

SQL Clients and Progress

By Allegro Consultants, Ltd.

Many reasons exist why someone would want to connect an SQL client to a Progress application database. There's virtually no end to the types of SQL client software packages available that provide high-end reporting capabilities, database utilities, and development platforms. Hundreds of third-party SQL tools are available – too many to list here. Here are just a few software packages available: Crystal Reports, ApexSQL, iSQL, and Adept Software.

So what does that mean to a Progress application user? Have you ever wanted more reporting capabilities than what the canned reports that came with your software offered? If not, then you are fortunate because hardly a day goes by for Allegro that a customer requests for custom or report modifications. Many customers feel there are little to no options to expand their reporting capability. Even if they had a reporting tool like Crystal Reports, they wouldn't understand how to generate a report because of poor database constructs from the vendor.

By creating an SQL broker against your Progress database, you are no longer bound by the limitations of your software application. This does not mean all of your problems are over, but you can obtain insight into your data that wasn't there before. This also means you expose your data to risk that wasn't there before as well. You must have a plan that includes the following: what data you will make available, who has access to the data, and what privileges you will grant to each user. You do not want just anyone being able to look at sensitive data such as the 'executive' payroll, so security should have a significant influence in the implementation of SQL access.

Security should be the number one consideration when adding SQL access to your mission critical application data. The second consideration should be what the SQL user should see when the connection is established. Many database schemas are difficult to understand due to; lack of support documentation, no field or table description data provided in the schema, and normalized data that only the architect would understand. Data security and data constraints can, and usually do, drive the need for schema views. With views, you can limit what the user sees, present the data in a more meaningful format, and even include calculated fields.

Once a well laid out SQL schema is created, a typical power-user can create his or her report(s) and not have to rely or wait on IT to schedule a developer. Often a tool like Crystal Reports can generate a report in the fraction of time that a programmer would take and at half the cost. Lastly, the possibility to misinterpret the requirement is removed when the user generates his or her report. I have been in environments where one power-user generated all of the Crystal reports and each user was given a Crystal Reports viewer. Two examples of viewers are [Easy Street Software's EasyView](#) and [Millet Software's DataLink Viewer](#), both offering BOXI compatibility at an affordable price. I personally like Millet Software because if you have questions or concerns, you can speak directly with the architect, Ido Millet.

A word of caution about your SQL client software, please make certain it is, or has the ability to be, Oracle SQL compliant. Out of the box, Crystal Reports is MS SQL compliant. However, it is has the option to be Oracle, MS, or IBM SQL compliant through a database registry entry. The Progress ODBC drivers available support the Oracle SQL implementation. Almost everything I did worked fine until I used a complex query that had an embedded LEFT-OUTER join. At that point, I found out that SQL wasn't just SQL.

These are all great reasons why adding an SQL broker to your Progress database makes smart sense, so the next logical question is how you create that connection. The remainder of this document will walk you through the steps to create a 'successful' SQL connection.

Create a SQL Broker

Don't create a single broker to be used for the purpose of both Progress and SQL connections; you are going to experience problems. For instance if you have a maximum of two agents available for a single broker that is defined for SQL and Progress, and both agents are initially started through Progress connections, they will remain as Progress agents until they are killed or the database is restarted. The error message you will see associated with this problem is as follows:

Exceeding permissible number of connections (7625)

Avoid this problem at all cost and start a separate broker. Also, consult with your database administrator to set the appropriate startup parameter to help ensure stability.

Establishing a SQL DB User

First and most important, this is a deployment requirement that can easily be executed incorrectly. There must be a valid SQL user setup in your database, with appropriate access rights, or no SQL client will be able to communicate with the database. This may be the single most important requirement to implement. If this step is not implemented correctly, you may find yourself fighting with permission and access issues for a long time. On a couple of occasions, I have found that correcting the problem was a struggle.

Users and Privileges

The Progress database has tables for managing user access from both a 4GL and a SQL client. The table “_user” contains login names and password for both the 4GL and SQL interfaces and both interfaces are using this table to validate login/passwords.

