display and help language for the Microsoft Dynamics CRM web client or the Microsoft Dynamics CRM for Outlook client, see [Step 3: Select the language to display the user interface and help](#).

6.1.8 Troubleshooting Options

This section describes how you can configure error reporting for your deployment. It also provides references to information about setting up traces to help find issues if you need to perform troubleshooting.

6.1.8.1 Windows Error Reporting

By default, automatic error reporting is not enabled in Microsoft Dynamics CRM. To send error reports to Microsoft, Windows Error Reporting (WER) must be enabled; to send error reports generated from the Microsoft Dynamics CRM web application, WER must be enabled on the computer where Microsoft Dynamics CRM Server is running. Similarly, to send reports generated from Microsoft Dynamics CRM for Outlook, WER must be enabled where Microsoft Office Outlook is running.

For more details, see [Enable Windows Error Reporting](#).

6.1.8.2 Trace Microsoft Dynamics CRM

In Microsoft Dynamics CRM, you can create trace files that monitor the actions that are performed by the server and client applications. Trace files are helpful when you have to troubleshoot error messages or other issues in Microsoft Dynamics CRM.

For more details about how to enable tracing and cautions of tracing, see [Monitor and troubleshoot Microsoft Dynamics CRM](#).

6.1.8.3 Trace Microsoft Dynamics CRM Reporting Extensions

In Microsoft Dynamics CRM, you can create trace files that monitor the actions that are performed by Microsoft Dynamics CRM Reporting Extensions for SQL Server Reporting Services. Trace files are helpful when you have to troubleshoot error messages or other issues in CRM Reporting Extensions.

For more details, see [Enable tracing for Microsoft Dynamics CRM Reporting Extensions for SQL Server Reporting Services](#).

6.2 Automated Provisioning

The following section provides a guide to automating provisioning tasks using .NET Framework code and the Microsoft Dynamics CRM SDK. It is intended to augment the SDK by focusing on a specific subset of tasks that are relevant to a service provider or an ISV with a need to enhance a Control Panel with the ability to provision tenant organizations for Microsoft Dynamics CRM. Before reading this section we strongly recommend you first become familiar with the SDK, as it provides a wide range of instructive and practical information. In addition to comprehensive API references, the SDK contains tutorials, sample code, conceptual overviews, security considerations, and helper class libraries for simplifying CRM development. You can find more information regarding the Microsoft Dynamics CRM SDK in the [Microsoft Dynamics CRM Developer Center](#).

The focus of this guide is to provide simple samples written in C# that demonstrate how to perform a variety of provisioning tasks using the Microsoft Dynamics CRM web services. These tasks include:

- Deploying and Managing Tenant Organizations
- Provisioning Business Units
- Provisioning Users, Teams, and Roles

Microsoft Dynamics CRM is natively multi-tenant, supporting hosting of multiple tenants in a single CRM deployment, but it does not mandate a specific Active Directory Organizational Unit tenant structure. It is important, therefore, that organization's and user accounts exist in Active Directory prior to attempting these samples.

6.2.1 Prerequisites

You must have Microsoft Visual Studio® 2010 or a later version and you must be able to log into Microsoft Dynamics CRM.

6.2.2 Using the CRM Dynamics Deployment Web Service to Provision Tenant Organizations

In Microsoft Dynamics CRM, you can use the deployment web service (Deployment Service) as a programmatic alternative to using the Deployment Manager. The Deployment Service is exposed as a Windows Communication Foundation (WCF) endpoint as shown in the following URL example:

```
http://myservername/xrmdeployment/2013/deploymentservice.svc
```

There are several options for interacting with this service in code including:

- Adding a Service Reference to your .NET project.
- Using the svcutil.exe utility to generate a proxy class for the service
- Adding a reference to the assembly **Microsoft.Xrm.Sdk.Deployment.dll**, which you can find in the **SDK\bin** folder.

Most of the time, you will choose to use the SDK assemblies in your development projects. Using the assemblies is the recommended developer scenario for Microsoft Dynamics CRM and Microsoft Dynamics CRM Online. These assemblies are in the **SDK\Bin** folder, along with the corresponding Visual Studio IntelliSense® XML files.

For the examples in this section, however, we will use the technique of adding a Service Reference to a C# console project for those scenarios where using the reference assemblies is not practical. Subsequent examples using the **IDiscoverService** and **IOrganizationService** will illustrate the use of the SDK reference

assemblies. For more information on using the Deployment Service, see [Deployment Web Service in Microsoft Dynamics CRM](#).

6.2.2.1 Create a Sample Deployment Service Project

The samples in this section will, for simplicity, be hosted in a C# Console project.

6.2.2.1.1 Creating a Console Application in Visual Studio

1. On the **File** menu in Microsoft Visual Studio, point to **New** and then click **Project**.
2. In the **New Project** dialog box, select a language in the **Installed Templates** box, and then select the **Console Application** template.
3. Type **CrmDeploymentSample** for the application name in the **Name** box, and in the **Location** box, type the path to where you want the application to be created, and then click **OK**.

6.2.2.2 Add the Deployment Service to your Sample Project

A service reference enables a project to access one or more Windows Communication Foundation (WCF) services. Use the **Add Service Reference** dialog box to search for WCF services in the current solution, locally, on a local area network, or on the Internet.

6.2.2.2.1 Adding a Reference to the Deployment Service

1. In **Solution Explorer**, right-click the name of the project that you want to add the service to, and then click **Add Service Reference**. The **Add Service Reference** dialog box appears.
2. In the **Address** box, enter the URL for the Deployment service, for example, **http://myservername/xrmdeployment/2013/deploymentservice.svc**, and then click **Go** to search for the service.
3. In the **Service** list, expand the node for the service that you want to use and select a service contract.
4. In the **Namespace** box, enter **CrmDeploymentService** as the namespace that you want to use for the reference.
5. Click **OK** to add the reference to the project.

A service client (proxy) is generated, and metadata describing the service is added to the **app.config** file. Double-click the **app.config** file to load it in the editor. Following is an example of the metadata that is added to the configuration file:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <system.serviceModel>
    <bindings>
      <basicHttpBinding>
        <binding name="None_BasicHttpBinding_IDeploymentService" closeTimeout="00:01:00"
          openTimeout="00:01:00" receiveTimeout="00:10:00" sendTimeout="00:01:00"
          allowCookies="false" bypassProxyOnLocal="false" hostNameComparisonMode="StrongWildcard"
          maxBufferSize="65536" maxBufferPoolSize="524288" maxReceivedMessageSize="65536"
          messageEncoding="Text" textEncoding="utf-8" transferMode="Buffered"
          useDefaultWebProxy="true">
          <readerQuotas maxDepth="32" maxStringContentLength="8192" maxArrayLength="16384"
            maxBytesPerRead="4096" maxNameTableCharCount="16384" />
          <security mode="TransportCredentialOnly">
            <transport clientCredentialType="Windows" proxyCredentialType="None"
              realm="" />
            <message clientCredentialType="UserName" algorithmSuite="Default" />
          </security>
        </binding>
      </basicHttpBinding>
    </bindings>
  </system.serviceModel>
</configuration>
```

