

Generate

Installation and Configuration

Version 4.2

*Updated 2022-03-22*

Table of Contents

[Installation 1](#_Toc10102643)

[System Requirements 1](#_Toc10102644)

[Installation Steps 1](#_Toc10102645)

[Database Server 1](#_Toc10102646)

[Web Server 1](#_Toc10102647)

[Web Application 2](#_Toc10102648)

[Background Application 5](#_Toc10102649)

[Permission Changes 10](#_Toc10102650)

[Active Directory (AD) 10](#_Toc10102651)

[Final Steps 11](#_Toc10102652)

[Update 12](#_Toc10102653)

[Database Server 12](#_Toc10102654)

[Web Application 13](#_Toc10102655)

[Background Application 14](#_Toc10102656)

[Data Migration 15](#_Toc10102657)

[Migration Process 15](#_Toc10102658)

[Location of ETL Code 15](#_Toc10102659)

[Operational Data Store (ODS) Structure and Reference Data 16](#_Toc10102660)

[ODS Tables Used by Generate 16](#_Toc10102661)

[ETL General Guidance 17](#_Toc10102662)

[ETL Table Guidance 17](#_Toc10102663)

[Optional Installations 21](#_Toc10102664)

[Active Directory Lightweight Directory Services (AD LDS) 21](#_Toc10102665)

# Installation

## System Requirements

1. Web Server:
   1. Windows 2012 R2 (or newer) with IIS 8.5 (or newer)
   2. At least 8 GB of RAM
2. Database Server:
   1. Windows 2012 R2 (or newer) with SQL Server 2012 (or newer)
   2. At least 16 GB of RAM (32 GB RAM recommended)
   3. At least 50 GB of free storage
3. Web Server and Database Server may be on the same machine, if desired.
4. .NET Core Runtime and Hosting Bundle (version 2.2) installed on Web Server.
5. Active Directory (AD) accessible from the Web Server. If no AD installation exists, AD LDS may be installed.
6. Application Initialization Module for IIS installed on the Web Server.

## Installation Steps

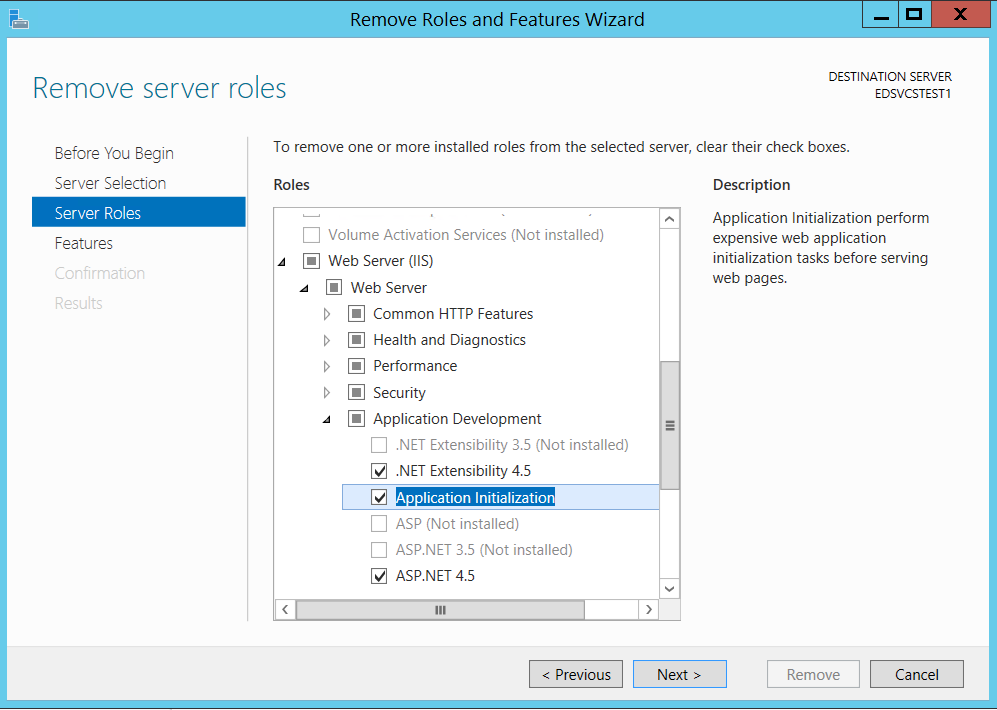
If you are upgrading from an earlier installation, please skip to the [Update](#_Upgrade) section. Please note that screenshots accompany certain installation steps.

### Database Server

1. Create an empty database called “generate” in SQL Server.
2. Create a SQL login called “generate”.
3. Using the Generate database backup file for this version (generate\_x.x.bak where x.x is the version number), restore to the new empty “generate” database. This backup file was created from SQL Server 2012 R2. If you are using a newer version of SQL Server and encounter difficulty with this restore, a newer version of the backup file can be provided.
4. Please note, the database contained within the backup file is called “generate-test”. Please make sure to restore to the “generate” database just created.
5. Restore “db\_owner” permissions to the “generate” user on the “generate” database.

### Web Server

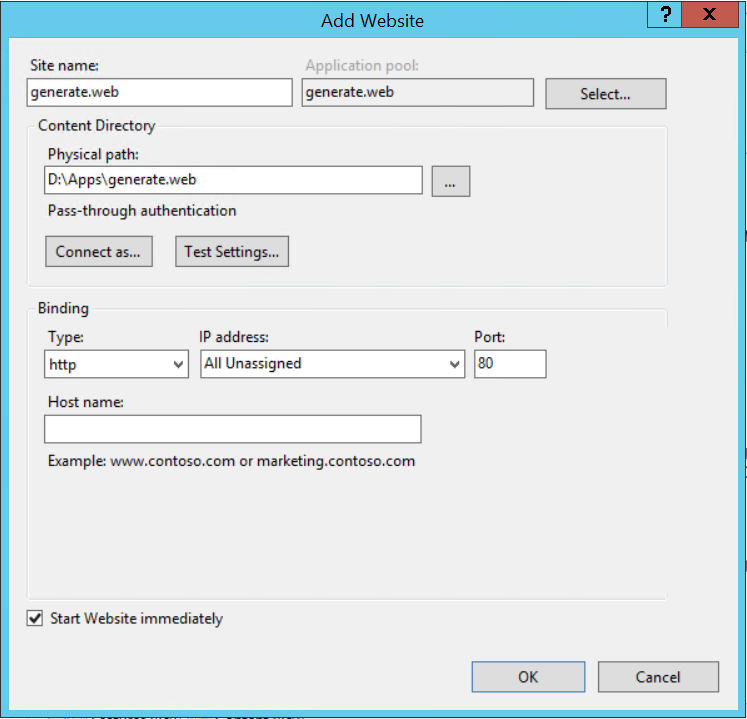
1. Ensure IIS is installed.
2. Ensure all pending Windows Updates are installed.
3. Install ASP.NET Core Runtime 6.0.11 Hosting Bundle. Installation files can be found here:
   1. <https://dotnet.microsoft.com/en-us/download/dotnet/thank-you/runtime-aspnetcore-6.0.11-windows-hosting-bundle-installer>
4. Execute iisreset at the command line or restart the server to pick up changes to the system PATH.
5. Install the Application Initialization Module for IIS. You can install the Application Initialization module via the Server Manager. You can find the module under Server Roles -> Web Server -> Application Developer -> Application Initialization. Please note, this is the only item that is required for this installation -- any other items checked in following screenshot are not necessary and may remain checked or unchecked depending on your environment.



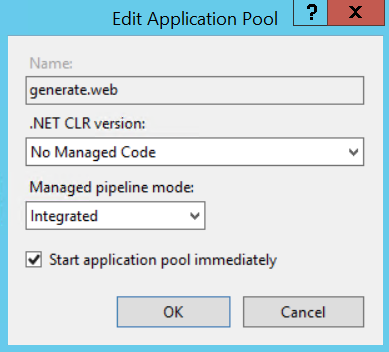
1. Please note, the Web Application will need to be accessible independent of the Background Application. You can achieve this by one of the following methods:
   1. Place the web application and background applications on different servers.
   2. Place both applications on the same server but assign different IP addresses to each website.
   3. Place both applications on the same server but assign different ports to each website.
   4. Place both applications on the same server but assign different host names to each website.

