Level: Intermediate
Works with: QuickPlace
Updated: 03-Sep-2002

Adding calendar roll-up to **QuickPlace**

by
Christopher
Pepin

Lotus QuickPlace is a powerful Domino-based application that provides virtual workplaces to help your teams collaborate and share knowledge. Out-of-the-box, QuickPlace offers (among many other features):

- Instant creation of workplaces
- Shared content
- Shared calendars
- Milestones and events

In addition, QuickPlace provides you with a high degree of customization, which you can do through a browser, the Domino Designer, and the QuickPlace developer's kit (QDK). This month, *LDD Today* includes two articles on how you can customize and extend QuickPlace to adapt it to your requirements. One article,"**Extending QuickPlace workflow**," describes a way to enhance QuickPlace workflow. The other article is the one you're now reading!

In this article, we demonstrate how to combine multiple QuickPlace calendars into one interface, a process we call *calendar roll-up*. As we explain how to do this, we'll also give you a general overview of QuickPlace architecture. In addition, all of the code used in this article is available to download from the **Sandbox**. This article assumes you're an experienced QuickPlace developer and are familiar with basic QuickPlace terminology.

Why a calendar roll-up?

The QuickPlace calendar is an especially critical component of your Place because you can use it to track important project-related events. For example, you can create a calendar page that describes the agenda for a meeting and insert the title of the page into the meeting time field. When meeting attendees look at the calendar, they see the page title in the relevant time slot and can click the title to display the agenda. Calendars are unique to each room in the QuickPlace. That is, each room has its own calendar, which is not displayed in the main room of the Place. (Note that by default in QuickPlace 3.0, calendars in inner rooms are hidden. You will need to modify the Room Options to make the calendar visible.)

There may be times when members would like to see a complete view of all project events, with calendar entries from inner rooms "rolled up" and displayed in the main room. You can implement this feature by using a QuickPlace PlaceBot. PlaceBots can be written in either LotusScript or Java, and they function very much like standard Domino agents. (The PlaceBot we use in this article is written in LotusScript.) PlaceBots can run on a scheduled basis or can be event triggered (for example, when a document is saved).

How the calendar roll-up PlaceBot works

When you create a new Place, three Domino databases are created in the process:

- Main.nsf contains the Place content and configuration information
- Contacts1.nsf contains the member list

```
Lotus Developer Domain: Adding calendar roll-up to QuickPlace
www.lotus.com/ldd/today.nsf
```

- Search.nsf is a site search database

If you create an inner room for a Place, QuickPlace creates a new Domino database to store content and configuration information for the inner room. In addition, a configuration document is added to the main room that "points" to the file name of the inner room.

Our calendar roll-up PlaceBot will run on a scheduled basis and do the following:
- Read documents in the main room to find the inner rooms of the Place
- Access the calendar entries of each inner room
- Copy calendar entries from each inner room to the main room

The first thing our PlaceBot will need to do is access the documents in the main room that contain the names and file paths to the inner rooms. The list of subrooms is contained in a Domino view called h_AreaView, which resides in the main room. The following LotusScript code demonstrates how to get a handle to this view:

```
Dim session as new notessession
Dim thisdb as notesdatabase
Dim catalogview as notesview
Dim dateTime As New NotesDateTime("12/01/94")
Dim collection as notesdocumentcollection

    Set thisdb = session.currentdatabase
    Set catalogview = thisdb.getview("h_AreaView")
```

Next, we'll use a while loop to process all the documents in the h_AreaView view. The documents contain a field called h_LocDBName that stores the file name of the inner room database, and another field called h_LocDirPath that stores the path. By default, Places are located in the QUICKPLACE subdirectory of the Domino data directory on the server. Concatenating QUICKPLACE\, the path, and file name together will give us the full file path of the database, as shown below:

```
While Not qpcatalogdoc Is Nothing
    Dim qpdb As New notesdatabase("", "")
    qpdatabase$ = qpcatalogdoc.h_LocDBName(0)
    qppath$ = qpcatalogdoc.h_LocDirPath(0)
    room$ = "QUICKPLACE\" & qppath$ & "\" & qpdatabase$
    Call qpdb.open("", room$)
```

In the last line of the preceding code, we get a handle to the inner room database of the QuickPlace.

