# Content Server

Version: 6.3

# Installing Satellite Server 6.3

Document Revision Date: Jun. 15, 2011

FatWire®
SOFTWARE

*Installing Satellite Server 6.3*
Document Revision Date: Jun. 15, 2011
Product Version: 6.3

**FatWire Technical Support**
www.fatwire.com/Support

**FatWire Headquarters**
FatWire Corporation
330 Old Country Road
Suite 303
Mineola, NY 11501
www.fatwire.com

Table of

# Contents

Chapter 1

# Satellite Server Configurations

Satellite Server is a product that works with your Content Server content management system to provide three things:

- An additional layer of caching, supplementing the layer of caching that is provided by the Content Server cache.
- The ability to quickly and economically scale your Content Server system by adding remote installations of Satellite Server.
- The ability to improve your website's performance and reduce the load on Content Server by moving content closer to the web site visitors who will view it.

This chapter introduces you to the configurations that you implement in order to receive these benefits.

You can configure Satellite Server in three ways:

- Co-Resident, which provides a second layer of caching.
- Co-Resident and Remote Combined, which improves performance and scalability.
- Remote Only (Delivery System), which you implement when you have enough remote Satellite Servers to make the co-resident Satellite Server unnecessary.

The following sections describe these configurations and what they are used for in greater detail.

# Co-Resident

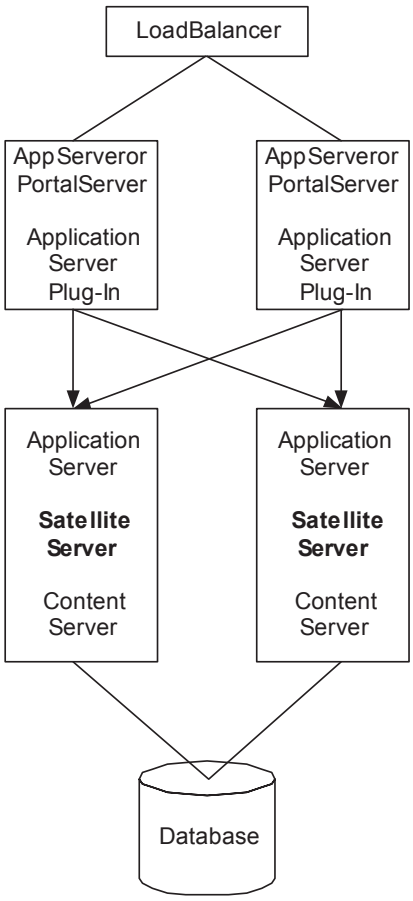
Content Server ships with a copy of Satellite Server that is installed on the same machine as your Content Server software. This is your **co-resident** Satellite Server. The co-resident Satellite Server provides a layer of caching in addition to that provided by Content Server's cache.

You will have a co-resident Satellite Server on all of your systems: development, management, delivery, and testing.

Satellite Server and the Content Server cache work in tandem to provide **double-buffered caching**, where copies of cached pages are stored in both the Satellite Server and the Content Server caches. For more information about double-buffered caching, see the caching chapter of the *Content Server Developer's Guide*.

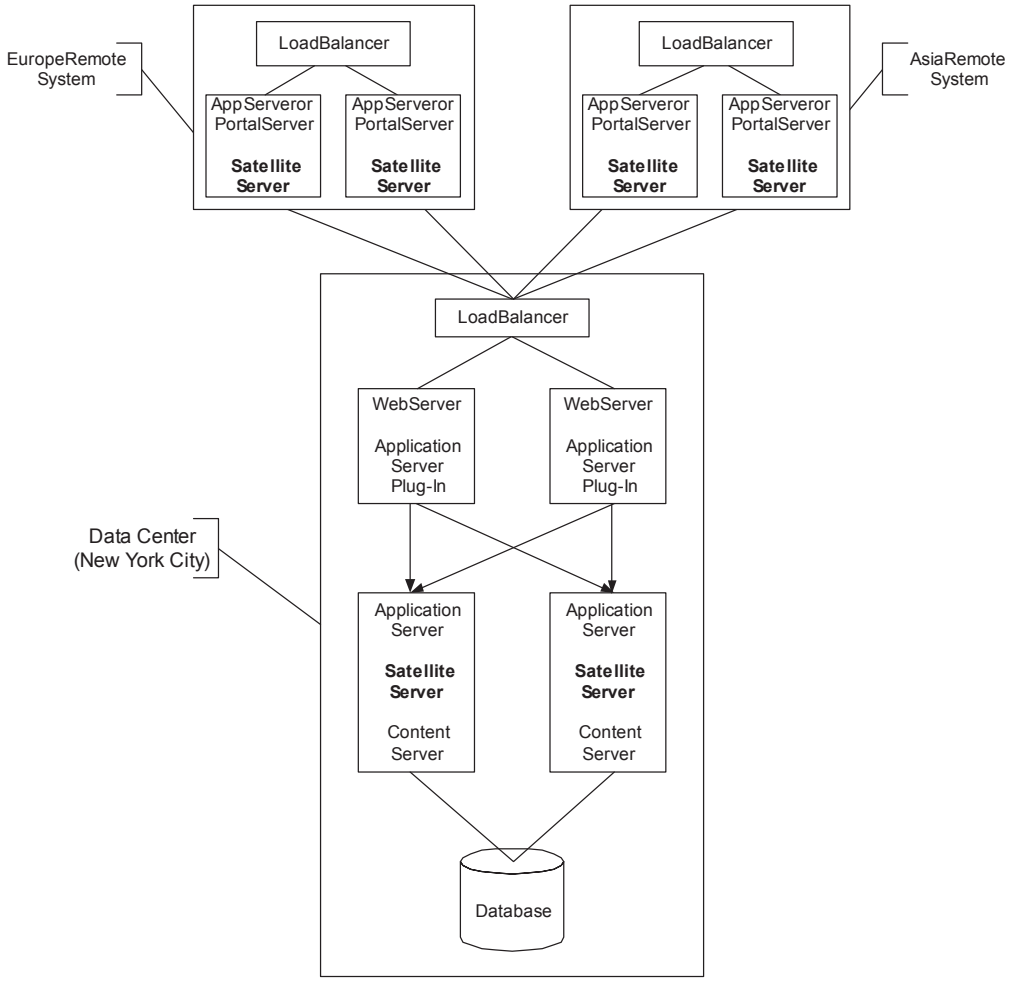
The co-resident Satellite Server installs automatically when you install Content Server. Co-resident Satellite Server will run automatically once Content Server is installed on your system. However, FatWire recommends that you tune your co-resident Satellite Server host to optimize your Content Server system's performance, as described in Chapter 3, "Tuning Satellite Server."