```

    </security>
  </binding>
</basicHttpBinding>
<customBinding>
  <binding name="CustomBinding_IDeploymentService">
    <security defaultAlgorithmSuite="Default" authenticationMode="SspiNegotiated"
      requireDerivedKeys="true" securityHeaderLayout="Strict" includeTimestamp="true"
      keyEntropyMode="CombinedEntropy" messageProtectionOrder="SignBeforeEncryptAndEncryptSignature"
messageSecurityVersion="WSSecurity11WSTrustFebruary2005WSSecureConversationFebruary2005WSSecurityPolicy11BasicSec
urityProfile10"
      requireSecurityContextCancellation="true" requireSignatureConfirmation="false">
    <localClientSettings cacheCookies="true" detectReplays="true"
      replayCacheSize="900000" maxClockSkew="00:05:00" maxCookieCachingTime="Infinite"
      replayWindow="00:05:00" sessionKeyRenewalInterval="10:00:00"
      sessionKeyRolloverInterval="00:05:00" reconnectTransportOnFailure="true"
      timestampValidityDuration="00:05:00" cookieRenewalThresholdPercentage="60" />
    <localServiceSettings detectReplays="true" issuedCookieLifetime="10:00:00"
      maxStatefulNegotiations="128" replayCacheSize="900000" maxClockSkew="00:05:00"
      negotiationTimeout="00:01:00" replayWindow="00:05:00" inactivityTimeout="00:02:00"
      sessionKeyRenewalInterval="15:00:00" sessionKeyRolloverInterval="00:05:00"
      reconnectTransportOnFailure="true" maxPendingSessions="128"
      maxCachedCookies="1000" timestampValidityDuration="00:05:00" />
    <secureConversationBootstrap />
  </security>
</binding>
<textMessageEncoding maxReadPoolSize="64" maxWritePoolSize="16"
  messageVersion="Default" writeEncoding="utf-8">
  <readerQuotas maxDepth="32" maxStringContentLength="8192" maxArrayLength="16384"
    maxBytesPerRead="4096" maxNameTableCharCount="16384" />
</textMessageEncoding>
<httpTransport manualAddressing="false" maxBufferPoolSize="524288"
  maxReceivedMessageSize="65536" allowCookies="false" authenticationScheme="Anonymous"
  bypassProxyOnLocal="false" decompressionEnabled="true" hostNameComparisonMode="StrongWildcard"
  keepAliveEnabled="true" maxBufferSize="65536" proxyAuthenticationScheme="Anonymous"
  realm="" transferMode="Buffered" unsafeConnectionNtlmAuthentication="false"
  useDefaultWebProxy="true" />
</binding>
</customBinding>
</bindings>
<client>
  <endpoint address="http://hex2010-crm01:5555/XRMDeployment/2011/Deployment.svc"
    binding="customBinding" bindingConfiguration="CustomBinding_IDeploymentService"
    contract="DeploymentService.IDeploymentService" name="CustomBinding_IDeploymentService">
    <identity>
      <userPrincipalName value="CONTOSO\administrator" />
    </identity>
  </endpoint>
  <endpoint address="http://hex2010-crm01:5555/XRMDeployment/2011/Deployment.svc/web"
    binding="basicHttpBinding" bindingConfiguration="None_BasicHttpBinding_IDeploymentService"
    contract="DeploymentService.IDeploymentService" name="None_BasicHttpBinding_IDeploymentService" />

```

```

</client>
</system.serviceModel>
</configuration>

```

Note

Two binding definitions were added to the configuration file. One of the bindings is of type **BasicHttpBinding** and the other is of type **CustomBinding**. All the examples that follow use the **CustomBinding** configuration which is identified by the name **CustomBinding_IDeploymentService** when instantiating the service proxy.

6.2.2.2.2 Adding a Using Declaration for the Deployment Service Proxy Class Namespace

1. In **Solution Explorer**, double-click the **Program.cs** file to open it in the editor.
2. Add a using declaration to the top of the **Program.cs** class file for the service proxy namespace:

```
using CrmDeploymentSample.CrmDeploymentService;
```

6.2.2.3 Create a New Tenant Organization

The following code example demonstrates how to create a new organization in Microsoft Dynamics CRM. The Deployment Service implements a request/response pattern for this operation using the **BeginCreateOrganizationRequest** and **BeginCreateOrganizationResponse** classes respectively. The operation is asynchronous, so there will be some latency between the time the call is made and when the organization is an active state in CRM.

6.2.2.3.1 Example Code C#

```

Guid operationId = Guid.Empty;
// instantiating the DeploymentServiceClient in a using statement ensures that the client
// communication channel is closed and the object is disposed when falling out of scope.

// CustomBinding_IDeploymenService is the name of the configuration setting for the CustomBinding
using (DeploymentServiceClient client = new DeploymentServiceClient("CustomBinding_IDeploymentService"))
{
    // Set properties for the new organization
    Organization organization = new Organization
    {
        BaseCurrencyCode = "USD",
        BaseCurrencyName = "US Dollar",
        BaseCurrencyPrecision = 2,
        BaseCurrencySymbol = "$",
        BaseLanguageCode = 1033,
        FriendlyName = "Alpine Ski House",
        UniqueName = "AlpineSkiHouse",
        SqlCollation = "Latin1_General_CI_AI",
        SqlServerName = "CRM01",
        SrsUrl = "http://CRM01/ReportServer",
        SqmIsEnabled = false
    };

    // Create a request for the deployment service
    BeginCreateOrganizationRequest request = new BeginCreateOrganizationRequest();
    request.Organization = organization;

    // Execute the request
    BeginCreateOrganizationResponse response = (BeginCreateOrganizationResponse)client.Execute(request);

```

```

// The operation is asynchronous, so the response object contains a unique identifier
// for the operation
operationId = response.OperationId;

}

```

6.2.2.4 Retrieve a Tenant Organization

After creating a tenant organization you can retrieve the details of the organization deployment object using the **Retrieve** method of the Deployment Service. An organization deployment object has five possible states as defined by the **OrganizationState** enum:

- Disabled
- Enabled
- Failed
- Maintenance
- Pending

6.2.2.4.1 Example Code C#

```

// instantiating the DeploymentServiceClient in a using statement ensures that the client
// communication channel is closed and the object is disposed when falling out of scope.

