Lotus. Developer Domain
The Technical Resource for Lotus software
Iris Today Archives

# Domino Import Service

by
Tara
Hall

Iris Today Archives

**Level:** Intermediate
**Works with:** Domino 5.0
**Updated:** 06/28/2000

*[Editor's note: Domino Import Service is no longer a supported product, so it is no longer available for download from the Lotus.com Web site or from any other IBM download site.]*

Have you invested money and time learning a particular Web authoring tool or HTML editor? Do you want to take advantage of Domino Web server features to enhance your existing Web site, but don't know Domino Designer? If you have some XML experience, then you can create Domino databases using the Domino Import Service 2.0, a utility that enables you to create Domino site databases (that is, Web sites hosted by Domino servers) without Domino Designer. You can use any Web authoring tool or HTML editor to create a Web site, then use Domino Import Service 2.0 to publish the Web site to a Domino server. With the Domino Import Service 2.0, Web developers with little or no knowledge of Domino Designer can develop Domino site databases.

## What is Domino Import Service 2.0?

The Domino Import Service 2.0 is a utility that publishes Web sites stored on a file system to a Domino server. Version 2.0 is a client-side application with a command-line interface. Previous versions of the Domino Import Service were server-side applications that worked with client -side applications, such as the Lotus Domino Fusion Connectors for NetObjects Fusion 3.0 and later, Lotus Domino Fusion Connector for NetObjects TeamFusion 3.0, and Domino Design Components 1.0 for Microsoft FrontPage 98. The Domino Fusion Connectors and Domino Design Components 1.0 used a Domino site component to store publishing information about a Web site. The components alone cannot publish a Web site to a Domino server, so they require you to install Domino Import Service on the Domino server to publish the Web site. Domino Import Service 2.0 uses a request file that you write instead of a Domino Site component to store publishing information. Because Domino Import Service 2.0 is a client-side application, when you publish the Web site with Domino Import Service 2.0, you can publish to any Domino server.

Though the Lotus Web Applications team continues to develop Domino Design Components, Domino Import Service 2.0 offers you the flexibility to choose your Web authoring tool or HTML editor. However, Domino Import Service 2.0 does not have the capability to create Domino design elements, such as Domino forms and views, which the Domino Design Components can create. The Domino Import Service 2.0 is capable of creating a fully functioning Domino site database to store a Web site. You can enhance the database with Domino forms and views using Domino Designer, but again, Domino Designer is not a requirement when using Domino Import Service 2.0.

Improvements in the Domino Import Service 2.0 not only offer more tool flexibility, but also enables you to publish to any Domino R5.0.x server, regardless of platform. Domino Import Service 2.0 eliminates the need to have experienced Domino Application Developers to create Domino site databases. Web developers without Domino Designer experience can create Web sites using tools with which they are familiar, and with some XML experience, can publish Web sites to Domino servers.

## How Domino Import Service 2.0 works

The Domino Import Service 2.0 publishes Web sites to new or existing Domino site databases. If you publish to a new database, Domino Import Service uses the Domino Design Components 2.0 template (ddc20.ntf) to create a local copy of the Domino site database.

The Domino site database stores all Web files, including HTML files, image files, Java applets, cascading style sheets (CSS), and so on. Domino Import Service stores HTML files in Domino pages. The Domino Import Service also translates URLs within the Web site to Domino URLs.

After the local copy of the Domino site database is created, Domino Import Service replicates the Domino site database from the local file system to a specified Domino server, removing the local copy once replication is complete. If you are a Lotus Notes user, you can publish the Domino site database locally, then replicate the database to a Domino server manually, thereby maintaining a local copy of the Domino site database.

## System requirements

Unlike earlier versions, the Domino Import Service 2.0 publishes Web sites without requiring a Lotus Notes client. To eliminate the Lotus Notes client requirement, the Domino Import Service includes the Notes Object Service (NOS). NOS provides the necessary dynamic link library (DLL) files to enable you to publish to a Domino server without having a Lotus Notes client. In addition to the features needed to create and to replicate a database, NOS also features support for Notes security. During installation of the Domino Import Service 2.0, if you do not have a Notes ID file, you are prompted to create one. As with any Notes ID, the Domino system administrator must certify the Notes ID file, and of course, you must have proper access to the Domino server to create databases on it.

Because of the inclusion of NOS, Domino Import Service 2.0 requires 25MB of disk space. However, at the time of this article was published, you must have enough disk space to create a local copy of the Domino site database that houses the Web site. The size of the Domino site database varies depending upon the size and number of Web pages and Web site assets that you publish. Domino Import Service creates the local database copy on the same drive where you install Domino Import Service 2.0.

