4D Ajax Framework v11 Install & Upgrade

The 4D Ajax Framework v11 component can either be installed into a 4D v11 SQL database or upgraded into a database with a previous version of the component. Thus, the 4DAF section of this guide is split into installation or upgrading.

- o Installing the 4D Ajax Framework v11 component in a 4D V11 SQL database
- o Upgrading the 4D Ajax Framework v11 component in a 4D V11 SQL database

Cal4D Component Installation

The final section of this guide involves instructions for Cal4D, which is a new Calendar component introduced as part of the 4D Ajax Framework v11 Release 5 (11.5).

4D Ajax Framework Install

> **Note:** Due to the Unicode support in 4D V11 SQL, there are two versions of the 4D Ajax Framework component. One has Unicode mode turned on and the other has it turned off. You must be sure to use the proper version for your database. Databases created in 4D V11 SQL will always have Unicode mode turned on and this cannot be modified. Databases upgraded from a previous version will default to having Unicode mode turned off. This can be turned On in the 4D Preferences under Compatibility.

Installing the Component

1. If one does not already exist, create a folder named "Components" next to your structure file.
2. Copy "4DAF v11.1 Unicode.4dbase" (from the installer's "For 4D v11 SQL" folder) into your new "Components" folder.

Installing the Plugins Folder

4D Ajax Framework makes use of the 4D Pack plugin when serving its web pages. If you do not currently have a "Plugins" folder next to your structure, create this folder and copy the 4D Pack plugin into it.

> **Note:** You can find the 4D Pack plugin in the latest version of your 4D installation. It is no longer bundled with the 4D Ajax Framework.

Installing the Web Pages

1. Copy the "Webfolder" folder next to your structure file

> **Note:** If "Webfolder" already exists next to your structure file be sure to copy the *contents* of the component's 'Webfolder' into your database's 'Webfolder'.

2. Launch the database with 4D V11 SQL.

Configuring 4D to Use the 4D Ajax Framework

Once installation of the component and files is complete, launch your database with 4D. You will now need to make some modifications to the Database Methods.
1. Open the Database method On Startup and add a call to *DAX_Dev_Initialize*.
2. Open the Database method On Server Startup and add a call to *DAX_Dev_Initialize*.
3. Open the Database method On Exit and add a call to *DAX_Dev_Shutdown*.
4. Open the Database method On Server Shutdown and add a call to *DAX_Dev_Shutdown*.
5. Open the Database method On Web Connection and use the "*Dax On Web Connection*" macro as shown below. This macro was installed with the 4DAF component.
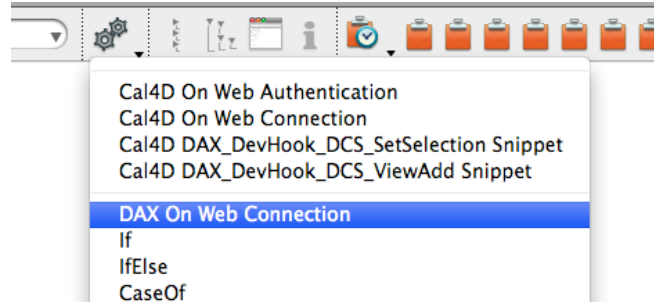
It will enter code like so:

6. Create a new project method named "Compiler_Web" (if you don't currently have one). Place a call to *DAX_Dev_CompilerWeb* in your Compiler_Web method.

> If your 4D database is not currently serving web pages, follow these additional steps. If you are adding the 4D Ajax Framework to an existing 4D web server, these options should already be correct.

7. Open the Database Preferences.
       On Windows, select Preferences... from the Edit menu.
       On Mac, select Preferences... from the Application menu.
8. Select Web -> Configuration.
9. Check "Publish Database at Startup".
10. Change the "TCP Port" for the Web Server if required.
11. Set the "Default HTML Root" to "Webfolder"
12. Set the "Default Home Page" to "index.html"
13. Set "Starting Mode" to "Non-contextual Mode"
14. Click the "OK" button to save your changes.
15. Quit and relaunch your database to start using 4D Ajax Framework.

4D Ajax Framework v11 requires at least 4D v11 SQL Release 5 (11.5)

Make sure that you are using at least 4D v11 SQL Release 5 (11.5), as it requires so to function properly.

4D Ajax Framework Upgrade

Upgrading to the latest version of 4D Ajax Framework v11 to run on 4D v11 SQL from a database previously running on 4D v11 SQL with an older component

Overview

A successful upgrade requires the following:
- Move the "Support" folder to the Resources folder (only for older versions of 4DAF)
- Update the 4D Ajax Framework Component
- Update the 4D Ajax Framework Library and Client
- Update the 4D Pack plugin
- Activate your Update

Move the "Support" Folder to Resources Folder

The "Support" Folder is found in the Extras folder within the 4DAF v11 component to the Resources folder next to the Host database.