### Web Application

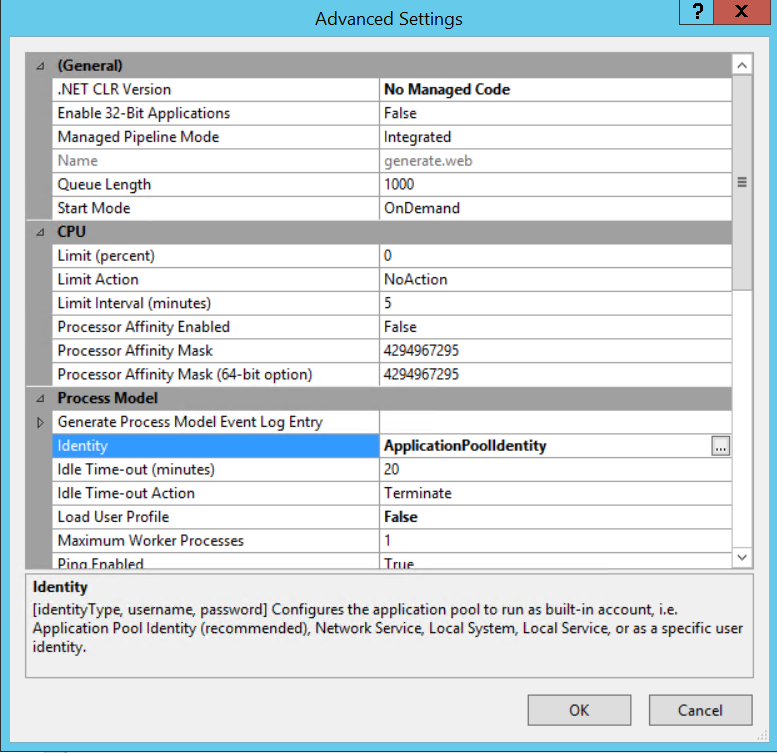
1. Extract contents of “generate.web\_x.x.zip” into a temporary directory.
2. Copy the “generate.web\_x.x” folder into a location where you want your web application files to reside.
3. In IIS, create a website called “generate.web” and point the physical path to the location of the web application files. Leave the application pool set to a new pool called “generate.web”. Adjust the IP Address, Port, and/or Host name as needed for your environment (as mentioned in the Web Server installation steps).



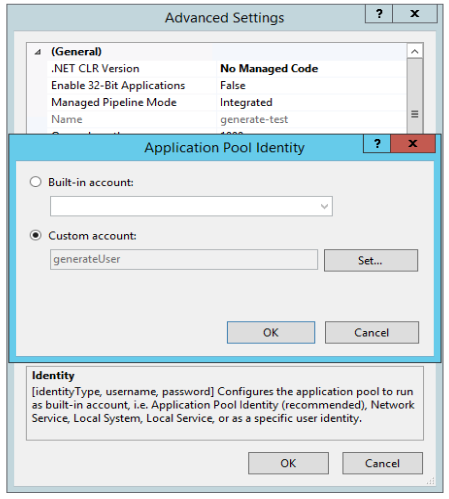
1. Set the application pool used by the generate.web website to “No Managed Code.”



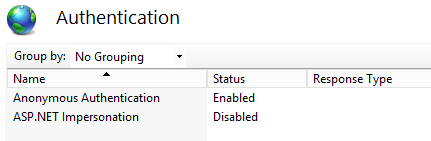
1. The application pool used by the Generate website needs to be run under a service account that has access to the Active Directory service. Usually, this account is a domain service account on the network. This can be changed by accessing the Advanced Settings of the newly created application pool.



1. From that dialog, change the Identity setting under the Process Model section to the domain service account.



1. Make sure that the Application Pool Identity user account has permission to Create and Modify files in the Updates and Logs directories of the website folder.
2. Ensure that Anonymous Authentication is enabled on the website.



1. Open the “appSettings.json” file located in the “Config” directory of the generate.web application folder. To retain proper formatting, Notepad++ is recommended over Notepad or Wordpad.
2. Ensure that the “AppSettings.Environment” value is set to “production.”   
   {

"AppSettings": {

"Environment": "production",

"ProvisionJobs": false,

"ADDomain": "ETSS.local",

"ADLoginDomain": "ETSS",

"ADPort": "389",

"ADContainer": "CN=Builtin,DC=ETSS,DC=local",

"UserContainer": "CN=Users,DC=ETSS,DC=local",

"ReviewerGroupName": "CN=Users,CN=Builtin,DC=ETSS,DC=local",

"AdminGroupName": "CN=Administrators,CN=Builtin,DC=ETSS,DC=local",

"BackgroundUrl": "http://192.168.1.2",

"BackgroundAppPath": "D:\\Apps\\generate.background"

},

"Data": {

"AppDbContextConnection": "Server=192.168.01.01;Database=generate;User ID=generate;Password=xxxxxxxxxxx;MultipleActiveResultSets=true;",

"ODSDbContextConnection": "Server=192.168.01.01;Database=generate;User ID=generate;Password=xxxxxxxxxxx;MultipleActiveResultSets=true;",

"RDSDbContextConnection": "Server=192.168.01.01;Database=generate;User ID=generate;Password=xxxxxxxxxxx;MultipleActiveResultSets=true;Connect Timeout=300;"

}

}

1. Replace the values of the SQL Server Connection String keys with values appropriate for the installed environment.

{

"AppSettings": {

"Environment": "production",

"ProvisionJobs": false,

"ADDomain": "ETSS.local",

"ADLoginDomain": "ETSS",

"ADPort": "389",

"ADContainer": "CN=Builtin,DC=ETSS,DC=local",

"UserContainer": "CN=Users,DC=ETSS,DC=local",

"ReviewerGroupName": "CN=Users,CN=Builtin,DC=ETSS,DC=local",

"AdminGroupName": "CN=Administrators,CN=Builtin,DC=ETSS,DC=local",

"BackgroundUrl": "http://192.168.1.2",

"BackgroundAppPath": "D:\\Apps\\generate.background"

},

"Data": {

"AppDbContextConnection": "Server=192.168.01.01;Database=generate;User ID=generate;Password=xxxxxxxxxxx;MultipleActiveResultSets=true;",

"ODSDbContextConnection": "Server=192.168.01.01;Database=generate;User ID=generate;Password=xxxxxxxxxxx;MultipleActiveResultSets=true;",

"RDSDbContextConnection": "Server=192.168.01.01;Database=generate;User ID=generate;Password=xxxxxxxxxxx;MultipleActiveResultSets=true;Connect Timeout=300;"

}

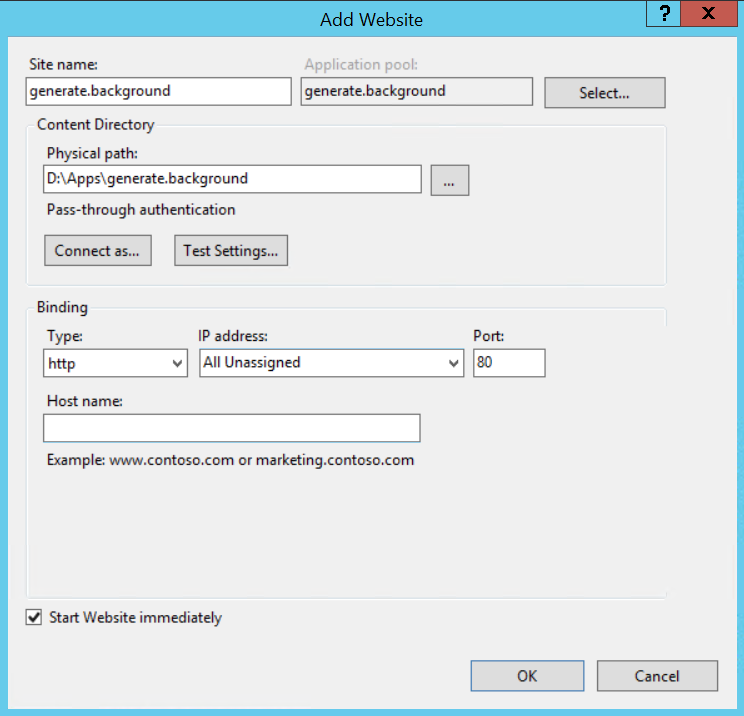
}

The SQL Connection String should be the same for all three connections.

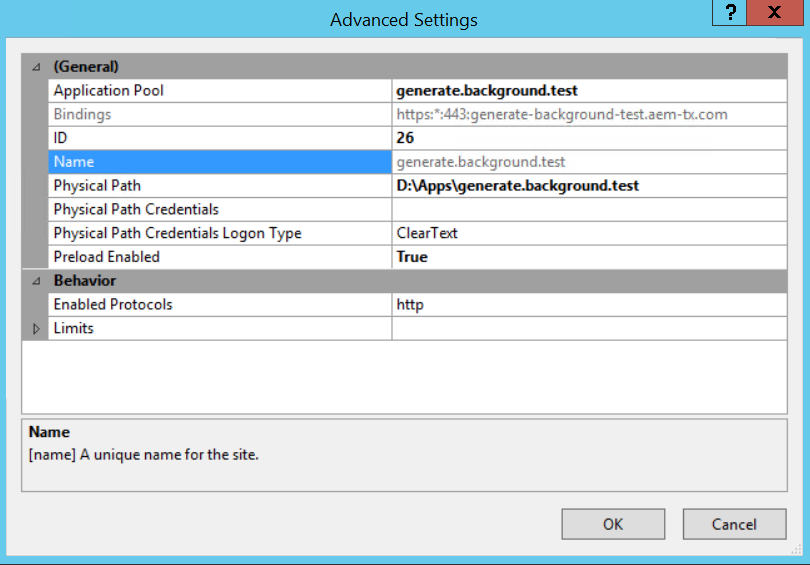
1. If the connection string contains a backslash (\), it should be escaped with an additional backslash to retain proper JSON syntax.

### Background Application

1. Please ensure that the Application Initialization Module for IIS was installed as part of step 5 of the Web Server installation instructions.
2. Extract contents of “generate.background\_x.x.zip” into a temporary directory.
3. Copy the “generate. background \_x.x” folder into a location where you want your background application files to reside.
4. In IIS, create a website called “generate.background” and physical path to the location of the background application files. Leave the application pool set to a new pool called “generate. background”. Adjust the IP Address, Port, and/or Host name as needed for your environment (as mentioned in the Web Server installation steps).

