crystal decisions™

Crystal Reports

Logging on to a database using Crystal Print Engine API calls

Overview

This document describes how to logon to a database at runtime by making direct Application Programming Interface (API) calls using the Crystal Report Print Engine (Crpe32.dll). This document is for use with Crystal Reports 5 and higher.

Contents

Introduction

When using a report that is using an ODBC data source or a SQL based database such as Microsoft SQL Server, a connection must be made to the database. The API Print Engine calls that allow for a new connection to the database are **PESetnthtablelogoninfo** and **PELogonServer**.

The **PELogoninfo** structure is used to pass the logon information to **PESetNthTableLogonInfo** and **PELogonServer**. The **PEGetNthTableLogonInfo** function retrieves into the **PELogOnInfo** structure the log on information required by a report.

Although, this document uses Microsoft Visual Basic code to demonstrate how these API Print Engine calls are used in an application, they can be used with other development languages.

PELogonServer

PELogonServer is not specific to a report, instead **PELogonServer** is called once to open a connection to a specified database. The connection remains open until **PELogOffServer** is called. **PELogonServer** is equivalent to clicking **Log On Server** from the **Database** menu in the Seagate Crystal Report Designer.

How to use PELogonServer

BOOL CRPE_API **PELogonServer**
(const char FAR *DLLName,
PELogonInfo FAR *logOnInfo)

Parameter	Description
DLLname	Specifies the name of a Seagate Crystal Reports DLL. This DLL is for the server or password protected non-SQL table you want to log on to, for example, "Pdsodbc.dll". Note that the DLL name must be enclosed in quotes. DLL names have the following naming convention: ▪ Pdb*.dll for standard (non-SQL) databases ▪ Pds*.dll for SQL/ODBC databases.
LogOnInfo	Specifies a pointer to the **PELogonInfo** structure.

NOTE	To find the DLLname for a report, click **Convert Database Driver** from the **Database** menu in the Crystal Reports Designer. Use the DLLname displayed beside the **From** field.

PESetNthTableLogonInfo

Before a report can be printed at runtime, there must be a connection to a specified database. Each time a report is printed, **PESetNthTableLogonInfo** connects the report to a database using the logon information from the **PELogonInfo** structure. Once the print job is complete, the report is automatically logged off from the database.

PESetNthTableLogonInfo is report specific and must be called for each report, including subreports as logging off is performed automatically when the print job is closed.

How to use PESetNthTableLogonInfo

```
BOOL CRPE_API PESetNthTableLogonInfo (
        short printJob,
        short tableN,
        PELogonInfo FAR *logOnInfo,
        BOOL PropagateAcrossTables);
```

Parameters	Description
printJob	Specifies the handle of the print job for which you want to set the table logon information.
tableN	**SPECIFIES THE NUMBER OF THE TABLE FOR WHICH YOU WANT TO SET LOGON INFORMATION. THE FIRST TABLE IS TABLE 0. THE LAST TABLE IS N-1.**
LogOnInfo	Specifies a pointer to the **PELogonInfo** structure.
PropagateAcross Tables	Indicates whether or not the program should apply the new logon information to any other tables in the report that have the same server and database names as the specified table.

NOTE	When setting **PropagateAcrossTables**, you may use either **True** or **False**. Setting the value to **True** will apply logon information to all other tables in the report that have the same server and database names. Setting the value to **False** will apply the logon information only to the selected tables. For example, when you create a report off a single database (one MDB file with multiple tables) set the **PropagateAcrossTables** parameter to **TRUE**. This insures that the changes are made to all tables in the MDB file

PELogonInfo

The **PELogonInfo** structure contains logon information that is used by the following function calls:

- **PEGetNthTableLogoninfo**
- **PESetNthTableLogonInfo**
- **PELogonServer**
- **PELogOffServer**.

These API Print Engine calls are used for logging on or off SQL and secured non-SQL databases.

How to use PELogonInfo

```
struct PELogonInfo
        {WORD structSize;
        char ServerName [PE_SERVERNAME_LEN];
        char DatabaseName [PE_DATABASENAME_LEN];
        char UserID [PE_USERID_LEN];
        char Password [PE_PASSWORD_LEN];};
```

Parameters	Description
StructSize	Specifies the size of the **PELogonInfo** structure. You must initialize this member to the correct structure size, for example, ***.StructSize*** = **PE_SIZEOF_LOGON_INFO**.
ServerName	Specifies the server name used to create the report. For ODBC databases, use the ODBC Data Source Name.
DatabaseName	Specifies the database name used to create the report.
UserID	Specifies the user ID necessary to log on to the server.
Password	Specifies the password necessary to log on to the server.

NOTE	• When you pass an empty string ("") for the **ServerName, DatabaseName, and UserID** parameters, Crystal Reports uses the value already set in the report. If you want to override a value, use a non-empty string (for example, "myUserID"). If you are using Visual Basic, all strings must be null-terminated. • If your report uses a Microsoft Access database via ODBC, the data source that is indicated in the **ServerName** parameter must include the complete path name for the MS Access database.

