by
John
Chamberlain

Level: Beginner
Works with: Domino 5.0.6a
Updated: 05/01/2001

The Domino Web server has a special URL, http://myserver/?OpenServer, which generates a page containing a list of all the databases on the server. The database names are active links, so you can open a database just by clicking a name. This is a convenient shortcut for administrators or designers working on a Web site. However, the OpenServer URL isn't really appropriate for the public users of your site, for two reasons:

- Authorization for the OpenServer URL is controlled by the Server document setting "Allow HTTP clients to browse databases." This setting is all-or-nothing; either everybody can browse, or nobody can. You rarely want the general public to see the entire list of databases on your site, but there's no way to limit the URL so that just a trusted set of users can see the list.
- The HTML generated for the database list is hard-coded. You can't modify the appearance of the page, or specify which databases should be displayed.

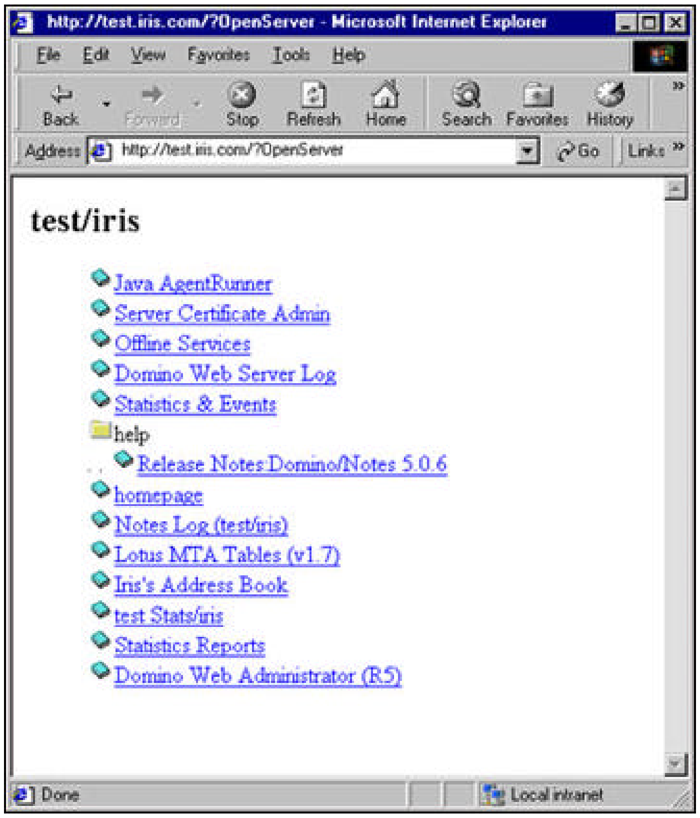
This article discusses a LotusScript agent that provides a flexible alternative to the OpenServer URL. You can copy the agent's script from the **Open Server agent sidebar** or download it in a text file from the **Iris Sandbox**. You can then use it to control who has access to the list of databases on your server and how that list appears.

This article assumes a basic understanding of Domino database design and administration. You can download the agent and use it as is, by following the instructions in the **Using the agent** section, below. An understanding of LotusScript will allow you to customize the agent as well.

The Open Server agent was developed and tested with Domino R5.0.6a, and should work with 5.0.6a and later releases.

How the agent works

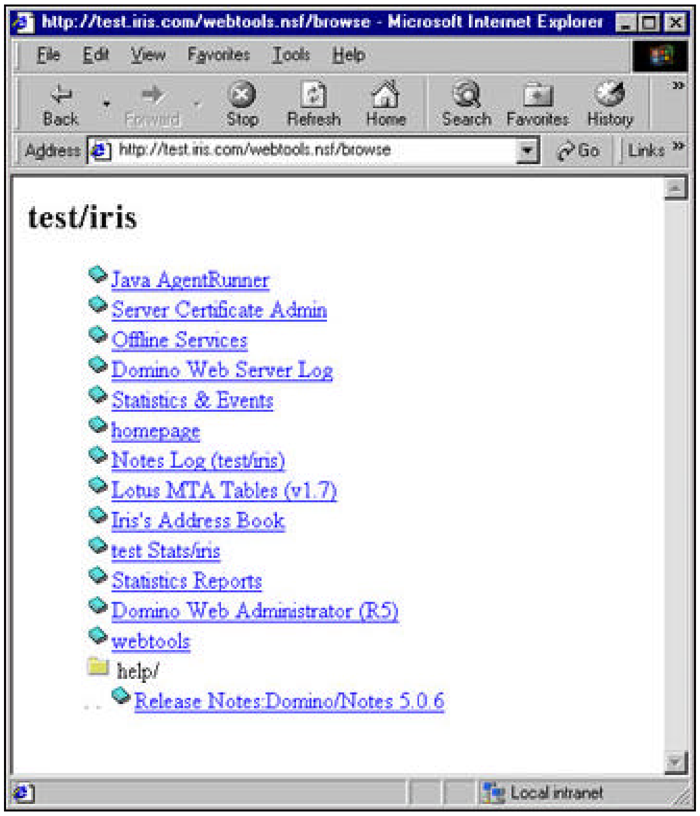
The OpenServer URL displays a list of all databases on the server, with active links to those databases:


http://test.iris.com/?OpenServer - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History
Address http://test.iris.com/?OpenServer
Go Links
Done
Local intranet

test/iris

- Java AgentRunner
- Server Certificate Admin
- Offline Services
- Domino Web Server Log
- Statistics & Events
- help
  - Release Notes:Domino/Notes 5.0.6
- homepage
- Notes Log (test/iris)
- Lotus MTA Tables (v1.7)
- Iris's Address Book
- test Stats/iris
- Statistics Reports
- Domino Web Administrator (R5)

The Open Server agent output mimics the OpenServer URL. In fact, it's a slight improvement, with all databases in a directory sorted before all subdirectories:


http://test.iris.com/webtools.nsf/browse - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History
Address http://test.iris.com/webtools.nsf/browse
Go Links
Local intranet

test/iris

- Java AgentRunner
- Server Certificate Admin
- Offline Services
- Domino Web Server Log
- Statistics & Events
- homepage
- Notes Log (test/iris)
- Lotus MTA Tables (v1.7)
- Iris's Address Book
- test Stats/iris
- Statistics Reports
- Domino Web Administrator (R5)
- webtools
- help/
  - . . Release Notes:Domino/Notes 5.0.6

Because the agent resides in a database, you can restrict who can run the agent by modifying the database ACL, as described in the **Using the agent** section, below.

This agent makes extensive use of the LotusScript "list" facility. A list, like an array, is a collection of objects, but a list is more flexible than an array because it can grow and shrink as elements are added and deleted. Also, elements in an array are identified by numeric subscripts, but elements in a list are identified by tags, which can be meaningful text strings. Since most of the information we are dealing with in this agent is in the form of short text strings, lists are the ideal way to organize and access the information.

The most important data structure in the agent is the DirList class. This class represents a directory in the Domino data-directory hierarchy. The "level" variable indicates how deep a directory is in the hierarchy, with the Domino data directory itself at level 1, a child of the data directory at level 2, and so on. The "prefix" variable represents a text string that should be displayed in front of the database names in a directory to indent them properly. The "entries" variable is a list of the databases in the directory.

```
Class DirList
    Public level As Integer
    Public prefix As String
    Public entries List As String
End Class
```

All of the agent code is in the Initialize event.

```
Sub Initialize
'
' Certain system databases are hidden by OpenServer.
' We list these databases here. You can modify
' this list if you want.
    Dim hidden List As String
    hidden("admin4.nsf") = ""
    hidden("busytime.nsf") = ""
    hidden("setup.nsf") = ""
'
' Create a list of databases on the server
'
    Dim pathmap List As DirList
```

To get the list of databases, we use the LotusScript NotesDbDirectory class. Every database returned by this class (except for ones we want to hide) will be added as a DirList object to the "pathmap" list. The next few lines of code are a standard way to set up a loop to use the NotesDbDirectory class.

```
    Set session = New NotesSession
    Set dbdir = session.GetDbDirectory("")
    Set db = dbdir.GetFirstDatabase(DATABASE)
    While Not (db Is Nothing)
        filePath = db.FilePath
' Check if this database is supposed to be hidden
        If (Iselement(hidden(filePath))) Then Goto NextDb
' Change Windows backslashes to forward slashes
        While (Instr(filePath,"\") > 0)
            Mid(filePath,Instr(filePath,"\")) = "/"
        Wend
```

Windows and Unix return different slashes as path delimiters. By standardizing the direction of the slashes right away, the rest of the code only has to deal with one kind of slash.

```
' Get actual subdirectory path (everything before the file name)
        path = Strleftback(filePath,"/")
' Normalize the path to start and end with a slash
        If path = "" Then path = "/" Else path = "/" & path & "/"
        If Not Iselement(pathmap(path)) Then
' Add this path to the directory list
```