Now that we have identified the inner room, we can access the calendar entries within it. Although the room contains a view showing calendar entries, this view is a standard calendar view. Therefore, copying items from this view will result in duplicate entries because repeating calendar events are actually the same document. To avoid this, our PlaceBot uses the db.search method to return a document collection of all calendar documents in the inner room, as shown in the code segment below. The formula is the same as the selection formula for the calendar. Note the formula for QuickPlace 2.0.*x* is different from the formula used in QuickPlace 3.0. In our example, we are running on QuickPlace 3.0:

```
Const V2QPCalSearchFormula = {(@IsAvailable(h_CalendarDate) & @IsAvailable(h_CalendarTime)) | (
h_isInCalendar = "1") & h_PublishedVersionUnid=""}
Const V3QPCalSearchFormula =
Const CALENDAR_VIEW = "DeleteCalendarEntries"

Set collection = qpdb.Search(V3QPCalSearchFormula,dateTime,0)
    For i = 1 To collection.Count
        Set calendardocument = collection.GetNthDocument(i)
        Set newcalendardoc = New NotesDocument( thisdb )
        Call calendardocument.CopyAllItems(newcalendardoc, True )
        newcalendardoc.QUICKPLACE = qpname$
        newcalendardoc.Form = "CalendarEntry"
        Call newcalendardoc.Save( True, True )
    Next i
```

The preceding code copies calendar documents from the inner room to the main room. It also adds a new field

```
Lotus Developer Domain: Adding calendar roll-up to QuickPlace
www.lotus.com/ldd/today.nsf
```

called QUICKPLACE to each document. This field is set to the name of the inner room and is used to display the name of the inner room in the calendar view. This makes it easy for users to see which inner room the calendar document originated in.

One final issue to consider is synchronizing calendar entries. The code described above simply copies calendar documents from the inner room to the main room each time it runs. This means you will end up with duplicate calendar entries. There are a number of ways to solve this problem. A simple solution is to have the PlaceBot delete all existing calendar entries in the main room at the beginning of its run. A disadvantage to this is that any calendar documents created in the main room itself will be deleted, although you can prevent this by instructing the PlaceBot to first perform a check and only delete documents in the main room that contain the QUICKPLACE field and originate in inner rooms. Our sample code file named Rollup Calendar - Basic.lss uses this method.

Another solution is to have the PlaceBot perform a comparison between calendar documents in the main room and calendar documents in inner rooms, and then to copy new calendar entries and update existing calendar entries. The sample file we provide called Rollup Calendar - Advanced.lss uses this technique. Note that this method does not take into account when calendar entries are deleted from inner rooms.

Before running this sample, make sure you, as creator/signer of the PlaceBot, have access to all appropriate QuickPlace databases and documents.

Creating the calendar roll-up PlaceBot

To create the calendar roll-up PlaceBot, you should first download the code sample files we provide in the **Sandbox** and place them in a local file folder on your PC. Then do the following:
1. Log in as manager to your QuickPlace's home page, and click Customize. Then click PlaceBots to see the PlaceBots page.
2. Click the New PlaceBot button to see the New PlaceBot page.
3. Fill in the sections of this page as follows:
- In the "What is the title of this PlaceBot?" field, enter the name you want to call your PlaceBot, for example, CalendarRollup.
- In the "When should this PlaceBot run?" field, click the "On Scheduled basis" radio button.
4. Click the Set Schedule button to display the PlaceBot Schedule page.
-In the "What pages should the PlaceBot affect?" section, choose "All the pages in this room" from the drop-down list.
- In the "How frequently do you want the PlaceBot to run?" section, select the frequency you would like the PlaceBot to run (monthly, weekly, daily, and so on). We recommend you run CalendarRollup daily, preferably at night when it's less likely that many people are working.
5. After completing the schedule click Next. This returns you to the New PlaceBot page. You must now load the code we have provided.
6. Click the folder icon near the bottom of the page and select either Rollup Calendar - Basic.lss or Rollup Calendar - Advanced.lss (depending on which method of calendar synchronizing you want to use) from the dialog box.
7. Click Done.

