9.1. AD LDS

Active Directory Lightweight Directory Services (AD LDS), formerly known as Active Directory Application Mode (ADAM), is an LDAP directory service that you can use to create a directory store (database) for use by directory-enabled applications. AD LDS is very similar to Active Directory Domain Services (AD DS).

- AD LDS is based on the X.500 directory standard.
- AD LDS uses the same database engine as AD DS. The database is stored in a file called Adamntds.nit.
- DNS is not required to implement an AD LDS database.
- AD LDS does not include many AD DS features such as Group Policy, Global Catalog, or Kerberos authentication.
- AD LDS does not require the deployment of domains or domain controllers.
- You can install an AD LDS on domain members or non-domain members with the following operating systems:
    - Microsoft Windows Server 2008/2012/2016 with a full feature set.
    - Any edition of Windows Server 2003 with a full feature set.
- Like AD DS, AD LDS uses a schema that defines the objects and attributes within the database. You can customize the schema to define your own objects or attributes.
- AD LDS is compatible and consistent with traditional Active Directory in that it manages and accesses application data using the same Application Programming Interfaces (API) as Active Directory, namely:
    - Active Directory API
    - Active Directory Service Interfaces
    - Lightweight Data Access Protocol
    - System.DirectoryServices
- Active Directory and AD LDS can exist and operate concurrently in the same environment.
    - AD LDS can be created for specific applications without worrying about any dependencies Active Directory may otherwise require of the application.
    - If desired, you can synchronize data from Active Directory into an AD LDS instance.
    - AD LDS can use AD DS for the authentication of Windows security principals.

An *instance* (also called a *service instance*) is a single running copy of AD LDS.

- Multiple instances of AD LDS (multiple distinct databases) can run simultaneously on the same computer.
- You can selectively start, stop, and restart instances running on a computer.
- You can replicate an instance between servers, meaning that multiple servers have the same copy of the database. Create a *configuration set* to group AD LDS instances that hold copies of the same directory partition or partitions.
- For instances within a configuration set, multimaster replication copies changes made from any instance to all other instances in the set. Replication does not occur outside of the configuration set.
- Use configuration sets to provide for fault tolerance, load balancing, and geographic placement of an AD LDS instance.
- You can configure the configuration set to exclude some partitions from replication. A single AD LDS instance can replicate all, or any subset of, the application directory partitions in its configuration set.

© Sergey Gorokhod
MCT/MCSE/MCITP/MCTS/MCSA/CCSE/CCSA/CCNA/A+®
E-mail: sergey@infosec.co.il
Mob: (+972) 526848757

9.2. AD LDS Configuration

Configuring AD LDS involves installing the AD LDS role through the Add Roles feature in Server Manager, and then adding one or more instances to the server. To configure a new instance, you will need to:

- Specify the instance name. The instance name must be unique on the server.
- Choose the LDAP and SSL port numbers used to connect to the instance.
    - If the server is a domain controller, or will be used as one, do not use port 389 for LDAP or 636 for SSL.
    - If AD DS is detected on the system, the configuration wizard automatically selects available ports above 50000.
- For new instances, specify whether or not to create an application directory partition.
    - If the application creates its own partition, do not create one during the install.
    - To create the partition, specify the distinguished name for the partition (for example: CN=Partition1,DC=westsim,DC=local).
- Select the service user account used by AD LDS and the user account who has administrator privileges to the instance.
- For new instances, select LDIF files to import. These files have predefined schema and object information.
- For a replica instance, select the user account with permissions to the source configuration set.
- For a replica instance, specify the source server (using its DNS or NetBIOS name), and select the partitions you want to replicate.

Use the following tools to manage AD LDS instances:

Tool	Description
Active Directory Lightweight Directory Services console	Use the Active Directory Lightweight Directory Services console to create new instances, start and stop instances, and launch common AD LDS management tools.

