001/\*  
002 \* Licensed to the Apache Software Foundation (ASF) under one or more  
003 \* contributor license agreements. See the NOTICE file distributed with  
004 \* this work for additional information regarding copyright ownership.  
005 \* The ASF licenses this file to You under the Apache License, Version 2.0  
006 \* (the "License"); you may not use this file except in compliance with  
007 \* the License. You may obtain a copy of the License at  
008 \*  
009 \* http://www.apache.org/licenses/LICENSE-2.0  
010 \*  
011 \* Unless required by applicable law or agreed to in writing, software  
012 \* distributed under the License is distributed on an "AS IS" BASIS,  
013 \* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
014 \* See the License for the specific language governing permissions and  
015 \* limitations under the License.  
016 \*/  
017package org.apache.commons.beanutils.converters;  
018  
019import java.io.File;  
020  
021/\*\*  
022 \* {@link org.apache.commons.beanutils.Converter} implementaion that handles conversion  
023 \* to and from <b>java.io.File</b> objects.  
024 \* <p>  
025 \* Can be configured to either return a <i>default value</i> or throw a  
026 \* <code>ConversionException</code> if a conversion error occurs.  
027 \*  
028 \* @version $Id$  
029 \* @since 1.6  
030 \*/  
031public final class FileConverter extends AbstractConverter {  
032  
033 /\*\*  
034 \* Construct a <b>java.io.File</b> <i>Converter</i> that throws  
035 \* a <code>ConversionException</code> if an error occurs.  
036 \*/  
037 public FileConverter() {  
038 super();  
039 }  
040  
041 /\*\*  
042 \* Construct a <b>java.io.File</b> <i>Converter</i> that returns  
043 \* a default value if an error occurs.  
044 \*  
045 \* @param defaultValue The default value to be returned  
046 \* if the value to be converted is missing or an error  
047 \* occurs converting the value.  
048 \*/  
049 public FileConverter(final Object defaultValue) {  
050 super(defaultValue);  
051 }  
052  
053 /\*\*  
054 \* Return the default type this <code>Converter</code> handles.  
055 \*  
056 \* @return The default type this <code>Converter</code> handles.  
057 \* @since 1.8.0  
058 \*/  
059 @Override  
060 protected Class<?> getDefaultType() {  
061 return File.class;  
062 }  
063  
064 /\*\*  
065 \* <p>Convert the input object into a java.io.File.</p>  
066 \*  
067 \* @param <T> The target type of the conversion.  
068 \* @param type Data type to which this value should be converted.  
069 \* @param value The input value to be converted.  
070 \* @return The converted value.  
071 \* @throws Throwable if an error occurs converting to the specified type  
072 \* @since 1.8.0  
073 \*/  
074 @Override  
075 protected <T> T convertToType(final Class<T> type, final Object value) throws Throwable {  
076 if (File.class.equals(type)) {  
077 return type.cast(new File(value.toString()));  
078 }  
079  
080 throw conversionException(type, value);  
081 }  
082}