

- 4 管理者ガイド -



4. 管理者ガイド

4.1 CSSJサーバー

ウェブ上で帳票やドキュメントを出力するといったサーバーサイド・アプリケーションからCSSJを利用する場合、通常はCSSJサーバーを常駐させ、ソケット通信を用いてアプリケーションからアクセスします。これは、データベース・サーバー(DBMS)などによく見られるやり方です。

アプリケーションからのアクセスのためには、言語ごとに用意されたドライバを用いますが、詳細については「開発者ガイド」に記述されています。このドキュメントでは、OSやネットワークの基本的な知識を持ったサーバー管理者向けに、CSSJサーバーの構成方法を説明します。

4.2 JavaVMのインストール

CSSJサーバーにはJavaVMがバンドルされていないため、Java2 SDKのバージョン1.4.1以降を事前にインストールする必要があります。Java2 SDKはSun Microsystems社(<http://java.sun.com/>)または他の互換VMベンダから入手可能です。

Java VMにはウィンドウ・アプリケーションで主に用いられるClient VMと、サーバー用のServer VMがありますが、CSSJサーバーはServer VMを使います。Java2 SDKではなくJava2 Runtime Edition(J2RE)ではServer VMが含まれていない場合があります。Sun Java2 SDK以外を使う場合は、Server VMが含まれていることを確認してください。

4.3 CSSJのインストール

CSSJサーバーのパッケージにはインストーラと、TARアーカイブがあります。インストーラは簡単にインストールできるため、特にWindows環境ではおすすめです。システム管理者がUNIX上でのJava環境の運用に精通している場合は、TARアーカイブからのインストールをおすすめします。

また、特殊なOSや、Java VMが通常とは異なる場所にインストールされている場合、インストーラが実行できないことがあります。そのような場合はTARアーカイブからインストールしなければなりません。

4.3.1 インストーラを使う場合

Windows

Windowsでは、`cssj-server-[version].exe`をダブルクリックして起動してください。後は指示に従ってインストールを完了させるだけです。なお、アンインストールはコントロールパネルの「プログラムの追加と削除」から行ってください。

インストールが完了したら、[ライセンスキーの配置](#)に進んでください。

Linux/UNIX/Mac OS X

Linuxのインストーラは`cssj-server-[version].bin`です。UNIXおよびMac OS Xでは`cssj-server-[version].sh`です。最初に、`chmod +x cssj-server-[version].bin`とするなどして、実行権限をつけておいて下さい。また、インストールはインストール先のディレクトリに書き込む権限のあるユーザー(一般にはroot権限)で行ってください。

X Window System上またはMac OS Xのターミナルから起動する場合は、以下のようにインストーラを実行することで、GUIのインストーラが起動します。後は、インストーラの指示に従ってインストールを完了してください。

```
Linuxの場合
# ./cssj-server-[version].bin

UNIX/Mac OS Xの場合
# ./cssj-server-[version].sh
```

例 4.1 インストーラの実行

X Window Systemが使えない場合、またはGUIのインストーラでフォントが化けるなどの問題が生じる場合は、コマンドラインでインストーラを実行することができます。コマンドライン・インストーラを起動するには、以下のように-consoleオプションを付け加えます。

```
Linuxの場合
# ./cssj-server-[version].bin -console

UNIX/Mac OS Xの場合
# ./cssj-server-[version].sh -console
```

例 4.2 コマンドラインでインストールする

なお、アンインストールはインストール先ディレクトリの中にできる _uninstディレクトリ内のuninstaller.binまたはuninstaller.shを実行してください。インストーラ同様に-consoleオプションを使うことができます。インストール後、ログ、作業ファイル、ライセンスキーなどの、後で作成されたファイルが残ります。必要なければ、ディレクトリごと削除しても構いません。

インストールが完了したら、[ライセンスキーの配置](#)に進んでください。

4.3.2 TARアーカイブからのインストール

cssj-server-[version].tar.gzはCSSJサーバーをTARアーカイブにまとめたものです。UNIXのtarコマンドあるいは適当な展開ツールで展開し、適当な場所に配置してください。