Adaminstall.exe	Use **Adaminstall.exe** to create an instance using an unattended file. See the following for two examples of answer files used with **Adaminstall.exe**: **Install a New Instance** The following answer file creates a new instance with an application partition and imports two LDIF files: **[ADAMInstall]** **InstallType=Unique** **InstanceName=MyFirstInstance** **LocalLDAPPortToListenOn=389** **LocalSSLPortToListenOn=636** **NewApplicationPartitionToCreate="CN=App1,DC=westsim,DC=us"** **DataFilesPath=C:\Program Files\Microsoft ADAM\instance1\data** **LogFilesPath=C:\Program Files\Microsoft ADAM\instance1\data** **ImportLDIFFiles="ms-inetorgperson.ldf" "ms-user.ldf"** ***Note**: In this example, the instance is being installed on a server that is not a domain controller. If the server was an AD DS domain controller, you would need to use different port numbers.* **Install a Replica Instance** The following answer file creates a new instance that is a replica of an existing instance. The answer file copies an existing application directory partition. **[ADAMInstall]** **InstallType=Replica** **InstanceName=MySecondInstance** **LocalLDAPPortToListenOn=50000** **LocalSSLPortToListenOn=50001** **SourceServer=Srv1.westsim.local** **SourceLDAPPort=389** **ApplicationPartitionsToReplicate="CN=App1,DC=westsim,DC=us"** **DataFilesPath=C:\Program Files\Microsoft ADAM\instance2\data** **LogFilesPath=C:\Program Files\Microsoft ADAM\instance2\data**

	Once your answer file has been created, type the following into a command prompt. Use quotation marks if the file name includes spaces. **c:\ADAM\adaminstall.exe /answer:C:\unattend.txt**
ADSI Edit	Use ADSI Edit to query, view, and edit objects and attributes in the directory. You must first connect and bind to the instance to use ADSI Edit to administer an AD LDS instance. You can perform the following tasks in ADSI Edit: • Create and AD LDS group or user. To do this, you must first import the optional user classes that are provided with AD LDS into the AD LDS schema. User classes are located in importable .ldf files found in the directory %windir%\ADAM on the computer where AD LDS is installed. • Add or remove members in an AD LDS group. • Disable or enable AD LDS user accounts to control whether or not a user can bind to the AD LDS directory. • Create a replication schedule for a configuration set.
Ldp.exe	Use **Ldp.exe** to perform LDAP operations against the directory. For example, you can search, modify, add, and delete objects and attributes.
Ldifde	Use **Ldifde.exe** to import, export, modify, and delete objects using LDAP Data Interchange Format (LDIF) files.
Active Directory Schema snap-in	Use the Active Directory Schema snap-in to view and manage AD LDS schema objects. By default, the AD schema snap-in must be registered and added to the console before you can make changes to the role. To register the Dynamic Link Library (DDL) for the AD Schema snap-in, type ***regsvr32 schmmgmt.dll*** at the command prompt.

Active Directory Sites and Services snap-in	Use the Active Directory Sites and Services snap-in to connect to your AD LDS instance to administer the replication of directory data among all sites in an AD LDS configuration set. This snap-in can be used by accessing the Active Directory Sites and Services option from Administrative Tools on the Start menu.
ADSchemaAnalyzer	Use ADSchemaAnalyzer to: • Quickly copy the schema from AD DS and import it into AD LDS. • Help migrate the AD DS schema to AD LDS, from one AD LDS instance to another, or from any LDAP-compliant directory to an AD LDS instance. • Load a target (source) schema, mark the elements that you want to migrate, and then export them to the base AD LDS schema. You can build your own schema using ADSchemaAnalyzer to make custom LDIF files if you are building a custom application. If you need to copy data from an existing AD DS deployment into the new AD LDS instance, it is difficult to build the schema file.
Dsacls	Use **Dsacls** to view, grant, or deny permissions on an object-by-object basis. Like AD DS permissions, objects in the directory inherit the permissions specified by the ACL that is located in the top-level object of each directory object partition, as well as obtaining permissions assigned directly to specific objects.

