JavaScript is disabled on your browser.

[Skip navigation links](#1fob9te)

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Use](http://docs.google.com/class-use/TransformerUtils.html)
* [Tree](http://docs.google.com/package-tree.html)
* [Deprecated](http://docs.google.com/deprecated-list.html)
* [Index](http://docs.google.com/index-all.html)
* [Help](http://docs.google.com/help-doc.html)
* [Prev Class](http://docs.google.com/org/apache/commons/collections4/Transformer.html)
* [Next Class](http://docs.google.com/org/apache/commons/collections4/Trie.html)
* [Frames](http://docs.google.com/index.html?org/apache/commons/collections4/TransformerUtils.html)
* [No Frames](http://docs.google.com/TransformerUtils.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* Constr |
* [Method](#3znysh7)
* Detail:
* Field |
* Constr |
* [Method](#tyjcwt)

org.apache.commons.collections4

## Class TransformerUtils

* [java.lang.Object](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true)
  + org.apache.commons.collections4.TransformerUtils
* public class TransformerUtils  
  extends [Object](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true)
* TransformerUtils provides reference implementations and utilities for the Transformer functor interface. The supplied transformers are:
  + Invoker - returns the result of a method call on the input object
  + Clone - returns a clone of the input object
  + Constant - always returns the same object
  + Closure - performs a Closure and returns the input object
  + Predicate - returns the result of the predicate as a Boolean
  + Factory - returns a new object from a factory
  + Chained - chains two or more transformers together
  + If - calls one transformer or another based on a predicate
  + Switch - calls one transformer based on one or more predicates
  + SwitchMap - calls one transformer looked up from a Map
  + Instantiate - the Class input object is instantiated
  + Map - returns an object from a supplied Map
  + Null - always returns null
  + NOP - returns the input object, which should be immutable
  + Exception - always throws an exception
  + StringValue - returns a java.lang.String representation of the input object
* Since v4.1 only transformers which are considered to be safe are Serializable. Transformers considered to be unsafe for serialization are:
  + Invoker
  + Clone
  + Instantiate