Log on using PELogonServer

The following example is written in MS Visual Basic 6. It demonstrates how to use **PELogonServer** and **PELogonInfo** to connect to a report that is using an ODBC connection.

NOTE	The Global32.bas (Global.bas for 16 bit) file is installed in your Seagate Crystal Reports folder. This file is used to declare all Crystal Report Engine API functions for Visual Basic and must be included in your application.

.

```
Private Sub Open_Job_Click()

Dim LogOnInfo As PELogonInfo


LogonInfo.StructSize = PE_SIZEOF_LOGON_INFO

LogonInfo.ServerName = "sqlkhai" + Chr$(0)

LogonInfo.DatabaseName = "Reports" + Chr$(0)

LogonInfo.UserID = "Vantech" + Chr$(0)

LogonInfo.Password = "Vantech" + Chr$(0)


SetLogonInfo = PELogonServer("PDSODBC.DLL" + Chr$(0),
LogonInfo)


‘Error checking to ensure that

'If the Logon was successful, notify the 'User with a
'success' message, else let the user know what error

'occurred when logging on and what the error number is.


If SetLogonInfo <> 0 Then

MsgBox "The job opened successfully for sqlsub.rpt"

MsgBox "The job number is" & job

Else

ErrorNum = PEGetErrorCode(0)

MsgBox "The job failed to open for sqlsub.rpt"

MsgBox "The error code is: " & ErrorNum

End If


End Sub
```

NOTE	• The **ServerName, DatabaseName, UserID**, and **Password** parameters for **PELogonInfo** can be found by clicking **Set Location** from the **Database** menu in the Crystal Report Designer. The dialog screen that appears will contain the members

	mentioned above. • When connecting to a report that uses ODBC, the **ServerName** parameter in **PELogonInfo** is referring to the ODBC Data Source Name. • When connecting to a report that uses a native connection, the **ServerName** parameter in **PELogonInfo** is referring to the physical server name.

Log on using PESetNthTableLogonInfo

The following example demonstrates how to use **PESetNthTableLogonInfo** and **PELogonInfo** to connect to a report that is using an ODBC Connection.

```
Private Sub Print_Click()
Dim LogOnInfo As PELogonInfo
LogonInfo.StructSize = PE_SIZEOF_LOGON_INFO
LogonInfo.ServerName = "SQL Server" + Chr$(0)
LogonInfo.DatabaseName = "techtest" + Chr$(0)
LogonInfo.UserID = "Vantech" + Chr$(0)
LogonInfo.Password = "Vantech" + Chr$(0)

Handle = PEOutputToWindow(Job, "sql.rpt", 0, 0, 700, 700,
0, 0)
Handle = PESetNthTableLogonInfo(Job, 0, LogOnInfo, True)
Handle = PEStartPrintJob(Job, True)
End Sub
```

NOTE	• The **ServerName, DatabaseName, UserID**, and **Password** parameters for **PELogonInfo** can be found by clicking **Set Location** from the **Database** menu in the Crystal Report Designer. The dialog screen that appears will contain the members mentioned above. • When connecting to a report that uses ODBC, the **ServerName** parameter in **PELogonInfo** is referring to the ODBC Data Source Name. • When connecting to a report that uses a native connection, the **ServerName** parameter in **PELogonInfo** is referring to the physical server name.

Log on using PEGetNthTableLogonInfo

The following example demonstrates how to use **PEGetNthTableLogonInfo** to return the **ServerName, DatabaseName** and **LogonID** from a report and populate the **PELogonInfo** structure

Since the password is not saved with the report, the **PEGetNthTableLogonInfo** function does not return a value for **Password**.

```
Private Sub Print_Click()


Dim LogOnInfo As PELogonInfo
Dim i As Integer

Dim Numberoftables As Integer

Dim GetLogoninfo As Integer

Dim errornum As Integer


LogonInfo.StructSize = PE_SIZEOF_LOGON_INFO

Numberoftables = PEGetNTables(Job)


For i = 0 To (Numberoftables - 1)

GetLogoninfo = PEGetNthTableLogoninfo(Job, i, LogOnInfo)

If GetLogoninfo = 0 Then

errornum = PEGetErrorCode(Job)

MsgBox "GetLogoninfo error number " & errornum

End If

LogonInfo.Password = "Vantech" + Chr$(0)

Handle = PESetNthTableLogonInfo(Job, i, LogOnInfo, True)

Next i


If PEStartPrintJob(Job, True) = 0 Then

errornum = PEGetErrorCode(Job)

MsgBox "PEStartPrintJob is not succeeding.  Error:  " &
errornum

End If

End Sub
```

Log on to a report that contains a subreport

When a report with a subreport is using a SQL based database or an ODBC data source, both the main report and subreport need to connect to the database. Use **PELogonServer** or **PESetNthTableLogonInfo** to connect to the database.

