Using the APLNext DataBase Interface Tool for Microsoft SQL Server

What is the Tool?

The APLNext Database Interface Tool simplifies the use of Microsoft SQL Server in an application system developed using APL+Win, VisualAPL, C#, etc.

The Tool employs the Microsoft ADO.Net (active data objects) class for .Net which has superior performance and features compared to previous technologies, such as ADODB, ODBC, MDAC, Jet, etc.

With the (no-cost) availability of Microsoft SQL Server Express database, scalable-to-enterprise-level data base software is more affordable and the logical choice for professionally-designed application systems.

The object model for the APLNext Database Interface Tools includes methods and properties to easily create a secure, high-performance application system data interface.

Refer to the “APLNext Database Interface Tools Overview.pdf” for more information.

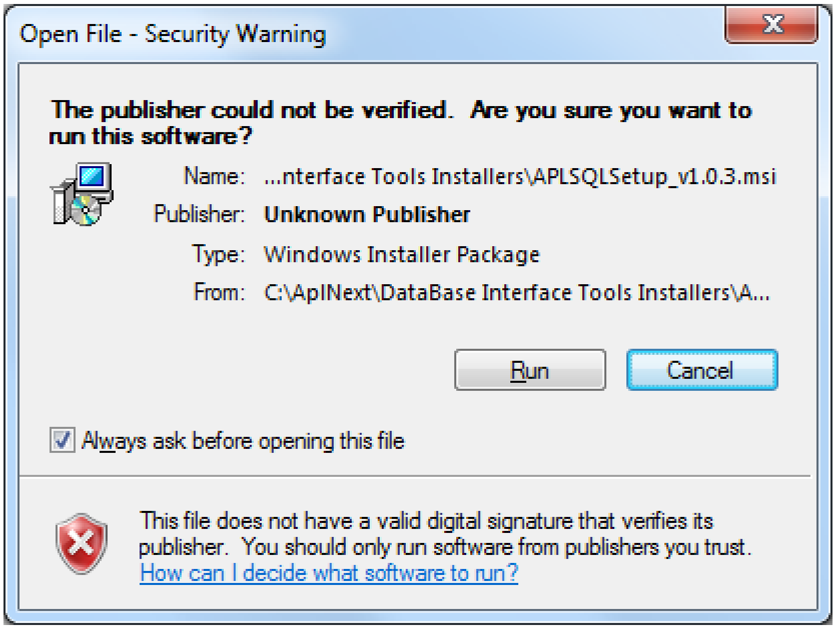
Obtaining the Tools:

The tool is software licensed by APL2000. For more information please contact sales@apl2000.com

Installing the Tools:

Administrator credentials on the target workstation are required to properly register the ActiveX interface for the tools so that they can be consumed by APL+Win. Double click the installer, e.g. APLSQLSetup_v1.0.3.msi, and follow the directions on the screen.

Click the Run button if the security warning dialog is presented:



Open File - Security Warning

The publisher could not be verified. Are you sure you want to run this software?

Name: ...nterface Tools Installers\APLSQLSetup_v1.0.3.msi

Publisher: **Unknown Publisher**

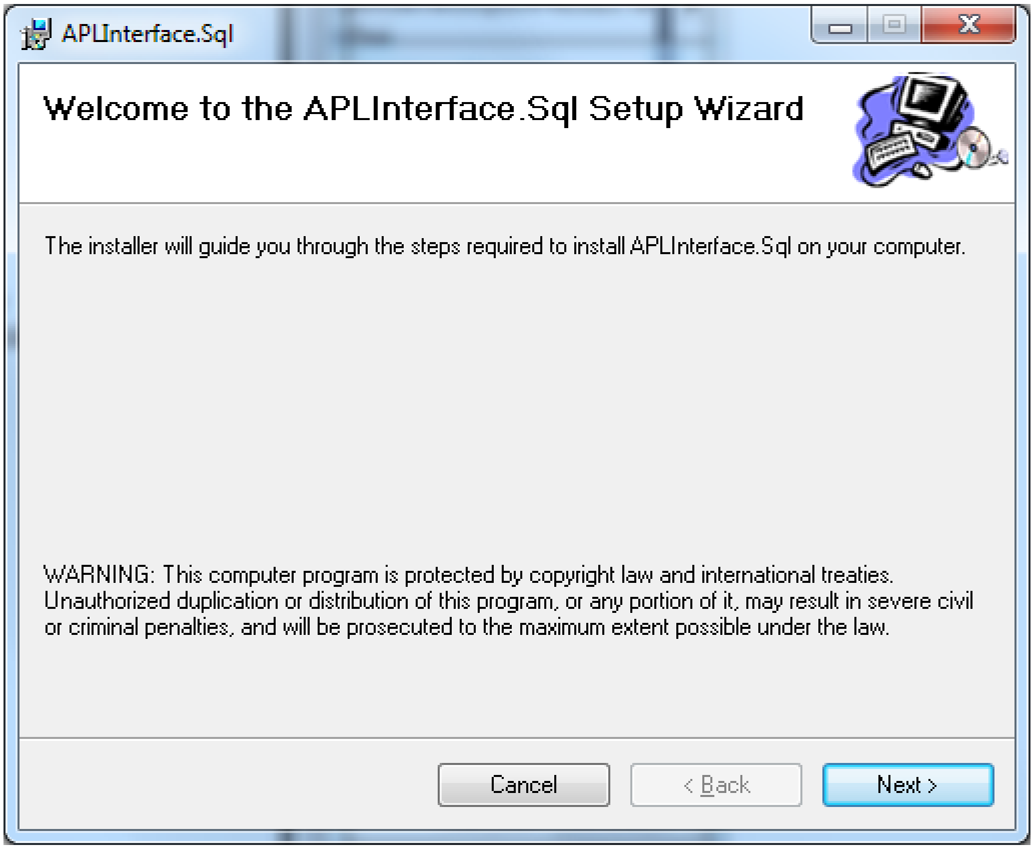
Type: Windows Installer Package

From: C:\AplNext\DataBase Interface Tools Installers\A...

☑ Always ask before opening this file

This file does not have a valid digital signature that verifies its publisher. You should only run software from publishers you trust.
How can I decide what software to run?

Click the Next button when the Welcome dialog is presented:

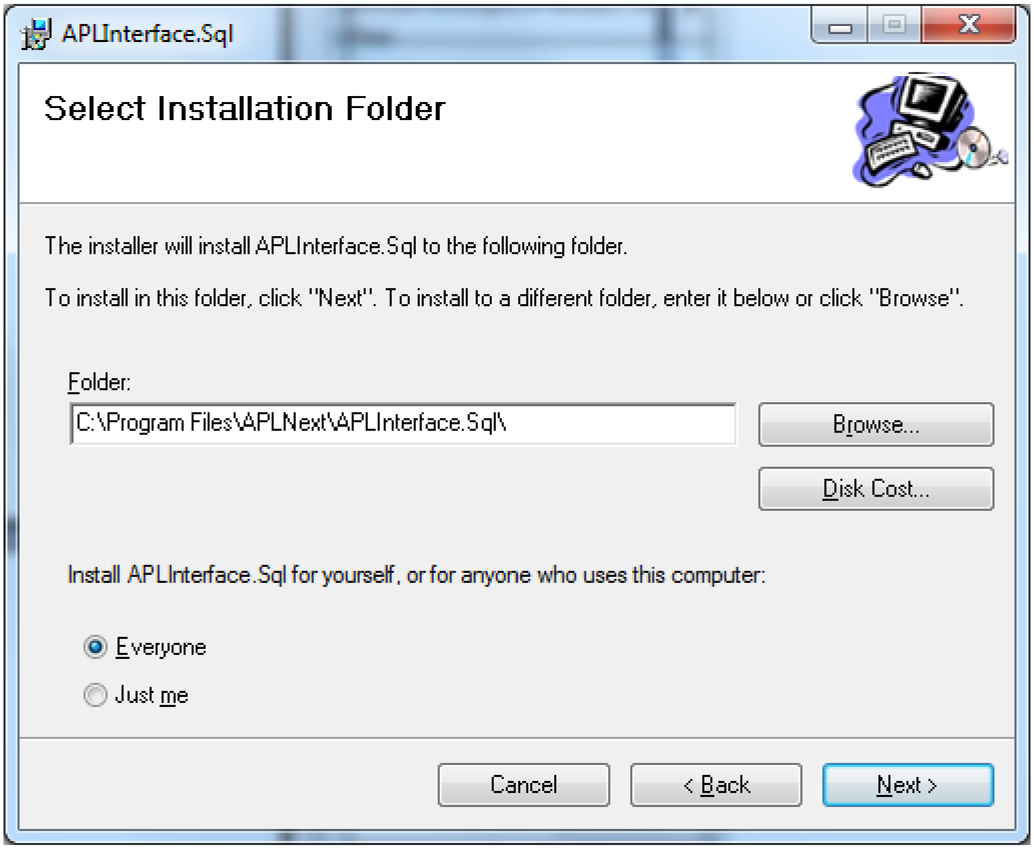


Welcome to the APLInterface.Sql Setup Wizard

The installer will guide you through the steps required to install APLInterface.Sql on your computer.

WARNING: This computer program is protected by copyright law and international treaties. Unauthorized duplication or distribution of this program, or any portion of it, may result in severe civil or criminal penalties, and will be prosecuted to the maximum extent possible under the law.

On the Select Installation Folder dialog, select ‘Everyone’ and if desired modify the target folder location and then click the Next button:



Select Installation Folder

The installer will install APLInterface.Sql to the following folder.

To install in this folder, click "Next". To install to a different folder, enter it below or click "Browse".

