A Dataset of API Directives

Martin Monperrus Michael Eichberg Elif Tekes Mira Mezini

December 9, 2011

This document complements the paper:

What Should Developers Be Aware Of? An Empirical Study on the Directives of API Documentation (Martin Monperrus, Michael Eichberg, Elif Tekes, Mira Mezini), In Empirical Software Engineering, Springer, volume Online Edition, 2011. (published online: Dec. 2, 2011)

It lists all directives that we identified as part of our empirical study. For each identified directive, we give its location in bold and the respective Javadoc snippet. This data is also available in XML format at `http://www.monperrus.net/martin/api-directives`. You can freely use this data as long as you cite the paper.

```
@ARTICLE{Monperrus2011a,
  author = {Martin Monperrus and Michael Eichberg and Elif Tekes and Mira Mezini},
  title = {What Should Developers Be Aware Of? An Empirical Study on the Directives of API Documentation},
  journal = {Empirical Software Engineering},
  year = {2011},
  volume = {Online Edition},
  comment = {published online: Dec. 2, 2011},
  doi = {10.1007/s1066401191864},
  ee = {http://www.springerlink.com/content/lk11x7x4v4h030j3/},
  publisher = {Springer},
  url = {http://www.monperrus.net/martin/An-Empirical-Study-On-the-Directives-of-API-Documentation.pdf},
  x-abbrv = {ESE}
}
```

Contents

1 Method Call Directive

1.1 Not Null Directive

CollatingIterator.addIterator(iterator) @param iterator the iterator to add to the collation, must not be null

AbstractLinkedMap.entryAfter(entry) @param entry the entry to query, must not be null

TransformedSortedSet.TransformedSortedSet(set, transformer) @param set the set to decorate, must not be null

AbstractBagDecorator.AbstractBagDecorator(bag) @param bag the bag to decorate, must not be null

UnmodifiableSortedSet.decorate(set) @param set the set to decorate, must not be null

TransformedCollection.decorate(coll, transformer) @param transformer the transformer to use for conversion, must not be null

UnmodifiableSet.decorate(set) @param set the set to decorate, must not be null

SynchronizedSortedSet.decorate(set) @param set the set to decorate, must not be null

AbstractSetDecorator.AbstractSetDecorator(set) @param set the set to decorate, must not be null

LazySortedMap.getLazySortedMap(map, factory) @param map the map to decorate, must not be null

Timestamp.Timestamp(timestamp, signerCertPath) @param timestamp is the timestamp's date and time. It must not be null

Timestamp.Timestamp(timestamp, signerCertPath) @param signerCertPath is the TSA's certificate path. It must not be null

SynchronizedBuffer.SynchronizedBuffer(buffer, lock) @param buffer the buffer to decorate, must not be null

UnmodifiableSortedMap.decorate(map) @param map the map to decorate, must not be null

AbstractReferenceMap.put(key, value) @param key the key to add, must not be null

AbstractReferenceMap.put(key, value) @param value the value to add, must not be null

DefaultKeyValue.DefaultKeyValue(entry) @param entry the entry to copy, must not be null

LoggingPermission.LoggingPermission(name, actions) @param actions Must be either null or the empty string.

AccessControlContext.AccessControlContext(context) Context must not be null

DefaultMapEntry.DefaultMapEntry(pair) @param pair the pair to copy, must not be null

FixedSizeSortedMap.decorate(map) @param map the map to decorate, must not be null

UnmodifiableList.UnmodifiableList(list) @param list the list to decorate, must not be null

AbstractSortedMapDecorator.AbstractSortedMapDecorator(map) @param map the map to decorate, must not be null

TransformedSortedSet.decorate(set, transformer) @param set the set to decorate, must not be null

ContentProposalAdapter.ContentProposalAdapter(control, controlContentAdapter, proposalProvider, keyStroke, autoActivationCharacters) May not be null.

ContentProposalAdapter.ContentProposalAdapter(control, controlContentAdapter, proposalProvider, keyStroke, autoActivationCharacters) May not be null

ListOrderedSet.decorate(set, list) @param list the list to decorate, must be empty and not null

DefaultedMap.decorate(map, factory) @param map the map to decorate, must not be null

Proxy.Proxy(type, sa) For instance, for types Http, and Socks, a SocketAddress must be provided.

(abstract) CallableStatement.setNull(parameterName, sqlType) Note: You must specify the parameter's SQL type.

UnmodifiableMapEntry.UnmodifiableMapEntry(pair) @param pair the pair to copy, must not be null

ExternalActionManager.CommandCallback.ExternalActionManager.CommandCallback(bindingManager, commandManager) @param bindingManager The binding manager which will provide the callback; must not be null.

ExternalActionManager.CommandCallback.ExternalActionManager.CommandCallback(bindingManager, commandManager) @param commandManager The command manager which will provide the callback; must not be null.

ExternalActionManager.CommandCallback.ExternalActionManager.CommandCallback(bindingManager, commandManager, activeChecker) @param bindingManager The binding manager which will provide the callback; must not be null.

ExternalActionManager.CommandCallback.ExternalActionManager.CommandCallback(bindingManager, commandManager, activeChecker) @param commandManager The command manager which will provide the callback; must not be null.

ExternalActionManager.CommandCallback.ExternalActionManager.CommandCallback(bindingManager, commandManager, activeChecker) @param activeChecker The callback mechanism for checking whether a command is active; must not be null

ExternalActionManager.CommandCallback.ExternalActionManager.CommandCallback(bindingManager, commandManager, activeChecker, checker) @param commandManager The command manager which will provide the callback; must not be null.

ExternalActionManager.CommandCallback.ExternalActionManager.CommandCallback(bindingManager, commandManager, activeChecker, checker) @param activeChecker The callback mechanism for checking whether a command is active; must not be null

(abstract) IContentAssistSubjectControl.removeSelectionListener(selectionListener) @exception IllegalArgumentException if listener is null

(abstract) IContentAssistSubjectControl.addSelectionListener(selectionListener) @exception IllegalArgumentException if listener is null

SynchronizedSortedBag.SynchronizedSortedBag(bag, lock) @param bag the bag to decorate, must not be null

Binding.toString() @return The string representation; never null.

SynchronizedBag.SynchronizedBag(bag, lock) @param lock the lock to use, must not be null

SynchronizedBag.SynchronizedBag(bag, lock) @param bag the bag to decorate, must not be nul

SynchronizedBag.SynchronizedBag(bag, lock) @throws IllegalArgumentException if bag is null

SynchronizedCollection.SynchronizedCollection(collection, lock) @param collection the collection to decorate, must not be null

SynchronizedCollection.SynchronizedCollection(collection, lock) @param lock the lock object to use, must not be null

SynchronizedCollection.SynchronizedCollection(collection, lock) @throws IllegalArgumentException if the collection is null

SynchronizedBuffer.SynchronizedBuffer(buffer, lock) @param lock the lock object to use, must not be null

SynchronizedBuffer.SynchronizedBuffer(buffer, lock) @throws IllegalArgumentException if the buffer is null

SynchronizedList.SynchronizedList(list, lock) @param list the list to decorate, must not be null

SynchronizedList.SynchronizedList(list, lock) @param lock the lock to use, must not be null

SynchronizedList.SynchronizedList(list, lock) @throws IllegalArgumentException if list is null

LazyList.decorate(list, factory) @param list the list to decorate, must not be null

LazyList.decorate(list, factory) @param factory the factory to use for creation, must not be null

LazyList.decorate(list, factory) @throws IllegalArgumentException if list or factory is null

LazyList.LazyList(list, factory) @param list the list to decorate, must not be null

LazyList.LazyList(list, factory) @param factory the factory to use for creation, must not be null

LazyList.LazyList(list, factory) @throws IllegalArgumentException if list or factory is null

LazyMap.getLazyMap(map, factory) @param factory the factory to use, must not be null

LazyMap.getLazyMap(map, factory) @throws IllegalArgumentException if map or factory is null

LazyMap.getLazyMap(map, factory) @param map the map to decorate, must not be null

LazyMap.getLazyMap(map, factory) @param map the map to decorate, must not be null

LazyMap.getLazyMap(map, factory) @param factory the factory to use, must not be null

LazyMap.getLazyMap(map, factory) @throws IllegalArgumentException if map or factory is null

LazyMap.LazyMap(map, factory) @param map the map to decorate, must not be null

LazyMap.LazyMap(map, factory) @param factory the factory to use, must not be null

LazyMap.LazyMap(map, factory) @throws IllegalArgumentException if map or factory is null

LazyMap.LazyMap(map, factory) @param factory the factory to use, must not be null

LazyMap.LazyMap(map, factory) @param map the map to decorate, must not be null

LazyMap.LazyMap(map, factory) @throws IllegalArgumentException if map or factory is null

LazySortedMap.decorate(map, factory) @param map the map to decorate, must not be null

LazySortedMap.decorate(map, factory) @param factory the factory to use, must not be null

LazySortedMap.decorate(map, factory) @throws IllegalArgumentException if map or factory is null

LazySortedMap.getLazySortedMap(map, factory) @param map the map to decorate, must not be null

LazySortedMap.getLazySortedMap(map, factory) @param factory the factory to use, must not be null

LazySortedMap.getLazySortedMap(map, factory) @throws IllegalArgumentException if map or factory is null

LazySortedMap.decorate(map, factory) @param map the map to decorate, must not be null

LazySortedMap.decorate(map, factory) @param factory the factory to use, must not be null

LazySortedMap.decorate(map, factory) @throws IllegalArgumentException if map or factory is null

LazySortedMap.getLazySortedMap(map, factory) @param factory the factory to use, must not be null

LazySortedMap.getLazySortedMap(map, factory) @throws IllegalArgumentException if map or factory is null

LazySortedMap.LazySortedMap(map, factory) @param map the map to decorate, must not be null

LazySortedMap.LazySortedMap(map, factory) @param factory the factory to use, must not be null

LazySortedMap.LazySortedMap(map, factory) @throws IllegalArgumentException if map or factory is null

LazySortedMap.LazySortedMap(map, factory) @param map the map to decorate, must not be null

LazySortedMap.LazySortedMap(map, factory) @param factory the factory to use, must not be null

LazySortedMap.LazySortedMap(map, factory) @throws IllegalArgumentException if map or factory is null

SynchronizedSortedSet.SynchronizedSortedSet(set, lock) @param set the set to decorate, must not be null

SynchronizedSortedSet.SynchronizedSortedSet(set, lock) @param lock the lock object to use, must not be null

SynchronizedSortedSet.SynchronizedSortedSet(set, lock) @throws IllegalArgumentException if set is null

SynchronizedSet.SynchronizedSet(set, lock) @param set the set to decorate, must not be null

SynchronizedSet.SynchronizedSet(set, lock) @param lock the lock object to use, must not be null

SynchronizedSet.SynchronizedSet(set, lock) @throws IllegalArgumentException if set is null

SynchronizedSortedBag.SynchronizedSortedBag(bag) @param bag the bag to decorate, must not be null

SynchronizedSortedBag.SynchronizedSortedBag(bag) @throws IllegalArgumentException if bag is null

SynchronizedBag.decorate(bag) @param bag the bag to decorate, must not be null

SynchronizedBag.decorate(bag) @throws IllegalArgumentException if bag is null

SynchronizedBag.SynchronizedBag(bag) @param bag the bag to decorate, must not be null

SynchronizedBag.SynchronizedBag(bag) @throws IllegalArgumentException if bag is null

AbstractSortedBagDecorator.AbstractSortedBagDecorator(bag) @param bag the bag to decorate, must not be null

AbstractSortedBagDecorator.AbstractSortedBagDecorator(bag) @throws IllegalArgumentException if list is null

AbstractBagDecorator.AbstractBagDecorator(bag) @throws IllegalArgumentException if list is null

SynchronizedCollection.SynchronizedCollection(collection) @param collection the collection to decorate, must not be null

SynchronizedCollection.SynchronizedCollection(collection) @throws IllegalArgumentException if the collection is null

AbstractCollectionDecorator.AbstractCollectionDecorator(coll) @param coll the collection to decorate, must not be null

AbstractCollectionDecorator.AbstractCollectionDecorator(coll) @throws IllegalArgumentException if the collection is null

SynchronizedBuffer.SynchronizedBuffer(buffer) @param buffer the buffer to decorate, must not be null

SynchronizedBuffer.SynchronizedBuffer(buffer) @throws IllegalArgumentException if the buffer is null

BlockingBuffer.BlockingBuffer(buffer) @param buffer the buffer to decorate, must not be null

BlockingBuffer.BlockingBuffer(buffer) @throws IllegalArgumentException if the buffer is null

AbstractBufferDecorator.AbstractBufferDecorator(buffer) @param buffer the buffer to decorate, must not be null

AbstractBufferDecorator.AbstractBufferDecorator(buffer) @throws IllegalArgumentException if list is null

SynchronizedList.decorate(list) @param list the list to decorate, must not be null

SynchronizedList.decorate(list) @throws IllegalArgumentException if list is null

SynchronizedList.SynchronizedList(list) @param list the list to decorate, must not be null

SynchronizedList.SynchronizedList(list) @throws IllegalArgumentException if list is null

UnmodifiableList.decorate(list) @param list the list to decorate, must not be null

UnmodifiableList.decorate(list) @throws IllegalArgumentException if list is null

UnmodifiableList.UnmodifiableList(list) @throws IllegalArgumentException if list is null

GrowthList.decorate(list) @param list the list to decorate, must not be null

GrowthList.decorate(list) @throws IllegalArgumentException if list is null

GrowthList.GrowthList(list) @param list the list to decorate, must not be null

GrowthList.GrowthList(list) @throws IllegalArgumentException if list is null

FixedSizeList.decorate(list) @param list the list to decorate, must not be null

FixedSizeList.decorate(list) throws IllegalArgumentException if list is null

FixedSizeList.FixedSizeList(list) @param list the list to decorate, must not be null

FixedSizeList.FixedSizeList(list) @throws IllegalArgumentException if list is null

AbstractListDecorator.AbstractListDecorator(list) @param list the list to decorate, must not be null

AbstractListDecorator.AbstractListDecorator(list) @throws IllegalArgumentException if list is null

AbstractIteratorDecorator.AbstractIteratorDecorator(iterator) @param iterator the iterator to decorate, must not be null

AbstractIteratorDecorator.AbstractIteratorDecorator(iterator) @throws IllegalArgumentException if the collection is null

AbstractMapIteratorDecorator.AbstractMapIteratorDecorator(iterator) @param iterator the iterator to decorate, must not be null

AbstractMapIteratorDecorator.AbstractMapIteratorDecorator(iterator) @throws IllegalArgumentException if the collection is null

AbstractListIteratorDecorator.AbstractListIteratorDecorator(iterator) @param iterator the iterator to decorate, must not be null

AbstractListIteratorDecorator.AbstractListIteratorDecorator(iterator) @throws IllegalArgumentException if the collection is null

AbstractOrderedMapIteratorDecorator.AbstractOrderedMapIteratorDecorator(iterator) @param iterator the iterator to decorate, must not be null

AbstractOrderedMapIteratorDecorator.AbstractOrderedMapIteratorDecorator(iterator) @throws IllegalArgumentException if the collection is null

UnmodifiableMap.decorate(map) @param map the map to decorate, must not be null

UnmodifiableMap.decorate(map) @throws IllegalArgumentException if map is null

UnmodifiableEntrySet.decorate(set) @param set the set to decorate, must not be null

UnmodifiableEntrySet.decorate(set) @throws IllegalArgumentException if set is null

FixedSizeMap.decorate(map) @param map the map to decorate, must not be null

FixedSizeMap.decorate(map) @throws IllegalArgumentException if map is null

FixedSizeMap.FixedSizeMap(map) @param map the map to decorate, must not be null

FixedSizeMap.FixedSizeMap(map) @throws IllegalArgumentException if map is null

UnmodifiableSortedMap.decorate(map) @throws IllegalArgumentException if map is null

FixedSizeSortedMap.decorate(map) @throws IllegalArgumentException if map is null

FixedSizeSortedMap.FixedSizeSortedMap(map) @param map the map to decorate, must not be null

FixedSizeSortedMap.FixedSizeSortedMap(map) @throws IllegalArgumentException if map is null

AbstractMapDecorator.AbstractMapDecorator(map) @param map the map to decorate, must not be null

AbstractMapDecorator.AbstractMapDecorator(map) @throws IllegalArgumentException if the collection is null

UnmodifiableOrderedMap.decorate(map) @param map the map to decorate, must not be null

UnmodifiableOrderedMap.decorate(map) @throws IllegalArgumentException if map is null

ListOrderedMap.ListOrderedMap(map) @param map the map to decorate, must not be null

ListOrderedMap.ListOrderedMap(map) @throws IllegalArgumentException if map is null

AbstractOrderedMapDecorator.AbstractOrderedMapDecorator(map) @param map the map to decorate, must not be null

AbstractOrderedMapDecorator.AbstractOrderedMapDecorator(map) @throws IllegalArgumentException if the collection is null

UnmodifiableMapEntry.UnmodifiableMapEntry(pair) @throws NullPointerException if the entry is null

UnmodifiableMapEntry.UnmodifiableMapEntry(entry) @param entry the entry to copy, must not be null

UnmodifiableMapEntry.UnmodifiableMapEntry(entry) @throws NullPointerException if the entry is null

DefaultKeyValue.DefaultKeyValue(pair) @param pair the pair to copy, must not be null

DefaultKeyValue.DefaultKeyValue(pair) @throws NullPointerException if the entry is null

DefaultKeyValue.DefaultKeyValue(entry) @throws NullPointerException if the entry is null

AbstractMapEntryDecorator.AbstractMapEntryDecorator(entry) @param entry the Map.Entry to decorate, must not be null

AbstractMapEntryDecorator.AbstractMapEntryDecorator(entry) @throws IllegalArgumentException if the collection is null

DefaultMapEntry.DefaultMapEntry(pair) @throws NullPointerException if the entry is null

DefaultMapEntry.DefaultMapEntry(entry) @param entry the entry to copy, must not be null

DefaultMapEntry.DefaultMapEntry(entry) @throws NullPointerException if the entry is null

SynchronizedSortedSet.decorate(set) @throws IllegalArgumentException if set is null

SynchronizedSortedSet.SynchronizedSortedSet(set) @param set the set to decorate, must not be null

SynchronizedSortedSet.SynchronizedSortedSet(set) @throws IllegalArgumentException if set is null

ListOrderedSet.ListOrderedSet(set) @param set the set to decorate, must not be null

ListOrderedSet.ListOrderedSet(set) @throws IllegalArgumentException if set is null

UnmodifiableSortedSet.decorate(set) @throws IllegalArgumentException if set is null

SynchronizedSet.SynchronizedSet(set) @param set the set to decorate, must not be null

