## Contents

1 Installing Aginity Pro  
1.1 Mac Install .................................................. 3  
1.2 Windows Install .................................................. 4  
1.3 Opening for the First Time .................................... 4  

2 Configuring Connections in Aginity Pro  
2.1 Downloading Drivers ............................................ 8  
2.2 Setting up Netezza Connections ................................ 9  
2.3 Setting up Redshift Connections ............................... 12  
2.4 Setting up Snowflake Connections ............................. 13  
2.5 Setting up Hive Connections .................................... 15  
2.6 Setting up SQL Server Connections ........................... 17  
2.7 Setting up Greenplum and Postgres Connections .......... 19  

3 Using the Database Explorer in Aginity Pro .................. 23  
3.1 Viewing the Database Data Dictionary ........................ 23  
3.2 Database Object Support by Platform ......................... 24  
3.3 Quick Actions from Explorer ................................... 25  

4 Using the Query Panel in Aginity Pro ........................ 29  
4.1 Writing your First Query ....................................... 29  
4.2 Executing to a File ............................................. 30  
4.3 Using Parameters in your Query ................................ 31  
4.4 Auto-Complete Capabilities ................................... 32  
4.5 Snippet support .................................................. 33  
4.6 Common Actions in Query Panel ............................... 33  
4.7 Interacting with Query Results ................................. 34  

5 Managing Files in Aginity Pro ................................. 39  
5.1 Working with File Projects ..................................... 40  

6 Using the Catalog in Aginity Pro ............................. 45  
6.1 Catalog Overview ............................................... 45  
6.2 Calling Catalog Items from SQL ............................... 47  
6.3 Adding Items to Catalog ........................................ 50  
6.4 Import and Export Items from Catalog ...................... 51
# Table of Contents

## 7 Using Discovery Panel in Aginity Pro
- 7.1 Searching Query Execution History ........................................... 55
- 7.2 Searching the Amp Pro Catalog ................................................. 56
- 7.3 Expanding the results ................................................................. 56
- 7.4 Doing More with the Discovery Panel .......................................... 56

## 8 Reference
- 8.1 Video Reference ................................................................. 57
- 8.2 Shortcut Key Reference ............................................................ 57
- 8.3 Frequently Asked Questions ...................................................... 57
- 8.4 SQL Reference Guides for Supported connections ......................... 58
- 8.5 Driver Support Reference .......................................................... 58

## 9 Troubleshooting ................................................................. 59

## 10 End User License for Aginity Pro ........................................... 61
Aginity Pro is our new SQL analysis tool. With cool new features and functions such as an active personal catalog, smart SQL discovery assistant, and more—beware, you may get spoiled. For more awesome features you can expect to see in Aginity Pro, check out the feature roadmap.

This documentation provides you with conceptual, procedural, and reference information to help you install and use Aginity Pro and find answers to questions you may have about it.

The documentation consists of the following sections:
The following sections will discuss installing Aginity Pro on both Windows and Mac. To start download the software if you have not from the Aginity Web Site

1.1 Mac Install

To start the Mac install click on the Pro “dmg” file you downloaded.

Drag the Aginity Pro.app icon into your Applications Directory. Once complete you can open the application from your Launchpad icon.
1.2 Windows Install

To start the Windows install click on the Pro “exe” file you downloaded.

You will be prompted to fill first choose whether to install as Only for me or Anyone who uses this computer (all users) on the machine. If you are installing the application on a VM or other shared infrastructure you should choose all users.

Secondly, pick the installation directory for the software.

Once complete you should be able to launch Aginity Pro.

1.3 Opening for the First Time

Once you have opened the application you will be presented with a clean instance of Aginity Pro.
Once you open Aginity Pro you will commonly do these Actions.

See also:

Add Connections
Configuring Connections in Aginity Pro

Aginity Pro allows you to set up connections to a variety of different database platforms. In order to do this, click the + symbol next to the “Connections” header in the left navigation pane of Aginity Pro. You will be presented with a list of existing connections, as well as options for adding another connection. Clicking the “New Connection” button will allow you to enter the necessary information for the connection you wish to create.

Below is a quick start video of creating connections in Aginity Pro

Note: Aginity Pro uses JDBC to connect to database platforms. It is recommended you refer to the JDBC guides for our supported platforms to learn about specific variables to use within the Advance Parameters section of Aginity Pro. We will provide links to that documentation where available.

