Progress
psdn
Software Developers Network℠

Getting Started with Crystal 9 and Progress®

Authors: Sébastien Haefelé and Jim Lundy
Edition 1, March 2003

PROGRESS
SOFTWARE

Contents

1 Introduction

The purpose of this white paper is to provide basic guidelines when starting to use Crystal Decisions™ Version 9 products with a Progress® Version 9.1D database, which Crystal Decisions products access through the Progress SQL-92 interface. As a result, this document is targeted towards a technical audience with knowledge about Crystal Decisions' product line and with some understanding of Progress and the SQL-92 standard.

Through the whole document an example of setup is provided in order to establish and use a connection to a standard Progress database (Sports2000) from Crystal Reports 9. The database was created with the following command:

```
prodb sports2000 sports2000
```

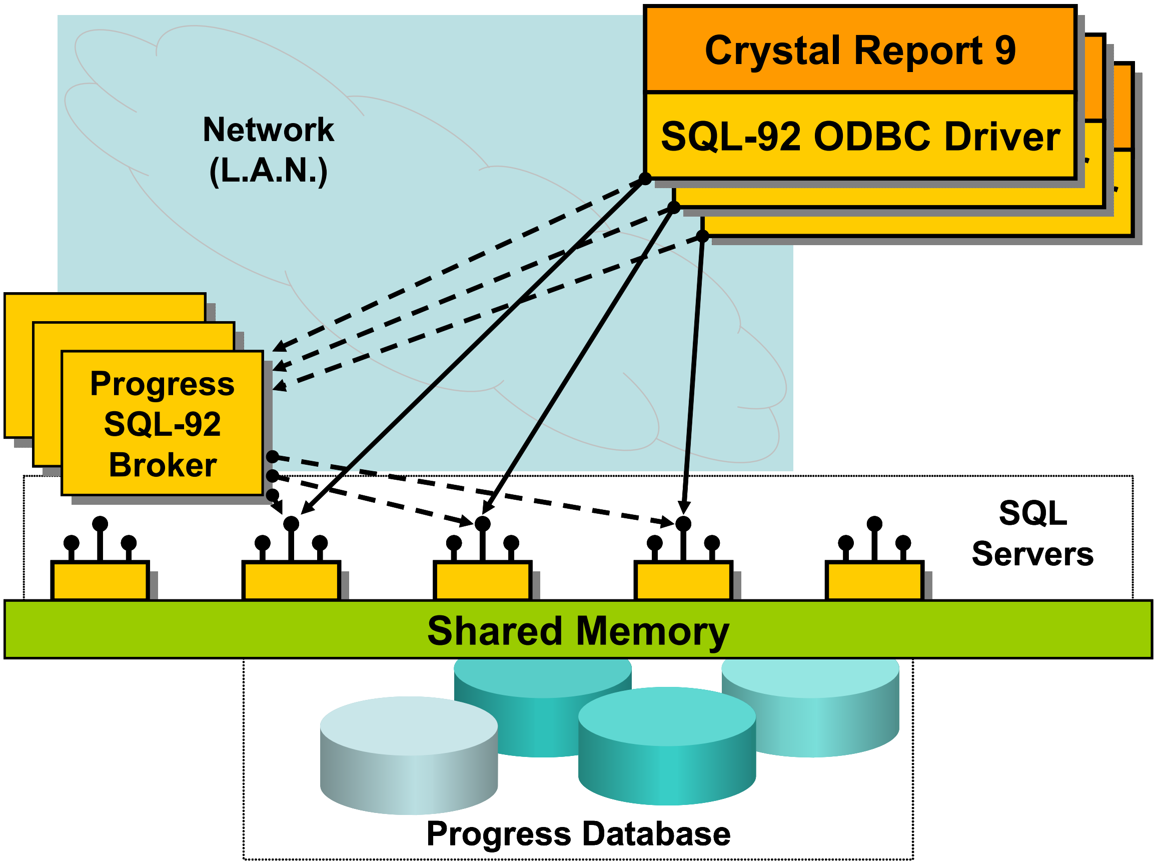
Specific details about this sample setup are provided in each sub-section entitled: "In Practice". They are also summarized at the end of this white paper in Appendix A.

2 Requirements

Before getting into the technical details we need to consider the licenses required to start using Crystal Reports 9 in combination with Progress 9.1D.

2.1 Architecture Overview

The general architecture used with this combination of products is composed of the elements illustrated below:



This illustration highlights the three key components involved.

From Crystal Decisions:

- Crystal Reports 9 (or any other products using an ODBC connection to access a database)

From Progress Software:

- A suitable ODBC driver to access the Progress database
- The Progress SQL-92 server

Each of these components will be covered with more detail in the following sections.

2.2 Crystal Decisions Requirements

Note: For clarity we will only consider in this document Crystal Decisions' Version 9 Product line.

Crystal Decisions provides a series of products among others:

- Crystal Reports editions:
    - Crystal Reports Professional
    - Crystal Reports Developer
    - Crystal Reports Advanced
- Crystal Enterprise Report Development Component (RDC)
- Crystal Enterprise Report Application Server (RAS)
- Crystal Enterprise Professional
- Crystal Enterprise Smart Reporting add-on

Crystal Reports 9 is only available on the Microsoft Windows® 32 operating system suite on Intel® platforms. It is certified and supported on:

- **Windows NT Workstation 4.0 SP6a**
- **Windows NT Server 4.0 SP6a**
- **Windows 2000 Professional SP2**
- **Windows 2000 Server**
- **Windows 2000 Advanced Server**
- **Windows XP Professional**
- **Windows98 Second Edition**
- **Windows ME**

The RAS and RDC products are available on Microsoft Windows 32 operating systems on Intel platforms. They are certified and supported on:

- **Windows NT Server 4.0 SP6a**
- **Windows 2000 Server**
- **Windows 2000 Advanced Server**

Crystal Enterprise Professional and Smart reporting are available on Windows 32 operating systems on Intel platforms, Sun® Solaris™ operating system on 32-bit and 64-bit platforms, and IBM® AIX®. They are certified and supported on:

- **Windows NT Server 4.0 SP6a**
- **Windows 2000 Server**
- **Windows 2000 Advanced Server**
- **Windows .NET Server**
- **Solaris 2.8**
- **AIX 5.1.2**

These products are available from your Progress Account Representative, Partner, or Distributor.

2.3 Progress Requirements

From Progress's point of view there are two products that are required to be installed. These are:

- The Progress SQL_92 ODBC Driver (included in the SQL-92 client access license)
- The Progress SQL-92 Server (included in the Progress Personal, Workgroup or Enterprise RDBMS licenses after performing a **Complete** installation)

Each of the above listed licenses is also included in other (more generic) licenses. For more information on this subject, you can find information about them in the following Progress manuals:

- *Progress Installation and Configuration Guide Version 9 for UNIX* - Appendix D: "Progress Version 9 Typical and Complete Installation Components"
- *Progress Installation and Configuration Guide Version 9 for Windows* - Appendix F: "Progress Version 9 Typical and Complete Installation Components"

2.3.1 The Progress SQL-92 Server

More specific and detailed information about requirements for the Progress SQL-92 server, can be accessed at: http://psdn.progress.com/library/sql.htm.

The white paper named **“Progress SQL92 Server: Installation, Environment, and Administration”** provides detailed information with regards to:

- Operating system availability
- License requirements
- Java™ requirements

However, the “survival kit” is listed below:

Table 1: First availability of the SQL92 server per operating system

Operating System	First Progress Version with the SQL92 Server
Citrix MetaFrame	9.1a
Compaq® Tru64	9.1a
DG/UX Intel	Not Available
HP-UX® 11 (Prgs 32-bit)	9.1a
HP-UX® 11 (Prgs 64-bit)	9.1b
IBM AIX®	9.1a
Linux Intel	9.1b
SCO OpenServer	Not Available
SCO UnixWare®	9.1b
Sun® Solaris™ Intel	Not Available
Sun® Solaris™ Sparc	9.1a
Microsoft Windows® 32 Intel	9.1a

Progress product licenses that will install the Progress SQL-92 server:

- 4GL Development
- Enterprise Database
- Personal Database
- ProVision *
- ProVision Development Server
- ProVision Plus *
- ProVision Plus Development Server
- Visual Translator *
- WebSpeed Development Server
- WebSpeed Workshop *
- Workgroup Database

Notes:

- Only a complete install of the above licenses will install the Progress SQL-92 server
- Licenses marked with * are only available on Windows.

2.3.2 The Progress ODBC Driver

This section lists the information that you need to make sure you have the required ODBC Driver:

- Operating system availability
- License requirements
- Java™ requirements

Table 2: First availability of the SQL92 ODBC driver per operating system

Operating System	First Progress Version with the SQL92 ODBC Driver
Citrix MetaFrame	9.1a
Compaq® Tru64	Not Available
DG/UX Intel	Not Available
HP-UX® 11 (Prgs 32-bit)	9.1a
HP-UX® 11 (Prgs 64-bit)	9.1b
IBM AIX®	9.1a
Linux Intel	9.1c
SCO OpenServer	Not Available
SCO UnixWare®	Not Available
Sun® Solaris™ Intel	Not Available
Sun® Solaris™ Sparc	9.1a
Microsoft Windows® 32 Intel	9.1a

These Progress licenses will install the Progress SQL-92 ODBC Driver:

- 4GL Development
- Client Networking
- Enterprise Database
- Personal Database
- Progress AppServer
- ProVision *
- ProVision Development Server
- ProVision Plus *
- ProVision Plus Development Server
- Secure AppServer
- SQL-92 Client Access #

- WebSpeed Development Server
- WebSpeed Enterprise Transaction Server
- WebSpeed Transaction Server
- WebSpeed Workshop *
- Workgroup Database

Notes:

- Only a **Complete** install of the above licenses installs the Progress SQL-92 ODBC Driver
- Licenses marked with # do not require a complete install to install the Progress SQL-92 ODBC Driver.
- Licenses marked with * are only available on Windows.

