Configuring Distributed ColdFusion

This document describes how to set up ColdFusion in a distributed configuration.

Contents

The content in this document was inadvertently dropped from the *Advanced ColdFusion Administration* manual.

Using ColdFusion in a Distributed Configuration

ColdFusion 4.5 can be configured in a distributed manner where the ColdFusion engine is running on a separate computer from the Web server. Running ColdFusion in this way might be called distributed or remote ColdFusion.

To run distributed ColdFusion, you must make the following changes to a standard installation:

- On the Web server side, you must notify the ColdFusion Web server plug-in that you want it to talk to a ColdFusion engine on another machine. You do this simply by making appropriate entries in an INI file.
- On the ColdFusion engine side, you must run an additional piece of software, known as the Network Listener Module, that listens for incoming ColdFusion requests and forwards them to the ColdFusion engine running on that machine. The ColdFusion engine itself is a standard release version of the engine with no special modifications to accommodate remoting.

In addition to allowing the ColdFusion engine to be located on a separate machine from the Web server, distributed ColdFusion provides the following unique capabilities:

- It allows the machine hosting the Web server to potentially be of a different architecture from the machine hosting the ColdFusion engine.
- It allows more than one Web server to be served by the same ColdFusion engine.

To provide some degree of security for the data being transferred between the Web server and the ColdFusion engine, that conversation is encrypted using a standard, 56-bit DES encryption algorithm.

Although it's possible for a ColdFusion engine to simultaneously service both local and remote requests, it is not possible for a single Web server to simultaneously dispatch both local and remote ColdFusion requests. When starting up, the ColdFusion Web server plug-in determines if it's to run in local or remote mode and remains in that mode until it's shutdown.

Distributed ColdFusion and clustering

The distributed ColdFusion configuration is not supported when ColdFusion is also configured for clustering. The reason is that the clustering component in ColdFusion, which runs as part of the Web server, needs to be able to communicate with the ColdFusion engine. This arrangement assumes that the ColdFusion engine and the Web server are on the same machine, which is not necessarily the case in a distrubuted configuration.

Configuring Distributed ColdFusion

Before trying to run ColdFusion in a distributed configuration, you must perform a standard installation of ColdFusion on all the machines involved. On the computer running the Web server, this guarantees that the ColdFusion server plug-ins are correctly loaded by the Web server. On the computer running the ColdFusion engine, this guarantees that the engine is set up and operating correctly.

Having complete, standard installations of ColdFusion available on all machines also provides a useful baseline environment so that validation can be done in the absence of the remote extensions. Should problems arise using ColdFusion in remote mode, it‘s possible to run ColdFusion locally to determine whether or not the problems are related to the distributed configuration.

If, after successfully testing your remote configuration, you wish, for security reasons, to disable the ColdFusion engine installed on the computer hosting the Web server, you can do this easily by renaming the following executable files in the `cfusion/bin` (Windows) `coldfusion/bin` (Solaris) directory:

- `cfserver`
- `cfrdsservice`
- `cfexec`

This prevents any ColdFusion server-side process from running while generally preserving your ColdFusion configuration.

Using the modified plug-in

In ColdFusion 4.5 all the Web server plug-ins are remote-capable so no special installation is required. All you need to do is let the plug-in know that you want to run in remote mode. You do this by putting the following information in an INI file and putting that file in the root directory of your ColdFusion installation on the machine running the Web server. That INI file must be named `cfremote.ini`. To enhance security, this INI file may be optionally set to be automatically deleted after being read at startup.

Here is a sample of the INI file with comments explaining what the various fields do. This sample may be cut and pasted and used as a template to get started.