Access rights and privileges are handled completely differently for each of the two interfaces; therefore, the management of this information varies significantly.

- The 4GL 'privileges' are stored directly in the _file and _field tables (those are the "CAN-*" entries)
- The SQL92 privileges are stored in the following 3 tables:
 - sysprogress.sysdbauth (for database wide privileges)
 - sysprogress.systabauth (for table / Views/ Stored procedures / sequences level privileges)
 - sysprogress.syscolauth (for constraint privileges at column levels - giving the ability to create Foreign Keys on the specific column)
 - SQL DBA Account: Is the user name under which the database was created.

Create New SQL User and Grant Access Rights

If no database administrator account exists for SQL then start with step one; otherwise proceed to step two.

- 1) Create database administrator account
 - a. Connect to the database using Data Administration tool. Supply the DBA user name and password when connecting.
 - b. Choose "Admin" > Security > "Edit User List"
 - c. Create Progress default DBA user "sqluser" as a user ID.
 - d. Assign a password, for example, "sqluser".
 - e. Connect to the Progress SQL Explorer using the above user name and password.
 - f. Exit the 'Data Administration' tool and stop the progress DB service. The SQL database service should continue to run; if not, you must start just the SQL service.

DO NOT create the 'SYSPROGRESS' login (unless you are forced to do so because the database was converted from V8 or because there is no other DBA for this database). This account is internal to the SQL92 engine. On a new database, if you attempt a connection with sysprogress/sysprogress (login/password) you should get this error message: **Access denied (Authorization failed)**

- 2) Create SQL user account and grant DBA access
 - a. Launch the Progress SQL Explorer tool or any SQL 92 interface
 - b. Connect to the SQL database service as user 'sysprogress'
 - c. Create a user and password and grant DBA privilege to that user using the following SQL commands:
 - i. create user 'sqluser', 'sqluser';
 - ii. commit;
 - iii. grant dba to sqluser;
 - iv. commit;

Provided the above instructions did not produce an error then the required user account has been created.

Who is the original DBA?

- Two primary ways to determine who is the DBA:
 - Look in the database log file and you will see the following message:
"Procopy get default SQL DBA session begin for "your name" on CON:.. (451)"
 - List all entries stored in "*dbname.sysdbauth*" and use a login that is not SYSPROGRESS. Any valid login and password should have sufficient privileges to execute this query.

NOTE: When the database is newly created, the password associated with this login is any character string (not an empty string).

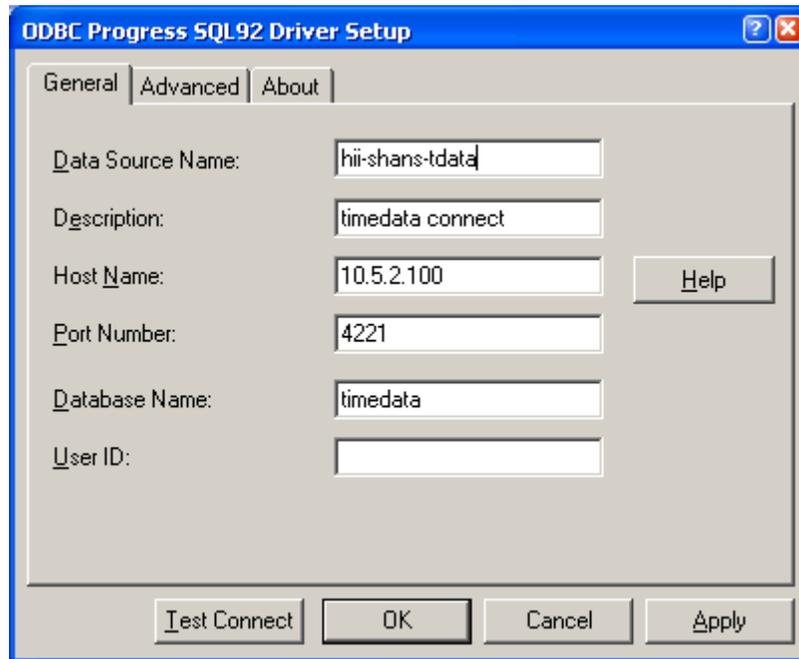
ODBC Drivers

Data Direct 4.1: driver is the most up-to-date SQL92 interface module for Progress 9.1D available today.

