Toad® Data Studio 1.0

User Guide

Copyright

The information in this document is provided in connection with Quest Software products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Quest Software products. EXCEPT AS SET FORTH IN THE TERMS AND CONDITIONS AS SPECIFIED IN THE LICENSE AGREEMENT FOR THIS PRODUCT, QUEST SOFTWARE ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL QUEST SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF QUEST SOFTWARE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Quest Software makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Quest Software does not make any commitment to update the information contained in this document.

If you have any questions regarding your potential use of this material, contact:

Quest Software Inc.
Attn: LEGAL Dept
4 Polaris Way
Aliso Viejo, CA 92656

Refer to our web site (www.quest.com) for regional and international office information.

Patents

Quest Software is proud of our advanced technology. Patents and pending patents may apply to this product. For the most current information about applicable patents for this product, please visit our website at www.quest.com/legal.

This product includes patent pending technology. For the most current information about applicable patents for this product, please visit our website at www.quest.com/legal.

Trademarks

Toad Data Studio 1.0
User Guide
Wednesday, January 3, 2024

Contents

1

Introduction

About Toad Data Studio

January 03, 2024

Toad® Data Studio is a heterogeneous database IDE intended to assist Data Engineers and other technical users to easily manage, query and develop against any database in your environment.

With Toad Data Studio, you can:

- Connect to almost any database or ODBC data source (see the Release Notes for a list of tested ODBC connections)
- Write SQL queries and join data across all platforms
- Automate and schedule frequent and repetitive tasks
- Streamline data collection
- Compare and synchronize databases
- Export data in a variety of file formats
- Version control SQL databases and objects

Toad Data Studio provides a full-featured Database Explorer, Query Builder, and Editor for the following databases:

- Oracle®
- SQL Server® and SQL Azure
- IBM® DB2® (LUW and z/OS)
- SAP® ASE, IQ, SQL Anywhere, and HANA
- Teradata®
- MySQL
- Snowflake®

Toad Data Studio also provides querying and reporting functionality via the following provider types:

- ODBC
- Business Intelligence and NoSQL data sources
- Microsoft® Excel®
- Microsoft Access®

ODBC Connections

The purpose of the ODBC provider is to offer basic connection and querying capabilities to any database that supports an ODBC 3.0 or later driver. For a list of databases tested with the ODBC provider, see the System Requirements in the Release Notes. This form of connectivity allows connections to databases such as Netezza, IBM iSeries, Ingres, and Vertica™. Because this form of connectivity is generic, it is not full-featured and may not be as robust as the fully-exploited providers for the other databases.

Business Intelligence and NoSQL Data Sources

Toad Data Studio provides users with the ability to connect to non-traditional data sources, such as Business Intelligence and NoSQL sources. You can connect to these data sources and explore data and objects, query data, report results, and automate tasks.

The following data sources are supported:

Business Intelligence	Oracle Business Intelligence Enterprise Edition (OBIEE), SAP Business Objects™, Salesforce®, Microsoft SQL Server Analysis Services, Google Analytics™, OData, and SharePoint®
NoSQL	Azure Table Services, Cassandra™, Cloudera Impala, DynamoDB, HBASE™, Hive™, MongoDB™

Best Practices

Toad provides tools that help you succeed in your query and reporting goals. Many of these tools are considered Best Practices and are beneficial for individual users as well as managers. The following table describes the benefits of using Toad to improve Best Practices:

Best Practice	Benefit
One tool	Connect to any database type for basic query needs. Using one tool for all platforms makes it easier to learn and train others. Managers can create common files shared by all users. This ensures consistency for connections, queries, and reports.
Shared connections	Easily share connections between users by importing or exporting them.
Project management	Organize files and make them easily accessible to provide significant time savings. The Project Manager provides an easy way to group Database Diagrams, queries, and reports. It also provides the following useful features: • Create multiple projects to manage and track related tasks. • Double-click files in the project to quickly load them for review or execution. You can also add favorite web sites and directories. • Email files to colleagues or send them to your local network share. • Use the To-Do list to track items you need to complete.
Automation	Tasks that are frequently repeated can be automated and scheduled using Automation. You can schedule long running queries to execute during off-hours and can automatically export the results to Excel. You can also add these tasks to the Project Manager, and then double-click them to execute them. Automation offers an extension to your work day without you having to be there.
Database Diagram	Database schemas can be complex and difficult for new users to understand. The Database Diagram displays the database relationships between tables to reduce the learning curve. It also provides the following useful features: • Add notes to help users learn more about specific tables and columns. • Hide columns that are rarely used so users can focus on critical information. • Add manual relationships to commonly used joins in queries. • Save, share, or use Database Diagrams as a template for building queries.
Query Builder	Create complex queries visually using the Query Builder. You can also: • Reverse engineer a query from the Editor.

Best Practice	Benefit
	• Create a cross-connection query between different database providers, such as Oracle and DB2. • Add Lookup tables to a column for easy filtering. • Use bind variables to optimize a query. • Annotate tables, relationships, or the actual diagram with notes. • Save or share the query for future use. • Add Query Builder files to the Project Manager for easy access during your sessions.
Excel	Build Excel reports and share them with colleagues. The One Click Export feature offers options for quickly building Excel reports, including pivot tables for data manipulation. If you want data in an exported report to reflect changes to the database, you can use the One Click Export feature to export the underlying query with the report so it can be refreshed with the latest updates.
Share Files and Reports	Store files and reports in libraries that can be shared with colleagues. You can specify a directory on a network share to share these files with others.

Understand Toad's Workflow

Most reporting projects have a general flow throughout their lifecycle that requires you to understand your data sources, create queries, generate reports based on query results, and automate scripts for repetitive tasks.

- Understand—Successful queries begin with a clear understanding of the data source to which you connect. If you understand the data and the objects, as well as their relationships to one another, you can build informed queries that are effective and more optimized than you could without this understanding.
- Query—Queries are the most important, and often the most time-consuming, aspect of your project. Once you have a clear understanding of the data source, you can begin to create your query.
- Report—Data retrieved from a query frequently needs to be shared with colleagues or management in the form of a report, such as a pivot grid or chart.
- Automate—Queries and reports usually need to run on a daily, weekly, or monthly basis. With Automation, you can create a workflow that includes connecting to data sources, importing/exporting data, executing scripts, running queries, and building and distributing reports on a scheduled basis.

2

Connect

Create Amazon Redshift Connections

Toad Data Studio allows you to create a connection to an Amazon Redshift database using ODBC connectivity.

Note: You must have Amazon Redshift ODBC driver 1.3.1.1000 (or later) installed before a native connection is created.

To install the Amazon Redshift ODBC Driver

Download and install the ODBC Driver from the Amazon Redshift Web site.

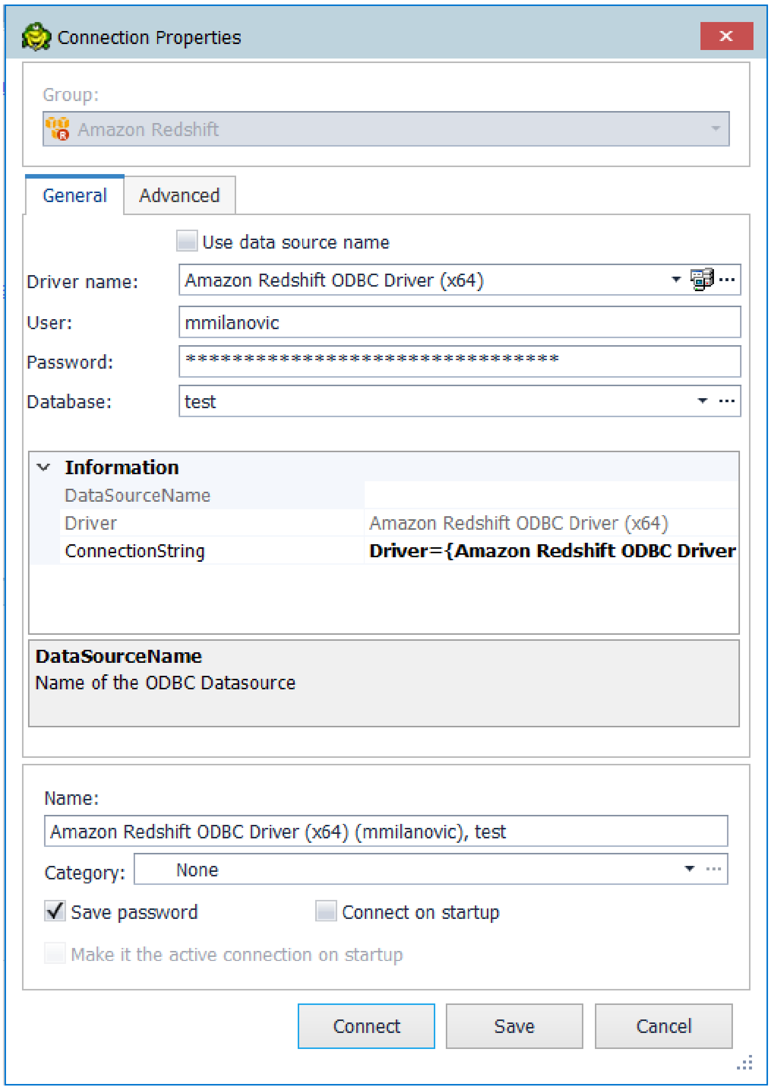
To create a new connection

1. Click on the **Connections** toolbar (ALT+F+N).
2. Select ***Amazon Redshift*** from the **Group** list box.
3. In the **Create New Connection** window configure the connection parameters.

General Tab	Description
Driver name	Uncheck the Use data source name checkbox and select Amazon Redshift (x64) Driver or Amazon Redshift ODBC Driver (x64) depending on the installed version.
User	Enter the username for your Amazon Redshift account.
Password	Enter the password for your Amazon Redshift account.
Database	Enter the name of the Database you would like to connect to or choose from the drop-down list.
ConnectionString	Add the connection string for the Redshift cluster in the ConnectionString field. Add the password string to the PWD= setting in the ODBC URL and paste the ODBC URL to the ConnectionString field: Driver={Amazon Redshift ODBC Driver (x64)};Server=&lt;server&gt;;Database=&lt;database&gt;;UID=&lt;user&gt;;PWD=&lt;password&gt;;Port=5439

4. Click **Connect** to save the connection and immediately connect to the database or click **Save** to save the

connection without connecting to the database.



Connection Properties

Group: Amazon Redshift

☐ Use data source name

Driver name: Amazon Redshift ODBC Driver (x64)

User: mmilanovic

Password: ********************************

Database: test

Information

DataSourceName	
Driver	Amazon Redshift ODBC Driver (x64)
ConnectionString	**Driver={Amazon Redshift ODBC Driver**

DataSourceName
Name of the ODBC Datasource

Name: Amazon Redshift ODBC Driver (x64) (mmilanovic), test

Category: None

☑ Save password ☐ Connect on startup

☐ Make it the active connection on startup

Tip: Connections are stored in the connections.xml file and can be found by clicking the *Application Data Directory* link in **Help | About**.

Create ODBC Connections

Toad Data Studio allows you to create a connection to a database that supports an ODBC 3.0 or later driver. ODBC connectivity provides basic querying capability, but may not be as full-featured as a native connection in Toad. For a list of databases tested with the ODBC provider, see the Release Notes.

This form of connectivity allows connections to databases such as Netezza, IBM iSeries, Ingres, and Vertica.

Note: Distinct values are not available in Code Completion for ODBC connections.

DSN Architecture (Bitness) Must Match Toad

To create a successful ODBC connection, the architecture (bitness) of the driver in the specified DSN must match the architecture (bitness) of Toad.

To create an ODBC connection

1. Click on the toolbar (ALT+F+N+O).
2. Select *ODBC Generic* from the **Group** list box.
3. **Use data source name**—Select to display data source names. Clear this check box if you want to display the driver names.
4. To create a data source to use in this connection, click in the **Data Source Name** field. Then click **Add**.
5. Select an ODBC driver from the list, and click **Finish**.

   Note: If creating an Oracle ODBC connection, select the ODBC driver provided by Oracle. The Microsoft ODBC for Oracle driver has less functionality than the driver provided by Oracle.

6. Specify the configuration properties required for the database in the Windows configuration dialog.

   Notes:

   - If you have an Oracle database with Unicode data, make sure to select *Force SQL_WCHAR Support* on the Workarounds tab of the windows Oracle ODBC Configuration dialog. If you do not, you will not be able to see the data.
   - If creating a MySQL ODBC connection to use in a cross-connection query, you must specify a database in the Windows MySQL ODBC Configuration dialog.

7. Specify the connection properties in the Create New Connection dialog. Review the following for additional information:

General	
Data source name	Select the driver or data source name you added in the previous steps.
User	Enter the user name to use when connecting.
Password	Enter the password to use when connecting. **Tip:** After connecting, you can set a master password to further secure your connection in **Tools | Options| Environment | Security**.
Database	Select a database or schema. Click ... to create a temporary connection and display available databases/schemas in the drop-down list.
Information	
Data Source Name	Displays the selected driver or data source name.
Driver	Displays the ODBC driver associated with the data source.
Advanced	
Default Table Column	Specify default options to use when creating a new table. **Column type**—Select the default data type to use when creating a table. **Column length**—For the selected column type, enter a default column length to use.
Advanced Options	**Block Cursor Size**—Specify the number of rows to return in a single fetch of data from the result set. **Disable multi-threading**—Multi-threading is disabled by default. Enable multi-threading only if supported by the driver. Multi-threading can improve performance, but can also result in unpredictable behavior if not supported by the driver. Default: *Selected*
Category	(Optional) Select an existing category or create a new one.

8. Click **Connect** to connect immediately while saving the connection information. Optionally, click **Save** to save the connection without connecting.

Tip: Connections are stored in the connections.xml file and can be found by clicking the *Application Data Directory* link in **Help | About**.

Tips for Working with ODBC Connections

Tip	Description
Filter DSN by bitness	To filter Data Source Names displayed in the New Connection dialog by the bitness of Toad, go to **Tools | Options | Database | ODBC**.
Cache object metadata	To cache object metadata for ODBC connections and retain it between sessions, go to **Tools | Options | Database | ODBC** and select to enable disk caching. Cached object metadata is retained until manually refreshed.
Refresh object metadata	To refresh object metadata for all ODBC connections, in the Object Explorer right-click an object and select **Refresh All**. Right-click an object and select **Refresh Schema** to refresh only objects in that schema.

ODBC Features

Feature	Description
Bypass schema/instance selection in SQL Editor	For ODBC connections, in the SQL Editor you can bypass selecting a schema/instance and specify this information through the script instead. This is useful if your script executes SQL against multiple schemas. To use this method, select **I will set schema/instance in the script** from the schema/instance drop-down list in the SQL Editor window. Then specify the schema/instance through your script. **Note:** When this option is selected, some code completion features, object actions, and object information tool tips are unavailable for this SQL Editor window.

Click here to view a video about creating connections in Toad Data Studio

Create Oracle Connections

You can create a new connection, or connect to an existing connection from the Create New Connection window.

Click here to view a video about creating connections in Toad Data Studio

Tip: You can configure Toad Data Studio to accept connection information from Toad for Oracle.

Client Connection or Direct Connection

You can connect to your database using a native Oracle database provider in Toad using one of the following methods:

Connection Method	Advantages	Disadvantages
Client connection	Supports LDAP	Requires an Oracle client

Connection Method	Advantages	Disadvantages
Client tab—Uses an installed Oracle Client or Oracle Instant Client to establish a connection to the database from Toad.		**Note:** Toad supports Oracle clients and Oracle Instant clients using version 9i or later.
Direct connection Direct Connect tab—Uses TCP/IP to connect directly to the database.	Does not require an installed Oracle Client Recommended for connecting to Oracle 8i databases	This method only supports common data types. Some advanced Oracle object types cannot be retrieved when connecting using this method. This method does not support LDAP. (To connect using LDAP, use the Client Connection method instead.) This method cannot be used in a cross-connection query.

Note: Quest recommends using one of the methods described in this table to connect to Oracle rather than an ODBC driver. Because ODBC connectivity is generic, it is not full-featured and may not be as robust as the fully exploited native database provider.

Configure the Oracle Instant Client

The Oracle Instant client installs only those files needed to connect to your database using OCI. Because this client does not write to the registry (no ORACLE_HOMES) or add to the environmental path, you must manually set environment variables so the client can be used with Toad.

To configure the Oracle Instant Client to work with Toad

1. Save the Oracle Instant Client files in any location on the computer where Toad is installed.
2. On your Windows desktop, right-click My Computer and select **Properties**.
3. Select the Advanced tab and click **Environment Variables**.
4. Click **New** under System Variables and complete the following:

Variable name	Variable Value
PATH	<*Path where Instant Client is installed*>;%PATH% For Example: C:\Program Files\OracleInstantClient;%PATH%
TNS_ADMIN	<*Path where tnsnames.ora file is installed*>

5. To use the Oracle Instant Client connection in a cross-connection query, you must also install an Oracle ODBC driver. (While the Oracle Client *full* install includes an ODBC driver, the Oracle Instant Client does not.)

Note: You cannot use an Oracle Direct Connection in a cross-connection query.

Create an Oracle Connection

To create an Oracle connection

1. Click on the toolbar (ALT+F+N).
2. Select *Oracle* from the **Group** list box.
3. Review the following for additional information:

Login Tab	**Description**
Database name	Select the alias to use when connecting. This list displays aliases from LDAP, as well as any local tnsnames.ora file. When you initially open the connection dialog, Toad retrieves the list of LDAP names from the LDAP server. Toad caches this list and displays it upon subsequent requests, within one Toad session. To refresh the list, click 🔄. **Note:** If you do not select a database, Toad uses the ORACLE_SID for the selected home. If an ORACLE_SID value does not exist and you do not specify a database, you cannot connect.
Password	Enter the password to use when connecting. **Tip:** After connecting, you can set a master password to further secure your connection in **Tools | Options| Environment | Security**.
Connect as	Select whether to connect as either SYSDBA or SYSOPER if you have the appropriate permissions. Otherwise, use the default.
Schema	Select a schema if you frequently work in a different schema than the one associated with your user name.
Explain plan table	(Optional) Specify the location of an existing explain plan table to use instead of Toad automatically creating one for you when you execute a statement. This is useful if you do not have necessary permissions to create explain plan tables or want to use an existing table.
Category	Select or create a category if you want to color code Editor tabs for a specific connection. This can help differentiate between development and production databases. You can also set an option to color code the Object Explorer pane and object editor windows (Create, Alter, Drop, etc.). 🎞 Click here to view a video of this feature.
Save password	Select this check box if you plan to use this connection in a scheduled Automation script. **Note:** This option is disabled if Toad was installed with **Prohibit saving passwords** selected.
Oracle Client Tab	**Description**
Current home	Modify the Oracle home to use, if needed.
TNSNames Editor	Click this button to edit your local TNSNames file. **Tip:** You can press CTRL+F to open the Find/Replace window to locate entries in the TNS Names Editor. Pressing F3 locates the next entry that matches your criteria.
SQLNET Editor	Click this button to edit SQLNET.ORA parameters. Before editing this file, you should create a backup copy. See *SQLNET.ORA Profile Parameters* in the Oracle documentation for more information.
Direct Tab	**Description**
SID	Enter the ORACLE_SID value for the database. This value is specified in the registry under each installed home.
Advanced Tab	**Description**
ODBC driver	If an Oracle Client is installed, the default is the Oracle ODBC driver. If an Oracle Client is **not** installed, the default is the Microsoft Access ODBC driver. **Tip:** You can specify the ODBC driver to use for all connections for this database provider in **Tools | Options | Database | Oracle**.

4. Click **Connect** to connect immediately while saving the connection information. Optionally, click **Save** to save the connection without connecting.

Note: Go to **Tools | Options | Database | Oracle** to specify default options to use in Oracle connections.

Tip: Connections are stored in the connections.xml file and can be found by clicking the *Application Data Directory* link in **Help | About**.

Create Snowflake Connections

Toad allows you to create a native Snowflake connection, which provides you with a full-featured Object Explorer, Query Builder and SQL Editor.

Note: You must have a Snowflake ODBC driver 2.24.2 (or later) installed before a native connection is created.

To install the Snowflake ODBC Driver

Download and install the ODBC Driver from the Snowflake Web site.

To create a new connection

1. Click on the **Connections** toolbar (ALT+F+N).
2. Select ***Snowflake*** from the **Group** list box.
3. Enter the connection information in the **Create New Connection** dialog. Review the following for additional information:
4. Review the following for additional information:

General Tab	Description
Host	Enter the name of the server hosting the database to which you want to connect.
Authentication	There are four types of authentication that you can use to connect to Snowflake in Toad. The required Login fields will change according to the type of authentication you select. All the authenticators are located in a dropdown list: • Snowflake – User account authentication • SSO – Single Sign-On via external browser • OAuth – Token-based authentication • MFA - Multi-Factor Authentication • Key pair - JSON Web Token (JWT)
User	Enter the username for your Snowflake account
Password	Enter the password for your Snowflake account
Token	Enter your authentication token (Only visible when connecting via OAuth)
Private key path	Enter your Private key (Only visible when connecting via Key pair)
Passphrase	Passphrase is required only if the keys are encrypted (Only visible when connecting via Key pair)

Port	The default port number for Snowflake is 443
Optional	**A connection can be made without providing this information**
Role	Enter the user role that you would like to connect to
Warehouse	Enter the name of the Warehouse you would like to connect to
Database	Enter the name of the Database(s) you would like to connect to
Schema	Enter the name of the Schema(s) you would like to connect to

4. Click **Connect** to save the connection and immediately connect to the database.

   or

   Click **Save** to save the connection without connecting to the database.

Authentication Notes

When using SSO or MFA, there is a way to maintain continuous and secure connectivity without needing to enter login credentials for each connection attempt to Snowflake (as well as to prevent additional browser loading whenever a new module is started in Toad Data Studio). To achieve this, check whether the correct parameters have been set in your account by executing the query below in your Snowflake cloud interface:

SHOW PARAMETERS IN ACCOUNT;

The output that will secure login credentials caching will have these values:

For SSO

allow_id_token = true;

For MFA

allow_client_mfa_caching = true;

If these values are set to false, then the following queries need to be executed:

For SSO:

ALTER ACCOUNT

SET allow_id_token = true;

For MFA:

ALTER ACCOUNT

SET allow_client_mfa_caching = true;

Note: If a Role or Warehouse isn’t chosen, the default settings from the Snowflake User account will be applied (it is considered best practice that a user sets a default Role and a default Warehouse in their Snowflake user account).

Tip: It is strongly advised to use only the Snowflake user account authentication when creating Automation scripts to avoid unnecessary workflow interruptions.

