Customizing Resource Reservations — Enable Users to Schedule Rooms Based on Room Attributes

By Mark Rose

If you administer the Notes/Domino Resource Reservations system for your organization, or if you are a developer supporting this system, you’ve probably heard from users who are tired of showing up for meetings, only to find that the reserved room does not have a crucial piece of equipment. Users may also complain that prior to making a reservation, they are spending too much time running around to check which rooms have the equipment they will need. This article shows some changes you can make to the Resource Reservations database to enable users to reserve rooms based on room attributes, such as telecommunications equipment, eating facilities, windows, video conferencing, whiteboards, or whatever is important to your users. Users will be able to select from a list of attributes when making a reservation. A customized search engine will then search for rooms that meet all of the user’s requirements.

After showing you how to implement the solution in the Resource Reservations system, I’ll show how to modify the Mail template to extend the solution to users who reserve resources through Notes Calendar and Scheduling features. Anyone with mid-level Domino application development skills (including experience with LotusScript and Calendar and Scheduling) can make the modifications. To assist you, I’ve included a template of the changes for the Resource Reservations database, including all of the design elements and search code, in a download available from THE VIEW Web site.

1 In the standard Resource Reservations database that ships with all versions of Notes/Domino, users can search for rooms based on location, capacity, and availability. However, they must schedule resources or equipment separately from the room.

2 At www.eVIEW.com, click THE VIEW Journal > Download Files > March/April 2004 > the title of this article. Scroll to the bottom of the abstract page for the download link.
The standard Resource Reservations database has different forms for administrators to define sites and describe resources, and for users to make their resource reservations. The form for making reservations is called the “Reservation” form in earlier releases and “New Reservation” in Notes/Domino release 6; I’ll refer to both forms as the Reservation form, for convenience. To get a look at how a reservation request might appear to a user after the modifications in this article, see Figure 1.

To integrate equipment resources with rooms in Resource Reservations, we will perform these actions:

• Add a form to establish room attributes as keywords, plus a view that presents room attributes in a keyword list
• Customize the Resource form by adding a field in which end users or administrators can enter and manage the attributes of each resource
• Modify the Reservation form, adding a field that enables end users to select the resource attributes they require
• Substitute a more sophisticated search engine that adds resource attributes to the standard search criteria — location, capacity, and availability

Note!
You can choose to add searchable attributes for any kind of resource (room or non-room). For brevity, this article illustrates altering room resources and mentions what differences to make for non-room resources.

Let’s start the alterations to the Resource Reservations system by setting up the form and view to manage the list of room attributes.
Creating the “Attribute Profile” Form and View

There are several techniques you could use to facilitate keyword maintenance — for example, you could set up keyword lookups. For this solution, we will use something I call the “each keyword is stored in a separate document” technique, but feel free to substitute your favorite technique.

By creating a new form and view for the keywords that indicate the attributes of rooms (such as the kinds of equipment or amenities each room has), we can facilitate keyword management without changing the design of the Resource Reservations database. If you know for sure that the list of room attributes will never change, you could hard-code them into the properties of the related fields in the Reservation form. However, hard-coding the attributes requires programming assistance when attributes change. Since equipment is always changing with new technologies (telecommunications and audio visual, for example), you most likely want to allow easy changes to the equipment attributes.

To begin, create a form called “Attribute Profile” that has a field named “Attribute” for entering the name of an attribute. (You can see how the form looks...)

---

There are several techniques you could use to facilitate keyword maintenance. For this solution, we will use something I call the “each keyword is stored in a separate document” technique.
in the Designer client in Figure 2 and in the Notes client in Figure 3.)

Next, create a view to display all of the Attribute Profile documents. I adapted an existing view in Resource Reservations that includes the standard action buttons New Site, New Resource, New Reservation, and Delete Site. (For example, in Domino 6, you could start with the Site view or the Resource view.) The Resource and Reservation forms will use an @Dbcolumn call to this view for a listing of any room’s attributes. To provide an interface for administrators to manage the list of attributes, create a new action button called “New Attributes” that opens the Attribute Profile form. Figure 4 shows the view I created. After creating some Attribute Profile documents, your view should look similar to the Attribute Profiles view shown in Figure 5. The view should sort by its first column, “Attribute Name.”

