

4D v11 SQL Database Access Files (4DLINK)

By Timothy Penner, Technical Services Team Member, 4D Inc.

Technical Note 08-30

Table of Contents

Table of Contents	2
Abstract	3
Introduction	3
What are 4DLINK files?	3
Why Use Them?.....	3
How does it work?	4
Creation of Files.....	4
Using 4DLINK Files.....	4
Anatomy of the 4DLINK file	5
Attributes for all 4DLINK Files	5
Attributes for Remote 4DLINK Files	5
Attributes for Local 4DLINK Files	6
4DLINK Examples	7
Example Database	9
Conclusion	11
Related Resources	11

Abstract

4D v11 SQL features a new file type called “database access file” or simply, 4DLINK.

This new file type allows 4D Developers to control the way in which 4D databases are launched, automatically log in users, automatically create a data file, etc. This feature goes beyond anything seen in both Path Documents and the CLI found in previous versions of 4D.

This Technical Note outlines how these files can be created and used.

A sample database for creating 4DLINK files is included.

Introduction

4D v11 SQL allows you to generate database access files containing parameters intended to automate and simplify opening or connecting to 4D databases. 4D has offered alternate ways of connecting to databases for some time now. Although the PTH files that were used to save 4D Server connection info are no longer supported, the command line interface (CLI) is still supported. However, the functionality of both has been rolled together into an XML file suffixed .4DLINK. 4DLINK files are much more powerful and flexible than the previous methods available.

What are 4DLINK files?

4DLINK files are XML files. 4DLINK files can save the address of a remote database as well as the connection identifiers, thus eliminating several operations for the user.

These access files can be used either to open a local database or to connect to a database published by 4D Server.

Note *In previous versions of 4D, this function was limited to saving access to 4D Server databases. It was available via a button in the 4D Client identification dialog box. The files generated had the .pth extension. These files are no longer supported by 4D v11 SQL; the functionality has been replaced by XML files suffixed .4DLINK.*

Why Use Them?

There are many reasons that make the use of 4DLINK files very practical. Simply put, 4DLINK files are available to save time when connecting to databases. 4DLINK files can include the default user credentials for either remote or local databases. A 4DLINK file can be used to launch a local database in MSC mode. The OnStartup method (of local databases) can be bypassed (skipped) via an attribute in the 4DLINK file. These are just some of the features available.

How does it work?

The database access files of 4D v11 SQL are XML files with the extension ".4DLINK".

Creation of Files

4DLINK files can be created using any text editor or the built-in XML commands in 4D. Furthermore 4D automatically creates database access files in order to populate the "recent database" list.

The 4DLINK files that are created automatically by 4D are placed in the local preferences folder of the user. In this folder, two directories are created: Local and Remote. The Local folder contains the ".4DLINK" files that can be used to connect to local databases and the Remote folder contains the ".4DLINK" files that can be used to connect to remote databases.

Local preferences folders in 4D v11 SQL Release 2 are found at the following locations:

- *Windows Vista*: C:\Users\UserName\AppData\Roaming\4D\Favorites v11\
- *Windows XP*: C:\Documents and Settings\UserName\Application Data\4D\Favorites v11\
- *Mac OS*: Users/UserName/Library/Preferences/4D/Favorites v11/

The files found in these directories are displayed by 4D in the Open Recent Databases submenu of the File menu.

The ".4DLINK" files can also be created with an XML editor and contain customized information such as the connection identifiers or the database opening mode.

4D provides a DTD describing the XML key and attributes that can be used to build a ".4DLINK" file. This DTD is named *database_link.dtd* and is found in the \Resources\DTD\ subfolder of the 4D application. A copy of this file, from 4D v11 SQL Release 2, is also included with this Technical Note.

Using 4DLINK Files

A .4DLINK access file can be used to launch the 4D application and open the target 4D database. There are three different ways to use it:

- Via a double-click or drag and drop onto the 4D application
- Via selection in the 4D open dialog box
- Via the Open Recent Databases submenu (file located in the local preferences folder)

Note: *4DLINK files of the "remote database" type can be copied and used on several different machines, without modification. Local 4DLINK files may need to be modified to use the correct paths.*

Anatomy of the 4DLINK file

The 4DLINK files are relatively simple and small in size. They consist of two lines. Here is an example:

```
<?xml version="1.0" encoding="UTF-8"?>
<database_shortcut structure_file="file:///C:/my.4DB"
data_file="file:///C:/my.4DD"/>
```

The first line identifies the file as valid XML, encoded in UTF-8. The second line is where all the magic happens. There is one single tag named <database_shortcut> with many acceptable attributes. Here is a list of the accepted attributes:

Attributes for all 4DLINK Files

Attribute	Description	Accepted Values	Default
is_remote	tells 4D whether this is a remote or local database	true, false	false
user_name	user name	text	""
password	password	text	""
md5_password	encrypted password	text	""
structure_opening_mode	0 -> normal 1 -> interpreted 2 -> compiled	0, 1, 2	0

Attributes for Remote 4DLINK Files

Attribute	Description	Accepted Values	Default
server_database_name	server database name without extension	text	""
server_path	could be IP address or DNS name	text	""
open_login_dialog	tells 4D to open the login dialog	true, false	false

Attributes for Local 4DLINK Files

Attribute	Description	Accepted Values	Default
open_in_custom_mode	open database in custom mode	true, false	false
open_tools	open database in MSC	true, false	false
create_structure_file	tells 4D to create structure file	true, false	false
structure_file	path to the structure file that should be used	text	""
create_data_file	tells 4D to create data file	true, false	false
data_file	path to the data file that should be used	text	""
skip_onstartup_method	tells 4D to skip the OnStartup method	true, false	false
definition_import_file	path to XML Structure Definition file	text	""
resources_import_file	internal use only		
data_opening_mode	1 -> default data file 2 -> select other data file 3 -> create new data file	1, 2, 3	1

Please note that all attributes are case sensitive, as well as the "true" and "false" values.

4DLINK Examples

Here are some example 4DLINK files:

- The following example will connect to the remote database named "Billing" located on server named "billings.company.com" using the username "Bob" and password "123". The Login Dialog will not be displayed:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<database_shortcut is_remote="true" password="123"
server_database_name="Billing" server_path="billings.company.com"
user_name="Bob"/>
```

The attributes were set as follows:

is_remote	true
user_name	Bob
password	123
server_database_name	Billing
server_path	billings.company.com

- The following example will create a local database structure saved at "file:///C:/temp/my.4db" and use the structure definition "file:///C:/4D/4DMeetings.4DB.xml" to create the structure:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<database_shortcut create_structure_file="true"
definition_import_file="file:///C:/4D/4DMeetings.4DB.xml"
structure_file="file:///C:/temp/my.4db"/>
```

The attributes were set as follows:

create_structure_file	true
definition_import_file	file:///C:/4D/4DMeetings.4DB.xml
structure_file	file:///C:/temp/my.4db

Note: For more info on XML Structure, see Technical Note 08-26, "XML Structure Import/Export".

- The following example opens the local database using the "file:///C:/4D/Test DB's/contacts/Contacts final/Contacts.4DB" structure file and the "file:///C:/4D/Test DB's/contacts/Contacts final/Contacts.4DD" data file in the Maintenance and Security Center (MSC):

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<database_shortcut data_file="file:///C:/4D/Test DB's/contacts/Contacts
final/Contacts.4DD" open_tools="true" structure_file="file:///C:/4D/Test
DB's/contacts/Contacts final/Contacts.4DB"/>
```

The attributes were set as follows:

```
data_file           file:///C:/4D/Test DB's/contacts/Contacts
                    final/Contacts.4DD
structure_file      file:///C:/4D/Test DB's/contacts/Contacts
                    final/Contacts.4DB
open_tools          true
```

- The following example opens the local database using the "file:///C:/4D/Test DB's/contacts/Contacts final/Contacts.4DB" structure file and the "file:///C:/4D/Test DB's/contacts/Contacts final/Contacts.4DD" data file and skips the onStartUp method:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<database_shortcut data_file="file:///C:/4D/Test DB's/contacts/Contacts
final/Contacts.4DD" skip_onstartup_method="true"
structure_file="file:///C:/4D/Test DB's/contacts/Contacts
final/Contacts.4DB"/>
```

