

4D Ajax Framework

Cal4D Guide
Windows/Mac OS



The Cal4D Product	3
Introduction	3
Features and Strategies.....	3
Usage and Requirements	3
Managing the Cal4D Component.....	4
Calendar Views	4
Displaying a Calendar	4
Calendar Views	6
Navigating Between Views	8
The Action Menu	9
Managing Calendars	13
The Manage Calendars Dialog	13
Managing Calendar Users.....	16
Managing the Calendar Server	17
Data Migration	18
Migrating Data into Cal4D.....	19
Cal4D Data Migration Assistant - Beta	19
Cal4D Component for Developers	22
Structure Definition	22
Time Zone Resource File.....	25
Language Reference	25
Cal4D HTML Pages for Developers	27
Creating the Calendar Object	27
Calendar methods.....	29
Installation and use	31
Examples	32
Further References	32

The Cal4D Product

Introduction

In order to fulfill most of our daily activities, we all use some type of organizing system. It can simply be an organizer in ones wallet, some "stickies" or Post-it notes that get placed on a desk at work, or even a secretary that provides verbal reminders. Because our days are so busy and hectic we rely on these things to help up get through our daily activities. Fortunately, the 4D community can now ease the pain of their busy customers by providing a calendar with their 4D applications.

Cal4D is a package which allows a 4D developer to add a full-featured calendar into any 4D v11 SQL Release 5 (11.5) or later application. The Cal4D v11.5 Unicode component supplied with 4D Web 2.0 Pack v11 Release 11.5 is exclusive to the 4D v11 SQL product line and cannot be used with previous versions of 4D. It is packaged as a 4D v11 SQL component and supports multiple calendars, taking advantage of many 4D v11 SQL features.

Features and Strategies

- Cal4D has been written to use the SVG renderer in 4D v11 SQL to display calendars.
- Cal4D uses the 4D SVG component to build all of the on screen graphics of the calendar. The 4D SVG component must be installed in the components folder of the host database.
- Cal4D uses 4D's SQL commands to create the five tables necessary in the host database to store calendar events, if the tables do not previously exist. The tables are named Cal4D_User, Cal4D_Group, Cal4D_Event, CalSRVR_Account, and CalSRVR_Subscription.
- Cal4D will display single events and banners similar to iCal™, Outlook™, Mozilla Sunbird™, and Google Calendar applications.
- Cal4D supports "Mini-Calendars" as an integral part of its "Calendar with Minis" window and as independent screen objects
- Cal4D supports pop-up calendars that can be used to set the date for any Date Field or Date Variable.
- Cal4D supports Time Zones, and applies time zone offsets to events being displayed.
- Cal4D supports the publishing of calendars for sharing with iCal™, Outlook™, Mozilla Sunbird™, and Google Calendar applications. It uses the iCalendar specification.
- There is no "Inherited Form" to install with Cal4D. Cal4D forms are built into the component and displayed from the component.

Usage and Requirements

The Cal4D product includes the following:

- A compiled version of the Cal4D component for Unicode databases.
- The Cal4D matrix database which contains the source code for the Cal4D component.
- A compiled version of the Cal4D Navbar component for Unicode databases.
- The Cal4D Navbar matrix database which contains the source code for the Cal4D Navbar component.
- A User Constants plug-in for running the matrix databases interpreted.
- Calendar HTML pages which are included in the 4D Ajax Framework Webfolder.

Note that the component is provided "open source". The developer is free to modify the source in the matrix databases as needed. Installation instructions are included in the *4D Ajax Framework - Installation and Upgrade Guide version 11.5*. Using version 11.5 of Cal4D requires:

- 4D v11 SQL Release 5 (11.5) or later
- The Cal4D 11.5 component
- The Cal4D Navbar 11.5 component
- The 4D SVG 11.5 component,
- The 4D Ajax Framework v11 Release 5 (11.5) component
- 4D InternetCommands plug-in
- 4D Pack plug-in
- User Constants plug-in (only needed when running Cal4D component or matrix database interpreted)

Managing the Cal4D Component

All the tools necessary to manage calendars are built into three methods:

- Cal4D_Cal_CalendarWithMinis
- Cal4D_User_Edit
- CalSRVR_Edit_Subscribers.

It is strongly recommended calendar events not be edited in a Cal4D_Event table form! It is very easy to get the fields out of balance if edited by hand. Great care has been taken in the tools contained within Cal4D to properly maintain events so that they are displayed correctly in the component window and web interface, and produce valid iCalendar files for sharing and exporting. Manually editing events in the table can invalidate them for the iCalendar specification and should only be done by developers who have familiarized themselves with the iCalendar requirements.

Cal4D_Cal_CalendarWithMinis opens a window that contains a list of all calendars that belong to the current user and those owned by other users but are marked as “Others Can View”. It contains three main calendar views (Month, Week, and Day) which provide the user with the ability to view and manage calendar events. For quick view and navigation from month to month there are three “Mini-Calendars” which display the month currently shown in the larger calendar and the previous and next months as well. An “Action Menu” is included that supports related calendar tasks not supported by direct interaction with one of the calendar views.

Cal4D_User_Edit opens a window that allows the database Designer or Administrator to manage the list of calendar users.

CalSRVR_Edit_Subscribers opens a window that allows the database Designer or Administrator to manage the list of subscribers to restricted published calendars.

Calendar Views

Displaying a Calendar

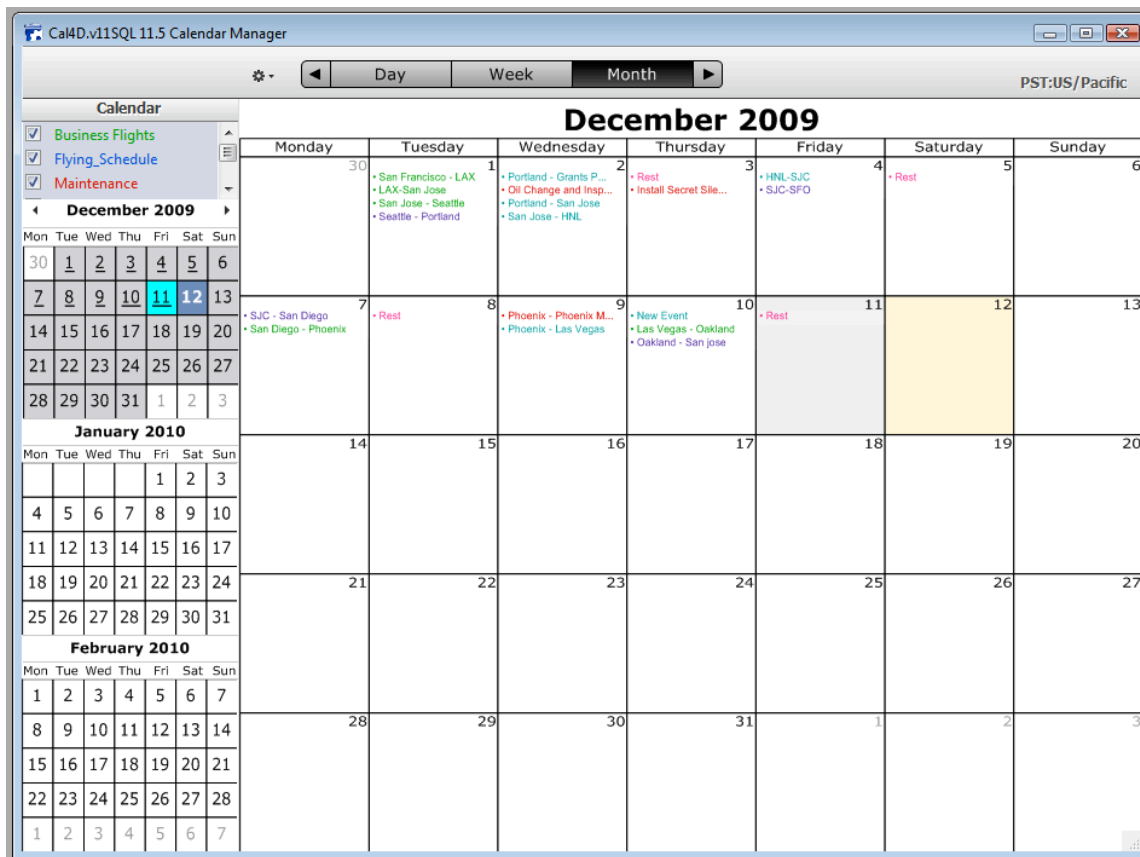
In 4D the Cal4D calendar operates in its own process. The developer must create a method in the host database that calls the component method **Cal4D_Cal_CalendarWithMinis** as shown here:

```
Cal4D_Cal_CalendarWithMinis (WindowTitle_T)
```

The text parameter is an optional parameter for the developer to designate the string that will be used as the Window Title. The component contains two variables for the developers use in accessing the calendar window and process:

- <>CAL4D_FullAndMinisProcName_T: contains the process name of the current CalendarWithMinis process.
- <>Cal4D_CalendarWindowRef_L: contains the window reference value of the window opened.