2.4 In Practice

Ensure, with the help of the information provided in the previous sections, that you meet:

- The Progress SQL-92 server requirements (Make sure you have the correct licenses and that it is properly installed.)
- The ODBC Driver requirements (Make sure you have the correct licenses and that it is properly installed.)
- The Crystal Reports requirements (Make sure you have the correct licenses and that the products are properly installed.)

Note: At this point in time we cannot guarantee that your connection will work as we have not yet tested anything. We have merely checked that the proper licenses are installed and that the architecture described can be applied to your environment.

3 Starting Your database

The first key element to have your configuration working is to ensure that your database server is up and running.

3.1 General information

There are many ways to start your database server. Detailed information is available at: http://psdn.progress.com/library/sql.htm.

The white paper titled **“Progress SQL92 Server: Installation, Environment, and Administration”** provides detailed information with regards to:

- Configuring your environment for all operating systems where the Progress SQL-92 server is available
- Starting and stopping the SQL-92 database server
- Basic guidelines to avoid server resource contention

To simplify this white paper, we are, from this point on, making the following assumptions:

- The operating system where the database is located is Windows.
- The database you want to use is the Sports2000 database and it is located in the C:\temp directory.
- The database server is started manually from the Progress environment (Proenv shortcut).
- The command line used is: “proserve sports2000 -S 5555” (using all the default values for other parameters).

3.2 In Practice

We have checked that the required licenses are installed (See section 2 if not). So let's move on to the next step.

As stated above, starting a progress database can be achieved in many different ways. However the most reliable way to start your database server in a test environment consists of following these steps:

- **On Windows:** Start Proenv (shortcut installed in the Progress group).
- **On UNIX:** Execute the proenv script located in your $DLC/bin directory.
  1. Change your current directory to the directory containing the database.
  2. Start the database server with the following parameter in your command or .pf file in order to enable networking (SQL-92 connections are always client-server based): "-S <service number>" (Where service number is the port number used by your Progress broker to help establish connections to your Progress database). For this example, we use:
     proserve sports2000 –S 5555

  3. Find the hostname or IP Address of the machine hosting your database server.

Note: For this example to continue, we will assume that the hostname is localhost and that the port on which the broker is listening is referenced by 5555.

4 Configuring Your ODBC Driver

The second key element to have your configuration working is to ensure that your ODBC driver is configured and can access your Progress database through its SQL-92 server.

4.1 General Information

There are different ways to configure your ODBC Connection depending on the operating system from which you will use Crystal Reports. Detailed information is available at: http://psdn.progress.com/library/sql.htm.

The white paper titled "SQL92 ODBC Configuration" provides detailed information in regards of:

- Configuring your environment for all operating systems where the ODBC driver is available
- Configuring a connection to be used from Crystal Reports while developing a report or while executing a report
- Basic guidelines to avoid ODBC resource contention (such as transaction isolation levels, cursors, etc.)

One thing you need to ensure is that your ODBC connection is working fine before working with Crystal Reports. Test it. For this purpose you can use several utilities:

- A freeware or open source utility which may require to be compiled on your operating system that allows testing ODBC configurations
- The Test Connection button available from the ODBC DataSource Administrator on Windows

- Microsoft Query which is a Windows-based utility available from The Microsoft Office suite after performing a complete install (or at least after ensuring this utility got installed)
- Winsql: which is shareware available for download at: http://www.indu-soft.com
- Other utilities that are available as shareware

4.2 Choosing the Proper Transaction Isolation Level

With Progress SQL-92, there are four transaction isolation levels. The isolation level for each ODBC Connection needs to be selected when creating a new ODBC Data Source configuration.

Each isolation level is defined in the following sections as well as the phenomena they allow or prevent while accessing data stored in your database. Since each isolation level is defined based on the different phenomena that can occur on your database, we will first define them.

4.2.1 SQL-92 Phenomena: Definitions

As stated above, each isolation level either allows or prevents certain phenomena. Let’s define these phenomena.

- **Dirty read**

This phenomenon occurs when a client session reads a row that has been inserted or modified by another transaction, but not committed. If the other transaction rolls back its changes, the transaction will read a row that never existed because it never committed.

- **Non repeatable read**

This phenomenon occurs when a client session reads rows from a set of tables before the rows accessed are modified or deleted by another transaction. It then reads again rows from the same set of tables and retrieves the modified rows after the other transaction commits its changes.

- **Phantom**

This phenomenon occurs when a client session reads a range of rows that satisfies a given search condition from a set of tables before new rows matching the same search condition are added by another transaction. Reading again this same range of rows retrieves the new rows after the other transaction has committed its changes.

4.2.2 Transaction Isolation Levels: Definitions

In the Progress implementation each isolation level is based on a specific locking scheme whose purpose is to either allow or prevent the different phenomena to occur or not while accessing your data.

- **READ UNCOMMITTED**

This isolation level allows “dirty reads”, “non repeatable reads”, and “phantoms”. It also allows a user application to read records that were modified by other applications but have not yet been committed.

Record Locking scheme: It ensures that when a record is read, no record locks are acquired. However it does require a shared schema lock to ensure that the format of the data does not change while reading records.

- **READ COMMITTED**

This isolation level prohibits “dirty reads”; allows “non repeatable reads” and “phantoms”. It disallows the reading of uncommitted/modified records. However, if a record is read multiple times in the same transaction, the state of the record can be different each time.

Record Locking Scheme: It ensures that when a record is read, a share lock is acquired on that record; the lock is held on the record for the time required to read that record. It also requires a shared schema lock to ensure that the format of the data does not change while reading records.

- **REPEATABLE READ**

This isolation level prohibits “dirty reads” and “non repeatable reads”; it allows “phantoms”. It disallows the reading of uncommitted modified records. If a record is read multiple times in the same transaction, the state of the record remains the same.

Record Locking Scheme: It ensures that when a record is read, a share lock is acquired on that record and held until the end of the current transaction (up to a commit transaction point or up to the connection is properly closed). It also requires a shared schema lock to ensure that the format of the data does not change while reading records.

- **SERIALIZABLE**

This isolation level prohibits “dirty reads”, “non repeatable reads”, and “phantoms”. If an application executes the same SELECT statement more than once within the same transaction, the same set of rows is retrieved every time. It guarantees that the concurrent transactions will not affect each other, and that they will behave as if they were executing serially, not concurrently.

Record Locking Scheme: It ensures that when a table is accessed the entire table is locked with a lock of appropriate strength; the lock is held until the end of the transaction.

4.2.3 Choosing the Correct Isolation Level

Different isolation level mean: Different levels of data consistency. Table 3 summarizes this briefly.

Table 3: Isolation level and phenomena

	Dirty Read	Non repeatable Read	Phantom Read
Uncommitted Read	Permitted	Permitted	Permitted
Committed Read	Prevented	Permitted	Permitted
Repeatable Read	Prevented	Prevented	Permitted
Serializable	Prevented	Prevented	Prevented

There are two rules to keep in mind when choosing the proper transaction isolation level for your ODBC connections:

- **Higher isolation level means:**

  - Better data consistency (more lock constraints)
  - Lower performance
- **Lower isolation level means:**
  - Lower data consistency (fewer lock constraints)
  - Better performance

Based on this information we can provide basic guidelines for choosing the proper isolation level for the ODBC connection that is going to be used by Crystal Reports:

- Un-committed read should be used with reports that do not rely on data accuracy. Usually these same reports also process/access a high number of records. This allows getting best performances while executing your report with a minimum number of locks on your database. Examples of reports in this category:
  - Statistical information at the end of a month
  - Sales report covering the previous year(s)
  - Reports running on a Data Mart (database whose only purpose to be used with reporting applications – this database is created/updated on a regular basis in order to keep it accurate)
  - Reports running daily that accesses data hardly or never updated (archived or reference tables)
  - Reports running daily if the values displayed are only used as indicators (meaning that the values displayed do not require to be 100% accurate)
- Committed read should be used with reports running daily on data that is frequently modified. This allows getting good performances while executing your report on a 'live' database with an average number of record locks that are released immediately. Examples of reports in this category:
  - Reports running daily when the accessed data is regularly updated and that require 100% data accuracy at the time the record is processed (inventory reports for items in store)
  - Reports building snapshots of operations for monitoring purposes at any given time during the day (new values can be entered, existing values can be modified or even deleted from the database – i.e. stock exchange status report at specific times during the day)
- Repeatable read should not be used with reports as it does not add any value at the time the report is generated (especially when compared to Committed Read).
- Serializable should not be used with reports as it does not add any value at the time the report is generated (especially when compared to Committed Read).

4.3 In Practice

We have checked or performed the following:

- The required licenses are installed (See section 2 if not)
- Started a Progress database broker listening on port "5555" on a machine which has hostname "localhost"

Since Crystal Reports 9 and associated products are currently available on Windows only, we are not providing an ODBC driver configuration example on UNIX.

The first thing to test at this stage is making sure that an ODBC or JDBC connection to your Progress SQL-92 server can be established from the machine where your ODBC and JDBC drivers are installed. For this, please follow these steps:

1. Start the SQL Explorer (use the shortcut available in the Progress group under windows)
2. Connect the database using the following settings:
    - Host: Localhost
    - Service or Port: 5555
    - Database: sports2000
    - User: test
    - Password: test
3. There are cases at this stage:
    - Your database server does not return an error message (login validation is not enabled). As a proof you can execute the following statement:
        - SELECT * FROM pub.state
        - The above query returns the data contained the state table located in the pub schema (the schema where all 4GL tables are located).
    - Your database server returns an error message: "Access denied (Authorisation failed)" (login validation is enabled).

Note: in both cases you have successfully proven that you can establish the proper connection to your database. However, for the next steps to be successful we need to find the SQL-92 DBA account to access this database. (If you know it, please use this information above so that you can execute the SELECT statement successfully.)