SynchronizedSet.decorate(set) @throws IllegalArgumentException if set is null

SynchronizedSet.decorate(set) @param set the set to decorate, must not be null
SynchronizedSet.SynchronizedSet(set) @throws IllegalArgumentException if set is null
UnmodifiableSet.decorate(set) @throws IllegalArgumentException if set is null
MapBackedSet.decorate(map) @param map the map to decorate, must not be null
MapBackedSet.decorate(map) @throws IllegalArgumentException if set is null
AbstractSortedSetDecorator.AbstractSortedSetDecorator(set) @param set the set to decorate, must not be null
AbstractSortedSetDecorator.AbstractSortedSetDecorator(set) @throws IllegalArgumentException if set is null
AbstractBidiMapDecorator.AbstractBidiMapDecorator(map) @param map the map to decorate, must not be null
AbstractSortedBidiMapDecorator.AbstractSortedBidiMapDecorator(map) @param map the map to decorate, must not be null
AbstractSortedBidiMapDecorator.AbstractSortedBidiMapDecorator(map) @throws IllegalArgumentException if the collection is null
AbstractBidiMapDecorator.AbstractBidiMapDecorator(map) @throws IllegalArgumentException if the collection is null
AbstractOrderedBidiMapDecorator.AbstractOrderedBidiMapDecorator(map) @param map the map to decorate, must not be null
AbstractOrderedBidiMapDecorator.AbstractOrderedBidiMapDecorator(map) @throws IllegalArgumentException if the collection is null
TransformedBag.decorate(bag, transformer) @param transformer the transformer to use for conversion, must not be null
TransformedBag.decorate(bag, transformer) @throws IllegalArgumentException if bag or transformer is null
TransformedBag.decorate(bag, transformer) @param bag the bag to decorate, must not be null
TransformedBag.TransformedBag(bag, transformer) @param bag the bag to decorate, must not be null
TransformedBag.TransformedBag(bag, transformer) @param transformer the transformer to use for conversion, must not be null
TransformedBag.TransformedBag(bag, transformer) @throws IllegalArgumentException if bag or transformer is null
PredicatedBag.PredicatedBag(bag, predicate) @param bag the bag to decorate, must not be null
PredicatedBag.PredicatedBag(bag, predicate) @param predicate the predicate to use for validation, must not be null
PredicatedBag.PredicatedBag(bag, predicate) @throws IllegalArgumentException if bag or predicate is null
TransformedSortedBag.decorate(bag, transformer) @param transformer the transformer to use for conversion, must not be null
TransformedSortedBag.decorate(bag, transformer) @param bag the bag to decorate, must not be null
TransformedSortedBag.decorate(bag, transformer) @throws IllegalArgumentException if bag or transformer is null
URI.create(str) @throws NullPointerException If str is null
ActivationDesc.ActivationDesc(groupID, className, location, data, restart) @exception IllegalArgumentException if groupID is null
ReflectPermission.ReflectPermission(name, actions) @throws NullPointerException if name is null.
RuntimePermission.RuntimePermission(name, actions) @throws NullPointerException if name is null.
(abstract) PreparedStatement.setNull(parameterIndex, sqlType) Note: You must specify the parameter's SQL type.
CodeSigner.CodeSigner(signerCertPath, timestamp) @param signerCertPath The signer's certificate path. It must not be null.
CaseInsensitiveMap.CaseInsensitiveMap(map) @throws NullPointerException if the map is null
PredicateUtils.andPredicate(predicate1, predicate2) @param predicate1 the first predicate, may not be null
PredicateUtils.andPredicate(predicate1, predicate2) @param predicate2 the second predicate, may not be null
PredicateUtils.andPredicate(predicate1, predicate2) @throws IllegalArgumentException if either predicate is null
IteratorUtils.filteredIterator(iterator, predicate) @param predicate the predicate to use as a filter, not null
IteratorUtils.filteredIterator(iterator, predicate) @param iterator the iterator to use, not null
IteratorUtils.filteredIterator(iterator, predicate) @throws NullPointerException if either parameter is null
IteratorUtils.filteredListIterator(listIterator, predicate) @param listIterator the list iterator to use, not null
IteratorUtils.filteredListIterator(listIterator, predicate) @throws NullPointerException if either parameter is null
IteratorUtils.filteredListIterator(listIterator, predicate) @param predicate the predicate to use as a filter, not null
IteratorUtils.loopingIterator(coll) @throws NullPointerException if the collection is null
IteratorUtils.loopingIterator(coll) @param coll the collection to iterate over, not null
IteratorUtils.loopingListIterator(list) @param list the list to iterate over, not null
IteratorUtils.loopingListIterator(list) @throws NullPointerException if the list is null
MultiKey.MultiKey(keys, makeClone) @throws IllegalArgumentException if the key array is null
MultiKey.MultiKey(keys, makeClone) @param keys the array of keys, not null
BoundedBuffer.BoundedBuffer(buffer, maximumSize, timeout) @param buffer the buffer to wrap, must not be null
SplitMapUtils.writableMap(put) @param put to wrap, must not be null
ListUtils.subtract(list1, list2) throws NullPointerException if either list is null
PredicateUtils.allPredicate(predicates) @throws IllegalArgumentException if the predicates array is null
PredicateUtils.allPredicate(predicates) @throws IllegalArgumentException if any predicate in the array is null
PredicateUtils.allPredicate(predicates) @param predicates an array of predicates to check, may not be null
PredicateUtils.allPredicate(predicates) @param predicates a collection of predicates to check, may not be null
PredicateUtils.allPredicate(predicates) @throws IllegalArgumentException if the predicates collection is null
PredicateUtils.allPredicate(predicates) @throws IllegalArgumentException if any predicate in the collection is null
PredicateUtils.onePredicate(predicates) @throws IllegalArgumentException if the predicates array is nul
PredicateUtils.onePredicate(predicates) @throws IllegalArgumentException if any predicate in the array is null
PredicateUtils.onePredicate(predicates) @param predicates an array of predicates to check, may not be null
PredicateUtils.onePredicate(predicates) @param predicates a collection of predicates to check, may not be null
PredicateUtils.onePredicate(predicates) @throws IllegalArgumentException if the predicates collection is null
PredicateUtils.onePredicate(predicates) @throws IllegalArgumentException if any predicate in the collection is null
CollectionUtils.getCardinalityMap(coll) @param coll the collection to get the cardinality map for, must not be null
ListUtils.predicatedList(list, predicate) @param list the list to predicate, must not be null
ListUtils.predicatedList(list, predicate) @throws IllegalArgumentException if the List or Predicate is null
Float.valueOf(s) If s is null, then a NullPointerException is thrown.
CollectionUtils.isProperSubCollection(a, b) @param a the first (sub?) collection, must not be null
MapUtils.safeAddToMap(map, key, value) @param map the map to add to, may not be null
MapUtils.safeAddToMap(map, key, value) @throws NullPointerException if the map is null
class ReferenceIdentityMap Attempting to add a null key or value to the map will raise a NullPointerException.
class ReferenceMap Attempting to add a null key or value to the map will raise a NullPointerException.
TransformedSortedBag.TransformedSortedBag(bag, transformer) @param bag the bag to decorate, must not be null
TransformedSortedBag.TransformedSortedBag(bag, transformer) @param transformer the transformer to use for conversion, must not be null
PredicatedSortedBag.PredicatedSortedBag(bag, predicate) @param bag the bag to decorate, must not be null
PredicatedSortedBag.PredicatedSortedBag(bag, predicate) @param predicate the predicate to use for validation, must not be null
TransformedCollection.TransformedCollection(coll, transformer) @param coll the collection to decorate, must not be null
TransformedCollection.TransformedCollection(coll, transformer) @param transformer the transformer to use for conversion, must not be null
PredicatedCollection.PredicatedCollection(coll, predicate) @param coll the collection to decorate, must not be null
PredicatedCollection.PredicatedCollection(coll, predicate) @param predicate the predicate to use for validation, must not be null
TransformedBuffer.decorate(buffer, transformer) @param buffer the buffer to decorate, must not be null
TransformedBuffer.decorate(buffer, transformer) @param transformer the transformer to use for conversion, must not be null
TransformedBuffer.TransformedBuffer(buffer, transformer) @param buffer the buffer to decorate, must not be null
TransformedBuffer.TransformedBuffer(buffer, transformer) @param transformer the transformer to use for conversion, must not be null
PredicatedBuffer.PredicatedBuffer(buffer, predicate) @param buffer the buffer to decorate, must not be null
PredicatedBuffer.PredicatedBuffer(buffer, predicate) @param predicate the predicate to use for validation, must not be null
SetUniqueList.SetUniqueList(list, set) @param set the set to decorate, must not be null
SetUniqueList.SetUniqueList(list, set) @param list the list to decorate, must not be null
TransformedList.decorate(list, transformer) @param list the list to decorate, must not be null
TransformedList.decorate(list, transformer) @param transformer the transformer to use for conversion, must not be null
TransformedList.TransformedList(list, transformer) @param list the list to decorate, must not be null
TransformedList.TransformedList(list, transformer) @param transformer the transformer to use for conversion, must not be null
PredicatedList.PredicatedList(list, predicate) @param list the list to decorate, must not be null
PredicatedList.PredicatedList(list, predicate) @param predicate the predicate to use for validation, must not be null
LazyMap.decorate(map, factory) @param map the map to decorate, must not be null

LazyMap.decorate(map, factory) @param factory the factory to use, must not be null
LazyMap.decorate(map, factory) @param map the map to decorate, must not be null
LazyMap.decorate(map, factory) @param factory the factory to use, must not be null
ListOrderedSet.ListOrderedSet(set, list) @param set the set to decorate, must not be null
ListOrderedSet.ListOrderedSet(set, list) @param list the list to decorate, must not be null
PredicatedSortedSet.PredicatedSortedSet(set, predicate) @param set the set to decorate, must not be null
PredicatedSortedSet.PredicatedSortedSet(set, predicate) @param predicate the predicate to use for validation, must not be null
TransformedSet.decorate(set, transformer) @param set the set to decorate, must not be null
TransformedSet.decorate(set, transformer) @param transformer the transformer to use for conversion, must not be null
TransformedSet.TransformedSet(set, transformer) @param set the set to decorate, must not be null
TransformedSet.TransformedSet(set, transformer) @param transformer the transformer to use for conversion, must not be null
PredicatedSet.PredicatedSet(set, predicate) @param set the set to decorate, must not be null
PredicatedSet.PredicatedSet(set, predicate) @param predicate the predicate to use for validation, must not be null
CollectionUtils.containsAny(coll1, coll2) @param coll1 the first collection, must not be null
CollectionUtils.containsAny(coll1, coll2) @param coll2 the first collection, must not be null
CollectionUtils.addAll(collection, iterator) @param collection the collection to add to, must not be null
CollectionUtils.addAll(collection, iterator) @param iterator the iterator of elements to add, must not be null
CollectionUtils.addAll(collection, enumeration) @param collection the collection to add to, must not be null
CollectionUtils.addAll(collection, enumeration) @param enumeration the enumeration of elements to add, must not be null
CollectionUtils.addAll(collection, elements) @param collection the collection to add to, must not be null
CollectionUtils.addAll(collection, elements) @param elements the array of elements to add, must not be null
SynchronizedSortedBag.decorate(bag) @param bag the bag to decorate, must not be null
UnmodifiableSortedBag.decorate(bag) @param bag the bag to decorate, must not be null
UnmodifiableBag.decorate(bag) @param bag the bag to decorate, must not be null
UnmodifiableBoundedCollection.decorate(coll) @param coll the BoundedCollection to decorate, must not be null
SynchronizedCollection.decorate(coll) @param coll the collection to decorate, must not be null
SynchronizedBuffer.decorate(buffer) @param buffer the buffer to decorate, must not be null
UnmodifiableBuffer.decorate(buffer) @param buffer the buffer to decorate, must not be null
BlockingBuffer.decorate(buffer) @param buffer the buffer to decorate, must not be null
BlockingBuffer.BlockingBuffer(buffer, timeoutMillis) @param buffer the buffer to decorate, must not be null
DefaultedMap.DefaultedMap(map, defaultValueTransformer) @param map the map to decorate, must not be null
MultiKeyMap.decorate(map) @param map the map to decorate, not null
ListOrderedMap.decorate(map) @param map the map to decorate, must not be null
ListOrderedSet.decorate(set) @param set the set to decorate, must not be null
MapBackedSet.decorate(map, dummyValue) @param map the map to decorate, must not be null
UnmodifiableOrderedBidiMap.decorate(map) @param map the map to decorate, must not be null
UnmodifiableSortedBidiMap.decorate(map) @param map the map to decorate, must not be null
UnmodifiableBidiMap.decorate(map) @param map the map to decorate, must not be null
BufferUtils.unmodifiableBuffer(buffer) @param buffer the buffer to make unmodifiable, must not be null
TransformedBag.decorateTransform(bag, transformer) @param bag the bag to decorate, must not be null
TransformedBag.decorateTransform(bag, transformer) @param transformer the transformer to use for conversion, must not be null
PredicatedBag.decorate(bag, predicate) @param bag the bag to decorate, must not be null
PredicatedBag.decorate(bag, predicate) @param predicate the predicate to use for validation, must not be null
TransformedSortedBag.decorateTransform(bag, transformer) @param bag the bag to decorate, must not be null
TransformedSortedBag.decorateTransform(bag, transformer) @param transformer the transformer to use for conversion, must not be null
PredicatedSortedBag.decorate(bag, predicate) @param bag the bag to decorate, must not be null
PredicatedSortedBag.decorate(bag, predicate) @param predicate the predicate to use for validation, must not be null
TransformedCollection.decorateTransform(collection, transformer) @param collection the collection to decorate, must not be null
TransformedCollection.decorateTransform(collection, transformer) @param transformer the transformer to use for conversion, must not be null
PredicatedCollection.decorate(coll, predicate) @param coll the collection to decorate, must not be null
PredicatedCollection.decorate(coll, predicate) @param predicate the predicate to use for validation, must not be null
TransformedBuffer.decorateTransform(buffer, transformer) @param buffer the buffer to decorate, must not be null
TransformedBuffer.decorateTransform(buffer, transformer) @param transformer the transformer to use for conversion, must not be null
PredicatedBuffer.decorate(buffer, predicate) @param buffer the buffer to decorate, must not be null
PredicatedBuffer.decorate(buffer, predicate) @param predicate the predicate to use for validation, must not be null
TransformedList.decorateTransform(list, transformer) @param list the list to decorate, must not be null
TransformedList.decorateTransform(list, transformer) @param transformer the transformer to use for conversion, must not be null
PredicatedList.decorate(list, predicate) @param list the list to decorate, must not be null
PredicatedList.decorate(list, predicate) @param predicate the predicate to use for validation, must not be null
DefaultedMap.decorate(map, transformer) @param map the map to decorate, must not be null
DefaultedMap.decorate(map, transformer) @param transformer the transformer to use as a factory to create entries, must not be null
TransformedSortedSet.decorateTransform(set, transformer) @param set the set to decorate, must not be null
TransformedSortedSet.decorateTransform(set, transformer) @param transformer the transformer to use for conversion, must not be null
PredicatedSortedSet.decorate(set, predicate) @param set the set to decorate, must not be null
PredicatedSortedSet.decorate(set, predicate) @param predicate the predicate to use for validation, must not be null
TransformedSet.decorateTransform(set, transformer) @param set the set to decorate, must not be null
TransformedSet.decorateTransform(set, transformer) @param transformer the transformer to use for conversion, must not be null
PredicatedSet.decorate(set, predicate) @param set the set to decorate, must not be null
PredicatedSet.decorate(set, predicate) @param predicate the predicate to use for validation, must not be null
CollectionUtils.isSubCollection(a, b) @param b the second (super?) collection, must not be null
CollectionUtils.isSubCollection(a, b) @param a the first (sub?) collection, must not be null
CollectionUtils.isEqualCollection(a, b) @param a the first collection, must not be null
CollectionUtils.isEqualCollection(a, b) @param b the second collection, must not be null
CollectionUtils.addAll(collection, iterable) @param collection the collection to add to, must not be null
CollectionUtils.addAll(collection, iterable) @param iterable the iterable of elements to add, must not be null
BufferUtils.predicatedBuffer(buffer, predicate) @param buffer the buffer to predicate, must not be null
BufferUtils.predicatedBuffer(buffer, predicate) @param predicate the predicate used to evaluate new elements, must not be null
BufferUtils.transformedBuffer(buffer, transformer) @param buffer the buffer to predicate, must not be null
BufferUtils.transformedBuffer(buffer, transformer) @param transformer the transformer for the buffer, must not be null
CollectionUtils.union(a, b) @param a the first collection, must not be null
CollectionUtils.union(a, b) @param b the second collection, must not be null
CollectionUtils.intersection(a, b) @param a the first collection, must not be null
CollectionUtils.intersection(a, b) @param b the second collection, must not be null
CollectionUtils.subtract(a, b) @param a the collection to subtract from, must not be null
CollectionUtils.subtract(a, b) @param b the collection to subtract, must not be null
CollectionUtils.predicatedCollection(collection, predicate) @param collection the collection to predicate, must not be null
CollectionUtils.predicatedCollection(collection, predicate) @param predicate the predicate for the collection, must not be null
CollectionUtils.transformedCollection(collection, transformer) @param collection the collection to predicate, must not be null
CollectionUtils.transformedCollection(collection, transformer) @param transformer the transformer for the collection, must not be null
BagUtils.predicatedBag(bag, predicate) @param bag the bag to predicate, must not be null
BagUtils.predicatedBag(bag, predicate) @param predicate the predicate for the bag, must not be null
NullIsExceptionPredicate.evaluate(object) @throws FunctorException if input is null
WhileClosure.getInstance(predicate, closure, doLoop) @param closure the closure the execute, not null
WhileClosure.getInstance(predicate, closure, doLoop) @throws IllegalArgumentException if the predicate or closure is null

WhileClosure.getInstance(predicate, closure, doLoop) @param predicate the predicate used to evaluate when the loop terminates, not null

WhileClosure.WhileClosure(predicate, closure, doLoop) @param predicate the predicate used to evaluate when the loop terminates, not null

WhileClosure.WhileClosure(predicate, closure, doLoop) @param closure the closure the execute, not null

ClosureUtils.doWhileClosure(closure, predicate) @param closure the closure to call repeatedly, not null

ClosureUtils.doWhileClosure(closure, predicate) @param predicate the predicate to use as an end of loop test, not null

ClosureUtils.doWhileClosure(closure, predicate) @throws IllegalArgumentException if either argument is null

TreeBidiMap.put(key, value) @throws NullPointerException if the key is null

ListUtils.retainAll(collection, retain) @throws NullPointerException if either parameter is null

CollectionUtils.retainAll(collection, retain) @throws NullPointerException if either parameter is null

CompositeMap.putAll(map) @throws NullPointerException the specified map is null,

CompositeMap.putAll(map) or if this map does not permit null keys or values, and the specified map contains null keys or values.

CompositeMap.remove(key) @throws NullPointerException if the key is null and the composited map does not not permit null keys (optional).

(abstract) BidiMap.put(key, value) @throws NullPointerException (optional) if the map limits the values to non-null and null was specified

SynchronizedSortedBag.SynchronizedSortedBag(bag, lock) @param lock the lock to use, must not be null

SynchronizedSortedBag.SynchronizedSortedBag(bag, lock) @throws IllegalArgumentException if bag is null

MapUtils.transformedMap(map, keyTransformer, valueTransformer) @param map the map to transform, must not be null, typically empty

MapUtils.transformedMap(map, keyTransformer, valueTransformer) @throws IllegalArgumentException if the Map is null

MapUtils.transformedSortedMap(map, keyTransformer, valueTransformer) @param map the map to transform, must not be null, typically empty

MapUtils.transformedSortedMap(map, keyTransformer, valueTransformer) @throws IllegalArgumentException if the SortedMap is null

(abstract) MultiMap.remove(key, item) @throws NullPointerException if the key or value is null and null is invalid

(abstract) MultiMap.get(key) @throws NullPointerException if the key is null and null keys are invalid

(abstract) MultiMap.put(key, value) @throws NullPointerException if the key or value is null and null is invalid

(abstract) MultiMap.remove(key) @throws NullPointerException if the key is null and null keys are invalid

MapUtils.putAll(map, array) @param map the map to populate, must not be null

MapUtils.putAll(map, array) @throws NullPointerException if map is null

MapUtils.invertMap(map) @throws NullPointerException if the map is null

MapUtils.invertMap(map) @param map the map to invert, may not be null

AbstractHashedMap.isEqualValue(value1, value2) This implementation uses the equals method and assumes neither value is null.

AbstractHashedMap.isEqualKey(key1, key2) This implementation uses the equals method and assumes neither key is null.

EnumerationUtils.toList(enumeration) @param enumeration the enumeration to traverse, which should not be null.

EnumerationUtils.toList(enumeration) @throws NullPointerException if the enumeration parameter is null.

BagUtils.synchronizedBag(bag) @param bag the bag to synchronize, must not be null

BagUtils.synchronizedBag(bag) @throws IllegalArgumentException if the Bag is null

BagUtils.synchronizedSortedBag(bag) @throws IllegalArgumentException if the SortedBag is null

BagUtils.synchronizedSortedBag(bag) @param bag the bag to synchronize, must not be null

SynchronizedSortedBag.decorate(bag) @throws IllegalArgumentException if bag is null

SynchronizedCollection.decorate(coll) @throws IllegalArgumentException if collection is null

SynchronizedBuffer.decorate(buffer) @throws IllegalArgumentException if buffer is null

MapUtils.synchronizedMap(map) . @param map the map to synchronize, must not be null

MapUtils.synchronizedMap(map) @throws IllegalArgumentException if the map is null

MapUtils.lazyMap(map, transformerFactory) @param map the map to make lazy, must not be null

MapUtils.lazyMap(map, transformerFactory) @throws IllegalArgumentException if the Map or Transformer is null

MapUtils.lazyMap(map, transformerFactory) @param transformerFactory the factory for creating new objects, must not be null

MapUtils.synchronizedSortedMap(map) @throws IllegalArgumentException if the map is null

MapUtils.synchronizedSortedMap(map) @param map the map to synchronize, must not be null

MapUtils.lazySortedMap(map, transformerFactory) @throws IllegalArgumentException if the Map or Transformer is null

MapUtils.lazySortedMap(map, transformerFactory) @param map the map to make lazy, must not be null

MapUtils.lazySortedMap(map, transformerFactory) @param transformerFactory the factory for creating new objects, must not be null

ListUtils.synchronizedList(list) @param list the list to synchronize, must not be nul

ListUtils.synchronizedList(list) @throws IllegalArgumentException if the list is null

BufferUtils.synchronizedBuffer(buffer) @param buffer the buffer to synchronize, must not be null

BufferUtils.synchronizedBuffer(buffer) @throws IllegalArgumentException if the Buffer is null

BufferUtils.blockingBuffer(buffer) @param buffer the buffer to synchronize, must not be null

BufferUtils.blockingBuffer(buffer) @throws IllegalArgumentException if the Buffer is null

BufferUtils.blockingBuffer(buffer, timeoutMillis) . @param buffer the buffer to synchronize, must not be null

BufferUtils.blockingBuffer(buffer, timeoutMillis) @throws IllegalArgumentException if the Buffer is null

CollectionUtils.synchronizedCollection(collection) @throws IllegalArgumentException if the collection is null

CollectionUtils.synchronizedCollection(collection) @param collection the collection to synchronize, must not be null

SetUtils.synchronizedSortedSet(set) @param set the sorted set to synchronize, must not be null

SetUtils.synchronizedSortedSet(set) @throws IllegalArgumentException if the set is null

BagUtils.transformedBag(bag, transformer) @param bag the bag to predicate, must not be null

BagUtils.transformedBag(bag, transformer) @param transformer the transformer for the bag, must not be null

BagUtils.predicatedSortedBag(bag, predicate) @param bag the sorted bag to predicate, must not be null

BagUtils.predicatedSortedBag(bag, predicate) @param predicate the predicate for the bag, must not be null

BagUtils.transformedSortedBag(bag, transformer) @param bag the bag to predicate, must not be null

BagUtils.transformedSortedBag(bag, transformer) @param transformer the transformer for the bag, must not be null

SetUtils.predicatedSet(set, predicate) @param predicate the predicate for the set, must not be null

SetUtils.predicatedSet(set, predicate) @param set the set to predicate, must not be null

SetUtils.transformedSet(set, transformer) @param set the set to transform, must not be null

SetUtils.transformedSet(set, transformer) @param transformer the transformer for the set, must not be null

SetUtils.predicatedSortedSet(set, predicate) @param predicate the predicate for the sorted set, must not be null

SetUtils.predicatedSortedSet(set, predicate) @param set the sorted set to predicate, must not be null

SetUtils.transformedSortedSet(set, transformer) @param set the set to transform, must not be null

SetUtils.transformedSortedSet(set, transformer) @param transformer the transformer for the set, must not be null

UnmodifiableBoundedCollection.decorateUsing(coll) @param coll the BoundedCollection to decorate, must not be null

UnmodifiableCollection.decorate(coll) @param coll the collection to decorate, must not be null

BlockingBuffer.decorate(buffer, timeoutMillis) @param buffer the buffer to decorate, must not be null

SetUniqueList.decorate(list) @param list the list to decorate, must not be null

DefaultedMap.decorate(map, defaultValue) @param map the map to decorate, must not be null

PredicatedSortedMap.PredicatedSortedMap(map, keyPredicate, valuePredicate) @param map the map to decorate, must not be null

TransformedMap.TransformedMap(map, keyTransformer, valueTransformer) @param map the map to decorate, must not be null

AbstractHashedMap.entryNext(entry) @param entry the entry to query, must not be null

AbstractHashedMap.entryHashCode(entry) @param entry the entry to query, must not be null

AbstractHashedMap.entryKey(entry) @param entry the entry to query, must not be null

AbstractHashedMap.entryValue(entry) @param entry the entry to query, must not be null

