Lotus Developer Domain
The Technical Resource for Lotus software
LDD Today

A tour of Sametime 3.0 servlets

Level: Intermediate
Works with: Sametime 3.0
Updated: 03-Sep-2002

by
Isaac
Hands

Since version 2.0, the Sametime server has relied on Java servlet technology to handle server configuration and administration. With Sametime 3.0, the number of servlets running on the server triples. With many more servlets performing many more functions, Sametime administrators should understand what these servlets do, how they are accessed, and how they are protected. With a working knowledge of servlet technology, and more specifically, a working knowledge of the Sametime servlets, you will better understand how to deploy and maintain a Sametime 3.0 server.

This article assumes that you understand basic Java programming principles, but not necessarily Java servlets. You should know the basics of setting up a Domino 5 server and understand the concept of single sign-on authentication as described in the Domino 5 release notes. Also, a working knowledge of HTTP will be helpful. The primary purpose of this article is to increase your understanding of how the Sametime 3.0 server operates so that you can become a better Sametime administrator or applications developer.

Overview of servlets

Java servlets are classes that implement the servlet interface defined in the Sun servlet API specification. Unlike standalone Java applications, servlets do not have a "static void main" entry point, but instead rely on a servlet manager application for startup, shutdown, and service calls. The most common type of servlets, HTTP servlets, handle HTTP traffic by reading and writing HTTP requests and responses through helper classes found in the servlet API. For example, an HTTP server that is working with a servlet manager will typically inspect all HTTP traffic for URLs containing a keyword that should direct the data to a servlet. If the keyword is found, then the server will pass the HTTP request to its servlet manager. Next, the servlet manager reads the HTTP header information from the request and searches for a servlet name in the URL to identify which servlet should handle the request. After the servlet manager identifies a servlet, it is handed the HTTP header data along with the input and output streams for processing.

Servlet managers

In the days before J2EE platforms, most servlet managers were pretty simple. A convenient way to understand pre-J2EE servlet managers is to think of them as standalone Java applications that have a thread pool and a list of dynamically loaded classes, namely, their known servlets. Typically, an HTTP server will create one instance of a servlet manager on startup, and then destroy it on shutdown. As the manager starts up, it loads a list of servlet classes from a property file, creates a single instance of each, and then calls their init() methods with some environment data. Next, as HTTP requests come into the servlet manager, the manager calls into the service() method of each servlet object using different threads from its thread pool. When the servlet manager shuts down (that is, when the HTTP server shuts down), each servlet is given a chance to clean up through a destroy() method before it is deleted.

With the advent of J2EE servers, servlet managers have become much more complex. They can be vertically and horizontally cloned, implement session affinity, and have multiple thread pools per process. The Sametime 3.0

server relies on the Domino 5 servlet manager, which follows the pre-J2EE description very closely. For the purposes of understanding Sametime 3.0 servlets, this model of servlet managers is the most useful.

Domino 5 servlet manager
If the Domino servlet manager is enabled in the server document of a Domino server, it will be started and stopped by the nhttp.exe process. The Domino servlet manager implements an older version of the servlet API, version 2.0, and runs in an older version of the Java Runtime Environment, version 1.1.8. As a result, its compatibility with modern Java applications is limited.

Classes loaded by the Domino 5 servlet manager, including servlet classes, must be located in the classpath of the Domino Java Virtual Machine (JVM). This classpath is specified in two places for the servlet manager: first, in the JavaUserClasses entry in the Notes.ini file and second, in the Server document under the Java Servlets-Class path heading. All classpaths are not created equal, however, because a class loaded from the Server document classpath has limited functionality. It cannot make Java Native Interface (JNI) calls or perform any function that the JVM considers a security risk. Annoyingly, the Notes.ini classpath entry, although not limited by the same security constraints as the Server document entry, cannot be longer than 255 characters. Because Sametime uses 195 of these characters, additional products that need to add classpath entries can only use the remaining 65 characters. One way around this limitation is to move some of the classpath entries from Notes.ini into the Server document, but beware, you can safely move only certain entries because Sametime uses JNI and other features that are considered security risks to the JVM. The Sametime classpath entries that are required to be in the Notes.ini JavaUserClasses entry are as follows:

- stcore.jar
- log4j-118compat.jar
- xml-apis.jar
- xalan.jar
- xercesImpl.jar
- stmtgmanagement.jar

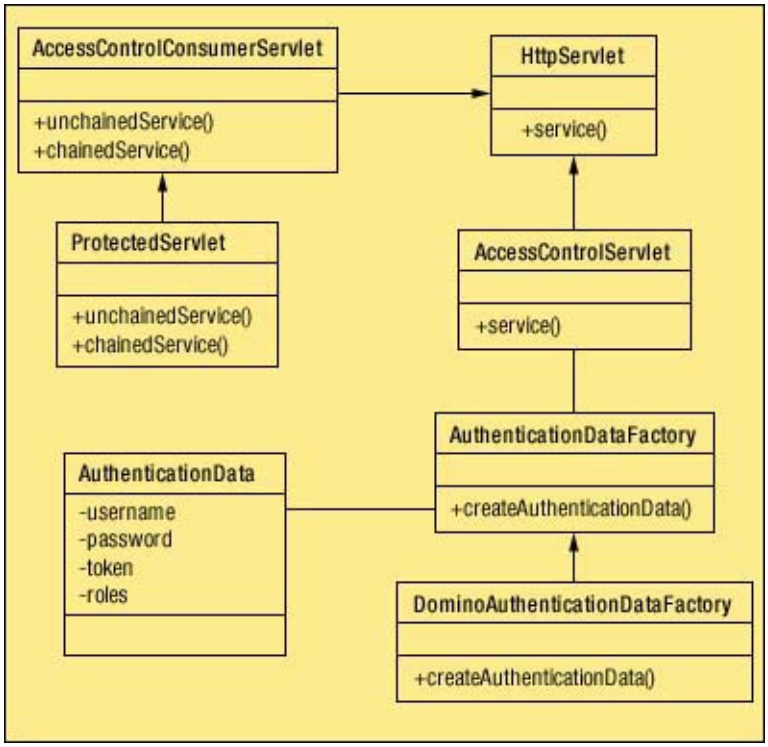
The following table lists all Sametime Servlet classpath entries and provides a brief description of their functionality:

Name	Description
Domino Java directory	Placeholder for future Java class files that must override version 3.0 class files
dsig.zip	Provides Base64 encoding and decoding from the W3C PICS project version 1.2
xml-apis.jar xercesImpl.jar xalan.jar	JAXP 1.0 API for XML/XSLT parsing and processing; uses the Apache project's Xalan implementation, version 2.3.1
stcore.jar	Core server-side Java classes for the Sametime server
stmtgmanagement.jar	Meeting Management API used internally by Sametime
STNotesCalendar.jar	Provides Sametime Meeting calendaring and scheduling features
log4j-118compat.jar	Version 1.1.3 of the Apache log4j product for diagnostic prints; this archive is Java 1.1.8 compatible because it has had the RollingFileAppenderBeanInfo class removed
ibmjsse.jar	IBM version 1.0.2 of the Java Secure Socket Extension (JSSE) API that provides SSL support
activation.jar	Part of the JavaBeans Activation Framework needed by the JavaMail API and used by the FileUpload Servlet
mail.jar	Part of the Java Mail API used by the FileUpload Servlet for MIME content parsing
Domino Data directory	Enables Sametime resource files to be loaded through relative file paths

Servlet authentication and access control framework

Sametime servlets perform many sensitive tasks that need to be protected from unauthorized use. However, the