If you have successfully established a connection but do not know the SQL-92 DBA account to use, or if you got the "Access denied (Authorisation failed)" error message, you need to find out who the SQL-92 DBA is on your database: There are three options:

- **In all cases, try first to:**
    1. Look in the database log file for the following message: "procopy get default SQL DBA session begin for **Administrator** on CON:. (451)". In this message **Administrator** is referenced as the default SQL-92 DBA for the progress SQL-92 database.
    2. In case you cannot find this message, please refer to the next bullets.
- **If login validation is not enabled:**
    1. Connect your SQL-92 database using the Progress SQL Explorer and execute the following query:

        SELECT * FROM sysprogress.sysdbauth WHERE dba_acc = 'y' AND GRANTEE <> 'sysprogress'
    2. The result returned should show at least 1 row. One of the values listed in the "Grantee" column is a proper SQL-92 DBA account that you should/can use from this point on.
    3. If the list displayed contains only 'sysprogress' then you need to create a proper DBA. Please refer to Appendix B.
- **If Login validation is enabled:**
    1. Find a valid User ID / Password that is currently defined in your database (use 4GL administration utility for this), or create a new one.
    2. Connect your SQL-92 database using the Progress SQL Explorer and execute the following query:

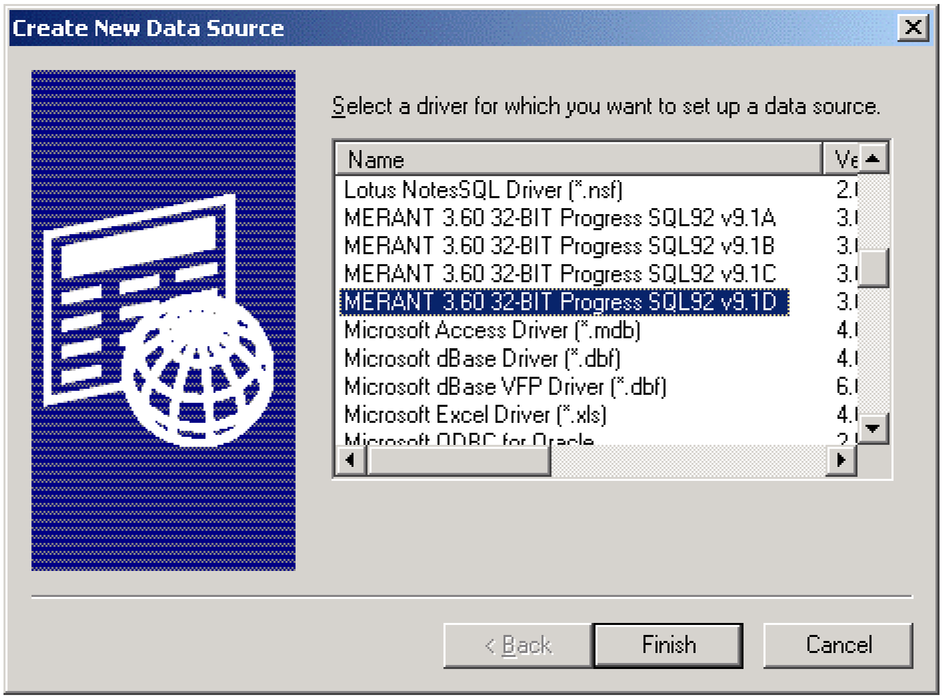
        SELECT * FROM sysprogress.sysdbauth WHERE dba_acc = 'y' AND GRANTEE <> 'sysprogress'
    3. The result returned should show at least 1 row. One of the values listed in the "Grantee" column is a proper SQL-92 DBA account that you should/can use from this point on.

4. If the list displayed contains only ‘sysprogress’ then you need to create a proper DBA. Please refer to Appendix B.

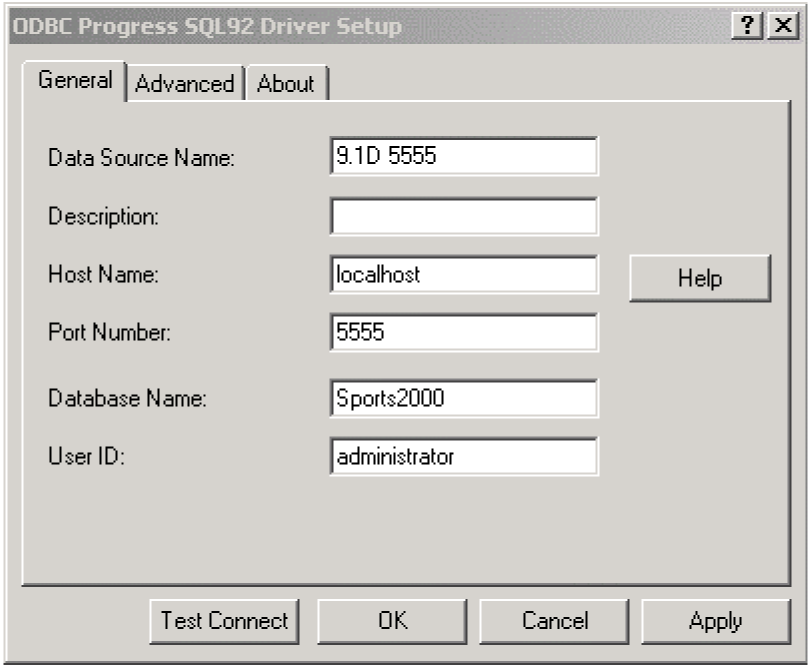
Note: Once you have found the SQL-92 DBA account to use, please make a note of it and keep it for future use.

Let’s move on to the next step: configuring your ODBC driver. With the information gathered during the previous stages, we can now create an ODBC configuration. Please follow these steps:

1. Start your “Data Sources (ODBC)” utility (on Windows XP it is located in your Control Panel under the ‘Administrative tools’).
2. Add a new User or System Data Source Name (DSN).
3. Select the “Progress SQL-92 v9.1D” ODBC driver as shown below:


Create New Data Source
Select a driver for which you want to set up a data source.
Name
Ve
Lotus NotesSQL Driver (*.nsf)
MERANT 3.60 32-BIT Progress SQL92 v9.1A
MERANT 3.60 32-BIT Progress SQL92 v9.1B
MERANT 3.60 32-BIT Progress SQL92 v9.1C
MERANT 3.60 32-BIT Progress SQL92 v9.1D
Microsoft Access Driver (*.mdb)
Microsoft dBase Driver (*.dbf)
Microsoft dBase VFP Driver (*.dbf)
Microsoft Excel Driver (*.xls)
< Back
Finish
Cancel

4. Click Finish on this dialog box.
5. On the next screen, in the “General” tab, enter the DSN Name: 9.1D 5555.
6. Enter the Host name (localhost), Port Number (5555), Database Name (Sports2000) and User ID (administrator) based on the information retrieved previously. The “General” tab should look like this:



Data Source Name: 9.1D 5555

Description:

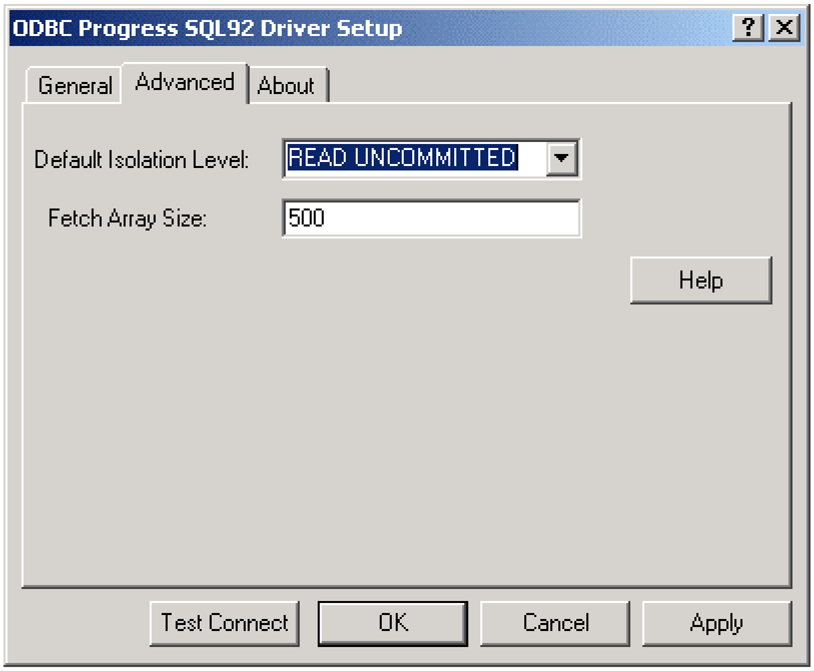
Host Name: localhost

Port Number: 5555

Database Name: Sports2000

User ID: administrator

7. In the “Advanced” tab, Select “Read Uncommitted” as the Default Isolation Level.
8. Enter a Fetch Array Size of 500 (default is 50). Your screen should now look like this:



Default Isolation Level: READ UNCOMMITTED

Fetch Array Size: 500

9. Click Apply.
10. Click Test Connect. (If you do it the other way around you may loose the entered settings.) A dialog box should pop up asking for the password. Please enter the proper password if “Login validation” is enabled or leave blank otherwise.

Notes:

- Setting the Default Isolation Level to Read Committed prevents dirty read. This enables better data accuracy; however it also requires setting locks on the different records accessed within the database. In this example, we assume that we do not require proper data accuracy while processing the report.
- The information that Crystal Reports 9 really needs for the next steps is:
    - The ODBC Data Source Name
    - The User ID used when connecting the Progress SQL-92 database. In our case we are using a DBA User ID
    - The Password associated with the User ID

From this point on we can start working with Crystal Reports 9.