The video below provides a more detailed explanation of how to use advanced parameters to illicit the desired behavior and successful connection to the database platform.
2.1 Downloading Drivers

The first step in configuring a connection is to download the JDBC driver for the platform you will be using. In most cases, we will provide access to the drivers which have been certified to work with Aginity Pro. In some cases such as when the vendor does not allow us to distribute or you want to use a different version, you will have to manually add the JDBC driver.

The video below provides a quick illustration of how to manage drivers.

See also:

Database Driver Reference Guide

To get started, in the Edit or Add Connection box you will click on Edit Drivers as shown below.

2.1.1 Automatic Download

As shown in the image below, if we are allowed to to download drivers you will be provided a link to do so and then be presented with one or more drivers to use and your done.
In some cases you may get an error message if your organization does not allow you to download drivers (jar files) typically supplied through a Maven repository. In this case please review the section below Manually Add Drivers.

### 2.1.2 Manually Add Drivers

If you have to manually add a driver it starts with first acquiring the JDBC jar for that platform and saving it somewhere on your machine. Once you acquire the JDBC jar you can upload it to Aginity Pro.

To do this you will hit the **Edit Driver or Edit Drivers Panel** section as shown above and you will be presented with the following screen.

You can name the Driver anything you want and provide a description.

The class name if not provided must be looked up typically in the JDBC reference manual from the platform vendor.

Finally if you click on the **Add File** button you will navigate to where you saved the Jar file and then click **Save Changes**. You are now ready to configure your connection.

### 2.2 Setting up Netezza Connections

You can set up a connection to a Netezza (IBM PureData for Analytics) appliance here. Simply populate the requested data into the form.
Note: Netezza does not allow us to distribute their drivers so you must obtain it from your DBA or directly from IBM Fix Central.
2.2.1 Using SSL with Netezza Connections

Aginity Pro supports SSL when setting up a Netezza connection. While setting up your Netezza connection simply select the level of security to enable from the dropdown selector. Your choices include:

- **Only Unsecured** - Specify this option when you want an unsecured connection to the Netezza system. If the Netezza system requires a secured connection, the connection is rejected.
- **Preferred Unsecured** - This argument is the default value. Specify this option when you would prefer an unsecured connection, but you accept a secured connection if the Netezza system requires one.
- **Preferred Secured** - Specify this option when you want a secured connection to the Netezza system, but you accept an unsecured connection if the Netezza system is configured to use only unsecured connections.
- **Only Secured** - Specify this option when you want a secured connection to the Netezza system. If the Netezza system accepts only unsecured connections, or if you are attempting to connect to a Netezza system that is running a release before release 4.5, the connection is rejected.

See also:

For more information on SSL refer to the link [Netezza SSL](#).

2.2.2 Connection Timeout

The connection timeout will tell the session when to disconnect. Typically set this to 10 minutes or 3600 seconds.

2.2.3 Advanced Parameters

For Netezza you can supply additional JDBC parameters using the **Advanced Properties** section.

See also:

Here is the link to other configurable parameters supported within [Netezza JDBC’s driver](#).
2.3 Setting up Redshift Connections

2.3.1 Using SSL Mode with Redshift Connections

Aginity Pro supports SSL when setting up a Redshift connection. While setting up your Redshift connection simply select the level of security to enable from the dropdown selector. Your choices include:

- **Disable** - SSL is disabled and the connection is not encrypted.
- **Prefer** - SSL is used if the server supports it. Amazon Redshift supports SSL, so SSL is used when you set
sslmode to prefer.

- **Allow** - SSL is used if the server requires it.
- **Require** - SSL is required.

**See also:**
For more information on Redshift SSL refer to this link

### 2.3.2 Advanced Parameters

For Redshift you can supply additional JDBC parameters using the **Advanced Properties** section.

**See also:**
Here is the link to other configurable parameters supported within Redshift JDBC’s driver

### 2.4 Setting up Snowflake Connections

You can set up a connection to a Snowflake here. Simply populate the requested data into the form.
2.4.1 Warehouse

A virtual warehouse, often referred to simply as a “warehouse”, is a cluster of compute resources in Snowflake. A warehouse provides the required resources, such as CPU, memory, and temporary storage, to perform the following operations in a Snowflake session. If you do not know which specific Warehouse to use speak to your Snowflake DBA.