**Modifying the Resource Form**

To allow administrators to designate or change the attributes describing a resource, you need to change the Resource form. Before starting, be sure to create a copy of the original Resource form and give it a name that you’ll recognize in the future (“Original Resource” for example). If something goes wrong, you can revert to this backup form.

On the Resource form, add a field named “Attributes” with the following properties:

---

**Figure 2** The “Attribute Profile” Form in the Designer Client

![Attribute Profile Form in Designer Client]

**Figure 3** The “Attribute Profile” Form in the Notes Client

![Attribute Profile Form in Notes Client]
Customizing Resource Reservations — Enable Users to Schedule Rooms Based on Room Attributes

- **Type** — Checkbox, Editable, Allow multiple values
- **Choices** — Use a formula for choices:

  ```
  @Dbcolumn("Notes":"NoCache";@Dbname; "Attributes";1)
  ```

The Choices formula uses `@Dbcolumn` to get the list of attributes from the Attribute Profiles view.

If you only intend rooms (not non-room resources) to have attributes, then add an appropriate hide-when formula to the Attributes field and label that hides these elements when the user is looking for...
non-room resources. Non-room resources have a Resource Type of 2, and room resources have a Resource Type of 1. Figure 6 shows a suitable hide-when formula.

A document created with the updated Resource form should now look something like the one shown in Figure 7, with the list of attributes at the bottom.

**Modifying the “Reservation” Form**

Now we’ll make adjustments to the Reservation form that will allow the end user to search on the new attributes. Again, be sure to make a copy of the original form for safety’s sake.

**Add an Attributes Field**

To accommodate the attributes from the Resource form, in the search section of the Reservation form, create a field named “Attributes” with the following properties:

- **Type** — Checkbox, Editable, Allow multiple values

- **Choices** — Use a formula for choices:

  ```javascript
  @Dbcolumn("Notes":"NoCache";@Dbname;"Attributes";1)
  ```

Again, note the choices formula that gets the attributes from the Attributes Profile view.

**Figure 6** The “Attributes” Field

**Hide-When Formula**

**Figure 7** A Modified “Resource” Document
The Attributes field provides an interface for users to select which attributes they require. If you only intend to offer searching for rooms (not non-room resources) by attributes, then add an appropriate hide-when formula to the two design elements, the Attributes field and label, so they are hidden when the user chooses a non-room resource. (Remember, non-room resources have a Resource Type of 2; room resources have a Resource Type of 1.) Figure 8 shows a suitable hide-when formula.

The Reservation form should now look something like the form in Figure 9, which shows some default settings.

**Creating the New Search Button**

There are two Search buttons on the standard Reservation (or New Reservation) form. The first Search button searches for a room that is available at a specific time, location, and capacity; it is hidden when the user chooses a non-room resource. The second Search button searches for a non-room resource available at a specific time, location, and capacity; it is hidden when the user chooses a room resource.

In this article, we are only going to enable searching for attributes of room resources, so we want to hide the original room Search button and add a custom room Search button on the form between the first and second Search buttons. The

---

**Figure 8** The Hide-When Formula to Apply to the “Attributes” Field and Label

<table>
<thead>
<tr>
<th>Field</th>
<th>Hide paragraph when document is</th>
<th>Hide paragraph if formula is true</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notes R4.5 or later</td>
<td>Previewed for reading</td>
<td>Resource Type=“1”</td>
</tr>
<tr>
<td>Web browsers</td>
<td>Previewed for editing</td>
<td></td>
</tr>
<tr>
<td>Mobile</td>
<td>Opened for reading</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Printed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Copied to the clipboard</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Embedded</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 9** The Modified “Reservation” Form
custom Search button will search for all rooms that are available at a specific time, location, and capacity and that include the user’s selected attributes. For example, to find a room that can handle a slide show, the user of the form in Figure 9 simply would select “Projection Unit” as a required room attribute and then click the Search button.

To ensure that you have the correct hide-when formula for the custom button, start the custom Search button with a copy of the original room Search button. Make a copy of the first Search button, click Enter to create a new line beneath the first button, and then paste the copy of the first button on the new line. To easily identify which is the custom Search button and which is the original Search button, add hidden text next to the original button and below the new custom button. You need to change the hide-when formula for the original Search button so that it is always hidden. You can see an appropriate formula and button identities in Figure 10.