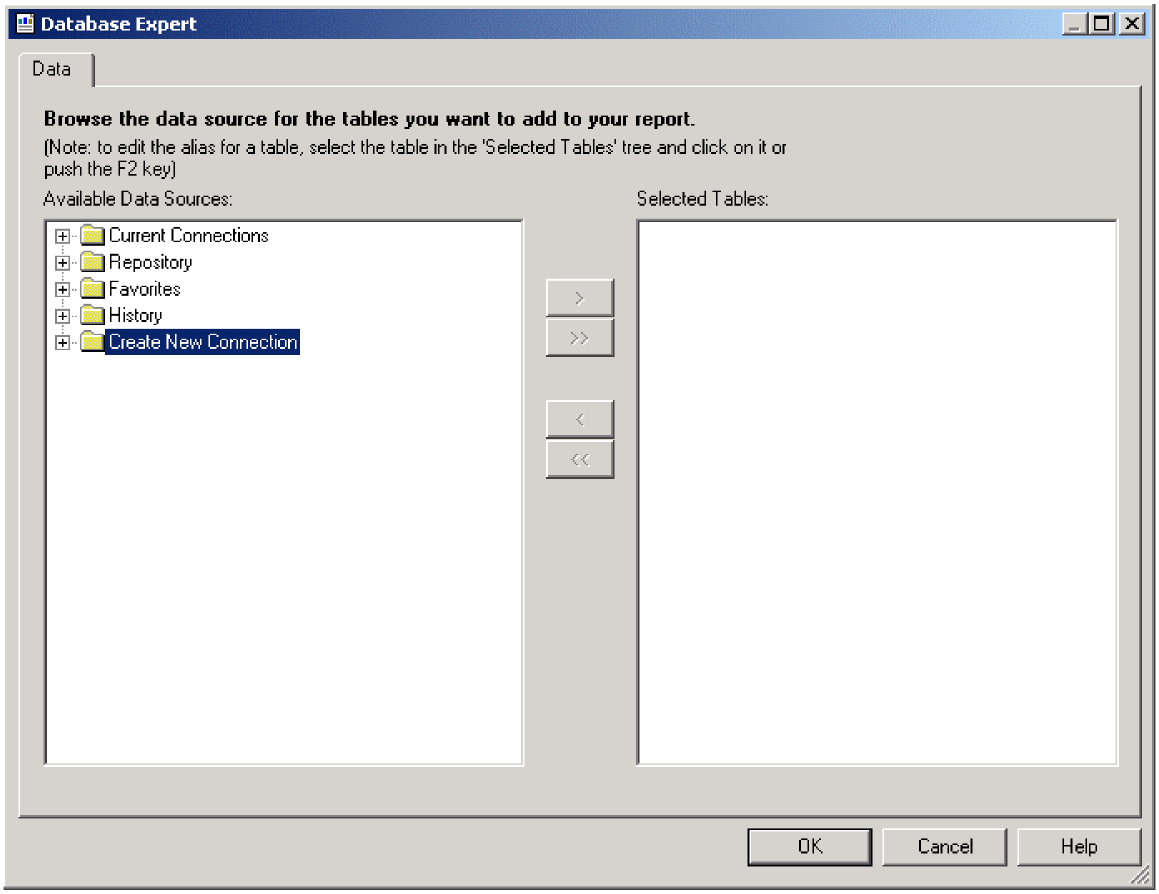
5 Connecting a Progress Database when Creating a New Report

The third and last key element to have your configuration working is to ensure that Crystal Reports 9 can use your newly created ODBC configuration and therefore access your Progress database through its SQL-92 server.

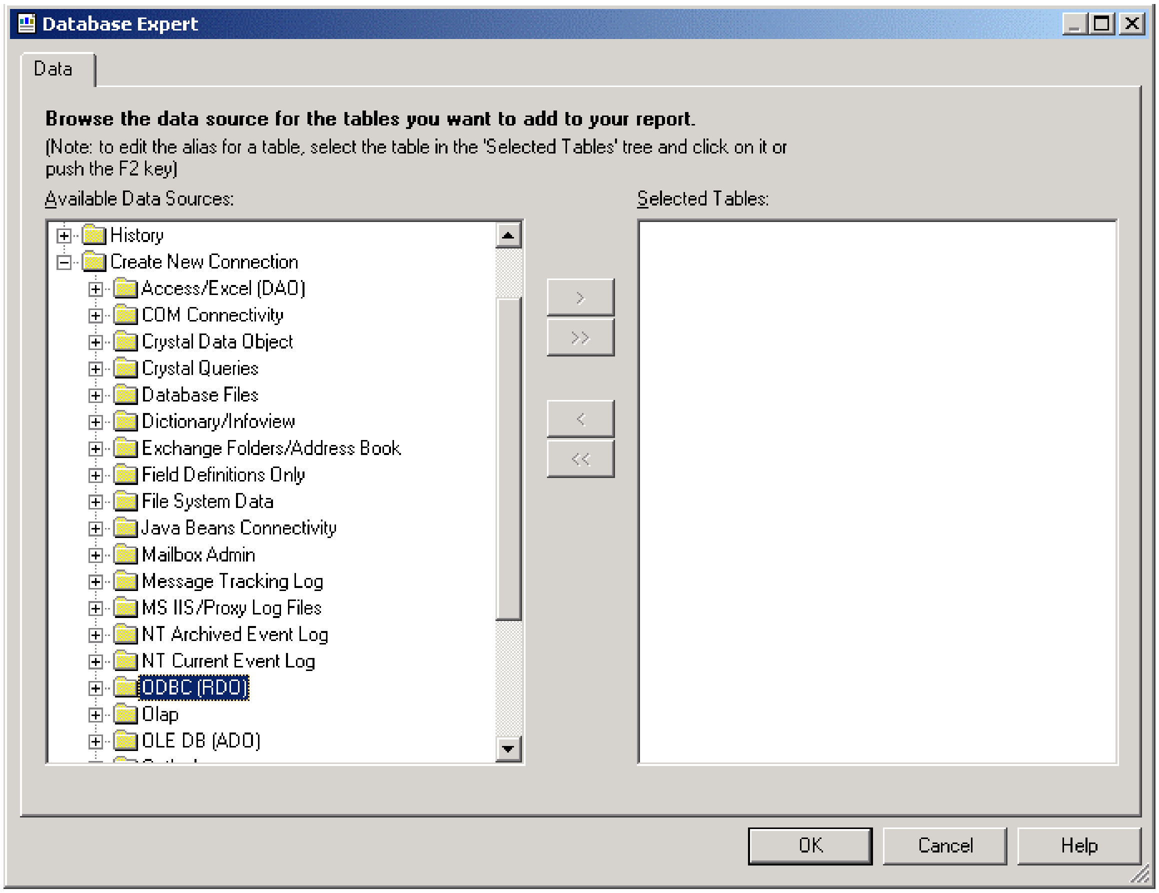
When starting to use Crystal Reports 9 to design a new report, the first required step is connecting a Progress SQL-92 database. Crystal Reports 9 provides easy access to ODBC Data Sources while creating a new report. These steps are provided in the next section.

5.1 In Practice

1. Start Crystal Reports.
2. Create a New Crystal Reports Document As a Blank Report (“File” Menu – “New” menu item). You will get the following dialog box:


Database Expert
Data
Browse the data source for the tables you want to add to your report.
(Note: to edit the alias for a table, select the table in the 'Selected Tables' tree and click on it or push the F2 key)
Available Data Sources:
Current Connections
Repository
Favorites
History
Create New Connection
Selected Tables:
>
>>
<
<<
OK
Cancel
Help

3. Under "Available Data Sources", expand "Create New Connection" and double-click ODBC (RDO).



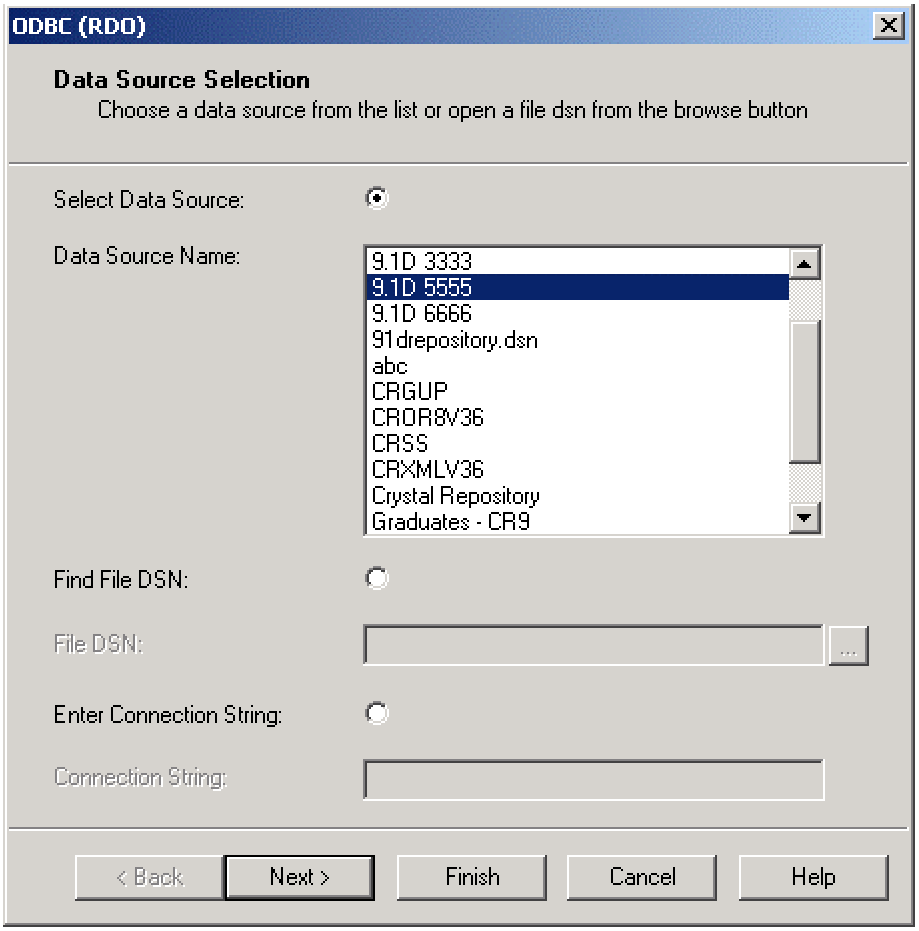
Browse the data source for the tables you want to add to your report.
(Note: to edit the alias for a table, select the table in the 'Selected Tables' tree and click on it or push the F2 key)