2.4.2 Role

This is a security role that may have been setup for you to use. You can leave this field blank if you do not know what Role to connect as.
2.4.3 Advanced Parameters

For Snowflake most connection detail is handled through the command line additions to the JDBC URI. You can supply additional JDBC parameters using the Advanced Properties section.

See also:

Here is the link to other configurable parameters supported within Snowflake JDBC’s driver

2.5 Setting up Hive Connections

You can set up a connection to a Hive databases here. Simply populate the requested data into the form.
Connection name
Demo Hive Server

Database driver

Host (required)
demo-spark.aginity.com

Port
10000

Subprotocol (required)
Hive 2

User
jschliit

Password
**********

Database
amp-demo.jschliit

Use Kerberos for authentication

Realm (required)
Enter realm

Kerberos Data Center FQDN (required)
Enter host fqdn

Service Name (required)
Enter service name

Advanced Properties
2.5.1 Picking Subprotocol

You can choose either Hive 2 or Spark SQL syntax when executing against Hive databases. You can choose either value in this drop down.

2.5.2 Setting up Kerberos connections

In a lot of cases organizations require the use of Kerberos with Hive so by clicking on the radio button for Use Kerberos for authentication you will be prompted to enter the following:

- Realm
- FQDN
- Service name

Note: The value of these parameters are typically set by your Hadoop administrator and it is best to work with them to obtain the appropriate values.

2.5.3 Advanced Parameters

You can supply additional Hive JDBC parameters using the Advanced Properties section.

See also:

Here is the link to other configurable parameters supported within Hive JDBC’s driver

2.6 Setting up SQL Server Connections

You can set up a connection to Microsoft SQL Server databases here. Simply populate the requested data into the form.
Note: We only support SQL Server compatibility levels 150 and 140 which is SQL Server 2017 and greater.

You can set up a connection to IBM DB2 warehouse databases here. Simply populate the requested data into the form.
2.6.1 Advanced Parameters

You can supply additional Hive JDBC parameters using the Advanced Properties section.

See also:
Here is the link to other configurable parameters supported within SQL Server JDBC’s driver

2.7 Setting up Greenplum and Postgres Connections

You can set up a connection to Greenplum and Postgres databases here. Simply populate the requested data into the form. Greenplum and Postgres share the same drivers so we will speak of them together.
2.7.1 Choosing Subprotocol (Only for Greenplum)

You can choose either generic Postgres or Pivotal Greenplum SQL syntax when executing against Greenplum databases. You can choose either value in this drop down.
2.7.2 Advanced Parameters

You can supply additional Postgres JDBC parameters using the Advanced Properties section.

See also:

Here is the link to other configurable parameters supported within Postgres JDBC’s driver

See also:

Navigating Database Explorer
CHAPTER 3

Using the Database Explorer in Aginity Pro

The Database Explorer panel in Aginity Pro can be toggled by using the Database Explorer and Catalog tabs on the left hand pane of the application as show below. The Database Explorer is used to navigate database objects based on the Connections defined in Aginity Pro. The Catalog browser is used to navigate SQL Analytics that you have personally stored.

Below is a quick start video of using the Database Explorer in Aginity Pro

3.1 Viewing the Database Data Dictionary

Once you have established a connection to a database you can browse the database dictionary catalogs. You can toggle to the Database Explorer browser by clicking on the tab as shown below. The database data dictionaries typically contains the tables, columns, views and other database objects available to query from the query panel.

Once you have selected the Database Explorer tab you can drill into each connection as shown below.
3.2 Database Object Support by Platform

The following image details what objects are visible from the browser for all of our currently supported platforms. We do releases every two weeks with Aginity Pro so if there is a feature you need please contact us at support@aginity.com.
### 3.3 Quick Actions from Explorer

Currently you can do the following quick actions from the **Database Explorer**

#### 3.3.1 SQL Statement Generation

To generate typical SQL statements from tables or views you can right click on the object, navigate to scripts and choose the appropriate script as shown below.
3.3.2 Selective SQL Generation on a Set of Columns

To generate SQL statements based on a selection of columns within a Table or View, multi-select the columns by holding Ctrl (Windows) or Command (Mac) and selecting the columns as shown below. Then generate the SQL statement.