To use the Domino Import Service 2.0, you must have Windows 95 with Microsoft Internet Explorer 4.x or later, Windows 98, or Windows NT 4.0. Because the Domino Import Service 2.0 takes advantage of certain Domino R5 design elements, Domino Import Service publishes to Lotus Domino R5.0.x servers only.

As stated earlier, Domino Import Service requires that you write a request file for publishing a Web site to a Domino server. The request file is written in XML, so knowledge of XML is required.

## Publishing with Domino Import Service 2.0

To publish with Domino Import Service 2.0, you create a request file using XML, then pass the request file to the Domino Import Service through a parameter on the Domino Import Service command-line. That parameter also invokes the Domino Import Service.

The request file is an XML file that contains all the publishing information needed to create the Domino site database. In the file, you specify which server to publish to, the name of the Domino database -- new or existing -- to store the Web site, error log file path, and other required information. An XML parser included with Domino Import Service validates all information in the request file against the document type definition file: imxsrequest.dtd.

## Building the request file

You can build the request file using any text editor, such as Windows Notepad, but make sure to save the file with an .XML extension. Before you publish, you can validate the XML request file using Microsoft Internet

Explorer 5.0. If you don't have Internet Explorer 5.0 installed, then validate your request file on the **STG XML validation form**.

The following is an example of a basic request file. A description of the file follows the example. This article describes only required elements and attributes needed for publishing. For a more comprehensive list of elements, attributes, and values, refer to the *Domino Import Service 2.0 Reference Guide* included with Domino Import Service 2.0. Not all values for required attributes are covered in this article.

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE imxsrequest SYSTEM "imxsrequest.dtd">
<imxsrequest>
<action specifier="publish" direction="import" qualifier="full"/>
<notesdb>
<server>Domino server name</server>
<path>directory\filename.nsf</path>
<title>Database title</title>
</notesdb>
<logfile level=″normal″>
<path>c:\directory\logfile.log</path>
</logfile>
<fsweb>
<path>c:\Web site directory</path>
<toppage type=″root″>index.html</toppage>
<filelist>
<url>c:\Web site directory\index.html</url>
<url>c:\Web site directory\images directory\image.gif</url>
<url>c:\Web site directory\filename.css</url>
</filelist>
</fsweb>
</imxsrequest>
```

Like any XML file, the request file begins with the XML element and the document type declaration, specifying the imxsrequest.dtd. The **<imxsrequest>** element is the parent element, so specify it at the beginning and at the end of the file as well as in the document type declaration. When specifying the imxsrequest.dtd, you may need to include the file path. The Domino Import Service expects the imxsrequest.dtd to reside in the same directory as the request file. When you install the Domino Import Service, you can find imxsrequest.dtd in the xml\lotus directory under the parent directory where you installed the Domino Import Service.

The <imxsrequest> element has five child elements, four of which are required and included in the example request file. The first child element is the **<action>** element. The **specifier** attribute determines that the Web site is to be published. The **direction** attribute determines that the files are imported to the server. The **qualifier** attribute determines that all files will be published, so if you publish a Web site more than once, all files are imported each time, not the modified files only. The specifier attribute has one other value available, but both the direction and qualifier attributes have only one value. All three attributes are required in the <action> element.

The **<notesdb>** element is parent to several other elements that provide information about where to publish the Web site.

Depending upon where you publish the Web site, the **<server>** element is optional. If you publish locally, you can omit the <server> element or include the element, but omit the value between the opening and closing tags like so: <server></server>. The <server> element specifies the Domino server name. Domino Import Service accepts the server name/organization/domain convention.

The **<path>** element specifies the file path and Domino database file name. The file path is relative to the Domino directory when published to a server, but is relative to the DIS\Data directory during a local publish.

The **<title>** element specifies the title of the Domino database to which you publish. You must include the <path> and <title> elements in the request file.

The **<logfile>** element and its child element **<path>** contain information about the error log file generated by the Domino Import Service. There are four values for the level attribute, all of which are described in the *Domino Import Service 2.0 Reference Guide*. The value normal is the default value. Normal returns all error and warning messages in the error log file. The <path> element specifies the drive, directory, and file name of the error log file.

The **<fsweb>** element and its child elements provide information about the Web site that you are publishing.

The **<path>** element  specifies the drive and directory where the Web site is stored on the local file system. The <path> element does not contain any files names.

The **<toppage>** element specifies the file name of the Web site home page. The root value in the type attribute indicates that the file specified is the root (or Home) page of the Web site.

The **<filelist>** element and its child element **<url>** specify the location of the Web site files on the file system. Specify the <filelist> element once and only once, but specify one <url> element for each Web site file, including the file named in the <toppage> element. The <url> element must be specified at least once. In the <url> element, enter the drive, directory, and file name of the file to be published.

You must specify every file in the Web site, unless the Web site files meet the following condition: if all Web site files are located in the directory specified in the <path> element or in subdirectories of that directory, then the <filelist> and <url> elements are optional. For instance, in our example request file, if all Web site files are located in C:\Web site directory or a subdirectory within the C:\Web site directory, then you can omit the <filelist> and <url> elements. Domino Import Service will publish the Web site files in the C:\Web site directory and any subdirectories.

For each Web site that you publish, you will need one request file.

## Submitting the request file to the Domino Import Service

Once you complete the request file, you can run Domino Import Service. To submit the request file to the Domino Import Service, open the MS-DOS Prompt, then enter the following command-line parameter:

```
c:\dis>dis.exe REQUEST=request.xml
```

In the example above, dis.exe invokes the Domino Import Service. The REQUEST parameter specifies the name of the request file to pass to the Domino Import Service. The example above assumes that the files dis.exe, request.xml, and imxsrequest.dtd reside in the same directory. If this is not the case, then make sure to specify the file paths to:

- dis.exe on the command line or type chdir drive:\directory at the command line prompt to change to the directory where dis.exe resides;

- the request file on the command line; and
- imxsrequest.dtd in the <!DOCTYPE> element.

Depending on where your files reside, you may need to perform all of the above tasks, or one or two of the tasks. To simplify the process, always store the request file and imxsrequest.dtd in the same directory.

If you have Notes in the environment path (as specified in your autoexec.bat file) the Domino Import Service fails to publish. The publishing failure occurs because rather than loading the NOS DLL files, Domino Import Service loads the Lotus Notes DLL files found in the environment path. The Notes DLL files do not support Domino Import Service.

If you run applications that require Notes in the environment path, then you may not want to permanently remove Notes from your path to use the Domino Import Service. You can temporarily set an environment path that excludes Notes. To do so, at the command line prompt, type set path=, then enter some or all of the environment path with the exception of the Lotus\Notes variable. After setting the path, submit the request file on the command line following the example above. Be aware that the path you set is available for the session. When you exit MS-DOS, the original path is reset.

## Domino Import Service and Domino Designer

After you publish the Web site, check the error log file to ensure that Domino Import Service published all files and that no errors occurred. If you published successfully, then you now have a Domino site database that users can view through a Web browser. As stated earlier, the Domino Import Service converts HTML files to Domino pages. If you open the Domino site database in Notes, likely the database will appear empty because the Domino Import Service did not create any Domino forms or documents to display in the database views.

If you have Domino Designer, open the Domino site database to view the pages. The Domino Import Service uses pass-thru HTML to preserve your coding. If you want, you can modify the pages in Domino Designer. If you publish the Web site again, the Domino Import Service will identify conflicts with the HTML files and the Domino pages that you modified. For instance, if you embedded a view on a page, the next time you publish the Web site Domino Import Service will recognize the change to the Domino page. To resolve the conflict, Domino Import Service creates a copy of the Domino page with the embedded view rather than overwriting the page and deleting the view. The Domino Import Service assigns the file name extension _copy1 to the page. You can then manually merge the designs changes into one page.

If you publish to a new Domino site database, then Domino Import Service applies the Domino Design Components 2.0 template to the database. You cannot specify another template when creating a new Domino site database. However, you can publish to an existing database that uses a template other than the Domino Design Components 2.0 template. Domino Import Service does not copy any design elements from the Domino Design Components template to the existing database, which can affect your launch options. When you publish to a new Domino site database, Domino Import Service launches the page specified in the <toppage> element of the request file.

If you publish to an existing database that does not use the Domino Design Components 2.0 template, you may need to change the launch option to open the Home page when viewed in a Web browser. To change the launch option in Notes or Designer, select the Domino site database, then choose File - Database - Properties. Select the Launch tab in the Database

Properties box. In the "When opened in a browser" field, select the "Open designated Page" option from the drop-down list. In the Page field, select the DDCRootPage. The DDCRootPage stores the name of the Home page specified in the request file. If you change the Home page in the request file and on the file system, the DDCRootPage redirects the Web browser to the correct page without you having to change the Web browser launch option in Designer again.

One approach for working with Domino Import Service is to treat the Web site files as source files, modify those files as necessary, then publish to a Domino site database. Any changes that you cannot make to the Web site files, such as embedding a view, make with Domino Designer.

## Conclusion

Domino Import Service simplifies building Domino site databases. You don't need to be an Application Developer with extensive Domino Designer experience to create databases. If you are a Web developer with HTML and XML experience, you can publish Web sites to a Domino server and start taking advantage of Domino features like replication and security.

**ABOUT TARA**
Tara Hall is a User Assistance writer for Lotus where she has worked for over a year. She is part of the Web Applications team and writes online help, programming guides, and release notes. She also is a member of the Notes UA Web team.

What do you think about this article?