The Cal4D_Cal_CalendarWithMinis command will open a window displaying the calendar, as shown here:



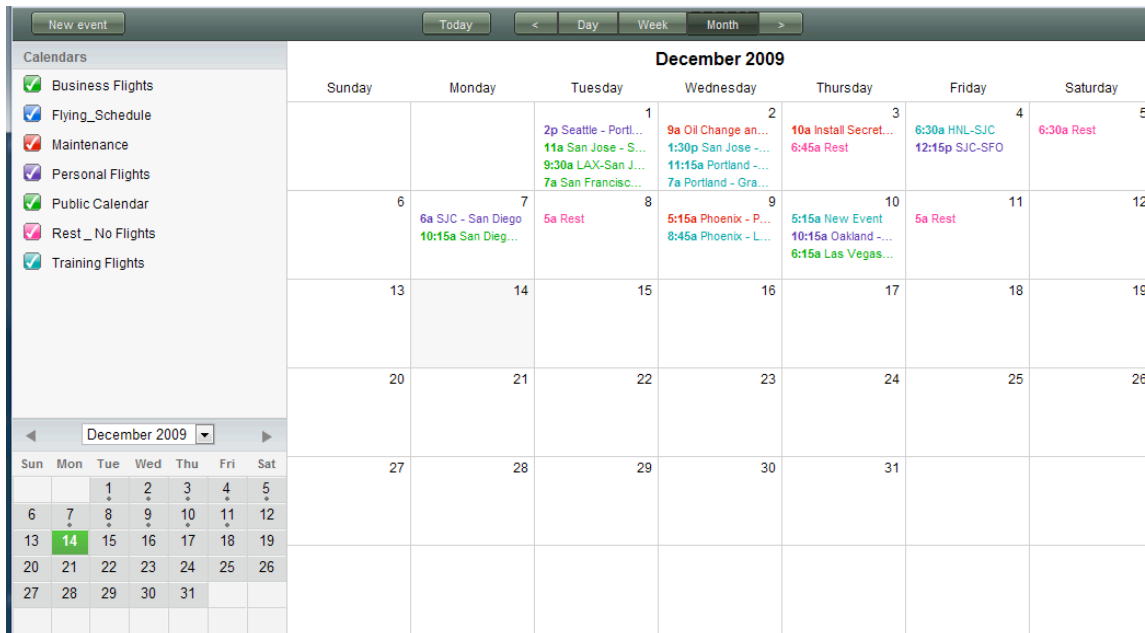
The Full Calendar with Minis, shown above, is a resizable window that displays a Full Calendar with events, three Mini Calendars, a navigator bar to choose the calendar view to display, and the Time Zone that the calendars are set to display in. Events are shown colorized on the calendar views and also shown in the mini calendars as dates with events are underlined. The image above is an example of Month View.

To access the web calendar make sure of the following three things:

- Ensure that your web server is on.
- Ensure that the 4D Ajax Framework v11 Release 5 (11.5) is installed correctly and that the default Webfolder set for your database contains calendar.html (this is in the Webfolder included with 4D Ajax Framework).
- Ensure that you have passed True to the Cal4D_OnStartup method called in your database's On Startup method to enable calendar web serving.

Once these things are set you should be able to access the calendar from any browser by going to: http://your_website/calendar.html (in this case the value "your_website" can be replaced with the IP address of your machine).

Here is an example of the web calendar:



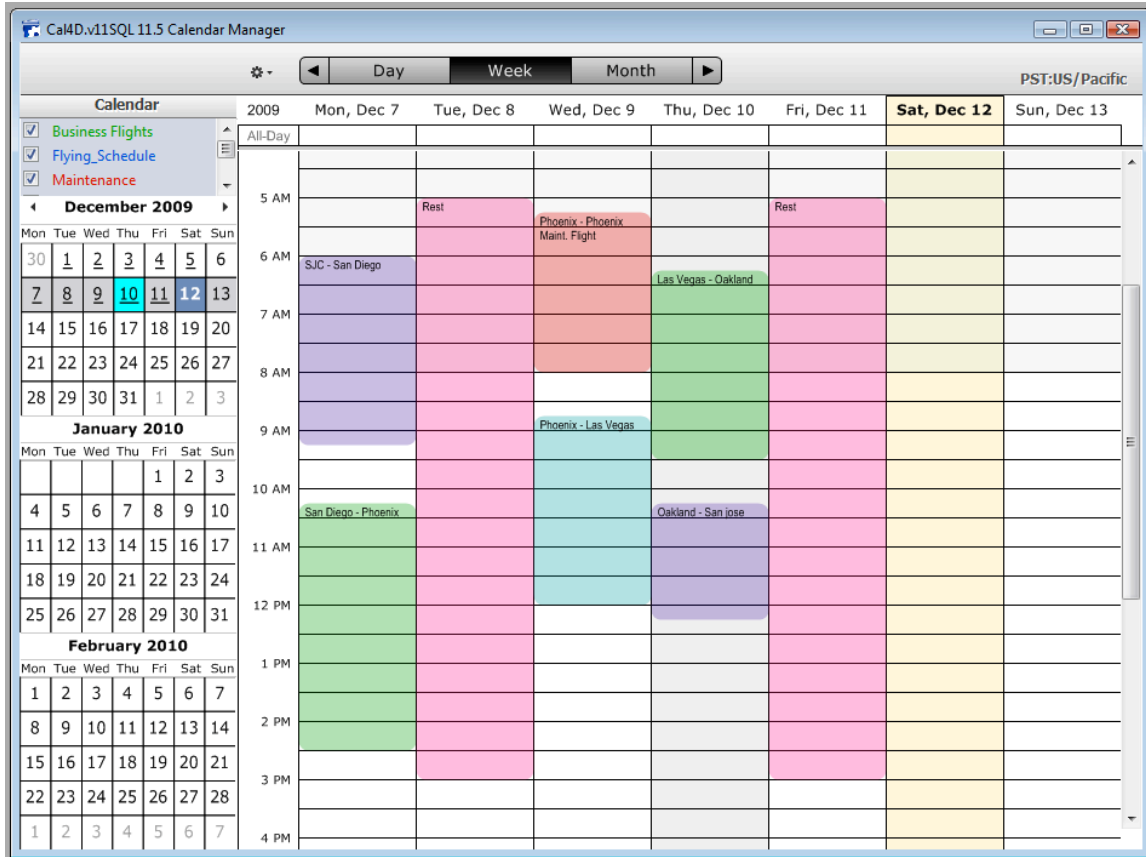
Calendar Views

Cal4D Month View Calendar

In Month View the calendar displays four, five, or six weeks, depending on the length of the current month. The events on days in the current month are displayed. Samples of month view are shown above under the **Displaying a Calendar** section.

Cal4D Week View Calendar

In Week View the calendar displays seven contiguous days that make up the current selected week. The week can have the first day of the week displayed Sunday, Monday, or Saturday. The time can be displayed AM PM format (shown) or Twenty-Four hour format (00:00-23:00). A sample of Week View is shown here:

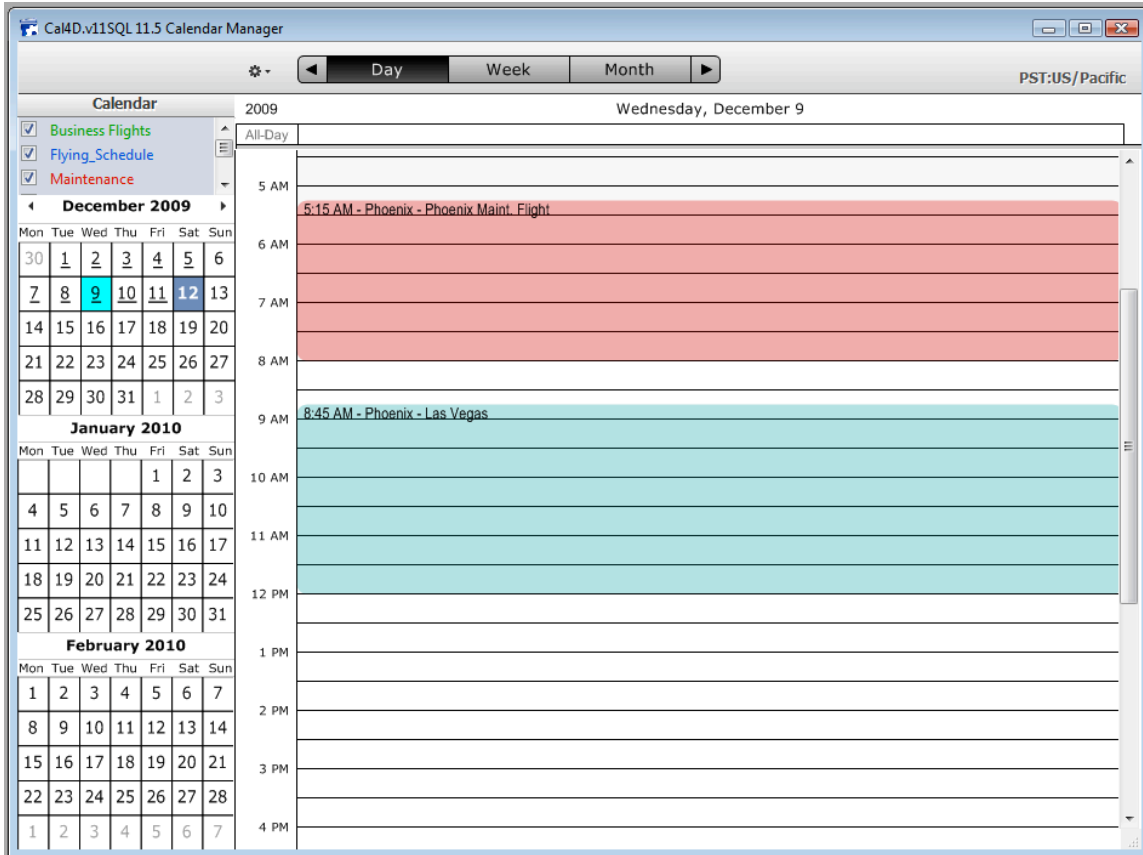


Graphically the whole-hour is represented with a heavy line and the half-hour is represented by a hairline. Though only these two time boundaries are presented, events are displayed on 15-minutes boundaries. Also notice the shading in the Mini-Calendar changes to highlight the current week displayed in the Week view calendar. This feature is highlighted below:



Cal4D Day View Calendar

In Day View the calendar displays one day that is the current selected day. The time can be displayed AM PM format (shown) or Twenty-Four hour format (00:00-23:00). A sample of Day View is shown here:



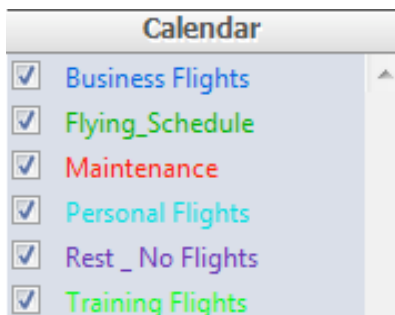
Graphically the whole-hour is represented with a heavy line and half-hour is represented by a hairline. Though only these two time boundaries are presented, events are displayed on 15-minutes boundaries. Notice that, like in week view, the shading in the Mini-Calendar changes to highlight the current day displayed in the Day view calendar.

Navigating Between Views

The Navigation and Display bar, shown below, is used to change the calendar views, Day, Week, or Month and arrow buttons on the left and right to navigate the calendars to the Previous and Next day, week, or month, depending on the currently selected view. This is the Navigation bar:



The “Calendars” list in the upper left corner of the window, shown below, allows you to display or hide calendars associated within the “Cal4D_Groups” table. Here is an example of the calendar list:



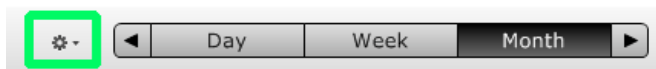
All calendars that are checked are displayed on the three calendar views. To hide a calendar from view, click the check box beside the calendar name and the events associated with that calendar will not be displayed in any of the three calendar views.

Double clicking on any calendar or in the empty area at the bottom of the list will bring up the calendar management dialog, shown on the right. In this dialog you can perform all the edits necessary to manage the calendars of the current user. Detailed instructions are given in detail later in this manual.

The calendar shown in bold is the default calendar. When the current user opens the Event Editor dialog to create a new event, the calendar that the event will be linked to is the default calendar. The default calendar cannot be deleted. To delete the default calendar, another calendar has to be designated as the default calendar.

The Action Menu

The action menu is the entry point for a number of standard actions needed for managing calendars and events. A detailed description of each action is described later in this manual. The Action Menu is only available via 4D, not in the Web Calendar. Access the Action Menu by clicking the cog next the Navigation Bar shown here:



- The “Refresh” menu item will refresh all calendars in the open window.
- The “New Event...” menu item will call **CREATE RECORD** for the Cal4D_Event table and then open the Cal4D Event Editor documented in the next section..

New Event...

When a new event is created the Event Editor dialog, shown below, is use to Create, Edit, and Delete calendar events. Here is an example of the event editor dialog:

Cal4D Event Editor

Calendar: Business Flights

Event Name: Color: ■

Location:

Type: Appointment Status: Confirmed

Start Time: 12/15/09 15:30

End Time: Tue, Dec 15, 2009 16:30

Time Zone: U:-0800:PST:US/Pacific

URL:

Description:

Delete Cancel Save

- **Calendar dropdown:** This menu contains all of the calendar names listed in the “Calendar” list box described above. When creating a new event it will be initially set to the calendar that is marked as the “Default” calendar. The user should set this menu to the desired calendar that the new event should be associated with.
- **Event Name:** The user should enter a short, but descriptive, title for an event. This is the text that will be displayed on the calendar views. An entry in this field is a required field.
- **Color:** The color popup menu presents a pallet of the twelve colors in the Cal4D color pallet. These colors are defined in the “User Constants.bundle” in the matrix database Plugins folder. There are actually twenty-four colors defined in this plug-in, built with the 4D Pop component, 4D Constants. Colors 13-24 are “complimentary” colors to the twelve primary colors. These complimentary colors are used by the web front end for selection highlighting and outlining. The “User Constants.bundle” plug-in is only required for the matrix database. During the compilation process the values in the plug-in replace to constants displayed in the source code.
- **Location:** This is an optional field. Any entry should refer to a physical location or address.
- **Type:** This menu is used to establish the event as one of three basic types shown in the image to the right.
 - Appointment events begin on a specific day at a specific time and end on a specific day at a specific time. An appointment event, such as a long travel event, e.g. plane flight from coast to coast, can cross midnight and in fact exceed more than 24 hours in length. It is recommended not to use Appointments for events which span many days as this can cause rendering issues.
 - All-Day events are twenty-four hours in length and begin at midnight (00:00:00) on day-one and end at midnight (00:00:00) the next day.
 - Multi-Day, also referred to as “Banner”, events begin at midnight (00:00:00) on day-one and end at midnight (00:00:00) on Day-N+1: Example, for a three day Multi-Day event scheduled for the 1st thru the 3rd, the Start time is 1st @ 12 AM and the End time is 4th @ 12 AM, a duration of three days or 72 hours. A user can enter start times and end times for these events, but they are ignored.

- **Status:** There are three possible values for the status of an event, Confirmed, Tentative, or Cancelled. Confirmed is the default value.
- **Start Time** and **End Time:** These values are set by using the form widgets shown on the right. Clicking on the calendar icon will bring up the Date-Picker widget shown at the right. For detailed instructions on navigating the Date-Picker see ***Navigating the Popup Calendar***, the next section. To establish a start or end date, click on a date on the calendar. To dismiss the date-picker without selecting a date, use the escape (Esc) key. To establish a start or end time use the time popup menu to the right of the calendar icon.
- **URL:** This, optional, string when entered in the URL should be a URL to a document available on the web or to another web site page.
- **Description:** This optional text is for entering a detailed description of an event.

Navigating the Popup Calendar

An example Popup Calendar is shown below:

December, 2009							
Today							
Wk	Mon	Tue	Wed	Thu	Fri	Sat	Sun
49		1	2	3	4	5	6
50	7	8	9	10	11	12	13
51	14	15	16	17	18	19	20
52	21	22	23	24	25	26	27
53	28	29	30	31			

Clicking on the Today button will snap the Popup Calendar to the Current Date.

The first navigation control to the right of the Today button is the Next Month popup menu. It navigates to the next month of the abbreviation selected from the current displayed month. For example, if “Jan” is selected the month displayed will be January of the next year, since that is the next January from the currently displayed month.

The second navigation control to the right of the Today button is the Next Years popup menu. It navigates to the year selected of the current displayed month. For example, if “2010” is selected the month displayed will be March of 2010. The menu always displays 12 future years.

The first navigation control to the left of the Today button is the Previous Month popup menu. It navigates to the previous month of the abbreviation selected from the current displayed month. For example, if “Oct” is selected the month displayed will be October of the previous year, since that is the previous October from the currently displayed month.

The second navigation control to the left of the Today button is the Previous Years popup menu. It navigates to the year selected of the current displayed month. For example, if “2004” is selected the month displayed will be March of 2004. The menu always displays 12 previous years.

Import iCalendar File...

To import a Calendar, from the Action Menu, select “Import iCalendar File...”. Cal4D reads the iCalendar, .ics, file format. Internet Calendaring and Scheuling Core Object Specification (iCalendar) was created by the Internet Engineering Task Force Calendaring and Scheduling Working Group. iCalendar is heavily based on the earlier vCalendar by the Internet Mail Consortium (IMC). iCalendar data files are plain text files and used or supported by the majority of popular calendaring applications.

