embarcadero

Product Documentation

ER/Studio Repository
Installation and Administration Guide

Version 6.7

Embarcadero Technologies, Inc. is a leading provider of award-winning tools for application developers and database professionals so they can design systems right, build them faster and run them better, regardless of their platform or programming language. Ninety of the Fortune 100 and an active community of more than three million users worldwide rely on Embarcadero products to increase productivity, reduce costs, simplify change management and compliance and accelerate innovation. The company's flagship tools include: Embarcadero® Change Manager™, CodeGear™ RAD Studio, DBArtisan®, Delphi®, ER/Studio®, JBuilder® and Rapid SQL®. Founded in 1993, Embarcadero is headquartered in San Francisco, with offices located around the world. Embarcadero is online at www.embarcadero.com.

September, 2014

CONTENTS

Welcome to ER/Studio Repository

ER/Studio Repository is a server-side model management system that solves the day-to-day challenges of modeling in a team environment, where model collaboration, versioning, security and component reuse are vital. The ER/Studio Repository allows multiple users to be extremely productive while collaborating on data and business process modeling projects with real-time concurrent access that allows team members to share and re-use assets across projects.

- Collaborative modeling with concurrent model and object access

- Model and object version management

- Standardized enterprise data dictionary

- Real-time objects status notification

About This Document

The ER/Studio Repository Installation and Administration Guide is the primary reference for the Repository. It provides both installation and administration information. In HTML Help format it serves as online help within ER/Studio DA. It is also distributed in PDF format for easy downloading and printing.

For more information, see the Embarcadero website at www.embarcadero.com.

Topics covered in this section include:

- [Installation](#)

- [Administration](#)

See Also

- [Upgrading From Previous Versions](#)

- [Repository Installation Checklist](#)

- [Installing the Repository](#)

Upgrading From Previous Versions

Use the following checklist to ensure ER/Studio Repository is correctly upgraded.

Upgrade Checklist

- Download ER/Studio Enterprise Upgrade. To download the most current release, go to embarcadero.com.

- Review the ER/Studio ReadMe. For the most current installation and usage information, see the ReadMe at embarcadero.com.

- Prepare to upgrade. For what to do before you upgrade to the latest version, see Preparing to Upgrade the Repository.

- In case you need to ER/Studio Team Server to match the features available in your version of ER/Studio Data Architect see ER/Studio Team Server Upgrade Notes.

Note: This procedure is for ER/Studio Enterprise installations only, which include the Repository. When you upgrade ER/Studio Data Architect, you must also upgrade ER/Studio Repository to correspond with the same build date as ER/Studio Data Architect.

Preparing to Upgrade the Repository

User preparations

1. Ensure you have backed up all your.dm1 files.

2. Check in all your diagrams and objects. If you try to check in a file from the previous version with the same name as a current file, the current file will become corrupted.

3. Rename all local copies of Repository diagrams in the active file directory as follows:

   1. Choose Repository > Options.

   2. Make note of the Active File Directory.

   3. Switch to the Windows Explorer and rename all the diagrams in the Active File Directory.

Administrator preparations

1. Ensure no one is logged into the Repository.

2. Backup the current Repository database.

3. Restart the Repository host computer.

Embarcadero Technologies 6

Note: If you are upgrading from a Repository version prior to 4.0, you can delete the C:\Program Files\Embarcadero\Repository\Data directory. This directory is not used with Repository versions 4.0 and later.

Upgrades from ER/Studio Repository 6.5/6.6 users to ER/Studio Repository 6.7

Keep the following in mind when updating for this release:

- Only a service upgrade is required.For this purpose:

1. Log on to Windows with local administrator privileges.

2. Start the *Repository* installation program and select *Repair Repository Services*. The installation program starts the following services:

3. Embarcadero Repository Communication Server

4. Embarcadero Repository Database Server

5. Embarcadero Repository Event and Dispatch Server

- No diagram data version upgrade is required. <u>User preparations</u> are not required.

- There are no database changes from version 6.5 or 6.6.

Upgrades from ER/Studio Repository 6

1. <u>Prepare</u> Repository to upgrade

2. Log on to Windows with local administrator privileges.

3. Start the *Repository* installation program and select *Repair Repository Services*. The installation program starts the following services:

4. Embarcadero Repository Communication Server

5. Embarcadero Repository Database Server

6. Embarcadero Repository Event and Dispatch Server

The Repository services and your existing database will be upgraded to the current version.

Upgrades from ER/Studio Repository prior to 6

The Repository upgrade must be done incrementally from one major version to the next until the target version is reached. You cannot skip major versions during the Repository upgrade as the installer for each major version only contains the database changes between the new version and the previous one.

When upgrading from earlier versions, such as from 5.5 to Repository 6.7, you must first upgrade the Repository to version 6.0, that will update the database, uninstall version 6.0, then upgrade to 6.7, and update the database.

Once the Repository upgrade is complete and you have installed the latest version of ER/Studio Data Architect, then you should upgrade your diagrams with the latest version of ER/Studio Data Architect.

Important Notes

- Always backup the Repository database before upgrading.

- You can install ER/Studio Data Architect 9.x and an earlier ER/Studio Data Architect version on the same workstation.

- Any *.dm1 files you use from the previous version will be copied and converted to the current version format when you open them for the first time. You can then save the diagram with a different name or use the same name to write over the old file. Previous versions of ER/Studio Data Architect may not read files generated on a higher major release.

- Upgrading diagrams can be time consuming and memory intensive. The number and size of ER/Studio Data Architect diagrams managed in the upgraded Repository dictates how long the upgrade process takes.

See Also

- [Release Notes](#)
- [Installing Embarcadero ER Studio Data Architect](#)
- [Licensing](#)

Repository Installation

Topics covered in this section include:

- [Repository Installation Checklist](#)
- [System Requirements](#)
- [Repository Database Server Requirements](#)
- [Planning for Repository Database Growth](#)
- [Preparing to Install ER/Studio Repository](#)
- [Installing the Repository](#)
- [Connecting to the Repository](#)
- [How to Specify Repository Ports](#)
- [Uninstalling ER/Studio Repository](#)

See Also

- [User Guide](#)
- [Repository Administration](#)

Repository Installation Checklist

Note: The following information is for installation only. If you are upgrading ER/Studio Repository, please see Upgrading From Previous Versions for important upgrade information.

The server-side Repository provides the ability for improved teamwork and enterprise collaboration. ER/Studio Repository is included in the Enterprise edition of ER/Studio. ER/Studio Repository is comprised of two components:

- **Repository Server**: The link between ER/Studio Data Architect and the Repository database. The server is responsible for creating the SQL code necessary to query the database to retrieve data or to insert data into the database. The Repository Server consists of the following components:

    o   Repository Communication Server

    o   Repository Event and Dispatch Server

    o   Repository Database Server

- **Repository Database**: Stores information about the Repository Diagrams and their objects.

Installation Checklist

Use the following checklist to ensure ER/Studio Repository is correctly installed and configured.

Done	N/A	Requirements
		Download ER/Studio Enterprise, which includes the Repository. You can download a 14-day trial version of ER/Studio Enterprise, which you can license later, from [www.embarcadero.com/downloads].
		Review the ER/Studio ReadMe. For the most current installation and usage information, see the ReadMe at [docs.embarcadero.com].
		Ensure system compatibility. For system requirements, see System Requirements.
		Optimize the repository database. For information on how to optimize Repository performance, see Preparing to Install ER/Studio Repository.
		Plan for repository growth. For information on how to estimate the required size of the Repository database, see Planning for Repository Database Growth.

		Install ER/Studio Repository For instructions on how to install the Repository, see [Installing the Repository](#).
		Connect to the Repository For instructions on how to connect to the Repository from ER/Studio Data Architect, see [Connecting to the Repository](#).
		Run and test ER/Studio Data Architect For instructions on how to use ER/Studio Data Architect, see the ER/Studio Data Architect Evaluation Guide at *[docs.embarcadero.com]*.