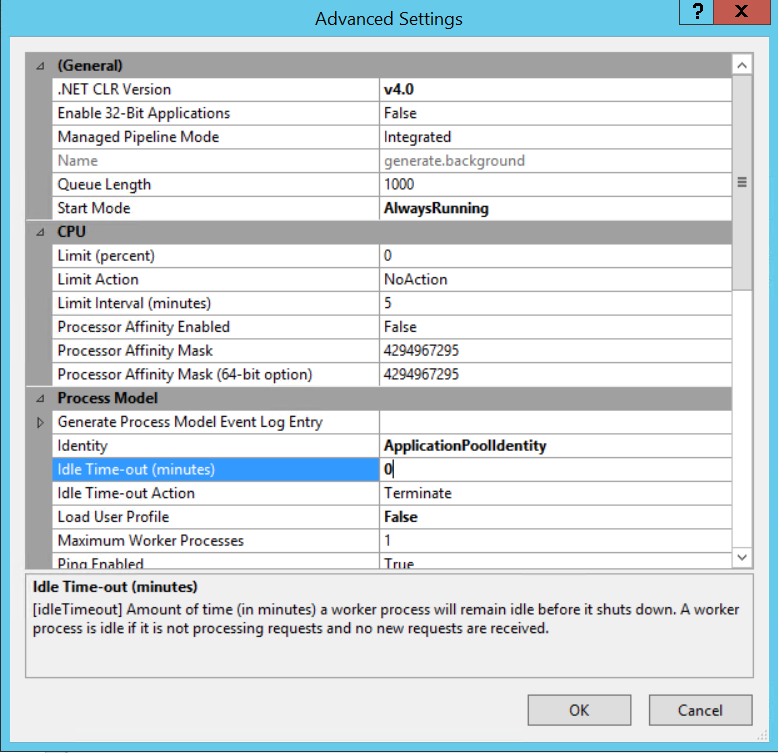


1. Edit the “Advanced Settings” of the generate.background website and make sure the following property is set:
   1. Preload Enabled = True

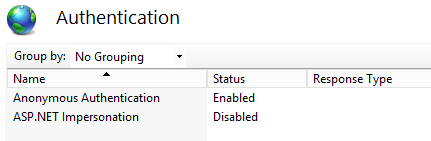


Please note, the names and paths in the preceding screenshot say “generate.background.test”, but will say “generate.background” in this installation.

1. Edit the “Advanced Settings” of the generate.background application pool and make sure the following properties are set:
   1. .NET CLR Version = v4.0
   2. Start Mode = AlwaysRunning
   3. Idle Time-Out (minutes) = 0



1. Make sure that the Application Pool Identity user account used by this website has permission to Create and Modify files in the Updates and Logs directories of the application folder.
2. Ensure that Anonymous Authentication is enabled on the website.



1. Open the “appSettings.json” file located in the generate.background application folder. To retain proper formatting, Notepad++ is recommended over Notepad or Wordpad.
2. Ensure that the “AppSettings.Environment” value is set to “production.”   
   {

"AppSettings": {

"Environment": "production",

"WebAppPath": "D:\\Apps\\generate.web"

},

"Data": {

"HangfireConnection": "Server=192.168.1.1;Database=generate;User ID=generate;Password=xxxxxxxxxxx;MultipleActiveResultSets=true;",

"AppDbContextConnection": "Server=192.168.1.1;Database=generate;User ID=generate;Password=xxxxxxxxxxx;MultipleActiveResultSets=true;Connect Timeout=3600;",

"ODSDbContextConnection": "Server=192.168.1.1;Database=generate;User ID=generate;Password=xxxxxxxxxxx;MultipleActiveResultSets=true;Connect Timeout=300;",

"RDSDbContextConnection": "Server=192.168.1.1;Database=generate;User ID=generate;Password=xxxxxxxxxxx;MultipleActiveResultSets=true;Connect Timeout=300;"

}

}

1. Replace the values of the SQL Server Connection String keys with values appropriate for the installed environment.

{

"AppSettings": {

"Environment": "production",

"WebAppPath": "D:\\Apps\\generate.web"

},

"Data": {

"HangfireConnection": "Server=192.168.1.1;Database=generate;User ID=generate;Password=xxxxxxxxxxx;MultipleActiveResultSets=true;",

"AppDbContextConnection": "Server=192.168.1.1;Database=generate;User ID=generate;Password=xxxxxxxxxxx;MultipleActiveResultSets=true;Connect Timeout=3600;",

"ODSDbContextConnection": "Server=192.168.1.1;Database=generate;User ID=generate;Password=xxxxxxxxxxx;MultipleActiveResultSets=true;Connect Timeout=300;",

"RDSDbContextConnection": "Server=192.168.1.1;Database=generate;User ID=generate;Password=xxxxxxxxxxx;MultipleActiveResultSets=true;Connect Timeout=300;"

}

}

The SQL Connection String should be the same for all three connections.

1. If the connection string contains a backslash (\), it should be escaped with an additional backslash to retain proper JSON syntax.
2. Replace the value of the WebAppPath key with a value appropriate for the installed environment. This should be the physical path of the generate.web application files (specified in step 3 of the Web Application installation steps).

{

"AppSettings": {

"Environment": "production",

"WebAppPath": "D:\\Apps\\generate.web"

},

"Data": {

"HangfireConnection": "Server=192.168.1.1;Database=generate;User ID=generate;Password=xxxxxxxxxxx;MultipleActiveResultSets=true;",

"AppDbContextConnection": "Server=192.168.1.1;Database=generate;User ID=generate;Password=xxxxxxxxxxx;MultipleActiveResultSets=true;Connect Timeout=3600;",

"ODSDbContextConnection": "Server=192.168.1.1;Database=generate;User ID=generate;Password=xxxxxxxxxxx;MultipleActiveResultSets=true;Connect Timeout=300;",

"RDSDbContextConnection": "Server=192.168.1.1;Database=generate;User ID=generate;Password=xxxxxxxxxxx;MultipleActiveResultSets=true;Connect Timeout=300;"

}

}

1. Open the “appSettings.json” file located in the “Config” directory of the generate.web application folder. Replace the value of the BackgroundUrl and BackgroundAppPath keys with values appropriate for the installed environment. If these keys do not exist in the .json file, please add them. BackgroundUrl refers to the URL of the generate.background application. BackgroundAppPath refers to the physical path of the generate. background application files (specified in step 3 of the Background Application installation steps).

{

"AppSettings": {

"Environment": "production",

"ProvisionJobs": false,

"ADDomain": "ETSS.local",

"ADLoginDomain": "ETSS",

"ADPort": "389",

"ADContainer": "CN=Builtin,DC=ETSS,DC=local",

"UserContainer": "CN=Users,DC=ETSS,DC=local",

"ReviewerGroupName": "CN=Users,CN=Builtin,DC=ETSS,DC=local",

"AdminGroupName": "CN=Administrators,CN=Builtin,DC=ETSS,DC=local",

"BackgroundUrl": "http://192.168.1.2",

"BackgroundAppPath": "D:\\Apps\\generate.background"

},

"Data": {

"AppDbContextConnection": "Server=192.168.01.01;Database=generate;User ID=generate;Password=xxxxxxxxxxx;MultipleActiveResultSets=true;",

"ODSDbContextConnection": "Server=192.168.01.01;Database=generate;User ID=generate;Password=xxxxxxxxxxx;MultipleActiveResultSets=true;",

"RDSDbContextConnection": "Server=192.168.01.01;Database=generate;User ID=generate;Password=xxxxxxxxxxx;MultipleActiveResultSets=true;Connect Timeout=300;"

}

}

### Permission Changes

For the Auto-Update functionality of Generate to work properly, you must ensure that the following permissions are set on the generate.web and generate.backgound directories:

* The Application Pool Identity user account used by the generate.background site, must have Create and Modify permissions on the entire generate.web application directory.
* The Application Pool Identity user account used by the generate.web site, must have Create and Modify permissions on the entire generate.background application directory.

### Active Directory (AD)

Generate makes use of Active Directory (AD) for authentication and authorization. If an existing installation of AD is not available, Active Directory Lightweight Directory Services (AD LDS) may be installed. For those instructions, please see the AD LDS instructions in the [Optional Installations](#_Optional_Installations) section.

The AD settings for Generate are configured in the “appsettings.json” file located in the “Config” folder of the web application directory. Please see the following for the configuration settings.

{

"AppSettings": {

"Environment": "production",

"ProvisionJobs": false,

"ADDomain": "ETSS.local",

"ADLoginDomain": "ETSS",

"ADPort": "389",

"IsSSLEnabled": false,

"ADContainer": "CN=Builtin,DC=ETSS,DC=local",

"UserContainer": "CN=Users,DC=ETSS,DC=local",

"ReviewerGroupName": "CN=Users,CN=Builtin,DC=ETSS,DC=local",

"AdminGroupName": "CN=Administrators,CN=Builtin,DC=ETSS,DC=local",

"BackgroundUrl": "http://192.168.1.2",

"BackgroundAppPath": "D:\\Apps\\generate.background"

},

"Data": {

"AppDbContextConnection": "Server=192.168.01.01;Database=generate;User ID=generate;Password=xxxxxxxxxxx;MultipleActiveResultSets=true;",

"ODSDbContextConnection": "Server=192.168.01.01;Database=generate;User ID=generate;Password=xxxxxxxxxxx;MultipleActiveResultSets=true;",

"RDSDbContextConnection": "Server=192.168.01.01;Database=generate;User ID=generate;Password=xxxxxxxxxxx;MultipleActiveResultSets=true;Connect Timeout=300;"

}

}

**ADDomain**: The name of the AD domain service. This should be a DNS resolvable name to the AD server.

**ADLoginDomain**: The AD domain where the user account is located.

**ADPort**: The port used to communicate with the AD server.

**IsSSLEnabled**: Set to “True” if SSL is required to communicate with the AD server.

**ADContainer**: The distinguished name of the root-level container common amongst the UserContainer, ReviewerGroupName, and AdminGroupName.