Mac users must right-click the component and select "Show Package Contents" to navigate within it.

Note that this step only has to occur once. Upgrades that involve the 4DAF Release 2 (v11.2) or later only require a two-step upgrade process:

1. Replace the Component in the Components folder with the latest one.
2. Update the Webfolder with contents from the latest component version.

Update the 4D Ajax Framework Component

1. Delete your existing 4DAF v11 component in the Components folder.
2. Drop the new 4DAF v11 component into the Components folder.

Update the 4D Ajax Framework Library and Client

Open the "Webfolder" folder inside the component in the "For 4D v11 SQL" folder.
1. Copy the contents of the component's Webfolder into your own Webfolder.
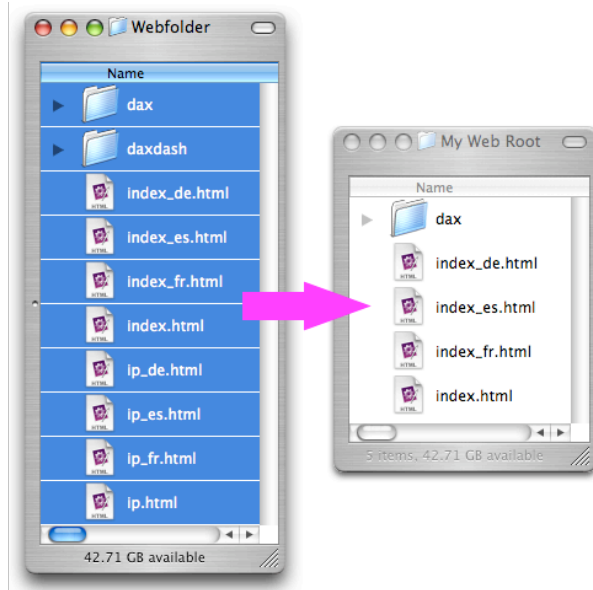
Copy the contents of the "Webfolder"...   ... into your web root folder.

In the image above "My Web Root" denotes the name of your default HTML root folder.
You can find the name of your default HTML root folder in 4D -> Preferences -> Web.

Update the 4D Pack Plugin:

The 4D Ajax Framework makes use of the 4D Pack plugin when serving its web pages. To update it, copy the latest 4D Pack plugin into it into your existing "Plugins" folder next to your structure.

Note: You can find the 4D Pack plugin in the latest version of your 4D installation. It is no longer bundled with the 4D Ajax Framework.

Activate your update:

1. Launch the database.
2. You may be prompted to update the structure XML files. Click "Yes".
3. Turn on Unicode mode in Preferences -> Compatibility, if necessary.

Cal4D Component Installation

Cal4D is a new component introduced as part of the 4D Ajax Framework v11 Release 5 (11.5).

If you are intending to install the Cal4D.v11SQL component into a host database that already contains tables named "Cal4D_User", "Cal4D_Group", "Cal4D_Event", "CalSRVR_Subscription", or "CalSRVR_Account" you will need to rename the existing tables. These tables are created by the component upon installation and their format is important for the component's ability to subscribe to and publish valid iCalendar formatted files.

This component needs to install these tables so they are complete. The Cal4D Manual includes instructions on a Data Migration Assistant that allows users to move any existing calendar ready data into the new Cal4D tables.

The Cal4D component can only be used with 4D v11 SQL Release 5 (11.5) or newer versions of 4D.

Cal4D Component Installation Instructions

These steps assume that the initial install of the 4D Ajax Framework v11 SQL Release 5 (11.5) has been completed. If you have not yet installed the 4DAF component and peripherals, do so and then return to installing the Cal4D component. The Cal4D component can only be installed in databases running in Unicode mode. It is also important that the first time the Cal4D component is installed that you are running 4D in local mode, not Client/Server. Once this is setup proceed to the following steps:

1   In the host database, place the Cal4D, 4D SVG, and Cal4D Navbar components in the Components folder. The 4D SVG component is not included with the 4D Web 2.0 Pack, but is in the installation of 4D v11 SQL Release 5 (11.5). Once the components are installed restart the host database.

2   In the host database Plugins folder place the 4D v11 SQL Release 5 (11.5) versions of 4D Pack and 4D InternetCommands plug-ins.