TransformedSortedMap.TransformedSortedMap(map, keyTransformer, valueTransformer) @param map the map to decorate, must not be null

AbstractLinkedMap.entryBefore(entry) @param entry the entry to query, must not be null

PredicatedMap.decorate(map, keyPredicate, valuePredicate) @param map the map to decorate, must not be null

PredicatedMap.PredicatedMap(map, keyPredicate, valuePredicate) @param map the map to decorate, must not be null

SplitMapUtils.readableMap(get) @param get to wrap, must not be null

TransformedMap.TransformedMap(map, keyTransformer, valueTransformer) @param map the map to decorate, must not be null

ListOrderedSet.decorate(list) @param list the list to decorate, must not be null

MapUtils.unmodifiableMap(map) @param map the map to make unmodifiable, must not be null

MapUtils.unmodifiableSortedMap(map) @param map the sorted map to make unmodifiable, must not be null

ListUtils.unmodifiableList(list) @param list the list to make unmodifiable, must not be null

CollectionUtils.addIgnoreNull(collection, object) @param collection the collection to add to, must not be null

CollectionUtils.unmodifiableCollection(collection) @param collection the collection to make unmodifiable, must not be null

BagUtils.unmodifiableBag(bag) @param bag the bag whose unmodifiable view is to be returned, must not be null

BagUtils.unmodifiableSortedBag(bag) @param bag the bag whose unmodifiable view is to be returned, must not be null

SetUtils.unmodifiableSet(set) @param set the set to make unmodifiable, must not be null

SetUtils.unmodifiableSortedSet(set) @param set the sorted set to make unmodifiable, must not be null

MapUtils.lazyMap(map, factory) @param map the map to make lazy, must not be null

MapUtils.lazyMap(map, factory) @param factory the factory for creating new objects, must not be null

MapUtils.lazySortedMap(map, factory) @param map the map to make lazy, must not be null

MapUtils.lazySortedMap(map, factory) @param factory the factory for creating new objects, must not be null

ListUtils.transformedList(list, transformer) @param list the list to predicate, must not be null

ListUtils.transformedList(list, transformer) @param transformer the transformer for the list, must not be null

ListUtils.lazyList(list, factory) @param list the list to make lazy, must not be null

ListUtils.lazyList(list, factory) @param factory the factory for creating new objects, must not be null

BoundedBuffer.decorate(buffer, maximumSize) @param buffer the buffer to decorate, must not be null

BoundedBuffer.decorate(buffer, maximumSize, timeout) @param buffer the buffer to decorate, must not be null

PredicatedSortedMap.decorate(map, keyPredicate, valuePredicate) @param map the map to decorate, must not be null

TransformedMap.decorate(map, keyTransformer, valueTransformer) @param map the map to decorate, must not be null

TransformedMap.decorate(map, keyTransformer, valueTransformer) @param map the map to decorate, must not be null

MapUtils.fixedSizeMap(map) @param map the map whose size to fix, must not be null

MapUtils.orderedMap(map) @param map the map to order, must not be null

MapUtils.fixedSizeSortedMap(map) @param map the map whose size to fix, must not be null

ListUtils.fixedSizeList(list) @param list the list whose size to fix, must not be null

BufferUtils.boundedBuffer(buffer, maximumSize) @param buffer the buffer to make bounded, must not be null

SetUtils.synchronizedSet(set) @param set the set to synchronize, must not be null

SetUtils.orderedSet(set) @param set the set to order, must not be null

TransformedMap.decorateTransform(map, keyTransformer, valueTransformer) @param map the map to decorate, must not be null

TransformedSortedMap.decorateTransform(map, keyTransformer, valueTransformer) @param map the map to decorate, must not be null

BufferUtils.boundedBuffer(buffer, maximumSize, timeoutMillis) @param buffer the buffer to make bounded, must not be null

MapUtils.predicatedMap(map, keyPred, valuePred) @param map the map to predicate, must not be null

MapUtils.predicatedSortedMap(map, keyPred, valuePred) @param map the map to predicate, must not be null

MultiKey.MultiKey(keys) @param keys the array of keys, not null

CollectionUtils.disjunction(a, b) @param a the first collection, must not be null

CollectionUtils.disjunction(a, b) @param b the second collection, must not be null

TransformedSortedMap.decorate(map, keyTransformer, valueTransformer) @param map the map to decorate, must not be null

SwitchClosure.getInstance(predicates, closures, defaultClosure) @param closures matching array of closures, cloned, no nulls

SwitchClosure.getInstance(predicates, closures, defaultClosure) @param predicates array of predicates, cloned, no nulls

SwitchClosure.SwitchClosure(predicates, closures, defaultClosure) @param predicates array of predicates, not cloned, no nulls

SwitchClosure.SwitchClosure(predicates, closures, defaultClosure) @param closures matching array of closures, not cloned, no nulls

SwitchTransformer.getInstance(predicates, transformers, defaultTransformer) @param predicates array of predicates, cloned, no nulls

SwitchTransformer.getInstance(predicates, transformers, defaultTransformer) @param transformers matching array of transformers, cloned, no nulls

SwitchTransformer.SwitchTransformer(predicates, transformers, defaultTransformer) @param predicates array of predicates, not cloned, no nulls

SwitchTransformer.SwitchTransformer(predicates, transformers, defaultTransformer) @param transformers matching array of transformers, not cloned, no nulls

ChainedClosure.ChainedClosure(closures) @param closures the closures to chain, not copied, no nulls

ChainedTransformer.ChainedTransformer(transformers) @param transformers the transformers to chain, not copied, no nulls

ChainedClosure.getInstance(closures) @param closures the closures to chain, copied, no nulls

ChainedTransformer.getInstance(transformers) @param transformers the transformers to chain, copied, no nulls

AbstractMapBag.AbstractMapBag(map) The map must be empty and non-null.

MultiKeyMap.put(key, value) Puts the key and value into the map, where the key must be a non-null MultiKey object.

MultiKeyMap.putAll(mapToCopy) Copies all of the keys and values from the specified map to this map. Each key must be non-null and a MultiKey object.

AbstractLinkedList.addNode(nodeToInsert, insertBeforeNode) @throws NullPointerException if either node is null

CursorableLinkedList.addNode(nodeToInsert, insertBeforeNode) @throws NullPointerException if either node is null

AbstractLinkedList.addNodeBefore(node, value) @throws NullPointerException if node is null

AbstractLinkedMap.entryBefore(entry) @throws NullPointerException if the entry is null

ComparatorUtils.booleanComparator(trueFirst) The comparator throws NullPointerException if a null value is compared.

IteratorUtils.arrayListIterator(array, start, end) @throws NullPointerException if array is null

PredicateUtils.transformedPredicate(transformer, predicate) @throws IllegalArgumentException if the transformer or the predicate is null

ObjectArrayIterator.ObjectArrayIterator(array, start, end) @throws NullPointerException if array is null

ObjectArrayListIterator.ObjectArrayListIterator(array, start, end) @throws NullPointerException if array is null

ArrayListIterator.ArrayListIterator(array, startIndex, endIndex) @throws NullPointerException if array is null

IteratorUtils.arrayIterator(array, start, end) @throws NullPointerException if array is null

IteratorUtils.arrayIterator(array, start, end) @throws NullPointerException if array is null

IteratorUtils.arrayListIterator(array, start, end) @throws NullPointerException if array is null

CompositeMap.put(key, value) @throws NullPointerException this map does not permit null keys or values, and the specified key or value is null.

MapUtils.predicatedMap(map, keyPred, valuePred) @throws IllegalArgumentException if the Map is null

SetUtils.predicatedSortedSet(set, predicate) @throws IllegalArgumentException if the Set or Predicate is null

SetUtils.predicatedSet(set, predicate) @throws IllegalArgumentException if the Set or Predicate is null

BagUtils.predicatedSortedBag(bag, predicate) @throws IllegalArgumentException if the SortedBag or Predicate is null

BagUtils.predicatedBag(bag, predicate) @return a predicated bag backed by the given bag @throws IllegalArgumentException if the Bag or Predicate is null

CollectionUtils.predicatedCollection(collection, predicate) @throws IllegalArgumentException if the Collection is null

BufferUtils.predicatedBuffer(buffer, predicate) @throws IllegalArgumentException if the Buffer or Predicate is null

TransformerUtils.asTransformer(closure) @throws IllegalArgumentException if the closure is null

TransformerUtils.asTransformer(closure) @param closure the closure to run each time in the transformer, not null

TransformerUtils.asTransformer(predicate) @throws IllegalArgumentException if the predicate is null

TransformerUtils.asTransformer(predicate) @param predicate the predicate to run each time in the transformer, not null

TransformerUtils.asTransformer(factory) @param factory the factory to run each time in the transformer, not null

TransformerUtils.asTransformer(factory) @throws IllegalArgumentException if the factory is null

BoundedBuffer.decorate(buffer, maximumSize, timeout) @throws IllegalArgumentException if the buffer is null

PrototypeFactory.getInstance(prototype) @throws IllegalArgumentException if the prototype is null

FactoryUtils.prototypeFactory(prototype) @throws IllegalArgumentException if the prototype is null

FactoryUtils.instantiateFactory(classToInstantiate) @throws IllegalArgumentException if the classToInstantiate is null

FactoryUtils.instantiateFactory(classToInstantiate, paramTypes, args) @throws IllegalArgumentException if the classToInstantiate is null

CompositeMap.containsValue(value) @throws NullPointerException if the value is null and this map does not not permit null values (optional).

KeyLookupFactory.setDefault(defaultLookup) @param defaultLookup the default look-up. Must not be null.

KeyFormatterFactory.setDefault(defaultKeyFormatter) @param defaultKeyFormatter the default key formatter. Must not be null.

HttpCookie.HttpCookie(name, value) @throws NullPointerException if name is null

ColorConvertOp.ColorConvertOp(srcCspace, dstCspace, hints) @throws NullPointerException if either srcCspace or dstCspace is null

(abstract) LoggingMXBean.setLoggerLevel(loggerName, levelName) @param loggerName The name of the Logger to be set. Must be non-null.

SerializablePermission.SerializablePermission(name, actions) @throws NullPointerException if name is null.

ObjectArrayIterator.setArray(array) @throws NullPointerException if array is null

MapUtils.verbosePrint(out, label, map) @param out the stream to print to, must not be null

MapUtils.verbosePrint(out, label, map) @throws NullPointerException if the stream is null

BoundedBuffer.BoundedBuffer(buffer, maximumSize, timeout) @throws IllegalArgumentException if the buffer is null

CompositeMap.containsKey(key) @throws NullPointerException if the key is null and this map does not not permit null keys (optional).

interface Queue Even in the implementations that permit it, null should not be inserted into a Queue, as null is also used as a special return value by the poll method to indicate that the queue contains no elements.

IteratorUtils.toListIterator(iterator) @throws NullPointerException if iterator parameter is null

MapUtils.debugPrint(out, label, map) @param out the stream to print to, must not be null

MapUtils.debugPrint(out, label, map) @throws NullPointerException if the stream is null

AbstractSetDecorator.AbstractSetDecorator(set) @throws IllegalArgumentException if set is null

MultiKeyMap.MultiKeyMap(map) The map must not be null and should be empty or only contain valid keys.

URLStreamHandler.equals(u1, u2) This method requires that none of its arguments is null.

URLStreamHandler.sameFile(u1, u2) This method requires that none of its arguments is null.

TriggerSequence.startsWith(triggerSequence, equals) Must not be null.

StatusLineContributionItem.setText(text) @param text the text to be displayed, must not be null

(abstract) ExternalActionManager.ICallback.getAcceleratorText(identifier) @param identifier The identifier of the item from which the accelerator text should be obtained ; must not be null.

ActionHandler.ActionHandler(action) @param action the action. Must not be null.

StructuredSelection.StructuredSelection(elements) The given element array must not be null.

(abstract) IControlContentAdapter.insertControlContents(control, contents, cursorPosition) Insert the specified contents into the control's current contents. Must not be null.

BindingManager.removeBindingManagerListener(listener) @param listener The listener to be removed; must not be null.

Scheme.addSchemeListener(schemeListener) @param schemeListener the instance of ISchemeListener to register. Must not be null.

BindingManager.setPlatform(platform) @param platform The new platform; must not be null.

(abstract) Binding.getTriggerSequence() @return The object representing an input event that will trigger this binding; must not be null.

KeySequence.getInstance(keyStroke) @param keyStroke a single key stroke. Must not be null.

(abstract) ExternalActionManager.ICallback.addPropertyChangeListener(identifier, listener) @param identifier The identifier of the item to which the listener should be attached; must not be null.

BindingManager.getPartialMatches(trigger) @param trigger The prefix to look for; must not be null.

MessageDialogWithToggle.createToggleButton(parent) @param parent The composite in which the toggle button should be placed; must not be null.

PredicateUtils.eitherPredicate(predicate1, predicate2) @param predicate1 the first predicate, may not be null

PredicateUtils.eitherPredicate(predicate1, predicate2) @param predicate2 the second predicate, may not be null

ViewerLabel.setText(text) @param text String The label to set. This value should not be null.

(abstract) ExternalActionManager.IExecuteCallback.postExecuteSuccess(action, returnValue) Cannot be null.

class JarFile Unless otherwise noted, passing a null argument to a constructor or method in this class will cause a @link NullPointerException to be thrown.

InflaterOutputStream.write(b, off, len) @throws NullPointerException if b is null

NetworkInterface.getByInetAddress(addr) @throws NullPointerException If the specified address is null.

PrintStream.format(format, args) @throws NullPointerException If the format is null

LinkedBlockingQueue.offer(e) @throws NullPointerException if the specified element is null

(abstract) BlockingDeque.push(e) @throws NullPointerException if the specified element is null

System.load(filename) @exception NullPointerException if filename is null

(abstract) ExternalActionManager.IExecuteCallback.notEnabled(action, exception) @param exception The NotEnabledException, never null.

(abstract) ExternalActionManager.IExecuteCallback.notEnabled(action, exception) @param action The action contribution that caused the exception, never null.

DefaultDamagerRepairer.DefaultDamagerRepairer(scanner) The scanner may not be null

(abstract) ExternalActionManager.IExecuteCallback.postExecuteFailure(action, exception) @param exception The ExecutionException, never null.

PrintWriter.format(format, args) @throws NullPointerException If the format is null

AbstractSortedMapDecorator.AbstractSortedMapDecorator(map) @throws IllegalArgumentException if the collection is null

1.2 Null Semantics Directive

DefaultMapEntry.DefaultMapEntry(key, value) @param key the key for the entry, may be null

DefaultMapEntry.DefaultMapEntry(key, value) @param value the value for the entry, may be null

(abstract) DatabaseMetaData.getColumnPrivileges(catalog, schema, table, columnNamePattern) null means that the catalog name should not be used to narrow the search

IndexedPropertyDescriptor.IndexedPropertyDescriptor(propertyName, beanClass, readMethodName, writeMethodName, indexedReadMethodName, indexedWriteMethodName) @param readMethodName May be null if the property is write-only

IndexedPropertyDescriptor.IndexedPropertyDescriptor(propertyName, readMethod, writeMethod, indexedReadMethod, indexedWriteMethod) @param readMethod May be null if the property is write-only

LogRecord.setMessage(message) @param message the raw message string (may be null)

ContentProposalAdapter.setAutoActivationCharacters(autoActivationCharacters) If this parameter is null, then only a specified keyStroke will invoke content proposal

ContentProposalAdapter.ContentProposalAdapter(control, controlContentAdapter, proposalProvider, keyStroke, autoActivationCharacters) If this parameter is null, then only a specified keyStroke will invoke content proposal.

ContentProposalAdapter.ContentProposalAdapter(control, controlContentAdapter, proposalProvider, keyStroke, autoActivationCharacters) the IContentProposalProvider used to obtain content proposals for this control, or null if no content proposal is available

PreferencePage.setPreferenceStore(store) the preference store, or null

ContentViewer.setLabelProvider(labelProvider) @param labelProvider the label provider, or null if none

(abstract) DatabaseMetaData.getSchemas(catalog, schemaPattern) null means catalog name should not be used to narrow down the search.

(abstract) DatabaseMetaData.getTables(catalog, schemaPattern, tableNamePattern, types) null means that the catalog name should not be used to narrow the search

(abstract) DatabaseMetaData.getTables(catalog, schemaPattern, tableNamePattern, types) null returns all types

BindingManager.setBindings(bindings) @param bindings The new array of bindings; may be null.

MessageDialogWithToggle.open(kind, parent, title, message, toggleMessage, toggleState, store, key, style) @param store the IPreference store in which the user's preference should be persisted; null if you don't want it persisted automatically.

MessageDialogWithToggle.open(kind, parent, title, message, toggleMessage, toggleState, store, key, style) @param key the key to use when persisting the user's preference; null if you don't want it persisted.

MessageDialogWithToggle.openOkCancelConfirm(parent, title, message, toggleMessage, toggleState, store, key) @param store the IPreference store in which the user's preference should be persisted; null if you don't want it persisted automatically.

MemoryUsage.from(cd) @return a MemoryUsage object represented by cd if cd is not null; null otherwise.

MemoryNotificationInfo.from(cd) @return a MemoryNotificationInfo object represented by cd if cd is not null; null otherwise.

AbstractRulerColumn.setFont(font) the font or null to use the default font

MapUtils.verbosePrint(out, label, map) @param label The label to be used, may be null. If null, the label is not output.

MapUtils.verbosePrint(out, label, map) @param map The map to print, may be null. If null, the text 'null' is output.

MapUtils.debugPrint(out, label, map) @param label The label to be used, may be null. If null, the label is not output.

MapUtils.debugPrint(out, label, map) param map The map to print, may be null. If null, the text 'null' is output.

URL.URL(protocol, host, port, file, handler) Specifying a handler of null indicates that the URL should use a default stream handler for the protocol, as outlined for: java.net.URL#URL(java.lang.String, java.lang.String, int, java.lang.String)

(abstract) DatabaseMetaData.getPrimaryKeys(catalog, schema, table) null means that the catalog name should not be used to narrow the search

(abstract) DatabaseMetaData.getPrimaryKeys(catalog, schema, table) null means that the schema name should not be used to narrow the search

(abstract) DatabaseMetaData.getCrossReference(parentCatalog, parentSchema, parentTable, foreignCatalog, foreignSchema, foreignTable) null means drop catalog name from the selection criteria

(abstract) DatabaseMetaData.getCrossReference(parentCatalog, parentSchema, parentTable, foreignCatalog, foreignSchema, foreignTable) null means drop schema name from the selection criteria

(abstract) DatabaseMetaData.getCrossReference(parentCatalog, parentSchema, parentTable, foreignCatalog, foreignSchema, foreignTable) null means drop catalog name from the selection criteria

AttributedString.getIterator(attributes) If the list is null, all available attribute information should be made accessible.

AttributedString.getIterator(attributes, beginIndex, endIndex) If the list is null, all available attribute information should be made accessible.

X509CertSelector.setIssuer(issuerDN) If issuerDN is not null, it should contain a distinguished name, in RFC 2253 format.

PKIXParameters.setDate(date) If null, the current time is used.

X509CertSelector.setSubject(subjectDN) If subjectDN is not null, it should contain a single DER encoded distinguished name, as defined in X.501.

X509CertSelector.setIssuer(issuerDN) If issuerDN is not null, it should contain a single DER encoded distinguished name, as defined in X.501.

X509CertSelector.setSubjectKeyIdentifier(subjectKeyID) If subjectKeyID is not null, it should contain a single DER encoded value corresponding to the contents of the extension value (not including the object identifier, criticality setting, and encapsulating OCTET STRING) for a SubjectKeyIdentifier extension.

X509CertSelector.setSubjectPublicKey(key) If key is not null, it should contain a single DER encoded SubjectPublicKeyInfo structure, as defined in X.509.

(abstract) CertStoreSpi.engineGetCertificates(selector) Specify null to return all Certificates (if supported).

(abstract) CertStoreSpi.engineGetCRLs(selector) Specify null to return all CRLs (if supported).

CertStore.getCertificates(selector) Specify null to return all Certificates (if supported).

CertStore.getCRLs(selector) Specify null to return all CRLs (if supported).

(abstract) DatabaseMetaData.getColumnPrivileges(catalog, schema, table, columnNamePattern) null means that the schema name should not be used to narrow the search

Runtime.exec(command, envp, dir) @param envp array of strings, each element of which has environment variable settings in the format name=value, or null if the subprocess should inherit the environment of the current process.

Runtime.exec(command, envp, dir) @param dir the working directory of the subprocess, or null if the subprocess should inherit the working directory of the current process.

Runtime.exec(command, envp) @param envp array of strings, each element of which has environment variable settings in the format name=value, or null if the subprocess should inherit the environment of the current process.

Runtime.exec(cmdarray, envp, dir) @param envp array of strings, each element of which has environment variable settings in the format name=value, or null if the subprocess should inherit the environment of the current process.

Runtime.exec(cmdarray, envp, dir) @param dir the working directory of the subprocess, or null if the subprocess should inherit the working directory of the current process.

Runtime.exec(cmdarray, envp) @param envp array of strings, each element of which has environment variable settings in the format name=value, or null if the subprocess should inherit the environment of the current process.

Handler.setEncoding(encoding) @param encoding The name of a supported character encoding. May be null, to indicate the default platform encoding.

CollectionUtils.collect(inputCollection, transformer, outputCollection) @param inputCollection the collection to get the input from, may be null

CollectionUtils.collect(inputCollection, transformer, outputCollection) @param transformer the transformer to use, may be null

CollectionUtils.collect(inputCollection, transformer, outputCollection) @param outputCollection the collection to output into, may not be null

CollectionUtils.collect(inputIterator, transformer, outputCollection) @param inputIterator the iterator to get the input from, may be null

CollectionUtils.collect(inputIterator, transformer, outputCollection) @param transformer the transformer to use, may be null

CollectionUtils.collect(inputIterator, transformer, outputCollection) @param outputCollection the collection to output into, may not be null

ObjectGraphIterator.ObjectGraphIterator(root, transformer) @param root the root object, null will result in an empty iterator

ObjectGraphIterator.ObjectGraphIterator(root, transformer) @param transformer the transformer to use, null will use a no effect transformer

IteratorUtils.objectGraphIterator(root, transformer) @param transformer the transformer to use, see above, null uses no effect transformer

IteratorUtils.objectGraphIterator(root, transformer) @param root the root object to start iterating from, null results in an empty iterator

AbstractDualBidiMap.AbstractDualBidiMap(normalMap, reverseMap) Neither map is validated, so nulls may be passed in.