Available Data Sources:

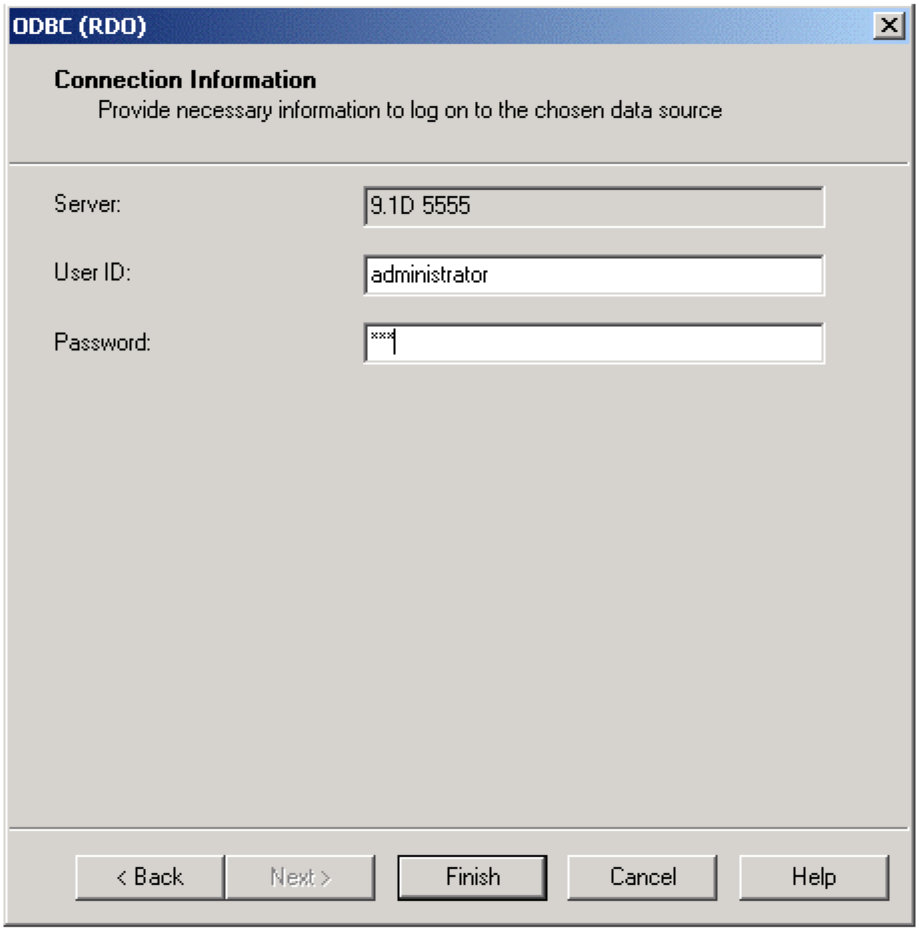
- History
- Create New Connection
  - Access/Excel (DAO)
  - COM Connectivity
  - Crystal Data Object
  - Crystal Queries
  - Database Files
  - Dictionary/Infoview
  - Exchange Folders/Address Book
  - Field Definitions Only
  - File System Data
  - Java Beans Connectivity
  - Mailbox Admin
  - Message Tracking Log
  - MS IIS/Proxy Log Files
  - NT Archived Event Log
  - NT Current Event Log
  - ODBC (RDO)
  - Olap
  - OLE DB (ADO)

Selected Tables:

You should get the following dialog box:



4. Select the Data Source we created in the previous steps (“9.1D 5555”) and click “Next”. You should get this dialog box:


ODBC (RDO)
Connection Information
Provide necessary information to log on to the chosen data source
< Back
Next >
Finish
Cancel
Help

Server: 9.1D 5555

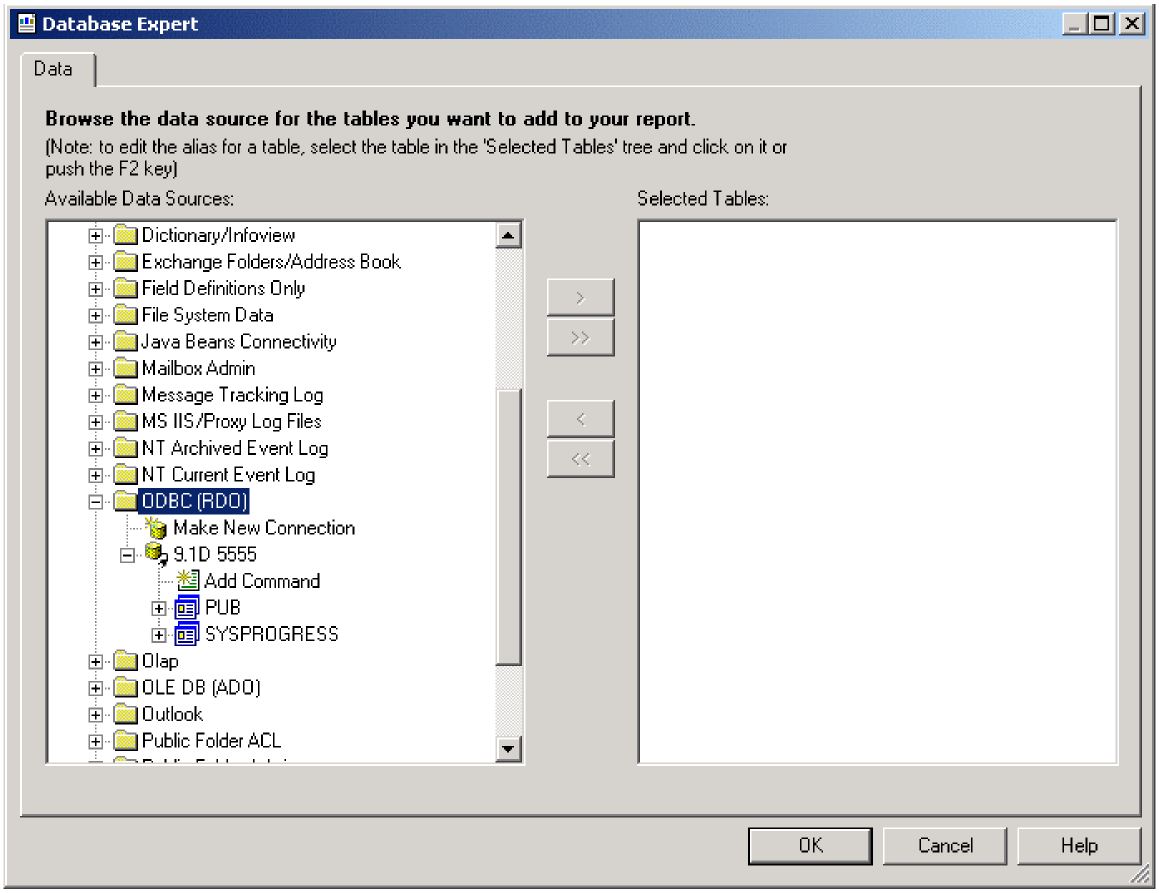
User ID: administrator

Password: ***

5. Provide your DBA User ID and Password if login validation is enabled. Type your DBA User ID only in any other case. Then click “Finish”.

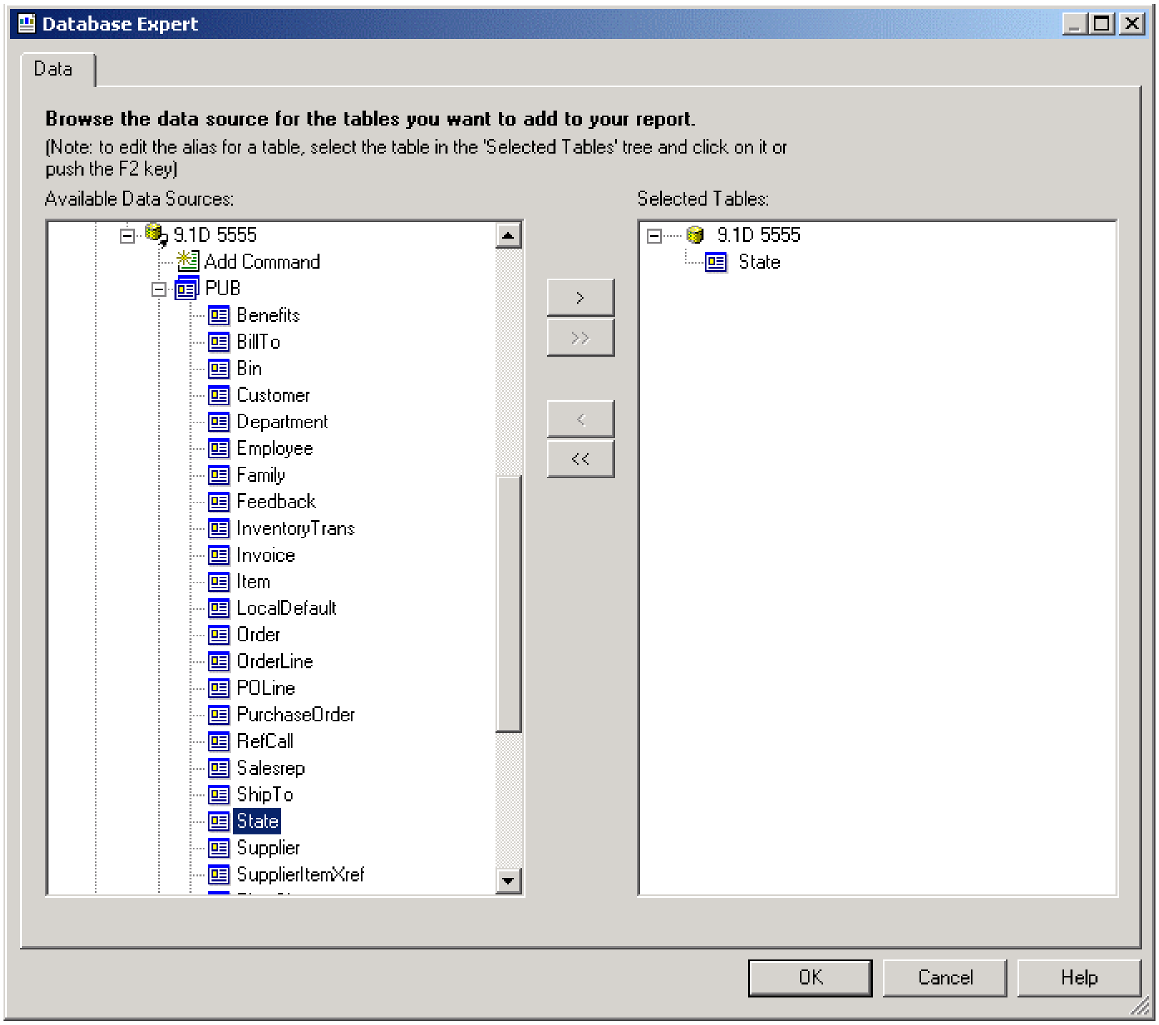
Note: At this stage you have connected Crystal Reports 9 to your Progress database. To demonstrate that we are indeed connected, we are now going to develop a very simple report on the State table located in the “pub” schema.