TARアーカイブにはJava VMの場所を、自動的に検索するラウンチャーが含まれないため、Java VMの場所を環境変数"JAVA_HOME"に設定する必要があります。環境変数はWindowsでは「マイコンピュータ」を右クリック 「詳細設定」タブをクリック 「環境変数」ボタンをクリックで設定可能です。UNIXでは起動スクリプトなどに記述します。詳細は各OSのマニュアルを参照してください。

環境変数JAVA_HOMEには、JavaVMのインストールディレクトリを設定してください。例えば、"C:\jdk1.4.1_06","/usr/local/j2sdk1.4.1_06","/opt/IBMJava2-ppc-142"といった値になります。

4.3.3 ライセンスキーの配置

CSSJサーバーはライセンスキーを設定するまでは、起動することができません。

弊社より入手されたライセンスキー・ファイルをresourcesディレクトリにコピーしてください。ライセンスキー・ファイルは正確に"license-key"という名前で行ってください。

4.3.4 管理プログラム(cssjadmin)

サーバーの起動には、CSSJサーバーの管理プログラム(cssjadmin)を使います。Windowsにインストーラでインストールした場合は、スタートメニューからCSSJ Serverのアイコンを選択することで、管理パネルが開きます。ここから、CSSJサーバーの起動・停止あるいは設定を行うことができます。

X Window SystemあるいはMac OS Xでは、cssjadminを引数無しでターミナルから実行することで管理パネルが開きます。

cssjadminのコマンドライン・インターフェースは、cssjadmin コマンド [引数...]という形式になります。

CSSJサーバーを起動するコマンドは、startです。以下に例を示します。

```
Windowsの場合
> .\cssjadmin.exe start

Linux/UNIX/Mac OS Xの場合
# ./cssjadmin start

WindowsでTARアーカイブからインストールした場合
> .\cssjadmin.bat start

Linux/UNIX/Mac OS XでTARアーカイブからインストールした場合
# ./cssjadmin.sh start
```

例 4.3 CSSJサーバーの起動

管理パネルの場合も、あるいはコマンドラインからCSSJサーバーを起動した場合も、一番最初にパスワードの設定を求められます。

パスワードは、ソケット通信でCSSJサーバーにアクセスする時の認証に必要なものです。パスワードを設定しない限り、CSSJサーバーにクライアントがアクセスすることはできません。設定したパスワード

ドはクライアントのプログラマに伝えてください。

パスワードの設定の後、アクセス制御の設定が必要です。これは管理パネルからも設定が可能ですがアクセス制御ファイルを直接修正しても構いません。アクセス制御ファイルはCSSJサーバーへのアクセスを許可または禁止するクライアントのIPアドレスを指定するものです。インストール直後の状態ではローカルマシン(127.0.0.1)からのアクセスだけが許可されています。設定を行うまでは、リモートからクライアントがアクセスすることはできません。

以上のセキュリティ機能はインストール直後にCSSJサーバーが無防備になることを防ぐためと、LAN内での簡単なアクセス制御を実現するためのものです。**CSSJサーバーを直接グローバルに公開することは推奨しません**。実運用においては、ファイアウォールなどで外部からのCSSJサーバーへのアクセスをブロックするなど、適切な処置を施してください。

4.3.5 パスワードの再設定

パスワードの再設定は管理パネルから可能ですが、コマンドライン・インターフェースも用意されています。以下のようにcssjadminでpasswdコマンドを実行してください。

```
# ./cssjadmin passwd
Enter password:secret
Retype password:secret
#
```

例 4.4 パスワードの再設定

これで、暗号化されたパスワードが後述する設定ファイル(cssjd.properties)に書き込まれます。パスワードの変更はサーバーが実行中でもすぐに反映されます。

4.3.6 制御パスワード

cssjadminがサーバーを停止したり状態を取得するためには、ローカルホストへのソケット通信を利用しています。その際の認証に用いるのが制御パスワードです。共用サーバーなどで、悪意のあるユーザーによって勝手にサーバーが制御されるのを防ぐために、制御パスワードとして意味のない文字列を設定しておくことをおすすめします。

```
# ./cssjadmin ctrlpasswd
Enter password:ddX#$sVg
Retype password:ddX#$sVg
#
```