For elements in the pathmap list, the value is a new DirList object, and the tag is the path name (for example, /help).

```
            Set pathmap(path) = New DirList
            Set subdir = pathmap(path)
            subdir.level = 0
            subdir.prefix = ""
' Determine how many levels deep this path is
```

By counting the number of path delimiters, we determine how deep this directory is in the hierarchy. We need to do this to set the level and prefix variables correctly.

```
            segment = Instr(path,"/")
            While (segment > 0)
                subdir.level = subdir.level + 1
                If subdir.level > 1 Then subdir.prefix = subdir.prefix & ". . "
                segment = Instr(segment + 1,path,"/")
            Wend
        End If
        Set subdir = pathmap(path)
' Save the name to be displayed for this database
```

The databases in the directory are stored as elements in the "entries" variable of the directory's DirList object. The value of an element is the database title, and the tag is the database's file name.

```
        If (db.Title <> "") Then
            subdir.entries(db.FileName) = db.Title
        Else
            subdir.entries(db.FileName) = "(No title: " & db.FileName & ")"
        End if
NextDb:
        Set db = dbdir.GetNextDatabase
    Wend
```

At this point, we have a complete list of all the databases on the server (minus the ones we want to hide). Now we can work on generating the actual HTML response.

The OpenServer URL prints out the databases sorted without regard to paths, so databases in child directories are mixed up with databases in parent directories. For example, it will print out /events4.nsf, /help/readme.nsf, /names.nsf.

The Open Server agent improves on this because it prints all the databases in a directory before any child directories. For the same example, the agent prints /event4.nsf, /names.nsf, /help/readme.nsf. Notice that the root Domino data directory is represented as /, so it will always sort before any child directory (such as /help), guaranteeing that all databases in the root will be printed first.

```
'
' Generate HTML, looping over directories in sort order
'
    Set server = New NotesName(session.CurrentDatabase.Server)
    Print "<h2>"; serverName.Abbreviated; "</h2><ul>"
    Do
' Search the directory list for the lowest-sorting path
```

The following code is a straightforward way for finding the lowest-ordered string in a collection of strings.

```
        lowest = ""
        Forall p In pathmap
            If lowest = "" Then
                lowest = Listtag(p)
            Elseif Listtag(p) < lowest Then
                lowest = Listtag(p)
            End If
        End Forall
        If lowest = "" Then Exit Do ' no entries left in list, so we're done
        Set subdir = pathmap(lowest)
        If subdir.level > 1 Then
```

For all directories except the root, we need to indent directory and database names by adding the prefix. Notice that we use {} (braces) to delimit strings in the Print statements. Although strings are traditionally delimited by double-quotation marks, you can use braces or vertical bars instead; and it's a good idea to do so if the strings themselves contain quotes. Otherwise you have to remember to double the embedded double-quotation marks, which looks really strange.

```
' Generate HTML for subdirectory name

            Print Mid(subdir.prefix,4);
            Print {<img src="/icons/afolder.gif"border=0>};
            Print Strright(lowest,"/",5,subdir.level - 1);{<br>}
```