The process of importing an iCalendar file determines which Calendar the file should be associated with. From the iCalendar file the Calendar name will first default to the “X-WR-CALNAME” element. If this element is not present within the iCalendar file then the import

method reverts back to the file name, minus the “.ics” extension. If it is found that a Cal4D calendar already exists with the name of the file being imported then all new records are marked as belonging to that calendar.

During the import events UUIDs are compared to the UUIDs contained within the iCalendar file.

- If an event is not found with a matching UUID a new record is created and populated with the data from the iCalendar file.
- If an event is found in the database with a UUID that matches the UUID being imported, then the method compares the Sequences numbers of the events. If the sequence number of the event being imported is greater than the one within the database then the event within the database is updated, otherwise the event is skipped.

Set Timezone...

Cal4D has extensive support for displaying calendar events in a time zone of preference. To facilitate the quick display of events, Cal4D stores two sets of date and time for each event, “Originators” and “Local” date and time. Typically the “Originators” time and “Local” time are the time zone in which an event was created. However, there are exceptions.

- When events are imported, either from an iCalendar file (.ics) or by subscription, the time zone in the import is saved as the originator’s time and date and the offset from GMT/UTC is used to adjust the originators time to the local time.
- When creating an event, if the situation exists where the event is going to take place in a different time zone, e.g. the user creating the events in the database is on the west coast and the event is going to take place on the east coast, the user can designate the Originators time zone to be US/Eastern and the system will apply the proper offset for “Local” to be three hours earlier in the day. This ensures that when calendars are published, the time zones and times will be correct for clients subscribing on the web, or importing to other calendar applications and are in the target time zone.

It is important to understand the difference between “Standard Time Zones” and “Political Time Zones”. Standards organizations divided the world into 24 longitudinal zones, each 15° wide. These are the “Standard Time Zones”. If it stopped there things would be simple, however, to facilitate national populations working on one common clock, governments around the world have adjusted these boundaries into more than 400 “Political Time Zones”. So it is the responsibility to the developer to allow for “Political Time Zones” to correctly set the local time zone. I recommend you refer to <http://www.worldtimezone.com/> when in doubt about a local time zone. This site also documents when Daylight Savings/Summer Time (DST) begins and ends at those locations that observe it. Note that there is no world wide standard for when DST begins and ends. Each nation declares its own DST begin and end dates.

In the upper right corner of the window is a text box that displays the Time Zone that has been set for the events. The code preceding the colon, in this case “EDT”, is the internationally recognized code of the time zone. Time Zones, their codes, and titles, in this case “US/Eastern”, are stored in the XLIFF localization files, in the components folder of the Cal4D contents.

A Time Zone string, such as “R:-0400:EDT:US/Eastern”, consists of:

- The Time Zone Letter designator, typically used by government agencies such as the military.
- The Time Zone time offset, in hours and minutes, from UTC (Coordinated Universal Time), formally known as GMT (Greenwich Mean Time).
- The International Time Zone Code. There are typically multiple codes for the same time zone. For example zone “W” can have any of the following, AHST: Alaska-Hawaii Standard Time, CAT: Central Alaska Time, or HST: Hawaii Standard Time. What code to display is a matter of choice and can be edited in the XLIFF file.

- Time Zone Title corresponding to the time zone code, if known. Not all time zones have recognized titles as show in the example.

To set the local time zone, from the Action Menu, open the Time Zone dialog shown below. The radio buttons are set based on the internationally published rules for determining time observance, in this case US Daylight Savings Time, DST, of the second Sunday in March and first Sunday in November. If the rules are different for DST in you local, then setting the correct radio button will immediately change the time zone list.

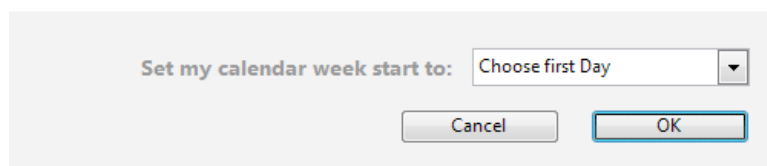
From the “Your Time Zone” menu select you local time zone, in the menu below are time zones coded for daylight savings time. This menu is in zone code order, A-Z. If you have trouble finding the time zone you wish to select, click the “More” button and it will open a dialog and the zones are listed in Content/Location name order.

When a change is made to the local time zone all events from the date the change is made forward are recalculated.

The local time zone set by the user is stored in a file named “TimeZone.xlf” in the Resources folder of the host database.

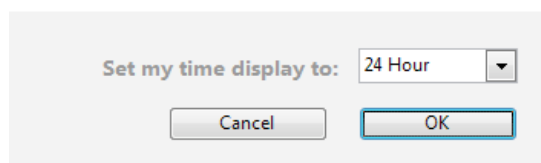
Set Week Start Day...

At startup Cal4D defaults to the week start day based on the time zone selected. However, if the user desires to have a different week start day it can be set from the dialog shown below:



Set Time Display...

At startup Cal4D defaults to the system time display format (System time long). However, if the user desires to have a different time display it can be set from the dialog shown below:

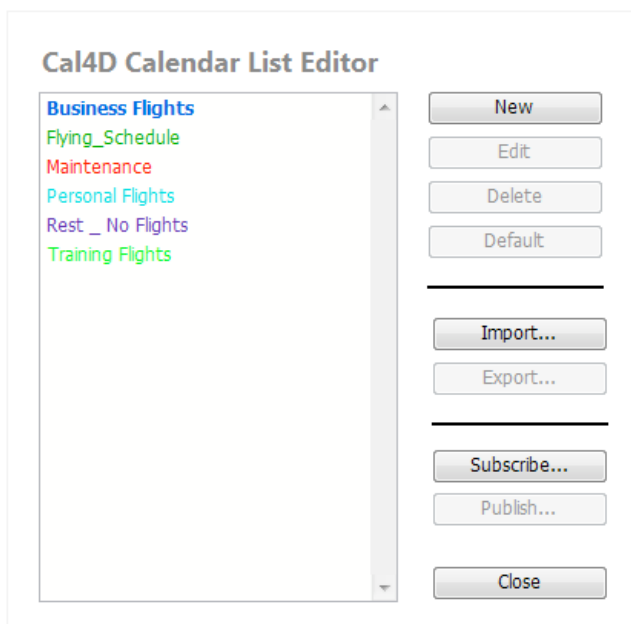


Managing Calendars

The Cal4D component allows a 4D user to manage multiple calendars. For instance, a calendar might be created for meetings, social events, travel, holidays, and work. Having multiple calendars will allow these tasks or events to be grouped in an orderly fashion. The calendar component provides the user with ways to distinguish these calendars from one another with the use of color as well as a feature to allow the calendars to be viewed publicly or privately. Along with these features, Cal4D provides the ability subscribe to, as well as export and import, iCalendar (.ics) files.

The Manage Calendars Dialog

The image below shows the Manage Calendars dialog:



Creating a New Calendar

Clicking on the “New” button will show the dialog on the right:

Calendar Name: This field allows the user to enter a name for the new calendar.

Default Color: This field allows the user to set the default color for the calendar. Note: this setting can be overridden by the color field in the event editor.

Other Users can View this Calendar: When enabled, allows other calendar users to read only or view the calendar events.

Other Users can Modify this Calendar's Events: This option becomes available when “Other Users can View this Calendar” is enabled. When enabled, allows 4D users to make changes or modifications to the calendar.

Edit a Calendar

When you double click on a calendar a Calendar Editor will open. If the calendar is a “Subscribed Calendar” the editor described in “Updating a Subscribed Calendar” below will open. For all other calendars the editor shown below will open.

Delete a Calendar

When the action chosen is to delete a calendar, the user will first be asked to confirm the action. If the user confirms the action, the calendar and all related events will be deleted from the host database.

Setting a Calendar to Default

The user's default calendar is the calendar which is the selected calendar with the user chooses to create a new calendar event. To set a calendar as the default calendar, select a calendar and then click the "Default" button.

Import a Calendar

When the "Import" button is clicked on the user will be presented with the document chooser dialog. The user selects an iCalendar (.ics) file the user calendar will be imported and integrated into the database. Exporting a calendar is handled in the same way with the "Export" button.

Subscribing to a Calendar

Clicking on the "Subscribe..." will prompt the user with the following dialog.

The component allows subscription to an iCal calendar. When subscribed to a calendar individual events within that calendar cannot be edited, moved, or deleted. The only alternative for deleting an event would be to delete the calendar or subscription.

- **Calendar URL:** This field allows the user to enter the address where the iCal calendar is being published. Note that SSL connections are supported but FTP published calendars are not supported.
- **User Name and Password:** If authentication is required the user name and password may be entered here.
- **Refresh:** (Disabled) This field allows the selection how often the subscribed calendar will be updated. This menu is currently disabled since there is a current background process to monitor its feature. It will be enabled in a future release. Until then the choices are:
 - Manually (Static)
- **Color:** This field determines the color of text for events that are from the subscribed calendar.