You should now be back to this screen:



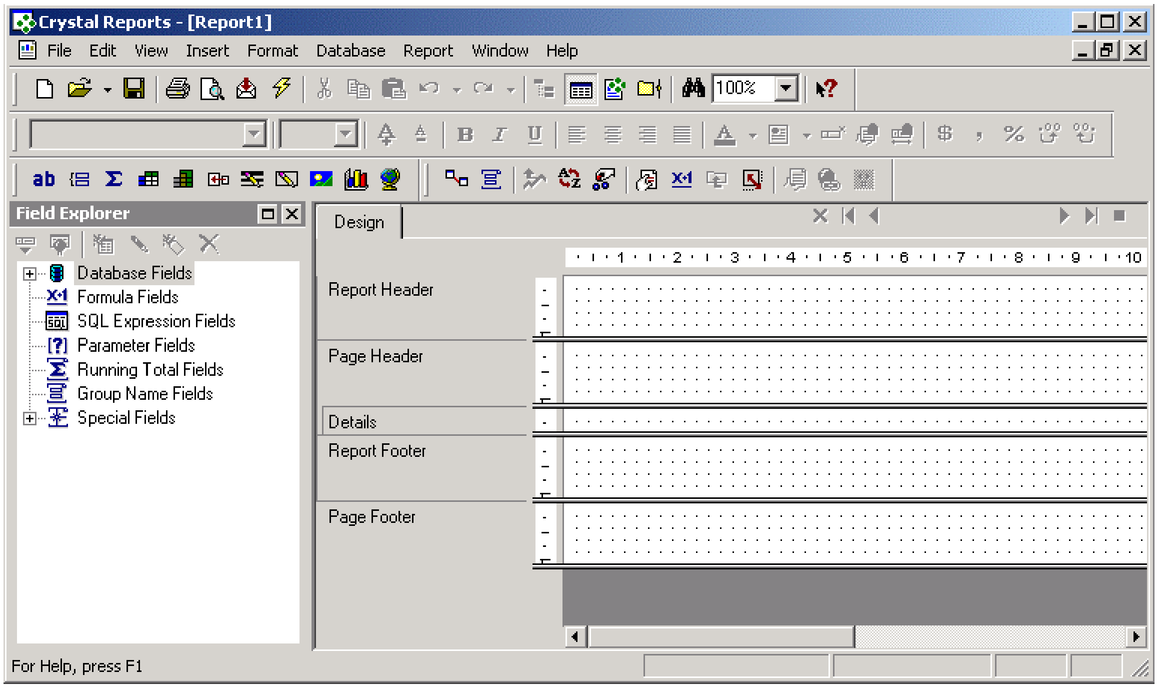
Browse the data source for the tables you want to add to your report.
(Note: to edit the alias for a table, select the table in the 'Selected Tables' tree and click on it or push the F2 key)

1. Under "9.1D 5555" expend the PUB section to retrieve a list of tables located in the pub schema. Select the State table so that it can be used in our report.

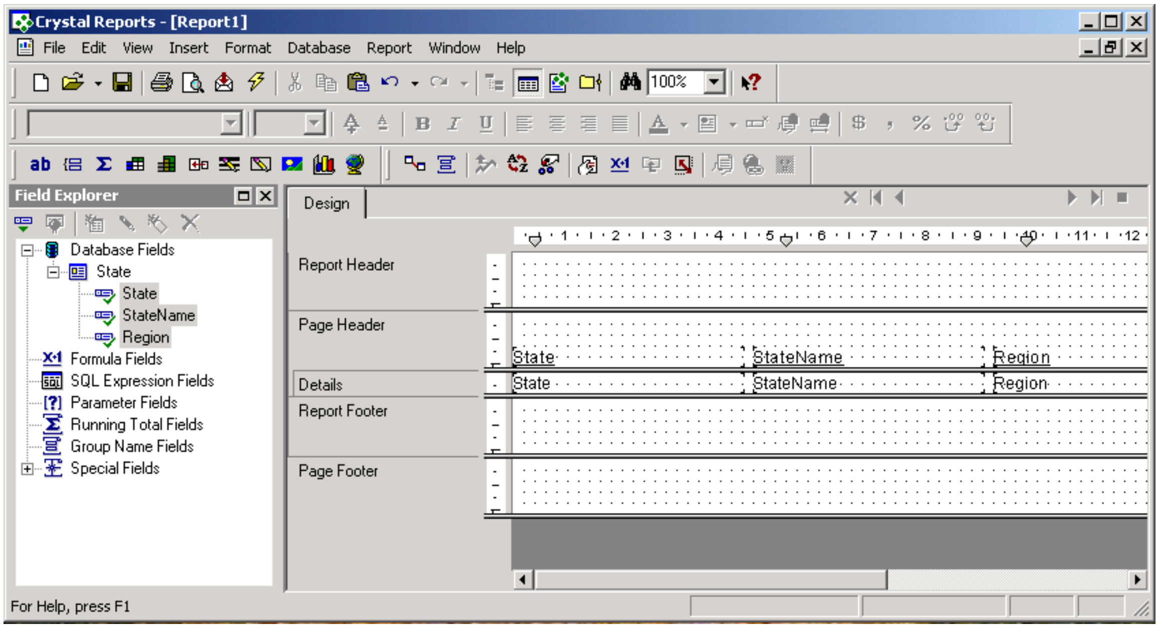


Browse the data source for the tables you want to add to your report.
(Note: to edit the alias for a table, select the table in the 'Selected Tables' tree and click on it or push the F2 key)

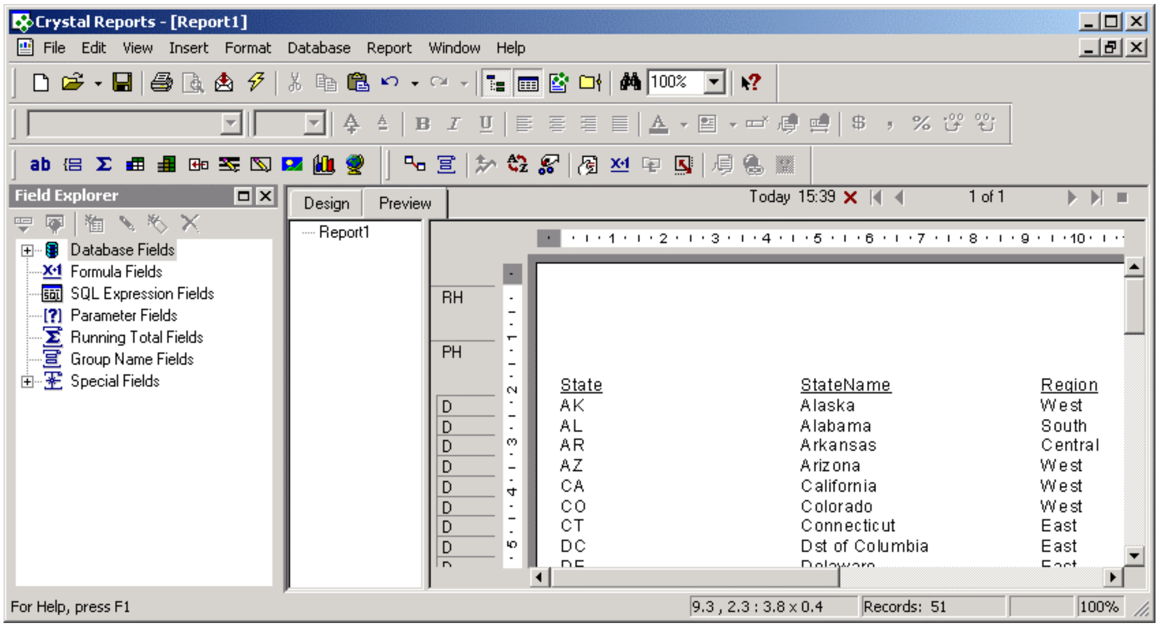
2. Click OK to return to the Crystal Reports main screen.
3. In the “View” menu, select the “Field Explorer” menu item (if not already done). Crystal Reports main window should now look like this:



4. Expand the Database Fields Node and select the three fields available in the State table.
5. Place each field in the Details sections under the Report Design tab (right side of your window). Your window may look like this:



6. Save your report
7. Run your report by pressing F5 on your keyboard. You should get the following on your screen:



State	StateName	Region
AK	Alaska	West
AL	Alabama	South
AR	Arkansas	Central
AZ	Arizona	West
CA	California	West
CO	Colorado	West
CT	Connecticut	East
DC	Dst of Columbia	East

5.2 *Conclusion of the Step-by-Step Example*

If you reach this result or a similar result, you have successfully:

- Configured and started your Progress SQL-92 database server
- Configured your ODBC Data Source connection
- Used Crystal Reports 9 to connect a Progress database through the SQL-92 server
- Developed a basic report demonstrating that your connection works fine

6 Securing Data Access at Design Time

While designing reports we also need to consider data security issues in relation to different User IDs.