The next section explains the new search-engine code for modifying the Search button.

**Creating the New Search Engine**

The Resource Reservations database has a search engine written in the Lotus formula language. The engine returns results as a listing, which works fine because the user is searching for one date, one capacity, and one location. However, searching by multiple room attributes is more efficient with code that loops through the listing and looks for each attribute, so I recommend using LotusScript for the new search function.

You can see the entire customized search-engine replacement (available in the download) in Figure 11. I’ll explain the code’s major points.

The customized Search button code uses an undocumented @function — @FindFreeResource — to find all available rooms that meet the user’s capacity, time,
and location requirements. Let’s go over briefly how the LotusScript call to @FindFreeResource is constructed.

To begin narrowing the search, the code uses a key argument to find just the rooms at the site the user requested in the Reservation document. The code in line 30 of Figure 11 assigns the Site variable with the value from the Reservation document. For rooms of that particular site, line 31 constructs this key argument string:

```
"CN=~/O=“+Site
```

(To return non-room resources based on attributes, we would use
```
"CN=~/"OU=ResourceCategory/O=“+Site
```
as the key argument in an equivalent line of code.)

The code in line 33 uses the Evaluate() routine to call @FindFreeResource. To further narrow the search, this code takes the room data from the string returned by the key argument and evaluates whether the rooms have field values that match the values in the Reservation document for the type of resource, start date and time, end date and time, and capacity.

Next, the code in lines 35 – 59 loops through the listing of available rooms returned by @FindFreeResource, comparing the attributes in the Resource Profile document for each room against the required attributes. In lines 61 – 70, the names of all rooms that match the attribute requirements are placed into an array.

Finally, the last block of code presents the search results to the user. If the search had results that match the user’s requirements, the code in line 74 presents the user with a prompt to display the list of rooms. If the user clicks “OK,” the code in lines 75 – 77 display the results for the user. If a search has no results, line 79 informs the user, “No rooms meet the required date, time, capacity, and room attributes.”

**Figure 11  The Code for the Customized “Search” Button**

```
1. Sub Click(Source As Button)
2. Dim s As New NotesSession
3. Dim w As New NotesUIWorkspace
4. Dim db As NotesDatabase ‘ Current Resource Reservations database
5. Dim UIDoc As NotesUIdocument ‘Current new Reservation UI document
6. Dim doc As NotesDocument ‘Current new Reservation backend document
7. Dim RequiredAttribs As Variant ‘List of required equipment
8. Dim Capacity As Integer ‘Required capacity
9. Dim ResourceType As String
10. Dim ResourceView As NotesView ‘ Resource view in current Resource Reservations database
11. Dim Site As String ‘Required Site
12. Dim fnn As notesname ‘Notesname for converting resources found to notesname format
13. Dim server As String
14. On Error Resume Next

15. ‘Initialize everything.
```

(continues on next page)
16. Set db = s.CurrentDatabase
17. Set UIDoc = w.CurrentDocument
19. Set SiteView = db.GetView("Resources")

20. RequiredAttribs = doc.Attributes
21. LowEquip% = Lbound(RequiredAttribs)
22. UpEquip% = Ubound(RequiredAttribs)
23. 'Set the number of required attributes.
24. If RequiredAttribs(0) = "" Then
25. numattribs% = 0
26. Else
27. numattribs% = (UpEquip% - LowEquip%) + 1
28. End If

29. 'Find all available resources that are of the correct site, availability, and capacity.

30. Site = doc.Site(0)
31. doc.Keyarg = "CN="/"O=" + Site
32. 'Find all available rooms for the desired site with the specified date/time and capacity requirements.
33. ResNameCheck = Evaluate("@FindFreeResource(Keyarg;ResourceType;StartDateTime;EndDateTime;Capacity;50)", doc)

34. 'Loop thru each resource to find all the resources that have the appropriate equipment and capacity.

