

Validación de normas de codificación de código java

Versión 1.0
ABRIL 2008



CONTROL DE CAMBIOS

Fecha	Versión	Cambios
08/04/2008	1.0	Primera versión del documento

1 TABLA DE CONTENIDO

1	<i>TABLA DE CONTENIDO</i>	3
2	<i>Introducción</i>	4
3	<i>Herramientas</i>	5
4	<i>Ficheros de reglas</i>	5
5	<i>Configuración de CheckStyle</i>	5
6	<i>Configuración de PMD</i>	8
7	<i>Configuración del Formateador (Formatter)</i>	9
8	<i>Configuración de CleanUp (Limpiar)</i>	10

2 Introducción

Un estándar de codificación adecuado y buen diseño orientado a objetos son prácticas que deben estar presentes en todos los proyectos.

Es necesario imponer una normalización a la hora de desarrollar código debido a diferentes factores que nos van a influir en nuestra facilidad de mantenimiento posterior y costes económicos derivados de tal actividad.

- Un porcentaje elevado del coste del software va asociado a su mantenimiento.
- Las aplicaciones son dinámicas, crecen y se modifican, requieren de mantenimiento y evolución.
- Las convenciones de código nos facilitarán la interpretación fácil de nuestro código, más rápida y más clara.
- Debemos pues garantizar el cumplimiento con estas convenciones.

Se deben utilizar las convenciones de código para el lenguaje de programación Java de SUN Microsystems.

Estas convenciones están disponibles en la url:
<http://java.sun.com/docs/codeconv/>

El uso de convenciones de código se debe realizar desde la primera línea de código que se implemente y no esperar a que se haya finalizado el desarrollo ya que en este caso el coste del cambio es muy elevado.

La herramienta Eclipse nos permite mediante determinados plugins validar que se están cumpliendo estas convenciones. ICM ha personalizado los ficheros de reglas para que se realicen las validaciones que ICM determine. Estos ficheros están disponibles en la web de soja.

La validación se debe realizar según se van codificando las clases. En el caso de que ya tengamos código desarrollado y queramos ajustarlo a esta codificación Eclipse dispone de utilidades que nos permiten formatear el código para que se cumplan muchas de las reglas que se validan.

3 Herramientas

Las herramientas que se van a utilizar para la validación son:

Herramientas	URL
Eclipse 3.3	http://www.eclipse.org/downloads/
Code Style Formatter de Eclipse	Incluido en eclipse
Code Style Clean up de Eclipse	Incluido en eclipse
Plugin de Checkstyle	http://eclipse-cs.sourceforge.net/
Plugin de PMD	http://pmd.sourceforge.net/