Securing data access is required because:

- Usually, there is information stored in your database that should only be available to specific users because of their function in the company where the report is used
- Crystal Reports 9 offers the ability to save data along with the report definition. In other words, at run time, a user can simply visualize the data stored in the report without actually accessing the database.

For these reasons it is wise to consider implementing data security either in the Progress RDBMS and/or in the report itself.

With the architecture described in the previous sections, there are three places where data access can/should be secured. These rely on different functionality available either with Progress SQL-92 or Crystal Reports 9. The three places are:

- At connection time with login validation enabled on the Progress database
- At table-access time by providing the proper privileges to a specific user in the Progress database
- At report-execution time when visualizing a report in Crystal Reports 9

Note: Before providing more details on these three places where we can secure data access, we need to remember two key rules with Progress SQL-92:

- By default, with any SQL-92 RDBMS, everything is prevented or forbidden. It is the role of a DBA to give sufficient privileges to a user so that he can access the adequate data in the database. If this is not properly defined in the database, the user accessing the database or data in your database should get the following error message: "Access denied (Authorisation failed) (7512)".
- In the Progress SQL-92 database User ID and privileges are independent (which is not common practice). Some of the key consequences are:
    - You can access a Progress database through its SQL-92 interface without enabling login validation. However the Progress SQL-92 interface requires a User ID at connection time.
    - A DBA can grant or revoke privileges to specific User ID (even if they are not previously created).

6.1 Through Login Validation

Before providing guidelines to implement login validation on a Progress RDBMS, there are **important** considerations we need to establish if you are implementing or have implemented this feature.

6.1.1 Key Mechanisms

First and foremost, login validation is defined in the following terms: It is the mechanism that checks the User ID and password at connection time with reference data stored in the RDBMS.

In the Progress RDBMS this information is stored in the pub."_user" table (from the SQL-92 interface) or simply _user (from the Progress 4GL interface). The same reference table is used by both interfaces. This means that enabling login validation on the database affects the Progress 4GL as well as the Progress SQL-92 interface.

The Progress SQL-92 interface does not permit "Blank user" connections. In other words, at connection time a User ID must be entered (even if login validation is not enabled). This is because SQL-92 privileges will be checked based on this same User ID. (Next section provides more details on this point.)

To enable login validation a DBA only needs to create a first user via:

- The Progress SQL-92 interface (Create users statement)
- The Progress Data Admin tool (Progress 4GL interface)

To disable login validation a DBA only needs to drop/delete all users via:

- The Progress SQL-92 interface (Drop users statement)
- The Progress Data Admin tool (Progress 4GL interface)

6.1.2 Key consequences

Creating/altering/dropping a user via SQL is equivalent to creating/maintaining/deleting a user from the Progress data administration utility (Progress 4GL interface):

- The user list maintained via the Progress SQL-92 interface (Create / alter / drop users) is updated for the Progress 4GL interface as well
- The user list maintained via the Progress data administration utility (Progress 4GL interface) is updated for the Progress SQL-92 interface as well

Considering an environment where a Progress RDBMS is accessed by a 4GL application and an SQL-92 application, the following should be noticed:
When no users are created:

- **4GL will not display a login dialog box.**
- **SQL will ALWAYS require a user ID** (password can be left blank). The User ID can be anything as there is no reference value.

When users are created:

- **4GL may display a login dialogue-box** if this is not handled in the application itself. In most cases, the 4GL interface also allows blank connections if not disabled by the DBA.
- **SQL ALWAYS requires a VALID login.**

6.1.3 Using Login Validation: Basic Guidelines

Now the question that should come to your mind is: "Should I use login validation?"

To answer this question you need to consider your run-time environment.

- If your Progress RDBMS is accessed only through the Progress SQL-92 interface: Yes you should enable login validation. Creating User IDs and Password will ensure that:
  - Proper users access the database (validated with the Password).
  - There is no typo in the User ID entered so that proper privileges are used when accessing data through your RDBMS. (See the next section for more information.)
- If your Progress RDBMS is accessed through both the Progress SQL-92 and 4GL interfaces and the 4GL already does not use login validation then you should not enable login validation unless:
  - You intend to modify your 4GL application and make full use of this feature.
  - Or your users do not mind getting a login dialog box when accessing the Progress RDBMS from this 4GL application.
- If your Progress RDBMS is accessed through the Progress SQL-92 and 4GL interfaces and the 4GL already uses login validation, then you have no choices left.

Notes:

- Login validation is not the only way to secure access to your Progress RDBMS. Enabling it is therefore not a requirement; it just makes it easier for a DBA to solve security-related issues and maintain privileges in your database.
- Login validation does not solve the whole problem as it does not offer the possibility to control access to specific tables in the database (This can only be achieved when setting SQL-92 privileges). However it offers security when connecting the database.
- When login validation is enabled, a user may get the following error message at connection time through the Progress SQL-92 interface: "Access denied (Authorisation failed) (7512)"

Implementing login validation on your database offers only part of a solution when securing access to data stored in your database. It is wise to use in your context (if you can) as it complements the security mechanism used when accessing data (see next section).

6.2 Through SQL-92 Privileges

The second place where data access can be secured is through the use of Progress SQL-92 privileges because they are checked each time an SQL request is processed. SQL92 privileges are different from 4GL privileges. They are defined only through the Progress SQL-92 interface when using the Grant and Revoke statements.

6.2.1 What Are SQL-92 Privileges?

Without having proper privileges granted, a user will get the following error message: “Access denied (Authorisation failed) (7512)” when attempting to access specific data.

This error message can be interpreted in the following ways:
- “A DBA did not Grant sufficient privileges to the user accessing specific data”
- The user has insufficient privileges to perform the action he requested on the database

This is the default behavior for newly created users accessing a database or for any SQL-92 user accessing a newly created database. The proper explanation of this behavior is clarified in any good SQL-92 manual and can be summarized in the following terms: “With SQL-92, every action is prevented unless otherwise specified by a DBA”.

6.2.2 Which Privileges Should a DBA Maintain?

In the context of this document, the only privileges that have some influence are defined for tables, views, and stored procedures.

For users to have access to tables and views, they needs “Select” privileges on each of them. Granting “Select” privileges gives full access to the table or view it is associated to.

For a user to have access to Progress SQL-92 stored procedures, he needs “Execute” privileges on each stored procedures.

These privileges can be granted to each user (or to PUBLIC) by an SQL-92 DBA or the owner/creator of these objects. They are created and suppressed with the SQL-92 GRANT and REVOKE statements (Information on these 2 statements is available in the Progress SQL-92 guide and reference manual)

All these privileges are stored in the sysprogress.systabauth table. In regards of this white paper’s context, a DBA would only need to use the following query to check which privileges are currently defined for users accessing non schema related tables:

```
SELECT grantee, tblowner, tbl, sel, exe
FROM sysprogress.systabauth
WHERE tblowner <> 'sysprogress'
AND NOT(tbl LIKE '\_%' ESCAPE '\');
```

Note: There is no need to create users to give them privileges with Progress SQL-92 because you may not want to use login validation while connecting a Progress database through SQL-92

6.2.3 Some Guidelines

Providing some guidelines is difficult in this context. However a DBA should ask himself the following questions when defining and later on maintaining privileges on tables, views or stored procedures:

- Is this table or view selectable by everybody? (Should its access be restricted to some users only?)
    - If yes then execute this type of statement: GRANT SELECT ON <Schema>.<Table Name> TO PUBLIC
    - If not then execute this type of statement: GRANT SELECT ON <Schema>.<Table Name> TO UserID (repeat this statement for each user after replacing UserID with the proper value)
- Is this Stored Procedure executable by everybody?
    - If yes then execute this type of statement: GRANT EXECUTE ON <Schema>.<Stored Procedure Name> TO PUBLIC
    - If not then execute this type of statement: GRANT EXECUTE ON <Schema>.<Stored Procedure Name> TO UserID (repeat this statement for each user after replacing UserID with the proper value)

Note: This is the proper way to control who can access tables, views or stored procedures in your database.

6.3 Through Password Validation in a Report

The last place where data access can be secured is through the Crystal Reports ability to require a password when visualizing a report on your screen. Using this mechanism requires a good understanding and practice of report development with Crystal Report.

6.3.1 Using Password Protection in Your Reports

Using password protection means implementing a mechanism in reports that display or even access sensitive data. This needs to be repeated for each report falling in this category.

This feature should be implemented with reports developed under Crystal Reports that contain sensitive information because:

- Crystal Reports can save data along with the report definition. If a the developer creating a new report is not careful enough, he can save confidential data with the report offering to any user the ability to visualize data saved with the report instead of re-processing it against your database.
- It offers a place to restrict access to the data visualized to specific users

However this protection mechanism comes with the following restrictions:

- It is independent from OS validation.
- It is independent from database login validation.

Implementing this mechanism needs to be considered when developing your reports and even better when writing the specifications of the reports you are about to develop.

For each report developed the same question needs to be answered: “Is this report displaying sensitive information?” If the answer is “yes”, then protecting your reports with a password validation should be considered (although not required if no data is saved with the report itself).

6.3.2 How to Implement a Password in Your Report

This is roughly based on four steps that need to be repeated for each report that contains data to which you need to secure access.

The steps are:

1. Open your report in Crystal Reports 9.
2. Create a new parameter named Password. (The prompting text could be "Enter your password to view restricted information". This parameter may be a string but requires to have the edit mask set to be password (meaning that characters are not displayed when the user is typing the password).
3. For the sections in your report displaying restricted information, change its properties so that it is conditionally suppressed if they do not enter the proper password value (the value itself is stored in the report and not in the database).
1. Save your report under a new name.

To see an example, you can:

1. Download the Crystal Reports Sample (zip file available from http://www.psdn.com/library/sql.htm).
2. Create on your machine a sports2000 database.
3. Start your database server as described in this document.
4. Configure an ODBC Data Source named "9.1D 5555" as described in this document.
5. Start Crystal Reports and open the report named: "Password Report.rpt" and execute it.
6. If you wish to see all the data, type "crystal" as the password.
7. If you wish to see only the data with the first letter outside of Range A-F type anything else.