**Hint:** Make sure you don’t have the table name selected and column names at the same time. If you do it will generate the SQL for the whole table.
### 3.3.3 Drag and Drop Capabilities

You can drag and drop any object from the DB Explorer into the Query Panel.

**Note:** Double-clicking expands the DB Explorer tree.

### 3.3.4 Copy JDBC Connection String

If you select any Connection (Top level of the tree) and select “Copy Connection string to Clipboard” you will get the full JDBC URI which is helpful for debugging. See this Knowledge Base Article for help with debugging your Aginity Pro connections.

See also:

- *Use Catalog Browser*
- *Use Query Panel*
Using the Query Panel in Aginity Pro

You will spend most of your time in the Query Panel writing SQL and saving it to your catalog in Aginity Pro. The video below will give you an overview of creating your first query.

4.1 Writing your First Query

Each tab in the Aginity Pro application can be set to different database connections. Once you establish that connection as shown below you can use the query panel like a text editor using SQL syntax specific to that database platform.

Once you write your code in Aginity Pro you have options to run it using the following scope:
Execution Type | Description
--- | ---
Run at cursor | When executing the query it will run the statement where the cursor is placed up until the semicolon.
Run all parts of the script in a sequence | Each statement in the window will be run sequentially and the output of each statement will be shown in the output and if the statement was a SELECT then a results panel will be displayed.
Run entire script as a batch | 
Abort Query | After executing a query you can direct Aginity Pro to send a request to the database to cancel the query.

See also:

*Note one of the greatest challenges in SQL is learning the different functional syntax from different database vendors. We are trying to make that easy by giving you links to the SQL reference manuals we support.*

### 4.2 Executing to a File

In version .17 of Aginity Pro we released the ability to use our editor commands to spool the results of a SQL statement(s) to a file(s).

To do this you must issue the following commands:

1. Use the `#+sql` command to tell the editor you will be writing SQL in this section.
2. You will then use the :output file command to tell the editor you will spool the results to an output file.
3. In the table below is a series of parameters and descriptions to be used when spooling.
4. Use the `#+begin` statement to tell the editor you from here until the `#+end` command you will have SQL to execute.
5. Write your SQL
6. End with `#+end`
7. Use the Run commands as you would any normal SQL statement as described in the Run Scope section above.

The code below is an example you can use to build off of.

```plaintext
-- This will write the SQL statement between `#+begin` and `#+end` to the file specified.

`#+sql`
`:=output file`
`:path '/users/jschlitt/Downloads/jeff1'
`:delimiter ','
`:null_value 'NULL'
`:quote_char ''
`:overwrite true`
`#+begin`
`select * from wb_pro_marketing_list `
`where emaildomain = 'aginity.com'
`#+end`
```
Note: We only support today the file type of csv. You can still change the delimiter but all files with have a csv extension.

The table below illustrates the current values accepted in execute to file commands:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Options</th>
</tr>
</thead>
</table>
| path      | On Mac this should be in the format “/dir/dir/filename”. On Windows it should be in the format “drive:directoryfilename”.
| delimiter | You can use any value but we recommend , or | or ^ or t for (tab)
| null_value| enter a character field for null typically ’ ‘ (empty string) or ’NULL’
| quote_char| enter how to enclose character string typically the quotation (")
| overwrite | whether or not to allow overwrite if file exists: true or false |

Note: At Aginity, we produce lists for marketing purposes using a single catalog item called “Product Daily Marketing Feeds” that execute twenty five queries and spool the results to twenty five csv files which we upload to our marketing engine all with the click of the Run button in Aginity Team.

4.3 Using Parameters in your Query

Parameters can be used in Aginity Pro by using the $ in any part of a SQL statement. When you execute a query the Aginity Pro application will ask you to fill in the value of the parameter before it sends the query to the database.

```sql
--sample parameter query
select * from landing_dim_customer where gender = $fill_in_gender_at_runtime limit 100;
```

You will have datatype choices at runtime as shown in the table below.

<table>
<thead>
<tr>
<th>Parameter Data Type</th>
<th>Action Taken</th>
</tr>
</thead>
<tbody>
<tr>
<td>As Is</td>
<td>Aginity Pro will send the value exactly as written with no casting. This is often used when using the parameter to fill in database table or column names. See example 2 below.</td>
</tr>
</tbody>
</table>

Hint: You can specify default values with parameters by using this syntax with your parameter

```sql
$fill_in_gender_at_runtime{‘M’}
```

When you run the query you will see the little gear icon as shown below. Clicking that will populate the default value.
4.4 Auto-Complete Capabilities

In Aginity Pro we have implemented auto-complete on several database platforms and will be deploying across all over time. The auto-complete capability has three capabilities.