© Sergey Gorokhod
MCT/MCSE/MCITP/MCTS/MCSA/CCSE/CCSA/CCNA/A+®
E-mail: sergey@infosec.co.il
Mob: (+972) 526848757

9.3. AD LDS Management

You should be aware of the following methods of AD LDS instance configuration:

Method	Description
Move an instance	An AD LDS replica instance can be moved into a site object by using the Active Directory Sites and Services. Membership in the Administrators group or equivalent is required to complete this procedure. 1. Open Active Directory Sites and Services from the Administrative tools in the Start menu. 2. Click on **Change Domain Controller**, then specify the name and the port number of the server that holds the AD LDS instances in the configuration set for which you want to create site objects. 3. Open the **Servers** container by double-clicking the **Sites** container, then double-clicking the site that contains the AD LDS instance that you want to move, then double-clicking the **Servers** container. 4. In the **Servers** container, right-click the AD LDS instance that you want to move, then click **Move**. 5. In the **Move Server** dialog box, select the site to which you want to move the AD LDS instance, then click **OK**.
Import data into an instance	Data can be imported into an AD LDS instance in the following ways: • Using the **Importing LDIF Files** page in the AD LDS Setup Wizard during setup of the instance. • Manually, by using the **Ldifde** command any time after creation of the instance. Prior to importing the data, use the **Ldifde** tool to export the data from an existing LDAP directory.

Create a replication schedule	A replication schedule for an AD LDS instance can be created using ADSI edit. • Scheduling replication is optional; the AD LDS replication schedule is set to the **Once per Hour** option by default. • Because intra-site AD LDS replication uses update notifications to replicate data, intra-site replication is only affected by a replication frequency schedule when no update notifications occur in the specified time frame.
Synchronize data	Use the ***Adamsync /sync*** command to synchronize data from an AD DS forest to the configuration set of an AD LDS instance. • The AD LDS instance must have been configured by importing the **MS-AdamSyncMetadata.LDF** file. • You must run the command each time you want to synchronize data.

AD LDS must verify the user's credentials or bind users into the directory through successful authentication before they can request directory data. Binding to an AD LDS instance takes place in the following ways:

- The user account resides directly in AD LDS. You can bind as an AD LDS security principal using **Ldp.exe** in Server Manager.
- The user account resides on a local computer or in an Active Directory Domain Services domain. You can bind as a Windows Security principal using the ADSI Edit snap-in.
- The user is bound through an AD LDS proxy object using redirection. This allows AD LDS to accept and process bind requests to an AD LDS proxy object that has the Security Identifier (SID) from an AD DS security principal as one of its attributes. A user who binds to an AD LDS instance through a proxy object receives membership in the Users group on each naming context that is held by the AD LDS instance.
  Bind redirection provides the following benefits:
  - Binding provides AD DS users with access to both AD LDS data and AD DS data using AD DS domain credentials for Single Sign-On (SSO).

    - AD LDS proxy objects can be used to store user data that is specific to a particular application in AD LDS, while using AD DS to store more widely used directory data.
    - Unlike other types of binding, bind redirection enables a user to bind to AD LDS by means of a simple bind while still using AD DS credentials.

You should know the following about configuring security principles and binding:

- Passwords can be set for security principals either by using ADSI Edit or by using **Ldp.exe** over an encrypted, non-SSL connection.
- You can set or modify a password for an AD LDS security principal using the ADSI Edit snap-in.
- AD LDS allows the use of Windows security principals for authentication and access control.
- Windows users can be members of AD LDS groups.
- A Windows user binding to an AD LDS instance receives membership only in the AD LDS groups to which that user has been explicitly added as a member.

© Sergey Gorokhod
MCT/MCSE/MCITP/MCTS/MCSA/CCSE/CCSA/CCNA/A+®
E-mail: sergey@infosec.co.il
Mob: (+972) 526848757

9.4. AD FS

Active Directory Federation Services (AD FS) is a secure solution that allows browser-based clients to access one or more protected Internet-facing applications regardless of whether the account and application are located in the same network or organization. AD FS:

- Secures trust relationships that makes a user's digital identity and access rights available to trusted partners, thus making the need for secondary credential requests unnecessary.
- Allows organizations to provide access to resources that are offered by trusted partners across the Internet.
- Retrieves user attributes from Active Directory.
- Uses Windows Integrated Authentication.
- Uses strong authentication technologies such as Kerberos, X.509 digital certificates, and smart cards.
- Uses Lightweight Directory Access Protocol (LDAP) binding to authenticate users when it is employed with AD LDS.
- Enables Single Sign-On (SSO), where an end user can access resources within multiple systems or organizations without having to repeatedly supply their logon credentials.