**UserContainer**: The distinguished name of the container where the user accounts are located.

**ReviewerGroupName:** The distinguished name of Generate Reviewer group.

**AdminGroupName:** The distinguished name of Generate Admin group.

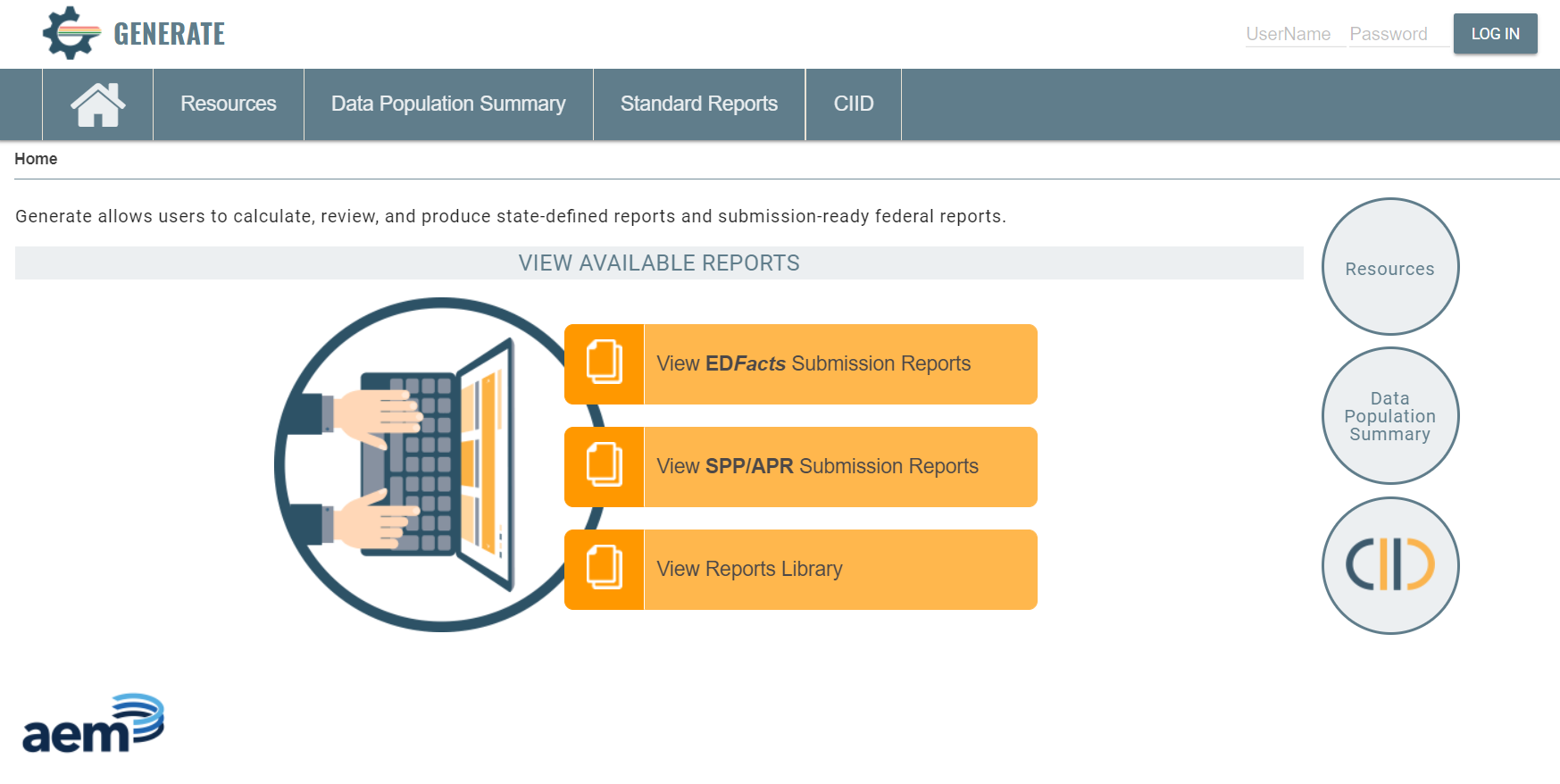
Generate makes use of two groups within Active Directory:

* Administrators
* Reviewers

Both AD roles must be configured in the AD instance and accessible by Generate. User access is provisioned via normal AD tools by adding or removing these roles to or from the desired users.

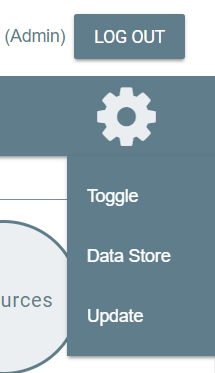
### Final Steps

1. Start up the web application in IIS.
2. Open the website in a browser to confirm that it loads properly.



# Update

If the current version of Generate is 3.1 or higher, updates should be performed using the Automatic Update functionality of Generate. This can be found in the Settings menu under the menu item named “Update”. This feature is available to users with the Admin role.



If you are upgrading from 3.0 to 3.1, or would prefer to do a manual update, please use the following instructions.

Please note, if your version of Generate is older than version 2.4, you cannot perform a manual update and must follow the new installation instructions.

### Database Server

Please note, if you are upgrading 3.0 to 3.1, please **remove all SQL Server Agent jobs** that were previously installed for Generate. These jobs are no longer required and will conflict with the operation of Generate if allowed to remain.

There are two options for upgrading the database server:

1. Using a PowerShell script
2. Running SQL scripts manually

The PowerShell method is quicker but requires some basic knowledge of PowerShell. The SQL script method will take longer but only requires knowledge of running SQL scripts. We strongly recommend using the PowerShell method as it is much quicker and less prone to manual error.

#### PowerShell Method

1. Create a backup of your database in case the update fails, and you need to revert the changes.
2. In the web application directory on the Web Server, copy the “DatabaseScripts” directory to the database server.
3. Open the “RunDatabaseScripts.ps1” file in a text editor.
4. If needed, modify the $sqlCmdPath value to suit your environment. This should be the path of your sqlcmd.exe executable.
5. Save the file and open a PowerShell session.
6. Change the path to the directory where the “RunDatabaseScripts.ps1” file is located.
7. Execute the PowerShell script with the following parameters relevant to your environment:
   1. Version
   2. Server
   3. Database
   4. User
   5. Password
8. Here is an example:

./RunDatabaseScripts.ps1 "2.5" "192.168.51.53" "generate" "generate" "password"

1. Confirm that the script executes successfully.
2. If the script fails, you may revert the update by restoring the backup created in the first step.

#### SQL Scripts Method

1. Create a backup of your database in case the update fails, and you need to revert the changes.
2. In the web application directory on the Web Server, copy the “DatabaseScripts” directory to the database server.
3. Open the “VersionScripts.csv” file in the “VersionUpdates\x.x” directory for the relevant version.
4. Execute each script listed in that file against the generate database in the exact order they are listed in the file. The first column specifies the relative path of the file and the second column specifies the file name. You can ignore the third column.
5. It is important that every script is executed in order and runs successfully.
6. If any script fails, you may revert the update by restoring the backup created in the first step.

### Web Application

1. Only perform these steps if the database update was successful.
2. Ensure the web server has the proper version of the .NET Core Runtime and Hosting Bundle referenced in the System Requirements. Keep in mind this may have changed since the last version of Generate.
3. In IIS, stop the generate.web website.
4. Make a backup of the generate.web website folder.
5. Delete all files within the generate.web website folder, except for the “appSettings.json” file located in the “Config” directory.
6. Extract contents of “generate.web\_x.x.zip” into a temporary directory.
7. Copy all files from the just extracted “generate.web\_x.x” folder, except for the “appSettings.json” file located in the “Config” directory, into the website folder
8. Start up the web application in IIS.
9. Open the website in a browser to confirm that it loads properly.

### Background Application

Please note, the Background Application did not exist prior version 3.1. Therefore, if you are performing a manual update of any version prior to 3.1, then you will need to perform the new installation instructions for the Background Application.

1. Only perform these steps if the database and web update were successful.
2. In IIS, stop the generate.background website.
3. Make a backup of the generate.background website folder.
4. Delete all files within the generate.background website folder, except for the “appSettings.json” file located in the “Config” directory.
5. Extract contents of “generate.background\_x.x.zip” into a temporary directory.
6. Copy all files from the just extracted “generate. background \_x.x” folder, except for the “appSettings.json” file located in the “Config” directory, into the website folder
7. Start up the web application in IIS.

# Data Migration

Generate makes use of data that have been migrated from a State Longitudinal Database System (SLDS) into the Operational Data Store (ODS) of Generate. The Extraction, Transformation, and Loading (ETL) of these data must adhere to the following guidelines.