Updating a Subscribed Calendar

When a user double-clicks on a calendar, if it is a calendar that the user has subscribed to the Calendar Editor will open. If any changes are made, to save those changes the user needs to click the "Post Changes" button. To refresh the data from the host Calendar URL the user needs to click the Refresh Subscription button.

Publishing to a Calendar

To publish a calendar making it accessible via the web, the user needs to select the target calendar and then click the Export button, shown below. This will open the Cal4D Publish Editor shown on the right.

Cal4D Publish Editor

URL:
 Example: MyWorkSchedule.ics (URL must end with ".ics")
 NOTE: "Space" characters in the title will be replaced with "32", its UNICODE value.

Password: (optional)
 Blank = Public Calendar
 Any entry, not an actual password, triggers a check of the "CalSRVR_Subscription" table for a userName and passWord that matches a record that is related to this calendar.

In the URL field the user will enter what will be the last element of the URL that will be used by a third party to subscribe to the calendar, e.g. Summit2009Agenda.ics. If a user leaves space characters within the URL element, Cal4D will substitute "32" for each space character into the text, e.g. Summit 2009 Agenda.ics becomes Summit32200932Agenda.ics, which is the recognized protocol for titling calendars shared over the web.

Managing Calendar Users

The concept of Calendar Users is all about the management of calendars, not the viewing of calendars. All calendars, except the most private should be marked as "Others Can View". This means that anyone, listed Calendar Users and others, will be able to view database calendars. Only listed Calendar Users can both view and edit calendars and calendar events. A Calendar User can give other Calendar Users permission to modify his, or her, calendars by setting the setting the calendar property "Others Can Modify" to true.

All calendars are related to a specific user. When Cal4D is first installed and initialized for the first time the user the user is asked for a name that will be used as his Calendar User Name. That user is, by default, allocated two calendars, Personal and Public. The Personal Calendar is, by default marked "Others Can View". The Public Calendar is marked as "Others Can View" and "Others Can Modify". There is only one calendar with the name Public in the system.

It is assumed that the person to encounter the calendar system during initialization will be a Designer or Administrator. Following that there should be a host database method that calls Cal4D_User_Edit. This call will open the dialog shown below that manages the Cal4D_User table:

Cal4D User Editor

Calendar User	Login Name
designer_user	Designer
admin_user	Administrator

Database Login Name:
None selected ▼

Calendar System User Name:

Cancel

Save User

New User

Delete User

Close

The dialog lists all records from the Cal4D_User table in the listbox on the left. The list box displays the Calendar Users name and Database Login name. On the right side of the dialog are the fields and buttons necessary to table management.

From this dialog records in the Cal4D_User table are Created, Modified, and Deleted. The use of this dialog is as follows:

- Creating a New User: Click the New User button, enter what will be the Calendar System User Name, select the new users Database Login Name from the popup menu and then click the Save User button.
- To edit a user, select the user from the list box. The users name will appear in the "Calendar Sys. User Name" field and the popup menu will be set to the users corresponding Database Login Name. Make any desired changes to these values and then click the Save User button.
- To delete a user, select the user from the list box and click the Delete User button. A confirm dialog will appear and when the dialog is accepted the Calendar user will be deleted.

NOTE: When a user is deleted, all related calendars and related events are also deleted.

Managing the Calendar Server

To support the ability to have private Published Calendars Cal4D installs two tables, CalSRVR_Account and CalSRVR_Subscription, to manager a user pass worded system for Published Calendar Subscription.

To manage these tables a method in the host database needs to be created that calls the Cal4D method CalSRVR_Edit_Subscribers. This method will display the Cal4D Calendar Server Accounts & Subscriptions Editor dialog shown below:

Cal4D Calendar Server Accounts & Subscriptions Editor

Calendar Server Accounts

Subscriber	Email Address	Date

Subscriber:

Email Address:

Date:

Password:

Calendar Subscriptions

Calendar	Date

Calendar:

Date:

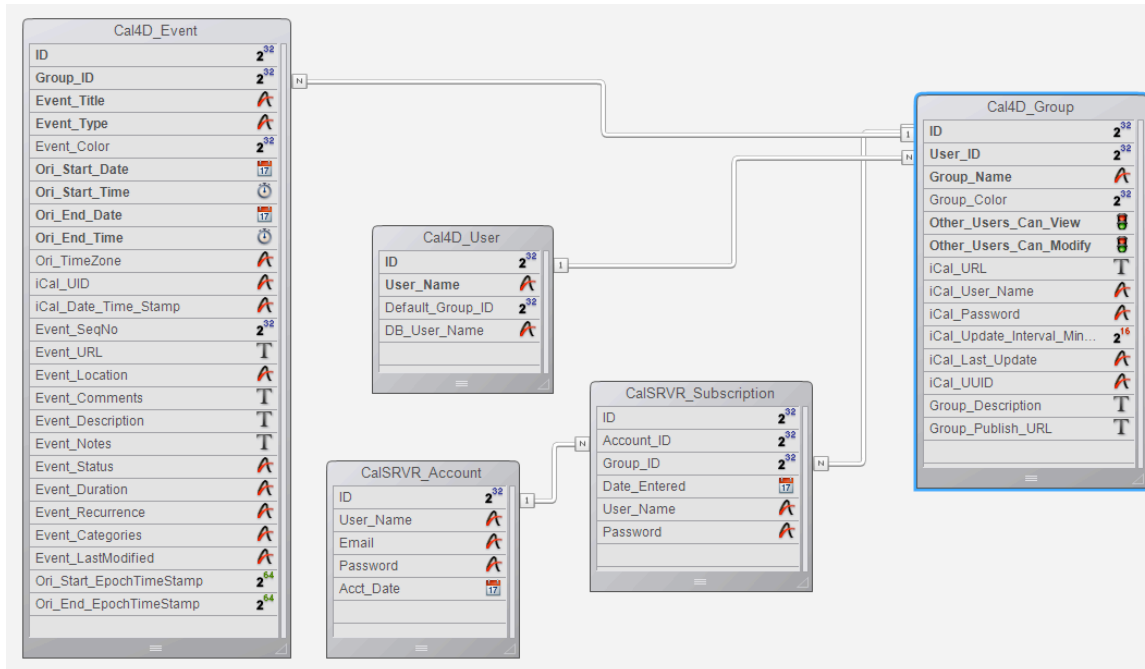
Password:

The left side of the editor window is for managing Calendar Server Accounts. In this area an Administrator can Create, Update, and Delete records in the CalSRVR_Account table.

The right side of the editor window is for managing Calendar Subscriptions. When an account record is selected in the “Calendar Server Accounts” list box, the subscriptions from the CalSRVR_Subscription table related to that record will be displayed the “Calendar Subscriptions” list box.

Data Migration

If you are installing Cal4D into a database that has existing data that you want to publish in calendars that data will have to be integrated into tables installed by the Cal4D component. The tables are shown in the image below:



The tables facilitate the ability of the component to import (subscribe) and export (publish) valid iCalendar format (.ics, RFC 5545) files and for the efficient display of calendars within a host database. For a more detailed discussion of these tables please refer to **Cal4D v11.5 Component for Developers - Structure Definition** later in this document.

Migrating Data into Cal4D

If your database already has the ability to export valid iCalendar (.ics) files, the easiest way to migrate data into the Cal4D tables is to export all of your calendars into .ics files. Next you want to use the Cal4D import function to integrate the data into the Cal4D tables. When using this method if a calendar of the name given in the .ics file does not exist within the Cal4D_Group table, it will be created and all of the events will be associated with that calendar as they are stored in the Cal4D_Event table.

Note: Before importing any exported files the contents of the file should be validated. This can be done for free and online with the iCalendar Validator. It is very important for each event have a Unique Identifier (UID). If you have not assigned a UID to each of your events you can download and use the 4D Tech Tip "How to Generate a UUID" to assign an iCalendar valid UID to each event before the .ics files are generated.

Cal4D Data Migration Assistant - Beta

If you have a database with tables named Cal4D_User, Cal4D_Group, Cal4D_Event, CalSRVR_Account and CalSRVR_Subscription you will need to rename the tables before the Cal4D component is installed. Cal4D will not install the proper tables as long as tables with these existing names exist in the host database. These tables are designed to provide the components ability to subscribe to and publish valid iCalendar formatted files.

Because these tables are so complex it is strongly recommended that you not attempt to migrate your data manually. There are numerous fields that are required or calculated. Once the Cal4D component is properly installed and the tables are created you can use the Cal4D Data Migration Assistant to transfer the calendar data as discussed below.