The following diagram illustrates a co-resident installation of Satellite Server:



# Co-Resident and Remote Combined

In addition to the co-resident Satellite Server, you can also install remote instances of Satellite Server as part of your delivery system. The remote Satellite Server systems are on hardware that is close to the web site's audience, as shown in the following diagram:




Remote installations of Satellite Server provide several benefits in addition to allowing double-buffered caching:

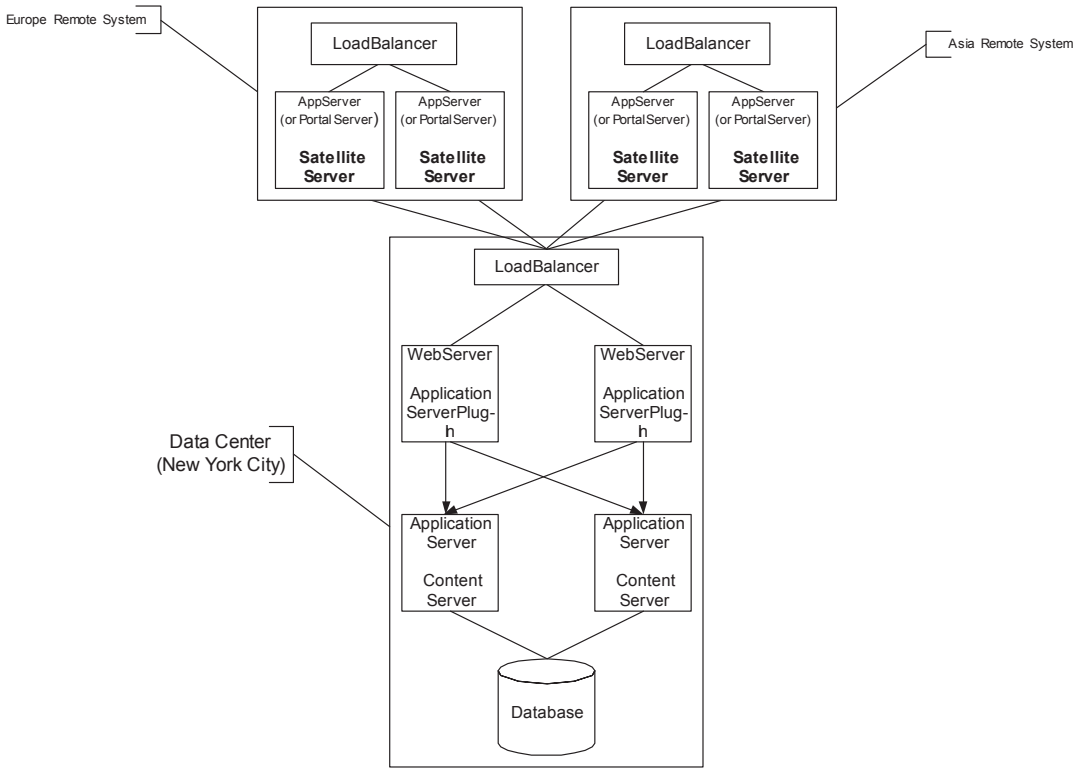
- They improve the performance of the web site by moving the content closer to its audience. In the preceding diagram, for instance, the main data center is located in New York City, while the secondary data centers are located in Europe and Asia.
- They remove load from Content Server. Because remote Satellite Servers do not require the same sort of hardware that a full installation of Content Server does, adding them to your Content Server system is a simple and economical way to make Content Server scalable.

For information on how to install and configure remote instances of Satellite Server, see Chapter 2, "Installing Remote Satellite Servers," and Chapter 3, "Tuning Satellite Server" in this book.

# Remote Only (Delivery System)

At some point, you may add enough remote Satellite Servers to your system that the co-resident installation of Satellite Server on your delivery system becomes unnecessary. Performance testing allows you to determine when you have reached this point; you will see less and less CPU utilization on your co-resident machine, and more CPU utilization on the remote Satellite Server machines.

When this occurs, you can disable your co-resident Satellite Server, creating the following configuration:




For instructions on how to disable your co-resident Satellite Server on the delivery system, and how to tune for this configuration, see Chapter 3, “Tuning Satellite Server.”

Chapter 2

# Installing Remote Satellite Servers

When you install remote instances of Satellite Server, you can install them on any FatWire-supported application server or portal server. If you plan to install a web server, you will need to configure it by referring to the application server documentation.

Note that installing and configuring remote instances of Satellite Server is an iterative process. You must initially install, configure, and test one remote Satellite Server, then install, configure, and test your other remote Satellite Server installations.

After you have completed the installation and initial configuration of your Satellite Server software, tune each Satellite host to achieve optimum performance. For more information about tuning Satellite Server, see Chapter 3, "Tuning Satellite Server."

To install and configure remote instances of Satellite Server, you must complete the following steps:

- Step 1. Install Required Hardware and Software
- Step 2. Expand the Installation File
- Step 3. Run the Installer
- Step 4. Register Satellite Server with Content Server
- Step 5. Configure the Web Server
- Step 6. Start Satellite Server
- Step 7. Test the Configuration
- Step 8. Install Satellite Server on Additional Remote Machines

# Step 1. Install Required Hardware and Software

Before you install Satellite Server, be sure that you have the required hardware and software.

FatWire frequently revises the specific software and hardware configurations that are supported by Content Server and Satellite Server. For the latest information, go to:

```
http://e-docs.fatwire.com/CS
```

Locate the product version of interest, and click the **Supported Platform List (SPD)** link.

## Networking Requirements

The connection between the Satellite Server hosts and the Content Server host is the primary limiting factor for performance of serving uncached data. The following table describes the minimum networking requirements:

| | |
|---|---|
| Connection Between Satellite Server Hosts and Content Server Hosts | 100 Mbps |
| Connection Between Load Balancer and Satellite Server Hosts | 100 Mbps |

## Load Balancer Requirements

You must have a load balancer. FatWire does not require a particular brand of load balancer, but we do recommend that you use a load balancer that supports session affinity, and the session affinity features should be enabled.

## Configuration Requirements

Your Satellite Server hosts must meet or exceed the following requirements:

**Table 1:** Solaris Requirements

| | |
|---|---|
| Operating System | Solaris 2.6 |
| CPU | Sun Ultra 10 (running at 440 MHz) |
| Physical Memory | 512 Megabytes (1 Gigabyte recommended) |
| Disk Space | 200 Megabytes |

**Table 2:** Windows Requirements

| | |
|---|---|
| Operating System | Windows |
| CPU | Pentium III (running at 500 MHz) |
| Physical Memory | 512 Megabytes (1 Gigabyte recommended) |
| Disk Space | 200 Megabytes |

FatWire recommends using a homogeneous set of Satellite Server hosts. A heterogeneous set of Satellite Server hosts is acceptable, but having one complicates performance tuning. If you decide to use a heterogeneous set of Satellite Server hosts, FatWire strongly recommends configuring your load balancer to distribute the load based on the relative strength of each machine.

### Satellite Server Contents

1. Satellite Server needs a full-featured servlet container, a servlet engine, and a Java Runtime Environment. If you want to install a full-featured web server for your servlet container or application server, refer to the application server documentation.
2. Create a domain in which to install Satellite Server. Do one of the following:
   - For WebLogic and Sun JES configurations that are not portal based, create an application server domain. The application itself is automatically created by the WAR file. For instructions on creating an application server domain, refer to the Content Server installation guide corresponding to your configuration.
   - For portal-based WebLogic configurations, create a portal domain and a portal web application before running the Satellite Server installer. Use WebLogic to create the portal web application, before running the Satellite Server installer (during the installation process, WebLogic automatically deploys the WAR file; the portal web application must exist before deployment of the WAR file can take place). For instructions on creating a portal domain and portal web application, refer to the Content Server installation guide corresponding to your configuration
   - For portal-based WebSphere and Sun JES configurations, the WAR file is automatically deployed by WebSphere and JES. The WAR file creates an application with the same name as the WAR file. You will need to rename the WAR file, as shown in “Run the Installer,” on page 12.

## Step 2. Expand the Installation File

The installation file is named `satelliteserver.tar` for Solaris installations and `satserv.zip` for Windows installations. Extract this file to a host machine as indicated in the sections below.