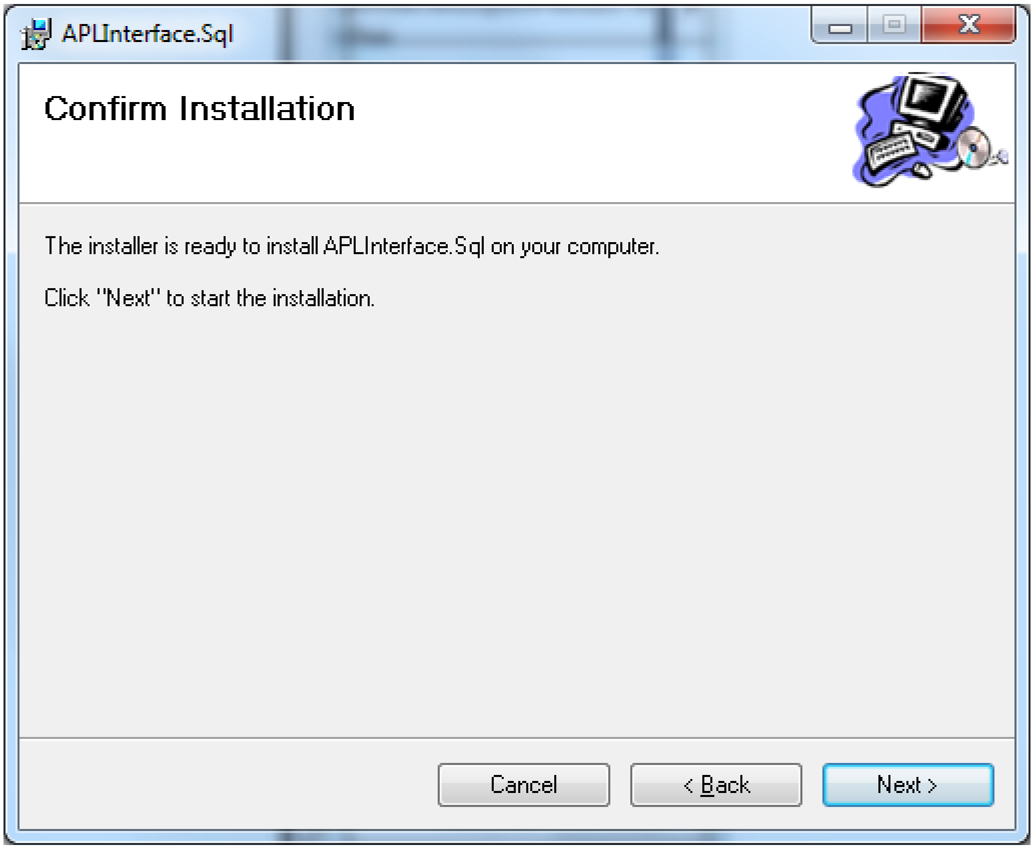
Folder: C:\Program Files\APLNext\APLInterface.Sql\

Install APLInterface.Sql for yourself, or for anyone who uses this computer:

● Everyone

○ Just me

Click the Next button when the Confirm Installation dialog is presented:

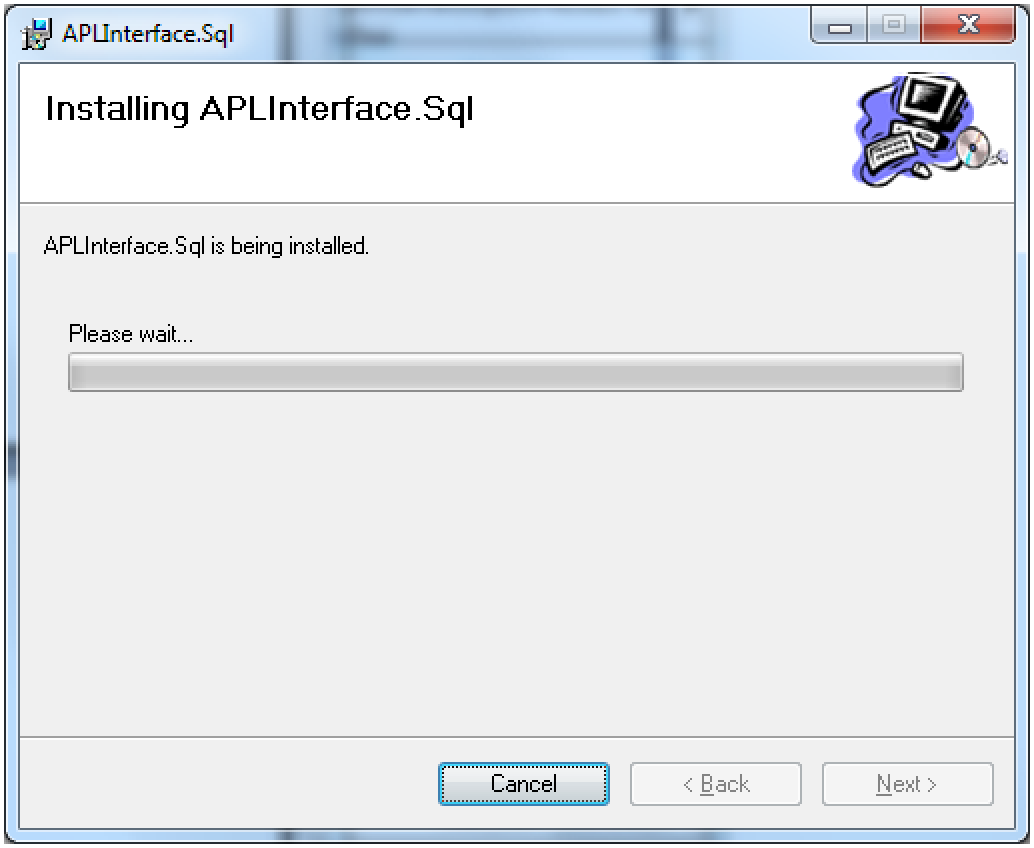


Confirm Installation

The installer is ready to install APLInterface.Sql on your computer.

Click "Next" to start the installation.

The Installing … dialog is presented while the files are copied to the target workstation and the tool is registered. During the installation a security warning may be issued by the workstation. The installer will check that the .Net Framework pre-requisite is installed before continuing the installation.

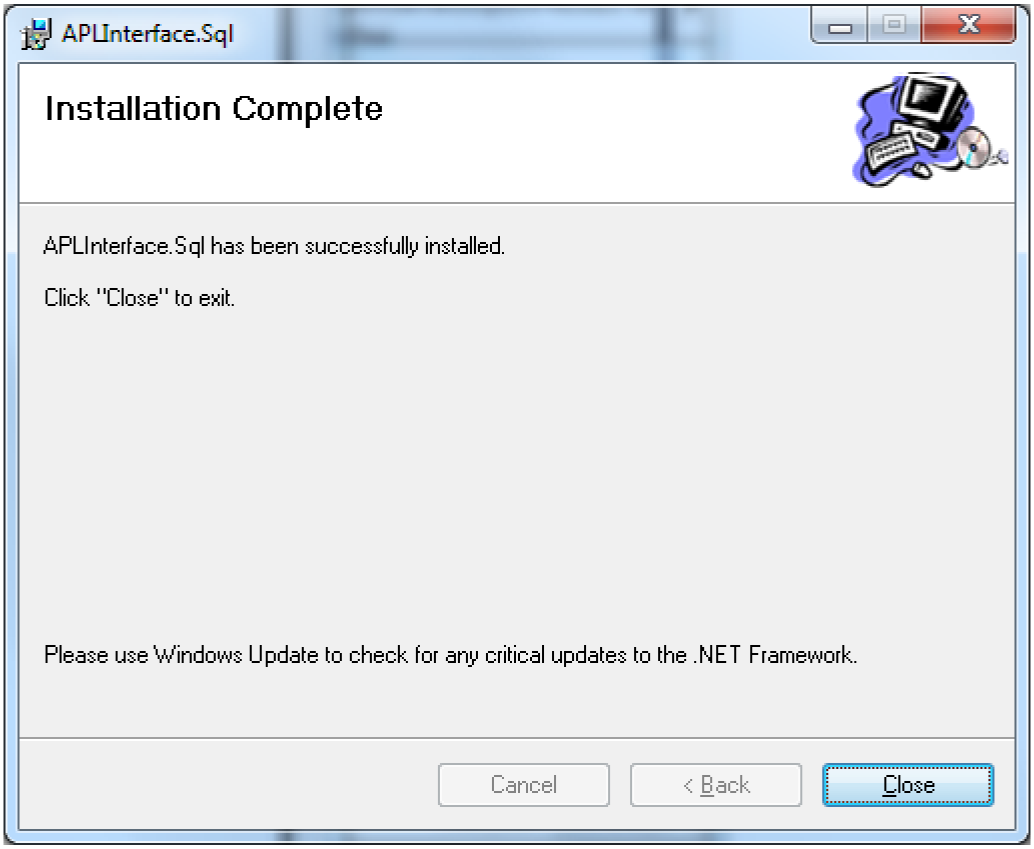


Installing APLInterface.Sql

APLInterface.Sql is being installed.

Please wait...

When the installation is complete, the following dialog is presented:



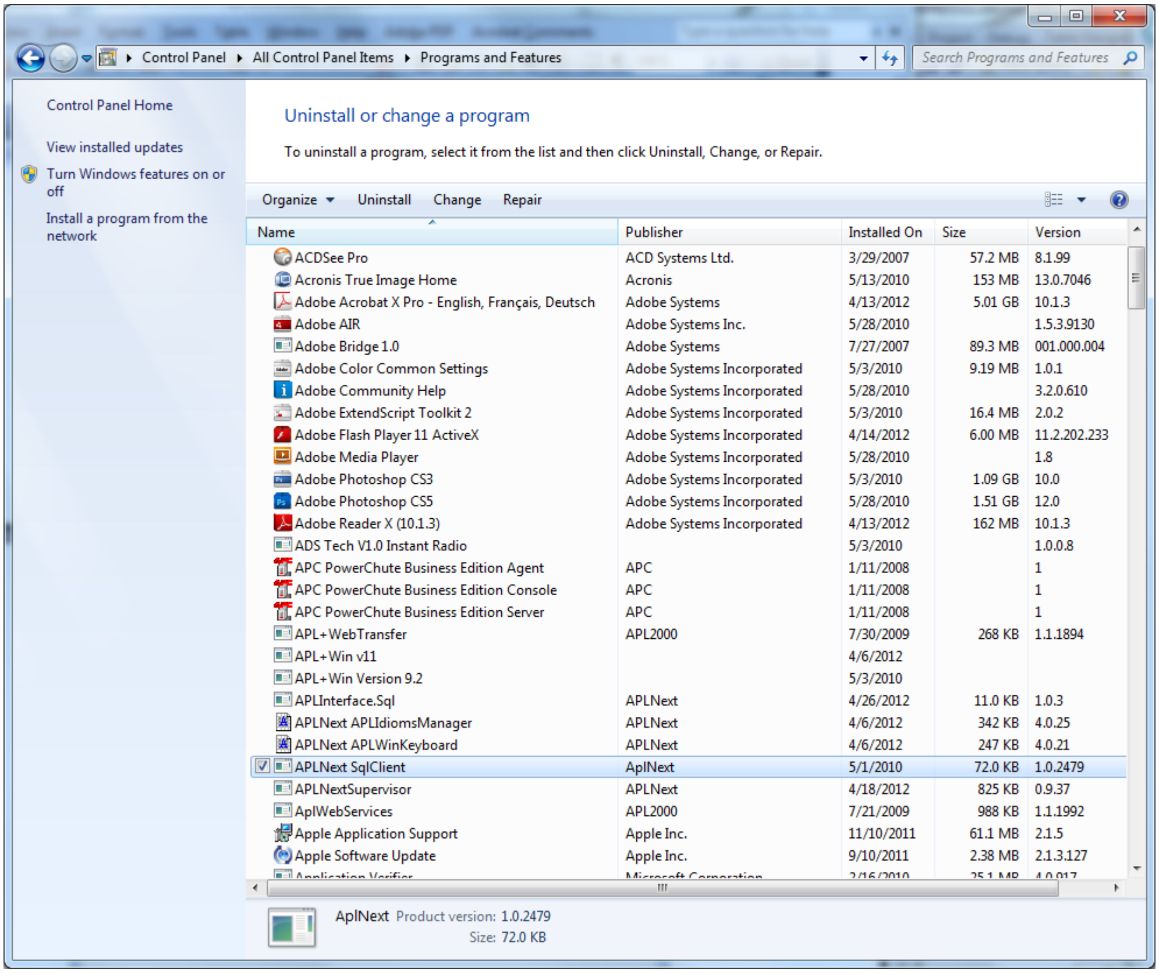
Installation Complete

APLInterface.Sql has been successfully installed.

Click "Close" to exit.

Please use Windows Update to check for any critical updates to the .NET Framework.

Successful installation may be verified using Start > Control Panel > Programs and Features:


Control Panel ▸ All Control Panel Items ▸ Programs and Features
Search Programs and Features
Control Panel Home
View installed updates
Turn Windows features on or off
Install a program from the network
Organize ▾
Uninstall
Change
Repair

Uninstall or change a program

To uninstall a program, select it from the list and then click Uninstall, Change, or Repair.

Name	Publisher	Installed On	Size	Version
ACDSee Pro	ACD Systems Ltd.	3/29/2007	57.2 MB	8.1.99
Acronis True Image Home	Acronis	5/13/2010	153 MB	13.0.7046
Adobe Acrobat X Pro - English, Français, Deutsch	Adobe Systems	4/13/2012	5.01 GB	10.1.3
Adobe AIR	Adobe Systems Inc.	5/28/2010		1.5.3.9130
Adobe Bridge 1.0	Adobe Systems	7/27/2007	89.3 MB	001.000.004
Adobe Color Common Settings	Adobe Systems Incorporated	5/3/2010	9.19 MB	1.0.1
Adobe Community Help	Adobe Systems Incorporated	5/28/2010		3.2.0.610
Adobe ExtendScript Toolkit 2	Adobe Systems Incorporated	5/3/2010	16.4 MB	2.0.2
Adobe Flash Player 11 ActiveX	Adobe Systems Incorporated	4/14/2012	6.00 MB	11.2.202.233
Adobe Media Player	Adobe Systems Incorporated	5/28/2010		1.8
Adobe Photoshop CS3	Adobe Systems Incorporated	5/3/2010	1.09 GB	10.0
Adobe Photoshop CS5	Adobe Systems Incorporated	5/28/2010	1.51 GB	12.0
Adobe Reader X (10.1.3)	Adobe Systems Incorporated	4/13/2012	162 MB	10.1.3
ADS Tech V1.0 Instant Radio		5/3/2010		1.0.0.8
APC PowerChute Business Edition Agent	APC	1/11/2008		1
APC PowerChute Business Edition Console	APC	1/11/2008		1
APC PowerChute Business Edition Server	APC	1/11/2008		1
APL+WebTransfer	APL2000	7/30/2009	268 KB	1.1.1894
APL+Win v11		4/6/2012		
APL+Win Version 9.2		5/3/2010		
APLInterface.Sql	APLNext	4/26/2012	11.0 KB	1.0.3
APLNext APLIdiomsManager	APLNext	4/6/2012	342 KB	4.0.25
APLNext APLWinKeyboard	APLNext	4/6/2012	247 KB	4.0.21
☑ APLNext SqlClient	AplNext	5/1/2010	72.0 KB	1.0.2479
APLNextSupervisor	APLNext	4/18/2012	825 KB	0.9.37
AplWebServices	APL2000	7/21/2009	988 KB	1.1.1992
Apple Application Support	Apple Inc.	11/10/2011	61.1 MB	2.1.5
Apple Software Update	Apple Inc.	9/10/2011	2.38 MB	2.1.3.127
Application Verifier	Microsoft Corporation	2/16/2010	25.1 MB	4.0.917

AplNext Product version: 1.0.2479
Size: 72.0 KB

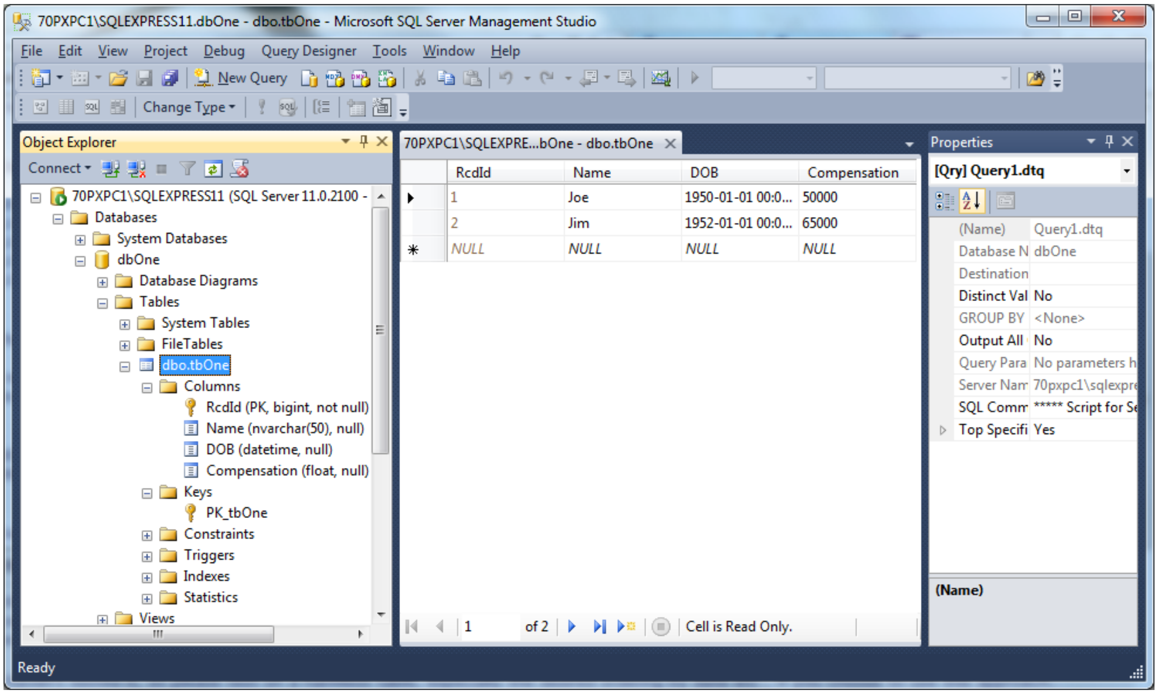
Microsoft SQL Server Must Be Installed and Running:

To use the tool Microsoft SQL Server must be installed and running on the target workstation or installed and running on a server to which the target workstation has access. Several versions of this Microsoft software are available. Microsoft SQL Express 2012 is available without charge to use in a test or production environment. Refer to the document “APLWin and MS SQL Express.pdf” for instructions on installing Microsoft SQL Express 2012 on the workstation.

A SQL Server Data Base Must Exist in the Running Instance of SQL Server:

An instance of a Microsoft SQL Server data base must exist on the target workstation or on a server to which the target workstation has access. The Microsoft SQL Server Management Studio is an excellent tool to establish this instance. Refer to the document “Microsoft SQL Server 2012 Management Studio.pdf” for instructions on creating a sample SQL data base and data base table.

The examples in the remainder of this document are based on the sample data base, tables and values illustrated in the above two documents. The initial state of the sample data base as displayed in the SQL Server Management Studio:


70PXPC1\SQLEXPRESS11.dbOne - dbo.tbOne - Microsoft SQL Server Management Studio
File
Edit
View
Project
Debug
Query Designer
Tools
Window
Help
New Query
Change Type
Object Explorer
Connect
70PXPC1\SQLEXPRE...bOne - dbo.tbOne
Properties
Ready

- 70PXPC1\SQLEXPRESS11 (SQL Server 11.0.2100 -
  - Databases
    - System Databases
    - dbOne
      - Database Diagrams
      - Tables
        - System Tables
        - FileTables
        - dbo.tbOne
          - Columns
            - RcdId (PK, bigint, not null)
            - Name (nvarchar(50), null)
            - DOB (datetime, null)
            - Compensation (float, null)
          - Keys
            - PK_tbOne
          - Constraints
          - Triggers
          - Indexes
          - Statistics
      - Views

	RcdId	Name	DOB	Compensation
▶	1	Joe	1950-01-01 00:0...	50000
	2	Jim	1952-01-01 00:0...	65000
*	NULL	NULL	NULL	NULL

1 of 2 Cell is Read Only.