AD FS is designed for organizations that have:

- At least one implementation of either AD DS or AD LDS.
- An environment in which multiple different operating systems are deployed.
- Computers that are domain-joined.
- One or more Web-based applications implemented.
- Computers that are connected to the Internet.

To understand how AD FS works, you need to be familiar with the following terms:

Term	Definition
Claim	A *claim* is a statement made about a digital identity. A claim includes: • Identity information, such as the user name • Group memberships • Capabilities, rights, and privileges A *claims-aware* application is an application that performs authorization for objects based on claims.
Security token	A *security token* is a digitally-signed object that contains claims for a given user. Following successful authentication, the security token is issued. A Windows NT *token-based* application is an application that uses Windows NT tokens (not claims) to authenticate users.
Security Token Service (STS)	A Security Token Service (STS) is a Web service that issues security tokens based on evidence provided by trusted entities.
Federation	A *federation* is a set of domains or realms that have been configured to trust each other. A *federated application* is another name for a claims-aware application.
Federation server	A *federation server* is a server running the Active Directory Federation Service role. Federation servers: • Host a security token service that takes user credentials and issues security tokens according to the user credentials. • Collect claims for users by examining user attributes stored in AD DS or AD LDS.
Federation trust	A *federation trust* is a one-way, non-transitive relationship between two organizations. Each trust has two members: • The *account* partner is the member that maintains the user accounts and is trusted to provide security tokens. The account partner is responsible for:

	- Storing user accounts in either an Active Directory store or an AD LDS store. - Collecting and authenticating user's credentials. - Building up claims for users. - Packaging claims into security tokens. - Issuing security tokens to users in the account partner realm. - The *resource* partner is the member that holds resources that need to be accessed by users. The resource partner is responsible for: - Validating the security tokens issued by the account partner. - Consuming (reading or interpreting) the claims that are packaged in security tokens to make authentication decisions regarding the level of resource access allowed. *Claim mapping* is the process of examining an incoming claim and filtering it to extract appropriate authorizations for a user. - Issuing security tokens for the applications. - Issuing cookies to the user accounts. These cookies allow the user to maintain Single Sign-On (SSO) login status when the user accesses multiple applications at the resource partner. The account partner can be compared to an Active Directory domain that contains user accounts that need access to resources that are located in another forest. The resource partner is like the Active Directory domain that holds the necessary resources.
Trust policy	A *trust policy* is the configured parameters that identifies trusted partners, certificates, account stores, claims, and the various properties of the entities that are associated with the federation service.
AD FS-enabled Web server	An AD FS-enabled Web server is a Web server that is running the necessary services to allow it to host claims-aware applications.

9.5. AD FS Configuration

You should be aware of the following when implementing AD FS:

- You can only install Active Directory Federation Services (AD FS) on domain members.
- AD FS requires Windows Server 2003 R2 or higher. It runs on only the Enterprise or Datacenter versions. The AD FS Web Agent can be installed on Standard edition servers.
- AD FS requires IIS, ASP.NET 2.0, and the .NET Framework 2.0.
- AD FS requires access to user accounts in either AD DS or AD LDS. Domain controllers require Windows 2000 with Service Pack 4 or higher.
- During installation, you select one of the following role services to install:

Role Service	Description
Federation Service	Installing AD FS runs the Federation Service on the server. The Federation Service: o Functions as a security token service. o Collects and verifies user credentials against Active Directory or ADAM to provide security tokens. o Routes authentication requests from user accounts in other organizations or from Internet clients. o Determines which organization will authenticate a user. o Verifies that tokens have been correctly signed by the correct partner.
Federation Service Proxy	The Federation Service Proxy forwards requests to federation servers that is not accessible from the Internet. The Federation Service Proxy: o Uses WS-Federation Passive Requestor Profile (WS-FPRP) protocols: ▪ To collect user credential information from browser clients.