Limitations of Snowflake Native Connections

- Connecting to Snowflake through Programmatic SSO (Okta only) is not currently supported.
- Semi-structured data types can be used for storing and analyzing purposes in Snowflake connections, but are not supported in the Data Compare/Sync Wizard because Toad will not be able to generate valid sync scripts unless they are modified before execution
- At the moment, there is no support for Code Snippets in Snowflake connections

Tip: Connections are stored in the connections.xml file and can be found by clicking the *Application Data Directory* link in **Help | About**.

Create SQL Server and SQL Azure Connections

You can create a new connection, or connect to an existing connection from the Create New Connection window. See "Troubleshoot SQL Server® Issues" on page 24 for more information about troubleshooting connection issues.

Click here to view a video about creating connections in Toad Data Studio

Note: For SQL Azure connections:

- To be able to utilize majority of Toad functionality please grant VIEW DEFINITION on the SQL Azure databases you want to work with.
- You can login to Windows Azure Platform by selecting **Tools | Cloud Computing | SQL Azure Portal** and providing your credentials.
- Make sure that no Azure firewall rule restricts your IP address from having access to Azure server. See "Create and Alter Azure Firewall Rules" in the online help for more information.

To create a new connection

1. Click on the toolbar (ALT+F+N).
2. Select *Microsoft SQL Server* or *Microsoft SQL Azure* from the **Group** list box.
3. Review the following for additional information:

Login Tab	Description
Server name	Enter the name of the server to use when connecting. Click … to display any servers running SQL Server that are currently active on the network. **Notes:** • For SQL Azure connections, enter InstanceName.database.windows.net in this field. • For named instances, enter the *MachineName\InstanceName* in this field.
Authentication	Select the type of authentication to use for this connection. • You can connect to an Azure SQL database with Active Directory authentication (either Active Directory Integrated Authentication or Active Directory Password Authentication). Client configuration is required. See https://learn.microsoft.com/en-us/sql/connect/ado-net/sql/azure-active-directory-authentication?view=sql-server-ver16

	- You can also connect to an Azure SQL database with Active Directory Universal with MFA. **Note:** You cannot create a SQL Azure connection using Windows Authentication.
Password	Enter the password to use when connecting. **Tip:** After connecting, you can set a master password to further secure your connection in **Tools | Options| Environment | Security**.
Application ID	Enter the application ID located in App registrations in Azure Active Directory. (This is a mandatory field when connecting to Microsoft SQL Azure via MFA.)
Redirect URI	Enter the appropriate redirect URI.(This is an optional field when connecting to Microsoft SQL Auzre via MFA which can be left empty unless you have specific requirements for configuring the Redirect URI.)
Category	Select or create a category if you want to color code Editor tabs for a specific connection. This can help differentiate between development and production databases. You can also set an option to color code the Object Explorer pane and object editor windows (Create, Alter, Drop, etc.). 🎞 Click here to view a video of this feature.
Encrypt connection	Select this option to enable encryption for the current connection provided that the certificate has been already installed on the server.
Advanced Tab	**Description**
Network protocol	Select one of the following: - Shared Memory - TCP/IP - Named Pipes **Note:** When establishing Named Pipes connection "Poll service status in connection list" option should be cleared.
Parameter Name/Value	You can add parameters to the connection. Toad includes the **Application Name** parameter with value *Toad Data Studio* by default.
OBDC driver	Select an ODBC driver to use for this connection when creating cross-connection queries. **Tip:** You can specify the ODBC driver to use for all connections for this database provider in **Tools | Options | Database | SQL Server**.

4. Click **Connect** to save the connection and immediately connect to the database.

   or

   Click **Save** to save the connection without connecting to the database.

Tip: Connections are stored in the connections.xml file and can be found by clicking the *Application Data Directory* link in **Help | About**.

To import connections (including Central Management Server connections)

1. Click .
2. Specify the location where the connections currently reside.

   Note: Review the following for additional information on CMS connections:

   - To import CMS connections, select a server from the list where the CMS is registered. The connection to this SQL Server instance should be created beforehand.
   - CMS is available beginning with Microsoft SQL Server 2008.

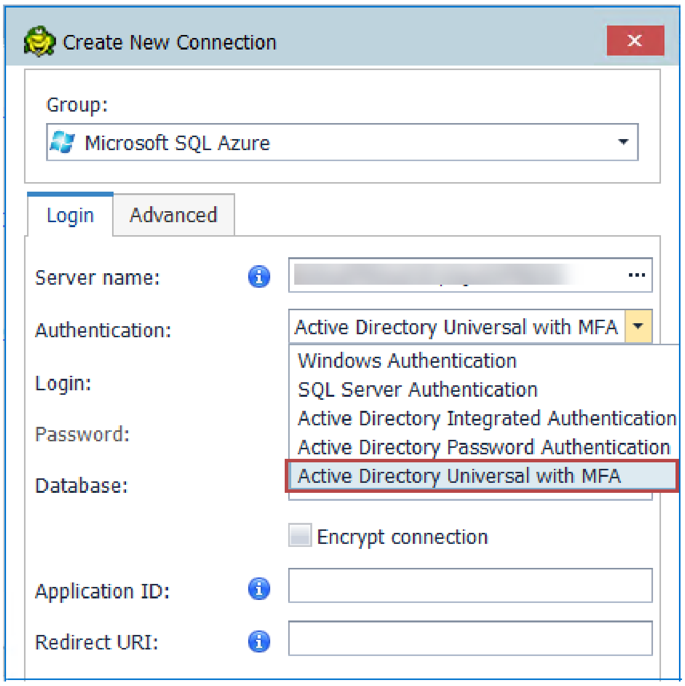
- User should have an appropriate login (Windows Authentication and grant to connect to selected server are required) created on the server to which he connects via CMS.

3. Select the connections to be imported.

Microsoft SQL Azure Connections with Multi-factor Authentication

To create this type of connection to **Microsoft SQL Azure**, **Active Directory Universal with MFA** needs to be selected from the Authentication field dropdown.

The next step is to enter a user name which has been configured for multi-factor authentication and put in the appropriate App ID. After filling out the required information, click on **Connect**.



Create New Connection

Group: Microsoft SQL Azure

Login | Advanced

Server name: [redacted]

Authentication: Active Directory Universal with MFA

Login:

Password:

Database:

□ Encrypt connection

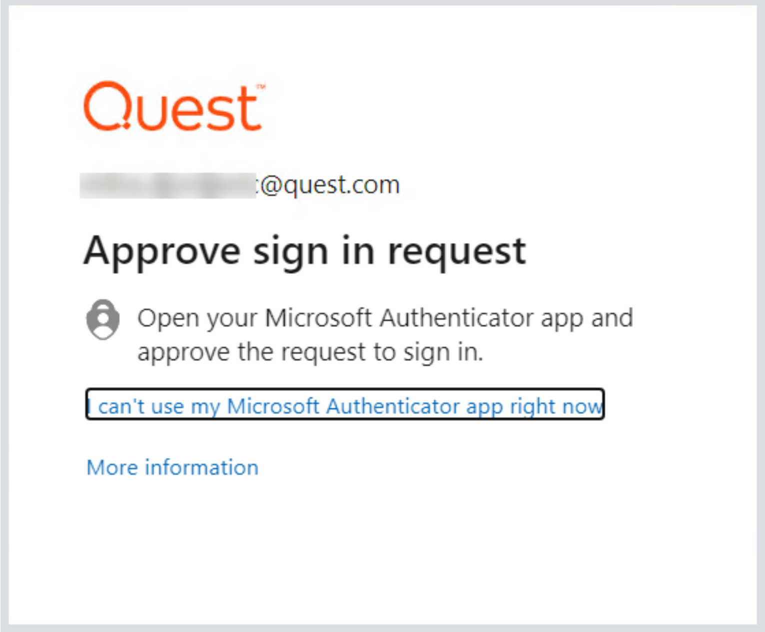
Application ID:

Redirect URI:

Windows Authentication
SQL Server Authentication
Active Directory Integrated Authentication
Active Directory Password Authentication
Active Directory Universal with MFA

The step that follows will enable you to select a user from the list or add another one by clicking on **Use another account.**

After that, you will be prompted to enter your credentials and depending on the type of authentication that the account has been configured with, you will undergo an additional authentication step, such as the one in the image below.



[redacted]@quest.com

Approve sign in request

Open your Microsoft Authenticator app and approve the request to sign in.

After completing the final authentication step, a successful connection to **Microsoft SQL Azure** will be made.

Create SharePoint Connections

Toad allows you to connect to SharePoint via OData service to view and query SharePoint lists as tables. Toad supports SharePoint 2010 and 2013. You can also create a connection to SharePoint Online.

To create a SharePoint connection

1. Click on the Connections toolbar (ALT+F+N).
2. Select *SharePoint* from the **Group** list.
3. Enter the connection information in the Create New Connection dialog. Review the following for additional information:

URL	Enter the full URL address to the SharePoint OData service endpoint for the SharePoint site to which you want to connect. The URL address should have the following format: http://<servername>/_vti_bin/listdata.svc
Authentication	Select the type of authentication to use for this connection. Toad offers two options: • Basic Authentication • Active Directory Universal with MFA The required login fields will be enabled or disabled according to the authentication type that is chosen.
Username	For SharePoint—Enter your Windows user name to use when connecting to this SharePoint site. For SharePoint Online—Enter the user name to use when connecting to this instance of SharePoint Online.
Password	Enter the password to use when connecting.

	Tip: After connecting, you can set a master password to further secure your connection in **Tools | Options| Environment | Security**.
SharePoint Online	Select this option if creating a connection to a SharePoint Online instance.
Category	(Optional) Select an existing category or create a new one.

Note: In NoSQL and Business Intelligence connections, Toad automatically saves the password in the connections.xml file as obfuscated text, as well as in Toad. To add additional password security, use Toad's Master Password feature.

4. Click **Connect** to connect immediately while saving the connection information. Optionally, click **Save** to save the connection without connecting.
5. Upon creating an initial connection, Toad automatically maps the data source objects. This process runs in the background, and until it is finished, you may experience a delay when attempting to access these objects.

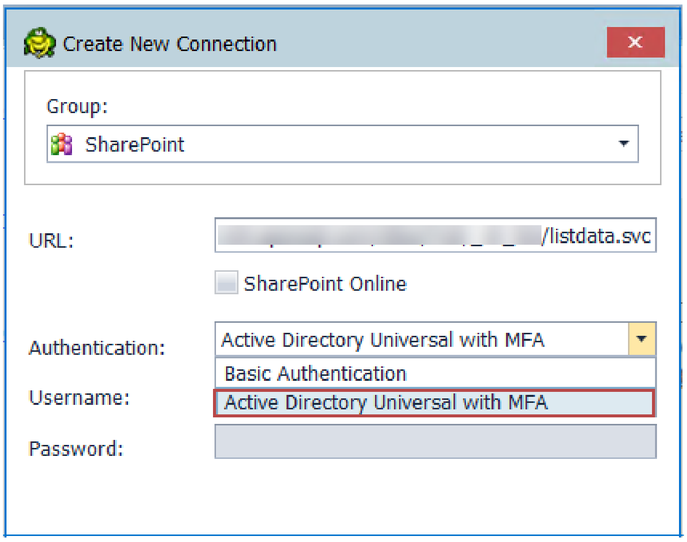
Tips:

- Connections are stored in the **connections.xml** file and can be found by clicking the *Application Data Directory* link in **Help | About**.
- You can specify a proxy server to use for Business Intelligence and NoSQL connections at **Tools | Options | Database | Data Services**.

SharePoint Connections with Multi-factor Authentication

To create this type of connection to **SharePoint**, **Active Directory Universal with MFA** needs to be selected from the Authentication field dropdown.

The next step is to enter a user name which has been configured for multi-factor authentication and click on **Connect**.



Create New Connection

Group: SharePoint

URL: [redacted]/listdata.svc

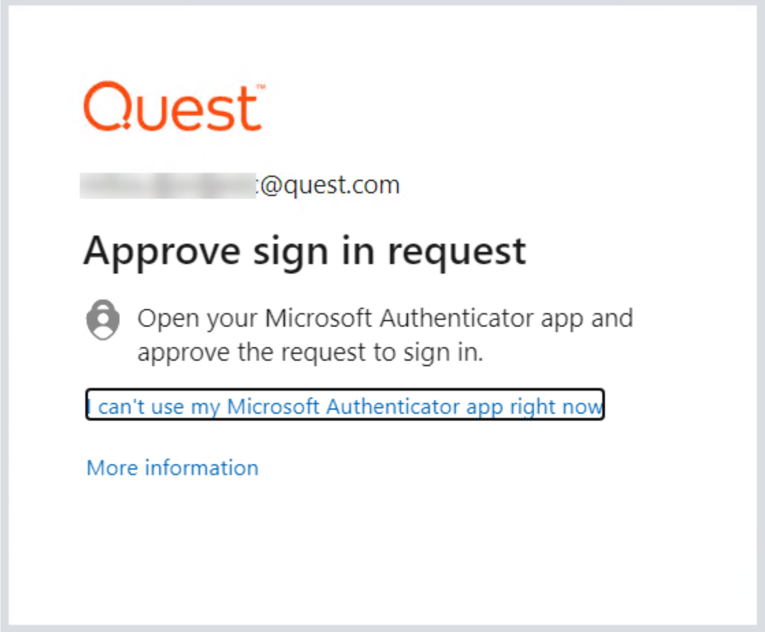
□ SharePoint Online

Authentication: Active Directory Universal with MFA

Toad Authentication Window will appear allowing you to select either **Windows Authentication** or **AzureADTrust** for the type of credentials which will be used for the connection.

The step that follows will enable you to select a user from the list or add another one by clicking on **Use another account.**

After that, you will be prompted to enter your credentials and depending on the type of authentication that the account has been configured with, you will undergo an additional authentication step, such as the one in the image below.



Quest

[redacted]@quest.com

Approve sign in request

Open your Microsoft Authenticator app and approve the request to sign in.

After completing the final authentication step, a successful connection to **SharePoint** will be made.

Limitations of SharePoint connections with Multi-factor Authentication

- Setting a default **SharePoint** connection in **Tools|Options|Environment** is currently available only for Basic Authentication

Understand Toad Connections

Review this topic to learn about open database connections in Toad.

Toad Open Connections

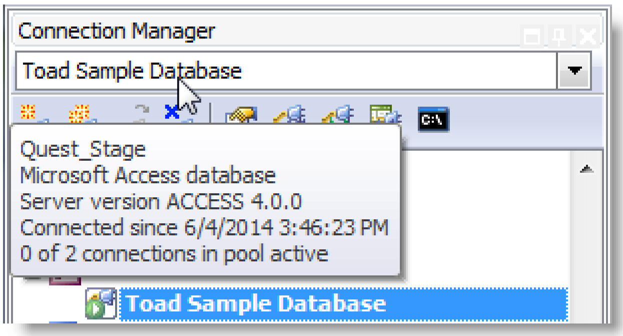
In Toad, an open connection in the Connection/Navigation Manager represents one or more actual connections to the database or data source. When you open a database connection in Toad:

- Toad creates an initial database connection for general use.
- Toad also creates a separate database connection for each Editor (the first time you execute a statement) and for each Data tab in the Database Explorer (the first time you select the tab and query data).

  The connection associated with the window (or tab) remains open until you close the window. If a window has an associated connection, but a statement is not currently executing, the connection is in an inactive state.

To view information for an open connection

- To view the number of actual database connections that exist for a Toad connection, including the number of active connections, place the cursor over the connection in the Connection/Navigation Manager pane or in the connection drop-down list.



Quest_Stage
Microsoft Access database
Server version ACCESS 4.0.0
Connected since 6/4/2014 3:46:23 PM
0 of 2 connections in pool active

To use the Background Processes pane

- When a statement is executing, a progress bar displays in the Status bar located at the bottom of the Toad application window. Click the executing progress bar to open the Background Processes pane.
- To cancel statement execution, use the Background Processes pane.

  Note: It is recommended that you cancel only your own processes, because terminating Toad activities could cause unexpected behavior.

Sharing a Connection Among Windows

You can instruct Toad to share an open connection among windows. In this way, the Editor windows and Data tabs can share the same connection when the connection is inactive.

To share a database connection among windows in Toad

1. Select **Tools | Options | Editor | General**.
2. Clear the checkbox for the **Use individual connection for each open editor** option.
3. When this option is not selected, Toad attempts to use a single connection for any new windows or query execution. For example, if you share a database connection, you can execute a query in the Editor, execute another query in a new Editor, and then select the Data tab from the same connection.

   If the single connection is in use when Toad attempts to execute it, the following occurs:

   - (DB2 and Oracle only) If your database supports executing multiple queries simultaneously on a shared connection, the query is placed in a queue and executes as soon as the current operation completes.
   - If your database does not support executing multiple queries on a shared connection, a dialog prompts you to either permanently associate the window with a new database connection, cancel the previous operation, or add the query to a queue until the current operation completes, and the execute it. If you place multiple queries in the queue, each one executes in the order that it was added to the queue.

Note: Although you can share a database connection, any commits apply to the selected window only.

Troubleshoot SQL Server® Issues

Connection Issues

Review the following solutions to help troubleshoot SQL Server connection issues in Toad.

Issue	Cause/Solution
Cannot specify a port number when connecting	Currently, Toad does not have a separate field for specifying a port number. **Solution:** You can connect if you know the port number for the instance by entering the following in the **Server name** field on the New Connection window: ServerName,Port or IPAddress,Port For example: 10.1.29.129,1234 (where 1234 is the port number of the instance)
Cannot connect to a remote instance in SQL Server 2005	Remote connections may not be enabled in SQL Server 2005. Also, named instances are configured to use dynamic ports and select an available port once the SQL Server Service starts. If you are connecting to a named instance through a firewall, configure the Database Engine to listen on a specific port, so that the appropriate port is opened in the firewall. **Note:** Remote connections are disabled by default in the Express and Developer editions of SQL Server 2005. **Solution:** To enable remote connections, see "How to configure SQL Server 2005 to allow remote connections" at: http://support.microsoft.com/kb/914277
"Path is not of a legal form" error message when connecting or "Could not find a part of the path C:\Program Files\Quest Software \Toad Data Studio\Template\Scripts\ InitialSqlServer.Sql" error message when connecting	You attempted to migrate settings from a previous install of Toad to the current install, but the settings file (InitialSqlServer.Sql) for the previous install was either removed, moved, or corrupt. You may also receive this error if you installed Toad and migrated settings from a previous version, and later uninstalled the previous version. **Solution:** You must specify a valid location for the InitialSqlServer.Sql file in Toad. ***To specify a valid location*** 1. Open the SQL Server options page under **Tools | Options | Database | SQL Server**. 2. Specify the location for the new install's InitialSqlServer.Sql file in the **Run Script on new connection** field. By default, this file is located in the following directory: C:\Program Files\Quest Software\Toad Data Studio *version* \Templates\Scripts **Note:** Clicking ... may result in a "Path not found error". If you receive this error, manually enter the location in the field.
Cannot establish Named Pipes connection	If you are trying to connect to the server using Named Pipes, "Poll service status in connection list" option should be cleared.

Issue	Cause/Solution
	Solution: 1. Open the options page under **Tools | Options | Database | General**. 2. Clear **Poll service status in connection list** checkbox.
Cannot access some functionality when working with Azure connection.	To utilize majority of Toad functionality VIEW DEFINITION grant is required for those databases you want to work with.

3

Understand

Determine the Best Method to Locate Objects

Toad offers a number of features to assist you in locating objects. The following table provides guidelines based on typical use cases:

If you know the....	Use this...
Column name, but do not know the table name	Use Object Search
String of DDL for an object	
Dependency for an object	Uses and Used By tabs in the Database Explorer, and then drill-down to the object from there
Criteria for excluding objects	Filter feature in the Object Explorer

Object Explorer

You can view database objects for the current connection with the Object Explorer. This feature is useful for dragging and dropping database objects into an open Editor, Query Builder, or other document window. Right-click the objects in the Object Explorer to access additional features such as multiple tasks and actions.

To open the Object Explorer

1. Select **View | Object Explorer**.
2. Review the following for additional information:

Schema (first field)	Select the schema you want to view. You can filter the list of schemas that display. See "Filter Databases and Objects" on page 27 for more information.
Object Filter (second field)	You can quickly set a filter in this field. To do so, enter a name or partial name of an object and press ENTER. Any advanced object filters you set display in this field. You can also perform advanced filtering based on case sensitivity, statistics, or other options. See "Filter Databases and Objects" on page 27 for more information. **Tip:** You can enable an option to filter using regular expressions in **Tools | Options | Explorer | General**.

Note: For tables and views, the columns, data types, and comments display at the bottom of this window.

Tips:

- To navigate to a previously selected object, click ←. You can also click → to move to the next selected object.
- To locate objects including procedures, functions, views, variable names and comments in an object's source code, use Object Search. See "Search for Objects" on page 32 for more information.
- To use a different method to select objects such as tabs, drop-downs, or a tree view, click ▦.
- To generate reports, export data, create SQL statements, or send objects to the Project Manager, SQL Modeler, or Master Detail windows, right-click the objects and select an option.
- To open the Background Processes window and cancel a query, click the progress bar at the bottom of the window.

Filter Databases and Objects

Toad allows you to filter the list of databases or objects that display in the Object Explorer. You can use filters to reduce the number of databases and/or objects to work with.

Types of Filters

You can create a *Quick* filter by entering a search string in the object filter field. You can also use the Filter dialog (click ▽) to choose additional filter options. If you use the Filter dialog, you can also name and save your filter.

Filter Type	Description	Behavior
Quick (unnamed)	Can be used to quickly filter objects for a single connection or schema. Create a Quick filter by entering a search string in the object filter field.	Applies to the current schema (or to objects in the current schema) only. Once applied, a Quick filter is in effect on that connection when you reconnect or relaunch Toad. Because Quick filters are not named, they cannot be applied to other connections by selecting from the filter list. You must recreate the Quick filter for each connection/schema.
Detailed (unnamed)	Use like a Quick filter, but add more detail by using the Filter dialog to enter filter criteria. Use this method to create an unnamed database filter.	-- same behavior as a Quick filter --
Named	Create and save named filters to reuse and to apply to other connections of the same data source type.	Once applied, a Named filter is in effect when you reconnect or relaunch Toad. Can be re-used on other connections of the same data source type (DBMS).

Note: If you rename a table that has a filter defined, it results in an orphaned filter. Orphaned filters are automatically removed unless you used a tool other than Toad to remove them.