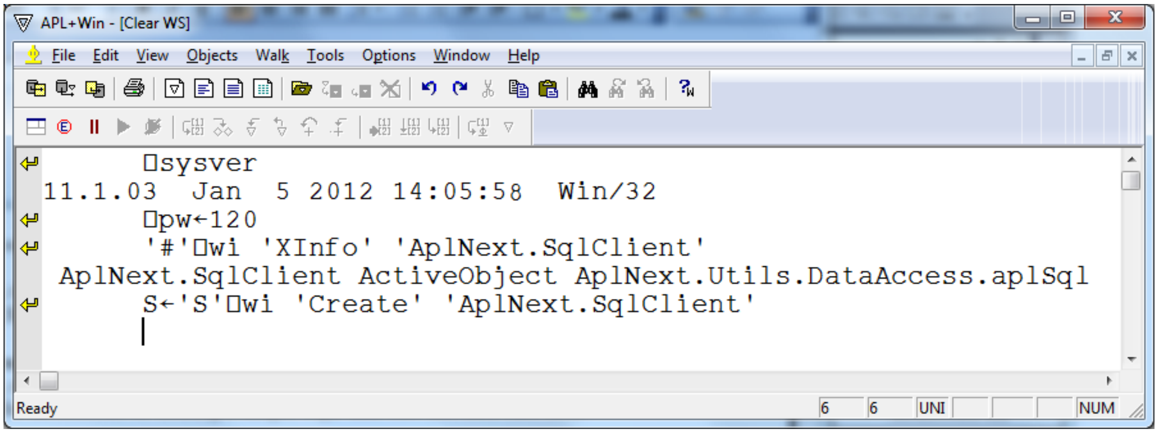
[Qry] Query1.dtq

(Name)	Query1.dtq
Database N	dbOne
Destination	
Distinct Val	No
GROUP BY	<None>
Output All	No
Query Para	No parameters h
Server Nam	70pxpc1\sqlexpre
SQL Comm	***** Script for Se
Top Specifi	Yes

(Name)

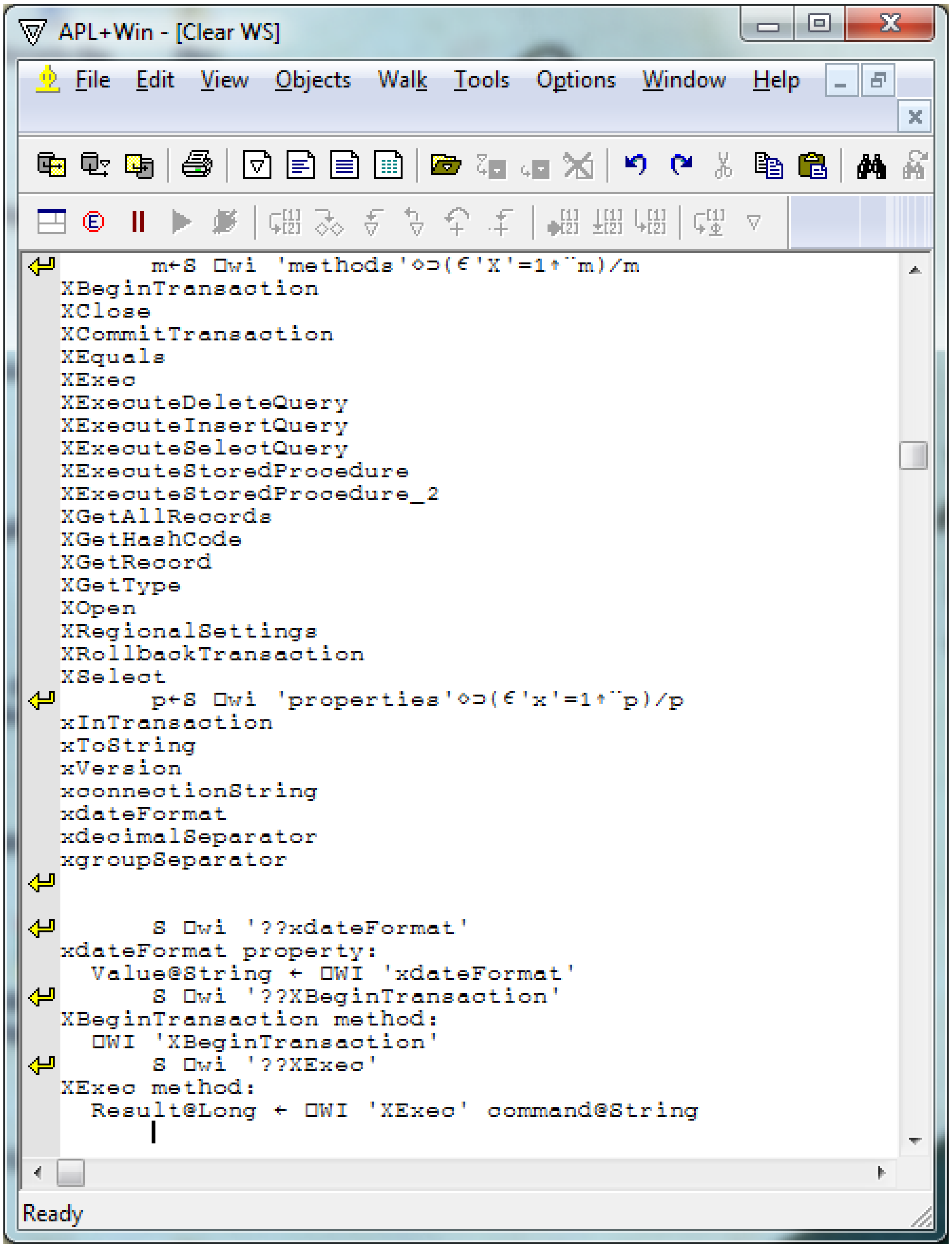
Start an APL+Win Session and Create an Instance of the Tools:

From the APL+Win programmer IDE use the []wi interface to create an instance of the tool:



```
      ⎕sysver
11.1.03  Jan  5 2012 14:05:58  Win/32
      ⎕pw←120
      '#'⎕wi 'XInfo' 'AplNext.SqlClient'
 AplNext.SqlClient ActiveObject AplNext.Utils.DataAccess.aplSql
      S←'S'⎕wi 'Create' 'AplNext.SqlClient'
```

Display Tool Methods (X…) and Properties (x…):
Technical documentation installed with the tool is provided for each of these. The APL+Win ActiveX syntax disclosure mechanism can be used too:



```
      m←S ⎕wi 'methods'⋄⊃(∊'X'=1↑¨m)/m
XBeginTransaction
XClose
XCommitTransaction
XEquals
XExec
XExecuteDeleteQuery
XExecuteInsertQuery
XExecuteSelectQuery
XExecuteStoredProcedure
XExecuteStoredProcedure_2
XGetAllRecords
XGetHashCode
XGetRecord
XGetType
XOpen
XRegionalSettings
XRollbackTransaction
XSelect
      p←S ⎕wi 'properties'⋄⊃(∊'x'=1↑¨p)/p
xInTransaction
xToString
xVersion
xconnectionString
xdateFormat
xdecimalSeparator
xgroupSeparator


      S ⎕wi '??xdateFormat'
xdateFormat property:
  Value@String ← ⎕WI 'xdateFormat'
      S ⎕wi '??XBeginTransaction'
XBeginTransaction method:
  ⎕WI 'XBeginTransaction'
      S ⎕wi '??XExec'
XExec method:
  Result@Long ← ⎕WI 'XExec' command@String
```

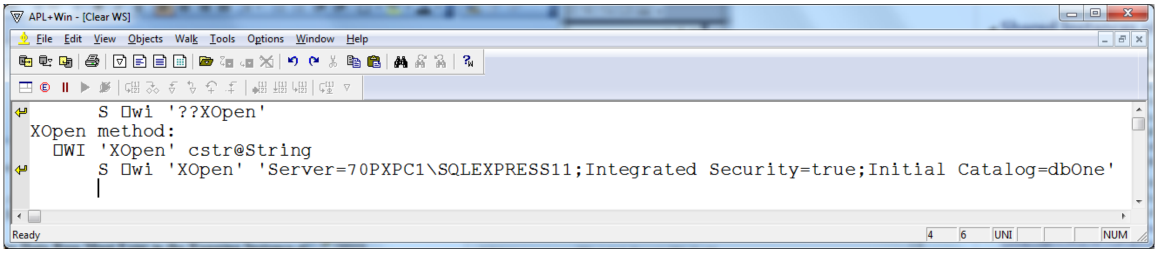
Open an SQL Data Base using a Valid Connection String:

Obtain the appropriate ‘connection string’ from your data base administrator. Additional property pairs may be required and used in the connection string.

For more information: http://www.connectionstrings.com/articles/show/all-sql-server-connection-string-keywords.

The sample connection string illustrated below includes:

- Server: workstationId\SQL Instance Name
- Integrated Security: true
- Initial Catalog: SQL data base name



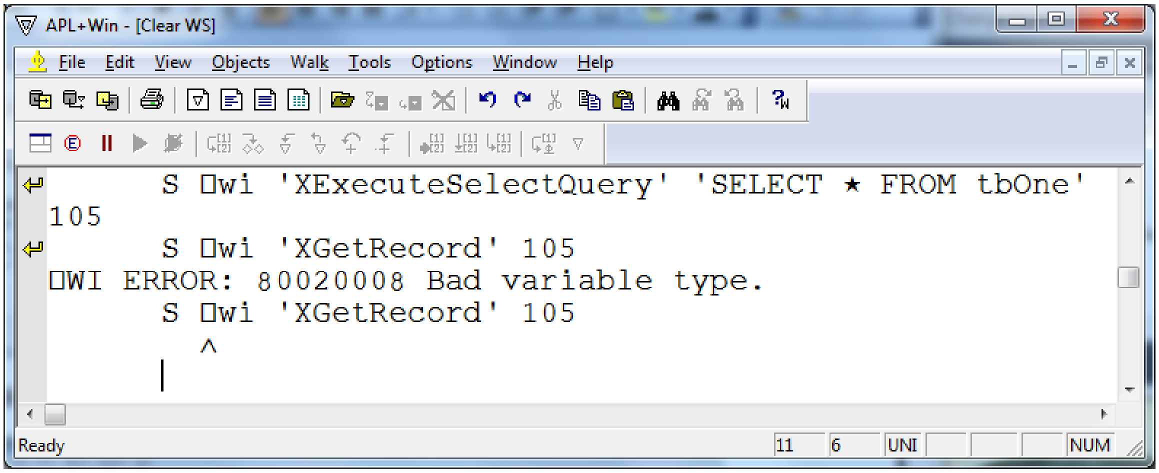
```
      S ⎕wi '??XOpen'
XOpen method:
  ⎕WI 'XOpen' cstr@String
      S ⎕wi 'XOpen' 'Server=70PXPC1\SQLEXPRESS11;Integrated Security=true;Initial Catalog=dbOne'
```

ExecuteSelectQuery on an SQL Data Base Table:

The XSelect method has an analogous argument syntax, but it returns the resulting record set *en masse*. It is suitable for record set results which can be entirely contained with the available memory.

The XExecuteSelectQuery method requires an SQL command statement and will return a pointer to the SQL data record set that is the result of the Select statement. The XGetRecord or XGetAllRecords method will return records in the record set.