System Requirements

Review the following requirements before you install ER/Studio Repository. Adhering to these requirements optimizes ER/Studio Repository performance.These hardware requirements are based on a five to 10 users environment.

WARNING: If you are installing the x64 version of our software, you must uninstall any previously installed x32 bit versions

Tip: For more than 10 users, ensure your hardware surpasses the previously stated hardware requirements to accommodate the number of transactions that increase as the number of users increases.

Hardware	Repository Database Server	Repository Server
Processor		Dual core or higher
RAM	As required for your database platform	2 GB
Disk Space		50 MB + additional DB growth
DVD Reader		Depending on install medium
Operating System	As supported for your database platform	Any of the following: • Windows Server 2008 or 2012 (32-bit and *64-bit modes) • Windows 8/8.1 • Windows 7 • Windows Vista • Windows XP • Windows 2003 (32-bit mode) For 64-bit OS, the 32-bit database client libraries are required if you are using the 32-bit Repository version.
Software	Transaction Control Protocol (TCP) up and running Western European character set (UNICODE and UTF8 are not supported)	TCP up and running
Privileges	• DBMS privileges to create objects on the server • Local administrator privileges	

Supported DBMS (see [Repository Database Server Requirements](#))

See Also

- [Installing the Repository](#)

- [Repository Database Server Requirements](#)

Repository Database Server Requirements

The following table lists the databases the Repository supports. In order for the Repository to communicate with the database, the machine running the Repository must also have the corresponding RDMS client utility installed. The RDMS client utilities must also be installed on the machine running ER/Studio Data Architect if you want to use ER/Studio Data Architect to reverse engineer databases originating at any of these platforms.

ER/Studio Repository Database Server Requirements

Supported Repository DBMS	Corresponding Client Utility Software for 32-bit Repository	Corresponding Client Utility Software for 64-bit Repository
IBM DB2 Universal 9.x, 10.x Server	32-bit version of IBM DB2 UDB Client Utilities	64-bit version of IBM DB2 UDB Client Utilities
Microsoft SQL Server 2005 - 2014	32-bit client libraries are already installed on Windows machine	64-bit client libraries are already installed on Windows machine
Oracle 10g, 11g and 12c	32-bit version of Oracle Client Utilities	Both 32-bit and 64-bit version of Oracle Client Utilities are required

Oracle

- Configure your server in one of the Oracle configuration tools: Net Easy Config or Net Configuration Assistant.

- Create a connection alias.

Microsoft SQL Server

- If you are using Microsoft SQL Server and use case-sensitive nomenclature, you may get connection error messages from the Repository. To avoid this, create an empty database with a non-case-sensitive character set (such as Latin_General_CI_AS), then initialize the database with the Repository Database Maintenance utility.

IBM DB2

- Configure an ODBC data source entry for the IBM DB2 database that you want to access from ER/Studio Data Architect.

- Create a connection alias.

- For all versions of an IBM DB2 UDB server, to ensure optimal Repository performance, you must have a user temporary tablespace and the database must be tuned so it does not run out of memory or log space. For more information, see Preparing to Install ER/Studio Repository.

- For all versions of IBM DB2 UDB server, the client connection to the Repository database must be registered as a system ODBC Data Source. The data source name must match the Database Alias specified in the client connection.

Important Notes

- You must use ER/Studio Data Architect 9.x or later with ER/Studio Repository 6.x.

- The DBMS Client Utility software must be installed before installing the Repository server.

Note: If you install the ER/Studio Repository Server and the ER/Studio Repository database on the same machine, consider surpassing the hardware requirements described previously.

See Also

- [Backing Up and Recovering the Repository](#)

- [Repository Installation Checklist](#)

Planning for Repository Database Growth

Regardless of the platform, allocate at least 300 MB of space to the database and implement auto extend so that the database can accommodate increases in the size and number of models. Also, allocate 50 MB of space to the database log file.

The table below describes the initial size of some sample Repository databases before any diagrams are added, and the approximate database sizing for sample small, medium, and large diagrams.

RDBMS	Initial Size (MB)	Small Diagram (MB)	Medium Diagram (MB)	Large Diagram (MB)
Oracle 8	6	0.5	14	64
Microsoft SQL Server 7	7.8	0.7	14.7	40.9
IBM DB2 UDB	3.2	0.8	14.6	50

The table below describes the characteristics of small, medium, and large sample diagrams:

Sample Diagrams	Entities	Attributes	Views	Relationships	DM1 File Size (KB)
Small	22	125	2	20	184
Medium	302	2516	32	606	2153
Large	722	10629	0	1542	6967

Note: If you are installing ER/Studio Repository 6.x, install ER/Studio Data Architect 9.x or later. Previous versions of ER/Studio Data Architect are not compatible with ER/Studio Repository 6.x.

See Also

- [Preparing to Install ER/Studio Repository](#).

- [Installing the Repository](#).

Preparing to Install ER/Studio Repository

For Microsoft SQL Server

If you are using Microsoft SQL Server and case-sensitive nomenclature, you may get connection error messages from ER/Studio Repository. To remedy this, create an empty database with a non-case-sensitive character set (such as Latin1_General_CI_AS), then initialize the database with the Repository utility. Also, ensure you are using Mixed Authentication for server security.

For IBM DB2 for LUW

If ER/Studio Repository will be stored on IBM DB2 for LUW, perform the following procedures to ensure optimal Repository performance:

- Creating an IBM DB2 for LUW User Temporary Tablespace.

- Tuning the IBM DB2 for LUW Database.

Creating an IBM DB2 for LUW User Temporary Tablespace

The DB2 UDB database must have a user temporary tablespace with at least a 4 KB page size to create the temporary tables.

Tip: Use a system-managed user temporary tablespace because when managed by the system the user temporary tablespace can grow automatically.

Tip: You can use DBArtisan to create a system managed user temporary tablespace.

Create a user temporary tablespace using the tool of your choice or you can use the following SQL:

Note: Before you run this script, modify the path for the container.

```
CREATE USER TEMPORARY TABLESPACE REPOTEMP

PAGESIZE 4096

MANAGED BY SYSTEM

USING('D:\DB2\NODE0000\SQL00002\SQLT0004.0')

// here you specify the physical location of the table space

EXTENTSIZE 32

PREFETCHSIZE 16

BUFFERPOOL IBMDEFAULTBP

OVERHEAD 24.10

TRANSFERRATE 0.90
```

Tuning the IBM DB2 for LUW Database

This procedure tunes the DB2 database so that it does not run out of memory and log space.

1. Increase the application heap size and application control heap size to 2048 × 4KB. ===

   Tip: To increase the application heap size, right-click the database in the DB2 Control Center, and then click Configure. Then, switch to Performance and modify the values for Application Heap Size (applheapsz) and Application Control Heap Size (appctlheapsz).

2. Increase the log file space to 10,000 × 4 KB (or 40 MB).

3. Change the number of primary and secondary log files from 3 and 2, to 10 and 10. This provides for a total of 400 MB of primary log space, which should be sufficient for medium to large diagrams.

   Tip: To increase primary and secondary log file sizes, use the Logs tab of the Database Configuration page in the IBM Control Center.

See Also

- [Installing the Repository](#)

- [Connecting to the Repository](#)

Installing the Repository

Note: You need an account on the DBMS server Database with administrator privileges to create a new Repository database.

1. Log on to Windows with local administrator privileges.

2. Start the *Repository 6.x* installation program.

3. Walk through the installation wizard.

   The options are straightforward, but make sure you choose the following option:

   o **Create new Repository database or initialize an existing, empty database.**

Important Notes

The following are notes for parts of the install that may require additional explanation.

If you have previously created the database using the DBMS specific tool:

- **Database Platform**: Select Repository current database platform.

- **Server/Connect String**: This string depends on the database platform of the current Repository database.

  o SQL Server: Specify the server machine name on which the Repository database is hosted.

  o Oracle: Connection alias generated with Oracle client.

  o DB2: Connection alias generated with DB2 client.

- **Database name**: Enter the current Repository database name.

- **User Name**: User name of the user with administrator privileges on the current database.