1. Display any words on the editor screen from above where your current cursor is
2. Display SQL Syntax like SELECT, WHERE, FROM all the way to database platform specific functions like CASE, DATEDIFF or CURRENT_DATE()
3. Display the database, schema, table, column as you are typing

Hint: To initiate just the database browser which shows database objects plus snippets hit the Ctrl+Space button otherwise as you type if auto-complete is supported it will automatically popup.

Note: Currently we support auto-complete on Netezza, Redshift and Snowflake
4.5 Snippet support

We have pre-wired some special snippets within the editor as well to make writing SQL simpler. The table below details those snippets that are currently available.

<table>
<thead>
<tr>
<th>Snippet</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>selall</td>
<td>will generate generic SELECT FROM</td>
</tr>
<tr>
<td>selwh</td>
<td>will generate generic SELECT FROM WHERE</td>
</tr>
<tr>
<td>upd</td>
<td>will generate an UPDATE statement</td>
</tr>
<tr>
<td>ins</td>
<td>will generate an INSERT statement</td>
</tr>
<tr>
<td>del</td>
<td>will generate a DELETE statement</td>
</tr>
<tr>
<td>expf</td>
<td>will generate a generic export to file statement</td>
</tr>
</tbody>
</table>

4.6 Common Actions in Query Panel

We will highlight common functions heavily used in the editor here. There is a complete list of available editor functions which can be accessed by hitting the Command+Shift+P on Mac or the Ctrl+Shift+P button within the editor or from the View->Command Palette menu item.

4.6.1 Renaming Tabs

You can rename the query tabs by double clicking and typing in a name as shown below.

4.6.2 Commenting SQL

In Aginity Pro you can use the typical SQL syntax to comment your code as shown below

```sql
-- This is one way to comment your SQL code
SELECT AVG(LineTotal)
FROM sales.SalesOrderDetail

/* This is another way to comment your SQL code */
```
4.6.3 Keeping Connection

The Keep Connection checkbox as show below will instruct Aginity Pro to maintain the connection to the database if the default behavior of the database is to disconnect after the execution of a query. You are still subject to an overall database connection timeout if your session connection exceeds the permitted database setting.

4.6.4 Find and Replace

Ctrl+F (Windows) or Command+F (Mac) will bring up the find and replace box as shown below.

4.7 Interacting with Query Results

Once your query has been executed you can use the features of the result grid to interact with the result set. The typical actions you will do are listed below.

- Reordering the Result Grid
- Sorting the Result Grid
- Filtering the Result Grid
- Exporting results to CSV

4.7.1 Reordering the Result Grid

Once a query has executed successfully, you may rearrange the columns in the Result Grid in order to more effectively perform subsequent analyses. Rearranging the Result Grid is simple: simply click, hold, and drag the column header for the column you wish to move. As you move it left or right, the grid will automatically adjust to reposition the column.
Once you are satisfied with the new position of the column, you may release the drag and the column will lock into place.

### 4.7.2 Sorting the Result Grid

After successfully executing a query, if the query has returned any rows you may sort the results in order to permit various types of analysis of your results.

In order to sort your results, simply click on the header row in the results grid. If the results are already being sorted by that column, then the sort order will toggle between ascending and descending.
4.7.3 Filtering the Result Grid

Data in the Result Grid may be filtered to make it easier to locate and analyze certain subsets of the resultant data, without requiring you to submit an entirely separate query.

To filter the result grid by values in a particular column, bring your mouse cursor to the column header for the column to which you wish to apply a filter. In the right-hand side on the header row the filter icon (three horizontal bars) will appear. Click on the filter icon and you will be presented with a pop-up menu that will allow you to enter your filter criteria.

By default, the filter criteria will select a “Contains” filter, but several other filter types exist and you may select any of them. Once you have selected the filter type you wish to apply, type the filter into the text box. The results will immediately begin to filter.
When a column is being filtered, a funnel icon will appear in its header. To remove the filter, you may re-enter the filter mode by clicking on the filter icon (three bars) in the header row, and simply delete the filter text.