Before Creating a Filter

A filter is associated with a specific list of objects from a specific database. Consider the following guidelines when setting filters:

- A filter applied to a set of objects such as tables in one specific database connection is not automatically applied to the list of objects in another database connection.
- A filter applied to a list of tables on one schema in the database is not automatically applied to the list of tables for a different schema in the same database.
- A filter set for a specific schema or object is applied when altering objects, using the SQL Editor, Master Detail Browser, etc., for that schema or object.

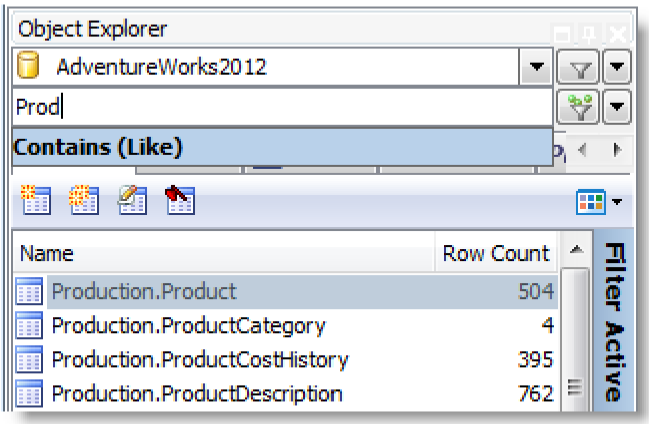
  Tip: To apply any currently-set filters to the code completion list in the editor, select the **Apply explorer filters** option in **Tools | Options | Editor | Code Completion**.

Create Filters

To create a Quick filter (objects only)

- Enter a search string to use for filtering in the object filter field (below the database field) in the Object Explorer.

  Note: As you enter a search string, the currently specified Search Condition displays in a blue banner below the object filter field. This Search Condition will be used in your filter. To change the Search Condition, click ▽ and select a different one from the Search Condition drop-down.



Name	Row Count
Production.Product	504
Production.ProductCategory	4
Production.ProductCostHistory	395
Production.ProductDescription	762

To create a Detailed filter

1. Click ▽ beside the database or object filter fields in the Object Explorer.
2. Specify criteria for the filter in the Properties tab. Review the following for additional information:

Search String	Enter a text string and search characters to use for filtering.
Search Condition	Select a search type to use in this filter. **Note:** You can set a default value for this field in **Tools | Options | Explorer | General**. If you don't specify a default value in Options, this field defaults to the search type you specified the last time you created or modified a filter (includes previous sessions).

	Tip: To include in your text string certain characters normally used in search logic, such as the underscore (_), use one of the following Search Conditions: • Contains (Exact) • Starts With (Exact) • End With (Exact)
Case Sensitive	In DB2 connections, this property is set to **True** by default.

3. To create custom SQL, select the SQL tab and then select the **Enable custom SQL** option. Modify the existing SQL. (The existing SQL reflects the criteria you specified in the Properties tab).
4. Select the Preview tab to review the list of databases or objects that are selected based on your filter criteria.

To create a Named filter

1. Click ▽ and create a Detailed filter (see above).
2. Enter a name for the filter and click 💾. Creating a name for the filter allows you to use it on other database connections.

Creating a new Quick filter from a current filter

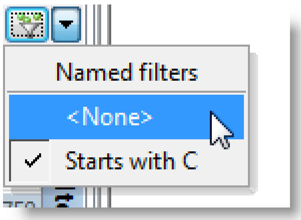
- If you apply a filter and then enter a search string into the object filter field *without* completely clearing the current filter in the field, the new search string is applied, as well as the properties specified for the old filter.
- If you apply a filter and then *completely clear* the object filter field before entering a new search string, all properties return to the default values.

  Note: The Search Condition also returns to the default value. This default is either the Search Condition you specified the last time you created or modified a filter or it is the new default you specified in Options.

Use Filters

To use Named filters

- To apply a Named filter, click the down-arrow beside ▽ and select the filter from the drop-down list.
- To clear a Named filter, click the down-arrow beside ▼ and select **<None>**.



Named filters

<None>

✓ Starts with C

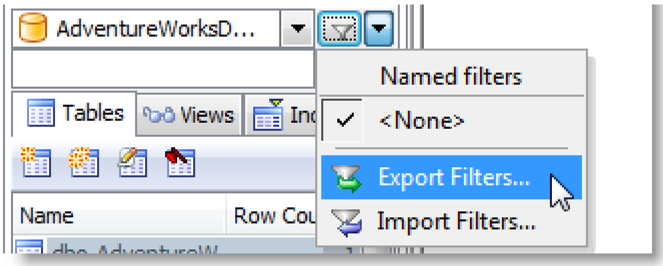
- To determine if a filter is applied, observe the filter icon.

Icon	Indicates
▽	No filter is applied (all databases/objects display).
▼	A filter is applied.

Import/Export Filters

You can export all named filters and you can export all filters defined for the current connection.

- To export Named filters or all filters defined for the current connection, click the down-arrow beside in the database field and select **Export Filters**.



- To import filters previously exported from Toad, click the down-arrow beside in the database field and select **Import Filters**. Browse to and select the filter file (.tfp) to import.

Tip: Click in the Database Explorer (Viewer) toolbar to clear all database, object, and data filters.

Tutorial: Understand a Table Using the Database Explorer

Scenario

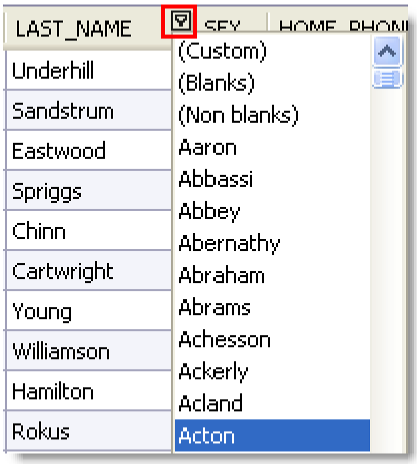
You need to build a query listing customers and want to know more about the table before writing the query.

Note: This exercise uses the Toad Sample Database. This connection does not have the relationship and script tabs. The tabs available can differ depending on the type of connection.

Note: This topic focuses on information that may be unfamiliar to you. It does not include all step and field descriptions.

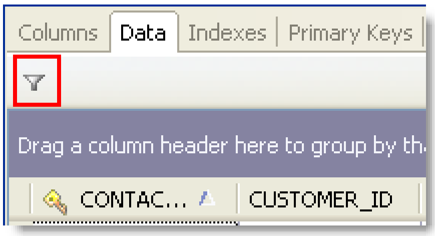
To understand a table using the Database Explorer

1. Connect to the Toad Sample (Access) database in the Navigation Manager.
2. Right-click the CONTACT table and select **View Details**.
3. View information for the following:
    a. Click the Indexes tab and view the index information.
    b. Click the Primary Keys and Foreign Keys tabs to view relationships.
4. Click the Data tab.
5. Place the cursor over the LAST_NAME column, click , and select *Acton*.



The only row that now displays is the one with Acton as the last name.

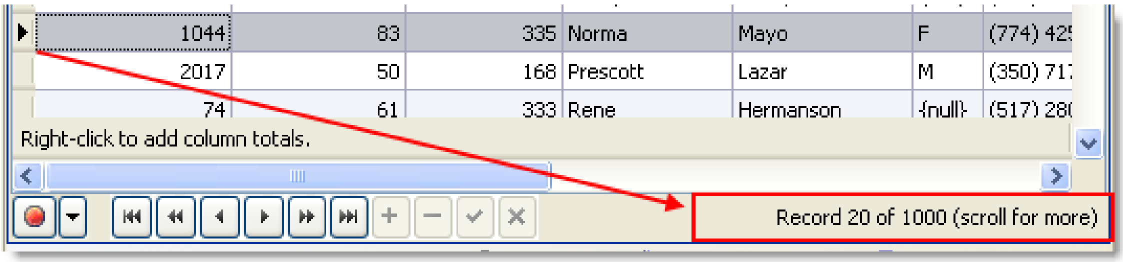
6. Click to the right of the LAST_NAME column again and select *All* to clear the filter.
7. Click the CONTACT_ID column to sort the rows in ascending or descending order.
8. Click in the top left of the Data tab.



9. Create the following filter in the Where Clause tab:

   CONTACT_ID > 2000

   The filter string displays at the bottom on the grid. Click ☒ to remove the filter.
10. Select a row of data in the grid. Notice that the row number changes in the bottom right of the status bar:

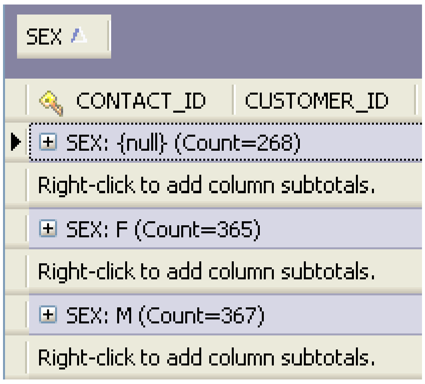


1044	83	335	Norma	Mayo	F	(774) 42
2017	50	168	Prescott	Lazar	M	(350) 71
74	61	333	Rene	Hermanson	{null}	(517) 28

Right-click to add column totals.

Record 20 of 1000 (scroll for more)

11. Under the SEX column, right-click the area below the last row in the grid that says "Right-click to add column totals" and select **Count**. The total number of rows that have either a M or F value display.
12. Click to the right of the SEX column and select *F*. The count value updates based on the new filter.
13. Click to the right of the SEX column again and select *All* to clear the filter.
14. Select the SEX column header and drag it the area above the columns that says "Drag a column header here to group by that column":



SEX

CONTACT_ID	CUSTOMER_ID

⊞ SEX: {null} (Count=268)

Right-click to add column subtotals.

⊞ SEX: F (Count=365)

Right-click to add column subtotals.

⊞ SEX: M (Count=367)

Right-click to add column subtotals.

You can expand each group to view the row data.

15. Switch the default view:
    a. Right-click within a group or row on the grid and select **Show | Card View**.
    b. Scroll through some records.
    c. Click **Customize** in the Card View window.
    d. Click **Filter** beside the ADDRESS_ID column and select *100*. Notice how the card view is filtered, but not the grid view.
    e. Close the Card View window.
16. Edit a row:
    a. Select a row in the FIRST_NAME column and edit the value.
    b. Press ENTER to update the row.
    c. Go to the BIRTH_DATE column and use the date picker to change the date.

       Each data type have an appropriate editor.

       Note: If you are using Oracle or a database that does not have auto commit on, you will need to press Commit at the left hand bottom of the window.
17. Update the number of rows in in the data:
    a. Select **Tools | Options | Database | General** from the menu.
    b. Enter 50 in the **Number of rows to initially fetch in data tab**.
    c. Click **OK**.
    d. Press F5 in the data grid. Notice the bottom right status bar now says "1 of 50 (scroll for more)".

       The option you set controls the number of rows fetched in the Data tab.
    e. Click the scrollbar and hold down to retrieve more.
    f. Right-click the grid and select **Read All Rows**.

Search for Objects

You can quickly locate an object rather than browsing through a list of objects in the Database Explorer orObject Explorer. Object Search locates the following:

- Names of objects for a string.
- Text in objects that include source code, which typically includes procedures, functions, triggers, data types, and views. However, the database provider determines which objects include source code.
- Variable names and comments in the object's source code.
- Column names for tables and views.

To search for objects

1. Click on the toolbar (F4).
2. Specify search criteria and click **Search**.

   Note: To append additional search results to the current search, click **Append** instead. If you click **Search**, new results replace the previous search results in the data grid.

Tips:

- To remove objects from the search results, right-click the objects in the data grid and select **Remove Items**. This is useful if you want to narrow the list of results to relevant objects.This does not remove the objects from your database.
- To create a DDL script from the search results, right-click the data grid and select **Create DDL Script**. You can then paste the results into the Editor.
- To perform any actions available for the object, such as creating, altering, dropping, exporting, and generating reports or SQL, right-click the results in the data grid and select an option.

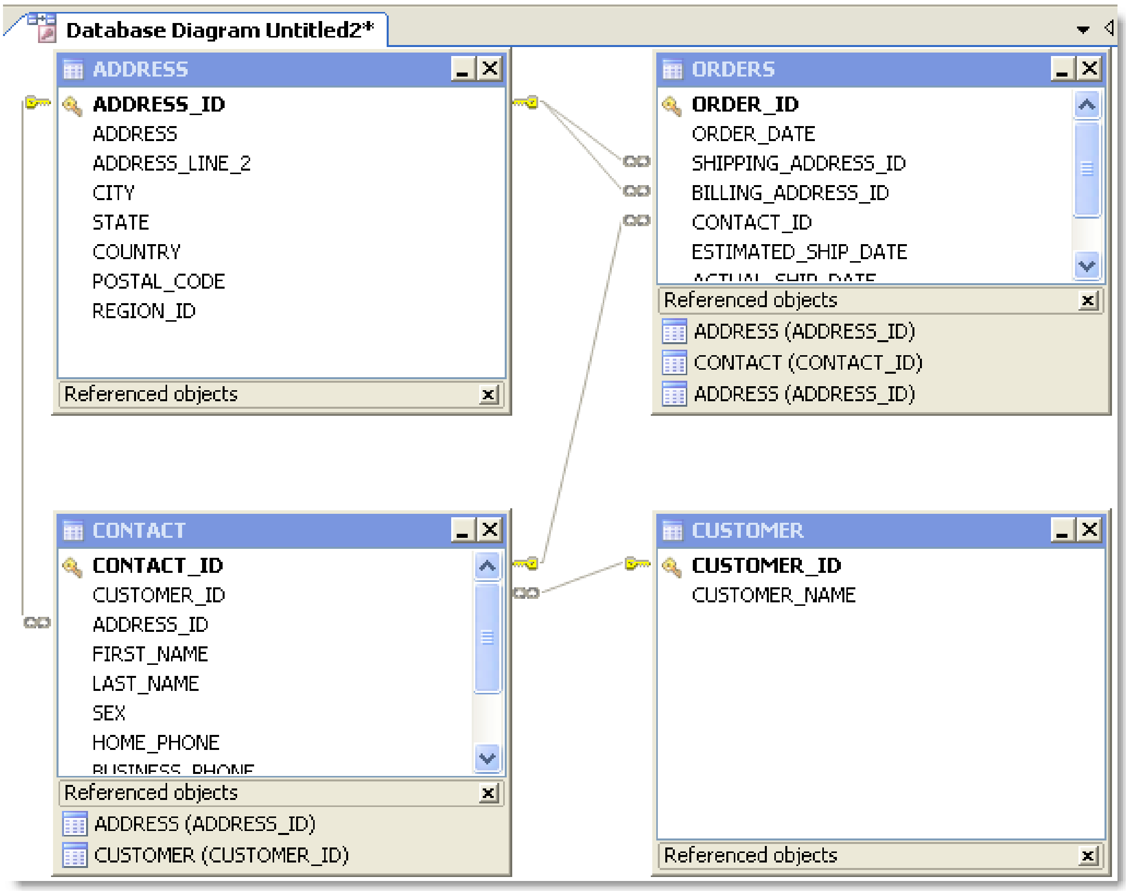
Tutorial: Create a Database Diagram to Use as a Query Template

Scenario

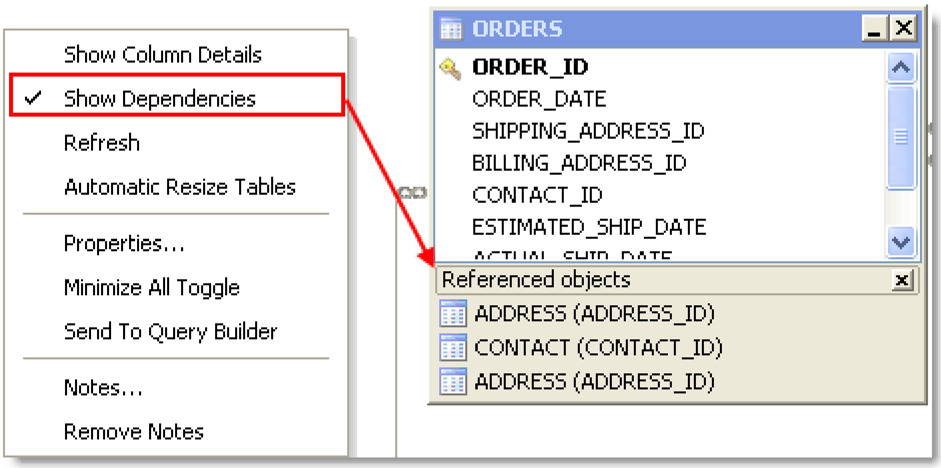
Your company creates several daily and weekly reports regarding customer sales. You need to create a Database Diagram that can be used as a template for these queries.

To create a Database Diagram to use as a query template

1. Connect to the Toad Sample (Access) database in the Navigation Manager.
2. Select **Tools | Diagram**.
3. Drag the CONTACT table in the Object Explorer to the Diagram pane.



4. Right click the Diagram pane and clear **Show Dependencies** to hide the Referenced Objects at the bottom of each table/view, as illustrated in the following screen capture.



5. Right-click the ORDERS table and select **Notes**.

6. Enter the following note in the window for other users:

   The shipping and billing ADDRESS_ID foreign key relations are not needed in most queries and should be hidden when building a query.

7. Because it is against company policy to display sex and a home phone number in reports, right click on each of these columns in the CONTACT table and select **Hide Column**.

   Tip: If you have a large number of columns you need to hide, you can right-click a table and select **Manage Hidden Columns** to quickly select them from a list.

8. Click **Add to Project** from the wizard bar at the bottom of the window and enter *DBDiagram_Test* asthe name for the file. This adds the file to the current project in the Project Manager under a Diagrams folder.
9. Close the Database Diagram window and open the Project Manager window.
10. Expand the Diagrams folder and double-click *DBDiagram_Test* to reload the file in a new Database Diagram window.
11. To share your project, hidden columns, and notes with colleagues, complete the following:
    a. Select **Tools | Options | Environment | Network Share**, and navigate to a shared drive where you want to store the Project Manager file.
    b. Select **Database | Cache** from the open Options window and set the following options:

Object An notation Cache	Description
Share object annotations	Select this checkbox and enter a shared drive location where users can access this file.
Share hidden column settings	Select this checkbox.

    c. Provide the location of the Network Share and Object Annotation Cache settings to colleagues who need to use this diagram.

Browse Data in Related Tables

Use the Master Detail Browser to browse data in parent tables and drill-down to data in related child tables. This is useful when you need to analyze or edit data. For example, suppose you need to view a list of all Nick's Flix stores and the inventory of comedy movies at each store. Using the Master Detail Browser, you specify the table that contains the list of stores as the parent table and specify the table that contains comedy movies as the child table to quickly locate comedies and analyze the inventory.

To browse tables using the Master Detail Browser

1. Select **Tools | Master Detail Browser** (ALT+T+A).
2. Drag one or more tables from the Object Explorer, Object Search, or Project Manager window to the Diagram pane at the top of the Master Detail Browser. Press CTRL+click to select multiple tables at once.

   Notes:

   - A line between tables indicates any existing relationships between the selected tables.
   - At least one parent table (table without any join arrows pointing to it) must be added to the window or an error message displays when returning results.
3. Join Columns.
4. Select the checkbox beside each column you want to include.

   Note: Any columns included in a join are automatically included in the results.
5. Click ▶ (F9) to retrieve data for the parent table initially. To view data in a child table, click + to the left of each row.

   or

   Click ➜ to retrieve data in the parent and child tables. If there is more than one child table, a tab displays for each child table.

6. From the data grid, you can:
   - Edit the data and commit your changes to the database.
   - Right-click the data grid and select **Send To | *option*** to add the data to a report, chart, or pivot grid.

Tips:

- Click to align tables and snap them to the grid.
- Select the parent table from the drop-down list in the toolbar to view data for a parent table if there are multiple parent tables.
- Click **X** on the table you want to remove in the Diagram pane to remove a table and any conditions (e.g., join, where clause) associated with the table.

4

Query

Tutorial: Visually Build a Query

The Query Builder enables you to create a query without writing or editing SQL statements. Even if you are familiar with SQL, the graphical interface makes it easier to create relationships and visualize the query.

Note: This procedure does not cover all of the possible steps of visually building a query. It only covers the steps required to build a query for the scenario. See Building Queries Visually for more information.

Scenario

Your company wants to improve its global sales, starting with countries where it has a customer base but generally low sales. You need to create a query that identifies countries with less than $500,000 in sales for each of the last four fiscal years.

Note: This scenario uses the SH (Sales History) sample schema that comes with Oracle 10g so you can follow the procedure.

To visually build a query based on this scenario

1. Connect to the database and select **Tools | Query Builder | Query Builder**.
2. Select the *SH* schema in the Object Explorer.
3. Drag the following tables from the Object Explorer to the Diagram pane: TIMES, SALES, CUSTOMERS, and COUNTRIES.

   Tips:

   - Select **Query Builder | Arrange Tables** (ALT+Q+A) to have Toad arrange the tables in the Diagram pane.
   - You can press CTRL and select a table, and then drag the table into the Editor pane. Click the list to create statements.
   - You can also press CTRL and select all of the tables in the Object Explorer, and then drag them into the Diagram pane. Toad automatically arranges the tables.

4. Add the following columns to the query by clicking each column:

Table	Column
COUNTRIES	COUNTRY_NAME
TIMES	FISCAL_YEAR
SALES	AMOUNT_SOLD

   Tips:

   - To view details for a table in the Diagram, click (F4). If a table is not selected, details for the last selected table display.

- To remove a column from the query, click ☒ or drag column off grid.

5. Complete the following steps to group the data and sum the sales:

    a. Select the **Group By** field in the COUNTRY_NAME column, and then click +. The **Group By** field is in the Criteria pane, which is below the Diagram pane.

    b. Select the **Group By** field in the FISCAL_YEAR column, and then click +.

    c. Select the **Aggregate Function** field in the AMOUNT_SOLD column and select *Sum*. This sums all of the sales per fiscal year per country.

    Note: If you select a group by clause or aggregate function for a column, the rest of the columns in the query must also have a group by clause or aggregate function.

6. Complete the following steps to select the fiscal year range:

    a. Select the **Where Condition** field in the FISCAL_YEAR column and click ⋯. The Where Condition window displays.

    b. On the **Form** tab, select *BETWEEN* in the **Operators** field.

    c. Select *1998* in the first **Constant** field.

    d. Select *2001* in the second **Constant** field and click **OK**.

    Note: The SH schema does not have current data. The 1998-2001 years provide the best data for the purposes of this scenario.