例 4.5 制御パスワードの再設定

4.3.7 アクセス制御ファイル

アクセス制御ファイルはリモートIPアドレスベースの簡単なアクセス制御を実現するもので、confディレクトリ内のaccess.txtです。管理パネルで編集できるほか、テキストエディタで直接編集しても構いません。アクセス制御はサーバーが実行中でもすぐに反映されます。

各行に、"allow 許可するIPアドレス"または"deny 禁止するIPアドレス"という形式で記述します。また、IPアドレスに"*"(アスタリスク)を指定すると、全てのIPアドレスが対象になることを意味します。

CSSJサーバーはクライアントから接続があった場合、アクセス制御ファイルの先頭の規則から順に検査するため、マッチする規則のうち、いちばん最初にあるものが適用されます。また、マッチする規則がない場合はアクセスを禁止します。

以下の設定例では、192.168.10.8からのアクセスは2行目の規則によって禁止されます。4行目の規則は意味を持ちません。また、最後に全てのIPアドレスに対してアクセスを許可しているため、192.168.10.10,192.168.10.8および192.168.0.11以外からのアクセスは全て許可されます。

```
deny    192.168.0.10
deny    192.168.0.8
deny    192.168.0.11
allow   192.168.0.8
allow   *
```

例 4.6 アクセス制御

4.3.8 設定ファイル

サーバーの設定は、confディレクトリ内のcssjd.proprtiesファイルに書かれます。管理パネルでの設定が反映されますが、テキストエディタで直接編集しても構いません。ただし、サーバーを実行中でもすぐに反映される設定と、サーバーの再起動が必要な設定があるので注意してください。各行に「プロパティ名=値」の形式で記述してください。

表 4.1 サーバーの再起動が必要なプロパティ

プロパティ名	タイトル	デフォルト値	説明
jp.cssj.cssjd.port	待ち受けポート	8099	クライアントプログラムが接続するポート番号です。
jp.cssj.cssjd.controlPort	制御ポート	8098	サーバーの状態の監視、停止といった制御を行うためのポートです。
jp.cssj.cssjd.backlog	バックログ	30	接続待ち行列のサイズです。この数のクライアントが接続待ち状態になると、他のクライアントは直ちに接続を拒否されます。拒否されたクライアント側では接続エラーとなります。
jp.cssj.cssjd.timeout	タイムアウト	180	クライアントとの通信の待ち時間の最大秒数です。これ以上の時間クライアントとの通信がなかった場合、クライアントとの接続が切断されます。
jp.cssj.cssjd.minThreads	最小スレッド数	10	起動直後に準備されるスレッドの数です。
jp.cssj.cssjd.maxThreads	最大スレッド数	20	同時に処理を実行できるクライアントの数の限界です。
jp.cssj.cssjd.warmup	ウォームアップ	true	trueに設定すると、起動直後にウォームアップのためにサンプルのファイルを1つ変換します。falseでは起動時にウォームアップを行いません。CSSJは最初のファイルの変換時にフォントの読み込みなど各種初期化処理を実行します。これには一定の時間がかかるため、ウォームアップを行うことで最初のファイルの変換が速くなります。

表 4.2 サーバーの再起動が不必要なプロパティ

名前	デフォルト値	タイトル	説明
jp.cssj.cssjd.controlCommand	CSSJCTRL	制御パスワード	サーバーの状態の監視、停止といった制御を行うためのパスワードです。ASCII文字列で指定してください。
jp.cssj.cssjd.paddword	N/A	パスワード	暗号化されたパスワードです。設定するにはcssjadminを使ってください。

4.4 サーバーの実行

4.4.1 待ち受けポート

CSSJサーバーのデフォルトの待ち受けポート番号は8099です。この番号は設定により変更することが出来ます。8099番ポートはアプリケーションが接続するためのポートです。また、デフォルトで8098番ポートは制御用ポートとなり、cssjadminが使用します。アプリケーションから接続されることはありません。