## Migration Process

When a user triggers an ODS Data Migration via the Generate application, the following steps are executed by the Generate application in this order:

1. The App.DataMigrationTask table is queried for all tasks with the following parameters:
   1. IsActive = 1
   2. RunBeforeGenerateMigration = 1
2. The stored procedures found in the App.DataMigrationTask.StoredProcedureName field from that query are then executed in ascending order of the App.DataMigrationTask.TaskSequence field.
3. If each of those tasks execute successfully, the Generate application performs its own data migration processes (e.g. establishing dimension data, reporting data, etc.).
4. If that process executes successfully, the App.DataMigrationTask table is again queried for all tasks with the following parameters:
   1. IsActive = 1
   2. RunAfterGenerateMigration = 1
5. The stored procedures found in the App.DataMigrationTask.StoredProcedureName field from that query are then executed in ascending order of the App.DataMigrationTask.TaskSequence field.

## Location of ETL Code

The ETL code should be placed in one or more stored procedures in the Generate database. They should reside in the App schema, or a new schema (as long as the generate SQL user has access to that schema).

For each stored procedure that is required, a new App.DataMigrationTask record should be created with the following values:

{

"StoredProcedureName": "[Stored procedure, along with any required parameters]",

"IsActive": 1,

"RunBeforeGenerateMigration": 1,

"RunAfterGenerateMigration": 0,

"TaskSequence": "[Sequence of stored procedure execution]",

"DataMigrationTypeId": [ID of the DataMigrationType record associated with the ODS migration],

}

## Operational Data Store (ODS) Structure and Reference Data

The Generate Operational Data Store (ODS) is based on the Common Education Data Standards (CEDS) Normalized Data Schema (NDS) and uses the same structure and reference data. Information about the NDS can be found here: <https://ceds.ed.gov/dataModelNDS.aspx>

When writing the ETL for Generate, the NDS documentation should be consulted for information regarding the purpose of each table/field, along with the reference data used.

## ODS Tables Used by Generate

Not including the reference data tables, the Generate application only uses those ODS tables required for the current functionality. Specifically, this includes the following:

* ODS.Assessment
* ODS.AssessmentAdministration
* ODS.AssessmentAdministration\_Organization
* ODS.AssessmentForm
* ODS.AssessmentPerformanceLevel
* ODS.AssessmentRegistration
* ODS.AssessmentResult
* ODS.AssessmentResult\_PerformanceLevel
* ODS.AssessmentSubtest
* ODS.K12SchoolStatus
* ODS.K12StaffAssignment
* ODS.K12StudentCohort
* ODS.K12StudentDiscipline
* ODS.K12StudentEnrollment
* ODS.K12SchoolGradeOffered
* ODS.K12OrganizationStudentResponsibility
* ODS.Incident
* ODS.Organization
* ODS.OrganizationCalendar
* ODS.OrganizationCalendarSession
* ODS.OrganizationIdentifier
* ODS.OrganizationPersonRole
* ODS.OrganizationProgramType
* ODS.OrganizationRelationship
* ODS.Person
* ODS.PersonCredential
* ODS.PersonDemographicRace
* ODS.PersonDisability
* ODS.PersonIdentifier
* ODS.PersonProgramParticipation
* ODS.PersonStatus
* ODS.ProgramParticipationCte
* ODS.ProgramParticipationMigrant
* ODS.ProgramParticipationSpecialEducation
* ODS.ProgramParticipationTitleIiiLep
* ODS.Role
* ODS.StaffCredential

As functionality is added to the Generate application, additional ODS tables may be used.

## ETL General Guidance

The Generate application will not remove or alter the data in the ODS in any way. Therefore, the ETL stored procedure code should be written in such a way that it can be run multiple times without failing. This will typically involve writing logic within the ETL to either delete or update the data as needed.

## ETL Table Guidance

When populating the ODS with data, the following guidelines should be followed.

#### ODS.Organization

This table is populated by the Generate application with one record with the name “Generate.” This record should not be removed or altered as it is used by the application. Additional data may be added to this table as long as they do not interfere with the seed data.

**Specific Organization Types:**

* State Education Agency (SEA) – this should be added with the following values:
* RefOrganizationTypeId that matches the following:

select RefOrganizationTypeId

from ods.RefOrganizationType ot

inner join ods.RefOrganizationElementType oet on ot.RefOrganizationElementTypeId = ot.RefOrganizationElementTypeId

where ot.Code = 'SEA' and oet.Code = '001156'

* Local Education Agency (LEA) – this should be added with the following values:
* RefOrganizationTypeId that matches the following:

select RefOrganizationTypeId

from ods.RefOrganizationType ot

inner join ods.RefOrganizationElementType oet on ot.RefOrganizationElementTypeId = ot.RefOrganizationElementTypeId

where ot.Code = 'LEA' and oet.Code = '001156'

* K12 School – this should be added with the following values:
* RefOrganizationTypeId that matches the following:

select RefOrganizationTypeId

from ods.RefOrganizationType ot

inner join ods.RefOrganizationElementType oet on ot.RefOrganizationElementTypeId = ot.RefOrganizationElementTypeId

where ot.Code = ' K12School' and oet.Code = '001156'

* Program (e.g. LEP or Special Education Program) – this should be added with the following values:
* RefOrganizationTypeId that matches the following:

select RefOrganizationTypeId

from ods.RefOrganizationType ot

inner join ods.RefOrganizationElementType oet on ot.RefOrganizationElementTypeId = ot.RefOrganizationElementTypeId

where ot.Code = ' Program' and oet.Code = '001156'

#### ODS.OrganizationIdentifier

Identifiers should be populated for both the State and NCES IDs for all Local Education Agencies and Schools.

#### ODS.OrganizationPersonRole

The pre-populated Role of “K12 Student” should be used whenever an OrganizationPersonRole record is being populated for a student. These records are used for school enrollment and program (e.g. LEP, Special Education, etc.) participation.

There should only be one OrganizationPersonRole record per combination of the following:

* Student
* Organization (e.g. school or program)
* School Year

If the source has multiple records, the ETL should have business logic to select the relevant record to load into the ODS.

#### ODS.OrganizationRelationship

Relationships between SEA, LEA, Schools, and Programs should be established using this table.

#### ODS.PersonIdentifier

At minimum, a state identifier for each student should be populated.

#### ODS.PersonStatus

LEP Status should be reflected in this table.

#### ODS.Role

This table is populated by the Generate application with two records: one for “K12 Student” and one for “K12 Personnel.” These records should not be removed or altered as they are used by the application. Additional data may be added to this table as long as they do not interfere with the seed data.

# Optional Installations

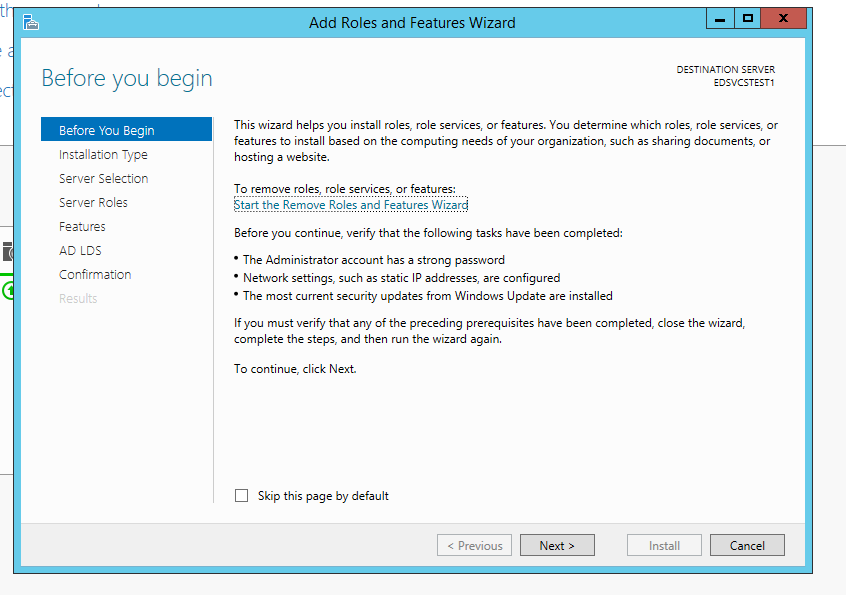
## Active Directory Lightweight Directory Services (AD LDS)

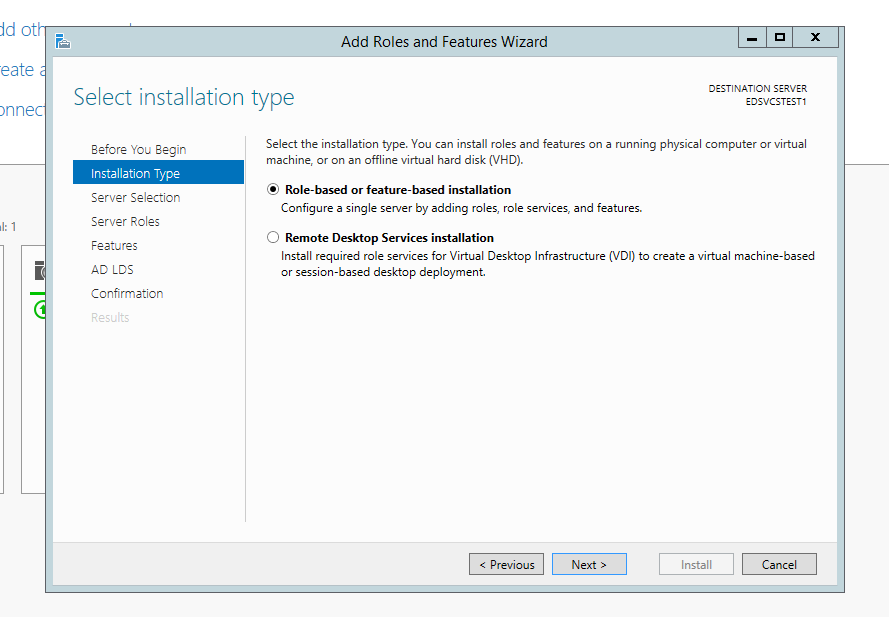
For those environments where Active Directory does not exist, Active Directory Lightweight Directory Services (AD LDS) can be used.