7. Complete the following steps to select countries with less than $500,000 in sales:

    a. Select the **Having Condition** field in the AMOUNT_SOLD column and click ⋯. The Having Condition window displays.

    b. On the **Form** tab, select < in the **Operators** field.

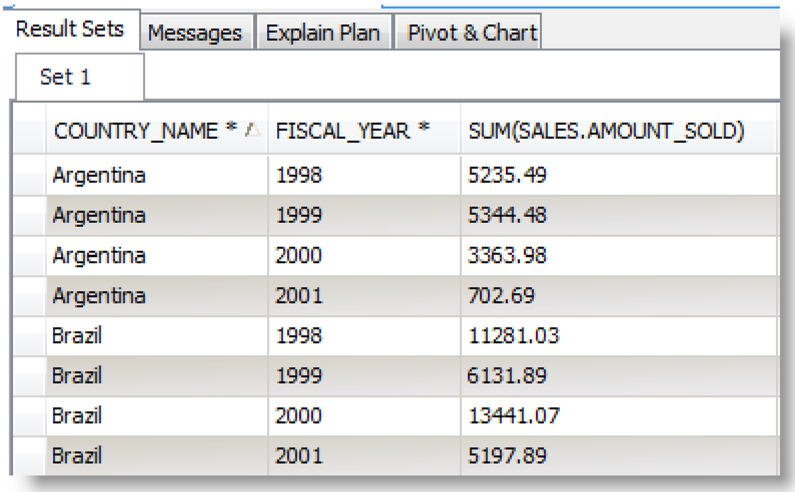
    c. Select *Sum* in the **Aggregate** field.

    d. Enter *500000* in the **Constant** field and click **OK**.

8. Select *Ascending* in the **Sort** field of the COUNTRY_NAME and FISCAL_YEAR columns.

9. Click ▷ Execute SQL.

    The query results display in the Results tab. See "View Result Sets" on page 50 for more information.



COUNTRY_NAME *	FISCAL_YEAR *	SUM(SALES.AMOUNT_SOLD)
Argentina	1998	5235.49
Argentina	1999	5344.48
Argentina	2000	3363.98
Argentina	2001	702.69
Brazil	1998	11281.03
Brazil	1999	6131.89
Brazil	2000	13441.07
Brazil	2001	5197.89

10. To save the query, right-click the Query Builder tab and select **Save File**.

    You can save a query and any results sets in a Toad editor file (.tef). This file format is useful for building scripts that have large result sets that you do not want to continue executing or for saving results when you have not finished building a script.

See Tutorial: Build a Subquery for a tutorial on creating a subquery.

Create Cross-Connection Queries

Use the Cross-Connection Query Builder or the Cross-Connection Editor to create a query that combines data from multiple, dissimilar databases. For example, you can create a query that joins managers from a SQL Server database to their employees in an Oracle database. (This is also often called a heterogeneous or cross-database query.) Toad uses an enhanced query engine to implement advanced execution processes that are specially designed to improve the performance of cross-connection queries.

Before creating a cross-connection query using an Oracle, Teradata or MySQL ODBC connection, please see the special considerations listed at the end of this topic.

To build a cross-connection query visually

1. Select **Tools | Build Query | Cross-Connection Query Builder**.

   Note: A blue banner along the right side of the Diagram pane indicates that you are in Cross-Connection Query mode.

2. Drag tables and views from the Object Explorer to the Diagram pane.

3. Complete the query in the same manner as you would a non-cross-connection query, but note the following exceptions.

Object Explorer Feature	In Cross-Connection Query Mode...
Options	Connection Treelist is the only view option available. In this option, connections, databases, tables, and views are displayed in a treelist in the Object Explorer window.
Filters	Filters are especially useful in cross-connection queries to reduce the number of objects displayed in the treelist in the Object Explorer window.

Note: Review the "Considerations and Limitations of Cross-Connection Queries" section below.

To compose a cross-connection query in the editor

1. Select **Tools | Edit | Cross-Connection SQL Editor**.

   Note: A blue banner along the right side of the Editor pane indicates that you are in Cross-Connection Query mode.

2. Compose the SQL statement in the same manner as you would a non-cross-connection query, but note the following exceptions.

SQL Editor Task	In Cross-Connection Query Mode...
Naming tables and views	To reference a table or view, you must use a cross-connection fully-qualified name. The format for a cross-connection fully-qualified name depends on the database. The following is an example using the SQL Server format: *<database connection string>.<database>.<schema>.<table>* *'sqlserverconn'.database.dbo.table*

SQL Editor Task	In Cross-Connection Query Mode...
	Note: For the *database connection string*, use the connection string displayed in the Navigation Manager window for each connection (or in the Name field in Connection Properties). **Tip:** Copy and paste a SQL statement from a single connection editor to a cross-connection editor and Toad automatically changes the syntax of the statement to include the cross-connection fully-qualified names.
Using alias names	You must use alias names for tables and columns in a cross-connection query.
Using code completion	In Cross-Connection Query mode, the scope of available options in code completion expands to include the databases, tables, and views for all the open database connections. **Tip:** To expand/collapse a database node in the code completion list, use CTRL+left arrow and CTRL+right arrow.

Tips:

- In a cross-connection code completion list, the selected database displays a descriptive tooltip containing database and connection information.
- Use the Database Diagram tool to create and save cross-connection relationships, and then send the tables with their relationships already created to the Query Builder.

Considerations and Limitations of Cross-Connection Queries

Review the following considerations and limitations before creating a cross-connection query.

Consideration/Limitation	Description
ODBC driver support	An ODBC driver can be selected for a single connection through each connection's Properties dialog or globally through **Tools | Options | Database**.
Using an Oracle connection in a cross-connection query	To use an Oracle connection in a cross-connection query, you must install an Oracle ODBC driver. **Note:** The Oracle Full Install client includes an ODBC driver. However, if you are using the Oracle Instant client (which may not include an ODBC driver), you may need to install an Oracle ODBC driver.
Oracle Direct Connection	You cannot use an Oracle Direct Connection in a cross-connection query.
Using a Teradata connection in a cross-connection query	To use a Teradata connection in a cross-connection query, you must install a Teradata ODBC driver.
Using a MySQL ODBC connection in a cross-connection query	To use a MySQL ODBC connection in a cross-connection query, you must specify a database for the connection.
Result sets	You cannot edit result sets.
Oracle Long Data Type	Cross-connection query does not support Oracle long data types or Oracle LOBs.

Reverse Engineer Queries

One way to build a query is to begin adding columns and creating joins in the Query Builder, and then send the query to the editor to add more complex functionality, such as a UNION join. You can also send a query from the editor to the Query Builder. For example, you inherited a large query when an employee left and you need to edit it to resolve an issue. After looking at the query in the editor for some time, you still cannot decipher it. So, you right-click the statement and select **Send to Query Builder**. Now that you can visually see how the tables are referenced, you can identify and resolve the issue.

Considerations and Limitations in the Query Builder

Consider the following when reverse engineering a query using the Query Builder:

General	**Description**
Single statement support	The Query Builder only supports one statement at a time. If you add multiple statements to the Query tab, only the first statement is used. If you make any changes to the statement on the Diagram tab, the other statements are lost.
Quote identifiers	If you selected the **Quote Identifiers** checkbox in **Tools | Options | Database | General** and generate a query from the Query Builder in SQL Server, the query cannot be reverse engineered.
ODBC support	You must use ANSI SQL for the query.
Excel support	Copying a generated query and attempting to reverse engineer that query results in an error. This issue occurs because the parser defaults to Oracle syntax, which does not understand quotation marks for fully qualified object names. To avoid this issue, clear the **Use fully qualified object names** and **Use fully qualified column names** checkboxes in **Tools | Options | Database | Query Builder** or manually edit the query.
Non-ANSI joins (Oracle and SQL Server)	When sending a query from the Editor to the Query Builder, Toad automatically uses ANSI joins. If you use Where clause joins, click to disable ANSI joins. This converts the statement to the correct join.
Where Condition	**Description**
Where condition does not display in the Criteria grid after sending it to the Query Builder from the Editor	If you send a query that contains a Where condition from the editor to the Query Builder, it displays in a Global Where clause bubble on the Diagram pane instead of in the **Where Condition** field in the Criteria grid. You can double-click the Global Where Clause bubble in the Diagram pane or can click to edit it. You can also use the **Where condition** fields in the Criteria grid to add new Where conditions. Although they display in the grid, they are added to the Global Where clause in the Query tab.
Where condition created in the Criteria grid of the Diagram tab disappears	If you define a Where condition in the Criteria grid and edit any part of the query in the Query tab, the Where condition is removed from the Criteria grid and placed in a Global Where Clause bubble. You can double-click the Global Where Clause bubble in the Diagram pane or can click to edit it.
Having Condition	**Description**
Having condition does not display in the Criteria grid	If you send a query that contains a Having condition from the editor to the Query Builder, it displays in a Global Having clause bubble on the Diagram

after sending it to the Query Builder from the Editor	pane instead of in the **Having Condition** field in the Criteria grid. You can double-click the Global Having Clause bubble in the Diagram pane or can click to edit it. You can also use the **Having condition** fields in the Criteria grid to add new Having conditions. Although they display in the grid, they are added to the Global Where clause in the Query tab.
Having condition created in the Criteria grid of the Diagram tab disappears	If you define a Having condition in the Criteria grid and edit any part of the query in the Query tab, the Having condition is removed from the Criteria grid and placed in a Global Having Clause bubble. You can double-click the Global Having Clause bubble in the Diagram pane or can click to edit it.
Query Builder Diagram Tab	**Description**
Comments and code regions removed after making changes in the Diagram tab	When reverse engineering a statement from the editor or the Query tab and switching to the Diagram tab, if the statement contains a block comment or code region and you make a change in the Diagram tab, the comment or code region is removed from the statement.
Same column cannot be used in the criteria grid	You cannot use the same column more than once in the criteria grid. Even if you create a separate alias for the second use of the column, that column is created as a calculated field
Execute with different query in the Diagram and Query tabs	If the query in the Diagram tab is different than the query in the Query tab, executing the query executes the SQL in the Query tab instead of the Diagram.
Save with different query in the Diagram and Query tabs	The Query Builder saves both the statement and diagram in the same file. The contents of the statement and diagram can be different without losing changes.

To reverse engineer a query

1. Right-click a statement in the Editor, and select **Send to Query Builder**.See "About Editing SQL" on page 46 for more information.

   or

   Click in the Query Builder to send a statement to the Editor. Skip the remaining steps and continue to edit the query. See "About Editing SQL" on page 46 for more information.

   Troubleshooting: If the query cannot be modeled in the Diagram tab, a message displays and the statement opens in the Query tab. This usually occurs because the graphical diagram cannot support some functionality in the statement. You can view an explanation for this in the Output window or by hovering over the syntax with a red underline () in the Query tab. To continue, correct the error and click to apply the change and open the Diagram tab.

2. Continue to build your query visually using the diagram.

3. Select the Query tab and make any additional edits. Indicators for an edited statement are as follows:

🟩	Inserted lines
⬜	Modified or removed lines
🟨	Modified characters
Use the toolbar + ~ ab in the lower-right corner of the Query tab to modify the color used for each indicator.	

> **Caution:** If you manually enter or update the statement in the Query tab, you must click to model the query in the Diagram tab. If you do not do this and make additional changes in the Diagram tab, any changes you made in the Query tab are lost.

4. Click ▶ to execute the query.

Tip: If you have multiple statements in the Editor and you want to send only one to the Query Builder, highlight the statement, then right-click it and select **Send to Query Builder**.

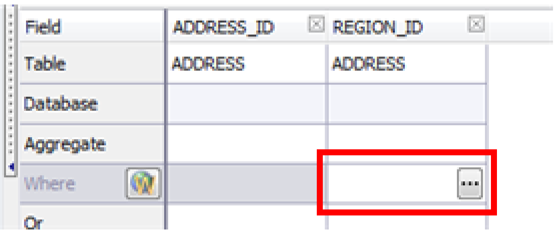
Tutorial: Build a Subquery

In this tutorial you will create a subquery to use to filter your selection of address IDs to only those addresses from specific regions.

To build a subquery

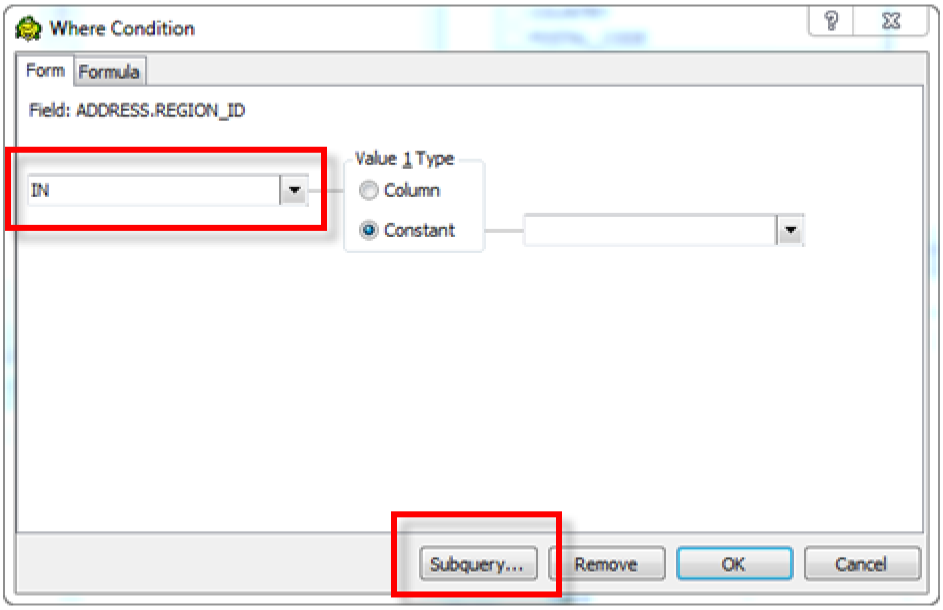
1. Connect to the Toad Sample (Access) database in the Navigation Manager.
2. Select **Tools | Query Builder | Query Builder.**
3. Drag the ADDRESS table to the Diagram pane.
4. Click these column names in the ADDRESS table to add the columns to the query:
   - ADDRESS_ID
   - REGION_ID
5. To add a subquery:

   a. Select the **Where** field in the REGION_ID column and click …. The Where Condition editor displays.



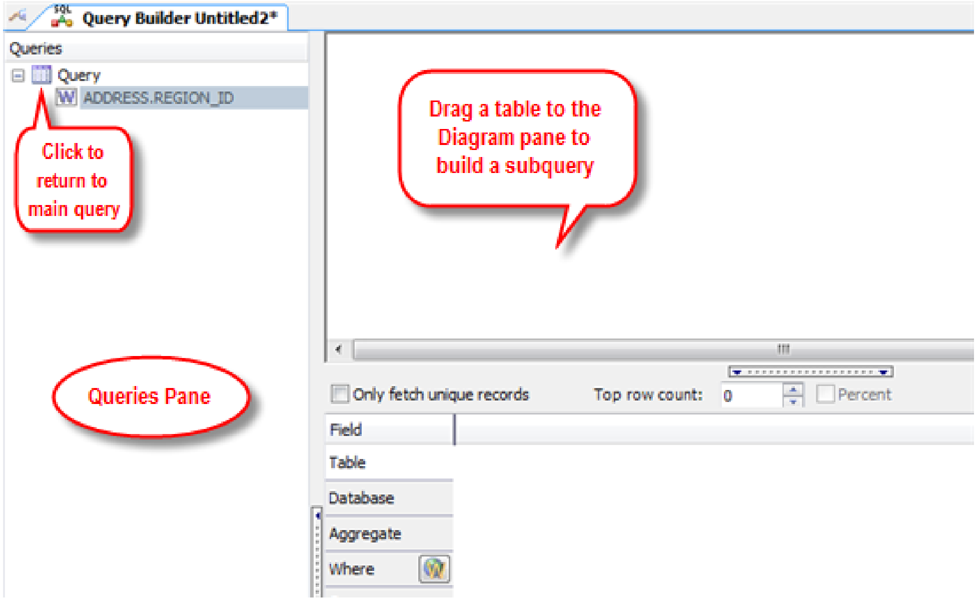
Field	ADDRESS_ID	REGION_ID
Table	ADDRESS	ADDRESS
Database		
Aggregate		
Where		…
Or		

   b. Select the *IN* operator from the list, and then click **Subquery**.



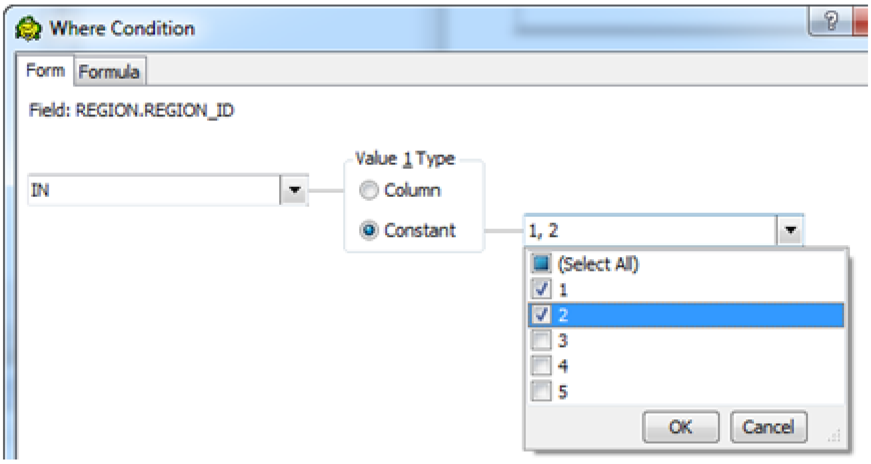
c. A child Query Builder window displays for you to use to build the subquery.

- The child (subquery) Query Builder window includes a Diagram pane, a Criteria pane, and a Queries pane for navigating back and forth between subqueries and the main query.



d. Drag the REGION table to the Diagram pane and select the REGION_ID column.

e. Select the **Where** field in the REGION_ID column and click ⋯. In the Where Condition editor, select the *In* comparison operator, select **Constant**, and select the values *1* and *2*.



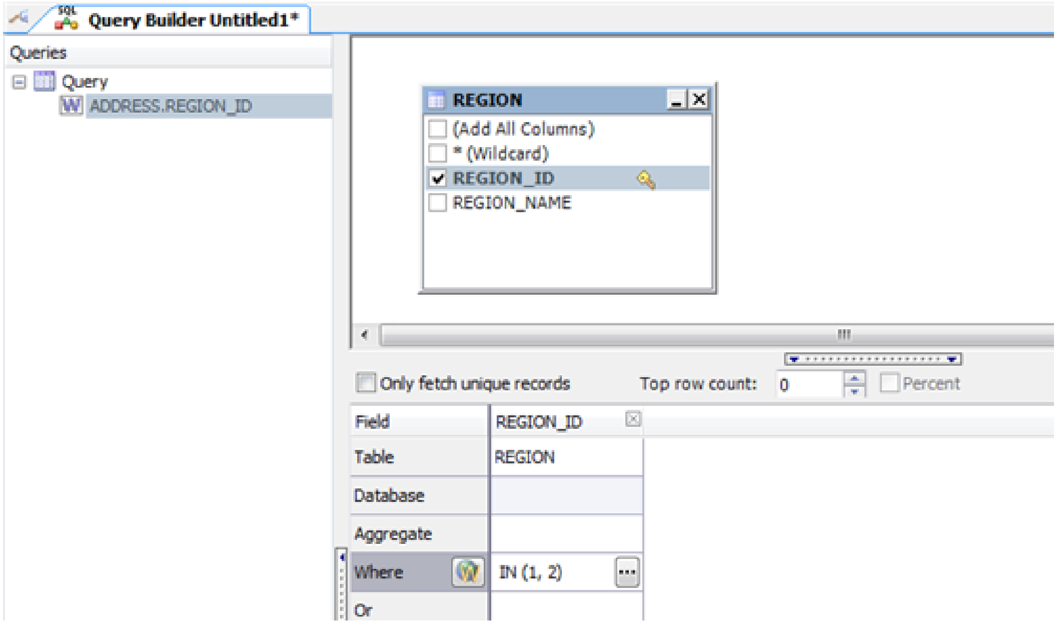
Field: REGION.REGION_ID

IN

1, 2

■ (Select All)
☑ 1
☑ 2
□ 3
□ 4
□ 5

f. Your subquery window should look something like this:



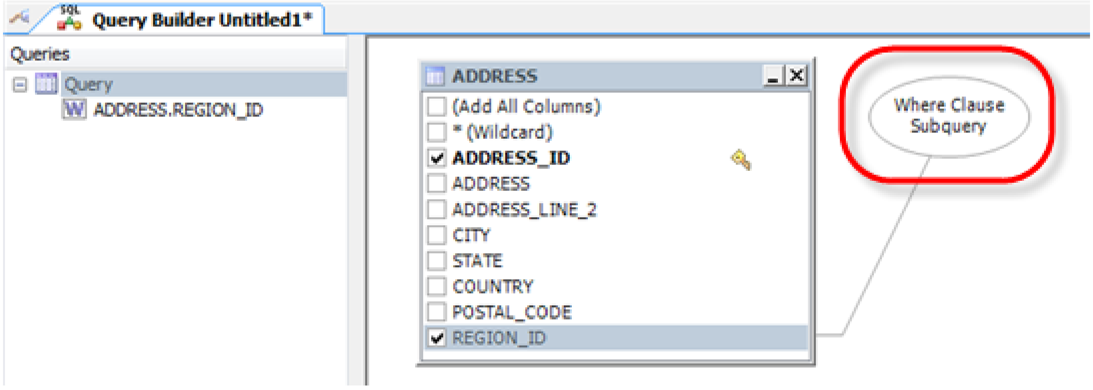
□ (Add All Columns)
□ * (Wildcard)
☑ REGION_ID
□ REGION_NAME

□ Only fetch unique records Top row count: 0 □ Percent

Field	REGION_ID
Table	REGION
Database	
Aggregate	
Where	IN (1, 2)
Or	

g. Click Execute SQL to execute the subquery. Select the Results tab to review the result set for the subquery.

6. Click the Query node in the Queries pane to return to the main query window. A call-out in the main query Diagram pane indicates that the statement contains a subquery and identifies the column it is on. Hover over the call-out to view the subquery statement.



□ (Add All Columns)
□ * (Wildcard)
☑ ADDRESS_ID
□ ADDRESS
□ ADDRESS_LINE_2
□ CITY
□ STATE
□ COUNTRY
□ POSTAL_CODE
☑ REGION_ID

7. Click Execute SQL to execute the main query.

About Editing SQL

The editor combines a powerful SQL and procedure editor into a single interface.