	▪ To relay requests from Web applications to the Federation Service. o Stores HTTP authentication, account partner, and sign-out cookies for clients to facilitate SSO. ***Note**: It is not necessary to deploy a separate server to act as a federation server proxy; federation servers perform this role automatically. However, if you use a proxy, it cannot be installed on a server running the Federation Service.*
Claims-aware Agent	The claims-aware agent is installed on Web servers that host claims-aware applications (Microsoft ASP.NET applications). The agent allows the claims-aware applications to query AD FS security tokens to make authorization decisions and personalize applications.
Windows NT Token-based Agent	The Windows token-based agent is installed on a Web server that hosts Windows NT token-based applications. It converts AD FS security tokens to impersonation-level Windows NT access tokens so they can be used by the token-based application.

- You cannot install the Federation Service and the Federation Service Proxy on the same computer. You can, however, install one or both Web agents together with either the Federation Service or the Federation Service Proxy.
- AD FS requires that each server have a certificate that is used for SSL communications. For implementations within your organization, you can issue a certificate from your own CA. When used with external organizations, obtain a server certificate from a third party.
- A federation server requires a certificate for issuing tokens; the token is digitally signed to verify that the token was issued by the federation server. This can be the same certificate used for SSL.
- When installing the Windows NT Token-based Agent, you must specify the Federation Service Proxy or the Federation Service that the agent communicates with.
- During installation, you can select an existing trust policy or create a new one.
- Installing an AD FS role also installs IIS if it does not yet exist.

Following installation, use the Active Directory Federation Services console to manage AD FS. For the Federation Service, you configure the following elements of the trust policy:

Component	Description
Organization Claims	Organizational claims identify the types of information about users that is understood and processed by AD FS. You can identify the following types of claims: • An *identity* claim is used to identify the user. Information that can be used for identity is the UPN name (such as **user1@mydomain.com**), an e-mail address, or a common name. • A *group* claim describes membership in a group or a role. For example, you list the names of groups to which the user belongs. • A *custom* claim is a custom attribute, such as an employee ID number, that you can define.
Account Stores	The account store identifies the database that holds the user accounts (either AD DS or AD LDS). When you enable the account store, AD FS can use account information for authentication. After you select and enable the account store, you must map the organizational claims to the attributes in the account store. For example, you would map the e-mail address identity claim to the e-mail address attribute for a user account. • The UPN claim is automatically mapped. • To map the e-mail address or common name claims, edit the existing mapping and specify the attribute using the LDAP syntax. • To map group claims, create a Group Claim Extraction. • To map custom claims, create a Custom Claim Extraction.

Applications	Configure applications to identify the claims- or Windows NT-based applications. When you configure the application, the Federation Service is authorized to issue tokens for the application, and users from the account store and account partner are given access. To configure the application: • Identify the application type (claims- or token-based). • Configure the URL to run the application. • Identify the types of claims that are accepted for the application. ***Note**: To enable a claim for use by an application, the claim must first be defined in the list of organizational claims.* In addition to configuring the AD FS settings, you will need to configure application settings in IIS.
Partner Organizations	When you define partner organizations, you create one side of the federation trust. • You create the trust on your server; the partner organization must create a corresponding trust. • When you define the trust, you identify the partner as either an account partner (i.e. the other organization maintains the user accounts) or as a resource partner (the other organization has the resources that your accounts need to access). • If you define an account partner, the other organization must define a corresponding resource partner definition to identify your organization. If you define a resource partner, the other organization must define a corresponding account partner definition to identify your organization. When you define the partner organization: • For an account partner, you must import the certificate of the federation server where the accounts reside.

	• For both partner types, select the scenario that identifies the trust relationship:
	  o Chose **Federated Web SSO with Forest Trust** to establish a trust with a target forest where an Active Directory forest trust already exists.
	  o Choose **Federated Web SSO** to establish a trust with a different organization when a forest trust does not exist.
	• Select the claims that will be used for the trust.
	  o For an account partner, you identify the UPN or e-mail suffixes that you will accept.
	  o For a resource partner, you identify the claims you will provide. **Note**: You can configure the claims to replace domain suffixes with a new value. For example, if your UPN suffix is **@westsim.com**, you can configure the claim to replace this with a new value, such as **@partner.com**.

The basic process for configuring AD FS is:

1. Install the AD FS role.
2. Configure claims to specify the user account information that will be included in the claims.
3. Enable a user account store, either AD DS or AD LDS.
4. Map claims to attributes in the user account store.
5. Identify applications that will be consumed (used), configuring the claims that are accepted by the application.
6. Define account or partner organizations.