### Windows

The installation file for Windows, `satserv.exe`, is a self-extracting ZIP archive. When you extract it, make sure that you retain the archived directory structure; otherwise the installer will fail.

### Solaris

You can untar the installation file into any target directory. However, for performance reasons, it is better to untar it into a directory on a local partition rather than an NFS-mounted directory on another host. Assuming the `satelliteserver.tar` file is in the current directory, the following command performs a verbose untar:

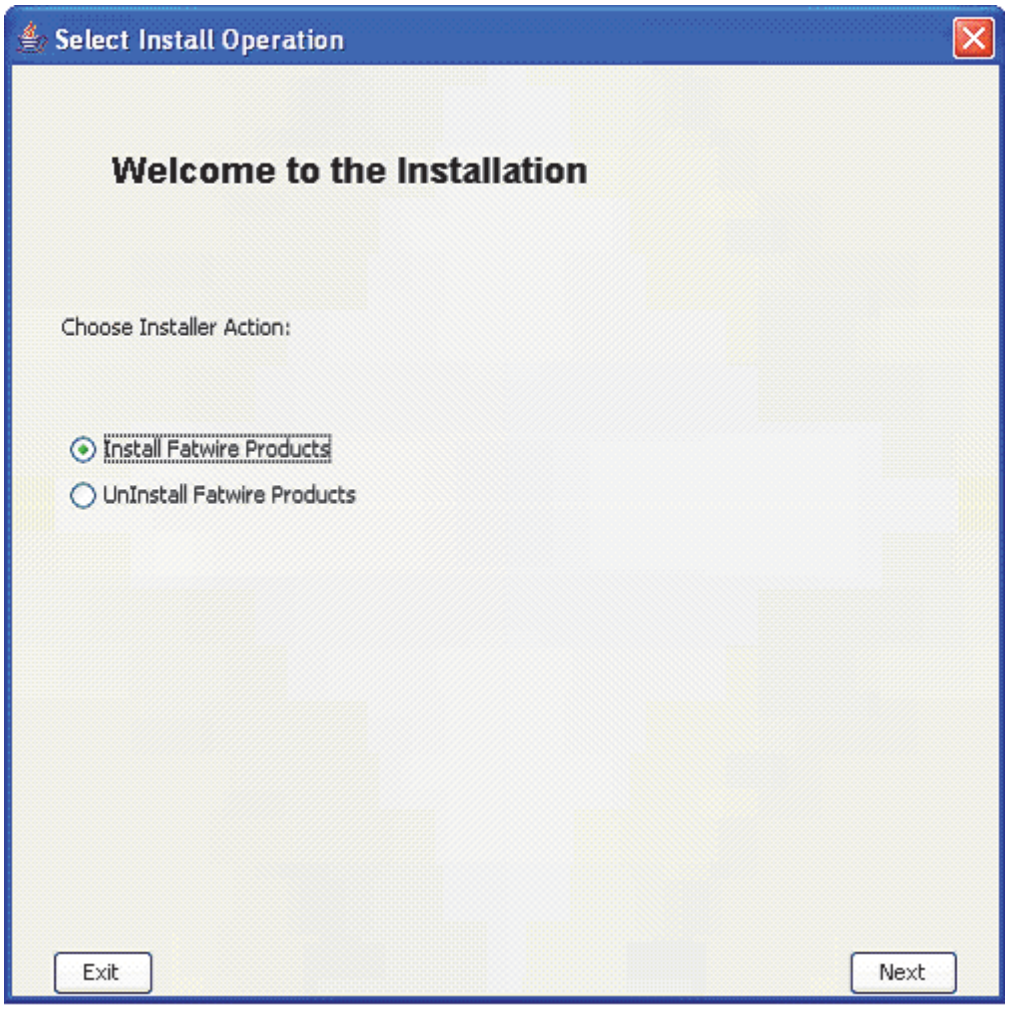
```
$ tar -xf satelliteserver.tar
```

Extracting `satelliteserver.tar` creates a subdirectory named `SatelliteServer`. Do *not* change the name of this directory or the names of any of its subdirectories.