Domino 5 server does not provide a standard way for servlet access to be controlled by access control lists (ACLs) similar to the way it provides ACLs for Web access of Domino databases. Moreover, the only way to enable user authentication of servlet access is to turn off anonymous access for the entire Domino Web server. In other words, the Domino server would force all HTTP users to authenticate, regardless of the resource they were requesting. You cannot limit servlet access to a specific list of users or allow anonymous access to a subset of servlets. Sametime 3.0 requires user-level access control of servlets, a feature not provided by Domino, so it uses a custom authentication and access control framework. The following diagram shows the custom Sametime servlet ACL framework:



All Sametime 3.0 servlets that require authentication and access control extend the AccessControlConsumerServlet class instead of the HttpServlet class directly. This class allows the servlet to receive authentication data about a user from a chained servlet called the AccessControlServlet. Users of Sametime servlets enter a URL that includes the AccessControlServlet first, and then the name of the desired servlet. For example, to access the Sametime Administration servlet, a user enters the following URL: http://hostname/servlet/auth/admin.

The word *auth* is the AccessControlServlet alias, and the word *admin* is the Sametime Administration servlet alias. Because auth is specified first in the URL, the HTTP request is first sent to the AccessControlServlet where user authentication and access roles are determined. Then, the auth servlet passes the user's credentials to the next servlet in the chain, the admin servlet. The Administration servlet receives the HTTP request data along with the user's credentials and determines whether or not the user is authorized to access it.

The first chained servlet: the AccessControlServlet

The AccessControlServlet reads HTTP request header data to determine if a user has been authenticated or not. If the user has been authenticated, then his name is retrieved through the getRemoteUser() call on the HttpServletRequest object. This user name is usually a fully qualified canonical name and has the most specificity that can be determined by the Domino server. For example, if the user logged in as J Smith, the getRemoteUser() method might actually return CN=Jimmy Smith/OU=Admins/O=Lotus depending on how the organization structure for the Domino domain is set up. Also, LDAP users are expanded to their full canonical name so that you can use Domino directory assistance to specify an LDAP directory for Sametime.

Next, after the name is read from the HTTP request data, the servlet tries to find a matching name in the ACL of a Domino database. If the user name has not been specified, as in the case of an anonymous user, the servlet tries to match the word anonymous in the database ACL. A setting in the servlets.properties file of the Domino server,

namely, an initialization argument for the AccessControlServlet called DominoDatabaseName, determines the actual database used for ACL matching. Sametime 3.0 ships with this argument set to stconfig.nsf; therefore, all servlet users must be listed in the ACL of that database. Some limitations of this name matching technique are that the ACL is not searched for wildcards, and nested groups are only searched to one level. Consequently, ACL entries such as */Domain and users that exist in groups within groups will not correctly match an authenticated user.

Finally, if the login name matches a name in the database ACL, the AccessControlServlet reads what roles are enabled for that user. Whether or not any enabled roles are found, an AuthenticationData object is created with the user's name passed to the next servlet in the chain. This next servlet is an instance of an AccessControlConsumerServlet. Most Sametime 3.0 servlets extend this class, protecting them from unauthorized access.

The second chained servlet: the AccessControlConsumerServlet
The AccessControlConsumerServlet, or Consumer servlet, first checks to see if the HTTP request was sent through the AccessControlServlet; in other words, it checks to see if it was chained. If not, then the Consumer servlet may reject the request right away. This rejection is controlled by an initialization argument for the Consumer servlet in the servlets.properties file called UnchainedAccessEnabled. If this setting is set to false, and the HTTP request was not chained through the AccessControlServlet, then the Consumer servlet responds to the request with the HTTP error condition SC_FORBIDDEN. If this setting is true, then it does not matter whether or not the request was chained through the AccessControlServlet.

Next, the Consumer servlet checks to see if the roles that are enabled for the user are adequate to execute the servlet. The roles that are checked against the user's roles are specified by an initialization argument for the Consumer servlet called AccessControl.Roles. This setting can vary for each servlet depending on the servlet's initialization arguments; however, Sametime 3.0 ships with most of its servlets protected by the role SametimeAdmin. This means that any user who tries to access a Sametime 3.0 servlet must have the SametimeAdmin role defined for him in the ACL of stconfig.nsf. After a user has been determined to have the proper roles enabled, the Consumer servlet processes the HTTP request.