MapUtils.transformedMap(map, keyTransformer, valueTransformer) @param valueTransformer the transformer for the map values, null means no transformation

MapUtils.transformedSortedMap(map, keyTransformer, valueTransformer) @param valueTransformer the transformer for the map values, null means no transformation

MapUtils.putAll(map, array) @param array an array to populate from, null ignored

PredicatedSortedMap.PredicatedSortedMap(map, keyPredicate, valuePredicate) @param keyPredicate the predicate to validate the keys, null means no check

PredicatedSortedMap.PredicatedSortedMap(map, keyPredicate, valuePredicate) @param valuePredicate the predicate to validate to values, null means no check

TransformedMap.TransformedMap(map, keyTransformer, valueTransformer) @param keyTransformer the transformer to use for key conversion, null means no conversion

TransformedMap.TransformedMap(map, keyTransformer, valueTransformer) @param valueTransformer the transformer to use for value conversion, null means no conversion

TransformedSortedMap.TransformedSortedMap(map, keyTransformer, valueTransformer) @param keyTransformer the predicate to validate the keys, null means no transformation

TransformedSortedMap.TransformedSortedMap(map, keyTransformer, valueTransformer) @param valueTransformer the predicate to validate to values, null means no transformation

PredicatedMap.decorate(map, keyPredicate, valuePredicate) @param keyPredicate the predicate to validate the keys, null means no check

PredicatedMap.decorate(map, keyPredicate, valuePredicate) @param valuePredicate the predicate to validate to values, null means no check

PredicatedMap.PredicatedMap(map, keyPredicate, valuePredicate) @param keyPredicate the predicate to validate the keys, null means no check

PredicatedMap.PredicatedMap(map, keyPredicate, valuePredicate) @param valuePredicate the predicate to validate to values, null means no check

CollectionUtils.addIgnoreNull(collection, object) @param object the object to add, if null it will not be added @return true if the collection changed

TransformedMap.decorate(map, keyTransformer, valueTransformer) @param keyTransformer the transformer to use for key conversion, null means no transformation

TransformedMap.decorate(map, keyTransformer, valueTransformer) @param valueTransformer the transformer to use for value conversion, null means no transformation

TransformedMap.decorate(map, keyTransformer, valueTransformer) @param keyTransformer the transformer to use for key conversion, null means no transformation

TransformedMap.decorate(map, keyTransformer, valueTransformer) @param valueTransformer the transformer to use for value conversion, null means no transformation

TransformedMap.decorateTransform(map, keyTransformer, valueTransformer) @param keyTransformer the transformer to use for key conversion, null means no transformation

TransformedMap.decorateTransform(map, keyTransformer, valueTransformer) @param valueTransformer the transformer to use for value conversion, null means no transformation

TransformedSortedMap.decorateTransform(map, keyTransformer, valueTransformer) @param keyTransformer the transformer to use for key conversion, null means no transformation

TransformedSortedMap.decorateTransform(map, keyTransformer, valueTransformer) @param valueTransformer the transformer to use for value conversion, null means no transformation

MapUtils.predicatedMap(map, keyPred, valuePred) @param keyPred the predicate for keys, null means no check

MapUtils.predicatedMap(map, keyPred, valuePred) @param valuePred the predicate for values, null means no check

MapUtils.predicatedSortedMap(map, keyPred, valuePred) @param keyPred the predicate for keys, null means no check

MapUtils.predicatedSortedMap(map, keyPred, valuePred) @param valuePred the predicate for values, null means no check

TransformedSortedMap.decorate(map, keyTransformer, valueTransformer) @param keyTransformer the predicate to validate the keys, null means no transformation

TransformedSortedMap.decorate(map, keyTransformer, valueTransformer) @param valueTransformer the predicate to validate to values, null means no transformation

PriorityBuffer.PriorityBuffer(comparator) @param comparator the comparator used to order the elements, null means use natural order

TransformerClosure.getInstance(transformer) @param transformer the transformer to call, null means nop

TransformedMap.TransformedMap(map, keyTransformer, valueTransformer) @param keyTransformer the transformer to use for key conversion, null means no conversion

TransformedMap.TransformedMap(map, keyTransformer, valueTransformer) @param keyTransformer the transformer to use for key conversion, null means no conversion

TransformedMap.TransformedMap(map, keyTransformer, valueTransformer) @param valueTransformer the transformer to use for value conversion, null means no conversion

PredicatedSortedMap.decorate(map, keyPredicate, valuePredicate) @param keyPredicate the predicate to validate the keys, null means no check

PredicatedSortedMap.decorate(map, keyPredicate, valuePredicate) @param valuePredicate the predicate to validate to values, null means no check

PriorityBuffer.PriorityBuffer(ascendingOrder, comparator) @param comparator the comparator used to order the elements, null means use natural order

PriorityBuffer.PriorityBuffer(capacity, comparator) @param comparator the comparator used to order the elements, null means use natural order

PriorityBuffer.PriorityBuffer(capacity, ascendingOrder, comparator) @param comparator the comparator used to order the elements, null means use natural order

SwitchClosure.getInstance(predicates, closures, defaultClosure) @param defaultClosure the closure to use if no match, null means nop

SwitchClosure.SwitchClosure(predicates, closures, defaultClosure) @param defaultClosure the closure to use if no match, null means nop

SwitchTransformer.getInstance(predicates, transformers, defaultTransformer) @param defaultTransformer the transformer to use if no match, null means return null @return the chained transformer

SwitchTransformer.SwitchTransformer(predicates, transformers, defaultTransformer) @param defaultTransformer the transformer to use if no match, null means return null

ClosureUtils.asClosure(transformer) @param transformer the transformer to run each time in the closure, null means nop

TransformerUtils.switchTransformer(predicates, transformers, defaultTransformer) @param defaultTransformer the default to call if no predicate matches, null means return null

TransformerUtils.invokerTransformer(methodName) If the input object is null, null is returned.

ListUtils.isEqualList(list1, list2) @param list1 the first list, may be null

ListUtils.isEqualList(list1, list2) @param list2 the second list, may be null

PredicateUtils.uniquePredicate() A null input object is accepted and will return true the first time, and false subsequently as well.

ExtendedProperties.interpolateHelper(base, priorVariables) When called the first time, priorVariables should be null.

FactoryUtils.instantiateFactory(classToInstantiate, paramTypes, args) @param args the arguments to pass to the constructor, can be null

FileFieldEditor.setFileExtensions(extensions) @param extensions a list of file extension, or null to set the filter to the system's default value

SQLPermission.SQLPermission(name, actions) @param actions should be null

Arrays.sort(a, c) A null value indicates that the elements' @linkplain Comparable natural ordering should be used.

File.createTempFile(prefix, suffix) may be null, in which case the suffix ".tmp" will be used

File.createTempFile(prefix, suffix, directory) @param suffix The suffix string to be used in generating the file's name; may be null, in which case the suffix ".tmp" will be used

URL.URL(protocol, host, port, file, handler) Specifying a handler of null indicates that the URL should use a default stream handler for the protocol, as outlined for: java.net.URL#URL(java.lang.String, java.lang.String, int, java.lang.String)

NetPermission.NetPermission(name, actions) the actions String is currently unused and should be null

ReflectPermission.ReflectPermission(name, actions) The actions should be null; they are ignored.

RuntimePermission.RuntimePermission(name, actions) the actions String is currently unused and should be null.

Arrays.binarySearch(a, key, c) A null value indicates that the elements' @linkplain Comparable natural ordering should be used.

Collections.binarySearch(list, key, c) A null value indicates that the elements' @linkplain Comparable natural ordering should be used.

WeakReference.WeakReference(referent, q) @param q the queue with which the reference is to be registered, or null if registration is not required

SoftReference.SoftReference(referent, q) @param q the queue with which the reference is to be registered, or null if registration is not required

PhantomReference.PhantomReference(referent, q) @param q the queue with which the reference is to be registered, or null if registration is not required

Method.invoke(obj, args) If the number of formal parameters required by the underlying method is 0, the supplied args array may be of length 0 or null.

Constructor.newInstance(initargs) If the number of formal parameters required by the underlying constructor is 0, the supplied initargs array may be of length 0 or null.

SecurityPermission.SecurityPermission(name, actions) @param actions should be null.

Logger.getAnonymousLogger(resourceBundleName) @param resourceBundleName name of ResourceBundle to be used for localizing messages for this logger. May be null if none of the messages require localization.

LogManager.removePropertyChangeListener(l) @param l event listener (can be null)

GroupMarker.GroupMarker(groupName) The group name must not be null or the empty string.

SQLInvalidAuthorizationSpecException.SQLInvalidAuthorizationSpecException(reason, SQLState, vendorCode, cause) @param cause the underlying reason for this SQLException (which is saved for later retrieval by the getCause() method); may be null indicating the cause is non-existent or unknown

SQLFeatureNotSupportedException.SQLFeatureNotSupportedException(cause) @param cause the underlying reason for this SQLException (which is saved for later retrieval bythe getCause() method); may be null indicating the cause is non-existent or unknown.

Policy.getInstance(type, params, provider) @param params parameters for the Policy, which may be null.

UnmodifiableMapEntry.UnmodifiableMapEntry(key, value) @param key the key for the entry, may be null

CollectionUtils.isNotEmpty(coll) @param coll the collection to check, may be null

CollectionUtils.filter(collection, predicate) @param collection the collection to get the input from, may be null @param predicate the predicate to use as a filter, may be null

CollectionUtils.isEmpty(coll) @param coll the collection to check, may be null

DefaultKeyValue.DefaultKeyValue(key, value) @param key the key for the entry, may be null

CollectionUtils.countMatches(input, predicate) @param input the @link Iterable to get the input from, may be null

CollectionUtils.countMatches(input, predicate) @param predicate the predicate to use, may be null

ControlDecoration.showHoverText(text) @param text the text to be shown in the info hover, or null if no text should be shown.

TrustAnchor.TrustAnchor(caPrincipal, pubKey, nameConstraints) Specify null to omit the parameter.

(abstract) IInputValidator.isValid(newText) Note that the empty string is not treated the same as null; it indicates an error state but with no message to display.

(abstract) DomainCombiner.combine(currentDomains, assignedDomains) This parameter may be null if there are no inherited ProtectionDomains.

Properties.storeToXML(os, comment, encoding) @param comment a description of the property list, or null if no comment is desired.

DefaultInformationControl.DefaultInformationControl(parent, toolBarManager, presenter) @param toolBarManager the manager or null if toolbar is not desired

GuardedObject.GuardedObject(object, guard) If the Guard object is null, then no restrictions will be placed on who can access the object.

Util.toString(array) @param array The array to convert; may be null

(abstract) ISourceViewer.setAnnotationHover(annotationHover) @param annotationHover the hover to be used, null is a valid argument

AccessController.doPrivileged(action, context) @param context an access control context representing the restriction to be applied to the caller's domain's privileges before performing the specified action. If the context is null, then no additional restriction is applied.

StructuredViewer.handleInvalidSelection(invalidSelection, newSelection) @param newSelection the selection after the update, or null if none

LegacyActionTools.extractAcceleratorText(text) @param text the text for the action; may be null.

ExtendedProperties.save(output, header) @param output an OutputStream, may be null

1.3 Number Range Directive

(abstract) Bag.remove(object, nCopies) If the number of copies to remove is greater than the actual number of copies in the Bag, no error is thrown.

BoundedBuffer.decorate(buffer, maximumSize) @param maximumSize the maximum size, must be size one or greater

BoundedBuffer.decorate(buffer, maximumSize, timeout) @param maximumSize the maximum size, must be size one or greater

AbstractLinkedMap.AbstractLinkedMap(initialCapacity, loadFactor, threshold) @param initialCapacity the initial capacity, must be a power of two @param loadFactor the load factor, must be ¿ 0.0f and generally ¡ 1.0f

AbstractHashedMap.AbstractHashedMap(initialCapacity, loadFactor, threshold) @param initialCapacity the initial capacity, must be a power of two @param loadFactor the load factor, must be ¿ 0.0f and generally ¡ 1.0f

UnboundedFifoBuffer.UnboundedFifoBuffer(initialSize) Constructs an UnboundedFifoBuffer with the specified number of elements. The integer must be a positive integer.

(abstract) RegistryHandler.registryImpl(port) The port must be non-zero.

BigDecimal.pow(n) The parameter n must be in the range 0 through 999999999, inclusive.

ServerSocket.setSoTimeout(timeout) The timeout must be ¿ 0.

FileLock.FileLock(channel, position, size, shared) @param position must be non-negative

FileLock.FileLock(channel, position, size, shared) @param size must be non-negative, and the sum position + size must be non-negative

Scrollbar.setBlockIncrement(v) The block increment must be greater than zero

Scrollbar.setUnitIncrement(v) The unit increment must be greater than zero.

NumberFormat.setMinimumIntegerDigits(newValue) minimumIntegerDigits must be ¡= maximumIntegerDigits.

DatagramSocket.setSoTimeout(timeout) The timeout must be ¿ 0.

MulticastSocket.setTimeToLive(ttl) The ttl must be in the range 0 ¡= ttl ¡= 255

Socket.setReceiveBufferSize(size) @param size the size to which to set the receive buffer size. This value must be greater than 0.

ServerSocket.ServerSocket(port, backlog, bindAddr) The port must be between 0 and 65535, inclusive.

ServerSocket.ServerSocket(port, backlog, bindAddr) The backlog argument must be a positive value greater than 0.

DatagramSocket.setTrafficClass(tc) The tc must be in the range 0 ¡= tc ¡= 255 or an IllegalArgumentException will be thrown.

DatagramSocket.DatagramSocket(port, laddr) The local port must be between 0 and 65535 inclusive.

Socket.setTrafficClass(tc) The tc must be in the range 0 ¡= tc ¡= 255 or an IllegalArgumentException will be thrown.

abstract class TimeZone Hours must be between 0 to 23

abstract class TimeZone Minutes must be between 00 to 59.

ServerSocket.ServerSocket(port, backlog) The backlog argument must be a positive value greater than 0.

MemoryHandler.MemoryHandler(target, size, pushLevel) @param size the number of log records to buffer (must be greater than zero)

NumberFormat.setMaximumIntegerDigits(newValue) maximumIntegerDigits must be ¿= minimumIntegerDigits.

ContentAssistant.setAutoActivationDelay(delay) (as of 3.6 a negative argument will be set to 0)

Socket.setSoTimeout(timeout) The timeout must be ¿ 0.

DatagramSocket.setSendBufferSize(size) @param size the size to which to set the send buffer size. This value must be greater than 0.

Date.Date(year, month, day) @param year the year minus 1900; must be 0 to 8099.

MulticastSocket.setTTL(ttl) The ttl is an unsigned 8-bit quantity, and so must be in the range 0 ¡= ttl ¡= 0xFF

Socket.setSendBufferSize(size) @param size the size to which to set the send buffer size. This value must be greater than 0.

ServerSocket.bind(endpoint, backlog) The backlog argument must be a positive value greater than 0.

JarFile.JarFile(file, verify, mode) The mode argument must be either OPEN_READ or OPEN_READ — OPEN_DELETE.

Vector.removeElementAt(index) The index must be a value greater than or equal to 0 and less than the current size of the vector.

DatagramSocket.setReceiveBufferSize(size) @param size the size to which to set the receive buffer size. This value must be greater than 0

ZipFile.ZipFile(file, mode) The mode argument must be either OPEN_READ or OPEN_READ — OPEN_DELETE.

Vector.setElementAt(obj, index) The index must be a value greater than or equal to 0 and less than the current size of the vector.

(abstract) MemoryPoolMXBean.setUsageThreshold(threshold) @param threshold the new threshold value in bytes. Must be non-negative.

(abstract) MemoryPoolMXBean.setCollectionUsageThreshold(threhsold) @param threhsold the new collection usage threshold value in bytes. Must be non-negative.

PipedReader.PipedReader(pipeSize) @exception IllegalArgumentException if pipeSize ¡= 0.

StringReader.mark(readAheadLimit) so this argument must not be negative, but is otherwise ignored

class SimpleDateFormat Hours must be between 0 and 23, and Minutes must be between 00 and 59. The format is locale independent and digits must be taken from the Basic Latin block of the Unicode standard.

class SimpleDateFormat woDigitHours must be between 00 and 23. Other definitions are as for general time zones.

CursorableLinkedList.cursor(fromIndex) @throws IndexOutOfBoundsException if the index is out of range (index ¡ 0 —— index ¿ size()).

(abstract) CallableStatement.registerOutParameter(parameterIndex, sqlType, scale) @param scale the desired number of digits to the right of the decimal point. It must be greater than or equal to zero.

BoundedBuffer.BoundedBuffer(buffer, maximumSize, timeout) @param maximumSize the maximum size, must be size one or greater

BoundedBuffer.BoundedBuffer(buffer, maximumSize, timeout) @throws IllegalArgumentException if the maximum size is zero or less

Arrays.copyOfRange(original, from, to) The initial index of the range (from) must lie between zero and original.length, inclusive.

Arrays.copyOfRange(original, from, to) The final index of the range (to), which must be greater than or equal to from, may be greater than original.length, in which case 0 is placed in all elements of the copy whose index is greater than or equal to original.length - from.

HttpCookie.setVersion(v) @param v 0 if the cookie should comply with the original Netscape specification; 1 if the cookie should comply with RFC 2965/2109

InetAddress.isReachable(netif, ttl, timeout) A negative value for the ttl will result in an IllegalArgumentException being thrown.

InetAddress.isReachable(timeout) A negative value will result in an IllegalArgumentException being thrown

InetAddress.isReachable(timeout) @throws IllegalArgumentException if timeout is negative.

URL.URL(protocol, host, port, file) Specifying a port number of -1 indicates that the URL should use the default port for the protocol.

(abstract) ThreadMXBean.getThreadInfo(id) @param id the thread ID of the thread. Must be positive.

(abstract) ThreadMXBean.getThreadInfo(id, maxDepth) @param id the thread ID of the thread. Must be positive.

Semaphore.Semaphore(permits) This value may be negative, in which case releases must occur before any acquires will be granted.

Semaphore.Semaphore(permits, fair) This value may be negative, in which case releases must occur before any acquires will be granted.

ByteArrayInputStream.pos This value should always be nonnegative and not larger than the value of count.

ByteArrayInputStream.count This value should always be nonnegative and not larger than the length of buf.

ECFieldF2m.ECFieldF2m(m, ks) A valid reduction polynomial is either a trinomial (Xm̂ + Xk̂ + 1 with m ¿ k ¿= 1) or a pentanomial (Xm̂ + Xk̂3 + Xk̂2 + Xk̂1 + 1 with m ¿ k3 ¿ k2 ¿ k1 ¿= 1), so ks should have length 1 or 3.

(abstract) Connection.prepareStatement(sql, autoGeneratedKeys) @param autoGeneratedKeys a flag indicating whether auto-generated keys should be returned; one of Statement.RETURN_GENERATED_KEYS or Statement.NO_GENERATED_KEYS

Float.intBitsToFloat(bits) If the argument is any value in the range 0x7f800001 through 0x7fffffff or in the range 0xff800001 through 0xffffffff, the result is a NaN. all NaN bit patterns, quiet or signaling, must be in the NaN range identified above.

Pack200.Unpacker.PROGRESS At a minimum, the unpacker must set progress to 0 at the beginning of a packing operation, and to 100 at the end.

GregorianCalendar.isLeapYear(year) To specify BC year numbers, 1 - year number must be given.

Arrays.copyOfRange(original, from, to) The initial index of the range (from) must lie between zero and original.length, inclusive.

(abstract) Statement.getMoreResults(current) @param current one of the following Statement constants indicating what should happen to current ResultSet objects obtained using the method getResultSet: Statement.CLOSE_CURRENT_RESULT, Statement.KEEP_CURRENT_RESULT, or Statement.CLOSE_ALL_RESULTS

Arrays.copyOfRange(original, from, to) The initial index of the range (from) must lie between zero and original.length, inclusive.

Double.longBitsToDouble(bits) If the argument is any value in the range 0x7ff0000000000001L through 0x7fffffffffffffffL or in the range 0xfff0000000000001L through 0xffffffffffffffffL, the result is a NaN. all NaN bit patterns, quiet or signaling, must be in the NaN range identified above.

(abstract) CallableStatement.registerOutParameter(parameterName, sqlType, scale) @param scale the desired number of digits to the right of the decimal point. It must be greater than or equal to zero

Arrays.copyOfRange(original, from, to) The initial index of the range (from) must lie between zero and original.length, inclusive.