INI files are also available online as part of the standard ColdFusion distrubution. On Windows, they can be found in the root directory of your ColdFusion installation. On UNIX they can be found in the subdirectory `/opt/coldfusion/CFRemoting`.

```
;-------------------------------------------------------------------
;
; Sample INI file for ColdFusion Remoting.
;
; Place this file in the root directory of your ColdFusion installation.
; It must be named "cfremote.ini".
;
; !IMPORTANT! * All values (the strings on the right hand side of
;               the equals sign) must be quoted using double
;               quotes.
;             * All info is case insensitive..
;             * Lines beginning with a semicolon are treated as
;               comments and are ignored.
;
; Use REMOTEPATH and LOCALPATH to map filenames from one host to
; another.
; For example, if you are running Apache webserver on Solaris and you
; want to connect to a ColdFusion server running on Windows NT, you
; might set these values as:
;
; LOCALPATH = “/usr/local/apache/htdocs”
; REMOTEPATH = “C:\Inetpub\wwwroot\”
;
; This would map any ColdFusion pages in /usr/local/apache/htdocs/fcml/
; to C:\Inetpub\wwwroot\cfml\

; Use this to turn on/off the remoting capability.
; Valid values: Yes, No.
;
REMOTING = "YES"

; Use this to specify the IP address of the remote computer running
; the ColdFusion Server.
;
; Valid values: a valid IP address, e.g.: 139.56.205.102.
;
IP = "205.181.21.61"

; Use this to specify the port on that computer on which the remote
; ColdFusion Network Listener Module is listening.
;
; Valid values: a valid port number (integer).
;
PORT = "1234"

; Use this to specify that the data sent between the machine running
; the Web server and the machine running the ColdFusion Application
; Server be encrypted.
;
; Valid values: Yes, No.
;
ENCRYPTION = "YES"

; Use this to specify the key used to encrypt the data.
;
; Valid values: any string of up to 127 ASCII chars.
;
KEY = "doglips"

; Use this to have this INI file be deleted after it is read at
; startup. (This is a security feature as it keeps your key from
; being read by others.)
;
; Valid values: Yes, No.
```

```
;
DELETE = "NO"

; Use this to write a message to the ColdFusion "webserver.log"
; confirming that remoting is active and what startup parameters (except
; the encryption key) were used.
;
; Valid values: Yes, No.
;
MESSAGE = "YES"
;
;---------------------------------------------------------------------
```

As with all warning and error messages from any of the ColdFusion Web server plug-ins, such text is written to the ColdFusion log file `webserver.log` in the log subdirectory of the directory into which you installed ColdFusion (on the machine hosting the Web server.) This file should be the first place you look if you encounter problems running ColdFusion in a distributed configuration since, for a variety of practical and security reasons, ColdFusion will not run in distributed mode if any information in the INI file is missing or incomplete.

The Network Listener Module (NLM)

The NLM is a stand-alone program that acts as a network front-end for the standard ColdFusion Server. It runs on the same computer on which the ColdFusion Server is running. It listens for incoming requests via TCP/IP and forwards them on to the local ColdFusion Server. The ColdFusion Server then processes those requests, returning the results to the listener module which, in turn, returns them via the original TCP/IP connection. It is a silent, background process with no user interaction. On NT, it runs as an NT service. On UNIX, it runs as a daemon. For debugging or other special purposes, it may also be run as a command line program by specifying the appropriate command line option (-i) at startup.

Installing the module on Windows NT

On NT, the module consists of a single executable file, `cfdist.exe`. Before you can run the listener as an NT service, you must perform the following installation step.

To install the network listener module as a service:

1 Run the listener with the following special command line argument:

```
cfdist.exe -sINSTALL
```

2 If installation was successful, it should now appear on your Services list under the name ColdFusion NetListener. If it doesn't show up, look in the module's log file, `distributed.log` in the log subdirectory of your ColdFusion installation, for information about why the install failed.

Note
Once you've installed the module as an NT service you cannot move the executable file unless you uninstall and reinstall it in its new location.

Once installed as a service, you can start, stop, pause or continue the listener's operation as you would any NT service. You can start or stop the listener independent of any of the other ColdFusion services although, of course, the listener must be running to receive remote network requests. Note that when starting the service (from the NT Services Control Panel applet), you will need to specify -p switch and possibly the -k switch in the Startup Parameters box in the Services applet.

Please refer to the list of command line options below.

To uninstall the listener

Invoke `cfdist.exe` with the `-sREMOVE` command line option. Notice of successful removal will be written to the listener log.

Installing the module on UNIX

On UNIX, the listener module consists of a single executable file, in this case named simply `cfdist`. It is not necessary to perform any special installation step on UNIX.

To start the listener as a daemon:

Type the executable's name (without the -i switch) and the process will start. Because it's running as a daemon, the command will return immediately having launched the process in the background. You will probably use at least the -p switch when starting the daemon.

Please refer to the list of command line options below.