- **Password**: Password of the user with administrator privileges on the current database.

- **Table Tablespace**: Enter current tablespace name. Just for Oracle and DB2.

- **Index Tablespace**: Enter current index tablespace name. Just for Oracle and DB2.

If you choose to create the database using the Repository Installation Wizard:

- **Automatic Growth for Microsoft SQL Server 7.0 or later**: To allow for automatic growth of the database file, click Allow Growth, and then enter a Growth Rate. Express the growth rate as either a percentage of the current database size or the number of MB to add to the database when the database is becoming full. Select the maximum size of the file as either Unlimited or click SIze, and then enter the maximum file size in MB.

- Use the **New Oracle User** dialog to create a new user and provide temporary tablespace information for the user. The dialog is self-explanatory, except for the following user identification options:

  o **Password**: If selected, indicates that Oracle should identify the user with the password provided.

  o **Externally**: If selected, indicates that Oracle should verify the database user name against an existing operating system user name.

  o **Globally**: If selected, indicates that Oracle will permit access to the user by obtaining user name and password information from the security domain central authority. This option is only available for Oracle 8.

- **Error Log Viewer**: The Error Log View appears if there is a problem creating your Repository database. If you get an error, review the Repository Database requirements to make sure you meet them. For further assistance with creating your Repository Database, contact Embarcadero Technologies Technical Support.

See Also

- [Connecting to the Repository](#)
- [How to Specify Repository Ports](#)

Migrating the Repository

To migrate your Repository installation to a new server:

1. Log on to the new server with local administrator privileges.

2. Start the *Repository x.x* installation program.

3. Proceed through the installation wizard.

   The options are straightforward, but make sure you choose the following option:

   o **Connect to an existing ER/Studio Repository database**

If you are just migrating the Repository database to a new server you do not need to run the install procedure:

1. In your current Repository server, select Start > All Programs > Embarcadero ERStudio Repository x.x > ERStudio Repository Database Maintenance.

   The options are straightforward, but make sure you choose the following option:

   o **Connect to an existing ER/Studio Repository database**

Notes:

The following are notes for parts of the install that may require additional explanation:

- **Database Platform**: Select Repository current database platform.

- **Server/Connect String**: This string depends on the database platform of the current Repository database.

  o SQL Server: Specify the server machine name on which the Repository database is hosted.

  o Oracle: Connection alias generated with Oracle client.

  o DB2: Connection alias generated with DB2 client.

- **Database name**: Enter the current Repository database name.

- **User Name**: User name of the user with administrator privileges on the current database.

- **Password**: Password of the user with administrator privileges on the current database.

- **Table Tablespace**: Enter current tablespace name. Just for Oracle and DB2.

- **Index Tablespace**: Enter current index tablespace name. Just for Oracle and DB2.

See Also

- [Connecting to the Repository](#)

- [How to Specify Repository Ports](#)

Connecting to the Repository

1. Start ER/Studio Data Architect or ER/Studio Viewer.

2. From the *Main* menu, choose *Repository > Repository Options*.

3. In the *Server Machine* field, type the host name or IP address of the Repository server machine, and then click *OK*.

4. Log on to the Repository as follows:

Choose Repository > Log In, enter your user name and password, and then click *OK*.

You are now ready to use ER/Studio Data Architect with ER/Studio Repository. For details on using ER/Studio Repository, see the ER/Studio Data Architect User Guide.

Note: During the Repository Installation, a default user name and password of 'Admin' is set as the default (without the quotes). The Administrator should change the Admin password after the initial log in.

See Also

- How to Specify Repository Ports

- Installing Embarcadero ER Studio Data Architect

How to Specify Repository Ports

To change the ports that the server listens to you will need to add the following registry keys on the machine that the Repository Application Server is running on.

Server Side

These are the registry keys that need to be set.

On 32-bit Operating System:
HKEY_LOCAL_MACHINE\SOFTWARE\Embarcadero\Repository\Database\RepoServer PortHKEY_LOCAL_MACHINE\SOFTWARE\Embarcadero\Repository\Database\RepoProbe Port

On 64-bit Operating System:
HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Embarcadero\Repository\Database\RepoServer PortHKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Embarcadero\Repository\Database\ RepoProbe Port

These are DWORD values and you just need to enter in the port number as a decimal value. You will need to restart the services after setting the values.

Default ports:

- Communication Port: 54333

- Identification Port: 54332

- SSL Port: 54331

Client Side

On the client side, in **ER/Studio Data Architect** you can use the menu **Repository > Repository Properties**. Change to the **Server** tab page and specify the ports that the client communicates to. The register value *RepoServer Port* mentioned before corresponds to the **Communication Port** listed on the **Server** tab. The register value *RepoProbe Port* corresponds to the **Identification Port** listed on the **Server** tab.

The identification port uses a UDP broadcast to identify running Repository servers that can be connected to in order to provide a list to the user (for example, the list in the **Repository > Options** dialog). This is optional and should not cause any issues if it is unavailable.

The communication port provides all communication between the ER/Studio client and the Repository server services, such as *login* requests, *get diagram* requests, *checkout/check-in* requests, *event updates*, and any other operations requiring information from the Repository. All communication is initiated by the client via TCP/IP. This port is required for use of the Repository, although it can be changed to a different port number if necessary.

Note: The SSL port works like the Communications port, except all traffic is encrypted via SSL. The Repository listens on both the SSL port and the Communication port for connections. If a user chooses to use SSL in ER/Studio, then the connection will be made to the SSL port on

the server rather than the Communication port for all repository operations. To enable SSL and configure the port you can use menu **Repository > Repository Properties**.

The registry key that stores this port number can be found on the client machine is *RepoSSL Port* and is under: HKEY_CURRENT_USER\Software\Embarcadero\ER/Studio Data Architect x.x\Repository

Where x.x is the version of ER/Studio Data Architect.

See Also

- [Setting Repository and ER/Studio Team Server Options](#)

- [Working with ER/Studio Repository](#)

Uninstalling ER/Studio Repository

From the Windows Add or Remove Programs Control Panel, select Embarcadero Repository X.X, click **Remove**, and then follow the prompts to remove all ER/Studio Repository components.

Note: This does not remove the Repository database. The database administrator must remove the Repository database.

See Also

- Uninstalling ER/Studio Data Architect
- Installing Embarcadero ER Studio Data Architect

Repository Administration

This section is for the person who installed ER/Studio Data Architect and who is now configuring ER/Studio DA or administering ER/Studio Repository databases. It contains the following topics:

- [Repository Architecture and Design](#)
- [Managing Repository Files](#)
- [Establishing Security for the Repository](#)
  - o [Creating and Managing Roles](#)
  - o [Creating and Managing Users](#)
  - o [Creating and Managing Groups](#)

See Also

- [Upgrading From Previous Versions](#)
- [Repository Architecture and Design](#)
- [Managing Repository Files](#)
- [Establishing Security for the Repository](#)

Repository Architecture and Design
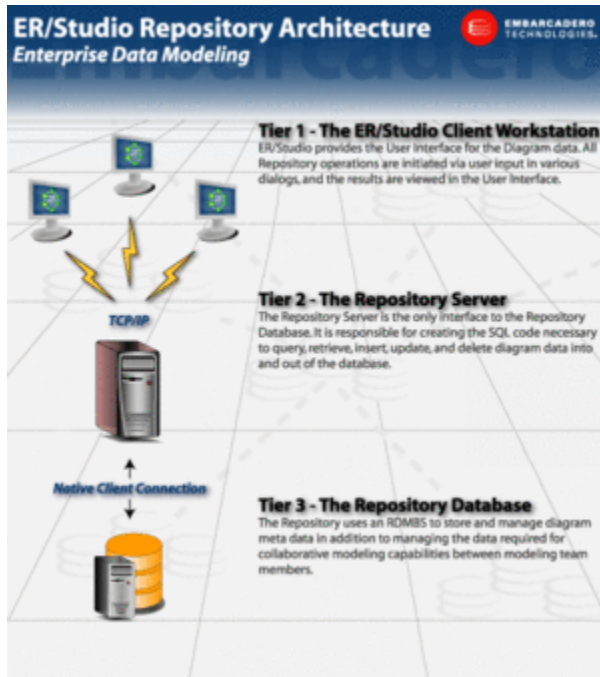
The Repository relies on a native client connection and the TCP/IP protocol to send and receive information between ER/Studio DA, the Repository Server, and the Repository Database. Communications between the client and server are secure. SSL support in the client and repository allows you to specify the common URL and port to support secure communications between the client and the repository.