35. matchcount% = 0
36. For all n In resnamecheck
37. Set fnn = New NotesName(n)
38. Set curdoc = SiteView.GetDocumentByKey(fnn.common)
39. If Not(curdoc Is Nothing) Then
40. attribs = curdoc.Equipment
41. foundattribs% = 0
42. AMatch% = 1 'By default all attributes are found.
43. If numattribs% = 0 Then 'If no attribute requirements then proceed.
44. 45. Else
46. Else check each attribute requirement against what the room provides.
47. For x% = LowEquip% To UpEquip%
48. For y% = Lbound(attribs) To Ubound(attribs)
49. If RequiredAttribs(x%) = attribs(y%) Then
50.          foundattribs% = foundattribs% + 1
51.    End If
52. Next
53. Next
54. End If
55. 'Check to make sure that we have a match for each required attribute.
56. If foundattribs% = numattribs% Then
57. Else
58.          AMatch% = 0 'Attributes not found.
59.    End If
60.
61. 'Place all matching resources in the matches array.
62. 'If Attribute found then add this resource to the array of matches.
63. If AMatch% = 1 Then
64.          matchcount% = matchcount% + 1
65.    Redim Preserve matches (1 To matchcount%) As String
66.    Set fnn = New notesname(curdoc.ResourceName(0))
67.    matches(matchcount%) = fnn.abbreviated
68.    End If
69.    End If
70. End Forall
71. 'Present all the matches for the user to select from and return the selected match.
72. If matchcount% > 0 Then
73.    'Display a prompt of all matches for the user to select from.
74.    ResourceName = w.Prompt( PROMPT_OKCANCELLELIST, "Search Results", "The following rooms are available for the required date, time, capacity and room attributes. Make your selection:" ; ";", matches )
75.    doc.resourcename = resourcename
76.    uidoc.Reload
77.    uidoc.Refresh
78. Else
79. test = w.prompt(PROMPT_OK,"Search Results","No rooms meet the required date, time, capacity and room attributes.")
80.    End If
81. ErrorRoutine:
82. End Sub
That completes the customizations to the Resource Reservations database. As always, test the new code thoroughly prior to implementing the modified database in a production environment.

**Extending the Solution to Calendar and Scheduling**

The modifications necessary to extend the attribute search functionality from our altered Resource Reservations database to the Mail template are far simpler than those we had to make to the Resource Reservations database. We only need to modify one subform to add the attribute selection list and adjust the search code.

While modifying the standard Mail template is not hard, you must exercise extreme caution when undertaking the task. A faulty Mail template can cause an organization great pain. If you have any hesitations about modifying the Mail template, then don’t do it — it is far better to be safe than sorry. If you decide to go ahead and make this additional change, make sure that you thoroughly test the enhanced Mail database prior to implementing it in a production environment.

End users reserve rooms and resources from the Mail template using the (_Calendar Entry) form. For example, the entry form for meetings (shown in Figure 12) has a “Find Room or Resource” button. This button launches the (Room Resource Scheduler) subform, which is where you make all of the modifications that allow users to include attributes in their searches for available rooms. You can see the subform in Figure 13. It’s similar to the Reservation form in the Resource Reservations database, so you can follow the same process and re-use much of the search code you already have there.

*Proceed with Caution!*

It is good programming practice to make a copy of each database element that will be modified and store the copies locally in the modified version of the database so that you can quickly revert back to the original design.

In addition, because the solution for Calendar and Scheduling in Mail involves customizing a standard Notes/Domino template, I recommend that you maintain a pure version of the template for recovery purposes.

**Figure 12**  The “Find Room or Resource” Action Button on the (_Calendar Entry) Form
**Architectural Assumptions**

Before we start, let’s review the architectural assumptions behind this extended solution:

- As recommended by IBM/Lotus, your company follows the best practice of establishing Resource Reservations databases on non-Mail servers. This means that users coming through Calendar and Scheduling are on different servers than the servers handling the Resource Reservations systems.

- There can be several or even many Resource Reservations databases. However, each Resource Reservations system is in the same time zone as the resource sites it supports, or it specifies the time zone of each resource (release 6 only). One site’s resources are not split among different Resource Reservations systems.

- Users will make reservations for meeting rooms from their homes or remote sites, as well as from their desktops. For example, a marketing manager in New Jersey can be visiting a division office in Ohio while planning a meeting at another corporate site, say in London or Singapore.