What are the Sametime 3.0 servlets?

Sametime 3.0 relies on nine different servlets to operate correctly. The following table summarizes each of the servlets, but more in-depth descriptions follow for each servlet, except the Access Control servlet discussed in the previous section.

Servlet name	**Access Control**
Alias	auth
Servlet Classname	com.lotus.sametime.admin.authentication.AccessControlServlet
Description	Determines credentials for authenticated users
Servlet name	**Administration**
Alias	admin
Servlet Classname	com.lotus.sametime.admin.AdminXPathRequestServletJAXP
Description	Generates Administration client HTML
Servlet name	**Bootstrap**
Alias	bootstrap
Servlet Classname	com.lotus.sametime.configuration.DominoBootstrapServlet
Description	Sets up environment before Sametime server starts
Servlet name	**File Upload**
Alias	fileupload
Servlet Classname	com.lotus.sametime.materials.servlets.FileUploadServlet
Description	Attaches uploaded materials to Sametime meetings
Servlet name	**Meeting Management API**
Alias	mmapi
Servlet Classname	com.lotus.sametime.meetingmanagement.remote.servlet.MMAPIServlet
Description	Creates, edits, and retrieves Sametime meeting data
Servlet name	**Record and playback**
Alias	rapfile
Servlet Classname	com.lotus.sametime.materials.servlets.RAPFileServlet
Description	Deletes record-and-playback files

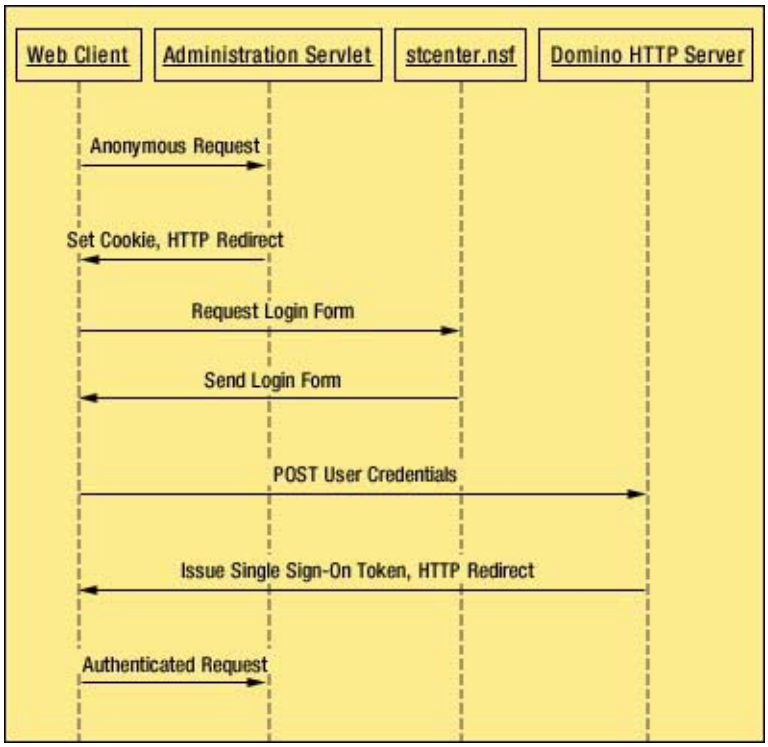
Servlet name	**Refresh**
Alias	refresh
Servlet Classname	com.lotus.sametime.materials.servlets.RefreshServlet
Description	Triggers a reload of meeting materials for a particular meeting
Servlet name	**Configuration**
Alias	scs
Servlet Classname	com.lotus.sametime.configuration.DominoConfigurationServlet
Description	Reads and writes server configuration data
Servlet name	**Notes Calendar**
Alias	stcal
Servlet Classname	com.lotus.sametime.calendar.NotesCalendarServlet
Description	Retrieves the correct meeting from a scheduled repeated meeting set