Since: 3.0

### Method SummaryAll Methods Static Methods Concrete Methods Deprecated Methods

|  |  |
| --- | --- |
| * + Modifier and Type | * + Method and Description |
| * + static <T> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<T,T> | * + [asTransformer](http://docs.google.com/org/apache/commons/collections4/TransformerUtils.html#asTransformer-org.apache.commons.collections4.Closure-)([Closure](http://docs.google.com/org/apache/commons/collections4/Closure.html)<? super T> closure) Creates a Transformer that calls a Closure each time the transformer is used. |
| * + static <I,O> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<I,O> | * + [asTransformer](http://docs.google.com/org/apache/commons/collections4/TransformerUtils.html#asTransformer-org.apache.commons.collections4.Factory-)([Factory](http://docs.google.com/org/apache/commons/collections4/Factory.html)<? extends O> factory) Creates a Transformer that calls a Factory each time the transformer is used. |
| * + static <T> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<T,[Boolean](https://docs.oracle.com/javase/7/docs/api/java/lang/Boolean.html?is-external=true)> | * + [asTransformer](http://docs.google.com/org/apache/commons/collections4/TransformerUtils.html#asTransformer-org.apache.commons.collections4.Predicate-)([Predicate](http://docs.google.com/org/apache/commons/collections4/Predicate.html)<? super T> predicate) Creates a Transformer that calls a Predicate each time the transformer is used. |
| * + static <T> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<T,T> | * + [chainedTransformer](http://docs.google.com/org/apache/commons/collections4/TransformerUtils.html#chainedTransformer-java.util.Collection-)([Collection](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true)<? extends [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<? super T,? extends T>> transformers) Create a new Transformer that calls each transformer in turn, passing the result into the next transformer. |
| * + static <T> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<T,T> | * + [chainedTransformer](http://docs.google.com/org/apache/commons/collections4/TransformerUtils.html#chainedTransformer-org.apache.commons.collections4.Transformer...-)([Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<? super T,? extends T>... transformers) Create a new Transformer that calls each transformer in turn, passing the result into the next transformer. |
| * + static <T> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<T,T> | * + [cloneTransformer](http://docs.google.com/org/apache/commons/collections4/TransformerUtils.html#cloneTransformer--)() Gets a transformer that returns a clone of the input object. |
| * + static <I,O> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<I,O> | * + [constantTransformer](http://docs.google.com/org/apache/commons/collections4/TransformerUtils.html#constantTransformer-O-)(O constantToReturn) Creates a Transformer that will return the same object each time the transformer is used. |
| * + static <I,O> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<I,O> | * + [exceptionTransformer](http://docs.google.com/org/apache/commons/collections4/TransformerUtils.html#exceptionTransformer--)() Gets a transformer that always throws an exception. |
| * + static <I,O> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<I,O> | * + [ifTransformer](http://docs.google.com/org/apache/commons/collections4/TransformerUtils.html#ifTransformer-org.apache.commons.collections4.Predicate-org.apache.commons.collections4.Transformer-org.apache.commons.collections4.Transformer-)([Predicate](http://docs.google.com/org/apache/commons/collections4/Predicate.html)<? super I> predicate, [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<? super I,? extends O> trueTransformer, [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<? super I,? extends O> falseTransformer) Create a new Transformer that calls one of two transformers depending on the specified predicate. |
| * + static <T> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<T,T> | * + [ifTransformer](http://docs.google.com/org/apache/commons/collections4/TransformerUtils.html#ifTransformer-org.apache.commons.collections4.Predicate-org.apache.commons.collections4.Transformer-)([Predicate](http://docs.google.com/org/apache/commons/collections4/Predicate.html)<? super T> predicate, [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<? super T,? extends T> trueTransformer) Create a new Transformer that calls the transformer if the predicate is true, otherwise the input object is returned unchanged. |
| * + static <T> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<[Class](https://docs.oracle.com/javase/7/docs/api/java/lang/Class.html?is-external=true)<? extends T>,T> | * + [instantiateTransformer](http://docs.google.com/org/apache/commons/collections4/TransformerUtils.html#instantiateTransformer--)() Gets a Transformer that expects an input Class object that it will instantiate. |
| * + static <T> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<[Class](https://docs.oracle.com/javase/7/docs/api/java/lang/Class.html?is-external=true)<? extends T>,T> | * + [instantiateTransformer](http://docs.google.com/org/apache/commons/collections4/TransformerUtils.html#instantiateTransformer-java.lang.Class:A-java.lang.Object:A-)([Class](https://docs.oracle.com/javase/7/docs/api/java/lang/Class.html?is-external=true)<?>[] paramTypes, [Object](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true)[] args) Creates a Transformer that expects an input Class object that it will instantiate. |
| * + static <I,O> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<I,O> | * + [invokerTransformer](http://docs.google.com/org/apache/commons/collections4/TransformerUtils.html#invokerTransformer-java.lang.String-)([String](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html?is-external=true) methodName) Gets a Transformer that invokes a method on the input object. |
| * + static <I,O> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<I,O> | * + [invokerTransformer](http://docs.google.com/org/apache/commons/collections4/TransformerUtils.html#invokerTransformer-java.lang.String-java.lang.Class:A-java.lang.Object:A-)([String](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html?is-external=true) methodName, [Class](https://docs.oracle.com/javase/7/docs/api/java/lang/Class.html?is-external=true)<?>[] paramTypes, [Object](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true)[] args) Gets a Transformer that invokes a method on the input object. |
| * + static <I,O> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<I,O> | * + [mapTransformer](http://docs.google.com/org/apache/commons/collections4/TransformerUtils.html#mapTransformer-java.util.Map-)([Map](https://docs.oracle.com/javase/7/docs/api/java/util/Map.html?is-external=true)<? super I,? extends O> map) Creates a Transformer that uses the passed in Map to transform the input object (as a simple lookup). |
| * + static <T> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<T,T> | * + [nopTransformer](http://docs.google.com/org/apache/commons/collections4/TransformerUtils.html#nopTransformer--)() Gets a transformer that returns the input object. |
| * + static <I,O> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<I,O> | * + [nullTransformer](http://docs.google.com/org/apache/commons/collections4/TransformerUtils.html#nullTransformer--)() Gets a transformer that always returns null. |
| * + static <T> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<T,[String](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html?is-external=true)> | * + [stringValueTransformer](http://docs.google.com/org/apache/commons/collections4/TransformerUtils.html#stringValueTransformer--)() Gets a transformer that returns a java.lang.String representation of the input object. |
| * + static <I,O> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<I,O> | * + [switchMapTransformer](http://docs.google.com/org/apache/commons/collections4/TransformerUtils.html#switchMapTransformer-java.util.Map-)([Map](https://docs.oracle.com/javase/7/docs/api/java/util/Map.html?is-external=true)<I,[Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<I,O>> objectsAndTransformers) Create a new Transformer that uses the input object as a key to find the transformer to call. |
| * + static <I,O> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<I,O> | * + [switchTransformer](http://docs.google.com/org/apache/commons/collections4/TransformerUtils.html#switchTransformer-java.util.Map-)([Map](https://docs.oracle.com/javase/7/docs/api/java/util/Map.html?is-external=true)<[Predicate](http://docs.google.com/org/apache/commons/collections4/Predicate.html)<I>,[Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<I,O>> predicatesAndTransformers) Create a new Transformer that calls one of the transformers depending on the predicates. |
| * + static <I,O> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<I,O> | * + [switchTransformer](http://docs.google.com/org/apache/commons/collections4/TransformerUtils.html#switchTransformer-org.apache.commons.collections4.Predicate:A-org.apache.commons.collections4.Transformer:A-)([Predicate](http://docs.google.com/org/apache/commons/collections4/Predicate.html)<? super I>[] predicates, [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<? super I,? extends O>[] transformers) Create a new Transformer that calls one of the transformers depending on the predicates. |
| * + static <I,O> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<I,O> | * + [switchTransformer](http://docs.google.com/org/apache/commons/collections4/TransformerUtils.html#switchTransformer-org.apache.commons.collections4.Predicate:A-org.apache.commons.collections4.Transformer:A-org.apache.commons.collections4.Transformer-)([Predicate](http://docs.google.com/org/apache/commons/collections4/Predicate.html)<? super I>[] predicates, [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<? super I,? extends O>[] transformers, [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<? super I,? extends O> defaultTransformer) Create a new Transformer that calls one of the transformers depending on the predicates. |
| * + static <I,O> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<I,O> | * + [switchTransformer](http://docs.google.com/org/apache/commons/collections4/TransformerUtils.html#switchTransformer-org.apache.commons.collections4.Predicate-org.apache.commons.collections4.Transformer-org.apache.commons.collections4.Transformer-)([Predicate](http://docs.google.com/org/apache/commons/collections4/Predicate.html)<? super I> predicate, [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<? super I,? extends O> trueTransformer, [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<? super I,? extends O> falseTransformer) Deprecated.  as of 4.1, use [ifTransformer(Predicate, Transformer, Transformer)](http://docs.google.com/org/apache/commons/collections4/TransformerUtils.html#ifTransformer-org.apache.commons.collections4.Predicate-org.apache.commons.collections4.Transformer-org.apache.commons.collections4.Transformer-) |