Notes:

- See "Editor Shortcut Keys" on page 48 for more information about keyboard shortcuts used in the Editor.
- Toad has implemented a large script mode to improve performance when opening files larger than 2 MB. By default, when you open a large file, some features such as word wrap, syntax highlighting, and the Script Map are disabled in the Editor. When a large script loads, a message indicates that you are in large script mode in the right margin of the Editor. You can cancel large script mode by clearing the option in **Tools | Options | Editor | General**.
- To show the Script Map, **Script Results** or **Group Execute** panes for the current Editor window select **Editor | Windows | <*pane*>**.

To edit SQL

1. Select **Tools | Edit | SQL Editor**.

   Note: You can query the worksheet from the Editor by entering a dollar sign ($) followed by a cell range to limit the results similar to the following statement:

   ```
   Select * from (MyWorksheet$A1:C10)
   ```

   You can also query the entire worksheet by entering a dollar sign ($) after the worksheet name; however, this returns empty rows and columns in the worksheet.

2. Review the following to help enter SQL:

   - Code completion
   - Script Map
   - Code regions
   - Code snippets
   - Toad views
   - SQL recall
   - Add/remove application code
   - View Objects at Cursor
   - Macros to record keystrokes

3. Click to format the code.

4. Right-click a block of code in the Editor to see additional actions available in the context menu. For example, you can turn a line into comment, uppercase or lowercase it, surround with snippet, set numbered bookmark, collapse or expand nodes, send to Query Builder and etc. You can also right-click an object (e.g. a table) and manage it as if in the Object Explorer.

5. Hold Alt + left-click and drag to select vertical blocks of text in Editor.

6. Select the block of code in the Editor, and click to validate the syntax.

   Note: Toad undoes any changes to the object and data in the database after executing the script (notice that the Result Sets tab is empty). To modify the object and data, you need to execute the script.

7. Click **Fetch *n* rows** to specify the number of rows to initially retrieve when executing the query. You can set a default value for this option in **Tools | Options | Database | General**.

8. Click to execute SQL on the current server or multiple servers at one time. In the Editor toolbar, the execute SQL buttons are grouped under . The button will retain the function of the last execute action for the active Editor window. If you switch to another Editor window, the button will change to reflect the last action for that window.

9. Select the block of code in the Editor and click to create a stored procedure. Stored Procedure creation dialog automatically recognizes procedure body and declared parameters.

Tips:

- Toad automatically saves a backup copy of any modified editor file in the Application Data Directory to avoid losing your work. When you restart Toad after an unexpected close, Toad checks the timestamp on the backup file against the original file, and opens the file that has the latest timestamp. The document recovery option is in **Tools | Options | Environment | General**.
- You can save the current connection, SQL script, executed results, filters, and group execute connections/results (if enabled) in a Toad editor file (.tef). This file format is useful for building scripts that have large result sets that you do not want to continue executing or for saving results when you have not finished building a script.
- Editor supports floating window mode.
- The script is automatically checked for syntax errors when executing current statement (F9), checking syntax, changing database, or refreshing the Script Map. The syntax errors are listed in the Script Results pane in the Messages tab.
- If you copy and paste a SQL statement from one editor window to another, Toad automatically rewrites the statement to match the syntax of the connection for the second editor window.

Configure Editor Options

You can specify options for the look and behavior of the SQL Editor by using the Toad Options dialog.

To configure Editor options

1. Select **Tools | Options**.
2. To specify one of the following options, select the corresponding page in the Options dialog.

Option Description	Page in Options Dialog
Display the tabs for the current connection only when using Tabbed windows	Select **Environment | Interface**
Change background color of the Editor	Select **Editor | General**
Use connection color as background color	Select **Editor | General**
Position Editor tab headers vertically	Select **Environment | Interface**
Set options to display line numbers, line modifications, and general text formatting	Select **Editor | General**
Specify options to automatically replace typed text with the correct substitution text	Select **Editor | Auto-replace**
Customize code formatting options including headers and layout characteristics	Select **Editor | Formatter**

Click here to view a video of this feature.

Editor Shortcut Keys

You can assign individual shortcut keys in Keyboard options. You can also customize Toad to use the shortcut keys of existing applications such as Toad for Oracle, SQL Navigator, or Query Analyzer in the Configuration Wizard (**Tools | Configuration Wizard**).

> **Caution!** If you have some global access keys assigned, make sure that Toad shortcut keys do not coincide with them. When using such duplicating shortcut keys in Toad, the third party application commands, assigned globally, will be performed instead of ones assigned by Toad.

The following table describes the default shortcut keys in Toad.

Frequently Used Functions	**Description**
CTRL+PERIOD	Displays the code completion list
F9	Execute current statement
F5	Execute all statements, in order, one at a time
CTRL+MINUS	Comments the selected line
CTRL+SHIFT+MINUS	Uncomments the selected line
CTRL+J	Collapses all code regions
CTRL+M	Expands all code regions
ALT+PAUSE	Stops script execution
Bookmark	**Description**
CTRL+SHIFT+0-9	Sets a numbered bookmark at the selected line
CTRL+0-9	Goes to the numbered bookmark
Case	**Description**
CTRL+L	Converts the selected text to lowercase
CTRL+U	Converts the selected text to uppercase
Clipboard	**Description**
CTRL+C	Copies the selected text to the clipboard
CTRL+X	Cuts the selected text for pasting in the clipboard
CTRL+V	Pastes text from the clipboard
Code Completion	**Description**
CTRL+PERIOD	Displays code completion list
ENTER	Selects an item and close the code completion list
CTRL+SPACE	Selects an item without closing the code completion list
CTRL+<right arrow>	Expands a node in the code completion list
CTRL+<left arrow>	Collapses a node in the code completion list
Comment	**Description**
CTRL+MINUS	Comments the selected line
CTRL+SHIFT+MINUS	Uncomments the selected line
Cursor	**Description**

HOME	Moves the cursor to the beginning of the line
CTRL+HOME	Moves the cursor to the beginning of the file
END	Moves the cursor to the end of the line
CTRL+END	Moves the cursor to the end of the file
Debugger	**Description**
SHIFT+F12	Step from cursor
F7	Step into
SHIFT+F7	Step out
F8	Step over
F11	Run to breakpoint
F12	Run to cursor
CTRL+B	Set breakpoint
CTRL+W	Add watch at cursor
Delete	**Description**
BACKSPACE	Deletes the selected text (if text is selected); otherwise, deletes the previous character
CTRL+BACKSPACE	Deletes text to the beginning of the previous word
CTRL+SHIFT+DELETE	Deletes the entire script
Find and Replace	**Description**
CTRL+F	Finds text
F3	Finds the next occurrence of text
SHIFT+F3	Finds the previous occurrence of text
CTRL+G	Goes to line number
CTRL+I	Performs an incremental search
CTRL+SHIFT+I	Performs a reverse incremental search
Indent	**Description**
TAB	Indents the selected line
SHIFT+TAB	Outdents the selected line
Macros	**Description**
CTRL+SHIFT+R	Starts/Stops recording a macro
CTRL+SHIFT+P	Plays the recorded macro
Miscellaneous	**Description**
F4	Invokes objects search
F4+SHIFT	Describes the object next to the cursor
F6	Toggles between the Editor and Data Grid
CTRL+R	Show/hide Results pane
CTRL+D	Sends SQL to Query Builder
SHIFT+F8	Recalls the previous SQL statement
CTRL+↑	Scrolls up the window.

CTRL+↓	Scrolls down the window
<auto-replace token>+SPACE	Automatically replaces a misspelled word, inserts symbols, and expands text. **Note:** You can set autoreplace in **Tools | Options | Editor | Autoreplace**.
CTRL+SPACE	Displays the Code Snippets folder browser
<code snippet shortcut name>+CTRL+SPACE	Inserts the code snippet template
CTRL+DELETE	Sets the column value for the current cell to Null in the data grid

View Result Sets

The Result Sets tab displays a data grid for executed SQL statements and scripts that return data.

If you occasionally closed the Results pane, you can restore it by selecting **Editor | Windows | Script Results**.

Visual Inspection Grid

You can easily review all (or the selected Result sets) in one Result Sets tab. Left-click and hold Shift or Ctrl to select more than one Result sets of your script. Select All results to review all in one tab.

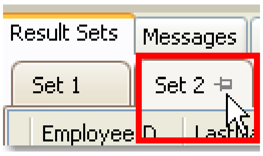
To enable view of all Result Sets in one Tab

- Check **Options | Database | Script results | Enable all results view**

Troubleshoot Data

- If you cannot edit data, click the red icon ● in the lower left corner of the grid, and then click **OK** on the window that displays. You do not need to edit fields in this window.
- By default, auto commit is enabled and any row changes you make are automatically submitted to the database. You can disable this option in **Tools | Options | Environment | Grid**.

Tips for Working with Data

To...	Do this...
Pin a result set so it is not overwritten by a subsequent query in the Editor or Query Builder.	Click the push pin beside the result set:  Result Sets Messages Set 1 Set 2 Employee D LastNa
Set the value of a cell to null.	Press CTRL+DELETE.
Toggle between the Editor and Data Grid.	Press F6.
Add column or group summary values	Right-click the data and select **Show | Summary Footer Panel**. Then, right-click the Summary Footer Panel below the

To...	Do this...
	column you want to add a summary total to and select an option. If you have grouped columns using the Grouping panel (**Show | Group Panel**), you can also add a summary total to grouped records.
Apply a summary function to the data (Sum, Min, Max, Count, Avg).	Right-click below the column you want to apply the summary function (but above the Navigation toolbar) and select an option.
Show or hide columns of data.	Right-click a column heading in the data grid and select **Column Chooser**.
Display data in a report, pivot grid, or chart.	Right-click the data and select **Send To |<*option*>**.
Compare differences between two data sets.	Right-click the data and select **Compare To**.
Display the data in a grid or card view; grid view is useful for viewing summary information at a glance; card view is useful for viewing information in greater detail.	Right-click a record and select **Show |Grid View** or **Card View**.
View multi-line text in the data grid as one-line text separated by spaces.	Right-click the data and select **Show | Multi-line Text as One-line**. When the option is cleared, only the first line of the multi-line text is shown. To see the full text hover a mouse over a data grid or expand it.

Compare Differences in Data

You can compare result sets from the current editor or a different editor in the Database Explorer's Data tab or the editor's Result Sets without having to save one of the sets to a file.

Notes:

- If you want to synchronize data, use Data Compare instead of the Data Diff Viewer.
- You cannot modify files in the Diff Viewer at this time.

To compare differences in data

1. Right-click data in the Editor's result set or Database Explorer's Data tab and select **Compare To | *Set_number***.

   You can also save a result set and later use it as a historical comparison.

2. Review the differences.

   Tips:

   - To switch the contents of the left pane to the right pane, and the contents of the right pane to the left, click ⇄.
   - You can also open files to compare by clicking … beside the drop-down list at the top of each pane.
   - You can show or hide tables that contain equal records, different records, etc., at any time using the toolbar.

Export Data with One Click Export

Use One Click Export to quickly export data to an Excel report. You can also export data and include a *linked query* in the file. A linked query lets you create a "live" Excel report that can be refreshed dynamically. Because a linked query contains the underlying SQL for a report, each time you refresh the report the query executes and retrieves the latest data. The dynamic nature of linked query reports makes them ideal for distribution to users who need to see changes to data on demand without waiting for new static reports to be distributed.

Notes:

- Before you can export to Excel, the Primary Interop Assemblies (PIAs) for your version of Excel must be installed. Typically, the PIAs are installed automatically when you install Microsoft Office.
- Use the Export wizard if you need to specify column delimiters and advanced field options when exporting data.

To export data in one click

1. Select the range of data to export or click any row in a data grid to export all data.
2. Right-click the selected range or data grid and select **Quick Export**.
3. Select an export option. Review the following for additional information:

Icon	Option	Description
	Excel File	Export data to an Excel file. The file is automatically saved under a default name in the default export folder.
	CSV File	Export data to a CSV file (comma separated value file). The file is

Icon	Option	Description
		automatically saved under a default name in the default export folder.
	HTML File	Export data to an HTML file. The file is automatically saved under a default name in the default export folder.
	Excel Instance	Export data to an Excel instance. Data is copied to a clipboard, Excel is launched, and a worksheet is populated with data. You can save the worksheet under a name and folder you specify or save it in the default export folder.
	Excel Instance at active cell	Export data to an active cell in Excel. First open Excel and select a cell. Then select the **Excel Instance at active cell** option.
	Excel Pivot Table	Export data to an Excel pivot grid report. See the Excel documentation for information on creating a Pivot Table report with external data.
	Excel Pivot Table at active cell	Export data to an active cell for an Excel Pivot Table report. First open Excel and select a cell. Then select the **Excel Pivot Table at active cell** option.
	Excel Linked Query	Export the underlying query for data to Excel.
	Excel Linked Query at active cell	Export the underlying query for data to an active cell in Excel. First open Excel and select a cell. Then select the **Excel Linked Query at active cell** option.
	Excel Linked Query with pivot table	Export the underlying query for data to an Excel Pivot Table report. See the Excel documentation for information on creating a pivot grid report with external data.
	Excel Linked Query at active cell with pivot table	Export the underlying query for data to an active cell for an Excel Pivot Table report. First open Excel and select a cell. Then select the **Excel Linked Query at active cell with pivot table** option.
	Excel Column Chart	Export data to an Excel chart instance. Data is copied to the clipboard, Excel is launched, and a chart is created with the data. You can then modify the chart's design and save the Excel file.

Tip: To refresh data in a linked query report, click in Excel's External Data toolbar. This executes the underlying query for the report. See "Share Excel Linked Queries" on page 54 for more information about creating and using linked queries.

Note: If the window containing data does not have the focus when you select a One Click Export option, the following message displays: "Export did not find any results sets to process."

Tips:

- To specify a default export folder, go to **Tools | Options | Environment | Export**.
- To always open a Save File dialog when exporting using One Click Export, go to **Tools | Options | Environment | Export** and select the option. This allows you to always customize the file name.

Share Excel Linked Queries

You can use linked queries to create "live" Excel reports that can be refreshed dynamically. Because a linked query contains the underlying SQL for a report, each time you refresh the report the query is executed and the latest data is retrieved. The dynamic nature of linked query reports makes them ideal for distribution to users who need to see changes to data without waiting for scheduled updates or for new reports to be distributed.

Notes:

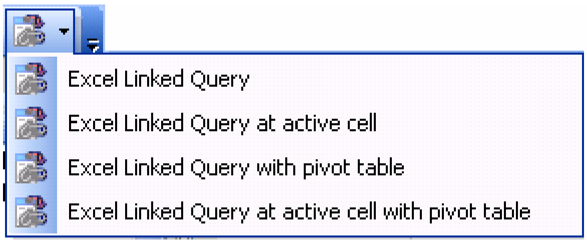
- Any user who opens an Excel linked query must have access to a user name and password for the queried database to refresh the data. If a user does not have access to the database, they can only see the original data in the spreadsheet.
- If the Excel linked query was created from an ODBC connection that uses a Data Source Name (DSN), the DSN must also exist on the computer where the Excel file is opened to refresh the data.

Scenario

You need to create an Excel linked query so that the Regional Sales Manager, can track whether the Sales Representatives in his region meet their monthly and quarterly goals. By providing a linked query, you only have to create the query once, and then the Regional Sales Manager can refresh the data at any time to view the latest sales information.

To create an Excel linked query

1. Connect to the database.
2. Create a query in the Query Builder or Editor.
3. Execute the query to display data results.
4. Select the result set and click one of the following options from the toolbar:


Excel Linked Query
Excel Linked Query at active cell
Excel Linked Query with pivot table
Excel Linked Query at active cell with pivot table

5. Save the Excel spreadsheet.
6. Email or upload the spreadsheet to a shared network folder to share it with colleagues.

To refresh data in an Excel linked query

1. Open the spreadsheet with the linked query in Excel. The original data from the query displays.
2. In Excel, select **Data | Refresh** to update the original data with the current data.
3. If prompted, enter a user name and password to connect to the database and refresh the data.

   or

   If the following error displays: "[Microsoft][ODBC Driver Manager] Data source name not found and no default driver specified", you must specify the DSN used by the query on your computer. Complete the following steps to resolve this issue:

   (Excel 2003)

a. Click **OK**.

b. Select the Machine Data Source tab, and then click **New**.

c. Use the wizard to select a **User Data Source** and create a DSN that points to the database.

d. Click **OK** to return to Excel and open an ODBC Driver Connect window.

e. Enter a user name and password to refresh the data.

5

Automate

Getting Started with Automation

Summary

This tutorial helps you get started with Automation by giving you an overview of the Automation module and then guiding you through the process of creating a basic script.

In this tutorial you will learn:

- How to use the Automation window
- How to create a basic script using the Select to File activity
- How to test and run a script
- How to schedule a script
- How to create a template for reusing an Automation activity

Watch a video version of this tutorial: Getting Started with Automation.

Note: The Automation module is disabled if Toad was installed with the **Prohibit saving passwords** option selected.

Introduction

Toad provides an Automation utility that allows you to easily script database activities and schedule them using the windows scheduler. In this way, database tasks can be automated to save you time. Toad Automation can increase your productivity by allowing you to automate tasks that you perform repeatedly.

The Benefits of Using Automation

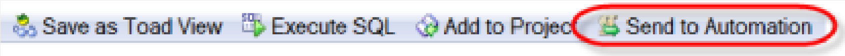
- Automating repeatable tasks allows you to be more productive and increases your efficiency.
- Toad Automation scripts are very flexible and extensible, allowing you to customized them to your needs.
- You can create and use variables in Automation scripts. Variables allow you to make your scripts flexible and to customize your scripts to different audience and company needs.
- You can incorporate a variety of templates into Automation scripts. Using templates in your scripts can help you get the most out of automating your tasks. Templates include: Microsoft Excel® files, Export Wizard templates, or Toad Report templates. You can even create an Automation activity template allowing you to reuse an activity.
- There is no limit to the number of tasks you can include in an automation script.
- Toad Automation scripts are conveniently scheduled and run by the Windows scheduler. You can track and manage your scheduled scripts using Toad or the Windows Scheduler interface.

* Did You Know *

Opening the Automation Window

There are several ways to open an Automation window.

- To build a script from scratch, use one of the following methods to open an Automation window:
  - Select **Tools | Automation**.
  - Click the **Automate** button in the main toolbar.
  - Click **Automation** in the Launch window.
- To build a script incorporating your current workflow, use the following steps:
  1. In the Query Builder or Editor, build or enter your SQL statement.
  2. Click **Send to Automation** in the Wizard bar.

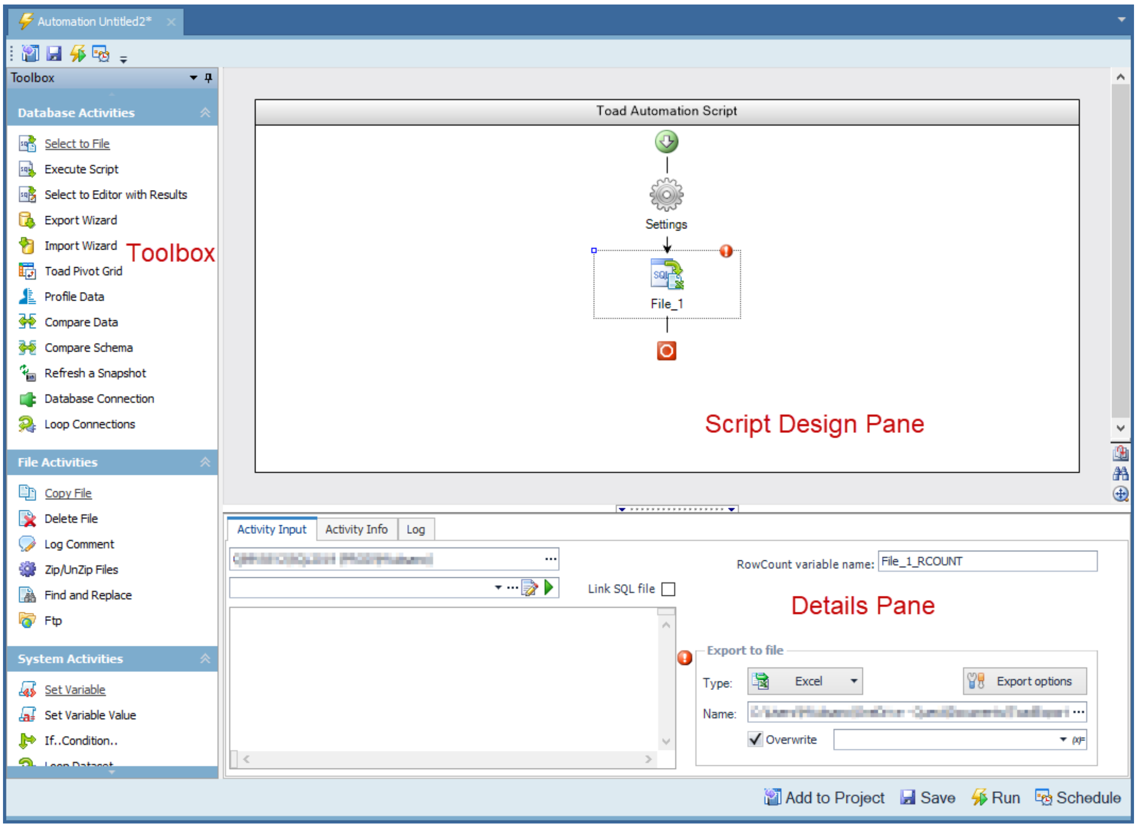
     

     Toad sends your SQL to the Automation Tutorial (Guided Tour) which opens in a separate window.
  3. Select a task and complete the wizard.
  4. After you click **Finish** in the wizard, an Automation window opens containing your new script. The script incorporates your SQL statement and the settings you specified in the wizard.

The Automation Window

1. To get started with Automation, open an Automation window by selecting **Tools | Automation**.

   The Automation window consists of a Toolbox pane (along the left side of the window), a design pane (top half of the window), and a details pane (bottom half of the window).



RowCount variable name: File_1_RCOUNT

Type: Excel

Name: [redacted]

☑ Overwrite

Toolbox—Stores the Automation activities. Click an activity in the Toolbox to add it to the script design pane. Activities are listed by category: Database, File, or System.

Script Design Pane—Graphically displays the script as a workflow. Use the script design pane to build an Automation script. Drag an activity to the design pane to add the activity to a script.

Note: You must drag the activity to an *Add Activity* icon ✚ in the Automation workflow. When you see the activity's icon displayed in the workflow, release the mouse button.