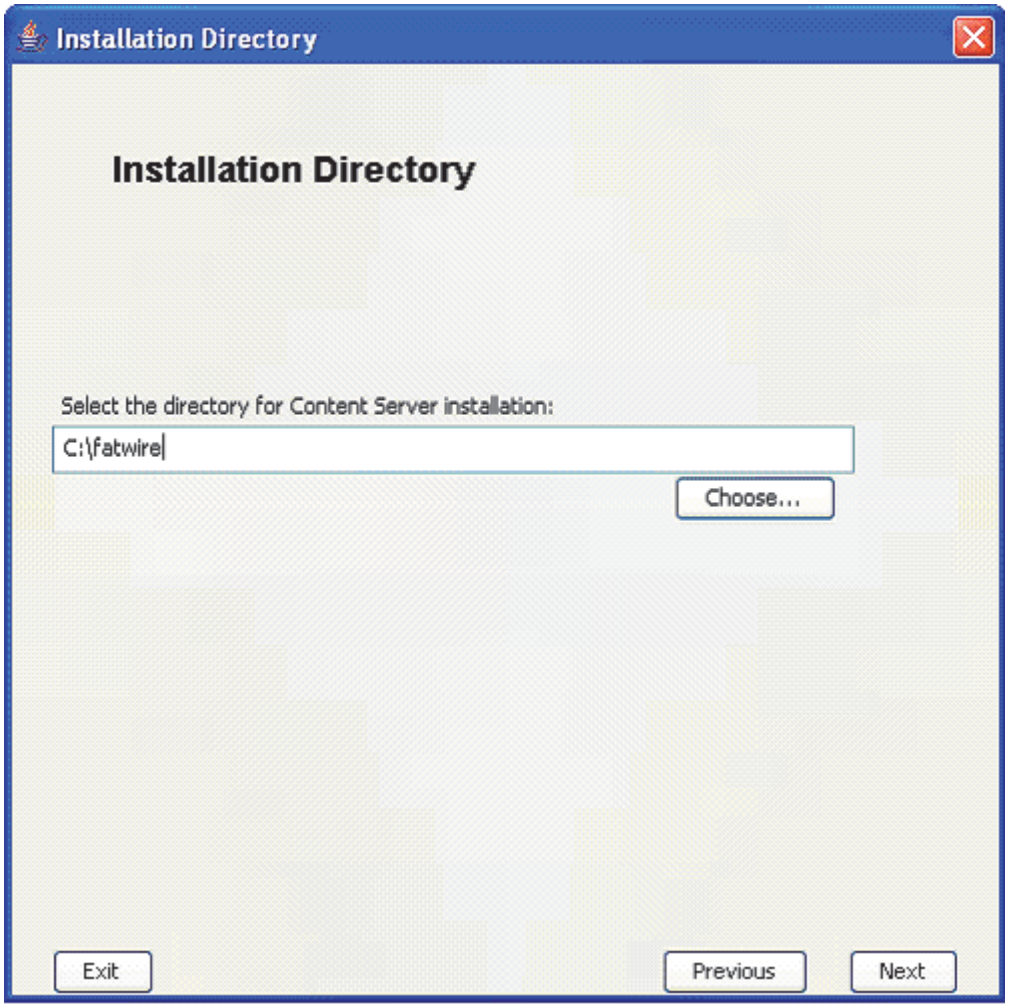
# Step 3. Run the Installer

**To install remote SatelliteServer**

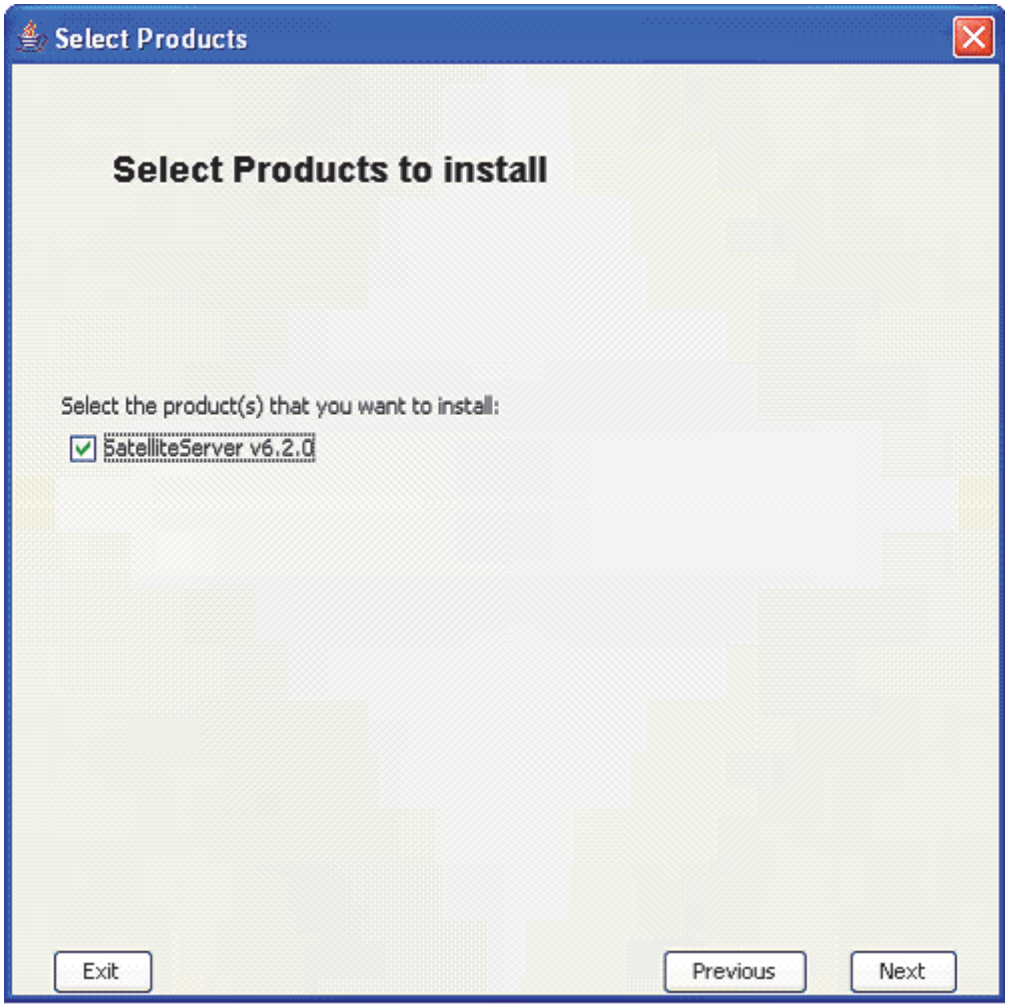
1. Run the installer script:
   - In Windows: `SSinstall.bat`
   - In *NIX: `ssinstall.sh`
2. Select **Install Fatwire Products** and click **Next.**



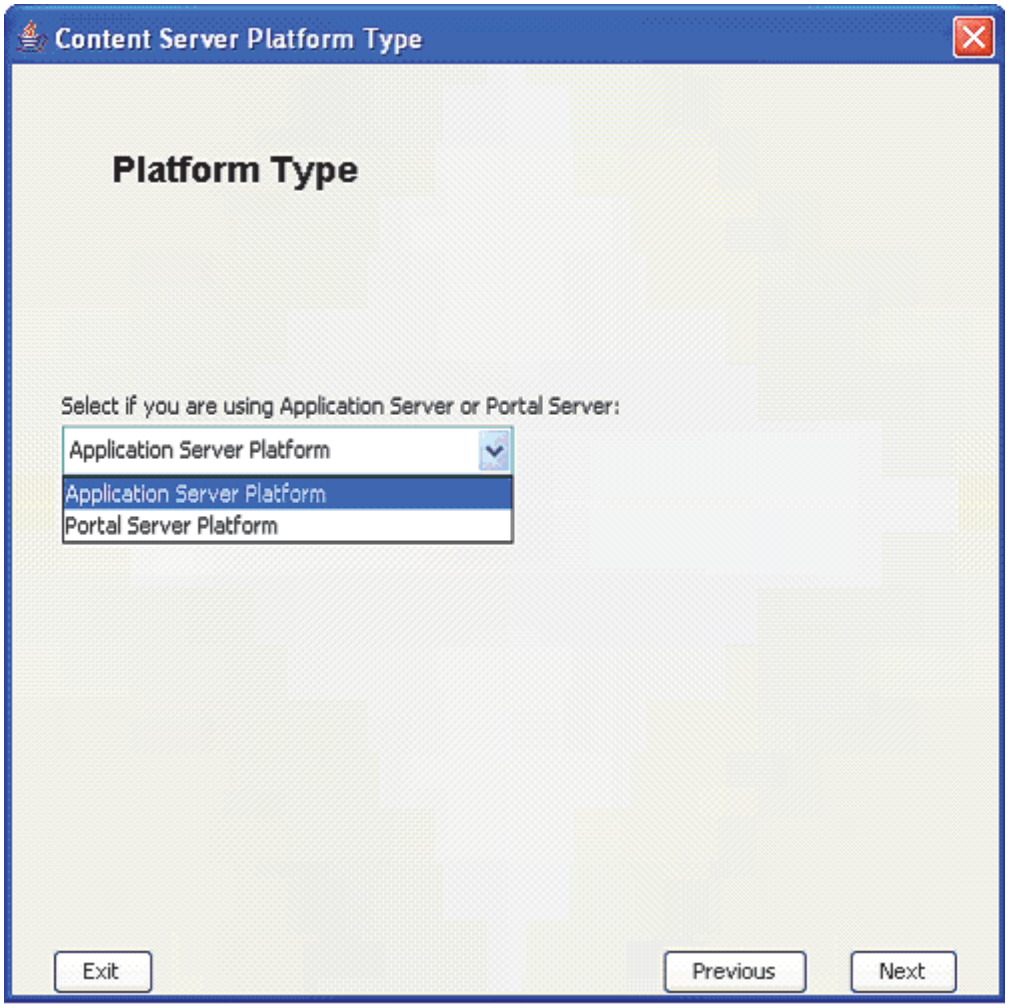
3. Enter the target installation path for Content Server. Make sure you have the required permissions.