User and Calendar Information

Create a host database method and call the Cal4D method Cal4D_MigrateDataAssistant. Executing this method will present the dialog shown below. It is a two page dialog. Page one establishes the data needed for the Cal4D_User and Cal4D_Group tables.

Cal4D Data Migration and Integration Assistant (Beta)

User and Calendar | Calendar Events

Calendar User

Name:

DB User:

Calendar Basics

Calendar Name:

Default Color:

Other Users can View this Calendar

Other Users can Modify this Calendar's Events

Publish or Subscribe

Activate the Publish or Subscribe areas

Published Information

URL:

Example: MyWorkSchedule.ics (URL must end with ".ics")

Password: (optional)

Blank = Public Calendar

Any entry, not an actual password, triggers a check of the "CalSRVR_Subscription" table for a userName and passWord that matches a record that is related to this calendar.

Subscribed Information

Calendar URL:

User Name: (optional)

Password: (optional)

In the Cal4D component calendar data management scheme, Users own Calendars and Calendars own Events. User information is stored in the Cal4D_Users table. Calendar information is stored in the Cal4D_Group table. Calendar Event information is stored in the Cal4D_Event table.

In the "Calendar User" group area you should enter a calendar system user name.

In the "Calendar Basics" group area the "Calendar Name" is the required entry. In the Data Migration Assistant the table Cal4D_Group is referred to as "the Calendar".

Publishing and Subscribing

There are two optional data areas of Calendar information, Publish information and Subscribe information. A calendar can have either additional Publish information or additional Subscribe information but not both.

- Publish Information: A “Published” calendar is a calendar that other calendar applications, such as iCal, Outlook, Google Calendar, et al. can subscribe to via the web.

In the URL field enter what will be last segment of the subscription URL, e.g. “mySchedule.ics”. Cal4D will parse the URL “http://<myIPAddress>/CalSRVR/mySchedule.ics” or “webcal://<myIPAddress>/CalSRVR/mySchedule.ics”.

The Password field is optional. It is only used if calendar is to be restricted to a given population of user. In this dialog the password is not the true password to access the calendar. It is merely a trigger for the user name and password to be verified against the users entered in the CalSRVR_Account and CalSRVR_Subscription tables.

- Subscribe Information: A “Subscribed” calendar is a calendar that has data populated from calendars that are shared or published on the web. Entering data in this area will not initiate a request for data refresh from the external calendar server. It will simply save the necessary information for calendar refreshment in the future.

There are numerous shared or published calendars, such as holiday and government calendars, available to the public at iCalShare.com, iCalWorld.com and others. In the URL field enter the full URL to the shared or published calendar. In the User Name field enter the name registered with the calendar server for the target calendar. In the Password field enter the registered password registered with the calendar server for the target calendar.

Calendar Events

The calendar events page allows you select the records to migrate into the Cal4D_Event table and associate with the calendar documented on the User and Calendar page of the dialog.

In the On Load form event the top popup/drop-down menu is populated with the list of tables in the database, minus the tables installed by the Cal4D component. From this table menu, select the source table for the data to be migrated to the Cal4D_Event table.

When a table is selected in this menu, the twenty field menus below will be populated with the field names of the source table. From these menus you select the field that will be mapped to the target fields in the Cal4D_Event table.

Once a source table is selected, clicking on the find icon will bring up the 4D standard query dialog for the selected table. Query for the records that will be related to the calendar record established on the User and Calendar page. The number of records that will be migrated into the Cal4D_Event table will be shown below the menu.

Those field pairs marked with a red dot are required. No data will be migrated if there is not a field menu choice for that pairing.

Those fields marked with yellow dots will be calculated if not selected.

- The “Type” field has restricted values of “Appointment”, “All-Day”, and “Banner”.
- The “Status” field has restricted values of “Confirmed”, “Tentative”, and “Cancelled”.
- The “Sequence” field, a long integer, will be set to one if no menu selection is made.
- The “UUID” field has to be a unique value. It is a forty character alpha field. If not selection is made and unique UUID will be calculated for each record.

- The “Date-Time Stamp” field contains a string in the format “YYYYMMDDTHHMMSSZ”. The string is calculated to GMT/UTC/ZULU time, not local time. If the source data is not calculated to this format, make no choice and it will be calculated as the record is created.
- The “Time Zone” field will receive the time zone selected with the current user first established his record during initialized.
- The “Color” field, a long integer, contains one of twelve colors from the Cal4D Color pallet. The color pallet and twelve colors values in the pallet is shown on the right. If the events in the source table do not have a custom color or if there is no choice in the field menu the value will default to the color of the calendar the selected events will be associated with.

Cal4D Component for Developers

Structure Definition

Cal4D_User

This table contains the list of Cal4D calendar system users.

Fields:

ID	Unique INT32 ID for each user, it is the primary key to the table.
User_Name	Calendar system user name
Default_Group_ID	The default record ID, an INT32, to the Cal4D_Group table that is to be used as the default group (calendar) name.
DB_User_Name	Calendar system user’s database login name.

Cal4D_Group

This table contains the list of Cal4D calendars. It is the one table to the Cal4D_Events table and the many table to the Cal4D_Users table.

Fields:

ID	Unique INT32 ID for each calendar, it is the primary key to the table.
User_ID	This is the foreign key to the Cal4D_User table and references the ID field in the Cal4D_User table.
Group_Name	Generally known as Calendar’s name.
Group_Color	The default RGB color for events associated with this calendar.
Other_Users_Can_View	A Boolean field that stipulates whether or not another user of the calendar system can view events that belong to this calendar.
Other_Users_Can_Modify	A Boolean field that stipulates whether or not another user of the calendar system can edit events that belong to this calendar.
iCal_URL	This field is used for “subscribing” to internet published calendars. A text field that contains a valid URL that starts with either “http” or “webcal”.
iCal_User_Name	A text field that contains the user name to subscribed calendars that require a user name and password.
iCal_Password	A text field that contains the password to subscribed calendars that require a user name and password.
iCal_Update_Interval_Minutes	Contains the number of minutes between updates to subscribed calendars. This is not supported in Cal4D v11.5.

iCal_Last_Update	Contains an ISO date-time string of the last time the subscription was updated.
iCal_UUID	The Universal Unique Identifier for this calendar group.
Group_Description	A general description of the purpose and contents of these events to be associated with this calendar.
Group_Publish_URL	<p>If this calendar is to be published this field will contain the last element to be included in Cal4D calendar server standard URL. Typically it is the calendar name plus ".ics". It is strongly recommended that this field not contain "space" characters.</p> <p><IP>/CalSRVR/< Group_Publish_URL >.</p>

Cal4D_Event

This table contains the event records of Cal4D calendar system.

Fields:

ID	Unique INT32 ID for each event, it is the primary key to the table.
Group_ID	This is the foreign key to the Cal4D_Group table. It references the ID field of the Cal4D_Group table.
Event_Title	This field contains the short description of the event.
Event_Type	<p>The entry in this field must be one of three values, "Appointment", "All-day", or "Banner".</p> <p>An "Appointment" event has a start date, start time, end date, end time, and time zone.</p> <p>An "All-day" event runs for 24 hours. It starts at 00:00:00 of the target day and end at 00:00:00 of the next day.</p> <p>A "Banner" event runs for 2 or more days. It starts at 00:00:00 of the target day and end at 00:00:00 of the Nth day plus one.</p> <p>The time of 00:00:00 of the next, or n-th plus 1, day is used in the Cal4D system, as well as all major calendaring applications, do not recognize 24:00:00 of a given day as a valid time.</p>
Event_Color	By default this takes the value from Group_Color when an event is created. It can be set to a custom color. Other calendaring systems do not recognize custom event colors but with exported and imported to another Cal4D system the custom event color is honored.
Ori_Start_Date	Start date of an event
Ori_Start_Time	Start time of an event
Ori_End_Date	End date of an event
Ori_End_Time	End time of an event
Ori_TimeZone	Time zone at the Cal4D database application.
iCal_UID	The Universal Unique Identifier for this calendar event.
iCal_Date_Time_Stamp	An ISO date-time string of the date and time when the event was created.
Event_SeqNo	The events sequence numbers. Zero at creation and incremented with each saved update of the event.
Event_URL	The URL of an associated file or web site.
Event_Location	Physical location for an event

Event_Comments	User comments regarding the event
Event_Description	Detailed description of the event
Event_Notes	User notes of an event
Event_Status	Event status value of “Confirmed”, “Tentative”, or “Cancelled”.
Event_Duration	An iCalendar, RFC 5545, definition for the duration of an event. Used with importing and subscribing to events. This is not supported in Cal4D 11.5.
Event_Recurrence	The iCalendar, RFC 5545, definition of the frequency of recurrence of an event. Cal4D v11.5 only recognized RRULE “Yearly” defined recurrences, such as holidays. Used with importing and subscribing to events. This is not supported in Cal4D 11.5.
Event_Categories	Stores any values contained in an iCalendar CATEGORIES event element encountered while importing or subscribing to a published calendar. This is not supported in Cal4D 11.5.
Event_LastModified	An ISO date-time string of the date and time when the event was last modified.
Ori_Start_EpochTimeStamp	The number of seconds that have passed since the UNIX Epoch date, 1 January, 1970 till event start.
Ori_End_EpochTimeStamp	The number of seconds that have passed since the UNIX Epoch date, 1 January, 1970 till event end.