Modifying the calendar view

The next step is to modify the QuickPlace calendar view so that it can display the QUICKPLACE field. To do this, you must edit the System\Calendar view in the main room (main.nsf).
1. Open main.nsf in Domino Designer and then open the System\Calendar view.
2. Replace the column formula for the fourth column with this formula:

```
@If( h_AllDayEvent="1"; "-All Day- " + QUICKPLACE + ": " + h_Name; @Left( @Text( h_CalendarTime;"S1");
5) + @Right( @Text( h_CalendarTime;"S1"); 3) + "-" +@Left(@Text(@Adjust(h_CalendarTime; 0; 0; 0; 0;
h_CalendarDuration; 0); "S1"); 5) + @Right( @Text( h_CalendarTime;"S1"); 3) + " " + QUICKPLACE + ": "
+h_Name)
```

3. Check that the view property "Prohibit design refresh or replace to modify" checkbox is selected so that this change will not be lost when the Domino design refresh server task runs.
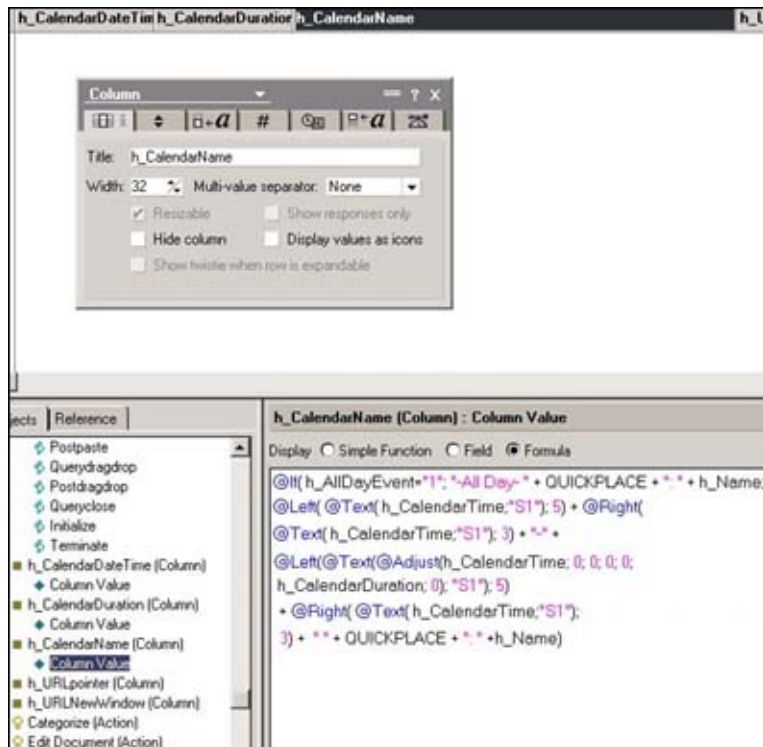
After you make the changes to the calendar view and save it, your Place members can now use calendar roll-up. When they open the main room in your Place and click Calendar, they'll see the following view:

Our PlaceBot can also be extended further to roll up calendars across multiple Places. For example, you can create a "Calendar" Place to store all calendar entries for all the Places on the QuickPlace server. To do this, have your PlaceBot loop through all the Places on the server and copy all calendar entries from all Places into the

Calendar Place. Note that if you have a large number of Places and long calendar entries, the roll-up calendar view can get so full of information, it may be difficult to read.

PlaceBots: The QuickPlace developer's friends

In this article, we've shown you a method for rolling up all inner Place calendars into a single calendar displayed in the main room. We've provided the LotusScript code for the PlaceBot so that you can implement this calendar roll-up at your own site, and we've included explanations that will help you modify and adapt our samples to your own particular needs.

Of course, there are many other ways you can use QuickPlace's flexible development platform to further extend its features. See the *LDD Today* article "**Extending QuickPlace workflow**" for a customization example that uses a Java PlaceBot, and you can find additional customization tips in the IBM Redbook, **Customizing QuickPlace**.

ABOUT THE AUTHOR
Christopher Pepin is a senior information technology specialist for IBM Software Solutions for Lotus (ISSL) in New York City. He has a MS degree in Computer Science and is a dual Principal Certified Lotus Professional in both Lotus Notes and Domino R5 Administration and Development. Chris has over nine years of experience with Lotus Notes and is responsible for architecting, designing, and developing total system solutions for customers.