4 Ficheros de reglas

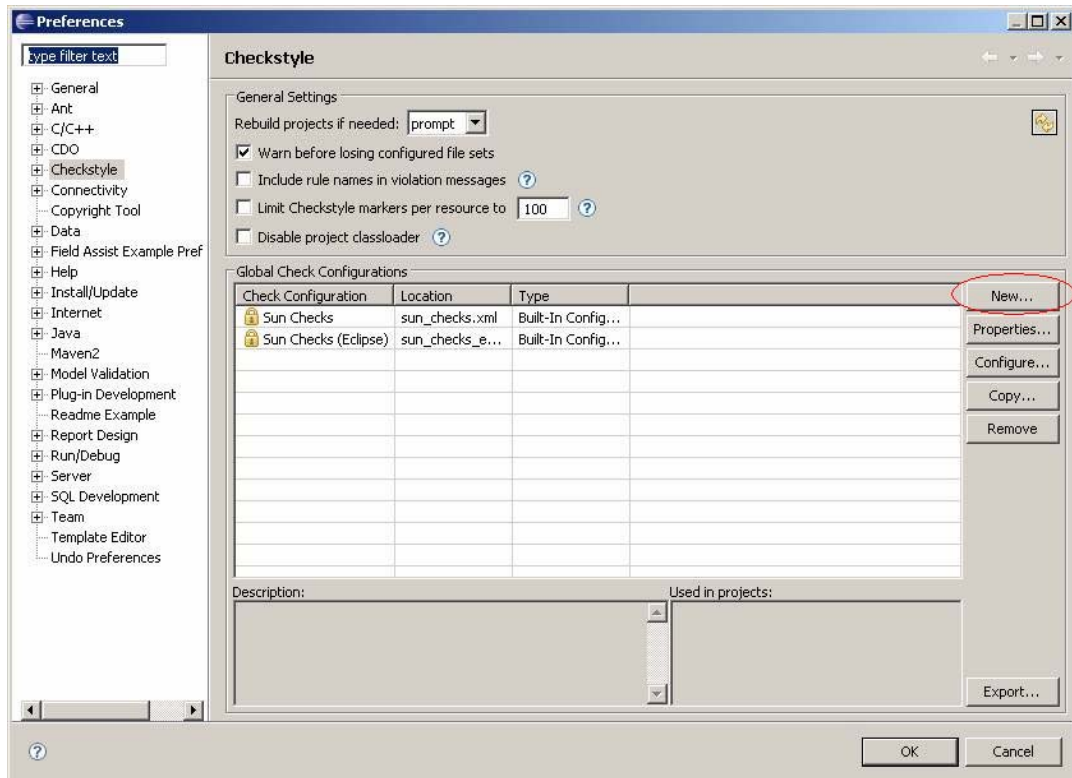
ICM proporciona los ficheros con las reglas de las herramientas anteriores en la web de soja (<http://gestiona.madrid.org/soja>) en el apartado Area de Aseguramiento de la Callidad/Recursos Adicionales.

Los ficheros son:

- REGLAS CC JAVA: Reglas de convenciones de codificación JAVA en ICM para los plugins de PMD y CheckStyle v.1.3
- PROFILES CC JAVA: Profiles para Eclipse de ayuda al cumplimiento de las reglas de convenciones de codificación JAVA en ICM v.1.3

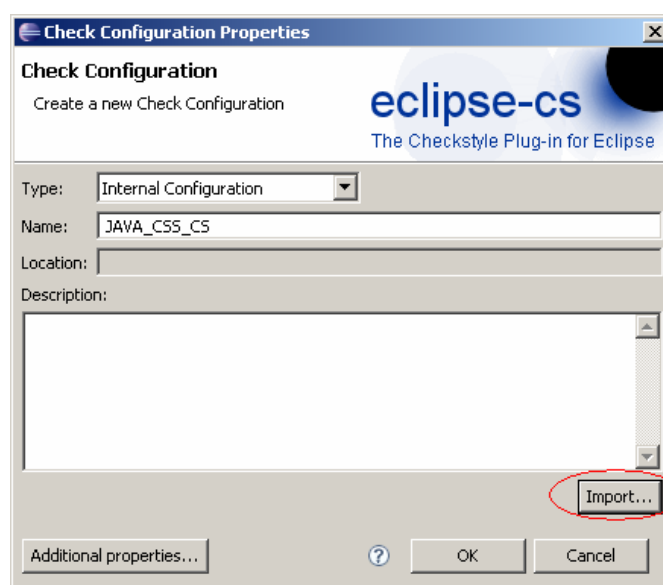
5 Configuración de CheckStyle

Para configurar el plugin de CheckStyle para aplicar las reglas de convenciones de codificación JAVA en ICM deberemos seleccionar la opción “Preferences” del menú “Window” y una vez dentro seleccionar CheckStyle como muestra la imagen siguiente

