

How to implement a reloadable MessageResources for Struts

How to implement a reloadable [MessageResources](#) for Struts

Commons Configuration support for reloadable files can be leveraged to implement a dynamic resource bundle for Struts. This is done easily by extending the [MessageResourcesFactory](#) and [MessageResources](#) classes provided by Struts.

[ConfigurationMessageResources](#)

```
/*
 * Copyright 2004 Emmanuel Bourg
 *
 * Licensed under the Apache License, Version 2.0 (the "License")
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *      http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

import java.net.URL;
import java.util.Locale;
import java.util.Map;

import org.apache.commons.configuration.Configuration;
import org.apache.commons.configuration.ConfigurationException;
import org.apache.commons.configuration.PropertiesConfiguration;
import org.apache.commons.configuration.reloading.FileChangedReloadingStrategy;
import org.apache.struts.util.MessageResources;
import org.apache.struts.util.MessageResourcesFactory;

/**
 * MessageResources based on reloadable configurations.
 *
 * @author Emmanuel Bourg
 */
public class ConfigurationMessageResources extends MessageResources {

    /** Configurations associées à leur Locale respective. */
    protected Map configurations;

    public ConfigurationMessageResources(MessageResourcesFactory factory, String config, boolean returnNull) {
        super(factory, config, returnNull);
    }

    public String getMessage(Locale locale, String key) {

        // get the configuration for the specified locale
        Configuration resource = getConfiguration(this.config + "_" + locale.getLanguage() + ".properties");

        if (resource == null || !resource.containsKey(key)) {
            // look for the key in the root configuration
            resource = getConfiguration(this.config + ".properties");
        }

        return resource != null ? resource.getString(key, null) : null;
    }

    /**
     * Load the specified configuration from the classpath and initialize it.
     */
}
```

```

    */
    private Configuration getConfiguration(String name) {

        Configuration configuration = null;

        URL url = Thread.currentThread().getContextClassLoader().getResource(name);

        if (url != null) {
            PropertiesConfiguration pc = new PropertiesConfiguration();
            PropertiesConfiguration.setDelimiter('\uffff'); // disable string splitting
            pc.setURL(url);
            pc.setReloadingStrategy(new FileChangedReloadingStrategy());

            try {
                pc.load();
                configuration = pc;
            } catch (ConfigurationException e) {
                e.printStackTrace();
            }
        }

        return configuration;
    }
}

```

ConfigurationMessageResourcesFactory

```

/*
 * Copyright 2004 Emmanuel Bourg
 *
 * Licensed under the Apache License, Version 2.0 (the "License")
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

import org.apache.struts.util.MessageResourcesFactory;
import org.apache.struts.util.MessageResources;

/**
 * Factory for ConfigurationMessageResources.
 *
 * @author Emmanuel Bourg
 */
public class ConfigurationMessageResourceFactory extends MessageResourcesFactory {

    public MessageResources createResources(String config) {
        return new ConfigurationMessageResources(this, config, this.returnNull);
    }
}

```

struts-config.xml

The last step is to declare the factory in the struts-config.xml file:

```
<?xml version="1.0"?>
<struts-config>

    ...

    <message-resources factory="ConfigurationMessageResourceFactory" parameter="yourResourceBundle"/>
</struts-config>
```