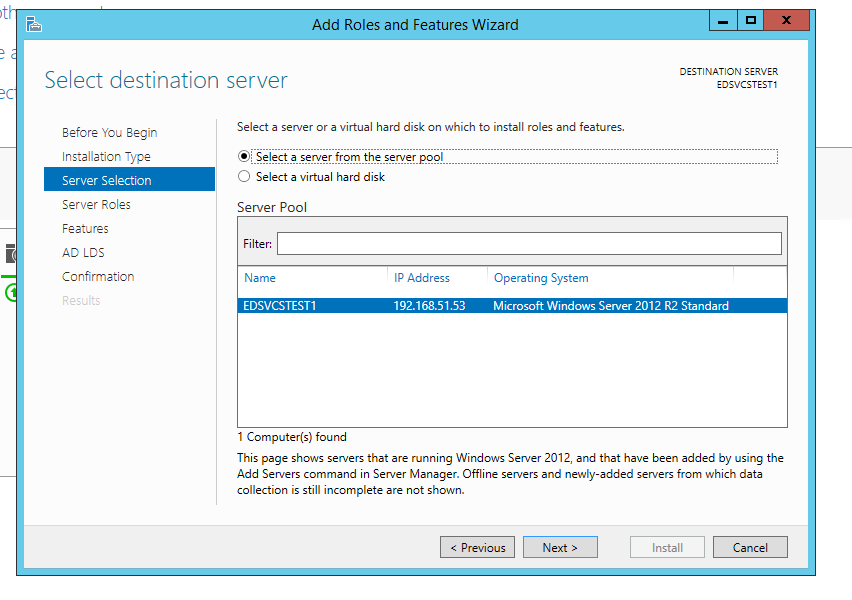
To install Active Directory Lightweight directory services (AD LDS) on a machine with Windows Server 2008 or higher, please use the following steps.

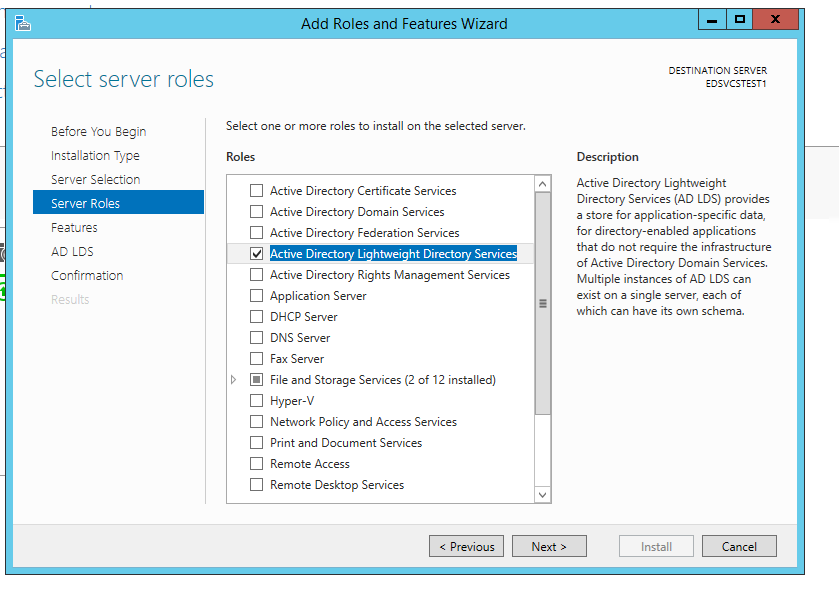
1. Click **Start**, and then click **Server Manager**.
2. In the console tree, right-click **Roles**, and then click **Add Roles**.
3. Review the information on the **Before You Begin** page of the Add Roles Wizard, and then click **Next**.
4. On the **Select Server Roles** page, in the **Roles** list, select the **Active Directory Lightweight Directory Services** checkbox, and then click **Next**.
5. Finish adding the AD LDS server role by following the instructions in the wizard.

Please use the following screen shots as a guide to install AD LDS.

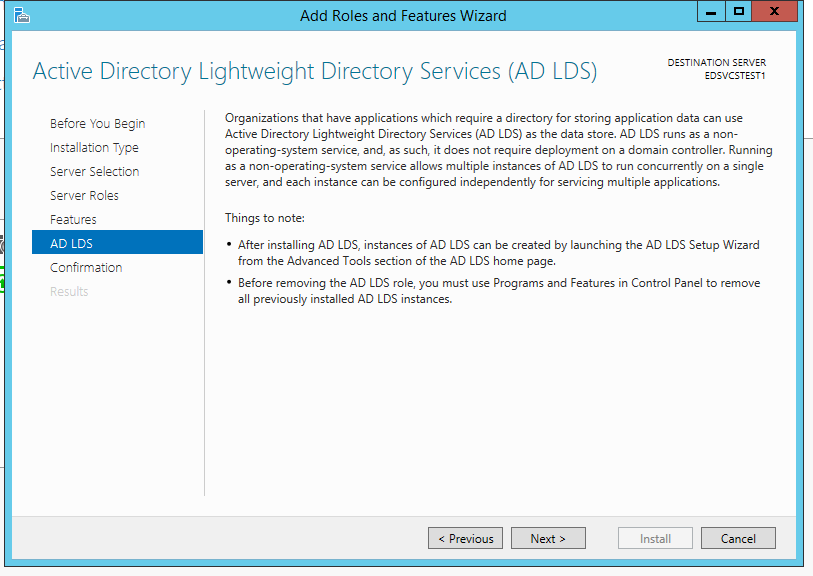


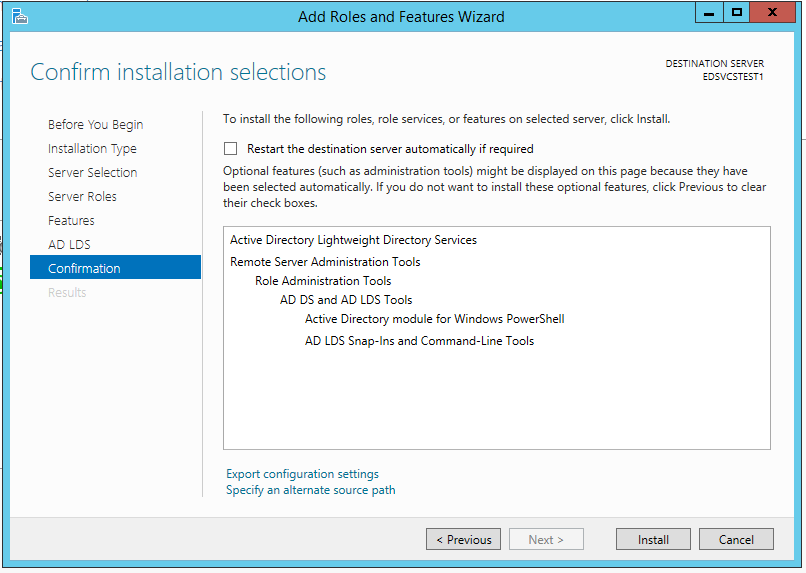




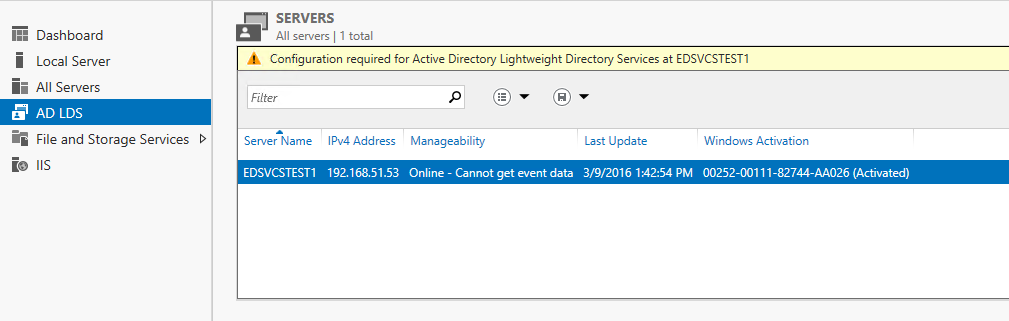


* Select AD LDS from the Roles and add the applicable features for the role.