Details Pane—Displays an activity’s settings. Click an activity in the design pane to display its configurable settings in the details pane. Use the details pane to specify the settings for each activity in the script.

2. In the Toolbox, find the Database Activities group. Single-click the **Select to File** activity to add it to the script design pane.

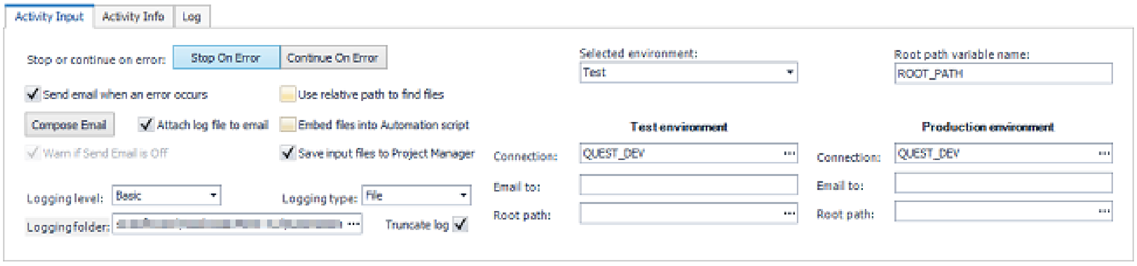
3. With the Select to File activity still selected in the script design pane, view the details pane. The details pane allows you to configure settings for the selected activity. The input fields change depending on which activity is currently selected in the design pane.

   Note: The Activity Input tab is the tab where you will specify most of an activity's settings.

4. Click ❗ next to the Select to File activity. Toad uses this icon to indicate that more information is required to complete the activity. Click or hover over the icon to display the missing information. Use this icon to help you determine which input fields are required.

   Note: This icon is also used to indicate a validation error, for example, when referencing a variable that has not been created yet.

5. Now click the **Settings** icon in the design pane. The details pane changes to display input fields for the script’s settings. You can use this area to specify script error handling, logging level, and run environment settings.



Activity Input | Activity Info | Log

Stop or continue on error: **Stop On Error** | Continue On Error

☑ Send email when an error occurs

Compose Email

☑ Attach log file to email

☑ Warn if Send Email is Off

□ Use relative path to find files

□ Embed files into Automation script

☑ Save input files to Project Manager

Logging level: Basic

Logging type: File

Logging folder: [redacted]

Truncate log ☑

Selected environment: Test

Root path variable name: ROOT_PATH

Test environment

Connection: QUEST_DEV

Email to:

Root path:

Production environment

Connection: QUEST_DEV

Email to:

Root path:

6. Click the **Stop on Error** button, if not already selected. This instructs the script to stop if an error is encountered.

   Note: You can also enable or disable the **Stop on Error** option for each individual activity. If you know that a particular activity will have an error, you can disable the **Stop on Error** option for just that activity. This option is located on the Activity Info tab for each activity.

7. In addition to selecting the **Stop on Error** option, enabling the **Send email when error occurs** and **Attach log file to email** options is also preferred. This can be very useful if your script encounters an error. Enabling your script to send an email notification *on error*, as well as *on success*, helps you to feel confident that your processes are running as expected.

   If you choose to enable these options, you must specify email settings. To specify email settings, click the **Compose Email** button.

8. Keep this Automation script window open, and proceed to the next section.

Create a Basic Script

1. If you kept the previous Automation window open, you will have a basic script consisting of one activity, the Select to File activity. Click on the Select to File activity in the design pane.

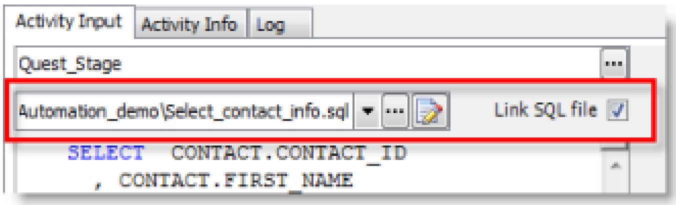
2. In the Activity Input tab, in the first text box, select a database connection, or use the default. The default database connection for each database activity is always the currently-active open connection. To select a different database connection, click ⋯ and browse to a different connection.

   Note: If you schedule a script to run against a password-protected database for which you did *not* save the password in Toad, the script will error. To save the password, right-click the connection in the Connection/Navigation Manager and select **Properties**. Enter the password and select the **Save Password** option.

   Tip: For Snowflake connections, it is considered best practice to use only the Snowflake user account authentication when creating Automation scripts to avoid unnecessary workflow interruptions.

3. Now select an input. Most activities require an input, and in the Select to File activity the input is a SQL statement. You can input a SQL statement using one of the following methods:

   - Manually enter a SQL statement in the editor box (deselect the **Link SQL file** option).
   - Click ⋯ to browse to and select a SQL file.
   - Select **Link SQL file**, then click ⋯ to browse to and select a SQL file. This option retains a link to the SQL file.
   - Click 📝 to open the SQL file (you previously selected) in the Editor where you can modify the SQL.
   - Select **Link SQL file**, then click the drop-down list to select a file from the currently open project in the Project Manager.



Quest_Stage

Automation_demo\Select_contact_info.sql

```
    SELECT   CONTACT.CONTACT_ID
      , CONTACT.FIRST_NAME
```

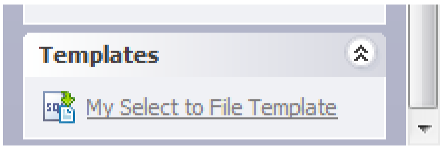
4. After selecting or entering your SQL statement, select an output. Select **Excel** in the **Type** field in the Export File section.

5. Click … in the **Name** field and select a location and enter a name for your new Excel file.

6. You can also add a suffix to the file name. In the suffix field, click the drop-down to select a datetime stamp, or click (x)= to build an expression. You can use this method to add a variable or a custom datetime stamp. Appending a datetime stamp to the file name allows you to create a unique file with each script execution.

7. You now have a complete activity, with input, output, and database connection specifications. You can save and reuse this activity in other scripts by saving the activity as a template. To do this, right-click the activity (after configuring all the settings) and select **Save As Template**. Enter a name for your template and click **OK** to save it.

   Your template now appears in the Toolbox in the Templates category.



8. Keep the Automation script window open and proceed to the next section.

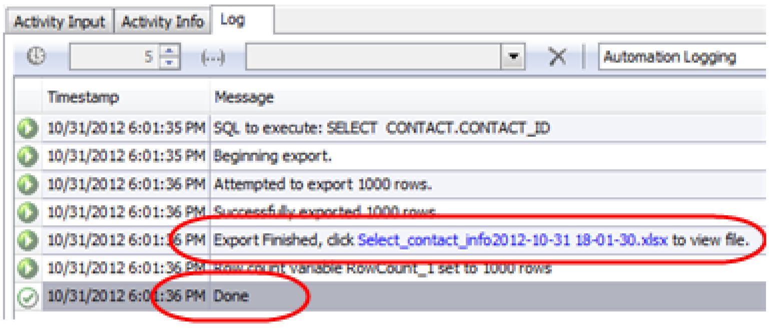
Test and Run Your Script

1. To test your script, click **Run** in the Wizard bar.

2. Toad will prompt you to save your script (if you have not already done so). In the Save As dialog, select a location, enter a file name, and click **Save**.

3. Toad then validates your entries and compiles the script. If these processes are successful, the script runs.

4. While the script is running, Toad displays your script’s execution progress in the Log tab. The message “Done” indicates script execution is finished.



```
Timestamp               Message
10/31/2012 6:01:35 PM   SQL to execute: SELECT  CONTACT.CONTACT_ID
10/31/2012 6:01:35 PM   Beginning export.
10/31/2012 6:01:36 PM   Attempted to export 1000 rows.
10/31/2012 6:01:36 PM   Successfully exported 1000 rows.
10/31/2012 6:01:36 PM   Export Finished, click Select_contact_info2012-10-31 18-01-30.xlsx to view file.
10/31/2012 6:01:36 PM   Row count variable RowCount_1 set to 1000 rows
10/31/2012 6:01:36 PM   Done
```

5. When a script produces an output file, a hyperlink to the output file is included in the Log. If your script ran successfully, click the output file hyperlink to view the new Excel file created by your script.

6. Click the **Settings** icon to return to the script’s settings.

7. If you require a more detailed Log, you can change the logging level. In the Settings Activity Input tab, select *Verbose* from the Logging level drop-down list to create a more detailed log.

   Note: If you want to change the location of the Log file, select a new location in the **Logging folder** input field.

8. Toad allows you to specify both a Test database connection and a Production database connection for each script. Toad also allows you to specify which connection to run your script against when the script executes. Both of these settings are made in the script's settings. This feature allows you to easily test your script in a test environment before running your script in the production environment.

   To specify a test and a production database, make the selections in the Connection text box under **Test environment** and **Production environment** in the Settings Activity Input tab. To select which of these databases to run your script against, select *Test* or *Production* from the **Select environment** drop-down list.

9. Click the Select to File activity to return to its settings. Then select the Activity Info tab.

10. If you remember from the overview of the Automation window, Toad allows you to enable or disable each activity in a script. This can be helpful when building a script, as it allows you to test one particular activity at a time by disabling the other activities.

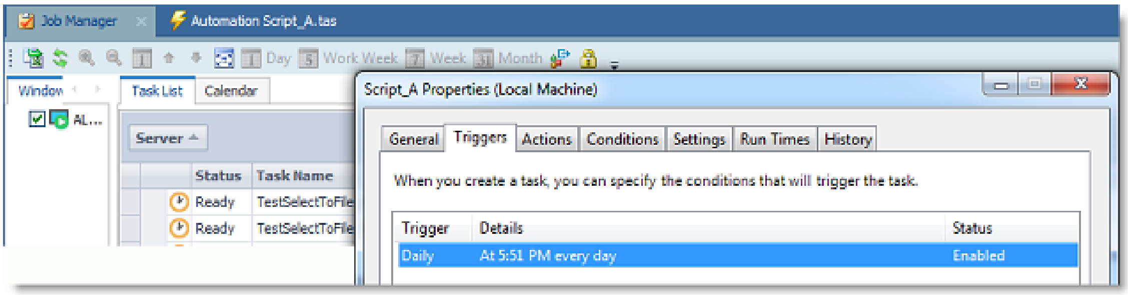
    In the Activity Info tab, click **Disabled** to disable the Select to File activity. Notice that the activity now appears shaded in the design pane. To enable the activity again, click **Enabled**, or right-click the activity in the design pane and select **Enabled**.

Schedule Your Script

Once you have thoroughly tested your script, you can schedule it.

1. To schedule your script, click **Schedule** in the Wizard bar.

2. The Job Manager window opens and a task properties dialog displays. The task properties dialog is pre-populated with the information required to run your script as a scheduled task. You only need to schedule it. (The Job Manager uses the Windows Scheduler.)

Status	Task Name
Ready	TestSelectToFile
Ready	TestSelectToFile

   When you create a task, you can specify the conditions that will trigger the task.

Trigger	Details	Status
Daily	At 5:51 PM every day	Enabled

3. To schedule the script, select the Triggers tab. Then select the scheduling trigger and click **Edit**. In the Edit Trigger dialog, ensure that the *On a schedule* trigger type is selected. Then specify a frequency, start time, interval, etc.

4. Click **OK** to save your schedule. Click **OK** to save and close the task properties dialog.

5. Your task (script) is now listed in the Task List in the Job Manager. Double-click your task in the list to open the task properties dialog again where you can make modifications, such as re-scheduling or changing the script location. Use the Task List to review the status of your tasks (scheduled scripts).

6. By default, the scheduled task runs under your Windows account. If you change your Windows password, you may also need to update the password in your scheduled task (depending on your version of Windows). Click 🔒 and select the task to update. Click **OK** and then enter your password when prompted.

7. After upgrading to a new version of Toad, you must migrate the scheduling tasks to the newer version. See "Upgrade Automation Scripts and Scheduling Tasks" in the online help for more information.

8. **Script Execution and Status.** When Toad is in the process of executing a scheduled script, you will see the Toad icon in the lower right corner of the Windows taskbar. To check the status of your scheduled scripts, open the Job Manager (**Tools | Administer | Job Manager**) and select the Task List.

Now you have learned how to increase your productivity by using Toad Automation to handle your repetitive reporting and deployment tasks!

Automate Exporting to Excel Reports

Summary

This tutorial walks you through the process of automating the task of exporting query results to simple Excel reports. You will create three different, and very useful, simple Excel reports.

In this tutorial you will learn:

- How to use Automation to export query results to Excel
- How to use the Select to File activity in Automation
- How to automate an Excel report that appends a new worksheet with each execution
- How to automate an Excel report that uses multiple queries to produce multiple worksheets

Introduction

After creating a result set in Toad, you may need to send the results to your end users. The easiest way to create a report containing query results is to export the data to an Excel file by right-clicking the data grid and selecting **Quick Export | File | Excel File**. This action creates a basic Excel spreadsheet containing the result set from the data grid.

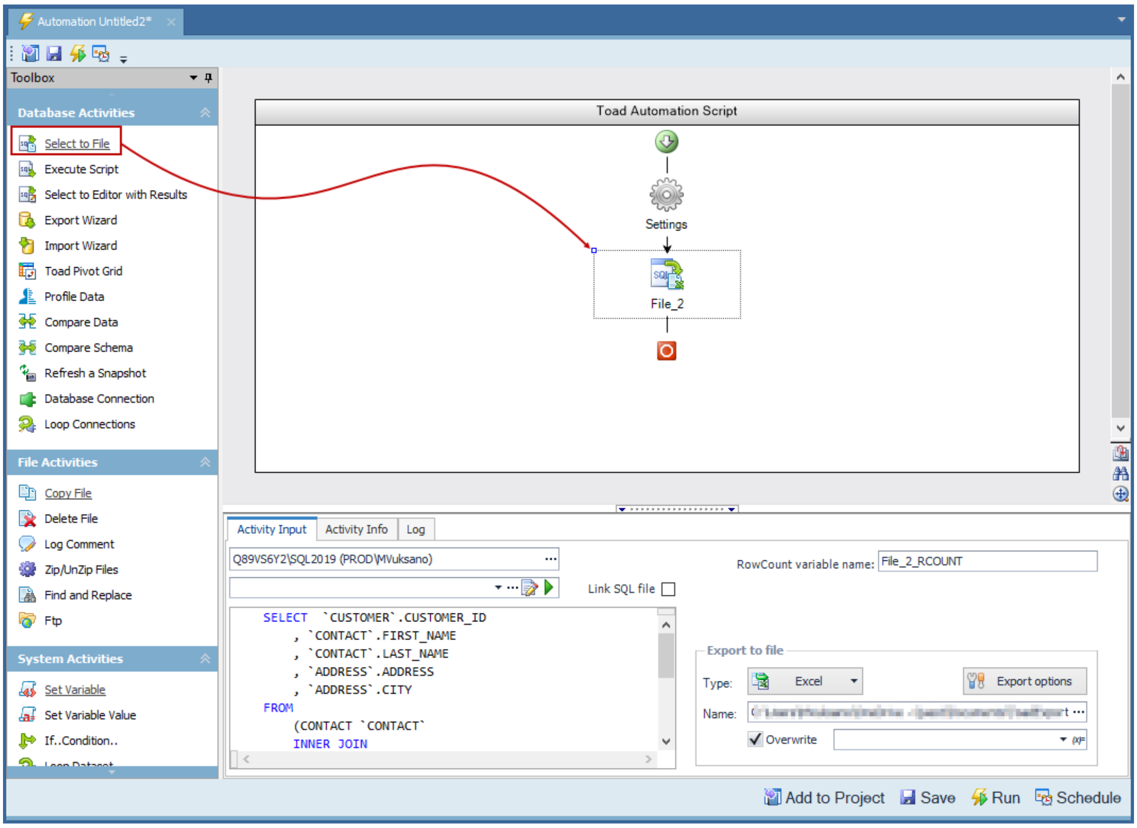
If you want to create more than a basic spreadsheet and automate the process, Toad Automation helps you build a variety of Excel reports, from simple worksheets to complex and attractive Excel reports.

In this introduction to automating Excel reports, you will learn how to automate the process of exporting data to a variety of simple Excel reports.

Note: To learn how to get started with Automation, see Getting Started with Automation.

Create a Simple Excel Report

1. Select **Tools | Automation**.
2. Click the **Select to File** activity in the Database Activities pane of the Toolbox, or drag the activity from the Toolbox to the Automation design window.



Q89VS6Y2\SQL2019 (PROD\MVuksano)

Link SQL file □

RowCount variable name: File_2_RCOUNT

```
SELECT  `CUSTOMER`.CUSTOMER_ID
      , `CONTACT`.FIRST_NAME
      , `CONTACT`.LAST_NAME
      , `ADDRESS`.ADDRESS
      , `ADDRESS`.CITY
FROM
      (CONTACT `CONTACT`
      INNER JOIN
```

Type: Excel

Name: [redacted]

☑ Overwrite

3. When the **Select to File** activity is selected in the design window, the Activity Input tab displays. Select a database connection in the first text box or use the default (the default is the currently-active open connection).

4. Then select a SQL file or enter a SQL statement using one of the following methods:

   - Manually enter a SQL statement in the editor (deselect the **Link SQL file** option).
   - Click ⋯ to browse to and select a SQL file.
   - Select **Link SQL file**, then click ⋯ to browse to and select a SQL file.
   - Click 📝 to open the SQL file (you previously selected) in an editor where you can modify the SQL.
   - Select **Link SQL file**, then click the drop-down list to select a file from the currently open project in the Project Manager.

5. Select **Excel** in the **Type** field in the Export to File section.

6. Click ⋯ in the **Name** field to select a location and enter a name for your new Excel file, or you can select an existing file.

   Note: To specify more-detailed options for the Excel file, click the **Export options** button.

7. (Optional) You can choose to create a unique file by adding a suffix. In the suffix field, click the drop-down to select a datetime stamp, or click (x)= to build an expression.

8. Select **Overwrite** if you want to overwrite the file.

   Note: If you did not add a suffix, you can deselect **Overwrite** to append a new worksheet to the Excel file with each script execution.

9. To save your Automation script, click **Save** in the Wizard bar. Then select a location and file name for your script.

10. To test your script, click **Run** in the Wizard bar.

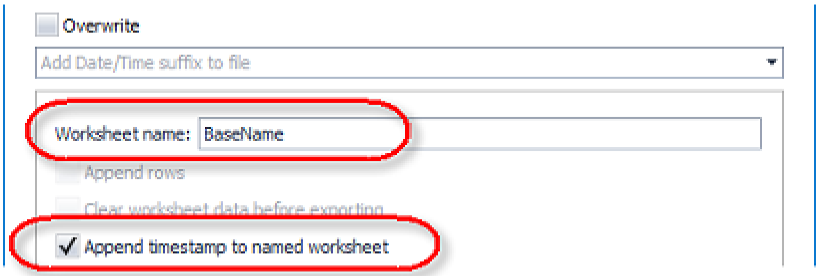
11. After the script executes, the Log tab displays. Click the hyperlink in the Log to view the Excel file.
12. To learn how to schedule your Automation script, see Schedule Your Script.

That was simple, wasn't it. Now move on to the next example to learn how to create a slightly more complex Excel report.

Create a Multi-Page Excel Report Using One Query

Suppose you want to create a new report each week from the same query while retaining the previous week's results. In this case, you want to append each week's results as a new worksheet to a single Excel file. This is an excellent option if you want to compare the results each week (or each day, month, etc.) to see how the data is changing. (And Toad has several data compare features in the application to let you do that.)

1. Select **Tools | Automation** to open a new Automation window.
2. Click the **Select to File** activity in the Toolbox to add it to the Automation design window.
3. Select a database connection.
4. Select a SQL file or enter a SQL statement using one of the methods described in the first example.
5. Select **Excel** in the **Type** field in the Export to File section.
6. In the **Name** field, click … to select a location and enter a name for your file.
7. Leave the suffix field blank (adding a date/time suffix could create a new file with each script execution and this is not the objective of this example).
8. Deselect **Overwrite** (you do not want to overwrite the file).
9. Now click **Export options**. The Export options dialog opens.
10. In the **Worksheet name** field, enter a name to be used as the base worksheet name.
11. Select **Append timestamp to named worksheet**. This will append the date and time (that the data was exported) to the base worksheet name. Click **OK** to save your settings and close the dialog.



☐ Overwrite

Add Date/Time suffix to file

Worksheet name: BaseName

☐ Append rows

☐ Clear worksheet data before exporting

☑ Append timestamp to named worksheet

12. Click **Run** to save and run your script.
13. Open the Excel file by clicking the link in the Log tab. Notice that the new worksheet's name includes the base name you specified suffixed with the timestamp.
14. Run the script a second time and open the file. A new worksheet with a new timestamp was added to the Excel file.

The output of this script is one Excel file containing multiple worksheets. Each worksheet contains a result set from one execution of the script. And each worksheet name records the date and time the result set was exported.

Create a Multi-Page Excel Report Using Multiple Queries

Another variation of the multi-page Excel report is to export the results of multiple queries. Each worksheet in the Excel file will be populated with the result set of a different query. This can be accomplished by using a SQL file that contains multiple SQL statements as the input file.

1. To start, open a new Automation window.
2. Click the **Select to File** activity to add it to the design window.
3. Select a connection.
4. Select a SQL file or enter a SQL statement using one of the methods described in the first example. However, this time use a file that contains multiple SQL statements (or enter multiple statements in the editor).
5. Select **Excel** in the **Type** field in the Export to File section.
6. Click ... in the **Name** field to select a location and enter a name for your new Excel file.
7. Click **Export options** and make sure the **Worksheet name** field is blank.

   Note: If the Worksheet name field is not blank, your script will export both result sets into the named sheet, and the second result set will overwrite the first.
8. Selecting to overwrite the existing file or to add a date/time suffix to the file name is optional in this example.
9. Click **Run** to save and run your script.
10. Open the new file by clicking the link in the Log tab. Notice that the file contains a separate worksheet for the result set of each query.

The output of this script is one Excel file containing multiple worksheets. Each worksheet contains the result set of a different query.

Schedule A Script

See Schedule Your Script to learn how to schedule the Automation script.

Automate Tasks

Use Automation to schedule a single script to run reports and queries, save results in Excel spreadsheets, and distribute results to the business community via email. You can schedule scripts that perform long-running tasks to run when database activity is light.

You can also use variables in automation scripts. With variables, you can update information that may be different each time you run the script, such as the department name, revenue, fiscal quarter, or product name. See "Using Variables in Automation" on page 82 for more information.