To stop the daemon process:

You need to kill it by its process ID. Use the `ps` command to get the PID and then kill the process as demonstrated below.

```
  ps -deaf  | grep cfdist | grep -v grep
```

It returns the PID in a string something like:

```
  ckintzin   980     1  0 15:48:12 ?        0:00 cfdist
```

The first number is the PID. Use it in the kill command to stop the process:

```
  kill -INT 980
```

Repeating the `ps` command should now return nothing, indicating the process is now dead.

Listener Module Command Line Options

The Network Listener Module (NLM) executable, `cfdist` (`cfdist.exe` on Windows NT) takes the following command line options at startup. Of these options, you'll probably only use the -p option on a regular basis.

Option	Description
-v	Verbose. This option prints out confirmation of the command line options in use, and on what port the program is running. It also prints information about each connection that comes in. This can be useful to confirm that requests are, indeed, reaching the remote computer.
-p*nnnn*	Port number where *nnnn* is the port number. If no port number is specified, the program automatically selects an unused port on which to run. In most cases, you use this option to guarantee that you're using the same port as the remote Web server.
-i	Interactive. Run from a command line not as a daemon/service. In order for verbose commentary to appear on the terminal, you must be running in interactive mode. Aside from the display of debugging output, however, there is no difference in operation between running the program from a command line or running it as a daemon/service.
-r	Reuse. (UNIX Only) If the specified port appears to be in use, try to use it anyway. Sometimes TCP/IP connections don't get closed down immediately. In those cases the connections can take a few minutes to timeout and close down. Unfortunately, these lingering connections will prevent the program from restarting on the same port because it thinks that port is in use. To overcome this and allow you to restart without waiting or switching to another port, you can use this option. Be careful, however, not to use this option indiscriminately as it could result in multiple versions of the listener running at the same time.
-k*xxxx*	Key for encryption. (where *xxxx* is the string used as the key) The key may be any string of printable ASCII chars up to 127 characters long.
-sINSTALL	Setup, install-mode. (NT Only) Install the process as an NT service. Its service name will be ColdFusion NetListener.
-sREMOVE	Setup, remove-mode. (NT Only) Uninstall the service.

The program will print out a list of available options along with a brief description of their purpose anytime you enter an unknown option at the command line.

Using the INI file to specify startup options

It is also possible to specify startup options for the listener in an INI file. This INI file is similar to the INI file required on the Web server side, but is available on the ColdFusion engine side as a convenience (since all the required information may be supplied as command line options at startup time.) Below is a template for this INI file. If used, it must be placed in the root directory of your ColdFusion installation (on the machine hosting the ColdFusion engine), and it must be named `cfdist.ini`. To enhance security, this INI file may be optionally set to be automatically deleted after being read at startup.

```
;--------------------------------------------------------------------
;
; Sample INI file for CFDist (AKA the "ColdFusion Listener Module").
;
; Place this file in the root directory of your ColdFusion installation.
; It must be named "cfdist.ini"
;
; !IMPORTANT! * All values (the strings on the right hand side of
;               the equals sign) must be quoted using double
;               quotes.
;             * All info is case insensitive..
;             * Lines beginning with a semicolon are treated as
;               comments and are ignored.
;

; Use this to specify the port at which to listen for incoming
; ColdFusion requests
;
; Valid values: a valid port number (integer).
;
PORT = "1234"

; Use this to specify that the data sent between the machine running
; the Web server and the machine running this program be encrypted.
;
; Valid values: Yes, No.
;
ENCRYPTION = "YES"

; Use this to specify the key used to encrypt the data.
;
; Valid values: any string of up to 127 ASCII chars.
;
KEY = "doglips"

; Use this to have this INI file be deleted after it is read at
; startup. (This is a security feature as it keeps your key from
; being read by others.)
;
; Valid values: Yes, No.
;
DELETE = "NO"
```

```
; Use this to write a message to the ColdFusion "remote.log" confirming
; that remoting is active and what startup parameters (except the
; encryption key) were used.
;
; Valid values: Yes, No.
;
MESSAGE = "YES"
;
;---------------------------------------------------------------------
```

The listener also writes various informative messages to the `remote.log` file in the log subdirectory of your ColdFusion installation (on the machine hosting the ColdFusion engine.)