4. Check the **SatelliteServer** box and click **Next**.



**5.** Select the desired platform type and click **Next**.

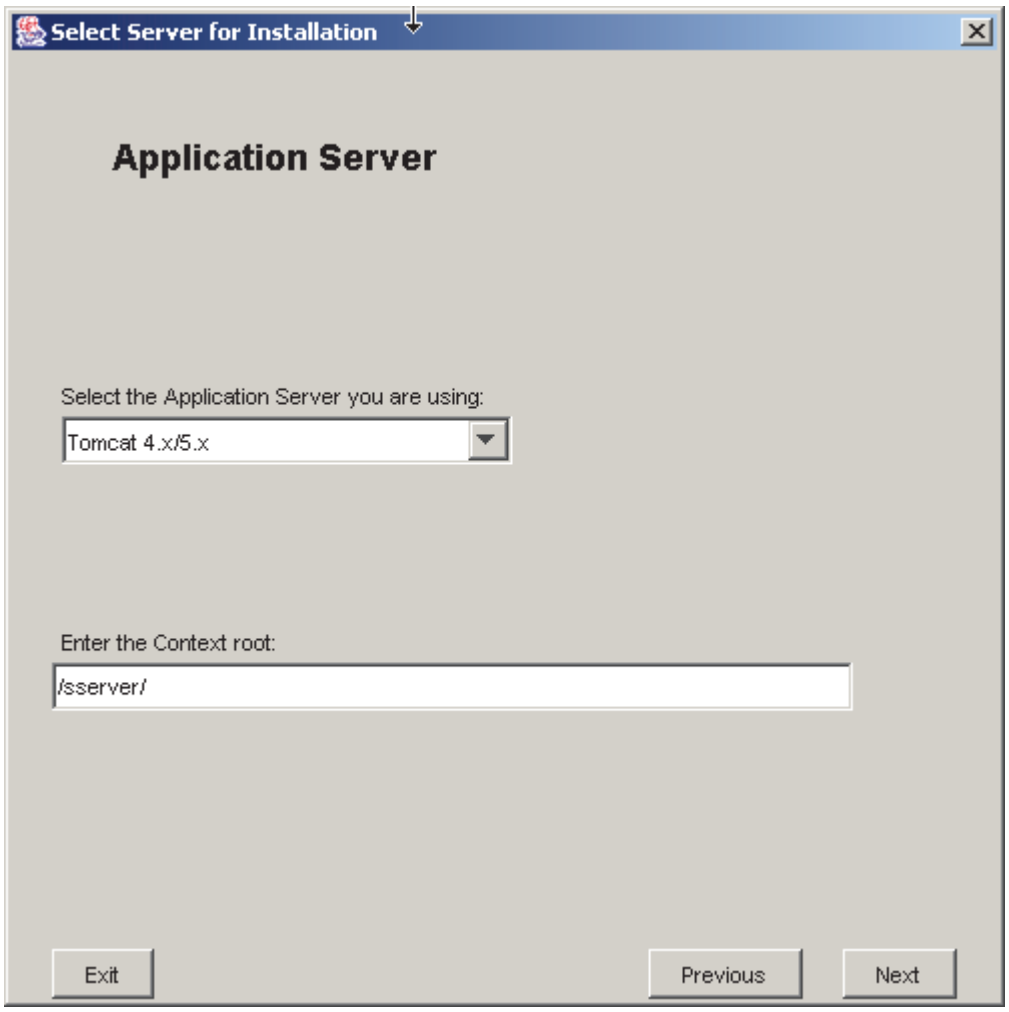


6. Select the desired application server, enter the context root, and click **Next**.

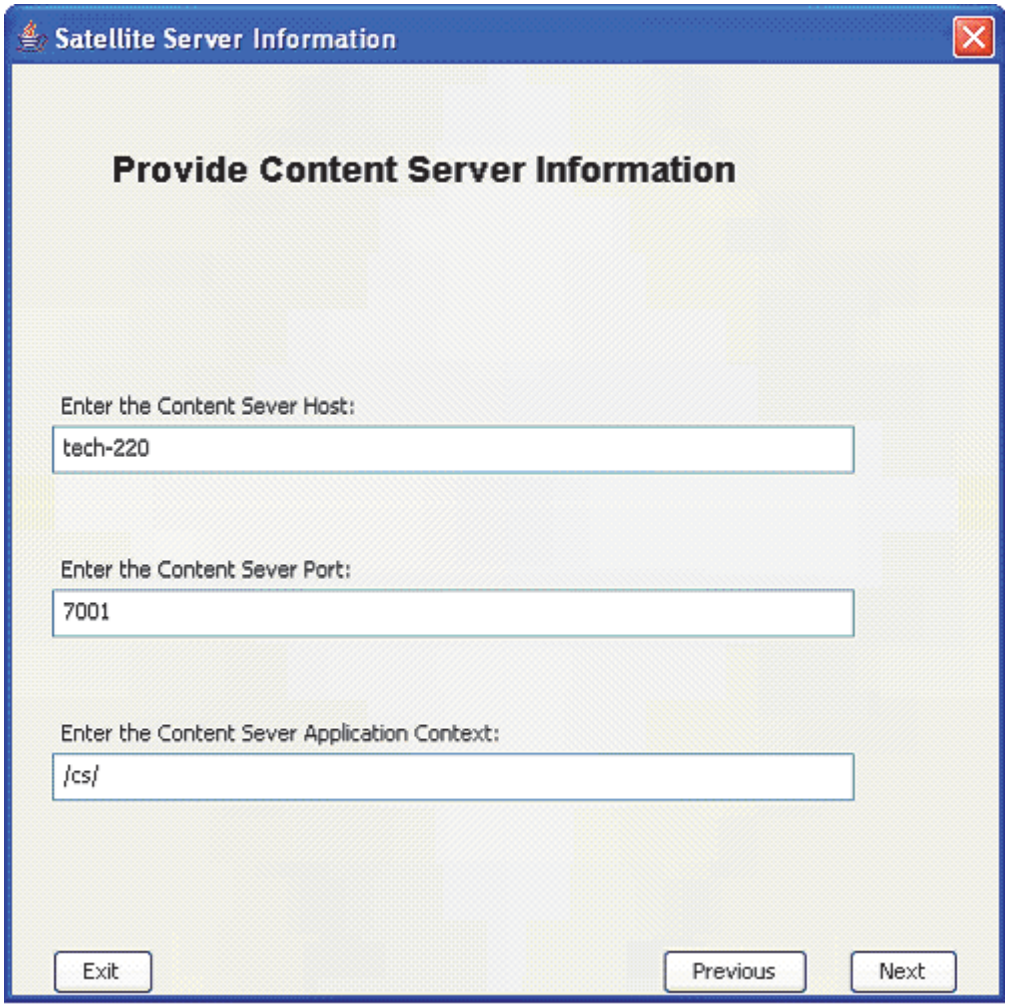
**Note**

For WebLogic portal installations, set the context root to the name of the portal web application that you have created using WebLogic Workshop. For all other configurations, set the context root to the name of the web application that you want to be created for Satellite Server.