// CustomBinding_IDeploymenService is the name of the configuration setting for the CustomBinding

using (DeploymentServiceClient client = new DeploymentServiceClient("CustomBinding_IDeploymentService"))
{

    //The EntityInstanceid class is used to identify the object you want to
    // retrieve. You have the option of using name or ID.
    EntityInstanceid id = new EntityInstanceid
    {
        Name = "AlpineSkiHouse"
    };
    //Retrieve the organization
    Organization organization = (Organization)client.Retrieve(DeploymentEntityType.Organization, id);

    client.Close();
}

```

Following is an example of an **Organization** class instance serialized to XML:

```

<?xml version="1.0" encoding="utf-8"?>
<Organization xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <ExtensionData />
  <BaseCurrencyCode>USD</BaseCurrencyCode>
  <BaseCurrencyName>US Dollar</BaseCurrencyName>
  <BaseCurrencyPrecision>2</BaseCurrencyPrecision>
  <BaseCurrencySymbol>$</BaseCurrencySymbol>
  <BaseLanguageCode>1033</BaseLanguageCode>
  <DatabaseName>AlpineSkiHouse_MSCRM</DatabaseName>
  <FriendlyName>Alpine Ski House</FriendlyName>
  <Id>273c0010-5973-43d4-b666-8d899ca8c6ab</Id>
  <SqlCollation>Latin1_General_CI_AI</SqlCollation>
  <SqlServerName>CRM01</SqlServerName>
  <SqlIsEnabled>>false</SqlIsEnabled>

```

```
<SrsUrl>http://crm01/reportserver</SrsUrl>
<State>Pending</State>
<UniqueName>AlpineSkiHouse</UniqueName>
<Version>5.0.9688.34</Version>
</Organization>
```

6.2.2.5 Modify a Tenant Organization

The following code example demonstrates how to modify an organization in Microsoft Dynamics CRM using the **Update** method of the Deployment Service. There are a very limited number of properties that can be modified on an organization deployment object. These are:

- **FriendlyName** – The display name of the organization.
- **SqlServerName** - The name of computer that is running the SQL Server database for the organization.
- **SrsUrl** – The URL of SQL Reporting Services instance used by the organization.

It is also necessary to place the organization in a **Disabled** state prior to editing these attributes. Once new values have been applied, the organization state can be changed back to **Enabled**. This is shown in the following example code.

6.2.2.5.1 Example Code C#

```
using (DeploymentServiceClient client = new DeploymentServiceClient("CustomBinding_IDeploymentService"))
{
    EntityInstanceId id = new EntityInstanceId
    {
        Name = "AlpineSkiHouse2"
    };

    // disable the organization
    Organization organization = (Organization)client.Retrieve(DeploymentEntityType.Organization, id);
    organization.State = OrganizationState.Disabled;

    client.Update(organization);

    // set the friendly name
    organization.FriendlyName = "Alpine Ski House Outfitters and Guides";

    client.Update(organization);

    // re-enable the organization
    organization.State = OrganizationState.Enabled;
    client.Update(organization);
}
```

6.2.2.6 Disable and Enable a Tenant Organization

The following code example demonstrates how to disable or disable an organization in Microsoft Dynamics CRM using the **Update** method of the Deployment Service.

6.2.2.6.1 Example Code C# - Disable

```
using (DeploymentServiceClient client = new DeploymentServiceClient("CustomBinding_IDeploymentService"))
{
```

```

EntityInstance id = new EntityInstance
{
    Name = "AlpineSkiHouse"
};

Organization organization = (Organization)client.Retrieve(DeploymentEntityType.Organization, id);
organization.State = OrganizationState.Disabled;
client.Update(organization);

client.Close();
}

```

6.2.2.6.2 Example Code C# - Enable

```

using (DeploymentServiceClient client = new DeploymentServiceClient("CustomBinding_IDeploymentService"))
{
    EntityInstance id = new EntityInstance
    {
        Name = "AlpineSkiHouse"
    };

    Organization organization = (Organization)client.Retrieve(DeploymentEntityType.Organization, id);
    organization.State = OrganizationState.Enabled;
    client.Update(organization);

    client.Close();
}

```

6.2.2.7 Delete a Tenant Organization

The following code example demonstrates how to delete an organization in Microsoft Dynamics CRM using the **Delete** method of the Deployment Service. To delete an organization, you must first set the organization state to **Disabled**. This is shown in the following example code.

6.2.2.7.1 Example Code C#

```

using (DeploymentServiceClient client = new DeploymentServiceClient("CustomBinding_IDeploymentService"))
{
    EntityInstance id = new EntityInstance
    {
        Name = "AlpineSkiHouse2"
    };

    // disable the organization
    Organization organization = (Organization)client.Retrieve(DeploymentEntityType.Organization, id);
    organization.State = OrganizationState.Disabled;
    client.Update(organization);

    // delete the organization
    DeleteRequest request = new DeleteRequest
    {
        EntityType = DeploymentEntityType.Organization,
        InstanceTag = id
    };
    DeleteResponse response = (DeleteResponse)client.Execute(request);
}

```

```
client.Close();  
}
```

6.2.2.7.2 *Remarks*

Deleting an organization does not automatically remove the database associated with the organization. This allows administrators to archive customer data, or recover customer data in the event of an accidental deletion. It will not be possible to create a new organization with the same database name until that database is removed from the system.

6.2.3 Using the CRM Dynamics Web Services to Provision Tenant Organization Objects

Microsoft Dynamics CRM provides two web services. These services can be used to identify your organization and to access Microsoft Dynamics CRM data.

6.2.3.1 IDiscoveryService Web Service

A single Microsoft Dynamics CRM installation can host multiple organizations on multiple servers. Therefore, it is important to specify which organization you want to access. The **IDiscoveryService** web service returns a list of organizations to which the specified user belongs, and the URL endpoint address for each organization. For more information regarding the **IDiscoveryService** web service, see the [IDiscoveryService Interface](#) topic of the Microsoft Dynamics CRM SDK.

6.2.3.2 IOrganizationService Web Service

The primary web service for accessing data and metadata in Microsoft Dynamics CRM is the **IOrganizationService** web service. For more information regarding the **IOrganizationService** web service, see the [IOrganizationService Interface](#) topic of the Microsoft Dynamics CRM SDK.

6.2.3.3 Add the Microsoft Dynamics CRM Assemblies to Your Project

Most of the time, you will choose to use the SDK assemblies in your development projects. Using the assemblies is the recommended developer scenario for Microsoft Dynamics CRM and Microsoft Dynamics CRM Online. These assemblies are in the **SDK\Bin** folder, along with the corresponding Visual Studio IntelliSense XML files.

6.2.3.3.1 *Adding a Reference to the Microsoft Dynamics CRM Assemblies*

1. In **Solution Explorer**, right-click the name of the project that you want to add the service to, and then click **Add Reference**. The **Add Reference** dialog box appears.
2. Select the **Browse** tab and navigate to the **SDK\Bin** folder.
3. Click the **Microsoft.Xrm.Sdk.dll** assembly and click **OK**.
4. Repeat this process for **Microsoft.Crm.Sdk.Proxy.dll**.

You will also need to know the endpoint address (URL) of the Discovery Service in your deployment topology. The URL of the Discovery Service takes the following form:

```
http[s]://<hostname[:port]>/XRMServices/2011/Discovery.svc
```

6.2.3.4 Retrieve Discovery Details for a Tenant Organization

The following code example demonstrates how to retrieve the discovery details for a tenant organization using a **RetrieveOrganizationRequest** and the **Execute** method of the **IDiscoveryService**.

6.2.3.4.1 Example Code C#

```
// set the endpoint uri
Uri serviceUri = new Uri("https://auth.consolidatedmessenger.com/XRMServices/2011/Discovery.svc");
IServiceConfiguration<IDiscoveryService> serviceConfig
    = ServiceConfigurationFactory.CreateConfiguration<IDiscoveryService>(serviceUri);

// create a client credential for authorization via IFD.
ClientCredentials credentials = new ClientCredentials();
credentials.UserName.Password = "Pass3word";
credentials.UserName.UserName = "contoso\\crmprovacct";
// Connect to the Discovery service.
// The using statement assures that the service proxy will be properly disposed.
using (DiscoveryServiceProxy proxy = new DiscoveryServiceProxy(serviceConfig, credentials))
{
    // Retrieve details about a single organization discoverable via the
    // Discovery service.
    RetrieveOrganizationRequest request = new RetrieveOrganizationRequest
    {
        UniqueName = "AlpineSkiHouse",
        AccessType = EndpointAccessType.Default,
        Release = OrganizationRelease.Current
    };

    // Execute the request
    RetrieveOrganizationResponse response = (RetrieveOrganizationResponse)proxy.Execute(request);
}

```

6.2.3.4.2 Remarks

The returned **RetrieveOrganizationResponse** object will contain the endpoints for the organization's web application, organization service, and organization data service. Also returned are properties of the organization. The following XML is a serialized example of the **RetrieveOrganizationResponse** object:

```
<RetrieveOrganizationResponse xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://schemas.microsoft.com/xrm/2011/Contracts/Discovery">
  <Detail>
    <Endpoints xmlns:d3p1="http://schemas.datacontract.org/2004/07/System.Collections.Generic">
      <KeyValuePairOfEndpointTypestringztYlk6OT>
        <d3p1:key>WebApplication</d3p1:key>
        <d3p1:value>https://alpineskihouse.consolidatedmessenger.com/</d3p1:value>
      </KeyValuePairOfEndpointTypestringztYlk6OT>
      <KeyValuePairOfEndpointTypestringztYlk6OT>
        <d3p1:key>OrganizationService</d3p1:key>
        <d3p1:value>https://alpineskihouse.consolidatedmessenger.com/XRMServices/2011/Organization.svc</d3p1:value>
      </KeyValuePairOfEndpointTypestringztYlk6OT>
      <KeyValuePairOfEndpointTypestringztYlk6OT>
        <d3p1:key>OrganizationDataService</d3p1:key>
        <d3p1:value>https://alpineskihouse.consolidatedmessenger.com/XRMServices/2011/OrganizationData.svc</d3p1:value>
      </KeyValuePairOfEndpointTypestringztYlk6OT>
    </Endpoints>
    <FriendlyName>Alpine Ski House</FriendlyName>
    <OrganizationId>d0b94f78-a069-45a1-b737-e65bf9fd44c1</OrganizationId>
    <OrganizationVersion>6.0.0.514</OrganizationVersion>
    <State>Enabled</State>
    <UniqueName>alpineskihouse</UniqueName>
    <UrlName>alpineskihouse</UrlName>
  </Detail>
</RetrieveOrganizationResponse>

```

6.2.3.5 Add a User Account to a Tenant Organization

The following code example demonstrates how to add a user account to a tenant organization using the **Entity** class and the **Create** method of the **IOrganizationService**.

Note

User must already exist in Active Directory.

6.2.3.5.1 Example Code C#

```
// constants
const string BusinessUnitEntityName = "businessunit";
const string BusinessUnitIdColumnName = "businessunitid";
const string ParentBusinessUnitIdColumnName = "parentbusinessunitid";
const string SystemUserEntityName = "systemuser";
const string FirstNameColumnName = "firstname";
const string DomainNameColumnName = "domainname";
const string LastNameColumnName = "lastname";

string organizationUrl = string.Empty;
Guid userId = Guid.Empty;
// use the discovery service to obtain the IOrganizationService endpoint for
// the specified organization.
Uri serviceUri = new Uri("https://auth.consolidatedmessenger.com/XRMServices/2011/Discovery.svc");

IServiceConfiguration<IDiscoveryService> serviceConfig
    = ServiceConfigurationFactory.CreateConfiguration<IDiscoveryService>(serviceUri);

// create a client credential for authorization via IFD.
ClientCredentials credentials = new ClientCredentials();
credentials.UserName.Password = "Pass3word";
credentials.UserName.UserName = "contoso\\crmprovacct";
// Connect to the Discovery service.
// The using statement assures that the service proxy will be properly disposed.
using (DiscoveryServiceProxy proxy = new DiscoveryServiceProxy(serviceConfig, credentials))
{
    // Retrieve details about a single organization discoverable via the
    // Discovery service.
    RetrieveOrganizationRequest request = new RetrieveOrganizationRequest
    {
        UniqueName = "AlpineSkiHouse",
        AccessType = EndpointAccessType.Default,
        Release = OrganizationRelease.Current
    };

    // Execute the request
    RetrieveOrganizationResponse response = (RetrieveOrganizationResponse)proxy.Execute(request);

    // retrieve the IOrganizationService Uri
    organizationUrl = response.Detail.Endpoints[EndpointType.OrganizationService];
}

// use URL obtained by the discovery service to connect to the IOrganizationService endpoint
Uri organizationUri = new Uri(organizationUrl);

IServiceConfiguration<IOrganizationService> orgServiceConfig
    = ServiceConfigurationFactory.CreateConfiguration<IOrganizationService>(organizationUri);
```

```

// The using statement assures that the service proxy will be properly disposed.
using (OrganizationServiceProxy orgProxy = new OrganizationServiceProxy(orgServiceConfig, credentials))
{
    // Retrieve the default business unit needed to create the user.
    QueryExpression businessUnitQuery = new QueryExpression
    {
        EntityName = BusinessUnitEntityName,
        ColumnSet = new ColumnSet(BusinessUnitIdColumnName),

        Criteria =
        {
            Conditions =
            {
                new ConditionExpression(ParentBusinessUnitIdColumnName,
                    ConditionOperator.Null)
            }
        }
    };

    // Get the business unit id from the returned entity
    EntityCollection entities = orgProxy.RetrieveMultiple(businessUnitQuery);

    Guid defaultBusinessUnitId = entities[0].Id;

    //Populate an entity with data for a new system user.
    Entity entity = new Entity(SystemUserEntityName);
    entity.Attributes.Add(DomainNameColumnName, "Contoso\\jimc_alpineskihouse");
    entity.Attributes.Add(FirstNameColumnName, "Jim");
    entity.Attributes.Add(LastNameColumnName, "Chen");
    entity.Attributes.Add(BusinessUnitIdColumnName, new EntityReference
    {
        Id = defaultBusinessUnitId,
        Name = BusinessUnitEntityName,
        LogicalName = BusinessUnitEntityName
    });

    userId = orgProxy.Create(entity);
}
// return the user id
return userId;

```

6.2.3.5.2 *Remarks*

This sample demonstrates the use of late-binding with the generic entity class versus early-binding with a strongly typed class. You can use **CrmServiceUtil.exe** utility included in the SDK to generate code to interact with a strongly typed class, as shown in the following example:

```

SystemUser user = new SystemUser
{
    DomainName = "contoso\\jimc_alpineskihouse",
    FirstName = "Jim",
    LastName = "Chen",
    BusinessUnitId = new EntityReference
    {
        LogicalName = BusinessUnit.EntityLogicalName,

```

```
Name = BusinessUnit.EntityLogicalName,  
Id = defaultBusinessUnit.Id  
}  
};  
userId = serviceProxy.Create(user);
```

For more information on late-bound versus early-bound, see the [Introduction to Programming Models for Microsoft Dynamics CRM](#) SDK topic. For more information on generating and using early-bound entity classes see the [Use the Early Bound Entity Classes in Code](#) SDK topic.

6.2.3.6 Retrieve an Existing User Account

The following code example demonstrates how to retrieve a user account by **DomainName** from a tenant organization using the **QueryExpression** and **SystemUser** classes, and the **RetrieveMultiple** method of the **IOrganizationService**.

6.2.3.6.1 Example Code

```
string organizationUrl = string.Empty;  
// use the discovery service to obtain the IOrganizationService endpoint for  
// the specified organization.  
Uri serviceUri = new Uri("https://auth.consolidatedmessenger.com/XRMServices/2011/Discovery.svc");  
  
IServiceConfiguration<IDiscoveryService> serviceConfig  
    = ServiceConfigurationFactory.CreateConfiguration<IDiscoveryService>(serviceUri);  
  
// create a client credential for authorization via IFD.  
ClientCredentials credentials = new ClientCredentials();  
credentials.UserName.Password = "Pass3word";  
credentials.UserName.UserName = "contoso\\crmprovacct";  
// Connect to the Discovery service.  
// The using statement assures that the service proxy will be properly disposed.  
using (DiscoveryServiceProxy proxy = new DiscoveryServiceProxy(serviceConfig, credentials))  
{  
    // Retrieve details about a single organization discoverable via the  
    // Discovery service.  
    RetrieveOrganizationRequest request = new RetrieveOrganizationRequest  
    {  
        UniqueName = "AlpineSkiHouse",  
        AccessType = EndpointAccessType.Default,  
        Release = OrganizationRelease.Current  
    };  
  
    // Execute the request  
    RetrieveOrganizationResponse response = (RetrieveOrganizationResponse)proxy.Execute(request);  
  
    // retrieve the IOrganizationService Uri  
    organizationUrl = response.Detail.Endpoints[EndpointType.OrganizationService];  
}  
  
// use URL obtained by the discovery service to connect to the IOrganizationService endpoint  
Uri organizationUri = new Uri(organizationUrl);  
  
IServiceConfiguration<IOrganizationService> orgServiceConfig  
    = ServiceConfigurationFactory.CreateConfiguration<IOrganizationService>(organizationUri);  
  
// The using statement assures that the service proxy will be properly disposed.  
using (OrganizationServiceProxy orgProxy = new OrganizationServiceProxy(orgServiceConfig, credentials))
```

```

{
    QueryExpression userQuery = new QueryExpression
    {
        EntityName = "systemuser",
        // retrieve all columns
        ColumnSet = new ColumnSet(true),

        Criteria =
        {
            Conditions =
            {
                new ConditionExpression("domainname",ConditionOperator.Equal,
                    "Contoso\\jimc_a1pineskihouse")
            }
        }
    };

    EntityCollection entities = orgProxy.RetrieveMultiple(userQuery);

    if (entities.Entities.Count > 0)
    {
        SystemUser user = entities[0].ToEntity<SystemUser>();

        //write out some key user properties
        Console.WriteLine("Id: {0}", user.Id);
        Console.WriteLine("DomainName: {0}", user.DomainName);
        Console.WriteLine("State: {0}", user.EntityState);
    }
}

```

6.2.3.6.2 Remarks

This sample demonstrates the use of early-binding with the strongly typed **SystemUser** class versus late-binding with the generic **Entity** class. The sample assumes that you have used the **CrmServiceUtil.exe** utility included in the SDK to generate code to interact with a strongly typed **SystemUser** class. For more information on generating and using early-bound entity classes see the [Use the Early Bound Entity Classes in Code](#) SDK topic.

6.2.3.7 Modify an Existing User Account

The following code example demonstrates how to retrieve a user account by **DomainName** from a tenant organization using the **QueryExpression** and **SystemUser** classes. The **Update** method of the **IOrganizationService** is then used to set properties on the user.

6.2.3.7.1 Example Code

```

string organizationUrl = string.Empty;
// use the discovery service to obtain the IOrganizationService endpoint for
// the specified organization.
Uri serviceUri = new Uri("https://auth.consolidatedmessenger.com/XRMServices/2011/Discovery.svc");

IServiceConfiguration<IDiscoveryService> serviceConfig
    = ServiceConfigurationFactory.CreateConfiguration<IDiscoveryService>(serviceUri);

// create a client credential for authorization via IFD.
ClientCredentials credentials = new ClientCredentials();

```

```
credentials.UserName.Password = "Pass3word";
credentials.UserName.UserName = "contoso\\crmprovacct";
// Connect to the Discovery service.
// The using statement assures that the service proxy will be properly disposed.
using (DiscoveryServiceProxy proxy = new DiscoveryServiceProxy(serviceConfig, credentials))
{
    // Retrieve details about a single organization discoverable via the
    // Discovery service.
    RetrieveOrganizationRequest request = new RetrieveOrganizationRequest
    {
        UniqueName = "AlpineSkiHouse",
        AccessType = EndpointAccessType.Default,
        Release = OrganizationRelease.Current
    };

    // Execute the request
    RetrieveOrganizationResponse response = (RetrieveOrganizationResponse)proxy.Execute(request);

    // retrieve the IOrganizationService Uri
    organizationUrl = response.Detail.Endpoints[EndpointType.OrganizationService];
}

// use URL obtained by the discovery service to connect to the IOrganizationService endpoint
Uri organizationUri = new Uri(organizationUrl);

IServiceConfiguration<IOrganizationService> orgServiceConfig
    = ServiceConfigurationFactory.CreateConfiguration<IOrganizationService>(organizationUri);

// The using statement assures that the service proxy will be properly disposed.
using (OrganizationServiceProxy orgProxy = new OrganizationServiceProxy(orgServiceConfig, credentials))
{
    // required for proper operation of the strongly typed classes
    orgProxy.EnableProxyTypes();

    // create a query to find the user
    QueryExpression userQuery = new QueryExpression
    {
        EntityName = "systemuser",
        // retrieve all columns
        ColumnSet = new ColumnSet(true),