Corresponding steps must also be performed at the partner organization to create the other side of the trust.

9.6. AD RMS

Active Directory Rights Management Services (AD RMS) is an additional server role that helps safeguard digital information from unauthorized use.

- *Usage policies* define users and the permitted actions they can perform on digital media. For example, you can prevent copying, printing, modifying, or even viewing files.
    - Usage policies remain with the file, regardless of where it is moved. This prevents intentional or unintentional misuse of electronic documents.
    - Administrators can define usage policy *templates* that configure predefined levels of access.
- The ability to work with protected content is controlled through various licenses.
    - A *client* license is issued to a user, and identifies the user as the owner of the content.
        - Only the owner can define or modify usage rights.
        - Usage rights and protected documents are created from within RMS-enabled applications, such as Microsoft Office 2007 Enterprise, Professional Plus, or Ultimate (one of these versions is required to create protected content).
    - A *publishing* license is issued for each protected document. This license contains the usage right information. Following issuance of the publishing license, the document content is encrypted by the AD RMS-enabled application.
    - A *use* license is issued to the recipient or user after AD RMS authenticates the user and verifies the usage rights defined for that user. Authorized users can only open rights-protected files on an RMS-enabled client computer and within an RMS-enabled browser or application.
- An AD RMS system has the following components:

Component	Description
AD RMS server	The AD RMS server is responsible for issuing licenses.
Database server	The database server stores configuration and policy information. The database can be hosted by the Windows Internal Database or another database server (such as a SQL server). ***Note****: Using the Internal Database is not recommended in a production environment.*
AD DS	The RMS server must be able to contact a domain controller for user authentication and other information. AD RMS uses Active Directory Domain Services (AD DS) to regulate access to all AD RMS users in the AD DS forest that have rights-protected content. If AD DS is not available, AD RMS cannot grant licenses to publish and consume rights-protected content.
AD RMS-enabled application	The RMS-enabled application is responsible for: o Providing the method to create digital content. o Allowing authors to set usage policies. o Encrypting content after the publishing license is obtained. o Enforcing usage policies by allowing only the defined actions to be taken based on the publishing and the use licenses.
AD RMS client	The AD RMS client facilitates communication between the server and the application. Windows Vista includes the AD RMS client by default; other client operating systems must have the AD RMS client installed.

- AD RMS can be implemented with Active Directory Federation Services (AD FS) to allow partner organizations access to controlled content.
- AD RMS trust policies allow users to share rights-protected content across internal or external Active Directory Domain Services (AD DS) forests.
- AD RMS supports the following trust hierarchies:

    - The *ISV* hierarchy is used for developing AD RMS-enabled applications.
    - The *production* hierarchy should be used for all production installations of AD RMS, unless you are developing an AD RMS-enabled application.
- AD RMS consists of the following services:
    - *Logging services* send logging information from each server in the AD RMS cluster to the logging database.
    - *Web services* provide communication between computers in the AD RMS cluster.

© Sergey Gorokhod
MCT/MCSE/MCITP/MCTS/MCSA/CCSE/CCSA/CCNA/A+®
E-mail: sergey@infosec.co.il
Mob: (+972) 526848757

9.7. AD RMS Installation

AD RMS has the following hardware and software requirements:

- The AD RMS role must be installed on a Windows Server 2008 or higher domain member or domain controller. It cannot be deployed on a server that is part of a workgroup. Domain controllers must be running Windows Server 2000 with Service Pack 3 or higher.
- Install AD RMS on a server that is a member of the same AD DS domain as the user accounts that will be consuming rights-protected content.
- It is recommended that the server hosting the AD RMS role be formatted with the NTFS file system.
- The server hosting the AD RMS role must have message queuing enabled.
- Users who acquire licenses or publish content must have an e-mail address configured as a property of the user account.
- The user account that installs AD RMS must have the ability to query the domain and create new databases on the database server. If Microsoft SQL Server 2005 or higher is used, the user account for installing AD RMS must be a member of the System Administrators database role or equivalent.
- All users and groups who use AD RMS to acquire licenses and publish content must have an e-mail address configured in Active Directory.
- After the installation, you cannot change the domain membership of the AD RMS server.
- The following Web services will be added automatically during installation if they do not already exist:
    - Internet Information Services (IIS)
    - ASP.NET must be enabled

The following table describes the configuration choices to make during AD RMS installation.