### Methods inherited from class java.lang.[**Object**](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true)[clone](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true#clone--), [equals](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true#equals-java.lang.Object-), [finalize](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true#finalize--), [getClass](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true#getClass--), [hashCode](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true#hashCode--), [notify](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true#notify--), [notifyAll](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true#notifyAll--), [toString](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true#toString--), [wait](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true#wait--), [wait](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true#wait-long-), [wait](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true#wait-long-int-)

### Method Detail

#### exceptionTransformer public static <I,O> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<I,O> exceptionTransformer() Gets a transformer that always throws an exception. This could be useful during testing as a placeholder.Type Parameters: I - the input type O - the output type Returns: the transformer See Also: [ExceptionTransformer](http://docs.google.com/org/apache/commons/collections4/functors/ExceptionTransformer.html)

#### nullTransformer public static <I,O> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<I,O> nullTransformer() Gets a transformer that always returns null.Type Parameters: I - the input type O - the output type Returns: the transformer See Also: [ConstantTransformer](http://docs.google.com/org/apache/commons/collections4/functors/ConstantTransformer.html)

#### nopTransformer public static <T> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<T,T> nopTransformer() Gets a transformer that returns the input object. The input object should be immutable to maintain the contract of Transformer (although this is not checked).Type Parameters: T - the input/output type Returns: the transformer See Also: [NOPTransformer](http://docs.google.com/org/apache/commons/collections4/functors/NOPTransformer.html)