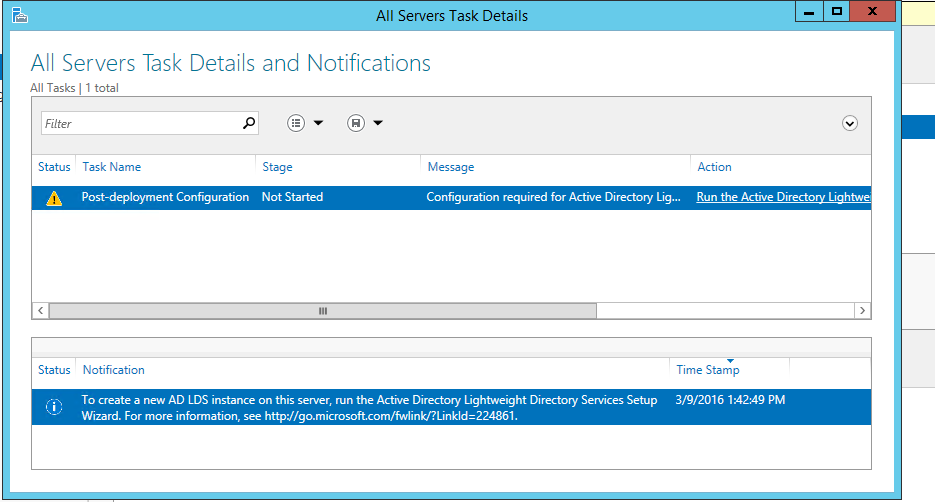


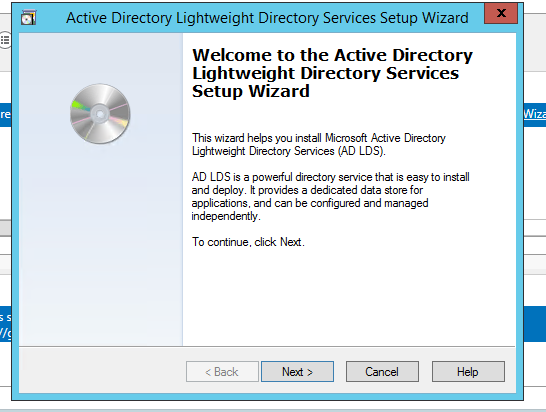


* Click Install.
* After AD LDS is installed, please follow the below steps to set up an active directory for Generate tool.
* From the Server Manager, click on AD LDS.

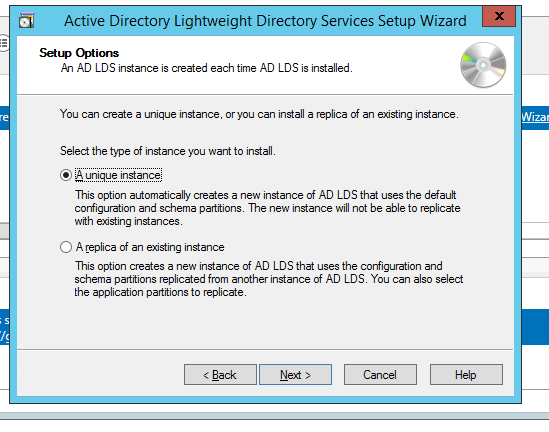


* Click on the “Configuration required” task and click on the Action link from the Task Window to open up the Active Directory Setup Wizard.

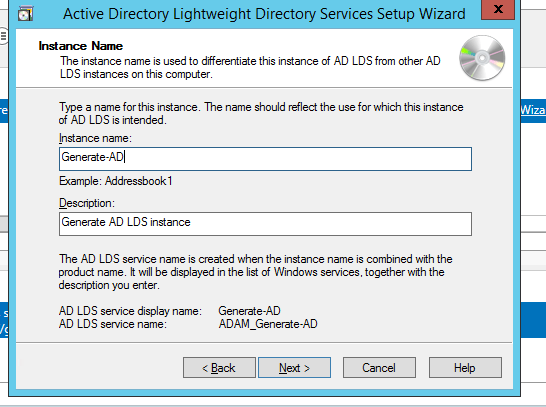


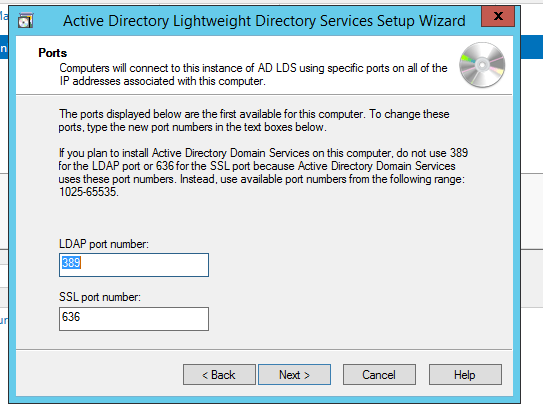


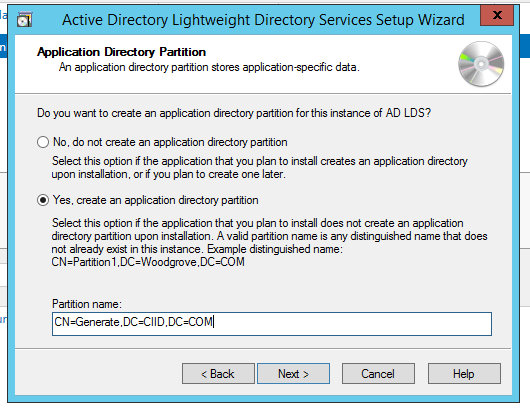
* Click Next.

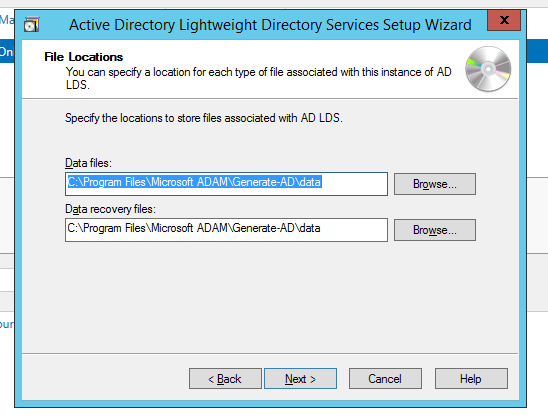


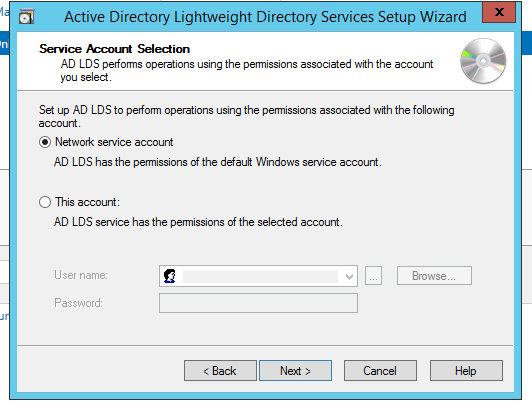
* Select “Unique Instance” option and click Next.