There is a metadata diagram of the Repository schema in the System Models directory (Repository300MetaModel.dm1). It includes relationships between entities as documentation whereas the actual database, for performance reasons, does not include FK constraints. The file shows the propagation chain which could be helpful for writing queries to run against the Repository.

For a complete list of software and installation requirements, and supported Repository Databases, see the Repository Database Server Requirements.

Repository Client

ER/Studio Data Architect provides the user interface for the Diagram data stored in the Repository. The ER/Studio Repository menu gives you access to all Repository functions and features. To access and use the Repository, you must have ER/Studio DA running.

You can access Repository metadata, search and create reports through ER/Studio Team Server.

Repository Server

The Repository Server links ER/Studio DA to the Repository Database. The Repository Server contains the SQL code necessary to query the database to retrieve data or to insert data into the database. The Repository Server runs this code when it receives a request to execute an operation that requires it to read data in the database and return the appropriate information and when it receives a request to execute an operation that requires it to write new data to the database.

Note: Because the Repository Server is responsible for inserting data into the database, and retrieving data from the database, you can only have one Repository Server for each Repository Database.

The Repository Server manages the state of the Repository data, which it stores in the Repository database. It records modeling change operations as requested by ER/Studio Data Architect users.

When the server receives a request, it does the following:

1. Checks to make sure the user has the necessary permissions to perform the operation.

2. If the operation requests data from the Repository database, the server retrieves the data from the database and sends it to the ER/Studio Data Architect instance from which the request originated.

3. If the operation attempts to store data in the Repository database, the Repository Server writes the data out to the database.

4. Notifies all ER/Studio Data Architect "clients" of the new state of the Repository and sends them the appropriate information to update the displays.

See Also

- Managing Repository Files

- Repository Database

Managing Repository Files

The Repository Server installation includes several folders, described below. Depending on your configuration options, particularly if the *-deleteoff* option is set on your server, you may want to manually remove older files collecting in your Repo In/Out folders as described here.

It is always safe to delete a **.pri** file or **.pro** file. It is also safe to delete any files with old dates since the files are generally processed immediately upon being written to the directory. This applies to **.out** files and **.err** files. Files with **.in** extension should not be deleted and should not be in the folders for more than the time it takes to process them, unless the Repository Services have been stopped.

Folder	Description
Repository	This folder contains the Repository subfolders, listed below. It also contains the executable files for the three Repository services, along with the Repository Server Licensing application, a copy of the license agreement, and a copy of a compiled help file that contains information about installing and licensing the Repository.
Logs	This is a subdirectory of the Repository folder. It is used to store log files that are used by each of the Repository services to store information about events or transactions.
RepoIn	This is a subdirectory of the Repository folder. It stores the request files received from an ER/Studio DA client until they are processed by the server. If you have older files collecting here: 1. Open the directory in the Windows Explorer. 2. Sort the files by Date Modified with the oldest files at the top. 3. Select all files starting from the top and stopping wherever you feel safe. A good idea is to select all files modified before the current day. 4. Delete these files.
RepoOut	This is a subdirectory of the Repository folder. It stores files being sent from the Repository server to the ER/Studio DA client until they are processed.
Utilities	This is a subdirectory of the Repository folder. It contains SQL scripts to run reports or to perform maintenance actions. The SQL Agent is used to run these scripts.

When installing the Repository, you can install the server and database on the same machine, or on different machines. Regardless of where you install each component, we recommend that you do a full backup of both components on a regular basis.

For more information about Repository maintenance please see:

- [Repository Database](#)

- [Moving the Repository](#)

- [Optimizing Repository Performance](#)

- [Contingency Planning](#)
- [Backing Up and Recovering the Repository](#)
- [Backup Copies](#)
- [Querying the Repository](#)

See Also

- [Best Practices](#)
- [Establishing Security for the Repository](#)

Repository Database

The Repository uses a database to store information about the Repository Diagrams and their objects. The database is updated each time a user checks out, checks in, adds, or deletes any element of a Diagram. In addition, the Repository database stores the history of certain Repository operations, as well as all security rules and data.

> ⚠️ **WARNING:** Do not directly update Repository tables because the database is a transactional system designed to handle collaborative model use. Updating the Repository database outside of ER/Studio DA can corrupt the data. The SQL scripts in the Embarcadero\Repository\Utilities\MasterScripts and SQLScripts directories are used by the Repository within ER/Studio DA and should be used outside ER/Studio DA only at the request and under the guidance of Embarcadero Support.

Note: You can query the data and select data from the Repository for reporting purposes. Commonly used SQL scripts in the Embarcadero\Repository\Utilities\QueryScripts and Embarcadero\ERStudio Data Architect X.X\SQLCode\Repo_Queries directories have been provided for your convenience. However, you can create your own scripts and to help you do this, a meta model is provided in the...\Program Files\Embarcadero\ERStudio Data Architect X.X\System Models directory.

See Also

- [Best Practices](#)
- [Moving the Repository](#)
- [Backup Copies](#)

Moving the Repository

The following describes how you can move the Repository server and database to a new machine or to a new database platform.

1. Backup the current Repository database or export the current Repository database schema, depending on the DBMS platform.

2. Restore the database on the new Repository database server.

3. Install the Repository application server on a new machine if necessary.

4. Run the program "ERStudio Repository Database Maintenance" on the machine where the new Repository application server is installed.

5. Choose the option "Connect to an existing ER/Studio Repository database".

6. Specify the connection information for the new database.

7. If necessary, provide the new hostname to Repository users so they can connect to the new Repository.

See Also

- [Best Practices](#)
- [Backing Up and Recovering the Repository](#)

Optimizing Repository Performance

The following tips can help optimize the performance of the Repository:

- **Dedicated Server**: The performance of the Repository is directly affected by the number and size of diagrams and the size of the data transfers between the client and the Repository. We recommend using a dedicated server for the Repository, unless you're using industrial-strength hardware such as multi-processor rack servers that can manage many applications simultaneously.

- **Separate the Repository Server from the Repository Database:** The Repository Server and Repository Database do not have to reside on the same machine, however such a configuration may reduce wait times due to reduced contention. If you decide to separate the Repository Server and the Repository Database, for optimal performance, the Server should reside in the same physical local as the users.

- **Clearing Repository Folders**: On the Repository Server machine there are *RepoIn* and *RepoOut* folders in the \...\Embarcadero\Repository\ folder. Sometimes network problems cause entries to be created in these folders. Many files in these folders can hinder Repository performance. Periodically purging the files in these folders should be done frequently, such as nightly, when no one is connected to the repository.

- **Anti-Virus**: Disabling scanning of Repository files and folders will prevent the anti-virus software from interfering with Repository operations. Since the Repository writes and deletes files in the RepoIn and RepoOut folders in the \...\Embarcadero\Repository\ folder, it is recommended to disable any anti-virus scanning of those folders on the server machine. The Repository also writes temporary files with the extensions.IN and.CNL to the installation folder and on older versions these were written to Windows\System32. It is recommended to disable scanning of files with those extensions in that folder.

- **For Oracle environments:**

  - **Separate Tablespaces**: Create separate tablespaces for Repository tables, *ERSTUDIO_DAT* and indexes, *ERSTUDIO_IDX*. The extent of each of these tablespaces should be 1 MB or larger. In some very large Repository environment it may be beneficial to further separate the larger tables and their indexes creating, for example *ERSTUDIO_DAT2* and *ERSTUDIO_IDX2* in addition to the original tablespaces.

