SettingTheJvmLocale

Information about how to setup the Java Virtual Machine (JVM) to use a specific locale, which governs character encodings, default formatting of dates, etc.

We won't copy existing JVM docs here, just give an overview of the problem and refer to existing documentation.

Reference information

For more detailed information see:

- i18n FAQ
- Introduction to i18n

Or search "user.language", "user.country" and/or "LC_ALL" on the web.

The problem

Although many parts of Cocoon use configurable character encodings, other parts of Cocoon or of the libraries that it uses rely on the current JVM locale.

If care is not taken to configure this locale correctly, processing non-ASCII characters might give unexpected results.

As the JVM is often able to select the "right" locale from the host environment, explicit configuration is often not needed, and problems might occur when moving to another host where the environment settings cause the JVM to use another locale.

Default Locale selection at JVM startup

At startup, the JVM tries to selects a Locale that matches the environment of the JVM process.

On unix/Linux systems, the system locale is defined by a set of environment variables (LANG, LC_ALL, etc.) which most JVMs use to set their default locale.

(can anyone add some info about Windows here? I guess the JVM uses the International settings from the control panel, according to the user identity of the JVM process).

Explicit Locale selection through system properties

System properties can also be used to define the locale, like for example

```
java -Duser.language=th -Duser.country=TH -Duser.variant=TH SomeClass
```

Using the Locale class

The locale can also be set programatically using the java.util.Locale class.

Locale availabilty

Note also that a given JVM has a limited set of Locale, some JVMs have only US locales for example. If you need a specific locale, make sure it is available in your JVM.