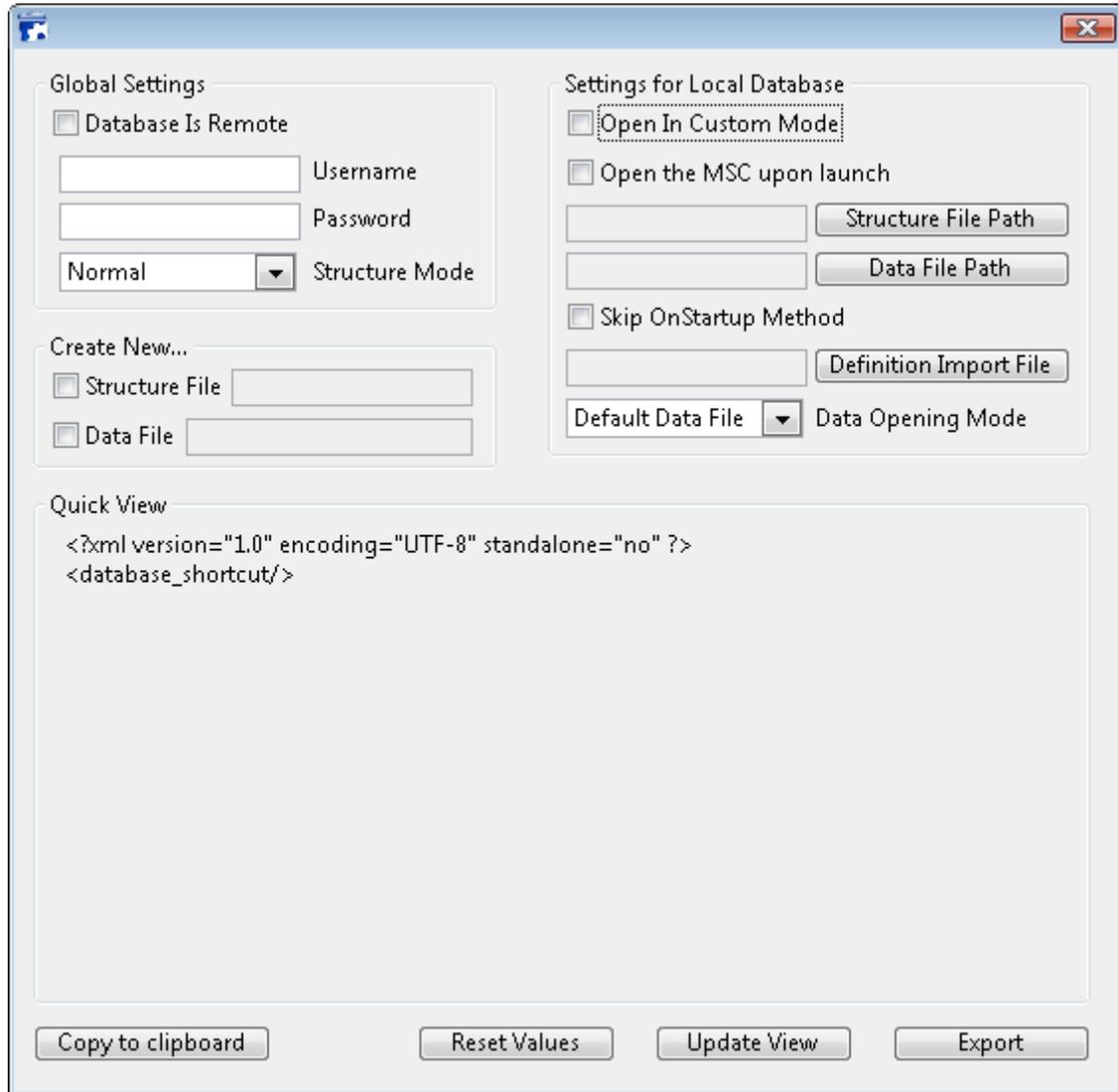
The attributes were set as follows:

```
data_file           file:///C:/4D/Test DB's/contacts/Contacts
                    final/Contacts.4DD
structure_file      file:///C:/4D/Test DB's/contacts/Contacts
                    final/Contacts.4DB
skip_onstartup_method true
```

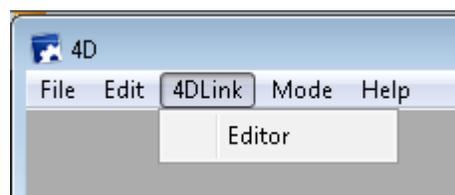
Example Database

The example database included with this Technical Note helps you create 4DLINK files. The database is interpreted, so you can view the code behind the forms.