#### cloneTransformer public static <T> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<T,T> cloneTransformer()

* + - Gets a transformer that returns a clone of the input object. The input object will be cloned using one of these techniques (in order):
      * public clone method
      * public copy constructor
      * serialization clone

Type Parameters: T - the input/output type Returns: the transformer See Also: [CloneTransformer](http://docs.google.com/org/apache/commons/collections4/functors/CloneTransformer.html)

#### constantTransformer public static <I,O> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<I,O> constantTransformer(O constantToReturn) Creates a Transformer that will return the same object each time the transformer is used.Type Parameters: I - the input type O - the output type Parameters: constantToReturn - the constant object to return each time in the transformer Returns: the transformer. See Also: [ConstantTransformer](http://docs.google.com/org/apache/commons/collections4/functors/ConstantTransformer.html)

#### asTransformer public static <T> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<T,T> asTransformer([Closure](http://docs.google.com/org/apache/commons/collections4/Closure.html)<? super T> closure) Creates a Transformer that calls a Closure each time the transformer is used. The transformer returns the input object.Type Parameters: T - the input/output type Parameters: closure - the closure to run each time in the transformer, not null Returns: the transformer Throws: [NullPointerException](https://docs.oracle.com/javase/7/docs/api/java/lang/NullPointerException.html?is-external=true) - if the closure is null See Also: [ClosureTransformer](http://docs.google.com/org/apache/commons/collections4/functors/ClosureTransformer.html)

#### asTransformer public static <T> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<T,[Boolean](https://docs.oracle.com/javase/7/docs/api/java/lang/Boolean.html?is-external=true)> asTransformer([Predicate](http://docs.google.com/org/apache/commons/collections4/Predicate.html)<? super T> predicate) Creates a Transformer that calls a Predicate each time the transformer is used. The transformer will return either Boolean.TRUE or Boolean.FALSE.Type Parameters: T - the input type Parameters: predicate - the predicate to run each time in the transformer, not null Returns: the transformer Throws: [NullPointerException](https://docs.oracle.com/javase/7/docs/api/java/lang/NullPointerException.html?is-external=true) - if the predicate is null See Also: [PredicateTransformer](http://docs.google.com/org/apache/commons/collections4/functors/PredicateTransformer.html)

#### asTransformer public static <I,O> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<I,O> asTransformer([Factory](http://docs.google.com/org/apache/commons/collections4/Factory.html)<? extends O> factory) Creates a Transformer that calls a Factory each time the transformer is used. The transformer will return the value returned by the factory.Type Parameters: I - the input type O - the output type Parameters: factory - the factory to run each time in the transformer, not null Returns: the transformer Throws: [NullPointerException](https://docs.oracle.com/javase/7/docs/api/java/lang/NullPointerException.html?is-external=true) - if the factory is null See Also: [FactoryTransformer](http://docs.google.com/org/apache/commons/collections4/functors/FactoryTransformer.html)

#### chainedTransformer public static <T> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<T,T> chainedTransformer([Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<? super T,? extends T>... transformers) Create a new Transformer that calls each transformer in turn, passing the result into the next transformer.Type Parameters: T - the input/output type Parameters: transformers - an array of transformers to chain Returns: the transformer Throws: [NullPointerException](https://docs.oracle.com/javase/7/docs/api/java/lang/NullPointerException.html?is-external=true) - if the transformers array or any of the transformers is null See Also: [ChainedTransformer](http://docs.google.com/org/apache/commons/collections4/functors/ChainedTransformer.html)

#### chainedTransformer public static <T> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<T,T> chainedTransformer([Collection](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true)<? extends [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<? super T,? extends T>> transformers) Create a new Transformer that calls each transformer in turn, passing the result into the next transformer. The ordering is that of the iterator() method on the collection.Type Parameters: T - the input/output type Parameters: transformers - a collection of transformers to chain Returns: the transformer Throws: [NullPointerException](https://docs.oracle.com/javase/7/docs/api/java/lang/NullPointerException.html?is-external=true) - if the transformers collection or any of the transformers is null See Also: [ChainedTransformer](http://docs.google.com/org/apache/commons/collections4/functors/ChainedTransformer.html)