For example, assume you are an analyst at Nick's Flicks movie rentals. You need to run inventory and movie rental reports on a daily basis and distribute them to managers of the inventory department. Using Toad, you can create an automation script that runs the reports, exports them to Excel spreadsheets, and emails the spreadsheets to the inventory department everyday at 2:00 AM.

Troubleshooting: You must surround a variable name with the # sign (for example, #sleep#).

Notes:

- Automation scripts are compatible with the version of Toad in which they are created.
- The Automation module is disabled if Toad was installed with the **Prohibit saving passwords** option selected.

To automate tasks

1. Select **Tools | Automation**.
2. Click **Settings** (icon) in the script design window and specify script settings. Enter settings in the bottom pane. Review the following for additional information:

Activity Input Tab	
Stop or continue on Error	Select **Stop On Error** to stop the script if an error occurs.
Send email when error occurs	Select this option to send an email on script error. If selected, you must also click **Compose Email** to compose the email message and to edit the **Email Server Settings**. **Note:** When you specify the error email settings, Toad saves your settings and uses them in all subsequent automation scripts.
Use relative path to find files	Select to convert all local file paths to relative paths in all activities in the script (network file paths remain unchanged). This is useful if you want to share your automation scripts with colleagues or place them on a shared network. Toad makes the change the next time you run or save the script. If you deselect this option later, the script reverts back to the original absolute paths.
Embed files into Automation script	Embeds Toad input files into the script. This makes it easier to share the script with colleagues. **Important:** To use this option, *deselect* it, build your script, select input files, then *re-select* this option. This allows Toad to collect and save the input file paths. This option embeds Data Compare, Pivot Grid, Toad Data Report, and Import/Export template files. Scripts used by the Run Automation Script activity are also embedded, as are SQL query files used as input by an Import/Export template. **Important:** Files linked using the **Link to File** option in the Select to File, Execute Script, and Select to Editor with Results activities are **not** embedded [.sql, Editor (.tef), and Query Builder (.tsm) files)]. Toad embeds the files the next time you run or save the script. If necessary, at a later time you can deselect this option and then re-establish links to the original files.
Truncate log	Select to overwrite the log file each time the script runs.
Selected environment	Use this area to specify a "test" database connection and a "production" database connection and then easily switch between the two. • **Test**—Select to run your script in test mode using the database connection and other settings you specified under **Test environment**. • **Production**—Select to run your script in production mode using the database connection and other settings you specified under **Production environment**. **Note:** The **Connection** field under both environments defaults to the currently-active open connection. **Variable names** • **Root path**—Enter a variable name or use the default name. Then use this variable name in your script wherever you want to apply the path defined under **Root path** in either the Test settings or the Production settings. The path that is applied depends on which environment (*Test* or *Production*) is selected at the time the script is executed. • **Run mode**—Enter a variable name or use the default name. The value assigned to this variable is either *Test* or *Production*, based on the environment selected at the time the script is executed.
Activity Info Tab	
Name	Enter a script name.

3. To build a script, single-click an activity in the Toolbox, or drag an activity from the Toolbox to the Automation script design window. Repeat this process to add additional activities to your script.

   Note: You must drag the activity to ✚, then release the mouse button after you see the activity's icon replace ✚.

4. Click an activity in your script workflow in the design window and then specify the activity's properties in the details pane (bottom pane). Repeat this process for each activity in your script. For a step by step tutorial on building an Automation script, see Getting Started with Automation.

   See the following topics for details about each activity and its properties:

   - Use Database Automation Activities
   - Use File Automation Activities
   - Use System Automation Activities

   Tip: You can save an activity and its settings as a template to reuse in other scripts. Right-click the activity in the design window and select **Save As Template**. After you create the first template, a Templates toolbox displays.

   Note: Click **Settings** in the design window at any time to go back and review the Automation script settings.

5. When you finish building your script, select one of the following actions from the Automation toolbar or the Wizard bar:

Add to Project	Add an automation script to the current project in the Project Manager. The script is automatically placed in the project's Automation Scripts folder.
Save	Save an automation script (.tas) in a folder on a local or network drive.
Run	Run an automation script before scheduling it. The script executes once and the execution log is displayed in the Log window. Resolve any problems that cause activities to fail, or disable failed activities before scheduling the script. **Tips:** • Press F5 to run the automation script. • To view the script log, select the Log tab or select **View | Output**.
Schedule	Schedule an automation script in the Job Manager. You can schedule a script to execute once or to run periodically at a defined interval. See "Execute and Schedule Automation Scripts" on page 86 for more information.

Important: To rename an Automation script file, do so through Toad (**File | Save File As**), not through Windows Explorer. Renaming a script file using Toad ensures that log files are created correctly and properly synced with their scripts.

Tips:

- Double-click an activity in the script design window to open the Activity Properties pane in a format used in earlier versions of Toad.
- Right-click within the script design window and select **Print Script** to print the script's workflow diagram.

Use Database Automation Activities

You can use the following activities in automation scripts.

Note: When building an automation script, you must have the appropriate connections open for database activities.

Tip: You can save an activity and its settings as a template to reuse in other scripts. Right-click the activity in the design window and select **Save As Template**. After you create the first template, a Templates toolbox displays.

Activity	Description
Select to File	Execute an existing SQL script, Query Builder file, or Editor file and export the results to an Excel, HTML, or CSV file. 1. Select a database connection. 2. Select a SQL file or enter a SQL statement. - Click the drop-down list to select a file from the currently open project in the Project Manager. - Click … to browse to and select a SQL file. - Click ✎ to open the SQL file in the Editor where you can modify the SQL file. - Enter a SQL statement in the editor. You must first deselect the **Link SQL file** option. 3. RowCount variable—Enter a name for the row count variable or use the default. The value assigned to this variable equals the number of rows returned from the activity. See "Using Variables in Automation" on page 82 for more information. 4. Export File Type—Select an export file type. 5. Export File Name—Click … to select a file location and enter a file name. To use a variable as a suffix, enter the hash symbol before and after the variable name (#VariableName#). 6. Suffix—Select a suffix to append to the file name. Select from the list of existing suffixes, or click (x)= to select a variable or build a new suffix using the expression editor. 7. Overwrite—Select to overwrite the file. If you did not add a suffix to the file name, you can deselect Overwrite to append a new worksheet to the Excel file with each script execution. 8. Export Options—Click to specify more export options. - Worksheet name—Enter a worksheet name. For an existing file, select the worksheet from the list. **Overwrite** must be *deselected* to display the list. To add a worksheet to an existing file, select *Append worksheet* from the list and enter a new name. - Append rows—Select to append to the selected worksheet. - Start export at—Enter the column and row position at which to begin the export. **Tip:** After exporting your file, you can use a subsequent activity in your script to email the report, to zip the report in a zip folder, or to copy the report to another location.
Execute Script	Execute an existing SQL script, Query Builder file, or Editor file. You can save the results of the script in a variable or export the results to Local Storage. 1. Select a database connection. 2. Select a SQL file or enter a SQL statement. - Click the drop-down list to select a file from the currently open project in the Project Manager. - Click … to browse to and select a SQL file.

Activity	Description
	- Click ✎ to open the SQL file in the Editor where you can modify the SQL file. - Enter a SQL statement in the editor. You must first deselect the **Link SQL file** option. 3. Save result set in variable—Enter a name for the result variable, which contains a data table with the results of the execution script. **Tip:** You can use the result variable with the Loop Dataset activity. See "Use System Automation Activities" on page 78 for more information. 4. Local Storage—To export the results to Local Storage, select a database and a new or existing table (or snapshot). Review the following for additional information: - RowCount variable—Enter a name for the row count variable or use the default. The value assigned to this variable equals the number of rows returned from the activity. See "Using Variables in Automation" on page 82 for more information. - Overwrite existing dataset—Select to overwrite an existing dataset with the same name.
Select to Editor with Results	Execute an existing SQL script or Query Builder query and save it and any results sets in a Toad editor file (.tef). This file format is useful for building scripts that have large result sets that you do not want to continue executing or for saving results when you have not finished building a script. 1. Select a database connection. 2. Select a SQL file or enter a SQL statement. - Click the drop-down list to select a file from the currently open project in the Project Manager. - Click … to browse to and select a SQL file. - Click ✎ to open the SQL file in the Editor where you can modify the SQL file. - Enter a SQL statement in the editor. You must first deselect the **Link SQL file** option. 3. Result saved in Editor file—You can select an Editor file from an open project or browse to select a file. Click ✎ to edit the selected file. **Tip:** You can use a variable in this field by entering the hash symbol before and after the variable name (#VariableName#). 4. Append results—Select to include the result set in the Editor file. Review the following for additional information: - RowCount variable—Enter a name for the row count variable or use the default. The value assigned to this variable equals the number of rows returned from the activity. See "Using Variables in Automation" on page 82 for more information. - Suffix—Select a suffix to append to the file name. Select from the list of existing suffixes, or click (x)= to select a variable or build a new suffix using the expression editor.
Export Wizard	Export data from databases, tables, views, or a query into a CSV, Excel, or other file.

Activity	Description
	Review the following for additional information: - Export template—You can select an existing template, edit an existing template, or create a new template. **Tip:** You can use a variable in the file name to select an *existing* template file by entering the hash symbol before and after the variable name (#VariableName#). - Use Automation Connection—Select this option to use the connection associated with the Export Wizard activity. Clear the checkbox to use the connection associated with the export template. - RowCount variable—Enter a name for the row count variable or use the default. The value assigned to this variable equals the number of rows returned from the activity. See "Using Variables in Automation" on page 82 for more information.
Import Wizard	Import data from a CSV, Excel, or other file into a new table or append to an existing table. You can also import data from one database to another. Review the following for additional information: - Import template—You can select an existing template, edit an existing template, or create a new template. **Tip:** You can use a variable in the file name to select an *existing* template file by entering the hash symbol before and after the variable name (#VariableName#). - Use Automation Connection—Select this option to use the connection associated with the Import Wizard activity. Clear the checkbox to use the connection associated with the import template. - RowCount variable—Enter a name for the row count variable or use the default. The value assigned to this variable equals the number of rows returned from the activity. See "Using Variables in Automation" on page 82 for more information.
Toad Pivot Grid	Open an existing Toad Pivot Grid file, execute the query, and export the data to Excel or other file formator Local Storage. **Note:** You must create a Toad Pivot Grid file (.tpg) before you can select it using this activity. Review the following for additional information: 1. Select a connection to execute the Toad Pivot Grid file against. 2. Select an existing Toad Pivot Grid file. You can browse for a file or select from the currently-open project. Click to edit the selected pivot file. 3. Export to file—Select a file type and enter a file name and location for your export file. You can use a variable in this field by entering the hash symbol before and after the variable name (#VariableName#). - To export to an Excel pivot table, select **Excel Pivot**. - Click **Export Options** to specify Excel file options. 4. Suffix—Select a suffix to append to the file name. Select from the list of existing suffixes, or click (x)= to select a variable or build a new suffix using the expression editor. 5. RowCount variable—Enter a name for the row count variable or use the default. The value assigned to this variable equals the number of rows returned from the activity. See "Using Variables in Automation" on page 82 for more information. 6. To export pivoted data to Local Storage, select a database and a new or existing table.

Activity	Description
Profile Data	Automate data profiling tasks. This activity profiles data from a selected query and exports a report containing the profiling results. 1. Select a database connection. 2. Select a SQL file or enter a SQL statement. - Click the drop-down list to select a file from the currently open project in the Project Manager. - Click … to browse to and select a SQL file. - Click ✎ to edit the selected file. - Enter a SQL statement in the editor. 3. To specify options for the profiling task, click **Profiling Options**. 4. Create Report—(Required) Select a location, select a file format, and enter a file name for the exported report (PDF, HTML, etc). 5. Suffix—Select a suffix to append to the file name. Select from the list of existing suffixes, or click (x)= to select a variable or build a new suffix using the expression editor. 6. Export Data Profiling File—(Optional) Select a location and enter a file name for the Toad Profiling Report file (.tpr). **Note:** You cannot attach this file in subsequent activities in the script. See Data Profiling for more information.
Visualize Data	Open an existing Toad Data Visualization file (.tdv), refresh the chart, and export contents as a report. Select from a number of output file formats, including PDF and Excel. - Data Visualization document—Select an existing Toad Data Visualization file. - Output file—Select a file location, file name, and file type for your output file. You can use a variable in this field by entering the hash symbol before and after the variable name (#VariableName#). - Suffix—Select a suffix to append to the file name. Select from the list of existing suffixes, or click (x)= to select a variable or build a new suffix using the expression editor.
Compare Data	Open a Toad Data Compare project (.dcp) and export the contents as a summary report and detailed object reports. Select from a number of output file formats for the summary report, including PDF and Excel. - Data Compare Project—Select an existing Toad Data Compare project file. - Summary Report—Select a file location, file name, and file type for the comparison summary report file. - Object reports folder—Select a destination folder for the detailed object comparison reports generated by the activity. - Suffix—Select a suffix to append to the file name. Select from the list of existing suffixes, or click (x)= to select a variable or build a new suffix using the expression editor. - Generate sync script—Select a location and enter a file name to generate a synchronization script. Click to configure options for the synchronization

<table>
<tr><th>Activity</th><th>Description</th></tr>
<tr><td></td><td>script.<ul><li>Execute script—Select to execute the generated script.</li><li>Compare result variable—Enter a variable name or use the default. This variable is assigned a value of <i>True</i> or <i>False</i> based on the results of the data comparison.</li></ul></td></tr>
<tr><td>Refresh a Snapshot</td><td>Use this activity to refresh a snapshot in Local Storage. This action will update the snapshot to reflect the current data in the source table.
Select one of the following:<ul><li>Local Storage<ul><li>Database—Select the database containing the snapshot to refresh.</li><li>Snapshot—Select the snapshot to refresh.</li></ul></li></ul></td></tr>
<tr><td>Database Connection</td><td>You can specify the database connection details within each applicable activity, eliminating the need to use the Database activity. You may continue to use the Database activity if you prefer.
<b>Note:</b> The database connection specified for the Database Connection activity takes precedence over the connection specified for the individual activity.</td></tr>
<tr><td>Loop Connections</td><td>Loops through multiple connections and executes the activities you specify. Add as many activities as you want to the loop. The Loop Connections activity can have more than one branch.<ul><li>Select connections to loop through—Click <b>Add Connections</b> to select one or more connections to loop through.</li><li>Connection description variable—Enter a new name for the variable or use the default name. The connection description variable is a SQL-type variable comprised of one row containing the connection information for the current connection.
Use the following column names to gather connection information from the connection description variable. The exact content of each column depends on the provider type.
<table>
<tr><th>Column Name</th><th>Description</th></tr>
<tr><td>DescriptionLong</td><td>Description of the connection details.</td></tr>
<tr><td>ProviderType</td><td>Provider type such as SQLServer, Oracle, or SAP ASE.</td></tr>
<tr><td>HostName</td><td>Hostname or instance for the connection.</td></tr>
<tr><td>UserName</td><td>User name.</td></tr>
<tr><td>TrueUserName</td><td>If the connection supports and uses Windows authentication (e.g., SQL Server connection) this column contains the current Windows Logon name as: domain\username (e.g., "PROD\bgates"). Otherwise, if database authentication is used, it contains the same as the UserName column.</td></tr>
<tr><td>Database</td><td>Default Database context for the connection.</td></tr>
</table></li><li>Next, add activities to the loop to execute for each connection in the connection list.</li></ul><b>Note:</b> The connections you specify in the Loop Connections activity take precedence over any connection specified in an individual activity.</td></tr>
<tr><td colspan="2"><b>Notes:</b></td></tr>
</table>

Activity	Description
	• To disable an activity, right-click the activity in the design window and select **Disabled**. This is helpful if you want to disable an activity while testing other activities. • The **Stop on Error** option for each activity allows you to select whether to stop or continue execution if Toad encounters an error during that activity. • To prevent an error in one activity from causing a failure at the script level, clear the **Generate Error** check box (Activity Info tab) for that activity. • Select the Log tab to review a script's execution log. • In the **Suffix** field, the drop-down list includes two predefined suffixes (Date and DateTime). The list also includes any suffixes you have created and saved since installing Toad. To delete a suffix from the list, select the suffix, click (x)=, delete the suffix in the **Expression** field, and click **OK**. • Automation does not support positional parameters denoted by a question mark. • For each applicable activity, you can specify the database connection in the activity's properties. The default connection is the currently-active open connection. • The connection associated with an activity takes precedence over the connection information in a report or SQL script, unless the report or SQL script information is a cross-connection query.

Use File Automation Activities

You can use the following activities in automation scripts.

Tip: You can save an activity and its settings as a template to reuse in other scripts. Right-click the activity in the design window and select **Save As Template**. After you create the first template, a Templates toolbox displays.

Activity	Description
Copy File	Copy or move one or more files to another location (for example, to a shared folder where they can be accessed by your business community). Review the following for additional information: • Source file—Select the file to copy. In this field, you can also use the asterisk (*) and question mark (?) wildcard characters, as well as variables, in the file name to help you select one or more files (in the same directory) to copy. To use this method: 1. Specify the file directory by browsing to and selecting one of the files. 2. Then modify the file name in the **Source file** field using wildcard characters and/or variables. • Copy to folder—Select the destination folder into which you want to copy or move the file or files. • Copy name—To rename the copied or moved file, enter a new name. **Note:** This field is only applicable when copying a single source file. • Suffix—Select a suffix to append to the file name. Select from the list of existing suffixes, or click (x)= to select a variable or build a new suffix using the expression editor. • Delete source file—Select to *move* the file from the source folder to the destination folder. Deselect to *copy* the file to the destination folder. • Retry—Specify the number of times to retry the copy action if an error or timeout occurs.

Activity	Description
	• Wait—Specify the number of seconds to wait between retries.
Delete File	Delete one or more files from one or more specified locations. You can also use this activity to delete files that are created previously in the script. Files to delete—Click **Add Files** to select one or more files to delete. In the File Collection Editor, do one or both of the following: • To delete files created previously in the script, select a file in the left pane and click **Add**. **Note:** When a file name is displayed for selection in the left pane, any variable names that are used in the file name are visible (#MyVariable#), but suffixes are not visible. • To select other existing files, click … to browse to and select the files. In this field, you can also use the asterisk (*) and question mark (?) wildcard characters, as well as variables, in the file name to help you select one or more files (in the same directory) to delete. To use this method: 1. Specify the file directory by browsing to and selecting one of the files. 2. Then modify the file name in the Files to Delete pane using wildcard characters and/or variables. **Note:** As an example of using wildcards and variables, using the file name MonthlyReport_#var#*.xls, where #var# is Feb, would delete all reports whose file name *begins* with "MonthlyReport_Feb" in the same directory.
Log Comment	Writes a comment to the scripts's log. Log message—Enter the text of the comment you want to insert into the log when this activity executes.
Zip/Unzip Files	Create a zip file (compressed folder) and add files to it. You can add files created previously in the script to the zip file. Use this activity to zip multiple files and then attach the zip file to an email using the Email activity. You can also use this activity to unzip a zip file. Review the following for additional information: **Zip Tab** • Archive name—Select a location and name for the zip file. You can use a variable in this field by entering the hash symbol before and after the variable name (#VariableName#). • Suffix—Select a suffix to append to the file name. Select from the list of existing suffixes, or click (x)= to select a variable or build a new suffix using the expression editor. • Password—Enter a password if you want to password protect the zip file. • Hide password—Select this option to mask password during input. • Encryption—Select whether to create a zip file without encryption (Standard) or to encrypt the zip file. **Note:** If you encrypt the zip file, you must have a zip utility that can support 128-bit or 256-bit decryption. • Zip all files in this directory—Select a directory. All files in the directory will be added to the zip file. You can use both this field and the **Source files** field to select files to add to the zip file. • Source files—Click **Add Files** to select one or more files to add to the zip file. In the File Collection Editor, do one or both of the following:

Activity	Description
	• To add files created previously in the script, select a file in the left pane and click **Add**. **Note:** When a file name is displayed for selection in the left pane, any variable names that are used in the file name are visible (#MyVariable#), but suffixes are not visible. • To select other existing files, click ... to browse to and select the files. In this field, you can also use the asterisk (*) and question mark (?) wildcard characters, as well as variables, in the file name to help you select one or more files (in the same directory) to add. To use this method: 1. Specify the file directory by browsing to and selecting one of the files. 2. Then modify the file name in the Files to Archive pane using wildcard characters and/or variables. **Unzip Tab** • Archive name—Select a zip file. • Password—If password-protected, enter the password. • Extract to folder—Select a location to extract the files to.
Find and Replace	This activity opens one or more files and performs find and replace actions inside the files. You can write the results to a different location and file name. Review the following for additional information: • Find/Replace—You can have multiple Find/Replace tasks within one activity. You can specify a different source file, save file, and action for each Find/Replace task. • Source file—Select the file in which to perform the find and replace action for the selected Find/Replace task. In this field, you can also use the asterisk (*) and question mark (?) wildcard characters in the file name to help you select one or more files (in the same directory). To use this method: 1. Specify the file directory by browsing to and selecting one of the files. 2. Then modify the file name in the **Source file** field using the wildcard characters. • Find—Enter the content to find. • Replace with—Select the type of replacement content: a value, contents from a file, or a variable value. • Set value—Enter the replacement value, select the replacement variable, or select a file containing replacement content. • Save file—Select a location and file name. *The source file is selected by default.* • Suffix—Select a suffix to append to the file name. Select from the list of existing suffixes, or click (x)= to select a variable or build a new suffix using the expression editor. **Note:** Use drag and drop or up/down arrows to change the order of multiple Find/Replace tasks in one activity.
FTP File	Upload or download one or more files using an FTP or SFTP connection. Review the following for additional information:

Activity	Description
	• Select an FTP connection—Select an FTP connection or define a new connection. **Note:** You can define a new connection prior to using the FTP activity by selecting **View | FTP Connections**. • Transfer ONLY if—Select this option to set conditions for FTP transfer, and then select a condition from the list. If you select this option, the FTP activity only runs if the condition you specify is met. • Local directory exists—Select this option to transfer only if a local directory exists, then browse to and select the directory. • Local file exists—Select this option to transfer only if a local file exists, then enter the name of the local file. • Remote directory exists—Select this option to transfer only if a remote directory exists, then browse to and select the directory. • Remote file exists—Select this option to transfer only if a remote file exists, then enter the name of the remote file. • Define Operations—Click to specify the files and other options for the FTP upload or download task. Review the following for additional information: ◦ Operation—Select *Upload* or *Download*. ◦ Use file mask—Select this option if you want to select files by using a file mask. Then specify file mask filtering options in the File Mask fields at the bottom of the page. ◦ Move files—Select this option to *move* the files from the source location to the target location instead of copying them. • Retry—Specify the number of times to retry the FTP connection if it fails. • Wait—Specify the number of seconds to wait between retries.