PriorityBuffer.PriorityBuffer(capacity, comparator) @param capacity the initial capacity for the buffer, greater than zero

ReferenceIdentityMap.ReferenceIdentityMap(keyType, valueType) @param valueType the type of reference to use for values; must be @link #HARD, @link #SOFT, @link #WEAK

ReferenceIdentityMap.ReferenceIdentityMap(keyType, valueType) @param keyType the type of reference to use for keys; must be @link #HARD, @link #SOFT, @link #WEAK

ReferenceIdentityMap.ReferenceIdentityMap(keyType, valueType, purgeValues) @param valueType the type of reference to use for values; must be @link #HARD, @link #SOFT, @link #WEAK

ReferenceIdentityMap.ReferenceIdentityMap(keyType, valueType, purgeValues) @param keyType the type of reference to use for keys; must be @link #HARD, @link #SOFT, @link #WEAK

ReferenceMap.ReferenceMap(keyType, valueType, capacity, loadFactor) @param keyType the type of reference to use for keys; must be @link #HARD, @link #SOFT, @link #WEAK

ReferenceMap.ReferenceMap(keyType, valueType, capacity, loadFactor) @param valueType the type of reference to use for values; must be @link #HARD, @link #SOFT, @link #WEAK

ReferenceMap.ReferenceMap(keyType, valueType, capacity, loadFactor, purgeValues) @param keyType the type of reference to use for keys; must be @link #HARD, @link #SOFT, @link #WEAK

ReferenceMap.ReferenceMap(keyType, valueType, capacity, loadFactor, purgeValues) @param valueType the type of reference to use for values; must be @link #HARD, @link #SOFT, @link #WEAK

ReferenceIdentityMap.ReferenceIdentityMap(keyType, valueType, capacity, loadFactor) @param keyType the type of reference to use for keys; must be @link #HARD, @link #SOFT, @link #WEAK

ReferenceIdentityMap.ReferenceIdentityMap(keyType, valueType, capacity, loadFactor) @param valueType the type of reference to use for values; must be @link #HARD, @link #SOFT, @link #WEAK

ReferenceMap.ReferenceMap(keyType, valueType, purgeValues) @param keyType the type of reference to use for keys; must be @link #HARD, @link #SOFT, @link #WEAK

ReferenceMap.ReferenceMap(keyType, valueType, purgeValues) @param valueType the type of reference to use for values; must be @link #HARD, @link #SOFT, @link #WEAK

ReferenceIdentityMap.ReferenceIdentityMap(keyType, valueType, capacity, loadFactor, purgeValues) @param keyType the type of reference to use for keys; must be @link #HARD, @link #SOFT, @link #WEAK

ReferenceIdentityMap.ReferenceIdentityMap(keyType, valueType, capacity, loadFactor, purgeValues) @param valueType the type of reference to use for values; must be @link #HARD, @link #SOFT, @link #WEAK

AbstractReferenceMap.AbstractReferenceMap(keyType, valueType, capacity, loadFactor, purgeValues) @param keyType the type of reference to use for keys; must be @link #HARD, @link #SOFT, @link #WEAK

AbstractReferenceMap.AbstractReferenceMap(keyType, valueType, capacity, loadFactor, purgeValues) @param valueType the type of reference to use for values; must be @link #HARD, @link #SOFT, @link #WEAK

ReferenceMap.ReferenceMap(keyType, valueType) @param keyType the type of reference to use for keys; must be @link #HARD, @link #SOFT, @link #WEAK

ReferenceMap.ReferenceMap(keyType, valueType) @param valueType the type of reference to use for values; must be @link #HARD, @link #SOFT, @link #WEAK

StaticBucketMap.StaticBucketMap(numBuckets) The number of buckets is never below 17, and is always an odd number (StaticBucketMap ensures this)

FileHandler.FileHandler(pattern, limit, count, append) The count must be at least 1.

(abstract) Clob.getSubString(pos, length) @param length the number of consecutive characters to be copied; the value for length must be 0 or greater

(abstract) Blob.getBytes(pos, length) @param length the number of consecutive bytes to be copied; the value for length must be 0 or greater

FileHandler.FileHandler(pattern, limit, count) The count must be at least 1

(abstract) Statement.execute(sql, autoGeneratedKeys) @param autoGeneratedKeys one of the following constants: Statement.RETURN_GENERATED_KEYS or Statement.NO_GENERATED_KEYS

BufferedWriter.write(s, off, len) If the value of the len parameter is negative then no characters are written. This is contrary to the specification of this method in the @linkplain java.io.Writer#write(java.lang.String,int,int) superclass, which requires that an @link IndexOutOfBoundsException be thrown.

BufferedInputStream.pos This value is always in the range 0 through count. If it is less than count, then buf[pos] is the next byte to be supplied as input; if it is equal to count, then the next read or skip operation will require more bytes to be read from the contained input stream.

Level.Level(name, value) In general clients of logging should use one of the constant Level objects such as SEVERE or FINEST.

Pack200.Packer.PROGRESS At a minimum, the unpacker must set progress to 0 at the beginning of a packing operation, and to 100 at the end.

SelectionProcessor.SelectionProcessor(document, tabWidth) @param tabWidth the tabulator width in space equivalents, must be ¿=0

Colors.palette(start, end, steps) @param steps the requested size, must be ¿ 0

Range.setLength(length) @param length the new length, must be ¿ 0

(abstract) ILineDiffer.revertSelection(line, nLines) @param nLines the number of lines to be reverted, must be ¿ 0.

Colors.computeHue(index) @param index the index of the color, must be ¿= 0

ContentProposalAdapter.setFilterStyle(filterStyle) @param filterStyle a constant indicating how keystrokes received in the proposal popup affect filtering of the proposals shown. FILTER_NONE specifies that no automatic filtering of the content proposal list will occur as keys are typed in the popup. FILTER_CHARACTER specifies that the content of the popup will be filtered by the most recently typed character. FILTER_CUMULATIVE is deprecated and no longer recommended. It specifies that the content of the popup will be filtered by a string containing all the characters typed since the popup has been open.

Hunk.line Must be in [0, numberOfLines] note the inclusive end; there may be a hunk with line == numberOfLines to describe deleted lines at then end of the document.

PreferenceDialog.setMessage(newMessage, newType) The valid message types are one of NONE, INFORMATION,WARNING, or ERROR.

MessageDialog.MessageDialog(parentShell, dialogTitle, dialogTitleImage, dialogMessage, dialogImageType, dialogButtonLabels, defaultIndex) @param dialogImageType one of the following values:

MessageDialog.NONE for a dialog with no image MessageDialog.ERROR for a dialog with an error image MessageDialog.INFORMATION for a dialog with an information image MessageDialog.QUESTION for a dialog with a question image MessageDialog.WARNING for a dialog with a warning image

MessageDialogWithToggle.open(kind, parent, title, message, toggleMessage, toggleState, store, key, style) @param kind the kind of dialog to open, one of @link #ERROR, @link #INFORMATION, @link #QUESTION, @link #WARNING, @link #CONFIRM, or #QUESTION_WITH_CANCEL.

BigInteger.BigInteger(numBits, rnd) @throws IllegalArgumentException numBits is negative.

(abstract) List.subList(fromIndex, toIndex) @throws IndexOutOfBoundsException for an illegal endpoint index value (fromIndex ¡ 0 —— toIndex ¿ size —— fromIndex ¿ toIndex)

Date.UTC(year, month, date, hrs, min, sec) @param month the month between 0-11. @param date the day of the month between 1-31. @param hrs the hours between 0-23. @param min the minutes between 0-59. @param sec the seconds between 0-59.

Object.wait(timeout, nanos) @param nanos additional time, in nanoseconds range 0-999999.

Thread.join(millis, nanos) @param nanos 0-999999 additional nanoseconds to wait. @exception IllegalArgumentException if the value of millis is negative the value of nanos is not in the range 0-999999.

MessageDialog.open(kind, parent, title, message, style) @param kind the kind of dialog to open, one of @link #ERROR, @link #INFORMATION, @link #QUESTION, @link #WARNING, @link #CONFIRM, or @link #QUESTION_WITH_CANCEL.

LinkedModeUI.setCyclingMode(mode) Sets the cycling mode to either of CYCLING_ALWAYS, CYCLING_NEVER, or CYCLING_WHEN_NO_PARENT, which is the default.

InetSocketAddress.createUnresolved(host, port) A valid port value is between 0 and 65535.

Date.setMonth(month) @param month the month value between 0-11

InetSocketAddress.InetSocketAddress(hostname, port) A valid port value is between 0 and 65535.

1.4 String Format Directive

LoggingPermission.LoggingPermission(name, actions) @param name Permission name. Must be ”control”.

BigDecimal.BigDecimal(val) The string must contain at least one digit in either the integer or the fraction.

ManagementPermission.ManagementPermission(name, actions) @param name Permission name. Must be either ”monitor” or ”control”.

ManagementPermission.ManagementPermission(name, actions) tive - Parameter: too vague): @param actions Must be either null or the empty string

HttpCookie.HttpCookie(name, value) The name must conform to RFC 2965.

URI.URI(str) An IPv6 address must be enclosed in square brackets (’[’ and ’]’) as specified by RFC 2732. The IPv6 address itself must parse according to RFC 2373.

class SocketPermission The host is expressed as a DNS name, as a numerical IP address, or as ”localhost” (for the local machine). The wildcard ”*” may be included once in a DNS name host specification. If it is included, it must be in the leftmost position, as in ”*.sun.com”.

class SocketPermission The format of the IPv6reference should follow that specified in RFC 2732: Format for Literal IPv6 Addresses in URLs

abstract class TimeZone The format is locale independent and digits must be taken from the Basic Latin block of the Unicode standard.

ActivationGroupDesc.CommandEnvironment.ActivationGroupDesc.CommandEnvironment(cmdpath, argv) The named program must be able to accept multiple -Dpropname=value options (as documented for the ”java” tool)

Byte.parseByte(s) The characters in the string must all be decimal digits, except that the first character may be an ASCII minus sign ’-’ (’\u002D’) to indicate a negative value

Byte.decode(nm) This sequence of characters must represent a positive value

Byte.parseByte(s, radix) The characters in the string must all be digits, of the specified radix

Integer.decode(nm) This sequence of characters must represent a positive value

Short.parseShort(s, radix) The characters in the string must all be digits, of the specified radix

SQLPermission.SQLPermission(name) @param name the name of this SQLPermission object, which must be setLog

SQLPermission.SQLPermission(name, actions) @param name the name of this SQLPermission object, which must be setLog

Integer.parseInt(s, radix) The characters in the string must all be digits of the specified radix (as determined by whether @link java.lang.Character#digit(char, int) returns a nonnegative value), except that the first character may be an ASCII minus sign ’-’ (’\u002D’) to indicate a negative value

Long.decode(nm) This sequence of characters must represent a positive value or a @link NumberFormatException will be thrown.

ManagementPermission.ManagementPermission(name) @param name Permission name. Must be either ”monitor” or ”control”.

Long.parseLong(s, radix) The characters in the string must all be digits of the specified radix (as determined by whether @link java.lang.Character#digit(char, int) returns a nonnegative value), except that the first character may be an ASCII minus sign ’-’ (’\u002D’) to indicate a negative value.

class Pattern In this class octal escapes must always begin with a zero.

Short.decode(nm) This sequence of characters must represent a positive value

URI.URI(scheme, authority, path, query, fragment) If a scheme is given then the path, if also given, must either be empty or begin with a slash character (’/’).

URI.URI(scheme, authority, path, query, fragment) If the authority contains a literal IPv6 address then the address must be enclosed in square brackets (’[’ and ’]’).

HttpCookie.setPath(uri) A cookie’s path must include the servlet that set the cookie, for example, /catalog, which makes the cookie visible to all directories on the server under /catalog.

ResourceBundle.Control.getNoFallbackControl(formats) The formats must be equal to one of @link Control#FORMAT_PROPERTIES, @link Control#FORMAT_CLASS or @link Control#FORMAT_DEFAULT.

URI.URI(scheme, userInfo, host, port, path, query, fragment) If a scheme is given then the path, if also given, must either be empty or begin with a slash character (’/’).

System.load(filename) The filename argument must be a complete path name.

Short.parseShort(s) The characters in the string must all be decimal digits, except that the first character may be an ASCII minus sign ’-’ (’\u002D’) to indicate a negative value.

Enum.valueOf(enumType, name) The name must match exactly an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)

File.createTempFile(prefix, suffix) The prefix string to be used in generating the file’s name; must be at least three characters long

File.createTempFile(prefix, suffix, directory) The prefix argument must be at least three characters long.

class MessageFormat An UnquotedPattern can contain arbitrary characters except single quotes, but curly braces within it must be balanced

Integer.parseInt(s) The characters in the string must all be decimal digits, except that the first character may be an ASCII minus sign '-' ('\u002D') to indicate a negative value.

Runtime.load(filename) The filename argument must be a complete path name, (for example Runtime.getRuntime().load("/home/avh/lib/libX11.so");).

class RuleBasedCollator Either the text-argument must already be present in the sequence, or some initial substring of the text-argument must be present. (e.g. "a ¡ b & ae ¡ e" is valid since "a" is present in the sequence before "ae" is reset). In this latter case, "ae" is not entered and treated as a single character; instead, "e" is sorted as if it were expanded to two characters: "a" followed by an "e". This difference appears in natural languages: in traditional Spanish "ch" is treated as though it contracts to a single character (expressed as "c ¡ ch ¡ d"), while in traditional German a-umlaut is treated as though it expanded to two characters (expressed as "a,A ¡ b,B ... &ae;\u00e3&AE;\u00c3"). [\u00e3 and \u00c3 are, of course, the escape sequences for a-umlaut.] Ignorable Characters For ignorable characters, the first rule must start with a relation

Date.parse(s) It also understands the continental U.S. time-zone abbreviations, but for general use, a time-zone offset should be used: "Sat, 12 Aug 1995 13:30:00 GMT+0430" (4 hours, 30 minutes west of the Greenwich meridian).

ResourceBundle.getBundle(baseName, locale, loader) The baseName argument should be a fully qualified class name.

abstract class ContentHandler The class name must be of the form: package-prefix.major.minor e.g. YoyoDyne.experimental.text.plain

HttpCookie.parse(header) @param header a String specifying the set-cookie header. The header should start with "set-cookie", or "set-cookie2" token; or it should have no leading token at all.

URLEncoder.encode(s, enc) The World Wide Web Consortium Recommendation states that UTF-8 should be used. Not doing so may introduce incompatibilites.

URI.create(str) @throws IllegalArgumentException If the given string violates RFC 2396

URL.URL(protocol, host, port, file) If IPv6 literal address is used, it should be enclosed in square brackets ('[' and ']'), as specified by RFC 2732; However, the literal IPv6 address format defined in RFC 2373: IP Version 6 Addressing Architecture is also accepted.

Long.parseLong(s) The characters in the string must all be decimal digits, except that the first character may be an ASCII minus sign '-' (\u002D') to indicate a negative value.

HttpCookie.setValue(newValue) With Version 0 cookies, values should not contain white space, brackets, parentheses, equals signs, commas, double quotes, slashes, question marks, at signs, colons, and semicolons. Empty values may not behave the same way on all browsers.

URLDecoder.decode(s, enc) Note: The World Wide Web Consortium Recommendation states that UTF-8 should be used. Not doing so may introduce incompatibilites.

Security.getAlgorithmProperty(algName, propName) The algorithm name should be a standard name. See Appendix A in the Java Cryptography Architecture API Specification & Reference for information about standard algorithm names.

ManagementFactory.newPlatformMXBeanProxy(connection, mxbeanName, mxbeanInterface) @param mxbeanName the name of a platform MXBean within connection to forward to. mxbeanName must be in the format of @link ObjectName ObjectName.

ResourceBundle.Control.getFormats(baseName) Strings starting with "java." are reserved for future extensions and must not be used by application-defined formats.

class Formatter If the conversion is 'a' or 'A', then the precision must not be specified.

class RuleBasedCollator For ignorable characters, the first rule must start with a relation (the examples we have used above are really fragments; "a ¡ b" really should be "¡ a ¡ b").

(abstract) PreparedStatement.setUnicodeStream(parameterIndex, x, length) The byte format of the Unicode stream must be a Java UTF-8, as defined in the Java Virtual Machine Specification.

Security.getProviders(filter) The selection criterion must be specified in one of the following two formats:

CertPathValidator.getInstance(algorithm, provider) The specified provider must be registered in the security provider list.

CertPathBuilder.getInstance(algorithm, provider) The specified provider must be registered in the security provider list.

AlgorithmParameters.getInstance(algorithm, provider) The specified provider must be registered in the security provider list.

CertificateFactory.getInstance(type, provider) The specified provider must be registered in the security provider list.

Signature.getInstance(algorithm, provider) The specified provider must be registered in the security provider list.

MessageDigest.getInstance(algorithm, provider) The specified provider must be registered in the security provider list.

KeyStore.getInstance(type, provider) The specified provider must be registered in the security provider list.

KeyPairGenerator.getInstance(algorithm, provider) The specified provider must be registered in the security provider list.

KeyFactory.getInstance(algorithm, provider) The specified provider must be registered in the security provider list.

SecureRandom.getInstance(algorithm, provider) The specified provider must be registered in the security provider list.

Policy.getInstance(type, params, provider) The specified provider must be registered in the provider list.

AlgorithmParameterGenerator.getInstance(algorithm, provider) The specified provider must be registered in the security provider list.

CertStore.getInstance(type, params, provider) The specified provider must be registered in the security provider list.

abstract class LocaleServiceProvider The file should contain a list of fully-qualified concrete provider class names, one per line.

Logger.getLogger(name, resourceBundleName) @param name A name for the logger. This should be a dot-separated name and should normally be based on the package name or class name of the subsystem, such as java.net or javax.swing

class Formatter Note that there must be at least one nonzero digit in a subnormal significand.

Logger.Logger(name, resourceBundleName) @param name A name for the logger. This should be a dot-separated name and should normally be based on the package name or class name of the subsystem, such as java.net or javax.swing.

class PropertyResourceBundle In that case, characters that cannot be represented in ISO-8859-1 encoding must be represented by Unicode Escapes, whereas the other constructor which takes a Reader does not have that limitation.

Float.valueOf(s) The rest of s should constitute a FloatValue as described by the lexical syntax rules: FloatValue: Signopt NaN Signopt Infinity Signopt FloatingPointLiteral Signopt HexFloatingPointLiteral SignedInteger HexFloatingPointLiteral: HexSignificand BinaryExponent FloatTypeSuffixopt HexSignificand: HexNumeral HexNumeral . 0x HexDigitsopt . HexDigits 0X HexDigitsopt . HexDigits BinaryExponent: BinaryExponentIndicator SignedInteger BinaryExponentIndicator: p P where Sign, FloatingPointLiteral, HexNumeral, HexDigits, SignedInteger and FloatTypeSuffix are as defined in the lexical structure sections of the of the Java Language Specification.

(abstract) CallableStatement.registerOutParameter(parameterIndex, sqlType, typeName) For a user-defined parameter, the fully-qualified SQL type name of the parameter should also be given, while a REF parameter requires that the fully-qualified type name of the referenced type be given.

Bidi.createLineBidi(lineStart, lineLimit) This call is not required if the entire paragraph fits on one line.

(abstract) PreparedStatement.setNull(parameterIndex, sqlType, typeName) To be portable, applications must give the SQL type code and the fully-qualified SQL type name when specifying a NULL user-defined or REF parameter. In the case of a user-defined type the name is the type name of the parameter itself.

Security.getProviders(filter) The key for any entry in such a map must be in one of the following two formats: * . The cryptographic service name must not contain any dots. The value associated with the key must be an empty string. A provider satisfies this selection criterion iff the provider implements the specified algorithm or type for the specified cryptographic service. * . The cryptographic service name must not contain any dots. There must be one or more space charaters between the and the . The value associated with the key must be a non-empty string. A provider satisfies this selection criterion iff the provider implements the specified algorithm or type for the specified cryptographic service and its implementation meets the constraint expressed by the specified attribute name/value pair.

(abstract) CertificateFactorySpi.engineGenerateCertificate(inStream) In the case of a certificate factory for X.509 certificates, the certificate provided in inStream must be DER-encoded and may be supplied in binary or printable (Base64) encoding. If the certificate is provided in Base64 encoding, it must be bounded at the beginning by ——BEGIN CERTIFICATE——, and must be bounded at the end by ——END CERTIFICATE——.

CertificateFactory.generateCertificate(inStream) In the case of a certificate factory for X.509 certificates, the certificate provided in inStream must be DER-encoded and may be supplied in binary or printable (Base64) encoding. If the certificate is provided in Base64 encoding, it must be bounded at the beginning by ——BEGIN CERTIFICATE——, and must be bounded at the end by ——END CERTIFICATE——.

abstract class Signature The algorithm name must be specified, as there is no default.

X509CertSelector.addSubjectAlternativeName(type, name) This byte array should contain the DER encoded name, as it would appear in the GeneralName structure defined in RFC 2459 and X.509. The encoded byte array should only contain the encoded value of the name, and should not include the tag associated with the name in the GeneralName structure. The ASN.1 definition of this structure appears below.

X509CertSelector.addPathToName(type, name) The name is provided as a byte array. This byte array should contain the DER encoded name, as it would appear in the GeneralName structure defined in RFC 2459 and X.509.

X509CRLSelector.setIssuerNames(names) If a name is specified as a byte array, it should contain a single DER encoded distinguished name, as defined in X.501.

(abstract) CallableStatement.registerOutParameter(parameterIndex, sqlType, typeName) To be portable, however, applications should always provide these values for user-defined and REF parameters.

Properties.load(reader) Note that a comment line cannot be extended in this manner; every natural line that is a comment must have its own comment indicator, as described below.

Properties.load(reader) Note that it is not sufficient to only examine the character preceding a line terminator sequence to decide if the line terminator is escaped; there must be an odd number of contiguous backslashes for the line terminator to be escaped.

Logger.getLogger(name) @param name A name for the logger. This should be a dot-separated name and should normally be based on the package name or class name of the subsystem, such as java.net or javax.swing

class SignedObject The algorithm name must be specified, as there is no default.

X509CertSelector.addSubjectAlternativeName(type, name) This byte array should contain the DER encoded name, as it would appear in the GeneralName structure defined in RFC 2459 and X.509. The encoded byte array should only contain the encoded value of the name, and should not include the tag associated with the name in the GeneralName structure. The ASN.1 definition of this structure appears below.

X509CertSelector.addPathToName(type, name) The name is provided in string format. RFC 822, DNS, and URI names use the well-established string formats for those types (subject to the restrictions included in RFC 2459). IPv4 address names are supplied using dotted quad notation. OID address names are represented as a series of nonnegative integers separated by periods. And directory names (distinguished names) are supplied in RFC 2253 format. No standard string format is defined for otherNames, X.400 names, EDI party names, IPv6 address names, or any other type of names. They should be specified using the @link #addPathToName(int type, byte [] name) addPathToName(int type, byte [] name) method.

X509CertSelector.setAuthorityKeyIdentifier(authorityKeyID) If authorityKeyID is not null, it should contain a single DER encoded value corresponding to the contents of the extension value (not including the object identifier, criticality setting, and encapsulating OCTET STRING) for an AuthorityKeyIdentifier extension.

(abstract) CallableStatement.setNull(parameterName, sqlType, typeName) To be portable, applications must give the SQL type code and the fully-qualified SQL type name when specifying a NULL user-defined or REF parameter.

X509CertSelector.setSubjectPublicKey(key) If key is not null, it should contain a single DER encoded SubjectPublicKeyInfo structure, as defined in X.509.

(abstract) CallableStatement.registerOutParameter(parameterName, sqlType, typeName) For a user-named parameter the fully-qualified SQL type name of the parameter should also be given, while a REF parameter requires that the fully-qualified type name of the referenced type be given. A JDBC driver that does not need the type code and type name information may ignore it. To be portable, however, applications should always provide these values for user-named and REF parameters.