#### ifTransformer public static <T> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<T,T> ifTransformer([Predicate](http://docs.google.com/org/apache/commons/collections4/Predicate.html)<? super T> predicate, [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<? super T,? extends T> trueTransformer) Create a new Transformer that calls the transformer if the predicate is true, otherwise the input object is returned unchanged.Type Parameters: T - the input / output type Parameters: predicate - the predicate to switch on trueTransformer - the transformer called if the predicate is true Returns: the transformer Throws: [NullPointerException](https://docs.oracle.com/javase/7/docs/api/java/lang/NullPointerException.html?is-external=true) - if either the predicate or transformer is null Since: 4.1 See Also: [IfTransformer](http://docs.google.com/org/apache/commons/collections4/functors/IfTransformer.html)

#### ifTransformer public static <I,O> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<I,O> ifTransformer([Predicate](http://docs.google.com/org/apache/commons/collections4/Predicate.html)<? super I> predicate, [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<? super I,? extends O> trueTransformer, [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<? super I,? extends O> falseTransformer) Create a new Transformer that calls one of two transformers depending on the specified predicate.Type Parameters: I - the input type O - the output type Parameters: predicate - the predicate to switch on trueTransformer - the transformer called if the predicate is true falseTransformer - the transformer called if the predicate is false Returns: the transformer Throws: [NullPointerException](https://docs.oracle.com/javase/7/docs/api/java/lang/NullPointerException.html?is-external=true) - if either the predicate or transformer is null Since: 4.1 See Also: [IfTransformer](http://docs.google.com/org/apache/commons/collections4/functors/IfTransformer.html)

#### switchTransformer [@Deprecated](https://docs.oracle.com/javase/7/docs/api/java/lang/Deprecated.html?is-external=true) public static <I,O> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<I,O> switchTransformer([Predicate](http://docs.google.com/org/apache/commons/collections4/Predicate.html)<? super I> predicate, [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<? super I,? extends O> trueTransformer, [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<? super I,? extends O> falseTransformer) Deprecated. as of 4.1, use [ifTransformer(Predicate, Transformer, Transformer)](http://docs.google.com/org/apache/commons/collections4/TransformerUtils.html#ifTransformer-org.apache.commons.collections4.Predicate-org.apache.commons.collections4.Transformer-org.apache.commons.collections4.Transformer-) Create a new Transformer that calls one of two transformers depending on the specified predicate.Type Parameters: I - the input type O - the output type Parameters: predicate - the predicate to switch on trueTransformer - the transformer called if the predicate is true falseTransformer - the transformer called if the predicate is false Returns: the transformer Throws: [NullPointerException](https://docs.oracle.com/javase/7/docs/api/java/lang/NullPointerException.html?is-external=true) - if either the predicate or transformer is null See Also: [SwitchTransformer](http://docs.google.com/org/apache/commons/collections4/functors/SwitchTransformer.html)

#### switchTransformer public static <I,O> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<I,O> switchTransformer([Predicate](http://docs.google.com/org/apache/commons/collections4/Predicate.html)<? super I>[] predicates, [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<? super I,? extends O>[] transformers) Create a new Transformer that calls one of the transformers depending on the predicates. The transformer at array location 0 is called if the predicate at array location 0 returned true. Each predicate is evaluated until one returns true. If no predicates evaluate to true, null is returned.Type Parameters: I - the input type O - the output type Parameters: predicates - an array of predicates to check transformers - an array of transformers to call Returns: the transformer Throws: [NullPointerException](https://docs.oracle.com/javase/7/docs/api/java/lang/NullPointerException.html?is-external=true) - if the either array is null [NullPointerException](https://docs.oracle.com/javase/7/docs/api/java/lang/NullPointerException.html?is-external=true) - if any element in the arrays is null [IllegalArgumentException](https://docs.oracle.com/javase/7/docs/api/java/lang/IllegalArgumentException.html?is-external=true) - if the arrays have different sizes See Also: [SwitchTransformer](http://docs.google.com/org/apache/commons/collections4/functors/SwitchTransformer.html)