Configuration Value	Description
Cluster	An AD RMS *cluster* is a single server or a group of servers running AD RMS that share AD RMS publishing and licensing requests from AD RMS clients. During installation, you can

	install a new cluster, or install the server into an existing cluster (if one exists). There are two types of clusters:
	
	• The *root* cluster provides AD RMS services for the Active Directory Domain Services (AD DS) domain in which it was installed. The root cluster is the first server in an AD RMS installation. Joining a server to a root cluster is the best way to increase the availability and redundancy of your deployment.
	• *Licensing-only* clusters provide licensing and publishing services. Licensing-only clusters:
	  o Are created in addition to the root cluster.
	  o Can be set with options such as rights templates, logging, membership in the super users group, and exclusion policies.
	  o Can be used to:
	  ▪ Isolate different departments within an organization by not establishing a trusted publishing domain between RMS clusters. This is makes it so content can only be consumed by users that have access to a given licensing server.
	  ▪ Support unique rights-management requirements of a department by setting up a licensing-only cluster that is dedicated to a group's needs, and configuring rights policy templates separately for that licensing-only cluster.
	  ▪ Support rights management for business partners that are part of an extranet.
	
	You must be a member of the local Administrators group to join servers to a licensing-only cluster.

Database location	The database server stores configuration, policy, and logging information. You can use one of the following database locations:

	• When you use the Windows Internal Database, the database is located on the AD RMS server and configured automatically. This option is intended for use in test environments only. When using this option, the AD RMS cluster is limited to a single server. • When you use a different database server, the database is saved on an external database such as SQL server. The database server can be running on the same computer or a different computer. You must use this option to have more than one server in a cluster.
Service account	AD RMS requires a user account as a service account for communicating with other services. • The service account cannot be the same as the user account used to install AD RMS. • If AD RMS is installed on a domain member, do not grant any additional rights to the service account. • If AD RMS is installed on a domain controller, add the user to the Domain Admins group. Otherwise, installation will fail with a message stating that the password cannot be verified.
Cluster key	The cluster key is used to digitally sign certificates and licenses. During setup, you need to: • Identify where the cluster key is stored, either locally or in a CSP storage location. If you choose a CSP location, you must copy the key to each new server in the cluster. • Configure an administrator password. The password is used to encrypt the cluster key. It is required for key recovery, to join a server to the cluster, or to restore from backup.

Cluster address	The cluster address is a URL that is used by clients to publish and consume rights-protected content. When defining the cluster address:
	
	• Choose a URL which is different from the server name. That way, if the server name changes in the future, you will not need to republish all content.
	• Create a separate DNS alias record for the AD RMS cluster URL and for the computer hosting the AD RMS configuration database. This is beneficial in the event that the AD RMS servers are retired, failure is experience, or the computer's name is changed. Having two separate alias records allows you to perform updates without having to publish all rights-protected files again.
	• When using Identity Federation Support, you must use SSL for the cluster address. If SSL is not enabled on the Web site, you will be prompted to select a certificate for SSL during installation.
	  o Use a Secure Sockets Layer (SSL) certificate issued from a trusted root certification authority when you install the AD RMS cluster. Self-signed certificates should only be used in a test environment.
	  o The SSL certificate must exist on the new server before it can be joined to an existing AD RMS cluster.

Service connection point (SCP)	A Service Connection Point (SCPs) allows AD RMS-enabled clients in your organization to find and access the AD RMS cluster. Registering an SCP places the cluster URL in Active Directory. SCPs can be registered in two ways: • During the installation of the AD RMS role. • The cluster **Properties** sheet in the Active Directory Rights Management Services console. This is also where you can modify an SCP.

	To register the AD RMS Service Connection Point (SCP) during installation, the installing user account must: • Have Write access to the Services container in AD DS. • Be a member of the AD DS Enterprise Admins group or equivalent.