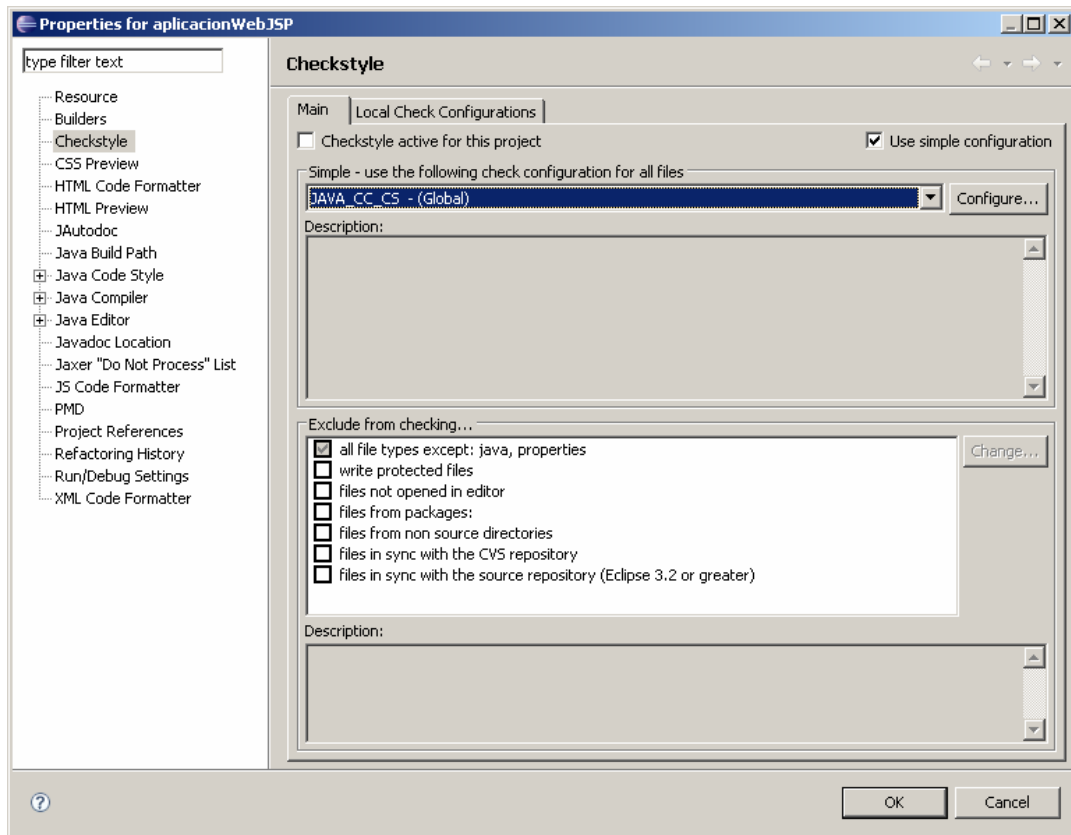


Pulsamos el botón New y aparecerá la siguiente ventana en la que se añadirá el nombre "JAVA_CSS_CS" para la nueva configuración y pulsaremos "Import" e importaremos el archivo JAVA.CC.CS.xml . Por último pulsamos OK en las dos ventanas.

El anterior fichero está disponible en el recurso adicional REGLAS CC JAVA en la web de soja_int en Área de Aseguramiento de la Calidad.

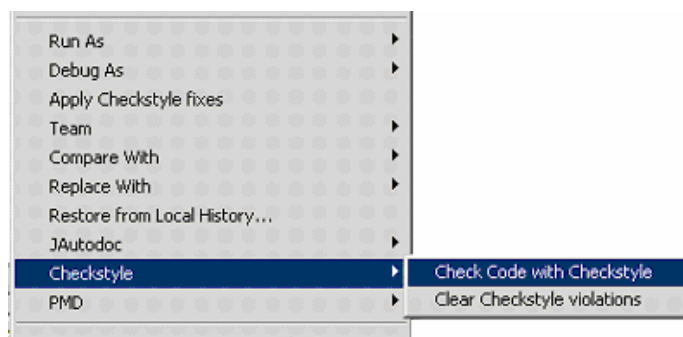


Para configurar nuestro proyecto seleccionamos la opción de menú Project → Properties y una vez dentro se selecciona la opción Checkstyle como muestra la imagen siguiente



En el panel “Simple – use the following check configuration for all files” seleccionamos JAVA_CSS_CS.