X509CertSelector.addSubjectAlternativeName(type, name) hed string formats for those types (subject to the restrictions included in RFC 2459). IPv4 address names are supplied using dotted quad notation. OID address names are represented as a series of nonnegative integers separated by periods. And directory names (distinguished names) are supplied in RFC 2253 format. No standard string format is defined for otherNames, X.400 names, EDI party names, IPv6 address names, or any other type of names. They should be specified using the @link #addSubjectAlternativeName(int type, byte [] name) addSubjectAlternativeName(int type, byte [] name) method.

(abstract) Preferences.getByteArray(key, def) Valid strings are Base64 encoded binary data, as defined in RFC 2045, Section 6.8, with one minor change: the string must consist solely of characters from the Base64 Alphabet; no newline characters or extraneous characters are permitted.

X509CertSelector.setIssuer(issuerDN) If issuerDN is not null, it should contain a distinguished name, in RFC 2253 format.

TrustAnchor.TrustAnchor(trustedCert, nameConstraints) The name constraints are specified as a byte array. This byte array should contain the DER encoded form of the name constraints, as they would appear in the NameConstraints structure defined in RFC 2459 and X.509.

TimeZone.getTimeZone(ID) @param ID the ID for a TimeZone, either an abbreviation such as ”PST”, a full name such as ”America/Los_Angeles”, or a custom ID such as ”GMT-8:00”. Note that the support of abbreviations is for JDK 1.1.x compatibility only and full names should be used.

Double.valueOf(s) The rest of s should constitute a FloatValue as described by the lexical syntax rules: FloatValue: Signopt NaN Signopt Infinity Signopt FloatingPointLiteral Signopt HexFloatingPointLiteral SignedInteger HexFloatingPointLiteral: HexSignificand BinaryExponent FloatTypeSuffixopt HexSignificand: HexNumeral HexNumeral . 0x HexDigitsopt . HexDigits 0X HexDigitsopt . HexDigits BinaryExponent: BinaryExponentIndicator SignedInteger BinaryExponentIndicator: p P where Sign, FloatingPointLiteral, HexNumeral, HexDigits, SignedInteger and FloatTypeSuffix are as defined in the lexical structure sections of the of the Java Language Specification.

RandomAccessFile.RandomAccessFile(file, mode) The mode argument specifies the access mode in which the file is to be opened. The permitted values and their meanings are:

Value

Meaning ”r” Open for reading only. Invoking any of the write methods of the resulting object will cause an @link java.io.IOException to be thrown. ”rw” Open for reading and writing. If the file does not already exist then an attempt will be made to create it. ”rws” Open for reading and writing, as with ”rw”, and also require that every update to the file’s content or metadata be written synchronously to the underlying storage device. ”rwd” Open for reading and writing, as with ”rw”, and also require that every update to the file’s content be written synchronously to the underlying storage device.

URLConnection.setRequestProperty(key, value) HTTP requires all request properties which can legally have multiple instances with the same key to use a comma-seperated list syntax which enables multiple properties to be appended into a single property.

class IDN The behavior of aforementioned conversion process can be adjusted by various flags:

If the ALLOW_UNASSIGNED flag is used, the domain name string to be converted can contain code points that are unassigned in Unicode 3.2, which is the Unicode version on which IDN conversion is based. If the flag is not used, the presence of such unassigned code points is treated as an error. If the USE_STD3_ASCII_RULES flag is used, ASCII strings are checked against RFC 1122 and RFC 1123. It is an error if they don’t meet the requirements.

Pack200.newPacker() If the system property java.util.jar.Pack200.Packer is defined, then the value is taken to be the fully-qualified name of a concrete implementation class, which must implement Packer.

ResourceBundle.getBundle(baseName, targetLocale, loader, control) Other strings starting with ”java.” are reserved for future extensions and must not be used for application-defined formats.

abstract class ResourceBundle The family should have a default resource bundle which simply has the same name as its family - ”MyResources” - and will be used as the bundle of last resort if a specific locale is not supported.

(abstract) IPartitionTokenScanner.setPartialRange(document, offset, length, contentType, partitionOffset) The range may no only contain complete partitions but starts at the beginning of a line in the middle of a partition of the given content type. This requires that a partition delimiter can not contain a line delimiter.

class XMLFormatter The XMLFormatter can be used with arbitrary character encodings, but it is recommended that it normally be used with UTF-8.

class URLEncoder The recommended encoding scheme to use is UTF-8. However, for compatibility reasons, if an encoding is not specified, then the default encoding of the platform is used.

TrustAnchor.TrustAnchor(caName, pubKey, nameConstraints) @param nameConstraints a byte array containing the ASN.1 DER encoding of a NameConstraints extension to be used for checking name constraints. Only the value of the extension is included, not the OID or criticality flag.

class Pattern This class also accepts dangling brackets but is strict about dangling metacharacters like +, ? and *, and will throw a @link PatternSyntaxException if it encounters them.

Character.isJavaIdentifierStart(ch) A character may start a Java identifier if and only if one of the following conditions is true:

@link #isLetter(char) isLetter(ch) returns true @link #getType(char) getType(ch) returns LETTER_NUMBER ch is a currency symbol (such as ”$”) ch is a connecting punctuation character (such as ”_”).

URLClassLoader.URLClassLoader(urls, parent) Any URL that ends with a ’/’ is assumed to refer to a directory. Otherwise, the URL is assumed to refer to a JAR file which will be downloaded and opened as needed.

class URLClassLoader Any URL that ends with a ’/’ is assumed to refer to a directory. Otherwise, the URL is assumed to refer to a JAR file which will be opened as needed.

IDN.toUnicode(input, flag) This method can handle both label and entire domain name, by assuming that labels in a domain name are always separated by dots.

class URLDecoder It is assumed that all characters in the encoded string are one of the following: ”a” through ”z”, ”A” through ”Z”, ”0” through ”9”, and ”-”, ”_”, ”.”, and ”*”. The character ”%” is allowed but is interpreted as the start of a special escaped sequence.

class KeyStore Whether aliases are case sensitive is implementation dependent. In order to avoid problems, it is recommended not to use aliases in a KeyStore that only differ in case.

class URL A URL can optionally specify a ”port”, which is the port number to which the TCP connection is made on the remote host machine. If the port is not specified, the default port for the protocol is used instead. For example, the default port for http is 80. An alternative port could be specified as:

http://www.socs.uts.edu.au:80/MosaicDocs-old/url-primer.html

PrintStream.format(format, args) @throws IllegalFormatException If a format string contains an illegal syntax, a format specifier that is incompatible with the given arguments, insufficient arguments given the format string, or other illegal conditions. For specification of all possible formatting errors, see the Details section of the formatter class specification.

Calendar.setLenient(lenient) With lenient interpretation, a date such as ”February 942, 1996” will be treated as being equivalent to the 941st day after February 1, 1996. With strict (non-lenient) interpretation, such dates will cause an exception to be thrown.

PKCS8EncodedKeySpec.PKCS8EncodedKeySpec(encodedKey) @param encodedKey the key, which is assumed to be encoded according to the PKCS #8 standard.

X509EncodedKeySpec.X509EncodedKeySpec(encodedKey) @param encodedKey the key, which is assumed to be encoded according to the X.509 standard.

(abstract) CurrencyNameProvider.getSymbol(currencyCode, locale) @param currencyCode the ISO 4217 currency code, which consists of three upper-case letters between ’A’ (U+0041) and ’Z’ (U+005A)

1.5 Method Parameter Type Directive

ActivationGroup.createGroup(id, desc, incarnation) The group class specified by the ActivationGroupDesc must be a concrete subclass of ActivationGroup and have a public constructor that takes two arguments: the ActivationGroupID for the group and the MarshalledObject containing the group’s initialization data (obtained from the ActivationGroupDesc

PortableRemoteObject.toStub(obj) @param obj the server object for which a stub is required. must either be a subclass of PortableRemoteObject

TreeSet.TreeSet() All elements inserted into the set must implement the @link Comparable interface.

TreeSet.TreeSet(c) All elements inserted into the set must implement the @link Comparable interface

TreeMap.TreeMap() all such keys must be mutually comparable: k1.compareTo(k2) must not throw a ClassCastException for any keys k1 and k2 in the map.

URI.compareTo(that) Compares this URI to another object, which must be a URI.

Timestamp.compareTo(o) Compares this Timestamp object to the given Date, which must be a Timestamp object.

SequenceInputStream.SequenceInputStream(e) created SequenceInputStream by remembering the argument, which must be an Enumeration that produces objects whose run-time type is InputStream

DateFormat.format(obj, toAppendTo, fieldPosition) param obj must be a Number or a Date.

(abstract) CertPathValidatorSpi.engineValidate(certPath, params) The CertPath specified must be of a type that is supported by the validation algorithm, otherwise an InvalidAlgorithmParameterException will be thrown.

CertPathValidator.validate(certPath, params) The CertPath specified must be of a type that is supported by the validation algorithm, otherwise an InvalidAlgorithmParameterException will be thrown.

Collections.min(coll, comp) All elements in the collection must be mutually comparable by the specified comparator (that is, comp.compare(e1, e2) must not throw a ClassCastException for any elements e1 and e2 in the collection).

Collections.max(coll, comp) All elements in the collection must be mutually comparable by the specified comparator (that is, comp.compare(e1, e2) must not throw a ClassCastException for any elements e1 and e2 in the collection).

Collections.sort(list, c) All elements in the list must be mutually comparable using the specified comparator (that is, c.compare(e1, e2) must not throw a ClassCastException for any elements e1 and e2 in the list).

Arrays.sort(a, fromIndex, toIndex) Furthermore, all elements in this range must be mutually comparable (that is, e1.compareTo(e2) must not throw a ClassCastException for any elements e1 and e2 in the array).

Arrays.sort(a, fromIndex, toIndex, c) All elements in the range must be mutually comparable by the specified comparator (that is, c.compare(e1, e2) must not throw a ClassCastException for any elements e1 and e2 in the range).

interface SortedSet Furthermore, all such elements must be mutually comparable: e1.compareTo(e2) (or comparator.compare(e1, e2)) must not throw a ClassCastException for any elements e1 and e2 in the sorted set.

interface SortedMap Furthermore, all such keys must be mutually comparable: k1.compareTo(k2) (or comparator.compare(k1, k2)) must not throw a ClassCastException for any keys k1 and k2 in the sorted map.

Collections.max(coll) Furthermore, all elements in the collection must be mutually comparable (that is, e1.compareTo(e2) must not throw a ClassCastException for any elements e1 and e2 in the collection).

Arrays.sort(a) Furthermore, all elements in the array must be mutually comparable (that is, e1.compareTo(e2) must not throw a ClassCastException for any elements e1 and e2 in the array).

TreeSet.TreeSet(comparator) All elements inserted into the set must be mutually comparable by the specified comparator: comparator.compare(e1, e2) must not throw a ClassCastException for any elements e1 and e2 in the set

TreeMap.TreeMap(comparator) All keys inserted into the map must be mutually comparable by the given comparator: comparator.compare(k1, k2) must not throw a ClassCastException for any keys k1 and k2 in the map

PreferenceNode.PreferenceNode(id, label, image, className) @param className the class name of the preference page; this class must implement IPreferencePage

PreferenceManager.addToRoot(node) @param node the node to add, which must implement IPreferenceNode

MultiPassContentFormatter.setMasterStrategy(strategy) @param strategy The master formatting strategy, must implement @link IFormattingStrategyExtension

MultiPassContentFormatter.setSlaveStrategy(strategy, type) @param type The content type to register this strategy with, must implement @link IFormattingStrategyExtension

ContentAssistantFacade.ContentAssistantFacade(contentAssistant) @param contentAssistant the content assistant which implements @link IContentAssistantExtension2 and @link IContentAssistantExtension4

class Pattern A regular expression, specified as a string, must first be compiled into an instance of this class.

abstract class EnumSet All of the elements in an enum set must come from a single enum type that is specified, explicitly or implicitly, when the set is created.

class EnumMap All of the keys in an enum map must come from a single enum type that is specified, explicitly or implicitly, when the map is created.

abstract class ListResourceBundle The first element of each pair is the key, which must be a String, and the second element is the value associated with that key.

TreeBidiMap.putAll(map) All keys and values must implement Comparable

TreeBidiMap.get(key) The key must implement Comparable.

TreeBidiMap.nextKey(key) The key must implement Comparable.

TreeBidiMap.containsValue(value) The value must implement Comparable

TreeBidiMap.previousKey(key) The key must implement Comparable.

MultiKeyMap.putAll(mapToCopy) Copies all of the keys and values from the specified map to this map. Each key must be non-null and a MultiKey object.

TreeBidiMap.containsKey(key) The key must implement Comparable.

CompositeSet.addComposited(c) @param c Must implement Set

TreeBidiMap.getKey(value) The value must implement Comparable.

TreeBidiMap.put(key, value) Both key and value must implement Comparable.

TreeBidiMap.remove(key) The key must implement Comparable.

TreeBidiMap.removeValue(value) The value must implement Comparable.

MarshalledObject.MarshalledObject(obj) @param obj the object to be serialized (must be serializable)

class ActivationGroupDesc The group's class must be a concrete subclass of ActivationGroup.

abstract class ActivationGroup The createGroup method has two requirements on the group to be created: 1) the group must be a concrete subclass of ActivationGroup, and 2) the group must have a constructor that takes two arguments:
the group's ActivationGroupID, and the group's initialization data (in a java.rmi.MarshalledObject)

TreeMap.TreeMap() All keys inserted into the map must implement the @link Comparable interface.

FactoryUtils.instantiateFactory(classToInstantiate, paramTypes, args) @throws IllegalArgumentException if the constructor doesn't exist

CollectionUtils.collect(inputCollection, transformer, outputCollection) @param the output type of the transformer - this extends O.

CollectionUtils.collect(inputIterator, transformer, outputCollection) @param the output type of the transformer - this extends O.

ObjectOutputStream.writeObject(obj) Exceptions are thrown for problems with the OutputStream and for classes that should not be serialized. All exceptions are fatal to the OutputStream, which is left in an indeterminate state, and it is up to the caller to ignore or recover the stream state. @throws InvalidClassException Something is wrong with a class used by serialization. @throws NotSerializableException Some object to be serialized does not implement the java.io.Serializable interface.

ObjectInputStream.readObject() Exceptions are thrown for problems with the InputStream and for classes that should not be deserialized. All exceptions are fatal to the InputStream and leave it in an indeterminate state; it is up to the caller to ignore or recover the stream state. @throws ClassNotFoundException Class of a serialized object cannot be found. @throws InvalidClassException Something is wrong with a class used by serialization. @throws StreamCorruptedException Control information in the stream is inconsistent. @throws OptionalDataException Primitive data was found in the stream instead of objects.

FormattingContextProperties.CONTEXT_PREFERENCES The property must implement java.util#Map.

FormattingContextProperties.CONTEXT_PARTITION The property must implement org.eclipse.jface.text#TypedPosition.

FormattingContextProperties.CONTEXT_REGION The property must implement org.eclipse.jface.text#IRegion.

FormattingContextProperties.CONTEXT_MEDIUM The property must implement org.eclipse.jface.text#IDocument.

Collections.sort(list) All elements in the list must implement the Comparable interface

FormattingContextProperties.CONTEXT_DOCUMENT The property must implement java.lang#Boolean.

(abstract) PreparedStatement.setObject(parameterIndex, x) The second parameter must be of type Object; therefore, the java.lang equivalent objects should be used for built-in types.

(abstract) PreparedStatement.setObject(parameterIndex, x) If the object is of a class implementing the interface SQLData, the JDBC driver should call the method SQLData.writeSQL to write it to the SQL data stream. If, on the other hand, the object is of a class implementing Ref, Blob, Clob, NClob, Struct, java.net.URL, RowId, SQLXML or Array, the driver should pass it to the database as a value of the corresponding SQL type.

(abstract) MultiMap.remove(key, item) @throws ClassCastException if the key or value is of an invalid type

(abstract) MultiMap.put(key, value) @throws ClassCastException if the key or value is of an invalid type

Pack200.newUnpacker() If the system property java.util.jar.Pack200.Unpacker is defined, then the value is taken to be the fully-qualified name of a concrete implementation class, which must implement Unpacker.

ClassLoader.getSystemClassLoader() The class is loaded using the default system class loader and must define a public constructor that takes a single parameter of type ClassLoader which is used as the delegation parent.

class ParsePosition The parseObject method in the various Format classes requires a ParsePosition object as an argument.

class FieldPosition One version of the format method in the various Format classes requires a FieldPosition object as an argument.

(abstract) PreparedStatement.setObject(parameterIndex, x, targetSqlType, scaleOrLength) The second argument must be an object type; for integral values, the java.lang equivalent objects should be used.

(abstract) CallableStatement.setObject(parameterName, x, targetSqlType, scale) The second argument must be an object type; for integral values, the java.lang equivalent objects should be used

MapUtils.multiValueMap(map, collectionClass) @param collectionClass the type of collections to return from the map (must contain public no-arg constructor and extend Collection)

interface SortedMap All keys inserted into a sorted map must implement the Comparable interface (or be accepted by the specified comparator).

(abstract) MultiMap.get(key) @throws ClassCastException if the key is of an invalid type

(abstract) CallableStatement.setObject(parameterName, x) The second parameter must be of type Object; therefore, the java.lang equivalent objects should be used for built-in types.

TreeMap.TreeMap(m) All keys inserted into the new map must implement the @link Comparable interface

Arrays.sort(a) All elements in the array must implement the @link Comparable interface

(abstract) SocketOptions.setOption(optID, value) For an option that has a notion of on and off, and requires a non-boolean parameter, setting its value to anything other than Boolean(false) implicitly enables it.

Arrays.sort(a, fromIndex, toIndex) All elements in this range must implement the @link Comparable interface

interface SortedSet All elements inserted into a sorted set must implement the Comparable interface (or be accepted by the specified comparator).

Collections.max(coll) All elements in the collection must implement the Comparable interface.

class Hashtable To successfully store and retrieve objects from a hashtable, the objects used as keys must implement the hashCode method and the equals method.

DelayQueue.toArray(a) @throws ArrayStoreException if the runtime type of the specified array is not a supertype of the runtime type of every element in this queue

LinkedList.toArray(a) @throws ArrayStoreException if the runtime type of the specified array is not a supertype of the runtime type of every element in this list

LinkedBlockingQueue.toArray(a) @throws ArrayStoreException if the runtime type of the specified array is not a supertype of the runtime type of every element in this queue

PriorityBlockingQueue.toArray(a) @throws ArrayStoreException if the runtime type of the specified array is not a supertype of the runtime type of every element in this queue

ArrayBlockingQueue.toArray(a) @throws ArrayStoreException if the runtime type of the specified array is not a supertype of the runtime type of every element in this queue

CopyOnWriteArrayList.toArray(a) @throws ArrayStoreException if the runtime type of the specified array is not a supertype of the runtime type of every element in this list

Vector.toArray(a) @throws ArrayStoreException if the runtime type of a is not a supertype of the runtime type of every element in this Vector

Collections.checkedList(list, type) Any attempt to insert an element of the wrong type will result in an immediate ClassCastException.

(abstract) NavigableMap.lowerKey(key) @throws ClassCastException if the specified key cannot be compared with the keys currently in the map

1.6 Method Parameter Correlation Directive

FileLock.FileLock(channel, position, size, shared) @param size must be non-negative, and the sum position + size must be non-negative

TextMeasurer.getLayout(start, limit) @param limit the index after the last character. Must be greater than start

DatagramPacket.DatagramPacket(buf, length, address, port) The length argument must be less than or equal to buf.length.

DatagramPacket.DatagramPacket(buf, length, address) The length argument must be less than or equal to buf.length.

DatagramPacket.setLength(length) The length must be lesser or equal to the offset plus the length of the packet's buffer.

Arrays.copyOfRange(original, from, to, newType) The initial index of the range (from) must lie between zero and original.length, inclusive.

Arrays.copyOfRange(original, from, to, newType) The final index of the range (to), which must be greater than or equal to from, may be greater than original.length, in which case null is placed in all elements of the copy whose index is greater than or equal to original.length - from.

TextMeasurer.insertChar(newParagraph, insertPos) @param insertPos the position in the text where the character was inserted. Must not be less than the start of newParagraph, and must be less than the end of newParagraph.

RescaleOp.filter(src, dst) The source and destination must have the same number of bands.

ColorConvertOp.ColorConvertOp(srcCspace, dstCspace, hints) In that case, the number of bands in the source Raster must match the number of components in srcCspace, and the number of bands in the destination Raster must match the number of components in dstCspace

KeyStore.setKeyEntry(alias, key, chain) If the protected key is of type java.security.PrivateKey, it must be accompanied by a certificate chain certifying the corresponding public key

KeyStore.setKeyEntry(alias, key, chain) If the underlying keystore implementation is of type jks, key must be encoded as an EncryptedPrivateKeyInfo as defined in the PKCS #8 standard.

DatagramPacket.DatagramPacket(buf, offset, length) The length argument must be less than or equal to buf.length

(abstract) FocusTraversalPolicy.getComponentBefore(aContainer, aComponent) aContainer must be a focus cycle root of aComponent or a focus traversal policy provider.

DatagramPacket.DatagramPacket(buf, offset, length, address) The length argument must be less than or equal to buf.length

ResourceBundle.Control.getControl(formats) The formats must be equal to one of @link Control#FORMAT_PROPERTIES, @link Control#FORMAT_CLASS or @link Control#FORMAT_DEFAULT.

Collections.copy(dest, src) The destination list must be at least as long as the source list.

DatagramPacket.DatagramPacket(buf, length) The length argument must be less than or equal to buf.length.

Long.parseLong(s, radix) The characters in the string must all be digits of the specified radix (as determined by whether @link java.lang.Character#digit(char, int) returns a nonnegative value), except that the first character may be an ASCII minus sign '-' ('\u002D') to indicate a negative value.

(abstract) ResultSet.updateObject(columnIndex, x, scaleOrLength) If the second argument is an InputStream then the stream must contain the number of bytes specified by scaleOrLength.

(abstract) FocusTraversalPolicy.getComponentAfter(aContainer, aComponent) aContainer must be a focus cycle root of aComponent or a focus traversal policy provider

TextMeasurer.deleteChar(newParagraph, deletePos) @param deletePos the position in the text where the character was removed. Must not be less than the start of newParagraph, and must not be greater than the end of newParagraph.