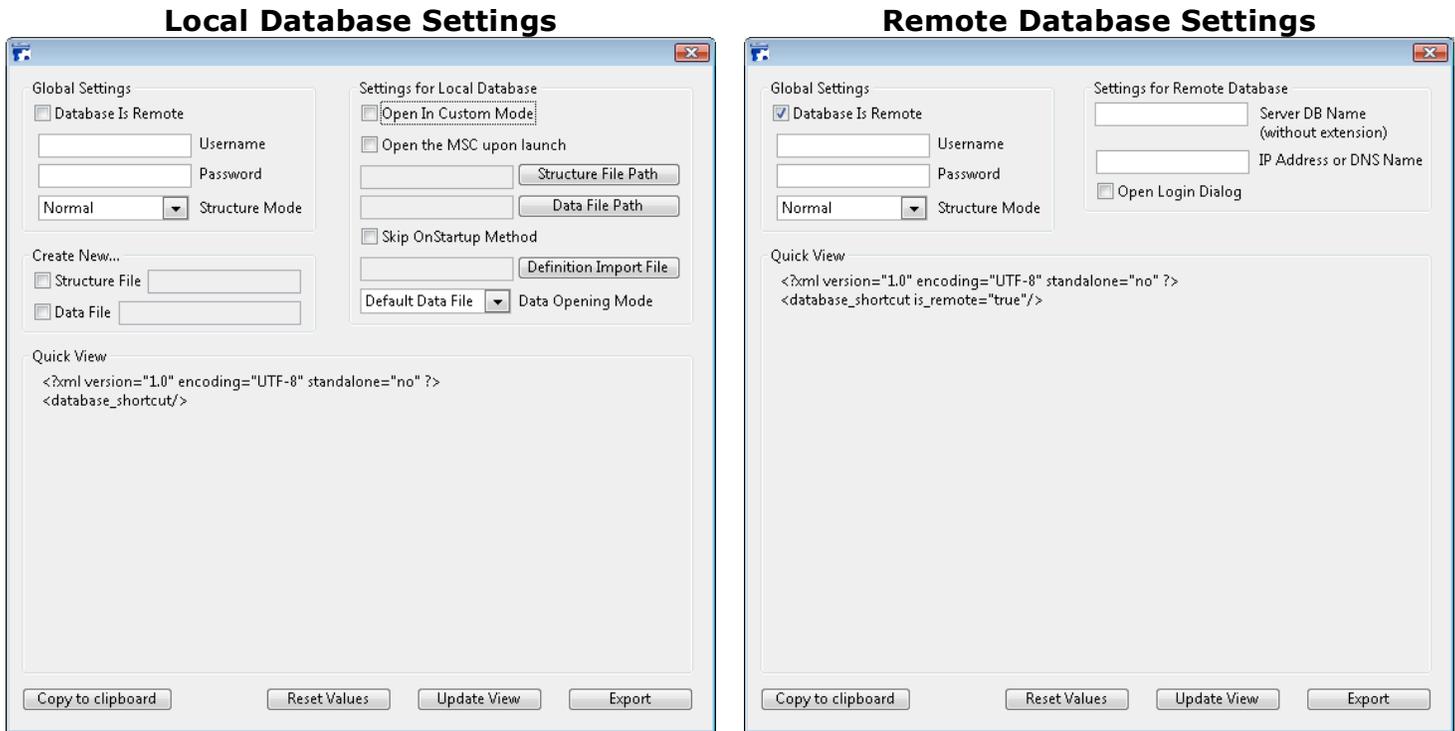
When the database is launched, it starts up in the Application environment. The 4DLINK Editor window is opened automatically upon startup of the database:



You can also open the 4DLINK Editor by choosing "Editor" from the 4DLINK drop down menu:



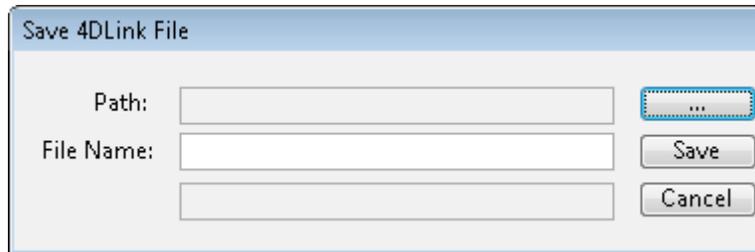
There is a different page for Remote and Local settings, as both would never be used together:



Clicking the "Update View" button updates the information presented in the "Quick View". You can reset the entire form by pressing the "Reset Values" button.

Clicking the "Copy to clipboard" button copies the contents of the Quick View area to the clipboard.

Clicking on the Export button brings up a save dialog, where you can select a folder to save the file in, and a file name:



Clicking the "..." button brings up the "Choose folder" window. Type in the name you would like it saved as into the "File Name" field. The 4DLINK extension is automatically added when you click "Save". Clicking Cancel brings you back to the editor.

Conclusion

This Technical Note described the general concepts of using the new 4DLINK files for saving database specific connection settings. This information should allow the 4D developer to write their own 4DLINK files and make reconnecting to databases even easier.

Related Resources

Technical Note 08-26: "XML Structure Import/Export in 4D v11 SQL":

<http://www.4d.com/knowledgebase?CaseID=50517>