The XGetRecord method can be repeated used to obtain the records in a record set returned by the XExecuteSelectQuery method. When there are no more records in a record set the value returned by XGetRecord method has shape 0 1 and APL+Win data representation 326 (32-bit pointer).

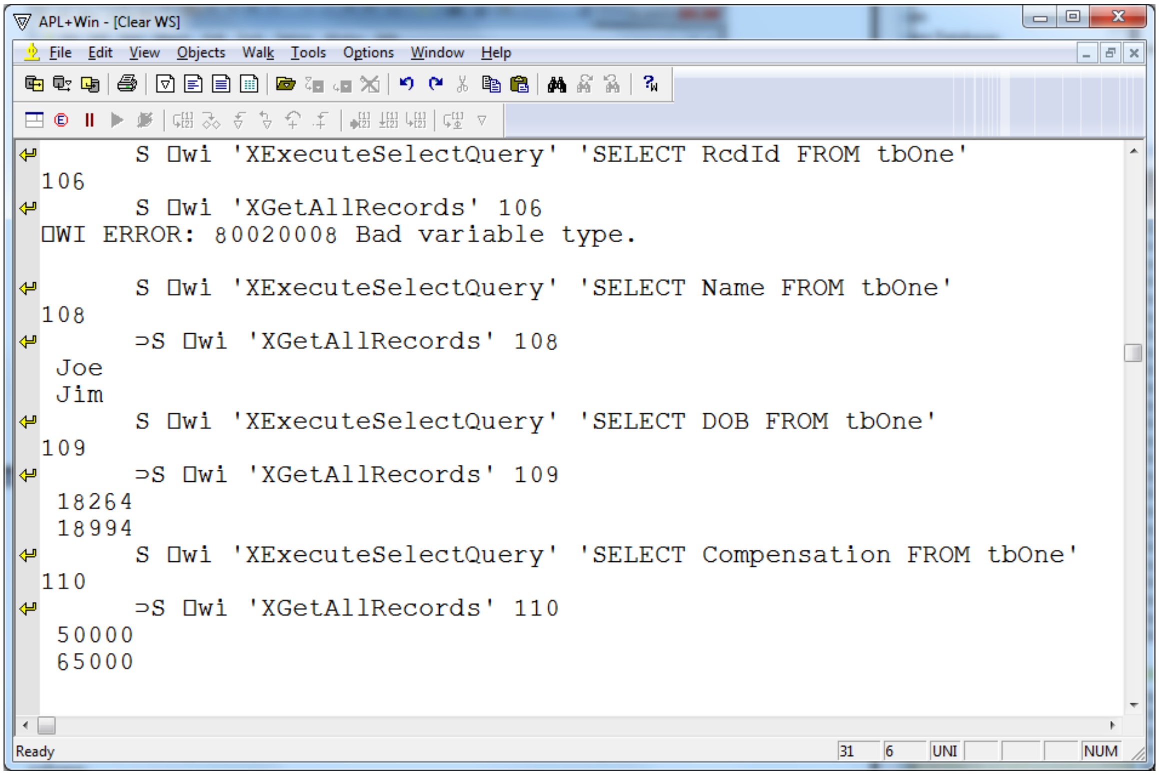


```
      S ⎕wi 'XExecuteSelectQuery' 'SELECT * FROM tbOne'
105
      S ⎕wi 'XGetRecord' 105
⎕WI ERROR: 80020008 Bad variable type.
      S ⎕wi 'XGetRecord' 105
        ^
```

In the sample data base this particular query failed because the data types being returned in the record set have no representation in APL. Recall that APL provides double, string, integer and bool data types, whereas Microsoft SQL Server includes many additional data types.

To be effective, the SQL Select statement will need to be more sophisticated.

Individually selecting the columns via separate Select statements will isolate the problematic column(s):

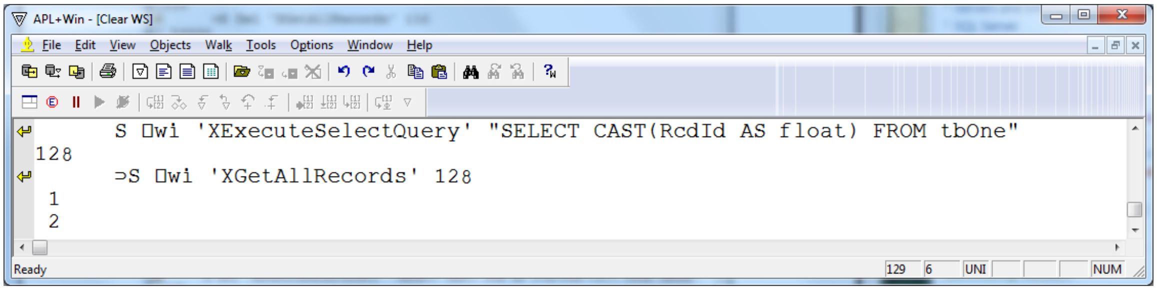


```
      S ⎕wi 'XExecuteSelectQuery' 'SELECT RcdId FROM tbOne'
106
      S ⎕wi 'XGetAllRecords' 106
⎕WI ERROR: 80020008 Bad variable type.

      S ⎕wi 'XExecuteSelectQuery' 'SELECT Name FROM tbOne'
108
      ⊃S ⎕wi 'XGetAllRecords' 108
 Joe
 Jim
      S ⎕wi 'XExecuteSelectQuery' 'SELECT DOB FROM tbOne'
109
      ⊃S ⎕wi 'XGetAllRecords' 109
 18264
 18994
      S ⎕wi 'XExecuteSelectQuery' 'SELECT Compensation FROM tbOne'
110
      ⊃S ⎕wi 'XGetAllRecords' 110
 50000
 65000
```

The above information indicates that the RcdId (Identity) and DOB (datetime) column values will need special handling in the SQL Select statement.
See here for more information:

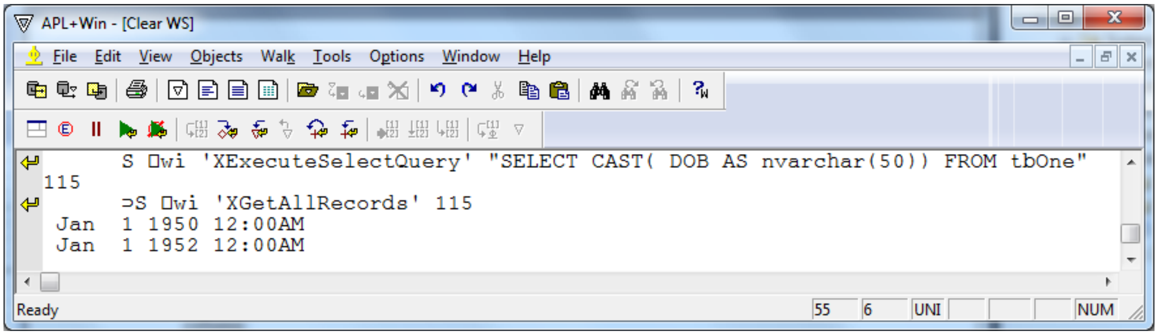
- Microsoft SQL Server data types: http://msdn.microsoft.com/en-us/library/ms187752.aspx
- Cast and Convert: http://msdn.microsoft.com/en-us/library/ms187928.aspx.

To successfully select the RcdId column (bigint SQL data type) cast it as float:



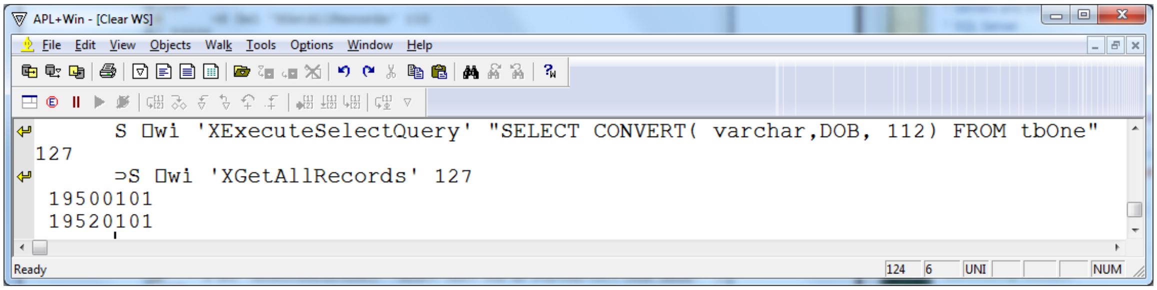
```
      S ⎕wi 'XExecuteSelectQuery' "SELECT CAST(RcdId AS float) FROM tbOne"
128
      ⊃S ⎕wi 'XGetAllRecords' 128
 1
 2
```

To successfully select DOB column (datetime SQL data type) cast it as a 50-character string:



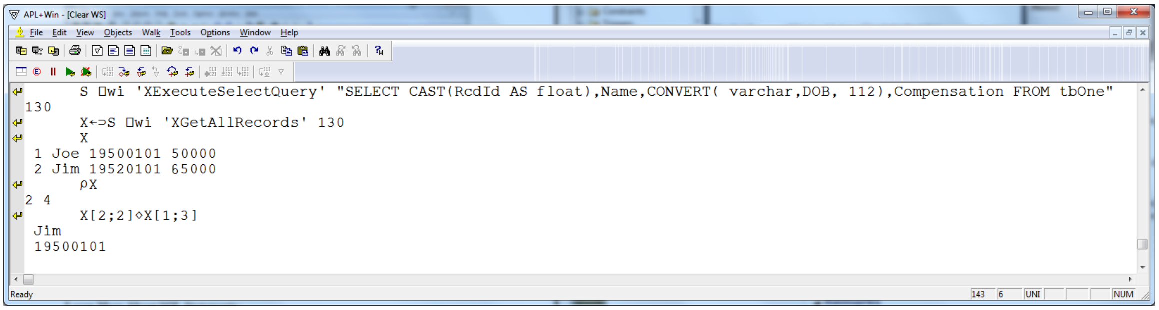
```
      S ⎕wi 'XExecuteSelectQuery' "SELECT CAST( DOB AS nvarchar(50)) FROM tbOne"
115
      ⊃S ⎕wi 'XGetAllRecords' 115
 Jan  1 1950 12:00AM
 Jan  1 1952 12:00AM
```

Alternately cast the DOB column values as varchar ISO style dates (yyyymmdd):



```
      S ⎕wi 'XExecuteSelectQuery' "SELECT CONVERT( varchar,DOB, 112) FROM tbOne"
127
      ⊃S ⎕wi 'XGetAllRecords' 127
 19500101
 19520101
```