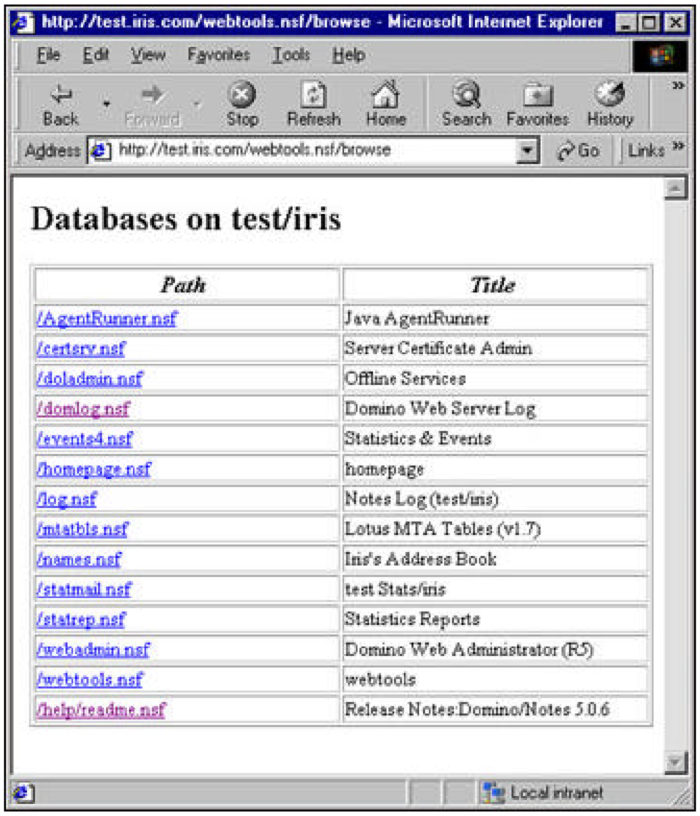
```
        End If
' Generate HTML for all entries in the database name list for this path
        Forall e In subdir.entries
            Print subdir.prefix;
            Print {<img src="/icons/abook.gif" border=0><a href="};
            Print lowest;Listtag(e);{?OpenDatabase">};e;{</a><br>}
        End Forall
'Done with this path, get it out of the list
        Erase pathmap(lowest)
    Loop

    Print "</ul>"
End Sub
```

Customizing the agent

Since the HTML output is completely controlled by the LotusScript code, you can easily customize the agent to change the layout of the page, to restrict browsing to certain databases or directories, or do anything else you want.

For example, you might change the layout of the page by displaying both the path and title for each database, and placing the output in a table. Here's what the new layout would look like:


http://test.iris.com/webtools.nsf/browse - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History
Address http://test.iris.com/webtools.nsf/browse
Go Links
Local intranet

Databases on test/iris

Path	Title
/AgentRunner.nsf	Java AgentRunner
/certsrv.nsf	Server Certificate Admin
/doladmin.nsf	Offline Services
/domlog.nsf	Domino Web Server Log
/events4.nsf	Statistics & Events
/homepage.nsf	homepage
/log.nsf	Notes Log (test/iris)
/mtatbls.nsf	Lotus MTA Tables (v1.7)
/names.nsf	Iris's Address Book
/statmail.nsf	test Stats/iris
/statrep.nsf	Statistics Reports
/webadmin.nsf	Domino Web Administrator (R5)
/webtools.nsf	webtools
/help/readme.nsf	Release Notes:Domino/Notes 5.0.6

To change to this layout, delete all the agent code after the comment "Generate HTML, looping over directories in sort order", and replace it with the following code:

```
'
' Generate HTML, looping over directories in sort order
'
    Set server = New NotesName(session.CurrentDatabase.Server)
    Print {<h2>Databases on }; server.Abbreviated; {</h2>}
```

```
' Generate HTML for the title row
    Print {<table width="100%" border=1>}
    Print {<tr><td width="50%">}
    Print {<div align=center><b><i>Path</i></b></div></td>}
    Print {<td width="50%">}
    Print {<div align=center><b><i>Title</i></b></div></td></tr>}
    Do
' Search the directory list for the lowest-sorting path
        lowest = ""
        Forall p In pathmap
            If lowest = "" Then
                lowest = Listtag(p)
            Elseif Listtag(p) < lowest Then
                lowest = Listtag(p)
            End If
        End Forall
        If lowest = "" Then Exit Do ' we're done
        Set subdir = pathmap(lowest)
' Generate HTML for all entries in the database list for this path
        Forall e In subdir.entries
            path = lowest & Listtag(e)
            Print {<tr><td width="50%"><font size=2>}
            Print {<a href="};path;{?OpenDatabase">};path;{</a></td>}
            Print {<td width="50%"><font size=2>};e;{</td></tr>}
        End Forall
 ' Done with this path, get it out of the list
        Erase pathmap(lowest)
    Loop
    Print {</table>}

End Sub
```

This code uses more HTML than the original agent in order to create the table. If you're not an HTML expert but you're handy with Domino Designer, then you can create a page in Designer with the layout you want, and check the page in a browser until it looks right. Then view the HTML source in the browser (in Microsoft Internet Explorer 5, choose View - Source), and copy-and-paste the HTML into your agent code.