Administration servlet
The purpose of the Administration servlet is to generate HTML pages for the Sametime Administration client. This servlet does not simply serve static HTML pages to Web users; it dynamically generates the HTML pages using XML and XSL stylesheets. Therefore, the Administration servlet responds to each HTTP request with an XML/XSLT transformation based on the requested URL. The stylesheets for the Administration client and its associated bitmaps are stored in the directory, C:\Lotus\Domino\Data\domino\html\sametime\stadmin in which C:\Lotus\Domino\Data is the Domino Data directory of the Sametime server. The XML for the Administration client is dynamically generated inside the Administration servlet based on several data sources.

The first XML source is the HTTP headers from the administration request. The Administration servlet uses these headers to determine which language to return to the client based on the client's locale. Secondly, the servlet generates XML based on query parameters passed in the HTTP request, such as Sametime server statistics and Sametime process names. And finally, the XML for the Administration client includes the output of the Configuration servlet, an XML fragment that includes all configuration data for the Sametime server. After all of the XML data has been collected and the XSL stylesheet has been chosen, the Administration servlet transforms these two pieces of data into the HTML of the Administration client.

To access the Administration client through the servlet, a Web user must have the proper access rights. In Sametime 3.0, Domino single sign-on (SSO) session cookies are required for Web access as opposed to previous versions of Sametime that relied on basic authentication. As a result, the Administration servlet redirects all unauthenticated users to the Sametime custom logon form. Before redirection, the servlet sets a cookie on the HTTP request called STAdminRedirect=true so that the request can be correctly routed back after authentication has occurred. The Sametime custom logon form is found in the stcenter.nsf database in a form called STLogonForm. After a user fills out the logon form and submits it, Domino generates an SSO token for that Web session and sets it as a cookie. The user is then routed back to the Administration servlet where the authentication token is read. In Sametime 3.0, if an authenticated user has the SametimeAdmin, SametimeMonitor, or DatabaseAdmin roles enabled in stconfig.nsf, that user can access the Administration client through the Administration servlet.

The following diagram shows how Sametime authentication works:



Bootstrap servlet
Before a Sametime 3.0 server can start, the Bootstrap Servlet must prepare its runtime environment. This servlet must be set to run as the first servlet in the servlets.properties file so that it can complete its work before any other Sametime processes start. It acts as a consistency check of the Sametime.ini and Notes.ini files. First, it checks the SametimeBootstrap setting in the Notes.ini file to see if it exists. This setting should point to the full filepath of the Sametime.ini file. If the setting does not exist, then a default setting is written to the Notes.ini file as follows:

SametimeBootstrap=<JVM "user.dir" value>/sametime.ini

The JVM user.dir value in a Sametime environment is the Domino program directory in which the Domino 5 server was installed. Several Sametime 3.0 processes read the SametimeBootstrap setting from Notes.ini to determine where they should get their additional environment context parameters.

The second task that the Bootstrap servlet performs is to setup parameters in the Sametime.ini file. All of these parameters appear under the [Config] section in the file and are described in the following table:

Parameter Name	Parameter Value
SametimeCluster	Fully qualified Notes name of the Sametime server
SametimeDirectory	Same as the "Directory" parameter in Notes.ini; usually the Domino Data directory
ConfigurationPort	Port number of the Domino Web server
ConfigurationHost	Fully qualified domain name of the Sametime server read from NetAddresses field in the Domino Server document
SametimeEventServerPort	Port value read from the MeetingServices document in stconfig.nsf
ConfigurationChangeListener.count ConfigurationChangeListener.classname ConfigurationChangeNotifier.count ConfigurationChangeNotifier.classname	Dynamically created Java objects that notify Sametime processes when Sametime configuration parameters change

Lastly, the Bootstrap servlet sets a parameter and value in the Notes.ini file SametimeBootstrapInitialized=1, which signals to other Sametime processes that the runtime environment has been set up. After the other Sametime processes have successfully read this parameter and started, it is reset to 0. If the Sametime Bootstrap servlet fails to start up and to complete all of its tasks, the Bootstrap servlet does not set the the Notes.ini parameter to 1, and the rest of Sametime does not start.