        Criteria =
        {
            Conditions =
            {
                new ConditionExpression("domainname", ConditionOperator.Equal,
                    "Contoso\\jimc_alpineskihouse")
            }
        }
    };

    // retrieve the user
    EntityCollection entities = orgProxy.RetrieveMultiple(userQuery);

    // update the user
    if (entities.Entities.Count > 0)
    {
        SystemUser user = entities[0].ToEntity<SystemUser>();
    }
}
```

```

user.Address1_City = "Seattle";
user.Address1_Country = "USA";
user.Address1_PostalCode = "98115";
user.Address1_StateOrProvince = "WA";

orgProxy.Update(user);
}
}

```

6.2.3.7.2 *Remarks*

For more information on generating and using early-bound entity classes see the [Use the Early Bound Entity Classes in Code](#) SDK topic.

6.2.3.8 Determine if an Existing User Account belongs to a Security Role

The following code example demonstrates how to retrieve a user account by **DomainName** from a tenant organization using the **QueryExpression** and **SystemUser** classes, and the **RetrieveMultiple** method of the **IOrganizationService**. In addition, a **Role** is retrieved and then the **SystemUser** is tested for membership in the role using the **QueryExpression** with **LinkEntity** instances.

6.2.3.8.1 *Example Code*

```

bool result = false;
string organizationUrl = string.Empty;
// use the discovery service to obtain the IOrganizationService endpoint for
// the specified organization.
Uri serviceUri = new Uri("https://auth.consolidatedmessenger.com/XRMServices/2011/Discovery.svc");

IServiceConfiguration<IDiscoveryService> serviceConfig
    = ServiceConfigurationFactory.CreateConfiguration<IDiscoveryService>(serviceUri);

// create a client credential for authorization via IFD.
ClientCredentials credentials = new ClientCredentials();
credentials.UserName.Password = "Pass3word";
credentials.UserName.UserName = "contoso\\crmprovacct";
// Connect to the Discovery service.
// The using statement assures that the service proxy will be properly disposed.
using (DiscoveryServiceProxy proxy = new DiscoveryServiceProxy(serviceConfig, credentials))
{
    // Retrieve details about a single organization discoverable via the
    // Discovery service.
    RetrieveOrganizationRequest request = new RetrieveOrganizationRequest
    {
        UniqueName = "AlpineSkiHouse",
        AccessType = EndpointAccessType.Default,
        Release = OrganizationRelease.Current
    };

    // Execute the request
    RetrieveOrganizationResponse response = (RetrieveOrganizationResponse)proxy.Execute(request);

    // retrieve the IOrganizationService Uri
    organizationUrl = response.Detail.Endpoints[EndpointType.OrganizationService];
}

// use URL obtained by the discovery service to connect to the IOrganizationService endpoint
Uri organizationUri = new Uri(organizationUrl);

```

```
IServiceConfiguration<IOrganizationService> orgServiceConfig
    = ServiceConfigurationFactory.CreateConfiguration<IOrganizationService>(organizationUri);

// The using statement assures that the service proxy will be properly disposed.
using (OrganizationServiceProxy orgProxy = new OrganizationServiceProxy(orgServiceConfig, credentials))
{
    // required for proper operation of the strongly typed classes
    orgProxy.EnableProxyTypes();

    // create a query to find the user
    QueryExpression userQuery = new QueryExpression
    {
        EntityName = "systemuser",

        ColumnSet = new ColumnSet("systemuserid"),

        Criteria =
        {
            Conditions =
            {
                new ConditionExpression("domainname",ConditionOperator.Equal,
                    "Contoso\\jimc_alpineskihouse")
            }
        }
    };

    // retrieve the user
    EntityCollection entities = orgProxy.RetrieveMultiple(userQuery);

    // determine if the user is in the role
    if (entities.Entities.Count > 0)
    {
        SystemUser user = entities.Entities[0].ToEntity<SystemUser>();
        // find the role
        QueryExpression query = new QueryExpression
        {
            EntityName = "role",
            ColumnSet = new ColumnSet("roleid"),
            Criteria = new FilterExpression
            {
                Conditions =
                {
                    new ConditionExpression
                    {
                        AttributeName = "name",
                        Operator = ConditionOperator.Equal,
                        Values = {"salesperson"}
                    }
                }
            }
        };

        // get the role
        Role salesRole = orgProxy.RetrieveMultiple(query).Entities.Cast<Role>().FirstOrDefault();
    }
}
```

```

if (salesRole != null)
{
    // Establish a SystemUser link for a query.
    LinkEntity systemUserLink = new LinkEntity()
    {
        LinkFromEntityName = "systemuserroles",
        LinkFromAttributeName = "systemuserid",
        LinkToEntityName = "systemuser",
        LinkToAttributeName = "systemuserid",
        LinkCriteria =
        {
            Conditions =
            {
                new ConditionExpression(
                    "systemuserid", ConditionOperator.Equal, user.Id)
            }
        }
    };

    // Build the query.
    QueryExpression linkQuery = new QueryExpression()
    {
        EntityName = "role",
        ColumnSet = new ColumnSet("roleid"),
        LinkEntities =
        {
            new LinkEntity()
            {
                LinkFromEntityName = "role",
                LinkFromAttributeName = "roleid",
                LinkToEntityName = "systemuserroles",
                LinkToAttributeName = "roleid",
                LinkEntities = {systemUserLink}
            }
        },
        Criteria =
        {
            Conditions =
            {
                new ConditionExpression("roleid", ConditionOperator.Equal, salesRole.Id)
            }
        }
    };

    // Retrieve matching roles.
    EntityCollection matchEntities = orgProxy.RetrieveMultiple(linkQuery);

    // if an entity is returned then the user is a member
    // of the role
    result = (matchEntities.Entities.Count > 0);
}
}
return result;

```

6.2.3.8.2 Remarks

For more information on generating and using early-bound entity classes see the [Use the Early Bound Entity Classes in Code](#) SDK topic.

6.2.3.9 Add an Existing User Account to a Role

The following code example demonstrates how to retrieve a user account by **DomainName** from a tenant organization using the **QueryExpression** and **SystemUser** classes, and the **RetrieveMultiple** method of the **IOrganizationService**. In addition, a **Role** is retrieved and then the **SystemUser** is associated with it using the **Associate** method of the **IOrganizationService**.

6.2.3.9.1 Example Code

```
string organizationUrl = string.Empty;
// use the discovery service to obtain the IOrganizationService endpoint for
// the specified organization.
Uri serviceUri = new Uri("https://auth.consolidatedmessenger.com/XRMServices/2011/Discovery.svc");

IServiceConfiguration<IDiscoveryService> serviceConfig
    = ServiceConfigurationFactory.CreateConfiguration<IDiscoveryService>(serviceUri);