Be aware of the following regarding AD RMS installation.

- AD RMS installation adds IIS if it is not already installed.
- To allow AD RMS to use the host-based firewall application that is installed and enabled by default in Windows Server 2008, the following port exceptions are created and enabled automatically when AD RMS is installed:
    - Port 80 is opened for HTTP.
    - Port 443 is opened for HTTPS or SSL communication.
- If you are using Microsoft SQL Server 2000 or later in a cluster or if the database is installed on a different server than the AD RMS server, you must make the following port exceptions on the database server that is hosting the AD RMS databases:
    - Port 1433 as the default Microsoft SQL Server listening port.
    - Port 445 as the SQL Server Named Pipes (used for provisioning the AD RMS server).
- *Decommissioning* is the process in which rights-protected content is decrypted when an AD RMS cluster is removed.
    - Until all rights-protected content has been decrypted, servers in the AD RMS cluster should remain available on the network in decommissioning mode.
    - No new content can be published as rights-protected while the AD RMS cluster is in decommissioning mode.

9.8. AD RMS Certificate

The following table lists the certificates and licenses that are used by AD RMS:

Certificate or License	Purpose
Server Licensor Certificate (SLC)	The Server Licensor Certificate (SLC) is a self-signed certificate created when the AD RMS server role is installed and configured on the first server in the cluster. Be aware of the following details: • The SLC generates a unique SLC for itself that establishes its identity, called *self-enrollment*, and has a validity time of 250 years. This enables the archiving of rights-protected data for an extended period of time. • A root cluster handles both certification, by issuing a rights account certificate (RAC), and licensing rights-protected content. Other servers added to the root cluster share an SLC. • In complex environments, licensing-only clusters can be deployed, which generate their own SLC. • With the SLC, AD RMS can operate in a network that is entirely isolated from the Internet. • The SLC contains the public key of the server.
Rights Account Certificate (RAC)	The Rights Account Certificate (RAC) establishes a user's identity in the AD RMS system. It is created by the AD RMS root cluster and provided to the user when first attempting to open rights-protected content. Be aware of the following details: • A standard RAC identifies a user by account credentials in the context of a specific computer or device and has a validity time measured in number of days. The default validity time for a standard RAC is 365 days. • A temporary RAC identifies a user based on account credentials only and has a validity time measured in

	number of minutes. The default validity time for a temporary RAC is 15 minutes. • The RAC contains the public key of the user and the private key of the user encrypted with the public key of the activated computer.
Client Licensor Certificate (CLC)	The Client Licensor Certificate (CLC) is created by the AD RMS cluster in response to a request from the client application. Be aware of the following details: • The CLC is sent to the client while it is connected to the organization's network and grants the user the right to publish rights-protected content when the client is not connected. • The CLC is tied to the RAC of the user, so that if the RAC is not valid or not present, the user is not able to access the AD RMS cluster. • The client licensor public key is contained in the CLC, along with the client licensor private key that is encrypted by the public key of the user who requested the certificate. It also contains the public key of the cluster that issued the certificate, which is signed by the private key of the cluster that issued the certificate. • The client licensor private key is used to sign publishing licenses.
Machine Certificate	The machine certificate is created on the client computer the first time that an AD RMS-enabled application is used. Be aware of the following details: • The AD RMS client in Windows Vista automatically activates and enrolls with the root cluster to create this certificate on the client computer. • This certificate identifies a lockbox on a computer or device that is correlated with the logged-on user profile.

	• The machine certificate contains the public key of the activated computer. The corresponding private key is contained by that computer's lockbox.
Publishing License	The publishing license is created by the client when content is saved with rights-protection. Be aware of the following details: • The publishing license specifies the users that can open the rights-protected content, under which conditions the content may be opened by the user, and the rights that each user will have to the rights-protected content. • The publishing license contains the symmetric content key for decrypting the content, which is encrypted with the public key of the server that issued the license.
Use License	The use license specifies the rights that apply to the rights-protected content in the context of a specific authenticated user. Be aware of the following details: • This license is tied to the RAC. If the RAC is not valid or not present, the use license cannot be used to work with the content. • The use license contains the symmetric content key for decrypting the content, which is encrypted with the public key of the user.