If your Resource Reservations architecture is quite large, you may need to tweak the basic solution to improve performance. The sidebar on page 17 offers some suggestions that can improve the user’s experience, especially in widely distributed architectures.

**Modifying the (Room Resource Scheduler) Subform**

Begin by making a backup copy of the (Room Resource Scheduler) subform in the Mail template. Name it something easy to find again, such as “(Room Resource Scheduler) — Original.”

The code in the new Search button will need two paths — one to the correct Resource Reservations system, and the other to the correct user calendar. To provide users with the list of room attributes from the Resource Reservations system that covers the desired site, the code on the subform needs to match the desired site to the correct Resource Reservations database. To update the user calendar, the code needs the name of the user’s mail server.
There are several ways we can find the values for the paths — the approach I suggest now is to add some hidden fields to the (Room Resource Scheduler) subform that have formulas for obtaining the necessary values from the Domino Directory on the user’s mail server. The best way to obtain the name of the user’s mail server is to use the @NameLookup function; when given a user name and a field name, this function returns the value of that field from the user’s Person document. (Since the user may be accessing Calendar and Scheduling remotely, you can only be sure of the user’s mail server if you obtain its name from the user’s Person document, which of course is replicated in all of the Domino Directories.) Once you have the path to the user’s mail server, use @DbLookup to find the server and file names of the Resource Reservations system that handles the desired site.

Place three hidden fields at the top of the (Room Resource Scheduler) subform — TmpMailServer, TmpSiteServer, and TmpSiteFileName. The formulas for obtaining the values for these three fields are presented in Figure 14.

Next, in the section of the (Room Resource Scheduler) subform where search criteria are entered, create a field called “Attributes” with the following properties:

- **Type** — Dialog list, Editable, Allow multiple values
- **Choices** — This formula makes use of the hidden fields we just created to find the room attributes on the local Resource Reservations system:

  ```plaintext
  @Dbcolumn("Notes":"NoCache";tmpSiteServer"tmpSiteFileName;"Attributes";1)
  ```

Figure 14 Hidden Fields for the (Room Resource Scheduler) Subform

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Type</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TmpMailServer</td>
<td>Text-Computed</td>
<td>@NameLookup([EXHAUSTIVE];@username;&quot;MailServer&quot;)</td>
</tr>
<tr>
<td>TmpSiteServer</td>
<td>Text-Computed</td>
<td>tmp := @Subset(@DbLookup(&quot;notes&quot;;&quot;nocache&quot;;tmpmailserver: &quot;names.nsf&quot;;&quot;($Rooms)&quot;;tmpsite;5);1); @If(@IsError(tmp);&quot;&quot;;tmp)</td>
</tr>
<tr>
<td>TmpSiteFileName</td>
<td>Text-Computed</td>
<td>tmp := @Subset(@DbLookup(&quot;notes&quot;;&quot;nocache&quot;;tmpmailserver: &quot;names.nsf&quot;;&quot;($Rooms)&quot;;tmpsite;6);1); @If(@IsError(tmp);&quot;&quot;;tmp)</td>
</tr>
</tbody>
</table>

- **Options** — Refresh choices on document refresh

The Attributes field provides an interface for users to select the attributes they require in a room or resource. Unless you expand the Resource Reservations customizations to include searching for non-room resources by attributes, you must add an appropriate hide-when formula to this Attributes field and field label, in the manner shown in Figure 15.

The (Room Resource Scheduler) subform should now present end users with a combo box of resource-attribute keywords like the one shown in Figure 16.
Customizing Resource Reservations — Enable Users to Schedule Rooms Based on Room Attributes

If your company has implemented a globally dispersed Resource Reservations architecture, and if your network is not up to supporting instant data exchanges, you may find that Calendar and Scheduling users accessing the enhanced Resource Reservations databases at remote sites experience performance problems. To improve performance in global architectures, you could complete more of the resource searching locally by taking advantage of the Mail-in database documents that are created as part of the Resource Reservations system.

Each Resource document has its standard content duplicated in a Mail-in database document in the Domino Directory. The Domino Directory automatically replicates the Mail-in database documents. By adding the attributes for each resource to its Mail-in database document, you would reduce the traffic between remote sites because a user’s attribute selection could be matched to the attributes in the local resource Mail-in database documents. Once the attributes are available locally, use an @NameLookup call on the (Room Resource Scheduler) subform to return the attribute values for resources from their Mail-in database documents. The @NameLookup function will find the Mail-in database documents in the closest Domino Directory.