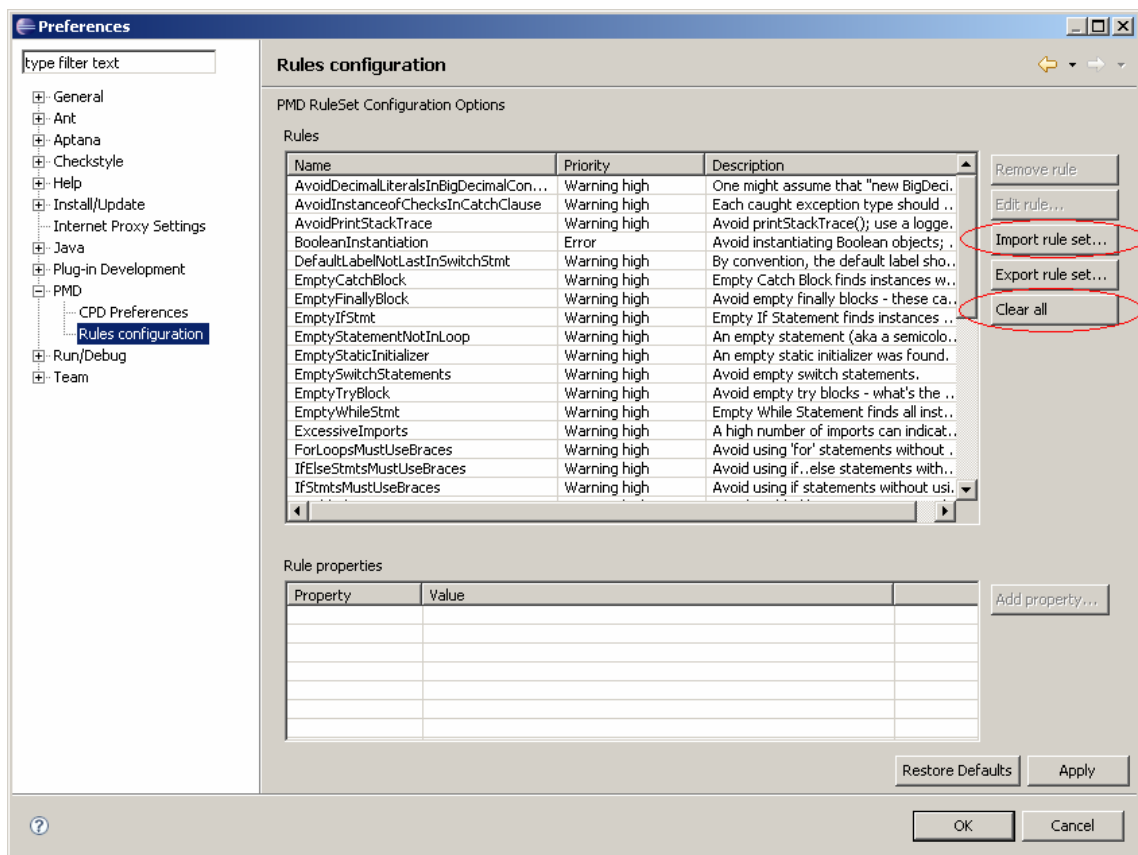
Para aplicar las reglas de CheckStyle sobre nuestro código pulsamos botón derecho del ratón y pulsamos sobre CheckStyle → Check Code with CheckStyle.



En las clases java donde no se cumpla alguna de las reglas aparecerá la línea marcada en rojo.

6 Configuración de PMD

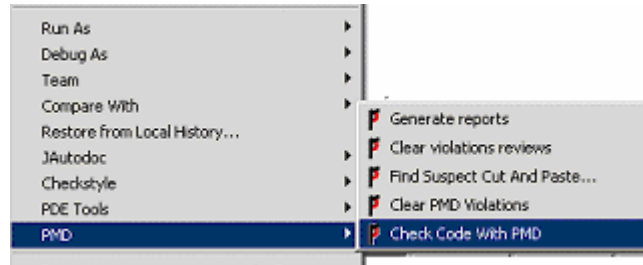
Para configurar el plugin de PMD para aplicar las reglas de convenciones de codificación JAVA en ICM deberemos seleccionar de nuevo la opción “Preferences” del menú “Window” y una vez dentro seleccionar PMD → Rules configuration como muestra la imagen siguiente



Pulsaremos la opción “Clear all” para quitar las reglas anteriores. A continuación pulsaremos el botón “Import rule set...” y seleccionaremos el fichero JAVA.CC.PMD.xml y pulsamos OK en ambas ventanas.