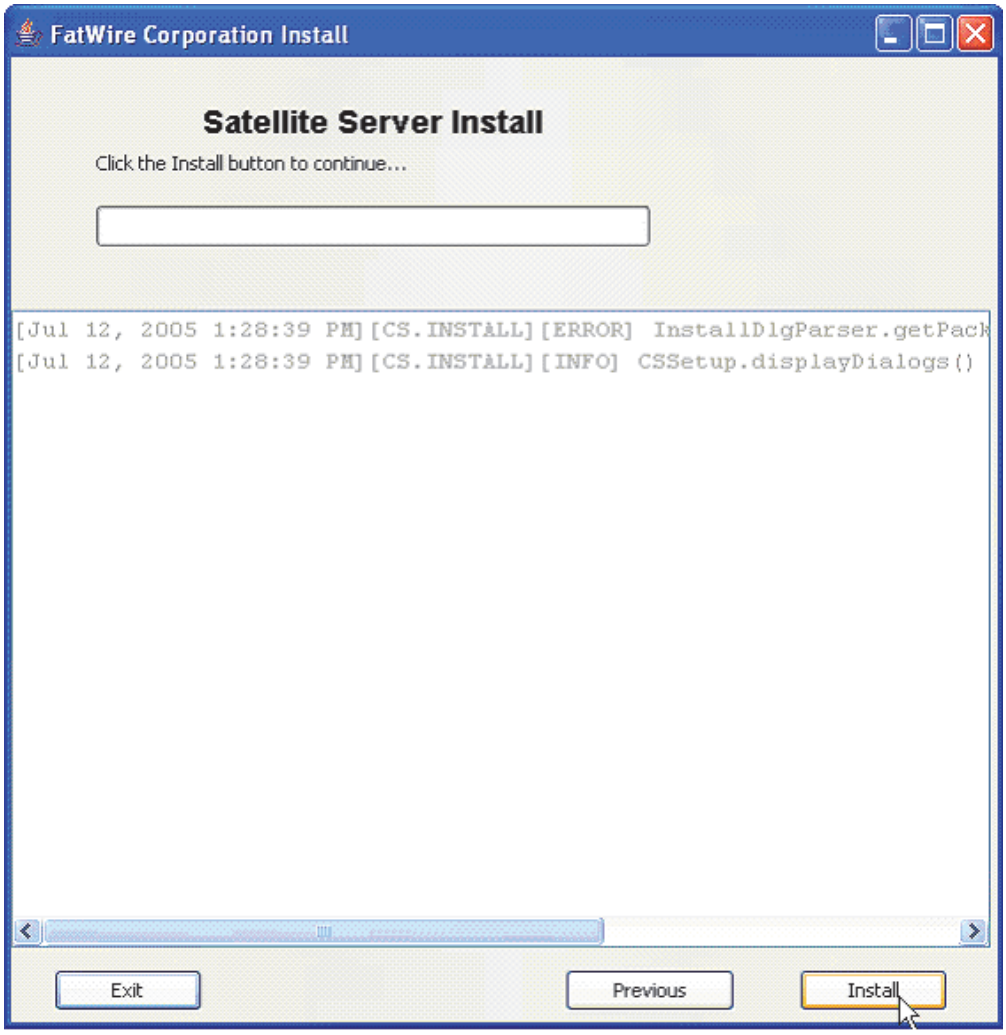
The CS installer will generate a WAR file named `cs.war`. For all portal installations other than WebLogic or WebSphere, you will rename the `cs.war` file (in step 9 on page 19) by giving it the name of the web application.



7. Enter the Content Server host name, port number, and application context URL SatelliteServer will be connecting to.



8. Click **Install** to start the installation process.

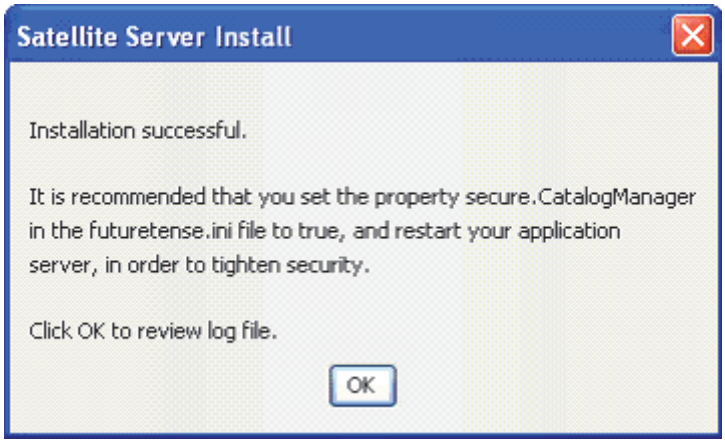


9. If necessary, rename the `cs.war` file (as instructed in step 6 on page 16). Deploy the war file to the server, then start (or restart) the server, and click **OK** to complete the installation.

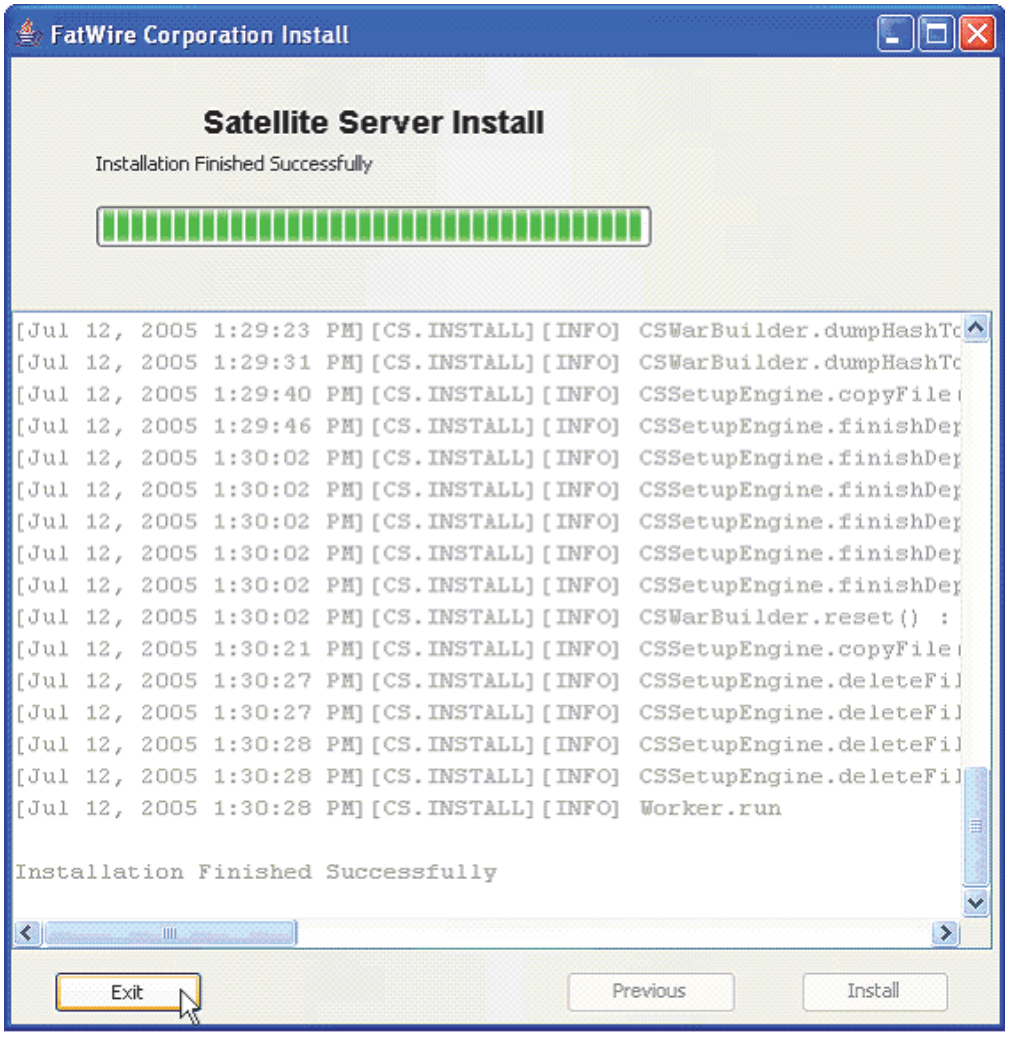
**Note**

If you are using WebLogic, the war file is automatically deployed. You need to either start or restart the server, and then click **OK** in the window shown below to complete the installation.

**10.** If server deployment was successful, you will be presented with a confirmation dialog box. Click **OK** to review the installation log.