#### switchTransformer public static <I,O> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<I,O> switchTransformer([Predicate](http://docs.google.com/org/apache/commons/collections4/Predicate.html)<? super I>[] predicates, [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<? super I,? extends O>[] transformers, [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<? super I,? extends O> defaultTransformer) Create a new Transformer that calls one of the transformers depending on the predicates. The transformer at array location 0 is called if the predicate at array location 0 returned true. Each predicate is evaluated until one returns true. If no predicates evaluate to true, the default transformer is called. If the default transformer is null, null is returned.Type Parameters: I - the input type O - the output type Parameters: predicates - an array of predicates to check transformers - an array of transformers to call defaultTransformer - the default to call if no predicate matches, null means return null Returns: the transformer Throws: [NullPointerException](https://docs.oracle.com/javase/7/docs/api/java/lang/NullPointerException.html?is-external=true) - if the either array is null [NullPointerException](https://docs.oracle.com/javase/7/docs/api/java/lang/NullPointerException.html?is-external=true) - if any element in the arrays is null [IllegalArgumentException](https://docs.oracle.com/javase/7/docs/api/java/lang/IllegalArgumentException.html?is-external=true) - if the arrays have different sizes See Also: [SwitchTransformer](http://docs.google.com/org/apache/commons/collections4/functors/SwitchTransformer.html)

#### switchTransformer public static <I,O> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<I,O> switchTransformer([Map](https://docs.oracle.com/javase/7/docs/api/java/util/Map.html?is-external=true)<[Predicate](http://docs.google.com/org/apache/commons/collections4/Predicate.html)<I>,[Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<I,O>> predicatesAndTransformers) Create a new Transformer that calls one of the transformers depending on the predicates. The Map consists of Predicate keys and Transformer values. A transformer is called if its matching predicate returns true. Each predicate is evaluated until one returns true. If no predicates evaluate to true, the default transformer is called. The default transformer is set in the map with a null key. If no default transformer is set, null will be returned in a default case. The ordering is that of the iterator() method on the entryset collection of the map.Type Parameters: I - the input type O - the output type Parameters: predicatesAndTransformers - a map of predicates to transformers Returns: the transformer Throws: [NullPointerException](https://docs.oracle.com/javase/7/docs/api/java/lang/NullPointerException.html?is-external=true) - if the map is null [NullPointerException](https://docs.oracle.com/javase/7/docs/api/java/lang/NullPointerException.html?is-external=true) - if any transformer in the map is null [ClassCastException](https://docs.oracle.com/javase/7/docs/api/java/lang/ClassCastException.html?is-external=true) - if the map elements are of the wrong type See Also: [SwitchTransformer](http://docs.google.com/org/apache/commons/collections4/functors/SwitchTransformer.html)

#### switchMapTransformer public static <I,O> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<I,O> switchMapTransformer([Map](https://docs.oracle.com/javase/7/docs/api/java/util/Map.html?is-external=true)<I,[Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<I,O>> objectsAndTransformers) Create a new Transformer that uses the input object as a key to find the transformer to call. The Map consists of object keys and Transformer values. A transformer is called if the input object equals the key. If there is no match, the default transformer is called. The default transformer is set in the map using a null key. If no default is set, null will be returned in a default case.Type Parameters: I - the input type O - the output type Parameters: objectsAndTransformers - a map of objects to transformers Returns: the transformer Throws: [NullPointerException](https://docs.oracle.com/javase/7/docs/api/java/lang/NullPointerException.html?is-external=true) - if the map is null [NullPointerException](https://docs.oracle.com/javase/7/docs/api/java/lang/NullPointerException.html?is-external=true) - if any transformer in the map is null See Also: [SwitchTransformer](http://docs.google.com/org/apache/commons/collections4/functors/SwitchTransformer.html)

#### instantiateTransformer public static <T> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<[Class](https://docs.oracle.com/javase/7/docs/api/java/lang/Class.html?is-external=true)<? extends T>,T> instantiateTransformer() Gets a Transformer that expects an input Class object that it will instantiate.Type Parameters: T - the output type Returns: the transformer See Also: [InstantiateTransformer](http://docs.google.com/org/apache/commons/collections4/functors/InstantiateTransformer.html)