DatagramPacket.DatagramPacket(buf, offset, length, address, port) The length argument must be less than or equal to buf.length.

Arrays.sort(a, c) All elements in the array must be mutually comparable by the specified comparator (that is, c.compare(e1, e2) must not throw a ClassCastException for any elements e1 and e2 in the array)

Arrays.copyOfRange(original, from, to) The final index of the range (to), which must be greater than or equal to from, may be greater than original.length, in which case 0f is placed in all elements of the copy whose index is greater than or equal to original.length - from.

Arrays.copyOfRange(original, from, to) The initial index of the range (from) must lie between zero and original.length, inclusive.

Arrays.copyOfRange(original, from, to) The initial index of the range (from) must lie between zero and original.length, inclusive.

Arrays.copyOfRange(original, from, to) The final index of the range (to), which must be greater than or equal to from, may be greater than original.length, in which case 0L is placed in all elements of the copy whose index is greater than or equal to original.length - from.

Arrays.copyOfRange(original, from, to) The final index of the range (to), which must be greater than or equal to from, may be greater than original.length, in which case '\\u000' is placed in all elements of the copy whose index is greater than or equal to original.length - from.

Arrays.copyOfRange(original, from, to) The initial index of the range (from) must lie between zero and original.length, inclusive.

(abstract) DatabaseMetaData.getTables(catalog, schemaPattern, tableNamePattern, types) @param types a list of table types, which must be from the list of table types returned from @link #getTableTypes

class MessageFormat A SubformatPattern must be a valid pattern string for the Format subclass used.

Arrays.copyOfRange(original, from, to) The initial index of the range (from) must lie between zero and original.length, inclusive.

Arrays.copyOfRange(original, from, to) The final index of the range (to), which must be greater than or equal to from, may be greater than original.length, in which case null is placed in all elements of the copy whose index is greater than or equal to original.length - from.

Arrays.copyOfRange(original, from, to) The initial index of the range (from) must lie between zero and original.length, inclusive.

Arrays.copyOfRange(original, from, to) The final index of the range (to), which must be greater than or equal to from, may be greater than original.length, in which case (byte)0 is placed in all elements of the copy whose index is greater than or equal to original.length - from.

class ConcurrentHashMap Ideally, you should choose a value to accommodate as many threads as will ever concurrently modify the table.

IteratorUtils.arrayListIterator(array, start, end) @throws IllegalArgumentException if end is before start

IteratorUtils.arrayIterator(array, start, end) @throws IllegalArgumentException if end is before start

IteratorUtils.arrayIterator(array, start, end) @throws IllegalArgumentException if end is before start

ArrayListIterator.ArrayListIterator(array, startIndex, endIndex) @throws IllegalArgumentException if end index is before the start

ObjectArrayIterator.ObjectArrayIterator(array, start, end) @throws IllegalArgumentException if end index is before the start

class ChoiceFormat When creating a ChoiceFormat, you must specify an array of formats and an array of limits. The length of these arrays must be the same.

ClassLoader.defineClass(name, b, off, len, protectionDomain) If name is not null, it must be equal to the binary name of the class specified by the byte array "b", otherwise a @link NoClassDefFoundError will be thrown.

ClassLoader.defineClass(name, b, off, len, protectionDomain) You should always pass in the binary name of the class you are defining as well as the bytes.

X509CertSelector.setSubjectKeyIdentifier(subjectKeyID) If subjectKeyID is not null, it should contain a single DER encoded value corresponding to the contents of the extension value (not including the object identifier, criticality setting, and encapsulating OCTET STRING) for a SubjectKeyIdentifier extension.

SecureClassLoader.defineClass(name, b, off, len, cs) @param b the bytes that make up the class data. The bytes in positions off through off+len-1 should have the format of a valid class file as defined by the Java Virtual Machine Specification.

(abstract) CallableStatement.registerOutParameter(parameterName, sqlType) If the JDBC type expected to be returned to this output parameter is specific to this particular database, sqlType should be java.sql.Types.OTHER.

(abstract) CallableStatement.registerOutParameter(parameterIndex, sqlType) If the JDBC type expected to be returned to this output parameter is specific to this particular database, sqlType should be java.sql.Types.OTHER.

Arrays.binarySearch(a, fromIndex, toIndex, key) The range must be sorted (as by the @link #sort(double[], int, int) method) prior to making this call.

Arrays.binarySearch(a, fromIndex, toIndex, key) The range must be sorted (as by the @link #sort(float[], int, int) method) prior to making this call.

Collections.sort(list) Furthermore, all elements in the list must be mutually comparable (that is, e1.compareTo(e2) must not throw a ClassCastException for any elements e1 and e2 in the list).

Arrays.copyOfRange(original, from, to) The initial index of the range (from) must lie between zero and original.length, inclusive.

Arrays.copyOfRange(original, from, to) The final index of the range (to), which must be greater than or equal to from, may be greater than original.length, in which case (short)0 is placed in all elements of the copy whose index is greater than or equal to original.length - from.

ClassLoader.defineClass(b, off, len) The bytes in positions off through off+len-1 should have the format of a valid class file as defined by the Java Virtual Machine Specification

ClassLoader.defineClass(name, b, protectionDomain) @param b The bytes that make up the class data. The bytes from positions b.position() through b.position() + b.limit() -1 should have the format of a valid class file as defined by the Java Virtual Machine Specification.

Arrays.copyOfRange(original, from, to) The initial index of the range (from) must lie between zero and original.length, inclusive.

Arrays.copyOfRange(original, from, to) The final index of the range (to), which must be greater than or equal to from, may be greater than original.length, in which case false is placed in all elements of the copy whose index is greater than or equal to original.length - from.

CollectionUtils.isProperSubCollection(a, b) The implementation assumes * a.size() and b.size() represent the total cardinality of a and b, resp. * a.size() i Integer.MAXVALUE

Arrays.copyOfRange(original, from, to) The initial index of the range (from) must lie between zero and original.length, inclusive.

Arrays.copyOfRange(original, from, to) The final index of the range (to), which must be greater than or equal to from, may be greater than original.length, in which case 0d is placed in all elements of the copy whose index is greater than or equal to original.length - from.

(abstract) PreparedStatement.setClob(parameterIndex, reader, length) The reader must contain the number of characters specified by length otherwise a SQLException will be generated when the PreparedStatement is executed.

(abstract) PreparedStatement.setBlob(parameterIndex, inputStream, length) The inputstream must contain the number of characters specified by length otherwise a SQLException will be generated when the PreparedStatement is executed.

(abstract) PreparedStatement.setNClob(parameterIndex, reader, length) The reader must contain the number of characters specified by length otherwise a SQLException will be generated when the PreparedStatement is executed.

ObjectArrayListIterator.ObjectArrayListIterator(array, start, end) @throws IllegalArgumentException if end index is before the start

AbstractDualBidiMap.AbstractDualBidiMap(normalMap, reverseMap) The two maps must be a matching pair, normal and reverse.

SecureClassLoader.defineClass(name, b, cs) @param b the bytes that make up the class data. The bytes from positions b.position() through b.position() + b.limit() -1 should have the format of a valid class file as defined by the Java Virtual Machine Specification.

KeyStore.setKeyEntry(alias, key, password, chain) If the given key is of type java.security.PrivateKey, it must be accompanied by a certificate chain certifying the corresponding public key.

(abstract) KeyStoreSpi.engineSetKeyEntry(alias, key, password, chain) If the given key is of type java.security.PrivateKey, it must be accompanied by a certificate chain certifying the corresponding public key.

BigDecimal.divideToIntegralValue(divisor, mc) . @throws ArithmeticException if divisor==0 @throws ArithmeticException if mc.precision @literal ¿ 0 and the result requires a precision of more than mc.precision digits.

BigDecimal.remainder(divisor, mc) @throws ArithmeticException if the result is inexact but the rounding mode is UNNECESSARY, or mc.precision @literal ¿ 0 and the result of this.divideToIntgralValue(divisor) would require a precision of more than mc.precision digits.

BigDecimal.divideAndRemainder(divisor, mc) @throws ArithmeticException if the result is inexact but the rounding mode is UNNECESSARY, or mc.precision @literal ¿ 0 and the result of this.divideToIntgralValue(divisor) would require a precision of more than mc.precision digits.

(abstract) ResultSet.updateObject(columnLabel, x, scaleOrLength) If the second argument is an InputStream then the stream must contain the number of bytes specified by scaleOrLength.

(abstract) PreparedStatement.setObject(parameterIndex, x, targetSqlType, scaleOrLength) If the second argument is an InputStream then the stream must contain the number of bytes specified by scaleOrLength. If the second argument is a Reader then the reader must contain the number of characters specified by scaleOrLength.

(abstract) CallableStatement.setBlob(parameterName, inputStream, length) The inputstream must contain the number of characters specified by length, otherwise a SQLException will be generated when the CallableStatement is executed.

ClassLoader.defineClass(name, b, off, len) @param b The bytes that make up the class data. The bytes in positions off through off+len-1 should have the format of a valid class file as defined by the Java Virtual Machine Specification.

(abstract) KeyStoreSpi.engineSetKeyEntry(alias, key, chain) If the protected key is of type java.security.PrivateKey, it must be accompanied by a certificate chain certifying the corresponding public key.

(abstract) CallableStatement.setClob(parameterName, reader, length) The reader must contain the number of characters specified by length otherwise a SQLException will be generated when the CallableStatement is executed.

(abstract) CallableStatement.setNClob(parameterName, reader, length) The reader must contain the number of characters specified by length otherwise a SQLException will be generated when the CallableStatement is executed.

Inflater.Inflater(nowrap) When using the ’nowrap’ option it is also necessary to provide an extra ”dummy” byte as input. This is required by the ZLIB native library in order to support certain optimizations.

Locale.toString() If both the language and country fields are missing, this function will return the empty string, even if the variant field is filled in (you can’t have a locale with just a variant– the variant must accompany a valid language or country code).

interface IPartitionTokenScanner In a simple case, a partition token scanner must always start at a partition boundary.

GridLayoutFactory.extendedMargins(left, right, top, bottom) Note that thes margins will be added to the ones specified by @link #margins(int, int).

CertificateFactorySpi.engineGenerateCertPath(inStream, encoding) The data is assumed to be in the specified encoding.

RandomAccessFile.read(b, off, len) @exception IndexOutOfBoundsException If off is negative, len is negative, or len is greater than b.length - off

1.7 Post-Call Directive

ThreadGroup.enumerate(list) If it is critical to obtain every active thread in this thread group and its subgroups, the caller should verify that the returned int value is strictly less than the length of list.

Policy.getPolicy() This value should not be cached, as it may be changed by a call to setPolicy.

AlgorithmParameters.getInstance(algorithm, provider) The returned parameter object must be initialized via a call to init, using an appropriate parameter specification or parameter encoding.

AlgorithmParameters.getInstance(algorithm) The returned parameter object must be initialized via a call to init, using an appropriate parameter specification or parameter encoding.

class AlgorithmParameters Once an AlgorithmParameters object is obtained, it must be initialized via a call to init, using an appropriate parameter specification or parameter encoding.

AlgorithmParameters.getInstance(algorithm, provider) The returned parameter object must be initialized via a call to init, using an appropriate parameter specification or parameter encoding.

class ObjectInputStream The method readObject is used to read an object from the stream. Java’s safe casting should be used to get the desired type.

File.renameTo(dest) The return value should always be checked to make sure that the rename operation was successful.

MessageDialogWithToggle.openYesNoQuestion(parent, title, message, toggleMessage, toggleState, store, key) @return the dialog, after being closed by the user, which the client can only call getReturnCode() or getToggleState()

MessageDialogWithToggle.openYesNoCancelQuestion(parent, title, message, toggleMessage, toggleState, store, key) @return the dialog, after being closed by the user, which the client can only call getReturnCode() or getToggleState()

MessageDialogWithToggle.openWarning(parent, title, message, toggleMessage, toggleState, store, key) @return the dialog, after being closed by the user, which the client can only call getReturnCode() or getToggleState()

MessageDialogWithToggle.openOkCancelConfirm(parent, title, message, toggleMessage, toggleState, store, key) @return the dialog, after being closed by the user, which the client can only call getReturnCode() or getToggleState()
MessageDialogWithToggle.openInformation(parent, title, message, toggleMessage, toggleState, store, key) @return the dialog, after being closed by the user, which the client can only call getReturnCode() or getToggleState()
MessageDialogWithToggle.openError(parent, title, message, toggleMessage, toggleState, store, key) @return the dialog, after being closed by the user, which the client can only call getReturnCode() or getToggleState()
MessageDialogWithToggle.open(kind, parent, title, message, toggleMessage, toggleState, store, key, style) @return the dialog, after being closed by the user, which the client can only call getReturnCode() or getToggleState()
(abstract) ColumnViewer.getViewerRowFromItem(item) callers can only use the viewer row locally and until the next call to this method.
MultiKey.getKey(index) If it is not then it must not be changed.
MultiKey.MultiKey(key1, key2, key3, key4) The keys should be immutable If they are not then they must not be changed after adding to the MultiKey.
MultiKey.MultiKey(key1, key2, key3, key4, key5) The keys should be immutable If they are not then they must not be changed after adding to the MultiKey.
MultiKey.MultiKey(keys) The keys should be immutable If they are not then they must not be changed after adding to the MultiKey.
class AbstractReferenceMap.ReferenceEntry If getKey() or getValue() returns null, it means the mapping is stale and should be removed.
ServiceLoader.iterator() To achieve laziness the actual work of parsing the available provider-configuration files and instantiating providers must be done by the iterator itself.
ImageDescriptor.createImage(device) The returned image must be explicitly disposed using the image's dispose call.
DelegatingStyledCellLabelProvider.getImage(element) The image is owned by the label provider and must not be disposed directly.
ResourceManager.createColor(descriptor) Any color allocated in this manner must be disposed by calling @link #destroyColor(RGB), or by an eventual call to @link #dispose(). @link Color#dispose() must never been called directly on the returned color.
ResourceManager.createImageWithDefault(descriptor) Creates an image, given an image descriptor. Images allocated in this manner must be disposed by @link #destroyImage(ImageDescriptor), and never by calling @link Image#dispose().
SourceViewer.createFormattingContext() After the use of the context, clients are required to call its dispose method.
ImageDescriptor.createImage(returnMissingImageOnError, device) Note: Even if returnMissingImageOnError is true, it is still possible for this method to return null in extreme cases, for example if SWT runs out of image handles.
Integer.toString(i, radix) If uppercase letters are desired, the @link java.lang.String#toUpperCase() method may be called on the result:
Integer.toString(n, 16).toUpperCase()
Integer.toHexString(i) If uppercase letters are desired, the @link java.lang.String#toUpperCase() method may be called on the result:
Integer.toHexString(n).toUpperCase()
SecurityManager.checkTopLevelWindow(window) If the method returns true, then the window can be shown without any special restrictions.
ResourceManager.createColor(descriptor) @link Color#dispose() must never been called directly on the returned color.
(abstract) ResourceManager.create(descriptor)
The resources returned from this method are reference counted and may be shared internally with other resource managers. They should never be disposed outside of the ResourceManager framework, or it will cause exceptions in other code that shares them. For example, never call @link org.eclipse.swt.graphics.Resource#dispose() on anything returned from this method.
InputStream.available() t is never correct to use the return value of this method to allocate a buffer intended to hold all data in this stream.
StructuredViewer.findItem(element) @return the corresponding widget, or null if none
interface Map.Entry These Map.Entry objects are valid only for the duration of the iteration; more formally, the behavior of a map entry is undefined if the backing map has been modified after the entry was returned by the iterator, except through the setValue operation on the map entry.
class FindReplaceDocumentAdapterContentProposalProvider Note: Clients must not make any assumptions about the returned proposals. This can change from release to release to adapt to changes made in @link FindReplaceDocumentAdapter.
ResourceManager.createFont(descriptor) The method @link Font#dispose() must never be called directly on the returned font.
PopupDialog.getForegroundColorExclusions() Return a list of controls which should never have their foreground color reset.

1.8 Return Value Directive

Executors.newSingleThreadScheduledExecutor(threadFactory) Unlike the otherwise equivalent newScheduledThreadPool(1, threadFactory) the returned executor is guaranteed not to be reconfigurable to use additional threads.
Executors.newSingleThreadScheduledExecutor() Unlike the otherwise equivalent newScheduledThreadPool(1) the returned executor is guaranteed not to be reconfigurable to use additional threads.
java/util/concurrent/atomic The atomic classes also support method weakCompareAndSet, which has limited applicability. On some platforms, the weak version may be more efficient than compareAndSet in the normal case, but differs in that any given invocation of the weakCompareAndSet method may return false spuriously (that is, for no apparent reason). A false return means only that the operation may be retried if desired, relying on the guarantee that repeated invocation when the variable holds expectedValue and no other thread is also attempting to set the variable will eventually succeed.
(abstract) IDialogPage.getMessage() @return the message, or null if none
(abstract) ThreadMXBean.getThreadInfo(ids, lockedMonitors, lockedSynchronizers) If a thread does not lock any object monitor or lockedMonitors is false, the returned ThreadInfo object will have an empty MonitorInfo array. Similarly, if a thread does not lock any synchronizer or lockedSynchronizers is false, the returned ThreadInfo object will have an empty LockInfo array.
(abstract) ISourceViewerExtension3.getQuickAssistAssistant() @return the quick assist assistant or null if none is configured
AbstractQueuedLongSynchronizer.getWaitingThreads(condition) Because the actual set of threads may change dynamically while constructing this result, the returned collection is only a best-effort estimate.
InetAddress.getCanonicalHostName() Best effort method, meaning we may not be able to return the FQDN depending on the underlying system configuration.
BindingManager.getActiveBindingsDisregardingContextFlat() @return All of the active bindings (Binding), not sorted in any fashion. This collection may be empty, but it is never null.
KeyBinding.getKeySequence() @return The key sequence; never null.
BindingManager.getPartialMatches(trigger) @return A map of triggers (TriggerSequence) to bindings (Binding). This map may be empty, but it is never null.
KeySequenceText.getKeySequence() @return The key sequence representation; never null.
KeyFormatterFactory.getEmacsKeyFormatter() @return The Xemacs formatter; never null.
KeyLookupFactory.getSWTKeyLookup() @return The SWT look-up table for key stroke format information; never null.
(abstract) FontDescriptor.createFont(device) @return a newly allocated Font (never null)
ViewerLabel.getTooltipText() @return @link String or null if the tool tip text was never set.
ExternalActionManager.getInstance() @return The singleton instance; this value is never null.
IAction.RESULT Clients should always assume that running the action succeeded in the absence of notification to the contrary.
LegacyActionTools.extractAcceleratorText(text) @return the accelerator text, or null
Dialog.getButton(id) @return the button for the ID or null
Dialog.getOKButton() If createButton was never called with this parameter, or if createButton is overridden, getOKButton will return null. @return the OK button or null
InflaterInputStream.available() Returns 0 after EOF has been reached, otherwise always return 1.
InflaterInputStream.available() @return 1 before EOF and 0 after EOF.
CertPath.getType() @return the type of Certificates in this certification path (never null)
PKIXCertPathValidatorResult.getPublicKey() @return the public key of the subject (never null)
(abstract) Statement.executeQuery(sql) @return a ResultSet object that contains the data produced by the given query; never null
PopupDialog.getDialogSettings() @return settings the dialog settings used to store the dialog's location and/or size, or null if the dialog's bounds should never be stored.
URI.getRawSchemeSpecificPart() @return The raw scheme-specific part of this URI (never null)
ActivationGroupDesc.CommandEnvironment.getCommandOptions() @return An array of the command options which will be passed to the new child command by rmid. Note that rmid may add other options before or after these options, or both. Never returns null.
KeyFormatterFactory.getDefault() @return The default formatter; never null.
BindingManager.getLocale() @return The active locale; never null.
Binding.getSchemeId() @return The scheme identifier; never null.
Binding.getContextId() @return The context identifier; never null.
JFaceResources.getFontDescriptor(symbolicName) @return the font descriptor (never null)
SWTKeySupport.getKeyFormatterForPlatform() @return an instance of IKeyFormatter appropriate for the current instance; never null.

KeyLookupFactory.getDefault() @return The default look-up; never null.

BindingManager.getCurrentConflicts() @return Read-only @link Map of the current conflicts. If no conflicts, then return an empty map. Never null

TreeNode.getChildren() @return The child nodes; may be null, but never empty.

JFaceResources.getHeaderFontDescriptor() @return the font descriptor (never null)

BindingManager.getActiveBindingsFor(parameterizedCommand) @return The array of active triggers (TriggerSequence) for a particular command identifier. This value is guaranteed to never be null, but it may be empty.

(abstract) LinkedModeUI.LinkedModeUITarget.getViewer() Returns the viewer represented by this target, never null.

System.getenv() The returned map will never contain null keys or values.

PolicyQualifierInfo.getPolicyQualifierId() @return the OID (never null)

(abstract) PreparedStatement.executeQuery() @return a ResultSet object that contains the data produced by the query; never null

URI.getSchemeSpecificPart() @return The decoded scheme-specific part of this URI (never null)

PolicyQualifierInfo.getEncoded() @return the ASN.1 DER encoded bytes (never null)

(abstract) OperatingSystemMXBean.getAvailableProcessors() @return the number of processors available to the virtual machine; never smaller than one.

Date.getSeconds() The value returned is between 0 and 61.

(abstract) CompiledScript.eval(context) Should return null if no value is returned by the script execution.

Math.cos(a) The computed result must be within 1 ulp of the exact result.

Math.cos(a) Results must be semi-monotonic

(abstract) Wrapper.unwrap(iface) @param iface A Class defining an interface that the result must implement

PopupDialog.createDialogArea(parent) The returned control's layout data must be an instance of GridData.

Dialog.createDialogArea(parent) The returned control's layout data must be an instance of GridData.

PageAttributes.getPrinterResolution() @return an integer array of 3 elements. The first element must be greater than 0. The second element must be must be greater than 0. The third element must be either 3 or 4.

PageAttributes.PageAttributes(color, media, orientationRequested, origin, printQuality, printerResolution) @param printerResolution an integer array of 3 elements. The first element must be greater than 0. The second element must be must be greater than 0. The third element must be either 3 or 4.

JobAttributes.getPageRanges() Ranges must be in ascending order and must not overlap.