The report should display a list of customers grouped by country. Depending on the password entered it may display the complete list of customers.

6.4 Conclusion

From the three places where mechanisms to secure access to sensitive data can be implemented, the most important one is through SQL-92 privileges. It defines exactly what a user can or cannot do with the data stored in your database.

Using Password validation, when visualizing a report, is good practice in general. However, it may have consequences because users who should have access to restricted information need to remember an additional password.

Implementing login validation on your database offers only part of a solution when securing access to data stored in your database. Although not required, it is wise to implement this mechanism in order to simplify the privilege management tasks a DBA must perform.

7 Developing a Report with Crystal 9 and Progress

We will review in this section some key areas while developing reports against a Progress database. These areas are listed here because they raised frequently asked questions or because they provide useful indications on what is possible when using Crystal Reports 9 with a Progress database.

7.1 Accessing Progress Extents from Crystal Reports 9

As Progress 4GL extents are not part of the SQL-92 standard, Progress provides a way to access these database columns and retrieve the information they contain in a readable fashion.

In the context of this white paper there are two recommended ways to access Progress extents and format them in the way you prefer. Each solution relies on features available through the Progress SQL-92 interface or within Crystal Reports 9. Each of these two techniques has its one advantage or raises its own issues and concerns. However both of these techniques are available to use with Crystal Reports 9 and Progress SQL-92

7.1.1 Using SQL-92 Views

On the Progress SQL-92 side the easiest way to help split the extents in different columns is to create a view.

Because this view is used only for reporting purpose we are only interested in splitting the values into a readable format where each value is in its own column.

This should be achieved through the use of the Pro_element function provided in Progress SQL-92. For detailed information on this function, please check the *Progress SQL-92 Guide and Reference* manual.

Here is an example to be used with the Progress Sports2000 database. You need first to create in your database the proper view (After you have connected your database server using the DBA User ID). The view definition is provided in the script below:

```
DROP VIEW mysalesrep;
CREATE VIEW mysalesrep (
      RepName,
      Region,
      SalesRep,
      January,
      February,
      March,
      April,
      May,
      June,
      July,
      August,
      September,
      October,
      November,
      December)
AS SELECT RepName,
      Region,
      SalesRep,
      To_number(Pro_element(MonthQuota,1,1)) 'January',
      To_number(Pro_element(MonthQuota,2,2)) 'February',
      To_number(Pro_element(MonthQuota,3,3)) 'March',
      To_number(Pro_element(MonthQuota,4,4)) 'April',
      To_number(Pro_element(MonthQuota,5,5)) 'May',
      To_number(Pro_element(MonthQuota,6,6)) 'June',
      To_number(Pro_element(MonthQuota,7,7)) 'July',
      To_number(Pro_element(MonthQuota,8,8)) 'August',
      To_number(Pro_element(MonthQuota,9,9)) 'September',
      To_number(Pro_element(MonthQuota,10,10)) 'October',
      To_number(Pro_element(MonthQuota,11,11)) 'November',
      To_number(Pro_element(MonthQuota,12,12)) 'December'
FROM pub.salesrep;
GRANT SELECT on mysalesrep TO PUBLIC;
COMMIT;
```

Now you can start Crystal Reports 9 and develop you own report using this view instead of the original pub.salesrep table. The above view is available to everyone running the report. This means that it is a one-time operation a DBA may need to perform on your Progress database.

7.1.2 Using Crystal Reports 9 SQL Commands

Another way to access individual elements in Progress extents is to actually create an SQL Command in Crystal Report that does the splitting for you.

To check how this works, you can:

1. Download the Crystal Reports Sample (zip file available at: http://www.psdn.com/library/sql.htm)
2. Create a Sports2000 database.
3. Start your database server as described in this document.
4. Configure an ODBC Data Source named “9.1D 5555” as described in this document.
5. Start Crystal Reports and open the report named: “Split extent Report.rpt” and execute it

The report will display the separated values for each element.

Copying this command to the repository available with Crystal Reports 9 will give you additional flexibility while developing other reports without requiring the assistance of a Progress SQL DBA. It basically enables you to reuse over and over again this same statement with any problem.

7.2 Outer Join Syntax Used with Crystal Reports 9

When using Crystal Reports 8 with Progress SQL-92, the following error occurred when performing an outer Join: “[MERANT][ODBC PROGRESS driver]The keywords LEFT OUTER JOIN were not found in an outer join escape sequence”.

The solution was to enter the following value in the Windows Registry:

- Under [HKEY_CURRENT_USER\Software\Seagate Software\Crystal Reports\DatabaseOptions\OuterJoin]
- Create a new string value with the following information:
  - Name: pgpro915
  - Value: PlusEqual

With Crystal Reports 9, this manipulation is no longer required.

To check it, you can:

1. Download the Crystal Reports Sample (zip file available at: http://www.psdn.com/library/sql.htm).
2. Create a Sports2000 database.
3. Start your database server as described in this document.

4. Configure an ODBC Data Source named “9.1D 5555” as described in this document.
5. Start Crystal Reports and open the report named: “OJ Report.rpt” and execute it.
6. Check the SQL statement with the Menu Item “Show SQL query…” located under the “Database” menu.

7.3 Using Dates and Time in Your Reports as Input Parameters or when Displaying Data

The format used in your report when displaying data or when entering value for input parameters is picked up from the operating system. Under Windows XP, this format is defined in the Control Panel in the “Regional and Language Options”.

If for instance on your PC you have specified a short date format like “dd-MM-yyyy”, dates will be displayed like this in your report: “31-12-2002”.

To check it, you can:

1. Download the Crystal Reports Sample (zip file available at: http://www.psdn.com/library/sql.htm).
2. Create a Sports2000 database.
3. Start your database server as described in this document.
4. Configure an ODBC Data Source named “9.1D 5555” as described in this document.
5. Start Crystal Reports and open the report named: “Date Report.rpt” and execute it.
6. The report asks for an input parameter that should be between “01-01-1997” and ”31-03-1998” the format for the value entered is based on your local date settings (Short date format) with minimum variation for ease of use.

The data display in the Order Date column should match your local date settings (short date format).

7.4 Calling SQL-92 Stored Procedures from Crystal Reports 9

In the Progress SQL-92 implementation, you can also use stored procedures. Crystal Reports 9 offers a couple of ways to access these database objects.

Using them in your report should be possible with the help of the following features:

- Using Crystal Reports 9 SQL commands
- Accessing stored procedures as tables in Crystal Reports 9

However there are currently some issues in this area which prevent proper access to and use of these Progress SQL-92 objects within a report. These issues are currently under investigation.

8 Conclusion

Once you have established and tested your connections from Crystal Reports to your Progress database, you are ready to start developing and running reports. Here’s a summary of the general steps involved:

- Check that you have the proper Crystal Decisions and Progress products installed. For Progress, a product including the SQL-92 server and ODBC driver is required.

- Configure the Progress database as an ODBC data source. Be sure to select the transaction isolation level that matches your performance and access requirements.
- Decide what kind of security your data requires and implement it through one of the mechanisms described in this white paper.
- Start the Progress database server.
- Start the Crystal Decisions product.
- If your reports are accessing Progress extents, views, or stored procedures, keep in mind some of the techniques for and issues around accessing these objects.

.

Appendix A: Notes on the Practical Example

You can replace any of the following values by their equivalents in your context:

Value Description	Values from the example:	Your value should be:
Data Source Name	9.1D 5555	The name you give to your Data Source in the ODBC Administrator utility. This name is re-used from Crystal Reports 9
Database Name	Sports2000	Your database name
Port Name or Port Number	5555	The port number or port name on which you have started your Progress Broker
Host Name or Host IP Address	Localhost	The Hostname or IP address of the machine hosting your Progress SQL-92 database server
User ID	Administrator	Your SQL-92 DBA User ID or a User ID with sufficient privileges to build or use reports against the database accessed
Password	(None)	The Password associated with the above User ID (only if login validation is enabled)
Fetch Array size:	500	The number of rows the driver retrieves when fetching from the server. This is not the number of rows given to the user.

Appendix B: Creating a Default SQL-92 DBA in Your Progress Database

In some very specific cases a Progress database has no SQL-92 DBA created (other then ‘sysprogress’ who is disabled by default).

When this situation occurs you can do the following:

If login validation is enabled, you need to disable it by doing the following:
1. Dump all the users defined in your database.
2. Delete all the users created in your database.

The detailed steps are as follows:
1. Stop every processes running on your database.
2. Make a back up of your current database.
3. Ensure you have disabled after imaging and 2 phase commit.
4. Create a copy of your database with the PROCOPY utility. Example of the command:
    a. procopy sports2000 sports3000
5. Start a database server on the newly created copy.
6. Look in your copied database log file for this message: “procopy session begin for Administrator on:. (451)” (Instead of Administrator you should find another Value. This name is the Default DBA User ID for Progress SQL-92.)

If you had login validation previously enabled, you need to recreate the users by doing the following:
1. Load all the users defined in your database (previously saved).
2. If the SQL DBA User ID is not in that list create a new User ID with that name.

You have now successfully created the proper SQL DBA for your Progress database.

Worldwide and North American Headquarters

Progress Software Corporation, 14 Oak Park, Bedford, MA 01730 USA Tel: 781 280 4000 Fax: 781 280 4095

Europe/Middle East/Africa Headquarters

Progress Software Europe B.V. Schorpioenstraat 67 3067 GG Rotterdam, The Netherlands Tel: 31 10 286 5700 Fax: 31 10 286 5777

Latin American Headquarters

Progress Software Corporation, 2255 Glades Road, One Boca Place, Suite 300 E, Boca Raton, FL 33431 USA Tel: 561 998 2244 Fax: 561 998 1573

Asia/Pacific Headquarters

Progress Software Pty. Ltd., 1911 Malvern Road, Malvern East, 3145, Australia Tel: 61 39 885 0544 Fax: 61 39 885 9473

PROGRESS SOFTWARE

www.progress.com