Combing these techniques, the SELECT query now returns appropriate APL+Win data type values:



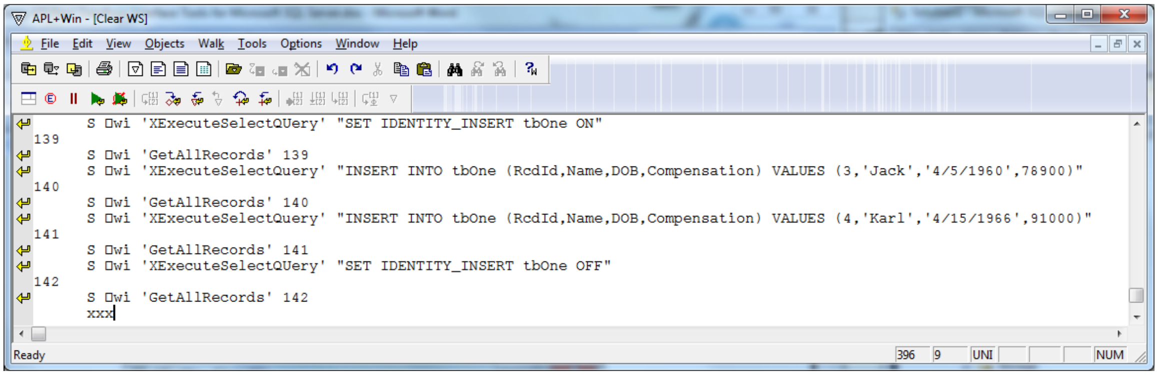
```
      S ⎕wi 'XExecuteSelectQuery' "SELECT CAST(RcdId AS float),Name,CONVERT( varchar,DOB, 112),Compensation FROM tbOne"
130
      X←⊃S ⎕wi 'XGetAllRecords' 130
      X
 1 Joe 19500101 50000
 2 Jim 19520101 65000
      ⍴X
2 4
      X[2;2]⋄X[1;3]
 Jim
 19500101
```

Using the ExecuteSelectQuery or Exec for INSERT Actions:

The ExecuteSelectQuery method can be used to execute any valid SQL statement. The Exec method operates with the same argument and result structure.

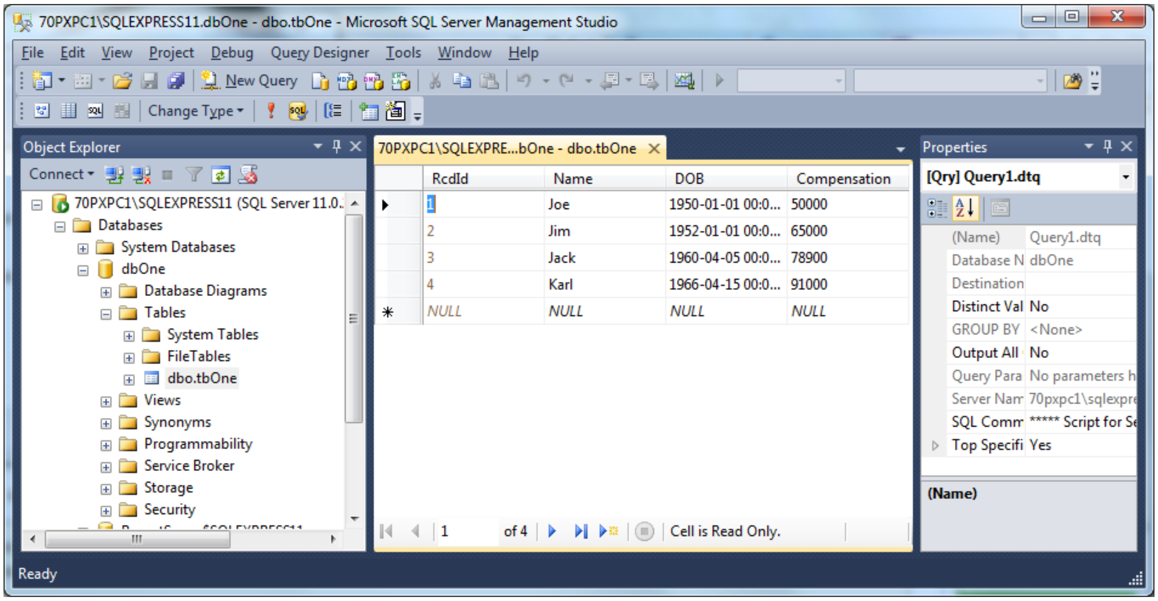
The insertion of data base records to a table which contains an Identity Column (RcdId in the sample) requires that the IDENTITY_INSERT property be ON. It should be set to OFF after completing the INSERT statement as it can be ON for only one table at a time in an SQL data base.

Two records are added to the data base with RcdIds 3 and 4:



```
      S ⎕wi 'XExecuteSelectQUery' "SET IDENTITY_INSERT tbOne ON"
139
      S ⎕wi 'GetAllRecords' 139
      S ⎕wi 'XExecuteSelectQuery' "INSERT INTO tbOne (RcdId,Name,DOB,Compensation) VALUES (3,'Jack','4/5/1960',78900)"
140
      S ⎕wi 'GetAllRecords' 140
      S ⎕wi 'XExecuteSelectQuery' "INSERT INTO tbOne (RcdId,Name,DOB,Compensation) VALUES (4,'Karl','4/15/1966',91000)"
141
      S ⎕wi 'GetAllRecords' 141
      S ⎕wi 'XExecuteSelectQUery' "SET IDENTITY_INSERT tbOne OFF"
142
      S ⎕wi 'GetAllRecords' 142
      xxx
```

The Microsoft SQL Server Management Studio reflects the effectiveness of the INSERT SQL statement:



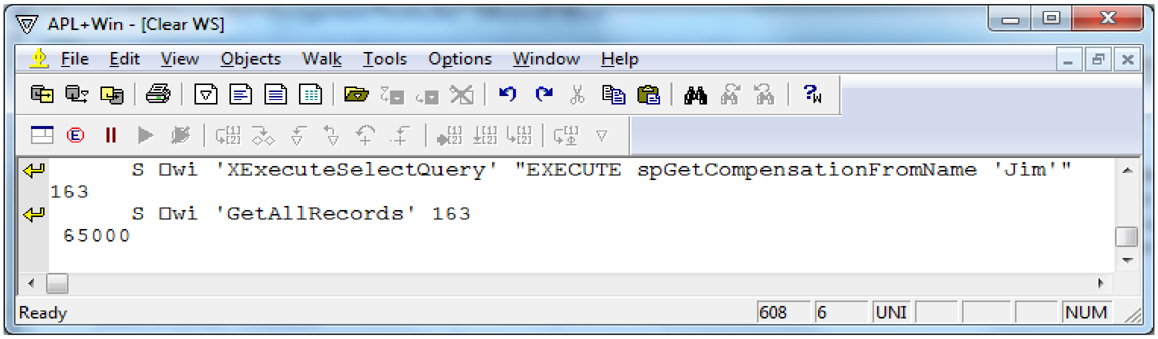
	RcdId	Name	DOB	Compensation
▶	1	Joe	1950-01-01 00:0...	50000
	2	Jim	1952-01-01 00:0...	65000
	3	Jack	1960-04-05 00:0...	78900
	4	Karl	1966-04-15 00:0...	91000
*	NULL	NULL	NULL	NULL

1 of 4 Cell is Read Only.

(Name)	Query1.dtq
Database N	dbOne
Destination	
Distinct Val	No
GROUP BY	<None>
Output All	No
Query Para	No parameters h
Server Nam	70pxpc1\sqlexpre
SQL Comm	***** Script for Se
Top Specifi	Yes

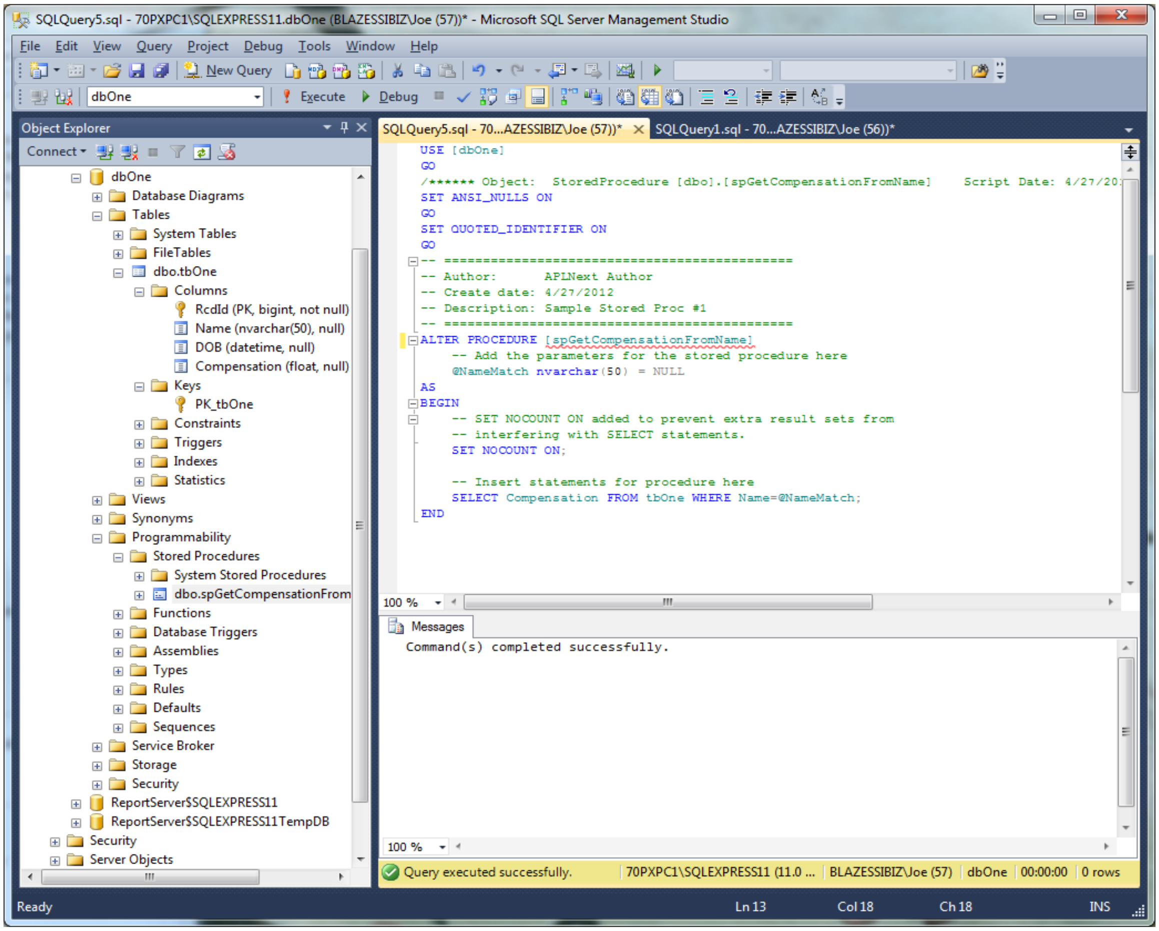
Using the ExecuteSelectQuery for EXECUTE Stored Procedure Action:

If the sample SQL sample data base the ‘spGetCompensationFromName’ stored procedure was implemented. This stored procedure has an argument (NameMatch) which will be used to filter the tbOne table records. The Exec method also has an analogous argument and result structure.



```
      S ⎕wi 'XExecuteSelectQuery' "EXECUTE spGetCompensationFromName 'Jim'"
163
      S ⎕wi 'GetAllRecords' 163
 65000
```

The Microsoft SQL Server Management Studio illustrates this stored procedure:

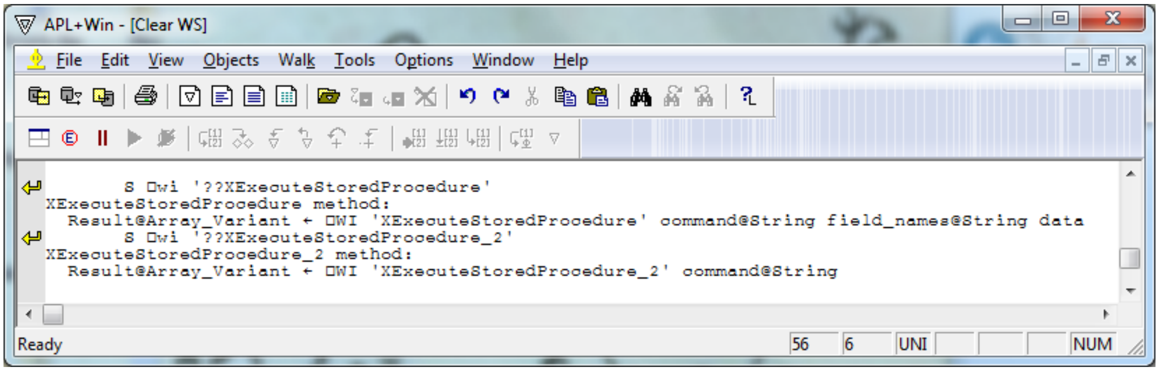


```
USE [dbOne]
GO
/****** Object:  StoredProcedure [dbo].[spGetCompensationFromName]    Script Date: 4/27/20
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:      APLNext Author
-- Create date: 4/27/2012
-- Description: Sample Stored Proc #1
-- =============================================
ALTER PROCEDURE [spGetCompensationFromName]
    -- Add the parameters for the stored procedure here
    @NameMatch nvarchar(50) = NULL
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    SELECT Compensation FROM tbOne WHERE Name=@NameMatch;
END
```

```
Command(s) completed successfully.
```

Using the ExecuteStoredProcedure Method:
Using the ExecuteStoredProcedure_2 Method:

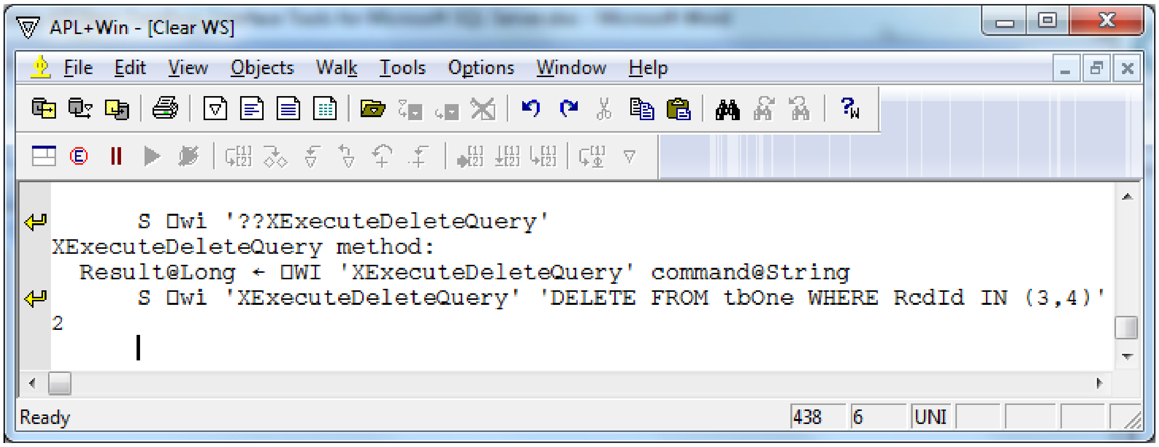
Stored procedures can have ‘out’ and ‘inout’ parameters, in which case these methods can be used. This is beyond the scope of this document.



```
      S ⎕wi '??XExecuteStoredProcedure'
XExecuteStoredProcedure method:
  Result@Array_Variant ← ⎕WI 'XExecuteStoredProcedure' command@String field_names@String data
      S ⎕wi '??XExecuteStoredProcedure_2'
XExecuteStoredProcedure_2 method:
  Result@Array_Variant ← ⎕WI 'XExecuteStoredProcedure_2' command@String
```

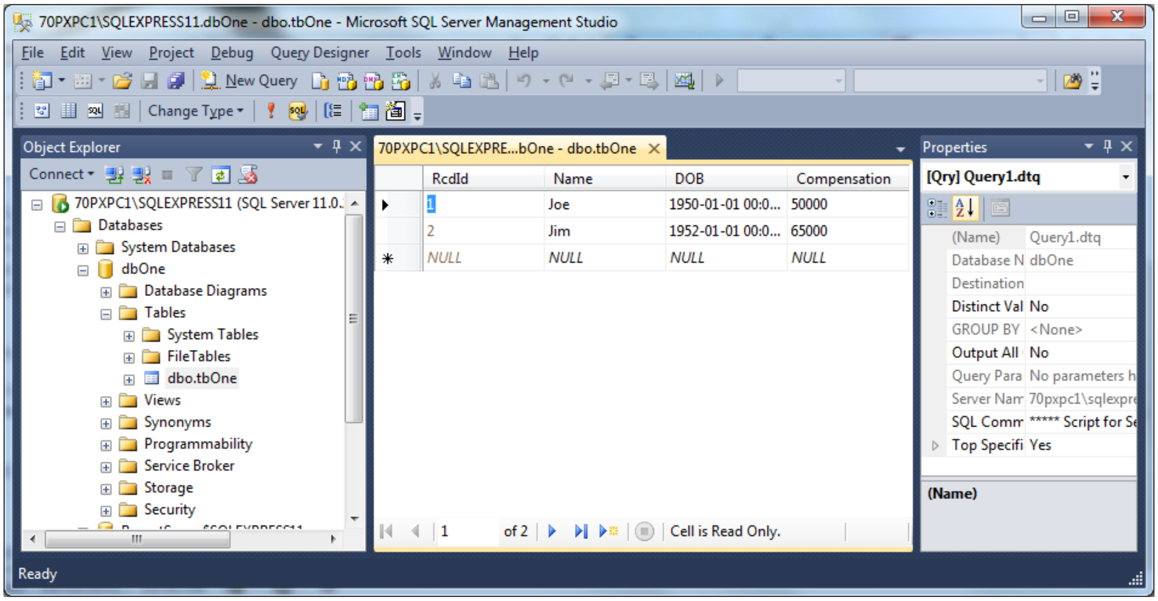
Using the ExecuteDeleteQuery:

In this example the tbOne records with RcdId values 3 or 4 are deleted from the table:



```
      S ⎕wi '??XExecuteDeleteQuery'
XExecuteDeleteQuery method:
  Result@Long ← ⎕WI 'XExecuteDeleteQuery' command@String
      S ⎕wi 'XExecuteDeleteQuery' 'DELETE FROM tbOne WHERE RcdId IN (3,4)'
2
```

The Microsoft SQL Server Management Studio reflects the effectiveness of the DELETE SQL statement:



	RcdId	Name	DOB	Compensation
▶	1	Joe	1950-01-01 00:0...	50000
	2	Jim	1952-01-01 00:0...	65000
✱	NULL	NULL	NULL	NULL

[Qry] Query1.dtq

(Name)	Query1.dtq
Database N	dbOne
Destination	
Distinct Val	No
GROUP BY	<None>
Output All	No
Query Para	No parameters h
Server Nam	70pxpc1\sqlexpre
SQL Comm	***** Script for Se
Top Specifi	Yes

(Name)

Using SQL Transactions:

Transactions are used to group several SQL statements into a block which must be entirely successfully completed to maintain data base integrity.