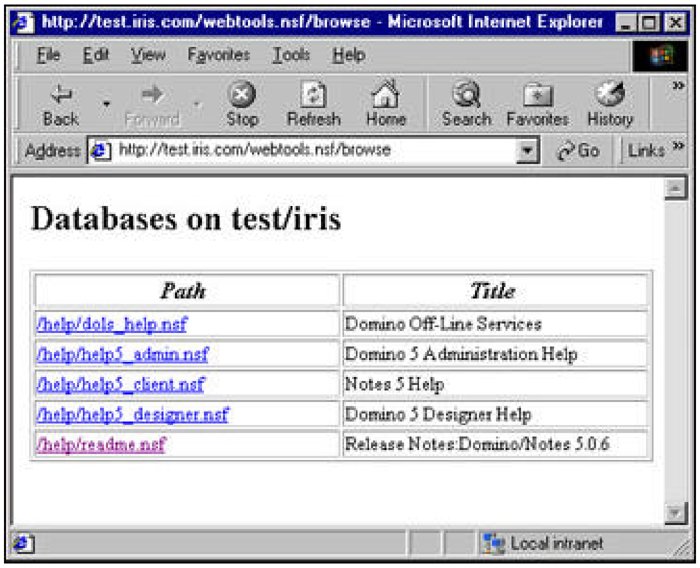
As another example, suppose you want to restrict the database list to a certain directory, for example, the Help directory. You can do this by adding one line of code to these lines:

```
    ' Normalize the path to start and end with a slash
    If path = "" Then path = "/" Else path = "/" & path & "/"
```

so that they read:

```
    ' Normalize the path to start and end with a slash
    If path = "" Then path = "/" Else path = "/" & path & "/"
    if Left(path,5) <> "/help" Then GoTo NextDb
```

If you make this change in addition to the table-layout change above, here's what you get in a browser. (We added some files to the directory to get a larger table.)


http://test.iris.com/webtools.nsf/browse - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History
Address http://test.iris.com/webtools.nsf/browse
Go Links
Local intranet

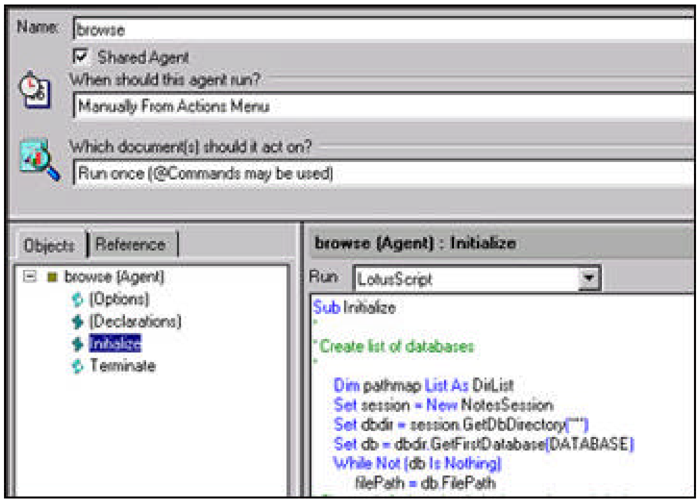
Databases on test/iris

Path	Title
/help/dols_help.nsf	Domino Off-Line Services
/help/help5_admin.nsf	Domino 5 Administration Help
/help/help5_client.nsf	Notes 5 Help
/help/help5_designer.nsf	Domino 5 Designer Help
/help/readme.nsf	Release Notes:Domino/Notes 5.0.6

Using the agent

To use the Open Server agent on your server, you place the agent in a database on the server. For easiest ACL control, create a new database to hold the agent. For example, create webtools.nsf. Then:

1. Open the database in Designer.
2. Go to the Agent pane and click the New Agent button.
3. Give the agent a simple name like Browse.
4. Click the Shared Agent checkbox.
5. Set "Which documents(s) should it act on?" to "Run once (@commands may be used)."
6. In the Run list, select LotusScript.
7. Select the Initialize event and delete the existing code in the event.
8. Copy the agent's script from the **Open Server agent sidebar** or download the text file openserveragent.txt from the **Iris Sandbox** and copy its contents.
9. Paste the agent's script into the event. (The Class definition will automatically move into the Declarations event.)



Name: browse
[✓] Shared Agent
When should this agent run?
Manually From Actions Menu
Which document(s) should it act on?
Run once (@Commands may be used)
Run: LotusScript

```
Sub Initialize

'Create list of databases

   Dim pathmap List As DirList
   Set session = New NotesSession
   Set dbdir = session.GetDbDirectory("")
   Set db = dbdir.GetFirstDatabase(DATABASE)
   While Not (db Is Nothing)
      filePath = db.FilePath
```

10. Save the agent. (If your ID doesn't have privileges to run agents on the server, resign the agent with an ID that does.)