  - **Rebuild Indexes**: The Repository index, *ERSTUDIO_IDX* should be rebuilt at least once a month. Depending on how many diagrams are added, the index may need to be rebuilt on a weekly basis.

  - **Analyze and Compute Statistics**: The response of queries to the Repository can be improved by running DBMS_STATS for recent Oracle version or the Analyze Table and Compute Statistics statements for older versions. These commands gather information about data distribution within the tables and indexes and update the data dictionary with this information. This enables the cost-based query optimizer to make more intelligent decisions about how to efficiently process SQL statements that access the tables or indexes.

- o **Oracle Logs**: Check In operations can cause checkpointing to occur which can interrupt other Repository actions to the detriment of response time. To minimize the effect checkpoint has, increase the size of the logs, set the *log_checkpoint_interva* l parameter in *init.ora* to zero, and reduce the size of the log buffer.

See Also

- Best Practices

- Backing Up and Recovering the Repository

Contingency Planning

To ensure optimal Repository performance and availability, backup the Repository schema frequently and designate another server as the backup server in case of hardware failure. Then, if necessary, you can install a new set of Repository services onto the backup server and point the Repository to a freshly restored backup.

See Also

- [Best Practices](#)

- [Backing Up and Recovering the Repository](#)

Backing Up and Recovering the Repository

You should implement a backup and recovery process for your Repository Database. Repository stores critical data in the Repository Server Data folder.

When you add your locally created and managed *.DM1 files to the Repository server, the file is managed in the Repository Database (existing within Oracle, SQL Server or DB2 tables). You can find these files in the ER/Studio Repository installation path. If you accepted the installation default, the files are located in:

- ...\Embarcadero\ERStudioX.X\Repository\Data

Note: ER/Studio Data Architect no longer uses overflow files. Long files are now added to the Repository Database.

The Repository Server is the only interface to the Repository database. It is responsible for managing the state of the Repository data. The Repository Server is basically a transaction server for the database. The Repository database also stores the transaction history as well as all security rules and data. Because the Repository server is responsible for data transactions with the database there can only be one Repository server for each Repository database.

See Also

- Best Practices
- Backup Copies

Backup Copies

In some cases it makes sense to keep a backup copy of a dm1 file. However, if you perform Repository operations on more than one copy of the file at a time it could lead to check in failures. The following best practices are for situations where it is necessary to keep a backup copy of a.dm1 file:

- Create a folder on your hard drive specifically for dm1 file backups. For example, add a folder named *backup* under your Active Directory folder.

- When making a backup copy, use Windows Explorer to move or copy the current file into the backup directory. If you plan on doing a "clean" get to retrieve the entire diagram from the Repository, use move so that your local diagram is not in the Active Files directory.

- If you need to merge changes from a backup into the current copy of the diagram, make sure no Repository operations are done on the backup copy. Complete the merge and close the backup copy before beginning the check in procedure.

See Also

- [Best Practices](#)
- [Backing Up and Recovering the Repository](#)

Querying the Repository

You can query your Repository when you want to run some simple SELECT reports. Below is an example of a simple cross-diagram report that selects a Diagram, Sub Models in the Diagram, and Entities in each Sub Model.

Note: Do not use the Repository for editing, deleting, or updating data. ER/Studio DA is the only application you should use for these operations.

Example

We created this example using ER/Studio Data Architect's Repository meta model. This data model is in the installed application directory:...\Program Files\Embarcadero\ERStudio Data Architect X.X\SystemModels\Repository450MetaModel.dm1

```
SELECT dbo.DiagramVer.Name as "Diagram Name",

dbo.SubModelVer.Name as "Submodel Name",

dbo.EntityVer.Name as "Entity Name",

dbo.EntityVer.TableName as "Table Name"

FROM dbo.Diagram, dbo.DiagramVer, dbo.Entity, dbo.EntityVer, dbo.Model,
dbo.SubModelVer, dbo.SubModel

WHERE (dbo.Entity.LatestVersionID = dbo.EntityVer.EntityVerID AND

dbo.Model.ModelID = dbo.Entity.ModelID AND

dbo.SubModel.LatestVersionID = dbo.SubModelVer.SubModelVerID AND

dbo.Model.ModelID = dbo.SubModel.ModelID AND

dbo.DiagramVer.DiagramID = dbo.Model.DiagramID AND

dbo.Diagram.LatestVersionID = dbo.DiagramVer.DiagramVerID)
```

See Also

- [Best Practices](#)

- [Backing Up and Recovering the Repository](#)

Establishing Security for the Repository

Security of the Repository contents is managed by the Repository Administrator through the Security Center by creating and then assigning users particular Groups, Roles and Privileges.

Note: To prevent an accidental lockout, the Admin user cannot be deleted.

For example, we create a user JimB, give him a Role and call it DBA. The DBA role has many privileges, but none allow the DBA to modify a logical model. A Repository administrator can associate JimB with any model in the Repository and he will not be able to modify the diagram's logical model, but he can view it.

The Administrator can restrict user and group access to Repository items at the project, diagram, model, submodel, and data dictionary levels. To restrict access to the Repository, the Administrator creates users and then assigns the users a default role or assign, which grants specific permissions to the user, or creates new roles to assign to users. The Administrator can alternatively create a group of users and assign a role to that particular group. In these ways, the Administrator can grant users full access to some Repository items and restrict or deny access to others items.

Before creating users, groups,and roles, the Administrator should understand the following Repository concepts:

- **Cascading Security**: To make it easy for Administrators to assign a role globally to users when there are many diagrams in the Repository, such as when the user must have the same access permissions to many of diagrams, the Administrator can assign a user and role to various 'levels' of the ER/Studio Repository.

For example, a Project can be created in the ER/Studio Repository that contains many diagrams. If you assign a user a role to this Project, any diagrams contained within the Project will be granted the same permission set. The Repository itself can act as this highest-level point to assign permissions that are cascaded down.

Higher levels can be overwritten by assigning the same user different roles at a lower level, for example UserA may have the Viewer role at repository level but Modeler role for a specific diagram or submodel.

- **Client Side Security Caching**: To promote the concept of detached modeling collaboration (e.g. being logged out of ER/Studio Repository and possibly detached from the network), all security associated with the user by diagram is cached on the client when the user logs into ER/Studio Repository.

Note: When security changes are made, users must re-login to update their permissions.

- **Super User**: Super User is a default Role created upon installation of ER/Studio Repository. The default 'Admin' user is assigned to the Super User role. This user can do anything to diagrams, users and roles managed in the ER/Studio Repository as it is granted all privileges. It is not removable or editable, and is only found at the 'Repository' level of the Repository Security area of the Security Center.

- **No Access**: *No Access* is a role that can be applied only at the project and diagram levels of the Repository. It is a quick and global mechanism to prevent any access whatsoever to diagrams managed in specific projects, or to individual diagrams themselves. For example, you can create a group and assign it to the no access role for a project, then any users added to the group will not have access.

- **Additive permissions**: Permission granted to a user by group membership are additive. The user will have all the permission granted to him plus the permissions afforded him by group membership.

Notes

- When a user gets a file from the Repository, ER/Studio Data Architect tracks the user and machine combination for that particular file. In addition, to enable users to model while they are "offline" or unable to connect to the Repository server, ER/Studio Data Architect stores information about the security rights of the user and the diagram with the file, which has the following effects:

- When a user is logged into the Repository, and attempts to open a DM1 file that was retrieved from the Repository by a different user/machine combination than the currently logged in user, ER/Studio Data Architect indicates that the file cannot be opened due to the conflict. This ensures that one user does not attempt to work on or check in changes to a diagram that was originally retrieved by a different user. Because the Repository keeps track of object check outs based on the user and machine, only the user on the machine that originally checked out an object can check it back in.

- Even if a user is not logged into the Repository, Repository DM1 files can be opened and worked on in ER/Studio Data Architect. ER/Studio Data Architect will load the security permission data for the user that originally retrieved the file from the Repository, and that security data will govern the rest of the working session until the user either logs in to the Repository or closes the file. If multiple files are open while the user is NOT logged in to the Repository they must all have been retrieved by the same user. Of course, this also applies when the user IS logged in.