Here are two alternatives for including the resource attributes within the Mail-in database documents:

- In each Resource Reservations database, create a scheduled "synchronization" agent to populate an Attributes field in each Mail-in database document with the attribute values in the respective Resource document. Pick a schedule to run the agent (daily/weekly/monthly) that covers how frequently your organization changes resources. In this alternative, during a search for resources, the @NameLookup on the subform would return the values in the Attributes fields of the Mail-in database documents.

- If you are really daring, instead of creating an agent, you could piggyback on the AdminP process that updates the Mail-in database documents for resources. One way to implement this is to alter the AdminP code so that it passes the values from the Attributes field of the Resource document via the Description field. In this case, during a search for resources, the @NameLookup on the subform would return the values in the Description fields of the Mail-in database documents.

Calendar and Scheduling Performance Tip!

Figure 16

The Expanded (Room Resource Scheduler) Subform
**Modifying the Search Button**

Finally, incorporate searching for resource attributes into the Search button’s actions. The LotusScript code for the (Room Resource Scheduler) subform’s Search button is listed in **Figure 17**. The complete code is in the download. It is almost the same as the code for the Search action button you added to the Reservation form. The differences in the subform’s Search code include these actions:

- Connect to the Resource Reservations database that covers the site selected by the user.
- Return the list of rooms that satisfy the search to the (Room Resource Scheduler) subform.

**Figure 17  The Code for the New “Search” Button on the (Room Resource Scheduler) Subform**

```plaintext
Sub Click(Source As Button)
    Dim s As New NotesSession
    Dim w As New NotesUIWorkspace
    Dim db As NotesDatabase ' Resource Reservations database for the site selected
    Dim UIDoc As NotesUIDocument 'Current new calendar entry UI document
    Dim doc As NotesDocument 'Current new calendar entry backend document
    Dim RequiredAttribs As Variant 'List of required attributes
    Dim Capacity As Integer 'Required capacity
    Dim ResourceType As String
    Dim ResourceView As NotesView ' Resource view in Resource Reservations database
    Dim Site As String 'Required Site
    Dim fnn As notesname
    Dim server As String

    On Error Resume Next

    'Initialize everything.

    Set UIDoc = w.currentdocument
    Set doc = uidoc.document
    'Get the Resource Reservations database of the Site selected.
    Set db = s.getdatabase(doc.tmpsiteserver(0),doc.tmpsitefilename(0))
    Set SiteView = db.getview("Resources")

    RequiredAttribs = doc.Attributes
    LowEquip% = Lbound(Equip)
    UpEquip% = Ubound(RequiredAttribs)
```

Customizing Resource Reservations — Enable Users to Schedule Rooms Based on Room Attributes

'Set the number of required attributes.
If RequiredAttribs(0) = "" Then
    numattribs% = 0
Else
    numattribs% = (UpEquip% - LowEquip%) + 1
End If

'Find all resources that are of the correct site and capacity.
'Loop thru each resource finding all the resources that have the appropriate attributes.
'Place all matching resources in the matches array.

tmpStartDateTime=Evaluate("@TimeMerge(@Date(@Year(StartDate); @Month(StartDate);
    @Day(StartDate));@Time(@Hour(StartTime); @Minute(StartTime); @Second(StartTime));
    StartTimeZone);","doc")

tmpEndDateTime=Evaluate("@TimeMerge(@Date(@Year(EndDate); @Month(EndDate);
    @Day(EndDate));@Time(@Hour(EndTime); @Minute(EndTime); @Second(EndTime));
    EndTimeZone);","doc")

doc.tmpStartDateTime = tmpStartDateTime
doc.tmpEndDateTime = tmpEndDateTime

Site = doc.tmpsite(0)
doc.Keyarg = "CN=*/O="+Site

'Find all available rooms for the specified date/time, site, and capacity requirements.