File Upload and Refresh servlets
Both of these servlets are part of the Sametime materials system, which is responsible for receiving, attaching, and transferring meeting attachments. The File Upload servlet is necessary for the Sametime server to take advantage of Microsoft Exchange integration. It provides a remote proxy for the integrated Exchange client to use for uploading materials to a Sametime server. Files are sent to the File Upload servlet as multipart MIME content, parsed by the servlet, and then attached to a Sametime meeting document in the Meeting Center database. The MIME parsing relies on the IBM implementations of both the JavaMail API version 1.2 and JavaBeans Activation Framework API version 1.01. The File Upload servlet is protected from any user who does not have the SametimeAdmin role defined in the ACL of stconfig.nsf.

The Refresh servlet is called if the materials for a particular Sametime meeting are changed after they have been uploaded to the Meeting server. A typical change might be to add or remove meeting attachments. This servlet notifies the rest of the Sametime servers of the change, so that the files can be reloaded as necessary.

Meeting Management API servlet
The Sametime Meeting Management API servlet is an HTTP proxy for the Sametime Meeting Management API. This API is used internally throughout the Sametime server and is not for public use. It allows a client to create, delete, and edit Sametime meeting data. This servlet is protected from access by any user who does not have the SametimeAdmin role defined in the ACL of stconfig.nsf.

Record-and-playback servlet
This utility servlet handles cleanup of record-and-playback (RAP) files that are generated on a Sametime server when a Sametime meeting is recorded. It is not used by the Sametime 3.0 server directly, but is necessary for compatibility with the planned version of Sametime that distributes Meeting servers over several machines. Any user that does not have the SametimeAdmin role defined in the ACL of stconfig.nsf cannot access it.

Configuration servlet
The Sametime 3.0 server processes rely on the Configuration servlet for retrieval of server configuration data and notification of changed data. This servlet permits access to a wide range of values, such as network ports, network addresses, Sametime log settings, Sametime meeting usage limits, and many others. By default, the servlet does not require authentication, but it should be locked down as soon as you deploy a Sametime server.

To secure the Configuration servlet, add a servlet initialization argument to the servlets.properties file in the Domino Data directory. The argument AccessControl.Roles=[SametimeAdmin] tells the servlet that only users with the role SametimeAdmin enabled on the stconfig.nsf database ACL can execute it. This setting is part of the Sametime custom authentication and access control framework, and it can specify any role or semicolon-delimited list of roles necessary to run the servlet. Next, create a special user account in the Domino program directory for servlet access. This account should not be tied to a particular person, and it needs to have an Internet password defined. After you create the account, add the user name of the account to the ACL of the stconfig.nsf database. You need to enable whichever roles were specified in the servlets.properties initialization argument for that user. Finally, you need to edit the Sametime.ini file in the Domino program directory to add the user name and Internet password for the special servlet user account. The following excerpt from the Sametime.ini file shows the format of the user credentials:

```
[Config]
…
SametimeAdminUsername=SpecialServletAccountUsername
SametimeAdminPassword=AccountPassword
…
```

Refer to the **Sametime 3.0 Release Notes** to learn how to protect this servlet from unauthorized access.

The Sametime 3.0 server processes read this file to get their credentials for accessing the Configuration servlet. After the user credentials are written to the Sametime.ini file and you start the Sametime 3.0 server, the user name and password will be Base64 encoded by a server process to help protect it from casual observers.

However, this sensitive information is not encrypted, so it should be guarded diligently.