Dicho archivo está disponible en el recurso adicional REGLAS CC JAVA en la web de soja_int en Área de Aseguramiento de la Calidad.

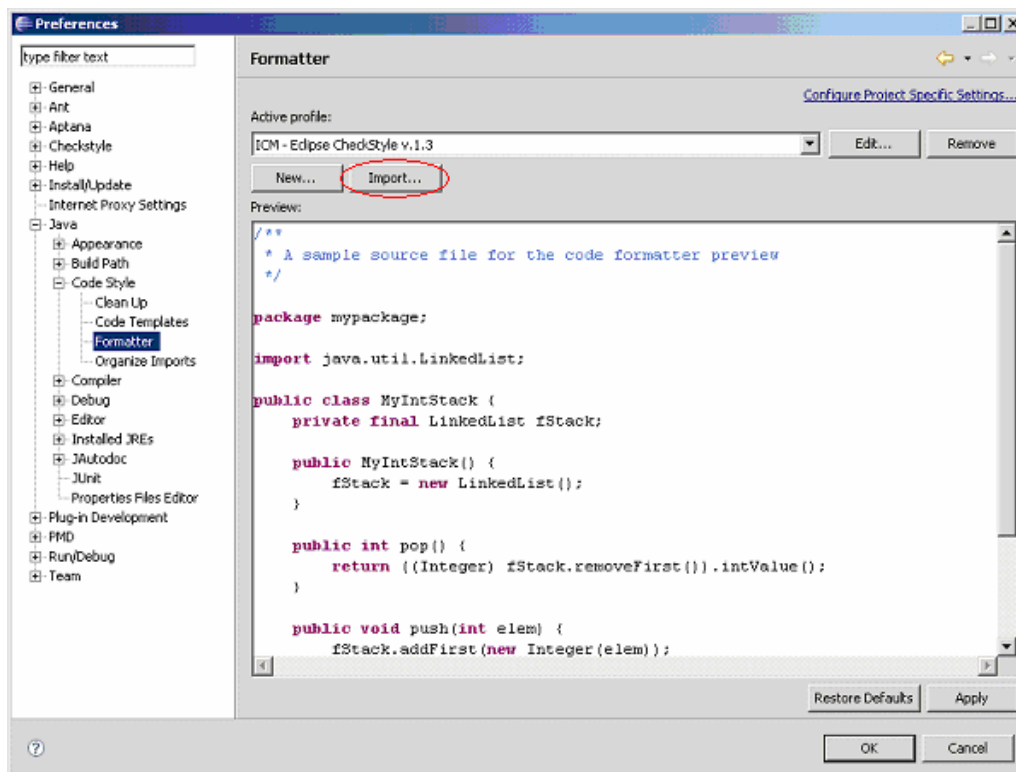
Para aplicar las reglas de PMD sobre nuestro código seleccionamos nuestro proyecto pulsamos botón derecho del ratón y pulsamos sobre la opción PMD → Check Code With PMD.



A continuación PMD chequeará el código y se abrirá la perspectiva PMD donde en las distintas ventanas se mostrarán las reglas que no se cumplen.