11. Click **Exit** to quit the installer.




# Step 4. Register Satellite Server with Content Server

You must now register Satellite Server with Content Server so that Content Server can properly manage the Satellite Server cache. Automatic cache management is a built-in feature of Content Server. If you are sure you do not want Content Server to manage the Satellite Server cache, you can skip this step. FatWire recommends that all Satellite Server installations be registered with their respective Content Server installations.

1. From a Windows system, open Content Server Explorer and log into Content Server as a user that has SiteGod privileges.

2. Click on the **System Satellite** tab. A table will appear. This table must be populated with specific values, and each row represents a unique Satellite Server system. Consult the table below for suggested values:

| Column | Value |
|---|---|
| id | A numerical (positive integer) value that identifies this Satellite Server. It must **not** be the same as any other value in the same column in any other row of the table. |
| description | A text description identifying this Satellite Server to users. It is used for reference purposes only. |
| protocol | The protocol on which Satellite Server is accepting requests. This is usually `http`. |
| host | The host name or IP address of the Satellite Server. This must be a host of the actual Satellite Server engine, not the load balancer. |
| port | The port on which Satellite Server is listening for requests. |
| satelliteservletpath | The part of the URL from the port number up to, and including, the name of the Satellite servlet. This is usually `/servlet/Satellite`. |
| flushservletpath | The part of the URL from the port number up to, and including, the name of the FlushServer servlet. This is usually `/servlet/FlushServer`. |
| inventoryservletpath | The part of the URL from the port number up to, and including, the name of the inventory servlet. This is usually `/servlet/Inventory`. |
| username | The username assigned to this Satellite Server. |
| password | The password assigned to this Satellite Server. |

3. From the **File** menu, select **Save** to save the changes.
4. Exit Content Server Explorer.

# Step 5. Configure the Web Server

For configuration instructions, refer to the product documentation of the web server you have chosen to install.

# Step 6. Start Satellite Server

Now you must restart the application server using your application server admin tool.

# Step 7. Test the Configuration

Before you install Satellite Server on other machines, test the first Satellite Server machine to make sure that it is communicating properly with Content Server.

**To test your configuration**

1. Configure your load balancer to send all Content Server requests to the first Satellite Server machine.
2. Using a browser, go to a Satellite Server URL. For example:
   `http://`*`yourhost:yourport`*`/servlet/Satellite?pagename=`*`MyPage`*
   where *`MyPage`* is any page on your Content Server system.
3. If you configured everything properly, your browser displays the selected page.

   If your browser did not display the selected page, review the following:

   - *Did you set up the load balancer properly?* Remember, for this test, every request for Content Server has to go to the Satellite Server machine. (The other machines haven't been set up yet, so they will not know how to handle these requests.)
   - *Did you set the Satellite Server properties properly?* In particular, make sure that you set the host and port to the proper values.
   - *Did you request an invalid page from Content Server?*

# Step 8. Install Satellite Server on Additional Remote Machines

After you have installed and tested Satellite Server on your first remote machine, you must install and configure Satellite Server on your other remote machines as follows:

- If your Satellite Server machines are not homogeneous, you will have to install software for each machine individually by using the installer, and following Steps 1 through 8 of the installation procedure.
- If your machines are homogeneous, however, you can expedite the installation process by deploying the same `cs.war` file to your other Satellite Server hosts.

  To deploy the `cs.war` file, complete the following steps:

  1. Make a copy of the first remote machine's `cs.war` file.
  2. Deploy the `cs.war` file to each of your remote Satellite Server hosts.
  3. Register the Satellite Server engines with Content Server.
  4. Test your remote Satellite Servers.

Chapter 3

# Tuning Satellite Server

After you have installed your Satellite Server hosts (or, in the case of the co-resident Satellite Server host, the host has been installed along with Content Server), you need to tune them in order to achieve the best performance on your Content Server system.

This chapter explains how to tune your Satellite Server hosts. It contains the following sections:

# Tuning the Co-Resident Satellite Server Host

Satellite Server stores pages both in memory and on disk. In the case of the co-resident Satellite Server host, this means that the Satellite Server shares memory with your Content Server installation.

To achieve optimum performance on a system with co-resident Satellite Server, you should adjust the `file_size` property, located in the `satellite.properties` file on the Content Server host.

`The file_size` property separates disk-cached pagelets and blobs from memory-cached pagelets and blobs. To set the `file_size` property, specify a size in kilobytes. Satellite Server caches any pagelet or blob larger than this size to disk, and caches any pagelet or blob smaller than this size to memory.

Setting `file_size` to `0` instructs Satellite Server to cache all pagelets and blobs to disk. Setting `file_size` to a large number (for example, `1,000,000`) instructs Satellite Server to cache all pagelets and blobs to memory. The appropriate setting for your system will be somewhere in between these two extremes.

To determine the proper setting for your system, experiment with values for this property, watching the memory usage on both Content Server and Satellite Server with each alteration. Your goal is to adjust the property so that Satellite Server stores as many items as possible in memory, while still allowing Content Server enough memory to run quickly.

# Tuning Remote Satellite Server Hosts

Because they do not share hardware or memory with your installation of Content Server, you tune your remote Satellite Server hosts differently than you would the co-resident host.

The following sections provide some tuning guidelines.

## Tuning Homogeneous Satellite Server Hosts

If every Satellite Server host has the same CPU, the same amount of physical memory, and the same amount of disk space, then each Satellite Server should have the same set of properties.

In order to determine the appropriate settings for your system, run performance tests while you experiment with various property values, noting which changes improve performance.

The following property values have an especially large impact on performance and should be tuned carefully:

- `readtimeout`
- `blocktimeout`
- `file_size`
- `expiration`
- `cache_max`

For more information about these properties, see “satellite.properties Properties,” on page 27.

For a complete listing of all of the Satellite Server properties, see the *Content Server Property Files Reference*.

After you have found the best settings for your system, you can copy the modified `satellite.properties` file to your other homogeneous remote Satellite Server hosts, as described in "Step 8. Install Satellite Server on Additional Remote Machines," on page 23.

## Tuning Heterogeneous Satellite Server Hosts

If your remote Satellite Server hosts have different strengths, consider adjusting the various caching parameters and your hardware configuration.

For example, if one host has significantly more physical memory than the others, then you might consider increasing the value of the `file_size` property to increase the number of pagelets that get cached in memory.

Evaluate each of the properties listed in "Tuning Homogeneous Satellite Server Hosts," on page 26, as their optimum values will differ with the differing hardware of each host.

You can also improve performance by tuning your hardware to take advantage of machines with more memory and processing power. To do this, configure your load balancer to send more requests to "stronger" hosts, and fewer requests to the hosts with less power and less memory.

# `satellite.properties` Properties

The properties described in this section are those that have the greatest impact on performance, and are the ones that you are most likely to tune. For a complete list of Satellite Server properties, see the *Content Server Property Files Reference.*

### readtimeout

Use the `readtimeout` property to specify a timeout period. The timeout period is the number of milliseconds that a remote Satellite Server waits for Content Server to fulfill a request. For example, `readtimeout` is set to 3000 (3-second wait time). Satellite Server requests a pagelet, but because of some network problem, Content Server fails to respond within three seconds. In this case, Satellite Server writes a message to the Satellite Server startup window.

Setting `readtimeout` to `0` (the default) means that the Java Runtime Environment establishes the timeout period.

This property is ignored by co-resident Satellite Servers.

### blocktimeout

Use the `blocktimeout` property to specify the number of seconds a request waits when another thread is in the process of requesting the same data from the host. Waiting for one thread to return helps reduce the load on the host server when the cache is empty; however, the benefit of a reduced load comes at the expense of individual user response time.