CalSRVR_Account

This table contains records for users that are subscribing to calendars that require a user name and password.

Fields:

ID	Unique INT32 ID for each user, it is the primary key to the table.
User_Name	Calendar subscriber system user name
Email	Calendar subscriber system email address
Password	Calendar subscriber system password
Acct_Date	Date of the account creation.

CalSRVR_Subscription

This table contains records for users that are subscribing to a particular calendar that require a user name and password.

Fields:

ID	Unique INT32 ID for each subscription, it is the primary key to the table.
Account_ID	This is the foreign key to the CalSRVR_Account table. It references the ID field of the CalSRVR_Account table.
Group_ID	This is the foreign key to the Cal4D_Group table. It references the ID field of the Cal4D_Group table.
Date_Entered	Date of the subscription creation.
User_Name	Calendar subscriber user name, normally the same as in [CalSRVR_Account] User_Name but can be different
Password	Calendar subscriber password, normally the same as in [CalSRVR_Account] Password but can be different

Time Zone Resource File

Cal4D creates and maintains a file in the Resource folder of the host database named TimeZone.xml. In this file there is a "User" element for each registered Cal4D Calendar System User. See example below:

```
<user>
  <name>DBMA</name>
  <zone letter="R" name="US/Eastern" observing="Standard" offset="-0500"
  zoneAbbr="EST"/>
  <weekStart day="1"/>
  <timeDisplay format="AM PM"/>
</user>
```

This file contains preferences for Time Zone, Week Start Day, and Time Display Format for each user. It should not be edited by outside of Cal4D.

Entries in this file is managed during the initial user establishment and with each user created or deleted within the Cal4D_User_Edit process.

If this file is deleted each registered user will be presented with the Timezone dialog, shown elsewhere in this documentation, for this information to be established and saved.

Language Reference

Cal4D_Cal_CalendarWithMinis

There are no parameters to this method. It creates its own process and opens the self contained calendar window described above.

Cal4D_DP_Widget (aDateVar_D {; aLeftVar_L; aTopVar_L}) -> aDate_D

To display a mini calendar, create a date variable or field and then a trigger object, such as an invisible or highlight button. In the example to the right the small calendar image is covered with a highlight button. The calendar image shown on the right is included with host demo database in the Resources folder.

In the button object include code similar to that shown below:

```
C_Date ($Date_D; PopDate_D)
  $Date_D:=Cal4D_DP_Widget (PopDate_D)
  If ($Date_D#!00/00/00!)
    PopDate_D:=$Date_D
  End if
```

If the variable or field that is passed as the argument to Cal4D_DP_Widget contains a valid date the Cal4D Popup Calendar will open with that date highlighted, otherwise it will open with the current date. The opened Popup Calendar is shown to the right. The optional parameters allow you to pass the coordinates where the popup window will be located.

Cal4D_EvtTabBar_Handler

This is a callback method that is used by then Cal4D_NavBar component to communicate with Cal4D.

Cal4D_Info (Selector_T) -> TEXT

This returns general information about the component. Selectors are “version” and “numeric_version”.

Cal4D_MigrateDataAssistant

There are no parameters to this method. It creates its own process and opens the self-contained window described in the Data Migration Assistant Guide.

Cal4D_NavBar_Handler

This is a callback method that is used by then Cal4D_NavBar component to communicate with Cal4D.

Cal4D_Quit

There are no parameters to this method. It creates its own process and signals any current Cal4D processes to terminate.

Cal4D_Startup ({StartCalSRVR {; CurrentUserName{; DBUserNameList}}})

Cal4D_Startup is called early in the host database startup process, in the database On Startup method. In this method Cal4D initializes all its variables and user preferences, such as time zone, week start day, etc.

If called without parameters the method will not install the CalSRVR tables. If they don't exist, it will call the 4D command “Current User” to gain the database login name of the default user, and call the 4D command “GET USER LIST” to gain a list of database users.

- **StartCalSRVR** : Boolean - If true, install the CalSRVR tables if they do not exist in the database and initialize the necessary component variables to support serving published calendars.
- **CurrentUserName** : String - If omitted or not passed Cal4D will use the 4D command “Current User” to learn the name of the current user.
- **DBUserNameList** : Pointer - If the host database is using a custom users and login system an array of the list of database users must be passed to the component by way of a pointer to the array.

Cal4D_User_Edit

There are no parameters to this method. It creates its own process and opens the self contained editor window described above.

CalSRVR_Edit_Subscribers

There are no parameters to this method. It creates its own process and opens the self contained editor window described above.

CalSRVR_Export (Path_T)

Provided a path to a volume on the computer or network, Cal4D will export a valid iCalendar (.ics) file of the current calendar, currently selected record in the Cal4D_Group table.

CalSRVR_IsOnWebConnection (URL_T; Header_T; ClientIP_T; ServerIP_T; User_T; Pass_T) -> Boolean

When the URL contains "GET /CalSRVR@" a valid iCalendar file is returned to the requesting IP address.

CalSRVR_OnWebAuth -> Boolean

This method tests the URL header for "/CalSRVR/" presents in the URL, if it is not present the value “True” is returned so that processes continue on to the “On Web Connection” method.

If "/CalSRVR/" is present in the URL header it is then tested for the presence of "Authorization". If it does not exist the assumption is that the request is for a Publish Public Calendar. When processing for a Public Calendar if the requested calendar is not found or the calendar requires a User Name and Password, the appropriate Internet error code will be returned.

When processing for a Private Calendar the User Name and Password is tested against the CalSRVR_Account and CalSRVR_Subscription tables. If the tests fail the appropriate Internet error code will be returned.

If all tests are passed the value TRUE is passed and processing continues to the CalSRVR_IsOnWebConnection method in the database On Web Connection method.

DAX_Cal_JSON_Tasker (TaskBuf_T ; CustomVarName_aP ; CustomVarValue_aP)

DAX_Cal_JSON_Tasker is called from the method DAX_DevHook_DCS_SetSelection. The interface for the call to this method was established when Cal4D was first installed and DAXCAL_AddFauxView was added to be call from the method DAX_DevHook_DCS_ViewAdd. This creates a table in the 4DAF "Virtual Structure" contained in ".../MyHostDB>.4dbase/Resources/Support/Tables.xml". This establishes the ability of the web front-end to be able to use the "Custom Variables" interface as a communications channel between the web front-end and the Cal4D component as the 4D Web Back-end.

The first parameter passed to DAX_Cal_JSON_Tasker is the value passed to the back-end with the name "DAXCAL". The second and third parameters are pointers to host database process text arrays.

Before calling DAX_Cal_JSON_Tasker there needs to be two process text arrays declared in the host database that will be passed to DAX_Cal_JSON_Tasker using pointers. When control is returned to the host database the arrays will contain the proper responses to the 4DAF web front-end.

Cal4D HTML Pages for Developers

Creating the Calendar Object

The function DAX_Calendar.init() is used to create the calendar object, as shown here:

```
myCalendar = new DAX_Calendar().init({
    mainCalendarNode    : document.getElementById('maincalendar'),
    miniCalendarNode    : document.getElementById('minicalendar'),
    startDay            : DAX_CalendarConstants.weekStartDay.mon,
    dateFormat          : DAX_CalendarConstants.dateFormats.dayMonthYear,
    timeFormat          : DAX_CalendarConstants.timeFormat.twentyFourHr,
    startDate           : new Date(2009, 7, 20),
    mainView            : DAX_CalendarConstants.mainView.month,
    hideMiniCalendarEventIndicator: true,
    readyCallback       : functionName,
    updateCallback      : functionName,
    readOnly            : boolean
});
```

Here are the parameters explained:

Parameter	Type	Description	Optional
mainCalendarNode	reference to a node	Add main calendar to passed node	
miniCalendarNode	reference	Add mini calendar to passed node	x

	to a node		
startDay	constant	First day of the week	x
dateFormat	constant	Time formatting	x
startDate	date	Calendar will initialize and load this date	x
mainView	constant	Calendar can initially display as day, week, or month	x
hideMiniCalendarEventIndicator	boolean	Disable mini calendar event notation	x
readyCallback	function	Called when init response comes in	x
updateCallback	function	Called when init response comes in	x
readOnly	boolean	Show calendar in read-only mode, events are not editable	x

mainCalendarNode

Pass a reference to the node where main calendar node will be added. DIV HTML node is recommended, with fixed width and height properties. Here's an example DIV node that could be used:

```
<div style="width: 700px; height: 500px" id="maincalendar"></div>
```

miniCalendarNode (optional)