- The BeginTransaction method initiates the block of processing steps

- Any number of SQL processing methods can be included in the transaction block

- If any error occurs between the BeginTransaction and EndTransaction block, the RollbackTransaction method is called to reverse the 'partially' completed processing block

- If no error occurs between the BeginTransaction and EndTransaction block, the CommitTransaction method is called

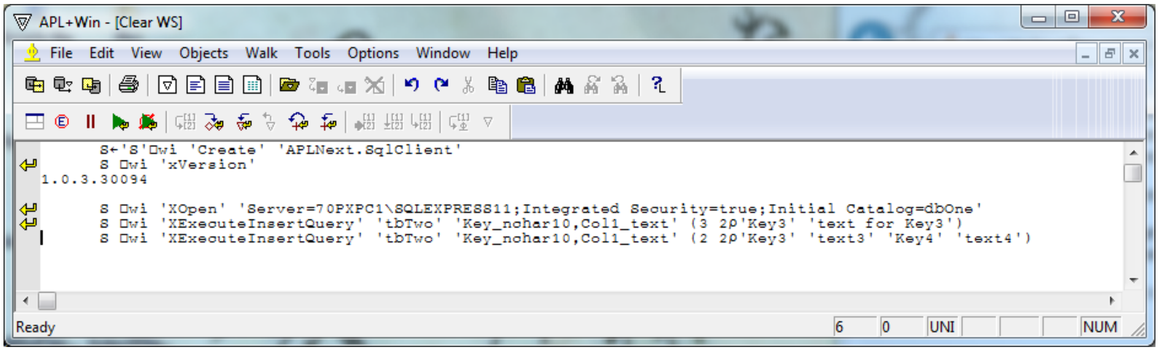
- The APL+Win :TRY, :CATCHALL and :ENDTRY control structures are useful to identify errors which may occur within an SQL transaction block

- The EndTransaction method ends the block of processing steps

Using the ExecuteInsertQuery Method:
The argument syntax for this method is:

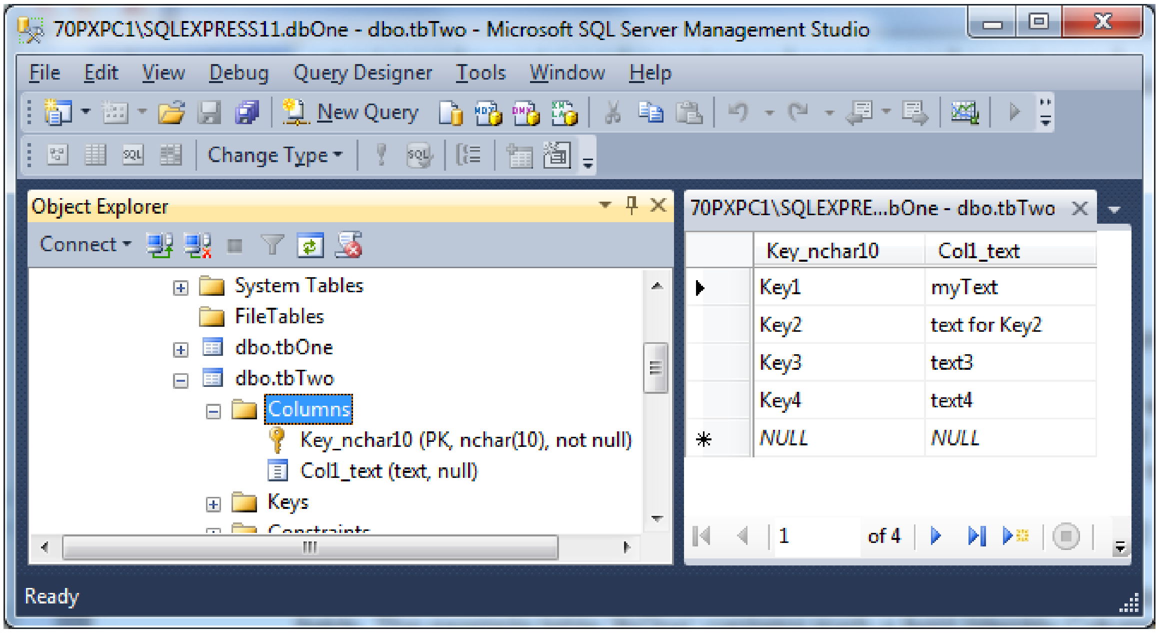
- Argument #1: table name: The name of the database table into which records will be inserted
- Argument #2: fields: A comma-separated string of fields for which values are provided in the records to be inserted. If any field
- Argument #3: data: An array of APL+Win data for the fields selected, with one row for each record to be inserted and columns corresponding to each of the fields specified in Argument #2.

This method can be used only if none of the fields in the table are ‘automatic’ fields. The example table ‘tbOne’ contains such a field (Identity Column) so an Insert SQL statement should use the Exec or ExecuteSelectQuery methods. In the example below an additional table, ‘tbTwo’ was defined in the database which has no ‘automatic’ fields.



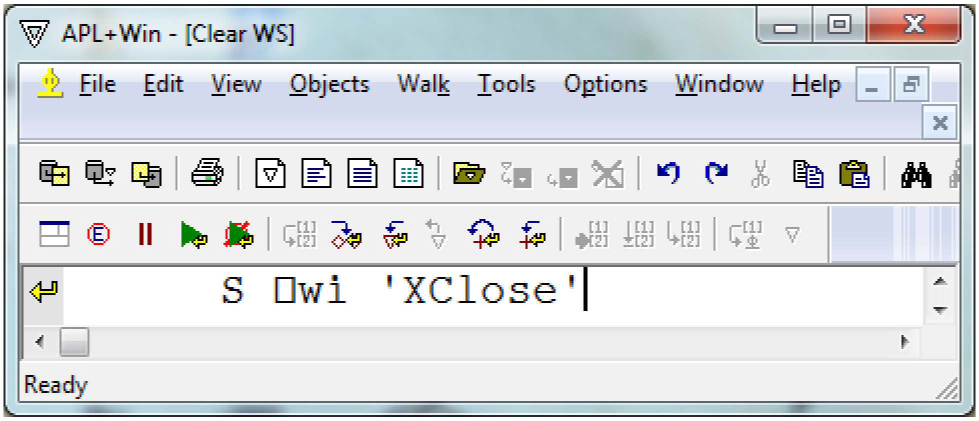
```
      S←'S'⎕wi 'Create' 'APLNext.SqlClient'
      S ⎕wi 'xVersion'
1.0.3.30094

      S ⎕wi 'XOpen' 'Server=70PXPC1\SQLEXPRESS11;Integrated Security=true;Initial Catalog=dbOne'
      S ⎕wi 'XExecuteInsertQuery' 'tbTwo' 'Key_nchar10,Col1_text' (3 2ρ'Key3' 'text for Key3')
      S ⎕wi 'XExecuteInsertQuery' 'tbTwo' 'Key_nchar10,Col1_text' (2 2ρ'Key3' 'text3' 'Key4' 'text4')
```

The Microsoft SQL Server Management Studio reflects the effectiveness of the ExecuteInsertQuery method:


70PXPC1\SQLEXPRESS11.dbOne - dbo.tbTwo - Microsoft SQL Server Management Studio
File Edit View Debug Query Designer Tools Window Help
New Query
Change Type
Object Explorer
Connect
System Tables
FileTables
dbo.tbOne
dbo.tbTwo
Columns
Key_nchar10 (PK, nchar(10), not null)
Col1_text (text, null)
Keys
Constraints
70PXPC1\SQLEXPRE...bOne - dbo.tbTwo
1
of 4
Ready

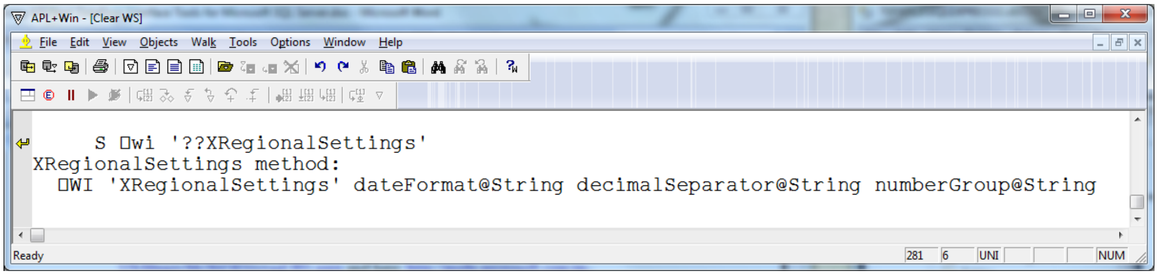
	Key_nchar10	Col1_text
▶	Key1	myText
	Key2	text for Key2
	Key3	text3
	Key4	text4
*	NULL	NULL

Close the Tool to Release Resources:



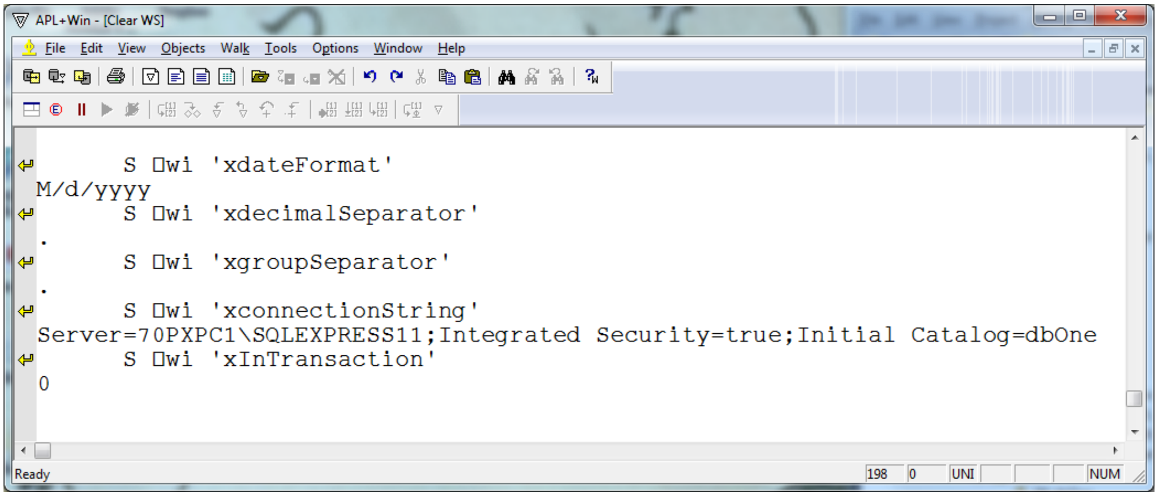
```
      S ⎕wi 'XClose'
```

Modify Regional Settings:



```
      S ⎕wi '??XRegionalSettings'
XRegionalSettings method:
  ⎕WI 'XRegionalSettings' dateFormat@String decimalSeparator@String numberGroup@String
```

Tool Properties:



```
       S ⎕wi 'xdateFormat'
M/d/yyyy
       S ⎕wi 'xdecimalSeparator'
.
       S ⎕wi 'xgroupSeparator'
.
       S ⎕wi 'xconnectionString'
Server=70PXPC1\SQLEXPRESS11;Integrated Security=true;Initial Catalog=dbOne
       S ⎕wi 'xInTransaction'
0
```

Learn More About SQL Statements:
General SQL statement information: http://www.w3schools.com/sql/default.asp
There are variants of the SQL statement language and syntax.
Microsoft SQL Server “Transact SQL” here: http://msdn.microsoft.com/en-US/library/bb264565(v=sql.90).aspx and here: http://msdn.microsoft.com/en-us/library/bb510741.aspx.