#### instantiateTransformer public static <T> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<[Class](https://docs.oracle.com/javase/7/docs/api/java/lang/Class.html?is-external=true)<? extends T>,T> instantiateTransformer([Class](https://docs.oracle.com/javase/7/docs/api/java/lang/Class.html?is-external=true)<?>[] paramTypes, [Object](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true)[] args) Creates a Transformer that expects an input Class object that it will instantiate. The constructor used is determined by the arguments specified to this method.Type Parameters: T - the output type Parameters: paramTypes - parameter types for the constructor, can be null args - the arguments to pass to the constructor, can be null Returns: the transformer Throws: [IllegalArgumentException](https://docs.oracle.com/javase/7/docs/api/java/lang/IllegalArgumentException.html?is-external=true) - if the paramTypes and args don't match See Also: [InstantiateTransformer](http://docs.google.com/org/apache/commons/collections4/functors/InstantiateTransformer.html)

#### mapTransformer public static <I,O> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<I,O> mapTransformer([Map](https://docs.oracle.com/javase/7/docs/api/java/util/Map.html?is-external=true)<? super I,? extends O> map) Creates a Transformer that uses the passed in Map to transform the input object (as a simple lookup).Type Parameters: I - the input type O - the output type Parameters: map - the map to use to transform the objects Returns: the transformer, or [ConstantTransformer.nullTransformer()](http://docs.google.com/org/apache/commons/collections4/functors/ConstantTransformer.html#nullTransformer--) if the map is null See Also: [MapTransformer](http://docs.google.com/org/apache/commons/collections4/functors/MapTransformer.html)

#### invokerTransformer public static <I,O> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<I,O> invokerTransformer([String](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html?is-external=true) methodName) Gets a Transformer that invokes a method on the input object. The method must have no parameters. If the input object is null, null is returned. For example, TransformerUtils.invokerTransformer("getName"); will call the getName method on the input object to determine the transformer result.Type Parameters: I - the input type O - the output type Parameters: methodName - the method name to call on the input object, may not be null Returns: the transformer Throws: [NullPointerException](https://docs.oracle.com/javase/7/docs/api/java/lang/NullPointerException.html?is-external=true) - if the methodName is null. See Also: [InvokerTransformer](http://docs.google.com/org/apache/commons/collections4/functors/InvokerTransformer.html)

#### invokerTransformer public static <I,O> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<I,O> invokerTransformer([String](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html?is-external=true) methodName, [Class](https://docs.oracle.com/javase/7/docs/api/java/lang/Class.html?is-external=true)<?>[] paramTypes, [Object](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true)[] args) Gets a Transformer that invokes a method on the input object. The method parameters are specified. If the input object is null, null is returned.Type Parameters: I - the input type O - the output type Parameters: methodName - the name of the method paramTypes - the parameter types args - the arguments Returns: the transformer Throws: [NullPointerException](https://docs.oracle.com/javase/7/docs/api/java/lang/NullPointerException.html?is-external=true) - if the method name is null [IllegalArgumentException](https://docs.oracle.com/javase/7/docs/api/java/lang/IllegalArgumentException.html?is-external=true) - if the paramTypes and args don't match See Also: [InvokerTransformer](http://docs.google.com/org/apache/commons/collections4/functors/InvokerTransformer.html)

#### stringValueTransformer public static <T> [Transformer](http://docs.google.com/org/apache/commons/collections4/Transformer.html)<T,[String](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html?is-external=true)> stringValueTransformer() Gets a transformer that returns a java.lang.String representation of the input object. This is achieved via the toString method, null returns 'null'.Type Parameters: T - the input type Returns: the transformer See Also: [StringValueTransformer](http://docs.google.com/org/apache/commons/collections4/functors/StringValueTransformer.html)

[Skip navigation links](#3o7alnk)

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Use](http://docs.google.com/class-use/TransformerUtils.html)
* [Tree](http://docs.google.com/package-tree.html)
* [Deprecated](http://docs.google.com/deprecated-list.html)
* [Index](http://docs.google.com/index-all.html)
* [Help](http://docs.google.com/help-doc.html)
* [Prev Class](http://docs.google.com/org/apache/commons/collections4/Transformer.html)
* [Next Class](http://docs.google.com/org/apache/commons/collections4/Trie.html)
* [Frames](http://docs.google.com/index.html?org/apache/commons/collections4/TransformerUtils.html)
* [No Frames](http://docs.google.com/TransformerUtils.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* Constr |
* [Method](#3znysh7)
* Detail:
* Field |
* Constr |
* [Method](#tyjcwt)

Copyright © 2001–2019 [The Apache Software Foundation](https://www.apache.org/). All rights reserved.