- The cached security data stored with each diagram is updated whenever an open Repository file is saved while logged in to the Repository, but if the file is not saved before closing it, then any changes to the security permissions for that user and diagram will not be cached with the file. So, if you plan to work offline, and the security settings have changed since the file was last saved, then you should open each file while logged into the Repository and save it before taking the files to work offline.

- Permission settings can be applied at the project or repository level, which might cause a conflict when two files are opened while a user is offline. If two files are opened that both have cached security information for the Repository level or for the same Project, and if the cached data differs, then the most recently saved data will be used and stored in both files if and when they are saved.

- If an admin makes changes to the security UI, such as changing permissions on folders to grant or revoke access or moving diagrams between projects with different permissions, users must re-login to update their permissions.

This section also includes the following:

- [Creating and Managing Roles](#)
- [Creating and Managing Users](#)
- [Creating and Managing Groups](#)
- [Granting and Prohibiting User Access to Repository Items](#)

See Also

- [Creating and Managing Roles](#)
- [Creating and Managing Users](#)
- [Creating and Managing Groups](#)
- [Granting and Prohibiting User Access to Repository Items](#)

Creating and Managing Roles

ER/Studio Data Architect lets you create customized roles with different sets of permissions. Repository Object Type Permissions are pre-defined privileges to operate on Repository items. You can assign them to a role, which gives users assigned to that role, permission to perform certain Repository operations. You can use the many available privileges to create specific roles to suit your environment.

For example, you can create and assign some roles like the following:

- **Repo Level Basic**, which has all of the Repository level permissions, assigned to the target user at the Repository level.

- **Logical Only Modeling**, which has all of the Diagram, Model, and Submodel level privileges that do not apply to the physical model, assigned to the target user at the diagram level for each diagram in the Repository or at least the ones that the target user is allowed to work on.

With the above roles and assignment, when the target user adds the diagram, the user can check it out, but cannot modify it. To allow the user modify the logical model immediately after adding it, the Admin would have to apply the "Logical Only Modeling" role to the target user at the Repository level.

- To access, create, update, and delete user information, a Repository administrator must have Repository Object Type Permission, Access Security Info and Update Security Info privileges.

Create, Update, and Delete Roles

1. Choose Repository > Security > Security Center.

2. Select the *Manage Roles* tab.

3. Click *New*.

4. Click an item in the *Repository Object Type* list and then in the *Repository Object Type Permission* area, assign the permissions you want to assign to the role for the selected object type.

5. Repeat step 4 for each *Repository Object Type*.

6. Click *Apply* and continue changing security settings and then when finished, click *OK* to exit the security center.

7. To apply the new role to existing users, see <u>Assigning a Role to a User/Group</u>.

8. Inform affected users that they should log out of the Repository and then log in again to receive the security updates.

Notes

- Once created, the Administrator can select the role and update, delete or rename it at any time. Renaming the role does not affect the privileges users have, but if you delete a

role users who had access to Repository items through that role, will no longer have access to those items.

- Users cannot modify objects without the necessary permissions, regardless of whether the user has the objects checked out.

- To facilitate immediate access to a newly added diagram (not projects) the Admin should set up privileges as follows: If a user gets a submodel and wants to add an entity, the user must have Create Entity permission in the model containing the submodel, as well as Add Member permission in that submodel. If the user deletes an entity, the user must have Remove Member permission in that submodel. In addition, if the user wants to select the "Delete From Model" check box, the user must have Delete Diagram Object permission in that model.

- Once you delete a role from the Repository, it will no longer be in the database.

- To access/create/update/delete user information, a Repository administrator needs to have the Access Security Info and Update Security Info Privileges applied to the Roles.

- Before you delete a role, you must unlink any diagrams that are assigned to it and delete any users assigned to it.

- The following lists the Repository Object Types, the permissions you can grant, and the common operations these permissions give access to:

Repository Object Types	Repository Object Type Permissions	Permitted Operations
Repository	Access Security Info	*Security Center*: View settings
	Update Security Info	*Security Center*: View and change settings
	Create Diagram	Add Diagram
	Update Diagram	Check Out Diagram, Check In Diagram Check Out Object(s), Check In Object(s) Undo Check Out Diagram, Undo Check Out Object(s) Redo Check Out Diagram, Redo Check Out Object(s)
	Delete Diagram	Delete Diagram If you delete a diagram in Repository, the file itself remains on the local disk.
	Create Enterprise Dictionary	Create Enterprise Dictionary
	Update Dictionary	Check Out Data Dictionary, Check Out Dictionary Object(s), Check In Data Dictionary, Undo Check Out Data Dictionary, Redo Check Out Data Dictionary

	Create Project	Create Project
	Delete Project:	Delete Project
Project	Add Project Member	Add Diagram to Project
	Remove Project Member	Remove Diagram from Project
Diagram	Bind Enterprise Dictionary	Create New Enterprise Data Dictionary, Bind Existing Enterprise Data Dictionary
	UnBind Enterprise Dictionary	Remove Enterprise Data Dictionary
	Compare Models	Run Compare/Merge Wizard
	Create Physical Model	Create Physical Model
	Delete Physical Model	Delete Model
	Set Named Release	Set Named Release
	Delete Named Release	Delete Named Release
	Rollback Diagram To Named Release	Rollback Diagram
	Update Diagram Properties	Edit Title Block Data, Edit Diagram Properties
	Create Data Flow	Create New Data Flow on the Data Lineage tab
	Delete Data Flow	Delete Data Flow from the Data Lineage tab

Data Dictionary	Create Dictionary Object	*Create*: Attachment Type, Attachment, Default, Rule, Data Movement Rules, Reference Value, User Datatype, Domain Folder, Domain, Reusable Trigger, Reusable Procedure, Library
	Update Dictionary Object	*Edit*: Attachment Type, Attachment, Default, Rule, Data Movement Rules, Reference Value, User Datatype, Domain Folder, Domain, Reusable Trigger, Reusable Procedure, Library
	Delete Dictionary Object	*Delete*: Attachment Type, Attachment, Default, Rule, Data Movement Rules, Reference Value, User Datatype, Domain Folder, Domain, Reusable Trigger, Reusable Procedure, Library
Logical Main Model	Create Diagram Object	*Create*: Entity, View, Relationship, View Relationship, Subtype Cluster, Subtype, Title Block
	Delete Diagram Object	*Delete*: Entity from Model, View from Model, Relationship from Model, View Relationship from Model, Subtype Cluster from Model, Subtype from Model, Title Block from Model
	Update Diagram Object	*Entity Editor*: Create/Modify/Delete: Attribute, Key, Key Attribute, Check Constraint *View Editor*: Modify View; Create/Modify/Delete: View Table, View Column *Key Editor*: Modify Key, Create/Modify/Delete Key Attribute *Relationship Editor*: Modify Relationship *Subtype Cluster Editor*: Modify Subtype Cluster *Edit Model*: Options, Properties
	Create Submodel	Create Submodel
	Delete Submodel	Delete Submodel
Logical SubModel	Add Member	*Submodel Editor*: Add to Submodel
	Remove Member	*Submodel Editor*: Remove from Submodel, Delete Entity from SubmodelRemove Database View *Delete*: Relationship from Submodel, View Relationship from Submodel, Subtype Cluster from Submodel, Subtype from Submodel
	Update Display Properties	*Move/Resize*: Entity/Table, View, Title Block, Text Block, Subtype Cluster *Color/Font Changes*: Entity/Table, View, Title Block, Text Block, Relationship Line, View Relationship Line, Subtype Cluster *Move*: Relationship Line, View Relationship Line *Create/Modify/Delete*: Text BlockChange Model Notation, Perform Layout, Zoom, Align Objects
Physical Model	Create Diagram Object	*Create*: Table, View, Relationship, View Relationship, Schema Object, Title Block
	Delete Diagram Object	*Delete*: Table, View, Relationship, View Relationship, Schema Object, Title Block