// create a client credential for authorization via IFD.
ClientCredentials credentials = new ClientCredentials();
credentials.UserName.Password = "Pass3word";
credentials.UserName.UserName = "contoso\\crmprovacct";
// Connect to the Discovery service.
// The using statement assures that the service proxy will be properly disposed.
using (DiscoveryServiceProxy proxy = new DiscoveryServiceProxy(serviceConfig, credentials))
{
    // Retrieve details about a single organization discoverable via the
    // Discovery service.
    RetrieveOrganizationRequest request = new RetrieveOrganizationRequest
    {
        UniqueName = "AlpineSkiHouse",
        AccessType = EndpointAccessType.Default,
        Release = OrganizationRelease.Current
    };

    // Execute the request
    RetrieveOrganizationResponse response = (RetrieveOrganizationResponse)proxy.Execute(request);

    // retrieve the IOrganizationService Uri
    organizationUrl = response.Detail.Endpoints[EndpointType.OrganizationService];
}

// use URL obtained by the discovery service to connect to the IOrganizationService endpoint
Uri organizationUri = new Uri(organizationUrl);

IServiceConfiguration<IOrganizationService> orgServiceConfig
    = ServiceConfigurationFactory.CreateConfiguration<IOrganizationService>(organizationUri);

// The using statement assures that the service proxy will be properly disposed.
using (OrganizationServiceProxy orgProxy = new OrganizationServiceProxy(orgServiceConfig, credentials))
{
    // required for proper operation of the strongly typed classes
    orgProxy.EnableProxyTypes();

    // create a query to find the user
```

```

QueryExpression userQuery = new QueryExpression
{
    EntityName = "systemuser",

    ColumnSet = new ColumnSet("systemuserid"),

    Criteria =
    {
        Conditions =
        {
            new ConditionExpression("domainname",ConditionOperator.Equal,
                "Contoso\\jimc_alpineskihouse")
        }
    }
};

// retrieve the user
EntityCollection entities = orgProxy.RetrieveMultiple(userQuery);

// add the user to a role
if (entities.Entities.Count > 0)
{
    SystemUser user = entities[0].ToEntity<SystemUser>();

    // find the role
    QueryExpression query = new QueryExpression
    {
        EntityName = "role",
        ColumnSet = new ColumnSet("roleid"),
        Criteria = new FilterExpression
        {
            Conditions =
            {
                new ConditionExpression
                {
                    AttributeName = "name",
                    Operator = ConditionOperator.Equal,
                    Values = {"salesperson"}
                }
            }
        }
    };

    // get the role
    Role salesRole = orgProxy.RetrieveMultiple(query).Entities.Cast<Role>().FirstOrDefault();

    // associate the user with the role
    if (salesRole != null)
    {
        orgProxy.Associate(
            "systemuser",
            user.Id,
            new Relationship("systemuserroles_association"),
            new EntityReferenceCollection() { new EntityReference("role", salesRole.Id) });
    }
}

```

```
}  
}
```

6.2.3.9.2 *Remarks*

For more information on generating and using early-bound entity classes, see the [Use the Early Bound Entity Classes in Code](#) SDK topic.

6.2.3.10 **Remove an Existing User Account from a Role**

The following code example demonstrates how to retrieve a user account by **DomainName** from a tenant organization using the **QueryExpression** and **SystemUser** classes, and the **RetrieveMultiple** method of the **IOrganizationService**. In addition, a **Role** is retrieved and then the **SystemUser** is disassociated with it using the **Disassociate** method of the **IOrganizationService**.

6.2.3.10.1 *Example Code*

```
string organizationUrl = string.Empty;  
// use the discovery service to obtain the IOrganizationService endpoint for  
// the specified organization.  
Uri serviceUri = new Uri("https://auth.consolidatedmessenger.com/XRMServices/2011/Discovery.svc");  
  
IServiceConfiguration<IDiscoveryService> serviceConfig  
    = ServiceConfigurationFactory.CreateConfiguration<IDiscoveryService>(serviceUri);  
  
// create a client credential for authorization via IFD.  
ClientCredentials credentials = new ClientCredentials();  
credentials.UserName.Password = "Pass3word";  
credentials.UserName.UserName = "contoso\\crmprovacct";  
// Connect to the Discovery service.  
// The using statement assures that the service proxy will be properly disposed.  
using (DiscoveryServiceProxy proxy = new DiscoveryServiceProxy(serviceConfig, credentials))  
{  
    // Retrieve details about a single organization discoverable via the  
    // Discovery service.  
    RetrieveOrganizationRequest request = new RetrieveOrganizationRequest  
    {  
        UniqueName = "AlpineSkiHouse",  
        AccessType = EndpointAccessType.Default,  
        Release = OrganizationRelease.Current  
    };  
  
    // Execute the request  
    RetrieveOrganizationResponse response = (RetrieveOrganizationResponse)proxy.Execute(request);  
  
    // retrieve the IOrganizationService Uri  
    organizationUrl = response.Detail.Endpoints[EndpointType.OrganizationService];  
}  
  
// use URL obtained by the discovery service to connect to the IOrganizationService endpoint  
Uri organizationUri = new Uri(organizationUrl);  
  
IServiceConfiguration<IOrganizationService> orgServiceConfig  
    = ServiceConfigurationFactory.CreateConfiguration<IOrganizationService>(organizationUri);  
  
// The using statement assures that the service proxy will be properly disposed.  
using (OrganizationServiceProxy orgProxy = new OrganizationServiceProxy(orgServiceConfig, credentials))  
{  
    // required for proper operation of the strongly typed classes
```

```

orgProxy.EnableProxyTypes());

// create a query to find the user
QueryExpression userQuery = new QueryExpression
{
    EntityName = "systemuser",

    ColumnSet = new ColumnSet("systemuserid"),

    Criteria =
    {
        Conditions =
    {
        new ConditionExpression("domainname",ConditionOperator.Equal,
            "Contoso\\jimc_alpineskihouse")
    }
    }
};

// retrieve the user
EntityCollection entities = orgProxy.RetrieveMultiple(userQuery);

// add the user to a role
if (entities.Entities.Count > 0)
{
    SystemUser user = entities[0].ToEntity<SystemUser>();
    // find the role
    QueryExpression query = new QueryExpression
    {
        EntityName = "role",
        ColumnSet = new ColumnSet("roleid"),
        Criteria = new FilterExpression
        {
            Conditions =
    {
        new ConditionExpression
        {
            AttributeName = "name",
            Operator = ConditionOperator.Equal,
            Values = {"salesperson"}
        }
    }
        }
    };

    // get the role
    Role salesRole = orgProxy.RetrieveMultiple(query).Entities.Cast<Role>().FirstOrDefault();

    // associate the user with the role

    if (salesRole != null)
    {
        orgProxy.Disassociate(
            "systemuser",
            user.Id,
            new Relationship("systemuserroles_association"),
            new EntityReferenceCollection() { new EntityReference("role", salesRole.Id) });
    }
}

```