On the server, start the http task. Open your browser and enter the URL http://myserver/webtools.nsf/browse. The database list for the server should appear.

Now set up the ACL for the agent's database (webtools.nsf, in this example) so that only those who should be able to browse the server have Reader access. A browser user only needs to have "Read public documents" access to the database to run the agent. If you want to restrict the agent to a certain set of users, you can change the ACL settings for -Default- (and Anonymous, if present) to No Access and then uncheck "Read/Write public documents." Then add the allowed users or user groups to the ACL and set them to Reader, or No Access with public access checked.

Finally, open the Server document and in the Basics section of the HTTP tab under the Internet Protocols tab, set the "Allow HTTP clients to browse databases" to No. Save the document and restart the http task. Now only those you have specified can run the agent and so view the server's database list.

About the Author
John Chamberlain is a Software Quality Architect on the Domino Web Server team.

Notes.net Iris Today

]

Open Server agent

Here is the complete code for the Open Server agent:

```
Class DirList
      Public level As Integer
      Public prefix As String
      Public entries List As String
End Class

Sub Initialize
'
' Certain system databases are hidden by OpenServer.
' We list these databases here. You can modify
' this list if you want.
      Dim hidden List As String
      hidden("admin4.nsf") = ""
      hidden("busytime.nsf") = ""
      hidden("setup.nsf") = ""
'
' Create a list of databases on the server
'
      Dim pathmap List As DirList
      Set session = New NotesSession
      Set dbdir = session.GetDbDirectory("")
      Set db = dbdir.GetFirstDatabase(DATABASE)
      While Not (db Is Nothing)
            filePath = db.FilePath
' Check if this database is supposed to be hidden
            If (Iselement(hidden(filePath))) Then Goto NextDb
' Change Windows backslashes to forward slashes
            While (Instr(filePath,"\") > 0)
                  Mid(filePath,Instr(filePath,"\")) = "/"
            Wend
' Get actual subdirectory path (everything before the file name)
            path = Strleftback(filePath,"/")
' Normalize the path to start and end with a slash
            If path = "" Then path = "/" Else path = "/" & path & "/"
            If Not Iselement(pathmap(path)) Then
' Add this path to the directory list
                  Set pathmap(path) = New DirList
                  Set subdir = pathmap(path)
                  subdir.level = 0
                  subdir.prefix = ""
' Determine how many levels deep this path is
                  segment = Instr(path,"/")
                  While (segment > 0)
                        subdir.level = subdir.level + 1
                        If subdir.level > 1 Then subdir.prefix = subdir.prefix & ". . "
                        segment = Instr(segment + 1,path,"/")
                  Wend
```

```
                End If
                Set subdir = pathmap(path)
' Save the name to be displayed for this database
                If (db.Title <> "") Then
                        subdir.entries(db.FileName) = db.Title
                Else
                        subdir.entries(db.FileName) = "(No title: " & db.FileName & ")"
                End If
NextDb:
                Set db = dbdir.GetNextDatabase
        Wend

'
' Generate HTML, looping over directories in sort order
'
        Set server = New NotesName(session.CurrentDatabase.Server)
        Print "<h2>"; server.Abbreviated; "</h2><ul>"
        Do
' Search the directory list for the lowest-sorting path
                lowest = ""
                Forall p In pathmap
                        If lowest = "" Then
                                lowest = Listtag(p)
                        Elseif Listtag(p) < lowest Then
                                lowest = Listtag(p)
                        End If
                End Forall
                If lowest = "" Then Exit Do ' no entries left in list, so we're done
                Set subdir = pathmap(lowest)
                If subdir.level > 1 Then
' Generate HTML for subdirectory name
                        Print Mid(subdir.prefix,4);
                        Print {<img src="/icons/afolder.gif" border=0>};
                        Print Strright(lowest,"/",5,subdir.level - 1);{<br>}
                End If
' Generate HTML for all entries in the database list for this path
                Forall e In subdir.entries
                        Print subdir.prefix;
                        Print {<img src="/icons/abook.gif" border=0><a href="};
                        Print lowest;Listtag(e);{?OpenDatabase">};e;{</a><br>}
                End Forall
 ' Done with this path, get it out of the list
                Erase pathmap(lowest)
        Loop

        Print "</ul>"
End Sub
```