Merant 3.60: I've had very good success using the Merant driver. However, quite a few postings on the PEG involve problems with this, which leads me to believe some potential platform issues exist.

Creating ODBC Connection

All PC's that will be running an SQL client tool must have an ODBC Data Source Name (DSN) defined in the ODBC registry. The following setup is a requirement for both.



- 1) **Data Source Name:** The name of the data source connection, which should be somewhat meaningful.
- 2) **Description (optional):** In addition to the name, you can further define the data source by adding a description that provides enough detail that even a novice will know what type of data they can expect to see from this connection. Only in an environment where there is only one data source should you not provide a description and even then it's still a good idea to provide one.
- 3) **Host Name:** The 'host' name or IP address of the server running the SQL data service.
- 4) **Database Name (optional):** Name of the database to which it will be connected.
- 5) **User ID (conditional):** It is simplest to enter your user ID here. However, you are then at risk of someone use your connection maliciously.

SQL Client Deployment Checklist

Step	Description	Responsible Entity	Complete?
1.	Add SQL administrator user account to the database	DBA	<input type="checkbox"/>
2.	Verify account exists and has DBA privileges	DBA	<input type="checkbox"/>
3.	Load ODBC Driver	Deployment Script	<input type="checkbox"/>
4.	Load ODBC DSN	Deployment Script	<input type="checkbox"/>
5.	Modify SQL client data connection	Deployment Script	<input type="checkbox"/>

Frequently Encountered ODBC Errors

Error #	Description	Resolve
8587	outer-join error: Developer does not have registry entry to define SQL command syntax (Oracle, MS, DB2)	Make certain the SQL client software is Oracle SQL compliant, and if not, change the required parameters.
8184	'Character string is too long:' comparison value exceeds length of defined field length	Verify that you're not comparing against a literal value that exceeds the defined length of the field being compared. If you experience this problem, you may be able to correct this problem by following the steps outlined in Progress Trouble report P24496. I have included this fix on the next page.
-20217	"Failed to retrieve data from the database"	This is an ambiguous error message. If for ANY reason the ODBC is unable to complete the task, this error is given. A 10k size limit with the Data Direct driver might be causing you problems. If the SQL client allows you to see the SQL statement that it created, then check the size of the query.

Progress to SQL Width Issues

ID: P24496
Title: "What is the DBTool?"
Created: 5-May-2003 Last Modified: 25-Jul-2005
Status: Verified

Goal(s): How to identify problems with the SQL width when using PeerDirect replication

Purpose: DBTool is a new character mode application that allows Progress to identify problems with the SQLWidth and record format errors in our database. It is primarily for use with replication using PeerDirect. This tool was first included in Service Pack 6 for Progress 9.1D.

Command Line Syntax: dbtool <dbname>

Options: DATABASE TOOLS MENU - 1.0b

-
1. SQL Width & Date Scan w/Report Option
 2. SQL Width Scan w/Fix Option
 3. Record Validation
 4. Record Version Validation
 5. Read database block(s)
 9. Enable/Disable File Logging
 - Q. Quit

Option 1: Will find the maximum field sizes and report them.

Option 2: Will find the maximum field sizes and update their width.

Option 3: Will validate the schema versioning of the records after records are updated by the tool

Option 4: Will validate the schema versioning before and after the records are updated by the tool

Option 5: Will display the amount of records find by recid, table and area

Option 9: Will enable or disable the logging of the tool , if enabled a dbtool.out file will be generated

The usage of this tool is very helpful when trying to determine if the SQLWidth of a given field has been exceeded and we need to repair this SQL width.

For Additional Help - Contact Allegro's support at (804) 553-1130 or at support@allegroconsultants.com.

© Copyright 2009, Allegro Consultants Ltd., All Rights Reserved