The default value is `45`. A value of `-1` means to wait until the previous thread returns. A value of `0` means to never wait.

This value must be tuned based on the host performance, average request size, and network latency. It is safe to use a large number or `-1`.

### cache_folder

Use this property to specify the directory into which Satellite Server caches pagelets to disk. By default, this value is empty, and Satellite Server will use the servlet context's temporary directory. To use your own value, specify an absolute path to a directory of your choice:

You can specify only one directory. The directory that you specify is not required to be on the same drive as `/SatelliteServer`. FatWire recommends that it is the same drive to improve performance.

### file_size

Use this property to separate disk-cached pagelets and blobs from memory-cached pagelets and blobs. You specify a size (in kilobytes). Satellite Server caches to disk any pagelet or blob larger than this size and caches to memory any pagelet or blob smaller than this size. For example, you set `file_size` to 4. Satellite Server caches to memory any pagelets smaller than 4 kilobytes and caches to disk any pagelets 4 kilobytes or larger.

To optimize Satellite Server performance, FatWire recommends that you experiment with this property.

Setting `file_size` to `0` instructs Satellite Server to cache all pagelets and blobs to disk. Setting `file_size` to a large number (for example, `1,000,000`) instructs Satellite Server to cache all pagelets and blobs to memory. If you have a large amount of memory or a relatively small web site, FatWire recommends caching everything to memory.

The `file_size` property can significantly influence performance. To optimize performance, maximize the amount of memory caching. Be careful not to exceed the host's memory capacity.

### expiration

The `expiration` property sets the default expiration time from for blobs when a cache expiration value is not specifically set for that item with the `satellite.blob` or `RENDER.SATELLITEBLOB` tag that generated the item.

Setting `expiration` as follows tells Satellite Server that blobs should never expire for time reasons:

```
never
```

Such objects are not guaranteed to stay in the cache indefinitely. For example, if the cache is full, Satellite Server still removes objects from cache based on an LRU (least recently used) algorithm.

Setting `expiration` as follows tells Satellite Server not to cache pages, pagelets, or blobs at all:

```
immediate
```

To set a specific set of expiration dates and times, assign a string that uses the following format for the `expiration` property:

*hh*:*mm*:*ss W*/*DD*/*MM*

The value of this property follows the syntax of a `TimePattern` object. The syntax definition is reproduced here for convenience.

**Table 3: TimePattern Syntax**

| Parameter | Legal Values | Description |
|---|---|---|
| *hh* | 0–23 | The hour. For example, `0` means midnight, `12` means noon, `15` means three in the afternoon, and so on. |
| *mm* | 0–59 | The number of minutes past the hour. |
| *ss* | 0–59 | The number of seconds past the minute. |
| *W* | 0–6 | The day of the week. For example, `0` means Sunday, `1` means Monday, and so on. |
| *DD* | 1–31 | The day of the month. |
| *MM* | 1–12 | The month of the year. For example, `1` means January, `2` means February, and so on. |

For example, the following expiration value means "3:30 in the afternoon every Monday and on the 15th of April":

```
15:30:00 1/15/4
```

If you specify a value for both *W* and *DD*, both values apply. Thus, pages expire on Monday (the *W* field) and on the 15th (the *DD* field). To indicate a day-of-week expiration only, place an asterisk in the *DD* field. For example, to indicate expiration at 3:30 in the afternoon every Monday in April, set the expiration value to:

```
15:30:00 1/*/4
```

To indicate a day-of-month expiration only, place an asterisk in the *W* field. For example, to indicate expiration at 3:30 in the afternoon on April 15, set the expiration value to:

```
15:30:00 */15/4
```

Setting the *hh*, *mm*, *ss*, or *MM* fields to an asterisk means all legal values. For example, to indicate expiration at 3:30 in the afternoon on Mondays and the 15th of **every** month, set the expiration value to:

```
15:30:00 1/15/*
```

You can also place multiple values for any of the six fields by separating the values with commas. To represent a range of values, use a minus sign. For example, the following expiration value represents 6:00 (morning), 1:00 (afternoon), and 5:00 (afternoon), Monday through Friday in June.

```
6,13,17:00:00 1-5/*/6
```

To indicate that pages must expire every 15 minutes, set the expiration value to the following:

```
*:15,30,45:0 */*/*
```

The default value is:

```
5:0:0 */*/*
```

This means that everything in the Satellite Server cache expires every day at 5:00 a.m.

### cache_check_interval

When a disk-cached page expires, Satellite Server does not immediately delete the page from the disk. Instead, Satellite Server removes this page from its list of active pages. Satellite Server does, however, contain a cache-pruning thread that runs periodically and deletes expired objects from the cache. Use the `cache_check_interval` property to define the period (in minutes) at which the cache-pruning program should run. The default value is `3600`, meaning that the cache-pruning program runs every 60 hours.

Do not set the `cache_check_interval` value too low; the cache-pruning program consumes a significant amount of resources. However, do not set `cache_check_interval` so high that your disk drive or memory fills up with expired pages.

**Note**

Satellite Server never serves expired pages. If a page is expired but is still in the cache, Satellite Server does not serve it.

### cache_max

Use this property to specify the maximum number of objects (pagelets and blobs) that can be cached (memory cache and disk cache combined) at a time. The default value is `500`, meaning that Satellite Server caches up to 500 objects at a time.

Satellite Server uses an LRU (Least Recently Used) algorithm to determine which objects must be removed from cache when the cache maximum is exceeded. For example, set the `cache_max` to `1000`. When Satellite Server receives a request to cache the 1001st object, Satellite Server removes the object that has not been used in the longest time.

Although you should set `cache_max` to a high level, note that each entry in Satellite Server's cache consumes memory. Also, note that setting `cache_max` to a very high value causes the cache-pruning program to take a longer time to run.

# Log Configuration

Satellite Server uses Apache's Jakarta Commons Logging to record all log messages. By default, no specific JCL configuration information is specified. As a result, JCL will record `INFO`, `WARN` and `ERROR` messages to the console. Users can specify detailed configuration information by placing an empty file called `commons-logging.properties` in the following directory:

*`$SatelliteServerRoot`*`/WEB-INF/classes`

and then editing the file using the Property Editor. The Property Editor provides detailed log configuration information about each property.

To open the Property Editor, run the `settings.bat` batch file (Windows) or the `settings.sh` script (Solaris). Open the `commons-logging.properties` file; it will open with several tabs. Under the `Loggers` tab, among other entries, you will see:

```
com.fatwire.logging.cs.satellite
com.fatwire.logging.cs.satellite.cache
com.fatwire.logging.cs.satellite.host
```

```
com.fatwire.logging.cs.satellite.request
```

These are the loggers that Satellite Server uses. Consult the property descriptions in the Property Editor for information about each logger, as well as the possible values. Under the `Factory` tab, you can choose the type of logger you want Satellite Server to use. By default, the Property Editor sets this to:

```
COM.fatwire.cs.core.logging.TraditionalLog
```

This allows you to write log messages to a log file that is configured under the TraditionalLog tab. (Note that the `logging.file` property is required.)

To send messages to the console, set the `org.apache.commons.logging.Log` property to either blank or `COM.FutureTense.Logging.StandardLog`. When you are done, save the changes, exit the Property Editor, and restart Satellite Server by restarting the application server. Consult the JCL website at `http://jakarta.apache.org/commons/logging/` for more information about JCL.

# Index

## T