If the subreport is using the same ODBC data source or database as the main report, **PELogonServer** can be used to connect to both the main report and subreport. Since **PELogonServer** does not connect to a specific report, both the main and subreport can use the same database connection.

If the subreport is using a different data source or database than the main report, use **PESetNthTableLogonInfo** to create a separate connection for the main report and subreport.

In the following example, a separate connection is made to the main report and subreport (each report is using a different data source).

```
Dim LogOnInfo As PELogonInfo

Dim LogonInfo2 As PELogonInfo
```

```
‘Logon information for the main report
LogonInfo.StructSize = PE_SIZEOF_LOGON_INFO
LogonInfo.ServerName = "sqlkhai" + Chr$(0)
LogonInfo.DatabaseName = "craze" + Chr$(0)
LogonInfo.UserID = "Vantech" + Chr$(0)
LogonInfo.Password = "Vantech" + Chr$(0)

‘Logon information for the subreport
LogOnInfo2.StructSize = PE_SIZEOF_LOGON_INFO
LogOnInfo2.ServerName = "sql65" + Chr$(0)
LogOnInfo2.DatabaseName = "techtest" + Chr$(0)
LogOnInfo2.UserID = "Vantech" + Chr$(0)
LogOnInfo2.Password = "Vantech" + Chr$(0)

‘open and logon to the main report
Job = PeOpenPrintJob(“C:\test\mainreport.rpt”)
SetLogonInfo = PESetNthTableLogonInfo(job, 0, LogOnInfo,
True)

If SetLogonInfo <> 0 Then
MsgBox "The job logged on successfully."
MsgBox "The job number is" & job
Else
ErrorNum = PEGetErrorCode(0)
MsgBox "The job failed to open."
MsgBox "The error code is: " & ErrorNum
End If

‘Open and logon to the subreport,
Handle = PEOpenSubreport(job, "mysubreport.rpt")
SetLogonInfo = PESetNthTableLogonInfo(job, 0, LogOnInfo2,
True)
```

PELogonServer and PESetNthTableLogonInfo differences

PELogonServer	PESetNthTableLogonInfo
Implied Logon - The database connection is not specific to a report. All reports use the same connection.	**Explicit Logon** - The database connection is report specific. That is, each report requires a separate logon.

PELogonServer	PESetNthTableLogonInfo
Requires the parameter: ***DatabaseDLL***	**DOES NOT REQUIRE THE PARAMETER:** *DATABASEDLL*
PEGetNthTableLogonInfo can be used to get the **PELogonInfo** parameter values only if the report print job has already been opened. Otherwise, the **PELogonInfo** parameters have to be hard coded.	**PEGetNthTableLogonInfo** can be used to get the **PELogonInfo** parameter values (except for the password that has to be specified)
Can be called before or after the report print job is opened.	Needs to be called for every print job that is opened (after **PEOpenPrintJob**).
The connection remains open (even if the print job is closed) until **PELogOffServer** is called.	**LOGGING OFF IS PERFORMED AUTOMATICALLY WHEN THE PRINT JOB IS CLOSED.**
Can only change a single data source in a report.	Can change multiple data sources in a single report.

Troubleshooting

The following are some common issues that are encountered when trying to logon to a database and possible solutions to these errors.

Error 599 – SQL Server Not Open

- Invalid information was passed to **PELogonServer** or **PESetNthTableLogonInfo.**

  Ensure that the **PELogonInfo** structure information is correct. In the Report Designer, click **Set Location** from the **Database** menu. The information listed at the bottom of the dialog screen should match that of the **PELogonInfo** structure.

- The members of the **PELogonInfo** call are not being null terminated (only applies to Visual Basic).

- The database client software should be installed on the computer.

- The directory where the database client is installed should be in the 'PATH' of the computer. (To check this, go to DOS, and type in PATH)

- If this error occurs on a test target computer, and not on the development computer, runtime files could be missing.

  For a listing of required runtime files, refer to the **Runtime File Requirement Help File**, found in the Crystal Report program group.

The data source has changed, but the report is still showing the original data.

- The report was saved with data. This can be confirmed by opening the report; if the report opens in preview mode, then it has saved data. To avoid this error message, open the report and on the **File** menu ensure **Save data with report** is unchecked.

NOTE	For further information about database connectivity issues at runtime, search the Crystal Decisions Knowledge Base at http://support.crystaldecisions.com/kbase.

Contacting Crystal Decisions for Technical Support

We recommend that you refer to the product documentation and that you visit our Technical Support web site for more resources.

Self-serve Support:

http://support.crystaldecisions.com/

Email Support:

http://support.crystaldecisions.com/support/answers.asp

Telephone Support:

http://www.crystaldecisions.com/contact/support.asp