Multiple columns may have active filters at the same time. To accomplish this, simply perform these steps on each column.

### 4.7.4 Exporting results to CSV

Once you have successfully executed a query you can elect to export those results in Comma Separated Value (CSV) format to a file outside of Aginity Pro. To do this, right-click anywhere in the results panel. Doing this will display a context menu from which you will select the option “Export to CSV”.

After selecting this option, you will be presented with a screen that permits you to select the location to which to save your results. Once you have identified the proper location, press the Save button to complete the export.

CSV files may subsequently be imported into a large number of different programs for further review and analysis.

See also:

- *Adding Query to Catalog*
- *Using Discovery Panel*
You can now open files with Aginity Pro! You’ll notice in the File menu commands for Open, Save and Save As. This commands will let you interact with files directly on your hard drive. By doing this you can now easily open your existing SQL files as well as start to govern your SQL in both files and in the Aginity Catalog. At Aginity we tend to use files to store SQL that we are in the process of developing and then use the Catalog as a place to store finished SQL assets for personal or collaborative use.

Once you click on a file menu item you will be presented with a typical file explorer for your O/S as shown below.
In the Query Panel the tab header will display with a symbol as shown below to let you know you are working with a file. Note the symbol * will denote that you have made changes and need to save the file. Even if you don’t save the file and exit Aginity Pro, you’ll changes will be there since we maintain state of all tabs regardless. You do need to actually save though if you open the SQL file in another editor.

5.1 Working with File Projects

On the left hand side of the object browser you’ll see a new tab for Project. This allows you to open an entire folder off your hard drive and manage the SQL Files within them. By default you will see a button prompting you to Open Project as shown below.
When you click Open you’ll be prompted with your O/S specific file explorer asking you to open a folder to displayed

**Note:** This is not asking you to Open a SQL File but instead an entire folder. If you want to just open a SQL File use the File - Open menu item.

Once you open the folder you will see the entire folder structure in the object browser as show below.
To open and start editing a file in Aginity Pro just double click.

Right clicking on any file within here presents you with several options listed and described below.

- **New file** - Will create a new file in the enclosed directory
- **New directory** - Will create a new folder in the enclosed directory
- **Cut file** - Will use clipboard to cut the file
- **Copy file** - Will use clipboard to copy the file
- **Rename** - Allows you to rename the file
- **Delete** - Delete the file; note this is an actual physical delete from disk not just within Aginity Pro
• Open with encoding - If you file uses a specific encoding you can’t specify that here
• Copy file path - Copy the file path
6.1 Catalog Overview

The query catalog is a unique capability of Aginity Pro allowing you to easily save, manage and use items stored in it. Let’s start with browsing the catalog.

The catalog is organized by any number of directories and sub-directories of queries as shown below.

See also:

Click Some link to best practice to see best practices for organizing catalogs.

Queries are given friendly names and can be used or accessed in several ways. The first way is to drag the catalog query into a query panel as shown in the two images below

You can also edit queries within the catalogs. By right clicking on a query you can change the SQL itself, the description and folder options.
Fig. 1: You can drag the query into the empty query panel.

Fig. 2: After you drag the item it will present the full catalog query for edit or execution.
6.2 Calling Catalog Items from SQL

One of the most innovative features of Aginity Pro is the ability to call catalog items from within SQL statements. The video and following examples below will give you basic syntax required to call a catalog item.

6.2.1 Single SQL SELECT Statement

In this example we have a single SQL SELECT statement in the catalog called `OrderAverage`.

This is the code in the Catalog.

```
SELECT AVG(LineTotal)
FROM sales.SalesOrderDetail
```

To call this from the query panel you would issue the following command. Note the highlighted reference to the object in the catalog.
### 6.2.2 Using Catalog Item to Facilitate Relationships

In this example we will predefine a join in the catalog called `ClientAddress` and then use it in the context of another SQL statement.

This is the code in the Catalog.
```sql
clients c
JOIN addresses a
ON c.clientsid = a.clientsid
```

To call this from the query panel you would issue the following command. Note the highlighted reference to the object in the catalog.
```sql
SELECT *
FROM ${ClientAddress};
```

This is the code passed to the database.
```sql
SELECT *
FROM clients c
JOIN addresses a
ON c.clientsid = a.clientsid;
```

### 6.2.3 Creating and Reusing Calculations

In this example we will predefine a Calculations in the catalog called `Residual` and then use it in the context of another SQL statement.