```
}  
}  
}
```

6.2.3.10.2 Remarks

For more information on generating and using early-bound entity classes see the [Use the Early Bound Entity Classes in Code](#) SDK topic.

6.2.3.11 Retrieve a Listing of Security Roles from a Tenant Organization

The following code example demonstrates how to retrieve a list of all the roles from a tenant organization using the **QueryExpression** and **Role** classes and the **RetrieveMultiple** method of the **IOrganizationService**.

6.2.3.11.1 Example Code

```
string organizationUrl = string.Empty;  
// use the discovery service to obtain the IOrganizationService endpoint for  
// the specified organization.  
Uri serviceUri = new Uri("https://auth.consolidatedmessenger.com/XRMServices/2011/Discovery.svc");  
  
IServiceConfiguration<IDiscoveryService> serviceConfig  
    = ServiceConfigurationFactory.CreateConfiguration<IDiscoveryService>(serviceUri);  
  
// create a client credential for authorization via IFD.  
ClientCredentials credentials = new ClientCredentials();  
credentials.UserName.Password = "Pass3word";  
credentials.UserName.UserName = "contoso\\crmprovacct";  
// Connect to the Discovery service.  
// The using statement assures that the service proxy will be properly disposed.  
using (DiscoveryServiceProxy proxy = new DiscoveryServiceProxy(serviceConfig, credentials))  
{  
    // Retrieve details about a single organization discoverable via the  
    // Discovery service.  
    RetrieveOrganizationRequest request = new RetrieveOrganizationRequest  
    {  
        UniqueName = "AlpineSkiHouse",  
        AccessType = EndpointAccessType.Default,  
        Release = OrganizationRelease.Current  
    };  
  
    // Execute the request  
    RetrieveOrganizationResponse response = (RetrieveOrganizationResponse)proxy.Execute(request);  
  
    // retrieve the IOrganizationService Uri  
    organizationUrl = response.Detail.Endpoints[EndpointType.OrganizationService];  
}  
  
// use URL obtained by the discovery service to connect to the IOrganizationService endpoint  
Uri organizationUri = new Uri(organizationUrl);  
  
IServiceConfiguration<IOrganizationService> orgServiceConfig  
    = ServiceConfigurationFactory.CreateConfiguration<IOrganizationService>(organizationUri);  
  
// The using statement assures that the service proxy will be properly disposed.  
using (OrganizationServiceProxy orgProxy = new OrganizationServiceProxy(orgServiceConfig, credentials))
```

```

{
    //required for correct functioning of generated strong types
    orgProxy.EnableProxyTypes();

    QueryExpression query = new QueryExpression
    {
        EntityName = Role.EntityLogicalName,
        ColumnSet = new ColumnSet("name", "roleid")
    };

    EntityCollection entities = orgProxy.RetrieveMultiple(query);
    // write the name and ID of each role to the console.
    foreach (Entity item in entities.Entities)
    {
        Role role = item.ToEntity<Role>();
        Console.WriteLine("Name: {0}. Id: {1}", role.Name, role.Id);
    }
}

```

6.2.3.11.2 Remarks

For more information on generating and using early-bound entity classes, see the [Use the Early Bound Entity Classes in Code](#) SDK topic.

6.2.3.12 Disable/Enable an Existing User Account

The following code example demonstrates how to retrieve a user account by **DomainName** from a tenant organization using the **QueryExpression** and **SystemUser** classes, and the **RetrieveMultiple** method of the **IOrganizationService**. A **SetStateRequest** is then used to place the user in a disabled state using the **Execute** method of the **IOrganizationService**.

6.2.3.12.1 Example Code

```

string organizationUrl = string.Empty;
// use the discovery service to obtain the IOrganizationService endpoint for
// the specified organization.
Uri serviceUri = new Uri("http://hex2010-crm01:5555/XRMServices/2011/Discovery.svc");

IServiceConfiguration<IDiscoveryService> serviceConfig
    = ServiceConfigurationFactory.CreateConfiguration<IDiscoveryService>(serviceUri);

// create a client credential for authorization via IFD.
ClientCredentials credentials = new ClientCredentials();
credentials.UserName.Password = "Pass3word";
credentials.UserName.UserName = "contoso\\crmprovacct";
// Connect to the Discovery service.
// The using statement assures that the service proxy will be properly disposed.
using (DiscoveryServiceProxy proxy = new DiscoveryServiceProxy(serviceConfig, credentials))
{
    // Retrieve details about a single organization discoverable via the
    // Discovery service.
    RetrieveOrganizationRequest request = new RetrieveOrganizationRequest
    {
        UniqueName = "AlpineSkiHouse",
        AccessType = EndpointAccessType.Default,
        Release = OrganizationRelease.Current
    };
};

```

```
// Execute the request
RetrieveOrganizationResponse response = (RetrieveOrganizationResponse)proxy.Execute(request);

// retrieve the IOrganizationService Uri
organizationUrl = response.Detail.Endpoints[EndpointType.OrganizationService];
}

// use URL obtained by the discovery service to connect to the IOrganizationService endpoint
Uri organizationUri = new Uri(organizationUrl);

IServiceConfiguration<IOrganizationService> orgServiceConfig
    = ServiceConfigurationFactory.CreateConfiguration<IOrganizationService>(organizationUri);

// The using statement assures that the service proxy will be properly disposed.
using (OrganizationServiceProxy orgProxy = new OrganizationServiceProxy(orgServiceConfig, credentials))
{
    // required for generated types to work properly
    orgProxy.EnableProxyTypes();

    QueryExpression userQuery = new QueryExpression
    {
        EntityName = "systemuser",
        ColumnSet = new ColumnSet("systemuserid"),

        Criteria =
        {
            Conditions =
            {
                new ConditionExpression("domainname",ConditionOperator.Equal,
                    "Contoso\\user1_AlpineBikeS608")
            }
        }
    };

    // retrieve the user
    SystemUser user = orgProxy.RetrieveMultiple(userQuery).Entities.Cast<SystemUser>().FirstOrDefault();

    // if the user is not null then disable it.
    if (user != null)
    {
        SetStateRequest request = new SetStateRequest();
        request.EntityMoniker = user.ToEntityReference();
        // sets the user to disabled
        request.State = new OptionSetValue(1);
        // required by request but always valued at -1 in this context
        request.Status = new OptionSetValue(-1);

        orgProxy.Execute(request);
    }
}
```

6.2.3.12.2 Remarks

In Microsoft Dynamics CRM it is not possible to delete a **SystemUser** instance. Instead you can disable a user so that they can no longer use the system. The user can then be re-enabled using a variation of the code example above:

```
SetStateRequest request = new SetStateRequest();
request.EntityMoniker = user.ToEntityReference();
// sets the user to enabled
request.State = new OptionSetValue(0);
// required by request but always valued at -1 in this context
request.Status = new OptionSetValue(-1);

orgProxy.Execute(request);
```

Note

Once you have created a user with a specific **DomainName** in an organization you cannot have another user with the same **DomainName**, even if the user is disabled.

For more information on generating and using early-bound entity classes, see the [Use the Early Bound Entity Classes in Code](#) SDK topic.