7 Configuración del Formateador (Formatter)

Es posible configurar eclipse para que formatee automáticamente el código de forma que cumpla el máximo número posible de objetivos de checkstyle, para hacer esto deberemos ir al menú de preferencias de eclipse (Window -> Preferences) y una vez hay ir al menú Java → Code Style → Formatter, como puede verse en la siguiente imagen

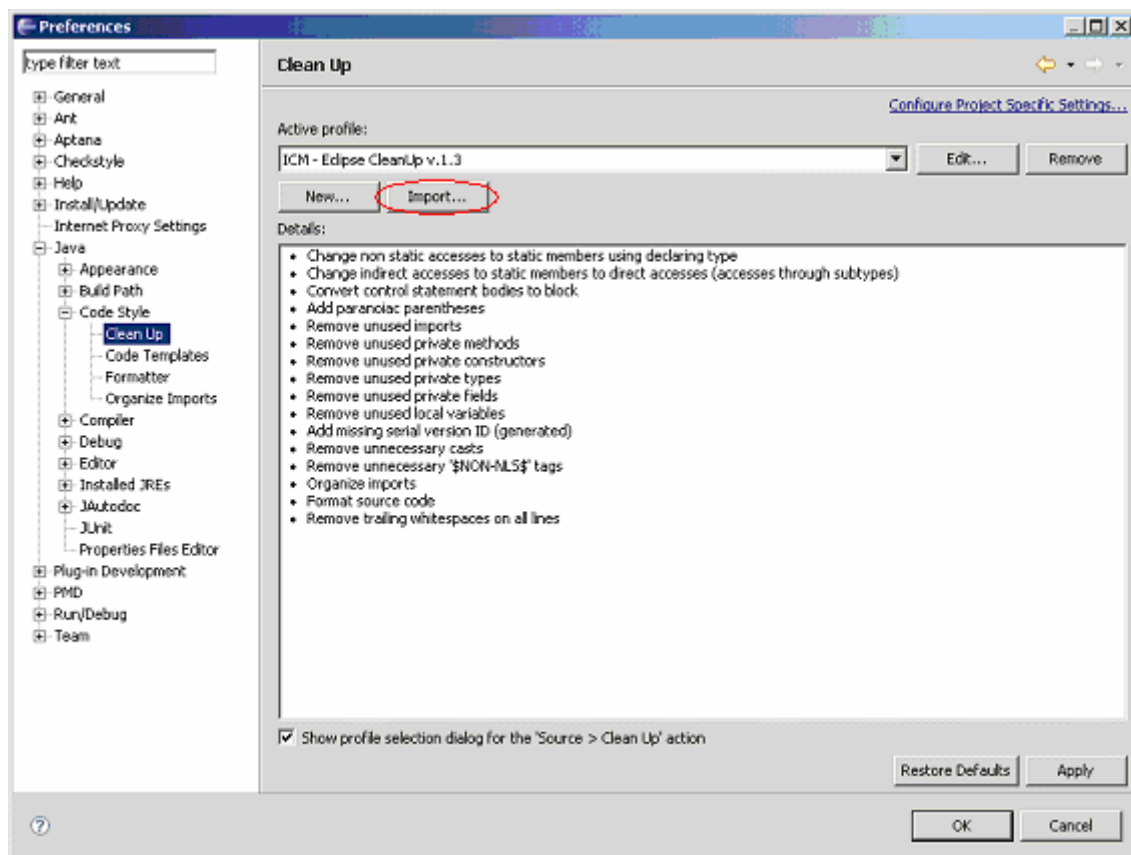


Una vez dentro de esta opción pulsaremos el botón “Import...” (“Importar”) e importaremos el archivo JAVA.CC.ECLIPSE formatter profile.xml.

Este fichero está disponible en el recurso adicional PROFILES CC JAVA en la web de soja_int en Area de Aseguramiento de la Calidad.

8 Configuración de CleanUp (Limpiar)

Además del formateador es posible configurar eclipse para que al hacer una limpieza de código este haga los cambios de acuerdo con los objetivos de checkstyle, para ello deberemos seleccionar de nuevo la opción “Preferences” del menú “Window” y una vez dentro seleccionar Java → Code Style → Clean Up como puede verse en esta imagen



En esta pantalla seleccionaremos “Import...” (Importar) e importaremos el siguiente archivo JAVA.CC.ECLIPSE cleanup profile.xml. Este fichero está disponible en el recurso adicional PROFILES CC JAVA en la web de soja_int en Area de Aseguramiento de la Calidad.