The Configuration servlet retrieves Sametime configuration data from the data sources summarized in the following table:

Data Source Name	Location
Stconfig.nsf	Domino Data directory
Stlog.nsf	Domino Data directory
Local Address Book (names.nsf)	Domino Data directory
Directory Assistance (da.nsf)	Domino Data directory
Sametime.ini	Domino program directory
Notes.ini	Domino program directory
Stappletver.txt	html/sametime directory
Buildinfo.txt	html/sametime directory
Stconnectver.txt	html/sametime/javaconnect directory

Confusingly, you do not use the Sametime server tab in the Domino Directory for Sametime configuration data. This tab held values for the Sametime 1.0 and 1.5 servers, but it has been deprecated since those versions because of the scheduling difficulty of keeping the Domino Directory template synchronized with the configuration needs of the Sametime server. The data that is currently retrieved from the Domino Directory for Sametime is not found in the Sametime tab. It includes items such as the local HTTP server settings, Sametime connection documents, and remote Sametime server names.

The ideal place to view and change any Sametime server configuration parameter is the Sametime Administration client, but you can manually edit all of the data sources with an appropriate editor. If this is done, however, the Sametime server will not pick up the manual changes until you stop and start the Configuration servlet through the Domino Web server. Additionally, some configuration changes require you to restart the entire Sametime server, depending on the parameter and how it is used.

Notes Calendar servlet
This servlet matches Notes online meeting identifiers with Sametime meeting identifiers. You can make appointments for Sametime meetings through the Notes Calendaring and Scheduling feature of a Notes client, where you're given an online meeting identifier. As the meeting appointment is accessed at the appropriate time, the Notes Calendar servlet matches the online meeting identifier with the appropriate Sametime meeting identifier, which is unique to a particular active Sametime meeting. If an online meeting is scheduled to repeat several times, then the servlet matches the Sametime meeting that is currently active or the next scheduled meeting if one is not active for the repeated set.

Troubleshooting Sametime servlets

In order for a Sametime 3.0 server to start, its Sametime 3.0 servlets must successfully initialize. Specifically, the Bootstrap and Configuration servlets must be started before any other Sametime processes will start. If an error occurs in any of the servlets, the Domino console usually shows Java exception messages when the HTTP server starts, but these are difficult to understand. A large portion of the Sametime server relies on the Apache Log4j package to collect runtime diagnostics, which makes it easier to troubleshoot. The SametimeDiagnostics.properties file in the Domino program directory controls these diagnostics prints. You can edit it to increase the verbosity of prints by changing all occurrences of the word INFO to DEBUG and restarting the Sametime server. The diagnostics output files, SametimeAdminDiagnosticsOutput.txt and SametimeDiagnosticsOutput.txt, are in the Domino program directory and contain debug information for the servlets.

If a Sametime server will not start up and the reason is unclear, one way to troubleshoot it is to disable all of Sametime except for the servlets. The easiest way to accomplish this is to remove the staddin task from the ServerTasks entry in the Notes.ini file and restart the Domino server. The Domino Web server will start, and hence the servlets, but the rest of Sametime will not. This should isolate the servlets and either implicate or clear them as the source of the problem. Remember to add the staddin task back to Notes.ini when you are finished troubleshooting.

An important pillar

Servlets are key to the Sametime 3.0 runtime environment. They are the first part of the Sametime server to start up, and they provide such essential services as meeting creation, administration, configuration, and materials management. Understanding the Sametime 3.0 servlets, how they are accessed, how they are protected, and how they can be controlled, will help you keep your servers running smoothly.

ABOUT THE AUTHOR

Isaac Hands works for IBM on the Sametime Development team in Lexington, Kentucky. His seven month-old son, Will, enjoys bouncing in a large plastic saucer, rolling across the floor to chew on the fringes of an area rug, and repeating the phrase, "DaDaDaDaDa." Isaac hopes to teach Will some new tricks by the time he writes another article for LDD. In fact, Isaac plans to teach Will how to write his next article for LDD, so stay tuned.