	Update Diagram Object	*Create/Modify/Delete*: Column, Index, Index Column, Check Constraint, View Table, View Column, Key Attribute, Key, Relationship, Subtype Cluster *Edit Model*: Options, PropertiesChange Database Platform
	Create Submodel	Create Submodel
	Delete Submodel	Delete Submodel
Physical SubModel	Add Member	*Submodel Editor*: Add to Submodel
	Remove Member	*Submodel Editor*: Remove from Submodel *Delete*: Table from Submodel, View from Submodel, Relationship from Submodel, View Relationship from Submodel, Schema Object from Submodel
	Update Display Properties	*Move/Resize*: Entity/Table, View, Title Block, Text Block, Physical Schema Object *Color/Font Changes*: Entity/Table, View, Title Block, Text Block, Relationship Line, Physical Schema Object *Move*: Relationship Line, View Relationship Line *Create/Modify/Delete*: Text BlockChange Notation, Perform Layout, Zoom, Align Objects
Data Flow Model	Create Data Flow Object	*Create*: Data Flow, Data Lineage Component, Data Stream, Source
	Update Data Flow Object	*Edit* Data Flow, Object, Transformation, Data Stream, Source *Check In*: Data Flow, Object, Transformation, Data Stream, Source *Check Out*: Data Flow, Object, Transformation, Data Stream, Source *Undo Check Out*: Data Flow, Object, Transformation, Data Stream, Source *Redo Check Out*: Data Flow, Object, Transformation, Data Stream, Source *Note*: The user must have permission to update the diagram in order to update data flow objects in the diagram.
	Delete Data Flow Object	*Delete*: Data Flow, Data Lineage Component, Transformation, Data Stream *Note*: Deleting a table\entity component from the Data Lineage window does not delete the table\entity from the model. *Note*: The user must have permission to delete the diagram in order to update data flow objects in the diagram.
Data Flow Display	Update Display Properties	*Move/Resize*: Transformation, Component, Data Flow *Color/Font Changes*: Data Lineage Background, Component, Transformation, Data Stream *Diagram And Object Display Options*: Change any option in this dialog to control how the data lineage diagram displays.Perform Layout, Zoom, Align Objects, Layout Data Stream, Straighten Data Stream, Remove All Bends

Assigning a Role to a User/Group

Assigning a role to a user/group grants the user/group all the permissions selected in the role.

1. Choose Repository > Security > Security Center.

2. Select a Repository item you want to assign permissions to.

   o **If the user/group is not currently assigned a role**, on the **Repository Security** tab, click the user name in the **Available Users** column and drag it onto a role name in the **Available Roles** column.

   o **If the user/group was already assigned a role**, on the **Repository Security** tab, click the user/group name the **Available Roles** column and drag it onto another role name in the **Available Roles** column.

3. Click **Apply** and then click *OK* to exit the **Security Center**.

4. Notify users/group members that they must log out and then log in again to receive the security updates.

See Also

- [Creating and Managing Users](#)
- [Creating and Managing Groups](#)

Creating and Managing Users

Managing users entails creating, changing, and deleting user account permissions and logins, checking in a users checked out documents under unusual circumstances. You can control the privileges granted to a user for a specific diagram, model, submodel or Enterprise Data Dictionary by associating the user with a role. Using Roles, you can choose the permissions granted a user for the Repository item.

- To access, create, update, and delete user information, a Repository administrator must have Repository Object Type Permission, Access Security Info and Update Security Info privileges.

Create, Edit, Delete, Deactivate, Reactivate, Log Out, and Check In Users

1. Choose Repository > Security > Security Center.

2. Click the **Manage Users** tab.

3. On the **Manage Users** tab, you can manage users by clicking their name in the list and then clicking a management option and following the prompts as required.

4. Continue making security changes and then click **OK** to exit the security center.

5. To assign permissions to users, see Assigning a Role to a User/Group.

6. Notify users that they must log out and then log in again to receive the security updates.

The following helps you complete the options of the *Manage Users* tab:

User Management area

Administrator User 'Force' Options area

Notes

User Management area

- **New**: Click **New** to add a new user through the **Create Repository User** dialog.

- **Directory Service User**: Select this option If you want the new user to login to the Repository using their Windows credentials. In this case ER/Studio Data Architect accesses the LDAP server on the network to verify the user's credentials. Directory service users can check the **Log in using current Windows account** option on the Repository login dialog to use their Windows user ID and password to log into the Repository.

- **Delete**: Select a user and then click Delete to delete the user. You cannot delete the Admin user.

- Deleting a user also deletes version history for that user, which means that version history for all actions done on various objects by that user will also be deleted. It may be advisable to instead deactivate the user. Deactivating a user does not delete their history.

Administrator User 'Force' Options area

When you use the Check In User option, you can choose how to handle the checked out objects.

- **Convert to non-exclusive**: Enables the user to continue making changes to the checked out items. Other users can then check out the same items. When checking in non-exclusively checked out items, if the item was changed by another user since it was checked out, the user can resolve the differences and choose to override the other user's changes or to update his local version with the changes found in the Repository.

- **Undo Check Out**: When the user next connects to the Repository, the status of his items will be updated to show that they are not checked out.

Notes

- You cannot delete or deactivate the Admin user, but you should change the Admin password from the default, which is Admin.

- The new user appears in the user list with a star indicating that this user does not yet have any roles assigned to it for any Repository items.

- Users must log out and log in again to receive any security updates.

- The Admin does not need to know the user's password to create a new password for the user.

- The Admin can look to the Manage Users tab to find out which users are logged in to the Repository and whether or not they currently have items checked out. Knowing if users have items checked out is particularly useful when upgrading the Repository, because after before upgrading the Repository the users should check all their items in and then do a clean get and check out of the items when the upgrade is complete. Knowing which users are logged in is useful when changing security center settings because after making any setting changes, the Admin should request online users to log off and log in again to receive the security center updates.

- The Administrator may want to user log a user out to receive security center updates. For example, a new diagram has been added to a project and the diagram shouldn't be accessible by all the users who have access to the project. The Administrator can make the necessary security changes for the project and then log out the users, forcing them to log in again to receive the security updates. Other unusual circumstances may arise when it is necessary for the Administrator to log a user out of the Repository.

Assigning a Role to a User

Assigning a role to a user grants the user all the permissions selected in the role.

1. Choose Repository > Security > Security Center.

2. Select a Repository item you want to assign permissions to.

- o **If the user is not currently assigned a role**, on the **Repository Security** tab, click the user name in the **Available Users** column and drag it onto a role name in the **Available Roles** column.

- o **If the user was already assigned a role**, on the **Repository Security** tab, click the user name the **Available Roles** column and drag it onto another role name in the **Available Roles** column.

3. Click *Apply* and then click *OK* to exit the **Security Center**.

4. Notify users that they must log out and then log in again to receive the security updates.

See Also

- Creating and Managing Roles

- Creating and Managing Groups

Creating and Managing Groups

You can use groups in the Repository Security Center to make it easier to handle permissions. You can assign roles to a group at the Repository, Project, and Diagram levels and then add or remove users from the group to apply the settings to the user quickly and easily.

Groups can be created two ways:

- Create a group in the Repository Security Center and then add users to it manually

- Create a group with the same name as an LDAP/Active Directory group, and any LDAP/Active Directory users that are added to the Repository would automatically be members of that group based on the LDAP/Active Directory settings.

