Shepherd.FTP Service

J&J Computer Consulting Shepherd Server Publications

Overview

The Shepherd.FTP Service implements an RFC 959 compliant File Transfer Protocol. Through Shepherd Directory Services (SDS), Shepherd.FTP provides support for virtual hosts with unique IP addresses. Shepherd.FTP also includes support for complex access control within a virtual file system based on the SDS configuration.

Installation

Unzip the Shepherd.FTP distribution to the Shepherd installation directory. This will install a sample ftp configuration for the directory in samples\ftp.ldif, the Service initialization file in services\ftp.svc, and the dynamic link library for the Shepherd.FTP Service in services\ftp.dll. Before getting started with Shepherd.FTP, you need to create any additional attributes required by Shepherd.FTP and create a ShepherdService object in the directory. To create the additional attributes, use the atsetup option in the shepherd.ini file to create the attributes found in setup\ftp.attrib. Refer to "Shepherd Installation and Configuration" for further information on configuring a ShepherdService object for Shepherd.FTP.

Configuration

Once you have a ShepherdService object available and Shepherd.FTP is running, you can start to configure access to the file system. Shepherd.FTP uses two container objects for controlling the file system. Those are:

- Volume (vol=myftpvolume, c=US)
- DirectoryAlias (dir=myftpdir, vol=myftpvolume, c=US)

Volume objects are used to define the root of a file system. Each volume object contains a **hostDomain** attribute that points to any internetDomain object in which it should be available. When a user logs on, the server IP address is used to find the internetDomain object in SDS. If found, all Volumes pointing to that internetDomain object are included as subdirectories off of the root path. This configuration is called a virtual root. If only one Volume is found for an internetDomain, the root is set to the path specified by the Volume object.

DirectoryAlias objects map directories in the file system to containers in SDS. A DirectoryAlias object can represent a directory relative to the physical file system in the same way the object is relative to the Volume in SDS. A DirectoryAlias object can also bring outside directories into the user's view of the FTP file system to allow a virtual structure.

To correlate SDS to the local file system, both Volume and DirectoryAlias objects include a **path** attribute that identifies the physical path of the object. The contents of path should be a distinguished path using forward slash characters (/) as separators. Do not include a trailing slash.

In addition to containers, Shepherd introduces the **FileAlias** object for mapping files outside or inside the physical file system of the Volume or DirectoryAlias objects. FileAlias objects should contain a full path specification that does not include the file name or a trailing slash. They should also include a filename attribute for the file's name on the physical file system. The cn (common name) portion of the distinguished name of the object in SDS is what is show to FTP users.

Access control works the same on all of these objects as it does within SDS. However, it's important to remember that the name of the Shepherd.FTP service object must be specified in the ACL. Wildcard service names in ACLs are discouraged due to the behavior differences between services. For instance, Shepherd.Admin with write access lets the user modify attributes in SDS whereas Shepherd.FTP with write access lets the user upload a file into the directory or file specified.

Log Files

Shepherd.FTP attempts to log several different sets of data and can use several log files. The logFile attribute of the ShepherdService object can point to one or more Log objects, but those objects must be identified appropriately with the **logType** attribute. Options for the logType attribute include:

- audit
- error
- debug

The audit log tracks the date, server IP address, client IP address, user name, number of bytes transferred, and time connected.

The error log has no specific format and is primarily used to log failed failures in the Shepherd.FTP service unrelated to a specific user action.

The debug log tracks all commands issued by a client system and the Shepherd.FTP server responses for those commands. The client IP address and user name are included for rebuilding the FTP conversation between the client and the Shepherd.FTP service.