これらのポートがインターネットに直接公開されることは好ましくありません。ファイアウォールなどで必要なクライアント以外からのアクセスをブロックしてください。

4.4.2 起動と停止

CSSJサーバーの起動と停止は管理パネルから行うことができますが、cssjadminにstartまたはstopオプションをつけることでコマンドラインから行えます。オプションをstartとすると起動、stopとすると停止します。

```
# ./cssjadmin start
```

例 4.7 起動オプション

```
# ./cssjadmin stop
```

例 4.8 停止オプション

サーバーが停止するとき、接続中の全てのクライアントが処理を終えるまで待ちます。強制的に停止する場合は、引数killを用いてください。このとき、クライアントとの接続が中断され、クライアント・アプリケーション側では接続エラーが発生します。

```
# ./cssjadmin kill
```

例 4.9 強制的に停止する

4.5 運用のヒント

4.5.1 パフォーマンス・チューニング

一般的な使用方法であれば、デフォルトの設定で概ね良好に動作します。ただし、ある状況では以下の調整を施すことで性能が改善されることがあります。

最大スレッド数を減らす

メモリやハードディスクの容量が小さな環境で動作させる場合、あるいは非常に複雑か、大きなドキュメントを処理する場合に、ディスクアクセスが急激に増えて動作が遅くなったり、OutOfMemoryエラーが発生することがあります。これは、多数のアクセスが集中し、サーバーが処理しきれなくなった場合におこります。最大スレッド数を減らすことである程度アクセスの集中を分散して処理効率をよくなり、処理しきれなくなりそうな場合はアクセスを拒否することで、OutOfMemoryエラーの発生を防ぐことができます。

最大スレッド数を増やす

メモリやハードディスクの容量が十分にあり、かつ複数CPUを搭載するなど高性能なハードウェアでは、非常にアクセスが集中した状態でもマシンの性能が十分に生かされない(CPU稼働率が100%にならない)場合があります。これは、実際のCPUの能力に対して、同時に実行している処理の数が少なすぎるのが原因です。最大スレッド数の数を増やすことで、より多くの処理を同時にこなし、CPUの稼働率を上げる効果があります。

4.5.2 負荷分散・多重化

不特定多数からアクセスが集中するようなサイトや、ノンストップで運用する必要のあるサイトでは負荷分散および多重化が必要になることがあります。ウェブサーバー(アプリケーションサーバー)とCSSJサーバーを別のハードウェアに分散するといったケースでは、CSSJサーバーとCSSJドライバを用いることで簡単に実現できます。しかし、複数のウェブサーバーとCSSJサーバーを使って負荷分散あるいは多重化する機能は、CSSJサーバーおよびドライバ自体にはありません。そのため、アプリケーション側が運用で対応することになります。

おそらく最も簡単な方法は、ウェブサーバーとCSSJサーバーの間に負荷分散装置(ロードバランサー)を入れることです。ウェブサーバーとCSSJサーバーの間の通信は、1)ソケットの接続 2)元データの送信 3)変換結果の受信 4)ソケットの切断 という単純なものです。この一連の処理は、CSSJサーバーを強制的に停止しない限り完了することが保証されます。そのため、負荷分散装置に接続されている複数のCSSJサーバーのうち1つを安全に停止することができます。ただし、ハードウェアの障害などの理由でCSSJサーバーが強制的に停止した場合は、接続中のクライアントは影響を受けます。

4.6 ログ

4.6.1 ログイングの概要

CSSJはアクセスログの記録や、エラー発生時の原因を調査するためにログを出力します。ログはlogsディレクトリに出力されます。デフォルトでは、警告メッセージがログファイルに出力されます。

ログの設定ファイルはconf/logging.propertiesです。出力するログを重要なエラーのみにとどめたい場合は、jp.cssj.level = INFOをjp.cssj.level = SEVEREに変更してください。逆に、アクセス記録などの詳細なログも含める場合は、jp.cssj.access.level = OFFをjp.cssj.access.level = INFOに変更してください。

なお、ログの設定はCSSJサーバーの再起動後に有効になります。

CSSJはJ2SE1.4からJavaの標準機能として追加されたログイング機能を用いています。より高度な設定方法はJavaのドキュメントか、書籍などを参照してください。