Pass a reference to the node where the mini calendar node will be added. DIV HTML node is recommended, with fixed width and height. Here's an example DIV node that could be used:

```
<div style="width: 300px; height: 200px" id="minicalendar"></div>
```

If a miniCalendarNode is not passed, mini calendar won't be created.

startDay (optional)

Sets week start day. Possible values are:

- DAX_CalendarConstants.weekStartDay.mon - week starts with Monday
- DAX_CalendarConstants.weekStartDay.sun - week starts with Sunday

The server will send a default value for this based on the server-side setting, if not otherwise specified.

dateFormat (optional)

Sets date formatting, used in event editor date entry fields. Possible values are:

- DAX_CalendarConstants.dateFormats.dayMonthYear - DD/MM/YYYY format
- DAX_CalendarConstants.dateFormats.monthDayYear - MM/DD/YYYY format

The server will send a default value for this based on the server-side setting, if not otherwise specified.

startDate (optional)

By default, calendar will display current date as initial position. Use Date object to override that. An example of valid Date object is "new Date(2009, 7, 20)", which returns August 20th, 2009.

mainView (optional)

Main view will display month view by default. Use this to override main view. Possible values are:

- DAX_CalendarConstants.mainView.month - month display
- DAX_CalendarConstants.mainView.week - week display
- DAX_CalendarConstants.mainView.day - single day display

readyCallback (optional)

Pass a function that will be called when calendar is initialized. The calendar in initialized state has received initial settings from the server, but didn't receive calendar list or data yet.

The function will receive the calendar instance reference. Here's an example of defining a function and passing it to new calendar:

```
function readyCallbackFunction(calendarInstance) {
    // callback code goes here
}

myCalendar = new DAX_Calendar().init({
    mainCalendarNode    : someNode,
    readyCallback       : readyCallbackFunction
});
```

updateCallback (optional)

Pass a function that is called when calendar receives data. At this point, the calendar will have a list of calendars available. The function will receive a calendar instance reference as the first parameter. See above example for readyCallback on how to call your callback function.

timeFormat (optional)

This sets time formatting used in day, week, month, and editor view. Possible values:

- DAX_CalendarConstants.timeFormats.twentyFourHr - 24 hour format
- DAX_CalendarConstants.timeFormats.amPm - AM/PM format

hideMiniCalendarEventIndicator (optional)

Disable mini-calendar dots that show if a certain date has an event(s). This will also suppress calls to 4D to request that information.

Calendar methods**goToDate**

```
myCalendar.goToDate(date);
```

Main (and mini, if displayed) calendar will jump to passed date as selected date. Here is an example:

```
myDate = new Date(2009, 7, 20); // aug 20th, 2009
myCalendar.goToDate(myDate);
```

showMainCalendarView

```
myCalendar.showMainCalendarView(mainViewConstant);
```

This switches the main calendar view to month, week, or single day view.

mainViewConstant can be:

- DAX_CalendarConstants.mainView.month
- DAX_CalendarConstants.mainView.week

- `DAX_CalendarConstants.mainView.day`

Example:

```
myCalendar.showMainCalendarView(DAX_CalendarConstants.mainView.week);
```

getCalendars

```
var calendarList = myCalendar.getCalendars();
```

Returns calendar objects with calendar name and unique id. Object structure is { name : string, calendarId : number }.

Example:

```
// get list of calendars
var calendarList = myCalendar.getCalendars();

// check number of calendars returned
var numberOfCalendars = calendarList.length;

// get the name of first calendar
var firstCalendarName = calendarList[0].name;
```

NOTE: this function shouldn't be called until calendar has returned a successful updateCallback.

showCalendars

```
myCalendar.showCalendars(calendarIds);
```

calendarIds is an array of calendar unique ids that will be visible. The calendar is redrawn after this is called. By default, all calendars are visible. Here is an example:

```
var calendarsToShow = [2,4,5]; // make 3 calendars visible
myCalendar.showCalendars(calendarsToShow);
```

hideCalendars

```
myCalendar.hideCalendars(calendarIds);
```

calendarIds is an array of calendar unique ids that will be hidden. The calendar is redrawn after this is called. By default, all calendars are visible. Here is an example:

```
var calendarsToHide = [2]; // hide one calendar
myCalendar.hideCalendars(calendarsToHide);
```

Relative date jumps

The following functions are self-explanatory and are used to change the current date according to the defined jump. The calendar is redrawn after they are called.

- `myCalendar.goToNextMonth();`
- `myCalendar.goToPreviousMonth();`
- `myCalendar.goToNextWeek();`
- `myCalendar.goToPreviousWeek();`
- `myCalendar.goToNextDay();`
- `myCalendar.goToPreviousDay();`

Installation and use

Deployment Mode

Include the following lines in HEAD tag:

```
<meta http-equiv="X-UA-Compatible" content="IE=8" />

<link rel="stylesheet" href="/dax/external/jquery_css/smoothness/jquery-
ui-1.7.2.custom.css" media="all" type="text/css" />
<link rel="stylesheet" href="/dax/calendar/calendar.css" media="all"
type="text/css" />

<!-- minified 4D Ajax Framework -->
<script language="javascript" type="text/javascript" charset="ISO-8859-1"
src="/dax/js/compile.js"></script>

<!-- minified 4DAF Calendar -->
<script language="javascript" type="text/javascript" charset="ISO-8859-1"
src="/dax/calendar/calendar_compiled.js"></script>

<!-- English localization for 4DAF (calendar loads localization
automatically) -->
<script language="javascript" type="text/javascript" charset="utf-8"
src="/dax/js/localization/resources_en.js"></script>
```

Developer Mode

Developer mode is using uncompressed files for easy debugging. Include the following lines in HEAD tag:

```
<!-- make sure that calendar renders in IE8 properly -->
<meta http-equiv="X-UA-Compatible" content="IE=8" />

<!-- jQuery UI style, used in date popup -->
<link rel="stylesheet" href="/dax/external/jquery_css/smoothness/jquery-
ui-1.7.2.custom.css" media="all" type="text/css" />

<!-- Calendar style sheet -->
<link rel="stylesheet" href="/dax/calendar/calendar.css" media="all"
type="text/css" />

<!-- JSON parser -->
<script language="javascript" type="text/javascript" charset="ISO-8859-1"
src="/dax/external/json2.js"></script>

<!-- jQuery -->
<script type="text/javascript" charset="utf-8" src="/dax/external/jquery-
1.3.2.min.js"></script>
<script type="text/javascript" charset="utf-8" src="/dax/external/jquery-
ui-1.7.2.custom.min.js"></script>

<!-- DAX framework -->
<script language="javascript" type="text/javascript" charset="ISO-8859-1"
src="/dax/js/framework.js"></script>

<!-- DAX calendar -->
<script type="text/javascript" charset="utf-8"
src="/dax/calendar/calendar_core.js"></script>
<script type="text/javascript" charset="utf-8"
src="/dax/calendar/calendar_connection.js"></script>
```

```

<script type="text/javascript" charset="utf-8"
src="/dax/calendar/calendar_header.js"></script>
<script type="text/javascript" charset="utf-8"
src="/dax/calendar/calendar_mini.js"></script>
<script type="text/javascript" charset="utf-8"
src="/dax/calendar/calendar_monthview.js"></script>
<script type="text/javascript" charset="utf-8"
src="/dax/calendar/calendar_columnview.js"></script>
<script type="text/javascript" charset="utf-8"
src="/dax/calendar/calendar_eventview.js"></script>

<!-- DAX calendar localization -->
<script type="text/javascript" charset="utf-8"
src="/dax/calendar/calendar_localization_en.js"></script>

<!-- DAX UI -->
<script type="text/javascript" charset="utf-8"
src="/dax/calendar/dax_ui.js"></script>
<script type="text/javascript" charset="utf-8"
src="/dax/calendar/dax_utilities.js"></script>

```

Initializing jQuery and DAX

The calendar requires that jQuery is initialized, and that DAX framework has the user successfully logged in. The following function makes sure that jQuery is ready:

```
jQuery(document).ready(function(){ // code goes here });
```

And following function is called when DAX user is logged in:

```
dax_loginSucess() { // code goes here }
```

Sample code to handle initialization follows:

```

// will be called when jQuery is ready
jQuery(document).ready(function(){

    // login to DAX
    dax_login('Username', 'Password');
})

// will be called when DAX login is done
function dax_loginSucess() {

    // create calendar here
    daxCalendar = new DAX_Calendar().init({
        mainCalendarNode : node
    });
}

```

Examples

Further examples can be found in the template pages in the extras folder included in the 4D Ajax Framework v11 Webfolder.

Further References

See the 4D Ajax Framework section of the 4D Web 2.0 Pack product page at:

<http://4d.com/us/products/4dweb20pack/ajax.html>

More information on Cal4D and the 4D Ajax Framework is available there, including screencasts of common techniques using the component and demo databases.