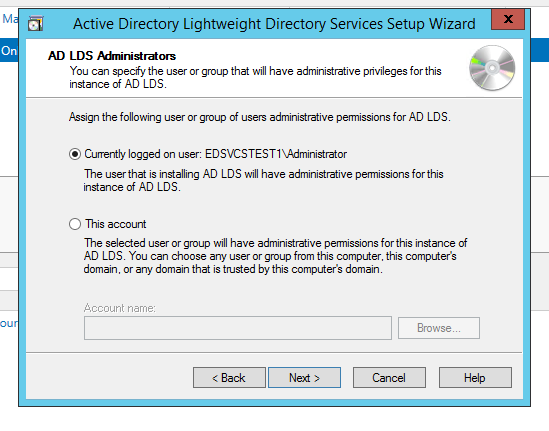


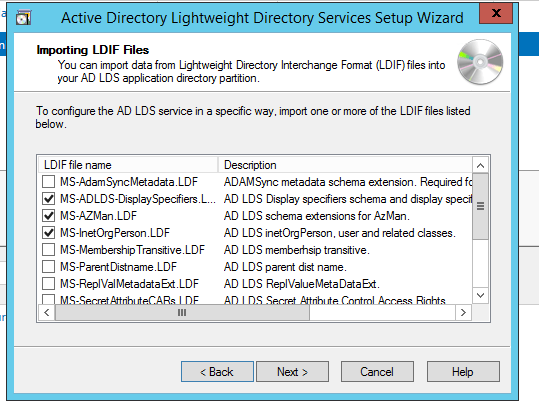


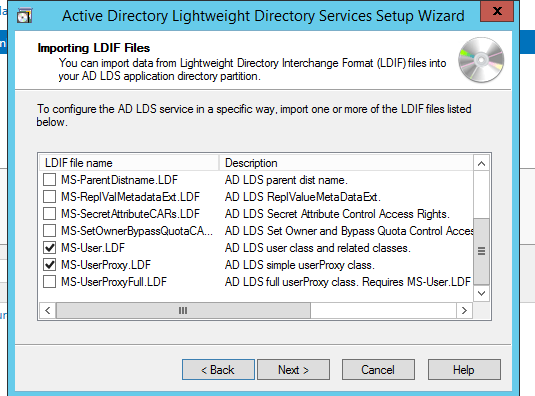


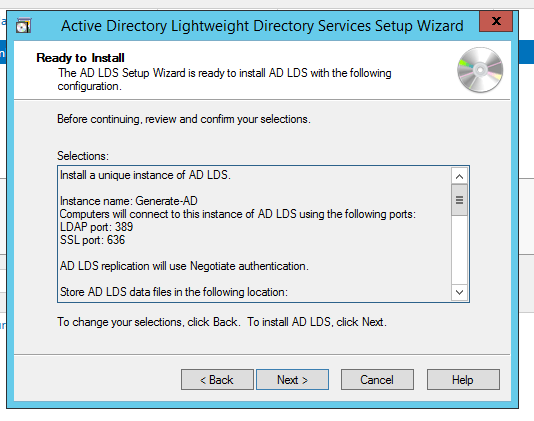


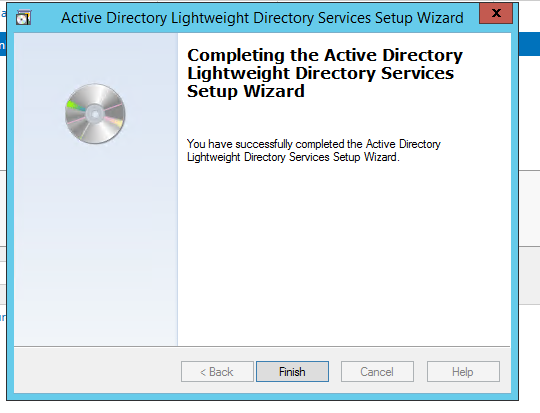








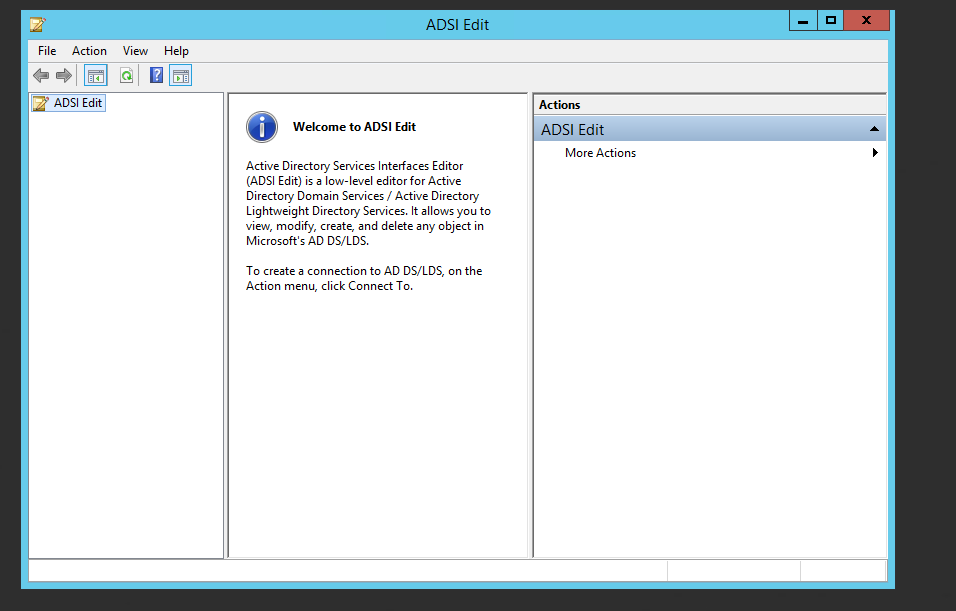


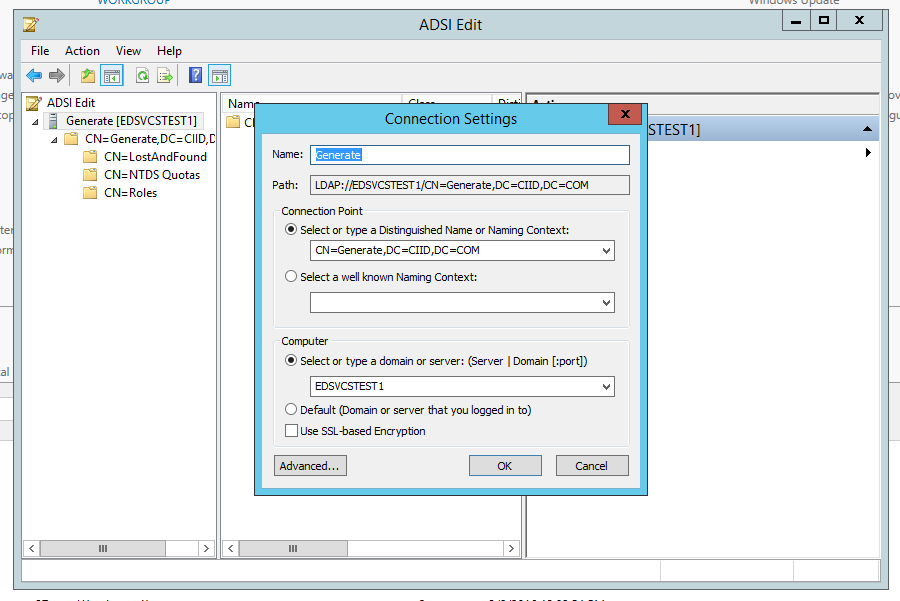


**Active Directory Service Interfaces (ADSI) Editor**

ADSI Edit is a tool used to connect to an active directory instance. It gets installed when AD LDS server role is installed.

Please use the following screenshots to connect to the AD LDS instance and set up the various roles and users for the Generate application.

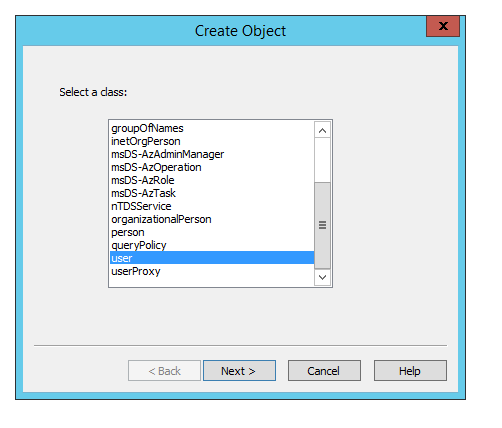




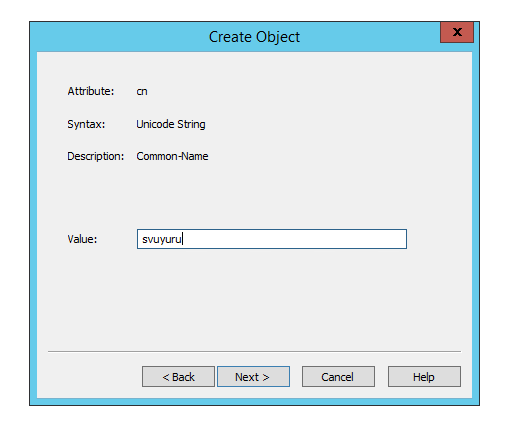
After connecting to the AD instance, a user should see 3 roles by default: Admin, Readers, and Users. A new “Reviewers” role will need to be added to the existing roles. The Generate tool has been configured to use Admin and Reviewers roles.

Please use the following steps to create a user for the generate tool in ADSI Edit.

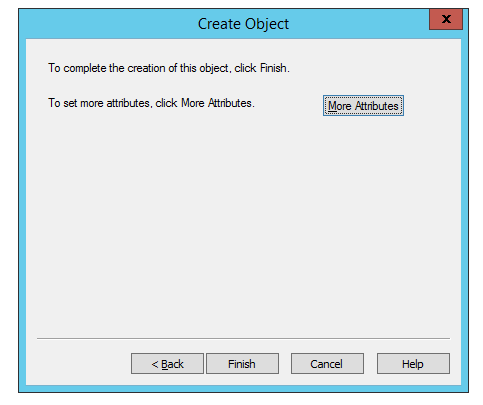
1. Right click on the Generate active directory in ADSI Edit and select New object.



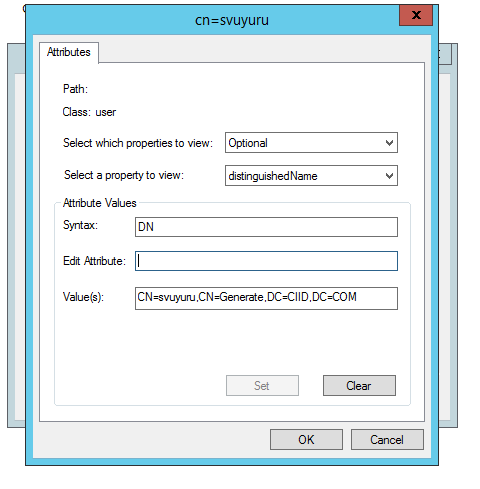
1. Click Next.



1. Enter the user name and click Next.



1. Click on the “More Attributes” button and set the following properties on the user object.



* distinguishedName: CN=svuyuru, CN=Generate, DC=CIID, DC=COM
* displayName : Sumanth Vuyuru
* givenName : Sumanth
* msDS-UserAccountDisabled : False
* userPrincipalName : svuyuru
* msDS-UserDontExpirePassword: True (Set this property if you don’t want to expire user’s password)
* ms-DS-UserPasswordNotRequired : True (Please reset the user password after the user has been created by right clicking on the user)

1. Go back and set “ms-DS-UserPasswordNotRequired” to False.

To add a user to a specific role, please add the user to member property of the role. Please use the following instructions.

1. Right click on the role and select Properties.
2. Go to member property and Edit it.
3. You should be able to add the user to that role using Distinguished Name (DN). Following is a sample DN that was added.

CN=svuyuru,CN=Generate,DC=CIID,DC=COM

