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;  
018  
019/\*\*  
020 \* <p>{@link BeanUtilsBean} implementation that creates a  
021 \* {@link ConvertUtilsBean2} and delegates conversion to  
022 \* {@link ConvertUtilsBean#convert(Object, Class)}.  
023 \* </p>  
024 \*  
025 \* <p>  
026 \* To configure this implementation for the current context ClassLoader invoke  
027 \* <code>BeanUtilsBean.setInstance(new BeanUtilsBean2());</code>  
028 \* </p>  
029 \*  
030 \* <p>  
031 \* BeanUtils 1.7.0 delegated all conversion to String to the converter  
032 \* registered for the <code>String.class</code>. One of the improvements in  
033 \* BeanUtils 1.8.0 was to upgrade the {@link Converter} implementations so  
034 \* that they could handle conversion to String for their type (e.g.  
035 \* IntegerConverter now handles conversion from an Integer to a String as  
036 \* well as String to Integer).  
037 \* </p>  
038 \*  
039 \* <p>  
040 \* In order to take advantage of these improvements BeanUtils needs to change  
041 \* how it gets the appropriate {@link Converter}. This functionality has been  
042 \* implemented in the new {@link ConvertUtilsBean#lookup(Class, Class)} and  
043 \* {@link ConvertUtilsBean#convert(Object, Class)} methods. However changing  
044 \* {@link BeanUtilsBean} to use these methods could create compatibility  
045 \* issues for existing users. In order to avoid that, this new  
046 \* {@link BeanUtilsBean} implementation has been created (and the associated  
047 \* {@link ConvertUtilsBean2}).  
048 \* </p>  
049 \*  
050 \* @see ConvertUtilsBean2  
051 \* @version $Id$  
052 \* @since 1.8.0  
053 \*/  
054public class BeanUtilsBean2 extends BeanUtilsBean {  
055  
056 /\*\*  
057 \* <p>Constructs an instance using new property  
058 \* and conversion instances.</p>  
059 \*/  
060 public BeanUtilsBean2() {  
061 super(new ConvertUtilsBean2());  
062 }  
063  
064 /\*\*  
065 \* <p>Convert the value to an object of the specified class (if  
066 \* possible).</p>  
067 \*  
068 \* @param value Value to be converted (may be null)  
069 \* @param type Class of the value to be converted to  
070 \* @return The converted value  
071 \*/  
072 @Override  
073 protected Object convert(final Object value, final Class<?> type) {  
074 return getConvertUtils().convert(value, type);  
075 }  
076}