3   In the database methods "On Startup" and "On Server Startup" add a call to the Cal4D component method "Cal4D_Startup". Make sure that a call has been made to "DAX_Dev_WaitForInitialization" before the "Cal4D_Startup" call. Cal4D_Startup requires two arguments and has a third optional argument:

1.   The first parameter, a Boolean, stipulates if calendar serving (CalSRVR) and support for the web front end to Cal4D will be active. Pass true to support calendar serving and web support, false to not support them.

2.   The second parameter is the database login name of the current user. Normally using the 4D command "Current User" will suffice.

3.   The third, parameter is optional and is to support a custom login system. It is a pointer to a text array that contains a list of the database login names. Cal4D assumes that the 4D password system is being used if this parameter is not passed and will call the 4D command GET USER LIST.

Here is a sample of what will be in your On Startup database method:

```
DAX_Dev_Initialize
DAX_Dev_WaitForInitialization
Cal4D_Startup (True;Current user)
```

2   In the database methods "On Exit" and "On Server Shutdown" place a call to the Cal4D method "Cal4D_Quit".

3   From the File menu select "Close Database". This will initiate the process of installing the necessary tables and methods. 4D will automatically reopen the database. At this point Cal4D will do four things:

     1.   Two configuration dialogs are presented. From these two dialogs enter your calendar user name, current time zone, and desired week start day as prompted.

     2.   The tables Cal4D_User, Cal4D_Group, and Cal4D_Event will automatically be added to the host database.   If you also added web support, two additional tables will be added that support Calendar Subscriptions, CalSRVR_Account and CalSRVR_Subscription.

     3.   A number of Macros that will be installed to assist with the installation process.

     4.   In the host database a method named "DAXCAL_AddFauxView" will be created.

4   To continue the installation process, open the database method "On Web Authentication" and from the Macros Menu select the Cal4D On Web Authentication macro. The code will be installed into the method, as shown below:

```
C_BOOLEAN($0;$acceptConnection_b)
C_TEXT($url_t;$1)
C_TEXT($header_t;$2)
C_TEXT($clientIP_t;$3)
C_TEXT($serverIP_t;$4)
C_TEXT($user_t;$5)
C_TEXT($pass_t;$6)

$url_t:=$1
$header_t:=$2
$clientIP_t:=$3
$serverIP_t:=$4
$user_t:=$5
$pass_t:=$6

$acceptConnection_b:=CalSRVR_OnWebAuth
$0:=$acceptConnection_b
```

5   Go to the database method "On Web Connection". As part of the 4D Ajax Framework installation you should have already used the Macro "DAX On Web Connection" which installed the necessary code for recognizing and processing a DAX command. Place the cursor where the second statement in the "Case of" statement is to be inserted, above the "ELSE", and from the Macros Menu select Cal4D On Web connection. The second case shown below will be inserted.

```
Case of
  : (DAX_Dev_IsDAXCommand ($url_t))
    DAX_Dev_OnWebConn
($url_t;$header_t;$clientIP_t;$serverIP_t;$user_t;$pass_t)
```

```
    : (CalSRVR_IsOnWebConnection
($URL_T;$header_t;$clientIP_t;$serverIP_t;$user_t;$pass_t))
End case
```

6   In the method "DAX_DevHook_DCS_ViewAdd" add a call to the method
     "DAXCAL_AddFauxView" by selecting the Cal4D DAX_DevHook_DCS_ViewAdd Snippet
     from the Macros Menu.

7   In the method "DAX_DevHook_DCS_SetSelection" insert into the Case of block the code
     shown below by using the Cal4D DAX_DevHook_DCS_SetSelection Snippet.

```
C_TEXT($1;$viewName_t)
$viewName_t:=$1

Case of
  : ($viewName_t="DAXCAL_Tasker")
    C_TEXT($value_T)
    $value_T:=DAX_Dev_GetWebVar ("DAXCAL")

    ARRAY TEXT(CustomVarName_aT;0)
    ARRAY TEXT(CustomVarValue_aT;0)
    DAX_Cal_JSON_Tasker ($value_t;->CustomVarName_aT;->CustomVarValue_aT)
    DAX_Dev_SetCustomVariables (->CustomVarName_aT;->CustomVarValue_aT)
    DAX_Dev_DCS_SetSelection (->CustomVarName_aT;->CustomVarValue_aT)

  : ($viewName_t="Customer Invoices")
...
End case
```

8   Create a Project method and add a call to "Cal4D_Cal_CalendarWithMinis". This will
     launch the self-contained calendar with minis, a list of installed calendars, and standard
     options. Use of this Calendar view is handled in the Cal4D Manual.