ResNameCheck=Evaluate("@FindFreeResource(Keyarg;tmpResourceType;tmpStartDateTime;
    tmpEndDateTime;tmpCapacity;50)",doc)

matchcount% = 0
Forall n In resnamecheck
    Set fnn = New fontsizename(n)
    Set curdoc = Siteview.GetDocumentByKey(fnn.common)
    If Not(curdoc Is Nothing) Then
        attribs = curdoc.Attributes
        foundattribs% = 0
        AMatch% = 1 'By default all attributes are found.
        If numattribs% = 0 Then 'If no attribute requirements then proceed.
            Else 'Else make sure to check each attribute requirement against what the room provides.

(continues on next page)
Figure 17 (continued)

For x% = LowEquip% To UpEquip%
    For y% = Lbound(attribs) To Ubound(attribs)
        If RequiredAttribs(x%) = attribs(y%) Then
            foundattribs% = foundattribs% + 1
        End If
    Next
Next
End If

' Check to make sure that we have a match for each required attribute.
If foundattribs% = numattribs% Then
Else
    AMatch% = 0 'Equipment not found
End If

'If Attribute found then add this resource to the array of matches.
If AMatch% = 1 Then
    matchcount% = matchcount% + 1
    Redim Preserve matches (1 To matchcount%) As String
    Set fnn = New notesname(curdoc.ResourceName(0))
    matches(matchcount%) = fnn.abbreviated
End If
End If
End Forall
If matchcount% > 0 Then
    doc.tmpsearchresults = Str(matchcount%) + " Room(s) are available at this time"
    If matchcount% = 1 Then
        doc.tmpResourceNameSingle = matches
        doc.tmpResourceName = matches
        doc.tmpResourceNameBlank=""
        uidoc.Refresh
        uidoc.Reload
    Else
        doc.tmpresourcename = matches
        doc.tmpresourcenamesingle = ""
        doc.tmpresourcenameblank=""
    End If
Customizing Resource Reservations — Enable Users to Schedule Rooms Based on Room Attributes

- Calculate the meeting start date and time and the end date and time and posts those values to the user’s Calendar Entry document in a Domino-recognizable date/time format.

```vba
uidoc.Refresh
uidoc.Reload
End If
Else
doc.tmpresourcenameblank = "No rooms meet the required date, time, capacity, and room attributes."
doc.tmpsearchresults = "0 Room(s) are available at this time."
doc.tmpresourcenamesingle = ""
doc.tmpresourcename = ""
uidoc.Reload
uidoc.Refresh
End If
ErrorRoutine:
End Sub
```

**Note!**

As always before starting a modification, make a copy of the original Search button code on the (Room Resource Scheduler) subform, paste it directly below the original button, and rename it to indicate that it is the original.

Make sure to change the hide-when formulas so that the proper button is displayed. To be consistent with the Resource Reservations customizations that add attribute searching to rooms only, the non-room Search button on this form should have a hide-when formula that hides this new button when searching for a non-room resource (when tmpResourceType = "2").

Once again, if you decide to implement these customizations in your environment, please be sure to thoroughly test the new system before moving it into production.

Now your Calendar and Scheduling users should have the same ability to find and schedule rooms based on the attributes of the room that you’ve given to your Resource Reservations users.

**Conclusion**

Including attributes in searches for rooms or resources makes life much simpler for everyone — users and administrators alike. End users spend less time looking for rooms with the correct resources, and meetings proceed more efficiently when the meeting room has the attributes crucial to the meeting’s progress, like whiteboards or teleconferencing equipment.
You can extend the technique I’ve shown you to include searching for non-room resources by their attributes or perhaps add functionality that solves other business problems. If you try this solution, please let me know how it goes.

In the next article, I’ll present a solution for moving resources from one Resource Reservations database to another.

Mark W. Rose is an IT Architect with IBM Software Services for Lotus (ISSL) in Philadelphia, Pennsylvania. He is fluent in all IBM Lotus Software products and technologies, ranging from Sametime and QuickPlace customization to messaging architectures. Mark has worked with Lotus Notes and Domino since the early 1990s. He focuses on application development and enterprise integration. Mark has customized the Resource Reservations database for several IBM clients over the last few years. His latest certifications include IBM Certified Application Developer and Advance System Administrator for Lotus Notes and Domino 6. You can contact Mark at mark_rose@us.ibm.com.