Create, Update, and Delete Group

1. Choose Repository > Security > Security Center.

2. Click the *Manage Groups* tab and select *New*.

3. Provide a name for the group and optionally, a description of the group.

4. *If you want to create a group based on an LDAP/Active Directory group*, click *Directory Service Group* and another option becomes available at the bottom of the dialog, *Create Repository Users for all members of this Directory Service Group*. If you select this option, then repository users in this group are automatically added to the list of Repository users. If you do not select this option, then you will need to create the users manually and as you add LDAP/Active Directory Service users, they will be added to the Repository groups that correspond to the LDAP/Active Directory Service groups that they are members of.
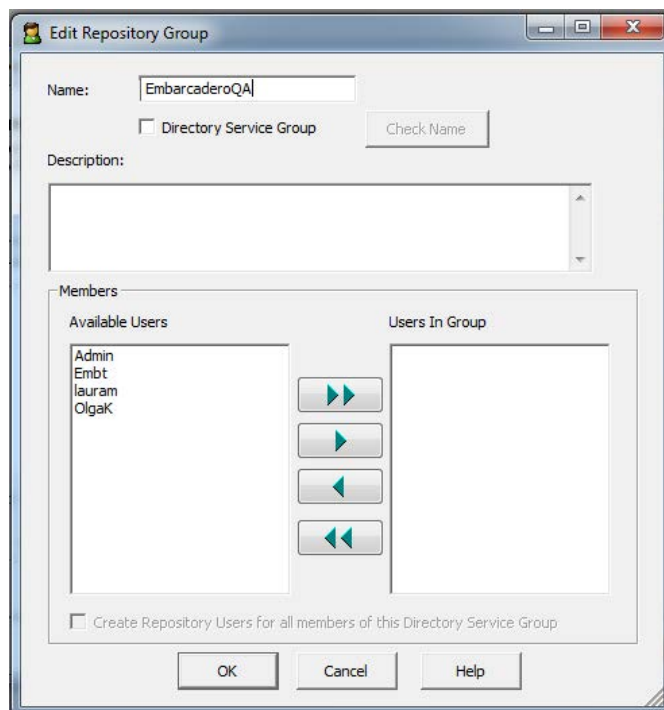
If you want to manually populate the group, do not select *Directory Service Group*, from the list of *Available Users*, select the group members by Ctrl-clicking each user, or click the double right-facing arrow to select all the users. You can also remove group members by clicking the left-facing arrow.

Notes

- *Delete a group*: On the *Manage Groups* tab, you can delete a group by selecting the group and clicking *Delete*. Deleting a group does not delete the group members but the group members will no longer have the permissions afforded to them by the group membership.

- *Edit a group*: On the *Manage Groups* tab, you can edit the group name and list of group members by selecting the group and clicking *Edit* and making your changes on the *Edit Repository Group* dialog.

Assigning a Role to a Group

Assigning a role to a group grants the group all the permissions selected in the role.

1. Choose Repository > Security > Security Center.

2. Select a Repository item you want to assign permissions to.

3. *If the group is not currently assigned a role*, on the *Repository Security* tab, click the group name in the *Available Users* column and drag it onto a role name in the *Available Roles* column.

If the group was already assigned a role, on the Repository Security *tab, click the group name the* Available Roles *column and drag it onto another role name in the* Available Roles *column.*

1. Click *Apply* and then click *OK* to exit the *Security Center*.

2. Notify users in the group that they must log out and then log in again to receive the security updates.

## Granting and Prohibiting User Access to Repository Items

On the Repository Security tab of the Security Center, users with the appropriate privileges can assign other users the appropriate access privileges for different objects, granting them permission to perform specific operations on a Repository item and prohibiting them from performing other operations on the same object. You can assign the access privileges of a role to individual users or to an entire group.

Assigning Users/Groups Roles for Accessing Repository Items

1. Log in to the Repository.

2. Choose Repository > Security > Security Center.

3. Click the *Manage Roles* tab; explore the available roles to determine if an existing role provides the access you want to grant the user. If necessary, create a new role with the appropriate permissions.

4. Click the *Manage Groups* tab; explore the available groups to determine if an existing group encompassed the users you want to authorize users. You can double click the name of a group to see the members of the group in the *Edit Repository Group* dialog. If necessary, create a new group. Assigning a role to a group is more efficient than assigning the role to the users one at a time.

5. Click the *Repository Security* tab.

6. In the *Repository Object* area, select a Repository item (diagram, object, submodel, data flow, transformation, or data dictionary) to which you want to grant or prohibit access.

7. Beneath *Available Users*, select a group or user to which you want to assign the role to the selected object and drag it onto the appropriate role in the *Available Roles* area.

8. To remove the role assignment of a user/group to a Repository item, click the Repository item, drag the user/group name from under role onto another role in the Available Roles area or back onto the Available Users list.
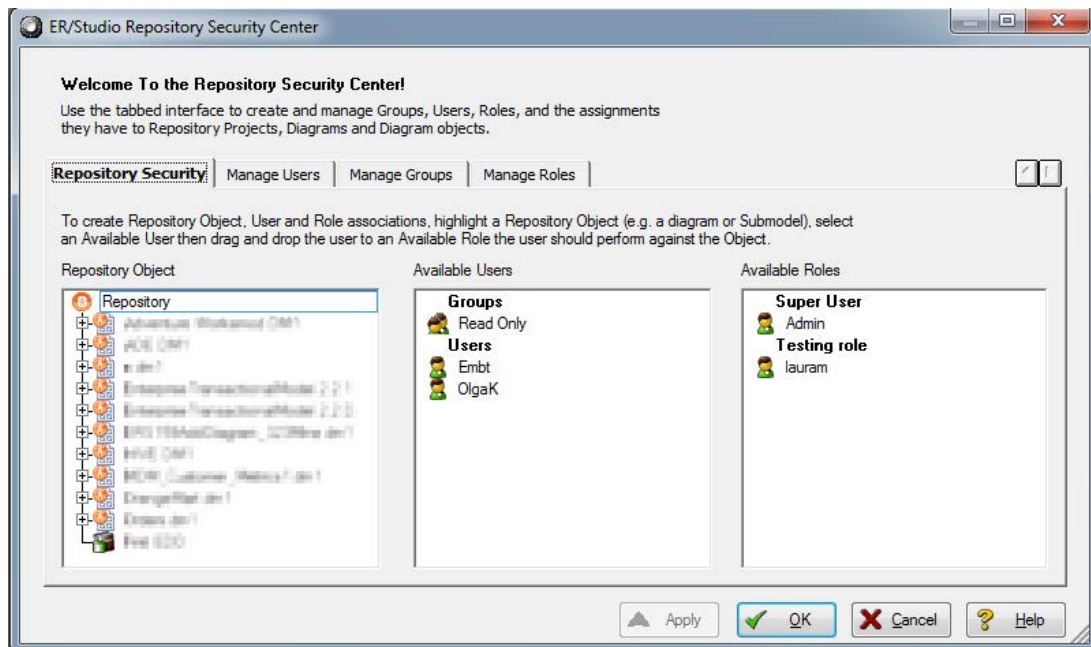
Notes

- All permissions available to a user through direct assignment to roles are added to the permissions available to any groups the user is a member of. Permissions are additive, so the user gets all the permissions assigned to the user whether they were assigned through a group or directly to the user.

- A user or group can only have one role for each object. But a user or group can have different roles for different objects.

- If a user is listed in the Available Users area, but cannot be selected, the user has been deactivated and must be reactivated before you can modify the user.

Prevent Users from Accessing Specific Repository Items

The system-defined No Access role can prevent a user from accessing selected projects or diagrams

1. Log in to the Repository.

2. Choose Repository > Security > Security Center.

3. In the *Repository Object* area, navigate to and then click to select a project or diagram you want to secure from specific users.

4. From the list of users in the *Available Users* area, click a user name or *CTRL-click* several names and then drag the users onto the *No Access* role in the *Available Roles* area.

5. Repeat Step3 and Step4 as required to secure other diagrams and projects.

6. Click *Apply*, continue changing security settings if required, and then click *OK* to exit the security center.

Notes

- If a user has been assigned the No Access role for an object, but that user is a member of a group with permissions for the object, then the user will have all permissions assigned to the group. All permissions available to a user through direct assignment to roles are added to the permissions available to any groups the user is a member of. Permissions are additive, so the user gets all the permissions assigned to the user whether they were assigned through a group or directly to the user.

Change User/Group Access to Repository Items

To change user/group access to Repository items, change the permissions of the role the user/group has for the Repository item or assign another role to the user/group for that particular Repository item. If required, you can force the user log out on the Manage Users tab of the Security Center, and then change the user's access privileges.

See Also

- [Creating and Managing Roles](#)
- [Creating and Managing Users](#)