This is the code in the Catalog.
```sql
1-total
```

To call this from the query panel you would issue the following command. Note the highlighted reference to the object in the catalog.
```sql
SELECT item,
@$\{Residual\}$
FROM table;
```

This is the code passed to the database.
6.2.4 Executing a Multi-Step script

In this example we will predefine a series of SQL statements in the catalog called **DailyRun** and then execute it by a single call. Any DML, (SELECT, CREATE, UPDATE, INSERT, DELETE) can be used.

This is the code in the Catalog.

```sql
CREATE TABLE xyz AS ...;
CREATE TABLE abc AS ...;
SELECT * FROM xyz JOIN abc...;
```

To call this from the query panel you would issue the following command. Note the highlighted reference to the object in the catalog.

```sql
@{DailyRun}
```

This is the code passed to the database.

```sql
CREATE TABLE xyz AS ...;
CREATE TABLE abc AS ...;
SELECT * FROM xyz JOIN abc...;
```

6.2.5 Using Parameters in Catalog

In this example we will use a parameter within a catalog item called **CitySelect**.

This is the code in the Catalog. Note the highlighted line has the parameter in it.

```sql
SELECT *
FROM addresses
WHERE city = $city;
```

To call this from the query panel you would issue the following command. Note the highlighted reference to the object in the catalog.

```sql
@{CitySelect}
```

This is the code passed to the database.

```sql
SELECT *
FROM addresses
WHERE city = 'Chicago';
```

**Note:** Upon execution of the command Aginity Pro will ask you to supply a parameterized value. For more information on parameters see Using Parameters.
6.3 Adding Items to Catalog

To add a new query to the catalog, highlight any part of a SQL statement(s), right click and select **Save selection to catalog** as shown below.

Once you save you will be asked to provide metadata about the query you are adding.

**Note:** It is important to fill out the description metadata as it will be used in searching. The more information you provide about the context of usage, key business terms and intended usage the better search results you will get.

You can also easily create new folders in the catalog or add SQL entries directly from the **Catalog Browser** by clicking the (+) to the right of the Catalog header.

When adding a new folder you will be presented with a basic text box to create your folder as shown below.
6.4 Import and Export Items from Catalog

You have the ability to Import and Export both your full catalog as well as any folder selected within the catalog.

**Note:** We highly recommend taking regular exports of your catalog to disk to prevent accidental loss of queries.

6.4.1 Export Steps

To Export your catalog start by either **Right Clicking** on a Catalog Folder or Using the Aginity Pro -> Export Catalog menu as shown below.

![Export Catalog Menu](image)

Your first step is to save either an entire folder or a selection of items within a folder.
The final step in exporting is to name the file and location you will save to as shown below.

6.4.2 Import Steps

Importing is valuable for several reasons. One is to restore SQL catalog items you may have lost or more importantly the ability to collaborate with other SQL developers and share your SQL making your a more efficient and consistent...
Aginity-Pro Documentation, Release 1

SQL developer.

The first step is to **Right Click** on a folder you want to import to and choose Import Catalog as shown below.

![Import Catalog](image)

**Note:** You can also use the Aginity Pro Menu -> Import Catalog. The main difference between using these two capabilities is the context of where you are importing items. If you choose to import from the Aginity Pro menu you must set the context where to import. If you **Right Click** on a folder the context is set to that folder.

After choosing to import you will choose the file to import. By default Aginity catalog files have the extension `.aginitypkg`.

After choosing your file you can specify the folder to import to and what to import as shown below.
Hit the **Import** button and your queries should be imported!

**See also:**

Aginity also maintains a Github repository of Catalog queries to be used. These queries are either administrative queries for specific platforms, data engineering queries that do powerful transformations against generic data or analytic queries showing how to do cool things in SQL.

Access the repositories [here](#)

**See also:**

*Using Query Panel*
CHAPTER 7

Using Discovery Panel in Aginity Pro

The Discovery Panel is a feature that is unique to Aginity Pro and which allows you to quickly identify and reuse code that you’ve previously executed or saved. Using the Discovery Panel begins with a search. Watch the video below to get a sense for how this feature works.