Notes:

- To disable an activity, right-click the activity in the design window and select **Disabled**. This is helpful if you want to disable an activity while testing other activities.
- The **Stop on Error** option for each activity allows you to select whether to stop or continue execution if Toad encounters an error during that activity.
- To prevent an error in one activity from causing a failure at the script level, clear the **Generate Error** check box (Activity Info tab) for that activity.
- Select the Log tab to review a script's execution log.
- In the **Suffix** field, the drop-down list includes two predefined suffixes (Date and DateTime). The list also includes any suffixes you have created and saved since installing Toad. To delete a suffix from the list, select the suffix, click (x)=, delete the suffix in the **Expression** field, and click **OK**.
- Automation does not support positional parameters denoted by a question mark.

Tips:

- For activities in which you can add a suffix to the output file, click (x)= to build a suffix using the expression editor.
- You can use the expression editor to select a variable created in a previous activity or select one of your stored bind variables.

Use System Automation Activities

You can use the following activities in automation scripts.

Tip: You can save an activity and its settings as a template to reuse in other scripts. Right-click the activity in the design window and select **Save As Template**. After you create the first template, a Templates toolbox displays.

Activity	Description
Set Variable	Automation variables are a powerful tool that can greatly extend your scripts by adding flexibility and customization. You can use variables to represent and store data that may be different each time the script executes. Variables can be used with almost every Automation activity. Use this activity to define one or more variables. Review the following for additional information: • Variable name—Enter a name for the variable. When you use the variable name later in the script, for example in a file name or in an expression, surround the variable name with the hash symbol (#MyVariable#). • Variable type—Specify a variable type. • SQL—Select *SQL* if you plan to use the result set of a query as the variable value. • DateTime—See Using Variables in Automation for more information about date variables. • Variable value—Enter a value or click (x)= to build an expression. • If you click (x)= to build an expression, select from the list of variables, operators, and functions. Click a category in the left pane to display a list of items in the right pane. Double-click an item to add it. Click **Test** to test the expression. • If you selected *SQL* as the variable type, the variable value options expand to allow you to specify a query. Select a database connection, then select a SQL file or enter a SQL statement. Toad uses the result set of the query as the value for your variable. See "Using Variables in Automation" on page 82 for more information about SQL type variables. • Prompt during run—Select this option to instruct Toad to prompt for a new variable value during manual execution. The new value is used for the current execution only. The *default value* specified in **Variable value** remains unchanged. The *default value* is used for a scheduled script. • Add—Click to specify an additional variable/value pair. Use Up and Down buttons to reorder list. To learn more about variables, see the following: • Using Variables in Automation
Set Variable Value	Use this activity to assign a new value to a variable previously defined in the current script or current activity. You can assign a new value to any variable type, except SQL. Review the following for additional information: • Variable name—Select a previously-defined variable. • Variable type—Displays the variable type of the selected variable.

Activity	Description
	• Variable value—Enter a new value or click (x)= to build an expression. • Add—Click to select an additional previously-defined variable. See Variable Scope in *Using Variables in Automation* to learn more about how to use this activity with the Set Variable activity.
If..Condition..	Execute one or more activities if a condition you specify evaluates to true. You can add several branches to the If Condition activity to simultaneously evaluate more than one condition. Use the expression editor to build an expression that evaluates a condition. Typically this includes a variable (user-defined or built-in). • Expression—Build your expression by selecting from the list of variables, operators, and functions. Click a category in the left pane to display a list of items in the right pane. Double-click an item to add it. Click **Test** to test the expression. This activity can have one or more branches. To delete a branch, select it and press **Delete**. To add a branch, right-click the activity (the outside loop) and select **Add Branch**. If you have multiple conditions, they do not have to be mutually exclusive. Multiple branches can be true. To learn more about variables, see the following: • Using Variables in Automation
Loop Dataset	Loops through each row in a dataset and performs the activities you specify. The dataset is the result set of a query you provide (the driving query) and is stored in the dataset variable. Add as many activities as you want to the loop. The Loop Dataset activity can have more than one branch. **Tip:** You also have the option to select any SQL-type variable created in a previous activity to use as the dataset variable. Activities that create SQL-type variables include Execute Script, Set Variable, or Loop Dataset. Review the following for additional information: • Dataset variable—Enter a new name for the dataset variable or use the default name. • Select dataset variable—(Optional) You can select one of the SQL-type variables created in a previous activity to use as the dataset variable. • Connection—Select a connection. • SQL script file—Select a SQL file to use as the driving query. After the script executes, the result set of the driving query is stored in the dataset variable. • Click the drop-down list to select a file from the currently open project in the Project Manager. • Click ... to browse to and select a SQL file. • Click ✎ to open the SQL file (you just selected) in the Editor where you can modify the SQL. **Note:** Click ▶ to preview the query results. • Link SQL file—Select to link the SQL script file you selected in the previous step. • SQL script—Enter a SQL statement to use as the driving query (if you did not select a SQL file). • RowCount variable—Enter a name for the row count variable or use the default. The value assigned to this variable equals the number of rows returned from the

Activity	Description
	activity. See "Using Variables in Automation" on page 82 for more information. **Tip:** In the Loop Dataset, you can use the following format to filter data from the dataset variable when referencing it: <*dataset variable name*>.<*filter parameter*>. Example: Loop_data_1_SQL.region_id, where region_id is a column name
While..	The While activity runs a loop while a condition you specify is true. The activity ends when the condition becomes false, or when the activity completes the selected number of loops. Each loop executes the activities you specify. Use the expression editor to build an expression that evaluates a condition. Typically this includes a variable (user-defined or built-in). • Expression—Build your expression by selecting from the list of variables, operators, and functions. Click a category in the left pane to display a list of items in the right pane. Double-click an item to add it. Click **Test** to test the expression. • Loop—Enter the number of times to loop the activity. If the condition never becomes false, Toad ends the activity after this number of loops. The default is *10*. This activity can have one or more branches. To delete a branch, select it and press **Delete**. To add a branch, right-click the activity (the outside loop) and select **Add Branch**. If you have multiple conditions, they do not have to be mutually exclusive. Multiple branches can be true. To learn more about variables, see the following: • Using Variables in Automation
Send Email	Use this activity to distribute reports produced by previous activities in the script or to send notification of script execution status. **Note:** To use this activity, an SMTP service must be running. Review the following for additional information: • From—Enter your email address or click ... to auto-populate this field with your email address. **Note:** If you encounter a security alert message from your email application, and you want to use this auto-populate feature, grant Toad access (at least temporarily). • To—Enter an email address or click ... to auto-populate this field with your email address. • Prompt—Select to convert the **To** field to a variable and instruct Toad to prompt for a new variable value during manual execution. The new value is used for the current execution only. The *default value* specified in **To** remains unchanged. The *default value* is used for a scheduled script. • Email Server Settings—Click this button to specify your SMTP email server settings. • Server—Enter the address of your SMTP email server or click ... to auto-populate this field. If Toad is unable to auto-populate this field, contact your email administrator for the SMTP email server address. **Note:** The email server setting automatically defaults to that specified in the script's Settings (click the Settings icon and then click the **Compose Email** button). • User name / Password—If a non-SSL connection requires authentication, enter credentials. If using a secure connection, enter credentials, select **Use Secure connection** and select SSL or TLS. • Subject, Body—Enter a subject line and enter text for the body of your email.

Activity	Description
	• Add Attachments—Click to add files created by this script or to browse for other files to add as attachments. In the **File Collection Editor** do one or both of the following: ◦ Select one or more previously created files in the left pane and click **Add**. ◦ Click ... and browse to select an existing file. In this field, you can also use the asterisk (*) and question mark (?) wildcard characters, as well as variables, in the file name to help you select one or more files (in the same directory) to add. To use this method: 1. Specify the file directory by browsing to and selecting one of the files. 2. Then, in the File Collection Editor, modify the selected file name (in the right pane) using wildcard characters and/or variables. • Use file for body—Select a file to use as the body of the email. This is useful if you want to include report contents in the body of the email instead of as an attached file. For example, in the Select to File activity you can select HTML as the output file format, and then use this HTML file as the body of the email. In another scenario, you can use a form letter as the body of your email. Create a form letter template and then customize the file with the Find and Replace activity. **Note:** In the **To**, **CC**, **Subject**, and **Body** fields: You can use a variable in this field by entering the hash symbol before and after the variable name (#VariableName#).
Run Program	Run a program with optional command-line arguments. Review the following for additional information: • Program—Select a batch file (.bat) or executable (.exe) file. • Arguments—Enter command-line arguments. You can also enter a variable, such as #sleep# in this field. **Note:** If you want to run a macro in Access, enter *databasename* /x *macroname* in this field. For example, if you have a macro that beeps and runs in your Northwind database, you would enter *Northwind 2007.accdb /x beep*. • Run directory—Select the directory to change to, when running the program. • Run style—Select the window state for the program when it runs. • Verb—Enter verbs used when running the program, such as "Print". • Wait for program to exit—Select this option to pause if there is an error. Select duration of wait time in minutes. • Return code variable—Enter a variable name or use the default name. You can use this variable with the If Condition activity. Return code (exit code) values of 16384 (4000 hex) or larger do not generate a script error in this activity. Other return code values do. This allows you to use your program to generate a custom return code value in this range and use it in subsequent activities without generating a script error. **Note:** If the script opens a command-line window, you must manually close the window once the script completes.
Run Automation Script	Run another automation script in your current automation script. Review the following for additional information: • Automation script—Select an automation script (.tas file). ◦ After selecting a script, click to open the child script in a new window.

Activity	Description
	• Return code variable—Enter a variable name or use the default name. The return code variable is a built-in variable. After the child script executes, the return code value is stored in this variable. Return code values are: ○ 0 = Successful execution ○ 999 = An error occurred during execution ○ 111 = An error occurred, but you selected continue on error. You can use this variable with the If Condition activity.
Pause	Stops an activity for a specified number of seconds before continuing. This is useful, for example, if you are using parallel activities and need to pause one branch while waiting for results or the complete execution of another branch.
Group Activities	Use this activity to group and organize two or more activities together.
Throw Error	Use this activity to generate an error and add a custom error message to the script execution log. Select **Stop On Error** to stop the script after the error is logged. This option is independent of the global Stop/Continue on Error option specified in script Settings or the Stop on Error option in the Activity Info tab. This activity generates a script exit code of 555.
Parallel	Run two or more activities in parallel. After adding the activity to the design window, hover over the activity to display a drop-down list of view options. View Parallel—Normal view. View Cancel Handler—Select to add cancel handler activity. View Fault Handlers—Select to add fault handler activity. This activity can have one or more branches. To delete a branch, select it and press **Delete**. To add a branch, right-click the activity (the outside loop) and select **Add Branch**.

Notes:

- To disable an activity, right-click the activity in the design window and select **Disabled**. This is helpful if you want to disable an activity while testing other activities.
- The **Stop on Error** option for each activity allows you to select whether to stop or continue execution if Toad encounters an error during that activity.
- To prevent an error in one activity from causing a failure at the script level, clear the **Generate Error** check box (Activity Info tab) for that activity.
- Select the Log tab to review a script's execution log.
- In the **Suffix** field, the drop-down list includes two predefined suffixes (Date and DateTime). The list also includes any suffixes you have created and saved since installing Toad. To delete a suffix from the list, select the suffix, click (x)=, delete the suffix in the **Expression** field, and click **OK**.
- Automation does not support positional parameters denoted by a question mark.

Using Variables in Automation

Automation variables are a powerful tool that can greatly extend your scripts. Variables can add flexibility and customization. Variables can be used in almost every Automation activity.

How to Create Variables

User-Defined Variables

You can use the Set Variable activity to create user-defined variables. See "Use System Automation Activities" on page 78 for more information.

Built-In Variables

Several Automation activities create variables by default. The following built-in variables are created by Automation activities.

- RowCount variable— This integer type variable is created by several activities. The value assigned to this variable is the number of rows returned by the query in the activity.
- Return code variable— This integer type variable is created by the Run Automation Script and the Run Program activities. The variable value is assigned based on the execution status of the script or program.
- Built-in SQL type variables—The Loop Dataset and Execute Script activities create a SQL type variable which is used to store the result set of a query. The Loop Connections activity creates a SQL type variable with only one row containing connection information.
- Activity Result variable—This variable captures the result of each activity's execution. If an error occurs, it captures the error message. The name of this variable is _ACTIVITY_RESULT.

The Automation script creates the following built-in variables.

- Root path variable—This string type variable is created by the script. The value assigned to this variable is the path you specified in the **Root path** field in either the Test environment or the Production environment area in script Settings. The path that is applied depends on which environment (*Test* or *Production*) is selected at the time the script is executed.
- Run mode variable—This string type variable is created by the script. The value assigned to this variable is either *Test* or *Production*, depending on the environment selected in the script Settings at the time the script is executed.

How to Use Variables

Where to Use Variables

The following are some examples of where you can use a variable (built-in or user-defined) in an Automation script.

- In the file name of a file exported by an activity.
- In the suffix appended to a file name.
- In an expression used in the While or If Condition activities.
- In a SQL statement as a bind variable for filtering data.
- In a SQL statement as a bind variable in an Import or Export template.

How to Use a Variable

To express the value of a variable, surround the variable name with hash symbols (#), for example *#myvar#*. The following are some examples of how variables might be used.

```
#myvar#_report.xlsx

Report_#Dept_Name#.xlsx

#File_1_RCOUNT# > 0
```

It is important to remember that when a variable is expressed, the variable value is substituted as text in place of the variable name (#myvar#). The only exception is when you use a variable as a bind variable in a SQL statement.

To use a variable as a bind variable in a SQL statement, use the following syntax format (regardless of what the address_id column type is):

SELECT * FROM contact WHERE address_id = :myvar

You can also use the Automation variable here as text substitution, but in this case single quotes might be required (because it is a text substitution) depending on the column type in the WHERE clause.

For example, if last_name is a character column, use the following syntax (single quotes are required):

SELECT * FROM contact WHERE last_name = ‘#myvar#’

If address_id is a non-character column, use the following syntax:

SELECT * FROM contact WHERE address_id = #myvar#

How to Use a String Variable in a Comparison Expression

When comparing a string variable to a string value in an expression, always surround the variable name and the string value with single quotes, for example:

'#myvar#' = 'Dog'

About SQL Type Variables

SQL type variables are created in two ways:

- You can create user-defined SQL type variables using the Set Variable activity.
- A built-in SQL variable is created by the Loop Dataset activity, the Execute Script activity, and the Loop Connection activity.

If you ask just for the variable value of a SQL type variable, Toad returns the value from the first row and column of the result set. However, you can filter by column using this syntax:

<variable name>.<column name>.

For example, Region.Region_Name returns the first value (first row) in the column Region_Name.

Dataset/SQL Variable - Special Format for Filtering

You can use the following syntax format to filter data from the dataset variable (or any SQL type variable):

<dataset variable name>.<filter parameter>

Example: Loop_data_1_SQL.region_id, where region_id is one column in the dataset variable

For example, use this format in a SQL statement (in a Select to File activity) to select and export data based on each region_id value in the dataset variable. The statement is executed with each loop of the Dataset activity.

SELECT * FROM address WHERE address.region_id = :Loop_data_1_SQL.region_id

SELECT * FROM address WHERE address.region_id = #Loop_data_1_SQL.region_id#

Bind Variables

You can store global bind variable/value combinations in Toad . After storing a global bind variable, that variable is available to use in Automation. Your stored global bind variables appear in the list of variables in the expression editor of applicable activities.

How to Create and Use Date Variables

In Automation scripts, a common use case is to create a variable to be used later in the script in a SQL statement (such as in the Select to File activity). When creating a variable for a date value, there are two variable types to choose from in the Set Variable activity: *String* or *DateTime*. The type you choose depends on how the value is to be used in the SQL.

- If the SQL expects a string value, use the *String* variable type.
- If the SQL expects a date/time value, use the *DateTime* variable type.

How to Define a Date Variable

	In the Set Variable Activity	
Example SQL	**Variable Type**	**Valid Values**

`SELECT * FROM contact`
`WHERE birth_date < to_date(:var_date, 'YYYY-MM-DD')`
`and CONTACT_ID < 20`
	String	'2017-7-7'

`SELECT * FROM contact`
`WHERE birth_date < :var_date and CONTACT_ID < 20`
	DateTime	To_date ('2017-07-07', 'YYYY-MM-DD')
		'2017-7-7'

Using a String. For a *String* variable type, surround the date value in single quotes. This ensures that date delimiters, such as the dash (-) or forward slash (/), do not cause Toad to interpret the value as an expression.

> ‘2017-7-7’ is treated as a string
>
> 2017-7-7 is evaluated to 2003

Variable Scope

When you are building an Automation script that uses a variable, it is important to understand the variable's scope of visibility. This section describes the concept of variable scope in Automation scripts.

Terms and Definitions

Term	Definition
Container	Let's define a container as any one of the following: • The script itself • An activity that contains other activities (like the Group, While, or Loop Dataset activities) • A branch of an activity (for example, a branch in the If Condition activity)
Levels	Suppose that a script is composed of nested containers. Each container is a different level with respect to variable scope. Level 1—The outermost container is the script itself. The script is comprised of activities. The script and its activities in the main workflow comprise the first level. Level 2—Some activities can contain other activities. "Container activities" (such as the Loop Dataset) are at the second level. Level 2 is nested below level 1. Level 3—Container activities can contain branches. A branch within a container activity is the third level. Level 3 is nested below level 2. And so on, until we get to the innermost level.
Container Activity	A container activity is an activity that contains other activities. The Loop Dataset, Group, If Condition, While, and Parallel activities are container activities.
Set Variable Activity	Use this activity to define a variable.
Set Variable Value Activity	Use this activity to assign a *new* value to a previously-defined variable.

Scope Rules

Basically, the scope of a variable includes all activities subsequent to the variable-defining activity AND either at or nested below the level at which the variable is defined.

The following are the rules for variable scope.

- **Use a variable in an expression**—You can use a previously-defined variable in an expression in an activity if the following are true:
  - The expression activity is subsequent to the variable-defining activity
  - AND the expression activity is at the same level or at a level nested below the variable-defining activity
- **Reassigning a variable value**—You can reassign the value of a variable (except a SQL type variable) at any point within the variable's scope. This task is accomplished using the Set Variable Value activity. See "Use System Automation Activities" on page 78 for more information.

  Once a variable value is reassigned, the new value applies to activities subsequent to the original Set Variable activity AND either at the same level or nested below the level of the original Set Variable activity.

  Example: You can use the Set Variable Value to turn a previously-defined variable into a counter (var=var+1) in the Loop Dataset activity.
- **Define a new variable with the same name**—You can redefine a new variable with the same name, but only within an activity that is nested one or more levels below the variable-defining activity. For example, if you defined a variable in the main workflow of your script (first level), you can define a new variable with the same name within a container activity (second level). The scope of this variable value is restricted to that container activity.

  You *cannot* define a new variable with the same name at the same level as the original.

Additional Examples to Illustrate Scope

- Run Automation Script activity—This activity is used in an Automation script (the parent script) in order to execute another Automation script (the child script). However, a variable defined in the parent script cannot be used in the child script.
- Loop Dataset activity—This activity is a container activity and can contain other activities, including a Set Variable activity. However, a variable defined in a Set Variable activity within the Loop Dataset cannot be used outside the Loop Dataset.

Execute and Schedule Automation Scripts

You can execute automation scripts on demand from the Automation window or the Project Manager. You can also schedule scripts to run on a recurring basis at a specific time. For example, schedule a script that performs long-running tasks to run when database activity is light.

When you schedule a script in Toad, the Job Manager creates a Windows scheduling task. Scheduled scripts run in batch mode. If using another scheduling application, specify "`-batch=true`" in the command. This executes the script in non-visual (batch) mode. See the procedure below.

To execute a script from the Automation window

1. Click **Run** in the Wizard bar. The script is compiled and executes immediately.
2. To view the script's execution log, select the Log tab.
3. If you want to schedule the script, first resolve any problems that caused activities to fail or disable failed activities. An activity can be disabled from its Activity Info window.

   Tip: To execute a script from the Project Manager, right-click the script and select **Execute**.

To schedule an automation script

1. With a script open in Automation, click **Schedule** in the Wizard bar. A Windows Scheduler task is created for the script and a task properties dialog opens.
2. To schedule the script, select the Triggers tab. Select the scheduling trigger for this task and click **Edit**.
3. In the Edit Trigger dialog, specify scheduling details. Click **OK** to save your changes and close the dialog.
4. To run the script when you are not logged on, in the General tab select **Run whether user is logged on or not**. Then enter your Windows credentials when prompted.
5. When finished, click **OK** to save your changes and close the task properties dialog. For detailed information about the task properties dialog, see "Schedule Windows Tasks" in the online Help.

Note: To run a scheduled script against a password-protected database, the password must be saved in the Connection Properties dialog.

To execute a script in batch mode

- To start Toad and execute a script in batch mode (non-visual mode), enter the following command and arguments:

  <path to the toad.exe application> -batch=true “*<path to Automation script>*”

  For example:

  C:\Program Files\Quest Software\Toad Data Studio 1.0\toad.exe -batch=true "*<path\scriptfilename.tas>*"

Notes:

- Automation scripts are compatible with the version of Toad in which they are created. To learn how to upgrade your scripts and migrate scheduled tasks that were created in a previous version of Toad, see "Upgrade Scripts and Scheduled Tasks" in the online Help.
- The Automation module is disabled if Toad was installed with the **Prohibit saving passwords** option selected.

About Us

Quest creates software solutions that make the benefits of new technology real in an increasingly complex IT landscape. From database and systems management, to Active Directory and Office 365 management, and cyber security resilience, Quest helps customers solve their next IT challenge now. Around the globe, more than 130,000 companies and 95% of the Fortune 500 count on Quest to deliver proactive management and monitoring for the next enterprise initiative, find the next solution for complex Microsoft challenges and stay ahead of the next threat. Quest Software. Where next meets now. For more information, visit www.quest.com.

Contact Quest

For sales or other inquiries, visit www.quest.com/contact.

Technical Support Resources

Technical support is available to Quest customers with a valid maintenance contract and customers who have trial versions. You can access the Quest Support Portal at https://support.quest.com.

The Support Portal provides self-help tools you can use to solve problems quickly and independently, 24 hours a day, 365 days a year. The Support Portal enables you to:

- Submit and manage a Service Request
- View Knowledge Base articles
- Sign up for product notifications
- Download software and technical documentation
- View how-to-videos
- Engage in community discussions
- Chat with support engineers online
- View services to assist you with your product

Index