By default, the Discovery Panel allows you to search both your query execution history and your Pro Catalog at the same time. However, by selecting one of the radio buttons in the Discovery Panel you can choose to focus on just one or the other if you prefer.

7.1 Searching Query Execution History

Start by typing in any free-text you want such as Sales as shown Below
The search engine will do a full text search and display all results.

### 7.2 Searching the Amp Pro Catalog

Buy clicking on the Catalog radio button you will restrict your search as shown below.

### 7.3 Expanding the results

Once you have performed a search, Amp Pro will dynamically update the results in the Discovery Panel to match the search term you have entered. You’ll be able to view a small amount of the query by default, but if more is required you can click on the arrows in the corner of the preview in order to expand the snippet further.

### 7.4 Doing More with the Discovery Panel

A placeholder for advanced filtering capabilities.
8.1 Video Reference

The following is a reference list of videos about Aginity Pro and how to use its features.

- 
- 
- 
- 
- 

8.2 Shortcut Key Reference

<table>
<thead>
<tr>
<th>Command</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command K + C (Mac) or Ctrl K + C (Windows)</td>
<td>Comment lines of code</td>
</tr>
<tr>
<td>Command K + U (Mac) or Ctrl K + U (Windows)</td>
<td>Uncomment lines of code</td>
</tr>
<tr>
<td>Command F (Mac) or Ctrl F (Windows)</td>
<td>Find and Replace Menu</td>
</tr>
</tbody>
</table>

8.3 Frequently Asked Questions

The section below is dedicated to commonly asked questions that we compile and provide answers to. Over time as things change we will keep this table up to date.
8.3.1 What Control do I have with Row Limit Preference?

By default we cap the row limit preference to 100K rows to discourage users from spooling too much data across your network. Trust us your network infrastructure folks do appreciate it. Fear not though, you have options. First, if you set the preference to 0, it will spool as much as you want until either you run out of memory or your query completes. Alternatively, we recommend spooling the output to a file if you are doing returns greater than 100K rows. Refer to the documentation on execute query to file.

8.4 SQL Reference Guides for Supported connections

As SQL developers ourselves we thought putting links to the databases we support in our documentation was a good idea so here they are!

- 
- 
- 

8.5 Driver Support Reference

The table below is a cross reference of driver versions by platform

<table>
<thead>
<tr>
<th>Platform</th>
<th>Driver Version</th>
<th>Class Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Netezza</td>
<td>Not distributed with software (nzjdbc.jar)</td>
<td>org.netezza.Driver</td>
</tr>
<tr>
<td>SQL Server</td>
<td>7.2.0 jre8</td>
<td>com.microsoft.sqlserver.jdbc.SQLServerDriver</td>
</tr>
<tr>
<td>Redshift</td>
<td>1.2.16.1027</td>
<td>com.amazon.redshift.jdbc.Driver</td>
</tr>
<tr>
<td>Greenplum and Postgres</td>
<td>42.2.5</td>
<td>org.postgresql.Driver</td>
</tr>
<tr>
<td>Hive</td>
<td>1.2.1</td>
<td>org.apache.hive.jdbc.HiveDriver</td>
</tr>
<tr>
<td>Snowflake</td>
<td>3.6.23, 3.6.28, 3.5.5</td>
<td>net.snowflake.client.jdbc.SnowflakeDriver</td>
</tr>
<tr>
<td>IBM DB2 Warehouse</td>
<td>db2jcc4</td>
<td>com.ibm.db2.jcc.DB2Driver</td>
</tr>
</tbody>
</table>
This page is dedicated to common issues or errors you may encounter when using Aginity Pro. We recommend you also post any new issues or questions to our Aginity Pro Forum.

The is a great resource for posting questions, searching for common solutions to features of Pro or just learning how to be efficient as a SQL analytic developer through samples, best practices and collaboration with other Pro users!

<table>
<thead>
<tr>
<th>Issue Number</th>
<th>Is this a Defect</th>
<th>Source of Error</th>
<th>Description of Error</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>No</td>
<td>Cannot Download Drivers via Connection Screen</td>
<td>In most cases your organization is blocking Aginity’s ability to download drivers from a Maven repository</td>
<td>Follow directions in the Configure Connections documentation to Manually Add Drivers.</td>
</tr>
</tbody>
</table>
End User License for Aginity Pro

Click here for end users license