Math.log1p(x) The computed result must be within 1 ulp of the exact result. Results must be semi-monotonic.

PredicatedMap.isSetValueChecking() Override to only return true when there is a value transformer.

NumberFormat.getAvailableLocales() It must contain at least a Locale instance equal to @link java.util.Locale#US Locale.US.

DecimalFormatSymbols.getAvailableLocales() Returns an array of all locales for which the getInstance methods of this class can return localized instances. It must contain at least a Locale instance equal to @link java.util.Locale#US Locale.US

DateFormatSymbols.getAvailableLocales() Returns an array of all locales for which the getInstance methods of this class can return localized instances. It must contain at least a Locale instance equal to @link java.util.Locale#US Locale.US

DateFormat.getAvailableLocales() Returns an array of all locales for which the get*Instance methods of this class can return localized instances. It must contain at least a Locale instance equal to @link java.util.Locale#US Locale.US

Collator.getAvailableLocales() Returns an array of all locales for which the getInstance methods of this class can return localized instances. It must contain at least a Locale instance equal to @link java.util.Locale#US Locale.US.

BreakIterator.getAvailableLocales() Returns an array of all locales for which the get*Instance methods of this class can return localized instances. It must contain at least a Locale instance equal to @link java.util.Locale#US Locale.US

(abstract) Collection.equals(o) Thus, a custom equals method for a collection class that implements neither the List nor Set interface must return false when this collection is compared to any list or set.

EnumSet.range(from, to) The returned set will contain the endpoints themselves, which may be identical but must not be out of order.

(abstract) ResponseCache.put(uri, conn) If the resource is to be cached, then put() must return a CacheRequest object which contains a WriteableByteChannel that the protocol handler will use to write the resource into the cache. If the resource is not to be cached, then put must return null.

CopyOnWriteArraySet.toArray() If this set makes any guarantees as to what order its elements are returned by its iterator, this method must return the elements in the same order.

(abstract) AbstractPreferences.childSpi(name) If a like-named child of this node was previously removed, the implementer must return a newly constructed AbstractPreferences node; once removed, an AbstractPreferences node cannot be "resuscitated."

Applet.getAppletInfo() An applet should override this method to return a String containing information about the author, version, and copyright of the applet.

ExternalActionManager.getCallback() This value may be null if there is no extra information.

Dialog.createButtonBar(parent) The returned control's layout data must be an instance of GridData.

Math.sin(a) The computed result must be within 1 ulp of the exact result. Results must be semi-monotonic.

Math.tan(a) The computed result must be within 1 ulp of the exact result. Results must be semi-monotonic.

Math.acos(a) The computed result must be within 1 ulp of the exact result. Results must be semi-monotonic.

Math.asin(a) The computed result must be within 1 ulp of the exact result. Results must be semi-monotonic.

Math.atan(a) The computed result must be within 1 ulp of the exact result. Results must be semi-monotonic.

Math.exp(a) The computed result must be within 1 ulp of the exact result. Results must be semi-monotonic.

Math.log(a) The computed result must be within 1 ulp of the exact result. Results must be semi-monotonic.

Math.log10(a) The computed result must be within 1 ulp of the exact result. Results must be semi-monotonic.

Math.hypot(x, y) The computed result must be within 1 ulp of the exact result. If one parameter is held constant, the results must be semi-monotonic in the other parameter.

Package.getSpecificationVersion() This version string must be a sequence of positive decimal integers separated by "."'s and may have leading zeros.

Proxy.getProxyClass(loader, interfaces) The resulting proxy class must not exceed any limits imposed on classes by the virtual machine. For example, the VM may limit the number of interfaces that a class may implement to 65535; in that case, the size of the interfaces array must not exceed 65535.

Math.expm1(x) The computed result must be within 1 ulp of the exact result. Results must be semi-monotonic. The result of expm1 for any finite input must be greater than or equal to -1.0.

Float.toString(f) How many digits must be printed for the fractional part of m or a? There must be at least one digit to represent the fractional part, and beyond that as many, but only as many, more digits as are needed to uniquely distinguish the argument value from adjacent values of type float. That is, suppose that x is the exact mathematical value represented by the decimal representation produced by this method for a finite nonzero argument f. Then f must be the float value nearest to x; or, if two float values are equally close to x, then f must be one of them and the least significant bit of the significand of f must be 0.

Double.toString(d) How many digits must be printed for the fractional part of m or a? There must be at least one digit to represent the fractional part, and beyond that as many, but only as many, more digits as are needed to uniquely distinguish the argument value from adjacent values of type double. That is, suppose that x is the exact mathematical value represented by the decimal representation produced by this method for a finite nonzero argument d. Then d must be the double value nearest to x; or if two double values are equally close to x, then d must be one of them and the least significant bit of the significand of d must be 0.

Math.cbrt(a) The computed result must be within 1 ulp of the exact result.

Math.cosh(x) The computed result must be within 2.5 ulps of the exact result.

Math.tanh(x) *

The computed result must be within 2.5 ulps of the exact result. The result of tanh for any finite input must have an absolute value less than or equal to 1. Note that once the exact result of tanh is within 1/2 of an ulp of the limit value of 1, correctly signed 1.0 should be returned.

(abstract) InvocationHandler.invoke(proxy, method, args) If the declared return type of the interface method is a primitive type, then the value returned by this method must be an instance of the corresponding primitive wrapper class; otherwise, it must be a type assignable to the declared return type.

Method.getGenericReturnType() If the return type is a parameterized type, the Type object returned must accurately reflect the actual type parameters used in the source code.

Field.getGenericType() If the Type is a parameterized type, the Type object returned must accurately reflect the actual type parameters used in the source code.

ObjectInputStream.resolveClass(desc) by subclasses to use an alternate loading mechanism but must return a Class object

(abstract) PreferencePage.createContents(parent) Any subclass returning a Composite object whose Layout has default margins (for example, a GridLayout) are expected to set the margins of this Layout to 0 pixels.

PreferencePage.createControl(parent) If a subclass that overrides this method creates a Composite that has a layout with default margins (for example, a GridLayout) it is expected to set the margins of this Layout to 0 pixels.

CursorableLinkedList.cursor() @return a new cursor iterator

SynchronizedBag.decorate(bag) @return a new synchronized Bag

TransformedBag.decorate(bag, transformer) @return a new transformed Bag

TransformedSortedBag.decorate(bag, transformer) @return a new transformed SortedBag

ObjectOutputStream.replaceObject(obj) This method should return the object to be substituted or the original object.

Math.sinh(x) The computed result must be within 2.5 ulps of the exact result.

Method.getGenericParameterTypes() If a formal parameter type is a parameterized type, the Type object returned for it must accurately reflect the actual type parameters used in the source code.

Method.getGenericExceptionTypes() If an exception type is a parameterized type, the Type object returned for it must accurately reflect the actual type parameters used in the source code.

Constructor.getGenericParameterTypes() If a formal parameter type is a parameterized type, the Type object returned for it must accurately reflect the actual type parameters used in the source code.

Constructor.getGenericExceptionTypes() If an exception type is a parameterized type, the Type object returned for it must accurately reflect the actual type parameters used in the source code.

Class.getGenericSuperclass() If the superclass is a parameterized type, the Type object returned must accurately reflect the actual type parameters used in the source code.

Class.getGenericInterfaces() If a superinterface is a parameterized type, the Type object returned for it must accurately reflect the actual type parameters used in the source code.

Float.toHexString(f) Note that there must be at least one nonzero digit in a subnormal significand.

Double.toHexString(d) Note that there must be at least one nonzero digit in a subnormal significand.

(abstract) CertPath.getCertificates() The List returned must be immutable and thread-safe.

PolicySpi.engineGetPermissions(codesource) f this operation is supported, the returned set of permissions must be a new mutable instance and it must support heterogeneous Permission types.

PolicySpi.engineGetPermissions(domain) the returned set of permissions must be a new mutable instance and it must support heterogeneous Permission types.

(abstract) X509Certificate.getSerialNumber() It must be unique for each certificate issued by a given CA (i.e., the issuer name and serial number identify a unique certificate).

interface CertPathValidatorResult All results returned by the @link CertPathValidator#validate CertPathValidator.validate method must implement this interface.

X509CRLSelector.getDateAndTime() The specified date must be equal to or later than the value of the thisUpdate component of the X509CRL and earlier than the value of the nextUpdate component.

(abstract) KeyStoreSpi.engineGetCertificateChain(alias) The certificate chain must have been associated with the alias by a call to setKeyEntry, or by a call to setEntry with a PrivateKeyEntry.

KeyStore.getCertificateChain(alias) The certificate chain must have been associated with the alias by a call to setKeyEntry, or by a call to setEntry with a PrivateKeyEntry.

CertificateFactorySpi.engineGenerateCertPath(certificates) The certificates supplied must be of a type supported by the CertificateFactory.

(abstract) KeyStoreSpi.engineGetKey(alias, password) The key must have been associated with the alias by a call to setKeyEntry, or by a call to setEntry with a PrivateKeyEntry or SecretKeyEntry.

KeyStore.getKey(alias, password) The key must have been associated with the alias by a call to setKeyEntry, or by a call to setEntry with a PrivateKeyEntry or SecretKeyEntry.

Policy.getPermissions(domain) @return a set of permissions granted to the specified ProtectionDomain. If this operation is supported, the returned set of permissions must be a new mutable instance and it must support heterogeneous Permission types

interface CertPathBuilderResult . All results returned by the @link CertPathBuilder#build CertPathBuilder.build method must implement this interface.

class CertificateFactory A certificate factory for X.509 must return certificates that are an instance of java.security.cert.X509Certificate, and CRLs that are an instance of java.security.cert.X509CRL.

DateFormatSymbols.getZoneStrings() If a zone does not implement daylight saving time, the daylight saving time names should not be used.

KeyFactory.getKeySpec(key, keySpec) keySpec identifies the specification class in which the key material should be returned. It could, for example, be DSAPublicKeySpec.class, to indicate that the key material should be returned in an instance of the DSAPublicKeySpec class.

(abstract) KeyFactorySpi.engineGetKeySpec(key, keySpec) keySpec identifies the specification class in which the key material should be returned. It could, for example, be DSAPublicKeySpec.class, to indicate that the key material should be returned in an instance of the DSAPublicKeySpec class.

(abstract) Permission.getActions() Subclasses should always return actions in what they consider to be their canonical form.

(abstract) PKIXCertPathChecker.getSupportedExtensions() All X.509 certificate extensions that a PKIXCertPathChecker might possibly be able to process should be included in the set.

NotPredicate.getPredicates() @return the predicate as the only element in an array

NullIsExceptionPredicate.getPredicates() @return the predicate as the only element in an array

NullIsFalsePredicate.getPredicates() @return the predicate as the only element in an array

NullIsTruePredicate.getPredicates() @return the predicate as the only element in an array

TransformedPredicate.getPredicates() @return the predicate as the only element in an array

TransformedMap.isSetValueChecking() Override to only return true when there is a value transformer.

IteratorUtils.loopingIterator(coll) @return a new looping iterator

IteratorUtils.loopingListIterator(list) @return a new looping iterator

ListUtils.subtract(list1, list2) @return a new list containing the results

(abstract) Driver.connect(url, info) The driver should return "null" if it realizes it is the wrong kind of driver to connect to the given URL.

CellEditor.dependsOnExternalFocusListener() Subclasses that hook their own focus listener should override this method and return false.

TextCellEditor.dependsOnExternalFocusListener() Subclasses that hook their own focus listener should override this method and return false.

Object.toString() The result should be a concise but informative representation that is easy for a person to read.

(abstract) Set.toArray() If this set makes any guarantees as to what order its elements are returned by its iterator, this method must return the elements in the same order.

(abstract) Collection.toArray() If this collection makes any guarantees as to what order its elements are returned by its iterator, this method must return the elements in the same order.

(abstract) ListResourceBundle.getContents() The first element of each pair is the key, which must be a String, and the second element is the value associated with that key.

(abstract) DataSource.getOutputStream() Note that a new OutputStream object must be returned each time this method is called

(abstract) RMIClassLoaderSpi.loadClass(codebase, name, defaultLoader) An implementation of this method must either return a class with the given name or throw an exception.

(abstract) RMIClassLoaderSpi.loadProxyClass(codebase, interfaces, defaultLoader) An implementation of this method must either return a proxy class that implements the named interfaces or throw an exception.

PageChangingEvent.getCurrentPage() In dialogs implemented by JFace, this will be an IDialogPage.

PageChangingEvent.getTargetPage() In dialogs implemented by JFace, this will be an IDialogPage.

PageChangedEvent.getSelectedPage() In dialogs implemented by JFace, this will be an IDialogPage.

Math.expm1(x) Note that once the exact result of ex - 1 is within 1/2 ulp of the limit value -1, -1.0 should be returned.

(abstract) ClassFileTransformer.transform(loader, className, classBeingRedefined, protectionDomain, classfileBuffer) If the implementing method determines that no transformations are needed, it should return null. Otherwise, it should create a new byte[] array, copy the input classfileBuffer into it, along with all desired transformations, and return the new array.

Throwable.printStackTrace() The backtrace for a throwable with an initialized, non-null cause should generally include the backtrace for the cause.

(abstract) ITextHoverExtension2.getHoverInfo2(textViewer, hoverRegion) Implementers have to ensure that @link ITextHoverExtension#getHoverControlCreator() returns @link IInformationControls that implement @link IInformationControlExtension2 and whose @link IInformationControlExtension2#setInput(Object) can handle the information objects returned by this method.

(abstract) IInformationControlExtension5.getInformationPresenterControlCreator() The returned information control creator must create information controls that implement @link IInformationControlExtension3 and @link IInformationControlExtension2, and whose @link IInformationControlExtension2#setInput(Object) accepts all inputs that are also supported by this information control.

(abstract) ICellEditorValidator.isValid(value) It is the responsibility of the implementor to fully format the message before returning it.

Locale.getAvailableLocales() It must contain at least a Locale instance equal to @link java.util.Locale#US Locale.US.

Calendar.getAvailableLocales() The array returned must contain at least a Locale instance equal to @link java.util.Locale#US Locale.US.

Pack200.Packer.PASS_FILE_PFX The resulting file names must match exactly as strings with their occurrences in the JAR file.

ResourceBundle.Control.getFormats(baseName) The list returned by this method must have at least one String.

ResourceBundle.Control.getFormats(baseName) However, the returned List must not be mutated after it has been returned by getFormats.

(abstract) Collection.toArray(a) If this collection makes any guarantees as to what order its elements are returned by its iterator, this method must return the elements in the same order.

LinkedHashMap.removeEldestEntry(eldest) It is permitted for this method to modify the map directly, but if it does so, it must return false (indicating that the map should not attempt any further modification).

CopyOnWriteArraySet.toArray(a) If this set makes any guarantees as to what order its elements are returned by its iterator, this method must return the elements in the same order.

ResourceBundle.Control.getCandidateLocales(baseName, locale) The last element of the list must be a @linkplain Locale#ROOT root locale if it is desired to have the base bundle as the terminal of the parent chain.

ResourceBundle.Control.getCandidateLocales(baseName, locale) If the given locale is equal to Locale.ROOT (the root locale), a List containing only the root Locale must be returned.

ResourceBundle.Control.getCandidateLocales(baseName, locale) However, the returned List must not be mutated after it has been returned by getCandidateLocales.

BitSet.equals(obj) That is, for every nonnegative int index k,
((BitSet)obj).get(k) == this.get(k)
must be true.

Math.pow(a, b) The computed result must be within 1 ulp of the exact result. Results must be semi-monotonic.

Deflater.deflate(b) A return value of 0 indicates that needsInput() should be called in order to determine if more input data is required.

Level.FINE The exact meaning of the three levels will vary between subsystems, but in general, FINEST should be used for the most voluminous detailed output, FINER for somewhat less detailed output, and FINE for the lowest volume (and most important) messages.

interface NavigableSet Subsets of any NavigableSet must implement the NavigableSet interface.

interface NavigableMap Submaps of any NavigableMap must implement the NavigableMap interface.

ResourceBundle.Control.getTimeToLive(baseName, locale) The value 0 indicates that a bundle must be validated each time it is retrieved from the cache.

SingletonMap.isEmpty() Checks whether the map is currently empty, which it never is.

ObjectArrayIterator.getArray() @return the array this iterator iterates over, or null if the no-arg constructor was used and @link #setArray has never been called with a valid array.

ArrayIterator.getArray() @return the array this iterator iterates over, or null if the no-arg constructor was used and @link #setArray(Object) has never been called with a valid array.

IteratorUtils.getIterator(obj) @return a suitable iterator, never null

abstract class AbstractMap This set should not support the add or remove methods, and its iterator should not support the remove method.

LogManager.readConfiguration(ins) Reinitialize the logging properties and reread the logging configuration from the given stream, which should be in java.util.Properties format.

abstract class AbstractPreferences Under these circumstances, however, all subsequent invocations of flush() and sync should return false, as returning true would imply that all previous operations had successfully been made permanent.

(abstract) CurrencyNameProvider.getSymbol(currencyCode, locale) If no symbol can be determined, null should be returned.

(abstract) TimeZoneNameProvider.getDisplayName(ID, daylight, style, locale) If daylight is true, the method should return a name appropriate for daylight saving time even if the specified time zone has not observed daylight saving time in the past.

(abstract) PredicateDecorator.getPredicates() The array may be the internal data structure of the predicate and thus should not be altered.

MultiKey.equals(other) To be equal, the other object must be a MultiKey with the same number of keys which are also equal.

(abstract) SortedBidiMap.inverseBidiMap() Implementations must return a SortedBidiMap instance, usually by forwarding to inverseSortedBidiMap().

(abstract) OrderedBidiMap.inverseBidiMap() Implementations must return an OrderedBidiMap instance, usually by forwarding to inverseOrderedBidiMap().

SingletonMap.previousKey(key) Gets the previous key before the key specified, always null.

(abstract) Bag.add(object) Since this method always increases the size of the bag, according to the @link Collection#add(Object) contract, it should always return true.

(abstract) DataSource.getInputStream() Note that a new InputStream object must be returned each time this method is called, and the stream must be positioned at the beginning of the data.

SingletonMap.nextKey(key) Gets the next key after the key specified, always null.

(abstract) Set.toArray(a) If this set makes any guarantees as to what order its elements are returned by its iterator, this method must return the elements in the same order.

PredicateUtils.invokerPredicate(methodName) The method must return either a boolean or a non-null Boolean, and have no parameters.

abstract class CertificateFactorySpi A certificate factory for X.509 must return certificates that are an instance of java.security.cert.X509Certificate, and CRLs that are an instance of java.security.cert.X509CRL

ArrayDeque.toArray() In other words, this method must allocate a new array

ArrayList.toArray() In other words, this method must allocate a new array

(abstract) AbstractPreferences.childSpi(name) The implementer must ensure that the returned node has not been removed.

ExceptionTransformer.transform(input) @return never

Math.atan2(y, x) The computed result must be within 2 ulps of the exact result. Results must be semi-monotonic.

Calendar.getMinimalDaysInFirstWeek() If the minimal days required must be a full week, this method returns 7.

DelayQueue.toArray() In other words, this method must allocate a new array

LinkedList.toArray() In other words, this method must allocate a new array

ExceptionFactory.create() @return never @throws FunctorException always

ExceptionPredicate.evaluate(object) @return never @throws FunctorException always

class SecureRandom A cryptographically strong random number minimally complies with the statistical random number generator tests specified in FIPS 140-2, Security Requirements for Cryptographic Modules, section 4.9.1. Additionally, SecureRandom must produce non-deterministic output. Therefore any seed material passed to a SecureRandom object must be unpredictable, and all SecureRandom output sequences must be cryptographically strong, as described in RFC 1750: Randomness Recommendations for Security.

Bidi.requiresBidi(text, start, limit) @return true if the range of characters requires bidi analysis

SignatureSpi.engineGetParameters() The returned parameters may be the same that were used to initialize this signature engine, or may contain a combination of default and randomly generated parameter values used by the underlying signature implementation if this signature engine requires algorithm parameters but was not initialized with any.

AbstractMapBag.retainAll(coll) @return true if this call changed the collection

Object.clone() By convention, the object returned by this method should be independent of this object (which is being cloned). To achieve this independence, it may be necessary to modify one or more fields of the object returned by super.clone before returning it.

abstract class AbstractCollection The iterator returned by the iterator method must implement hasNext and next.

(abstract) List.toArray() In other words, this method must allocate a new array even if this list is backed by an array

(abstract) Binding.getTriggerSequence() The triggers can be anything, but above all it must be hashable.

AbstractReferenceMap.ReferenceEntry.getKey() Gets the key from the entry. This method dereferences weak and soft keys and thus may return null. @return the key, which may be null if it was garbage collected

SetUniqueList.add(object) The List interface requires that this method returns true always.

(abstract) IPredicateRule.evaluate(scanner, resume) Evaluates the rule by examining the characters available from the provided character scanner. The token returned by this rule returns true when calling isUndefined, if the text that the rule investigated does not match the rule's requirements. Otherwise, this method returns this rule's success token.

(abstract) IRule.evaluate(scanner) Evaluates the rule by examining the characters available from the provided character scanner. The token returned by this rule returns true when calling isUndefined, if the text that the rule investigated does not match the rule's requirements

(abstract) IKeyFormatter.format(keySequence) It is required that unequal key sequences return unequal strings.

(abstract) ExternalActionManager.IExecuteCallback.postExecuteSuccess(action, returnValue) @param returnValue The command's result, may be null.

ContentProposalAdapter.getFilterStyle() @return a constant indicating how keystrokes in the proposal popup affect filtering of the proposals shown. FILTER_NONE specifies that no filtering will occur in the content proposal list as keys are typed. FILTER_CHARACTER specifies the content of the popup will be filtered by the most recently typed character. FILTER_CUMULATIVE is deprecated and no longer recommended.

(abstract) ICoolBarManager.getLockLayout() @return true if cool bar layout is locked, false otherwise

(abstract) ICompletionProposal.getImage() @return the image to be shown or null if no image is desired

BrowserInformationControl.BrowserInformationControl(parent, symbolicFontName, toolBarManager) @param toolBarManager the manager or null if toolbar is not desired

abstract class AbstractPreferences Even under these circumstances it is generally better to simply ignore the invocation and return, rather than throwing an exception.

(abstract) IStructuredContentProvider.getElements(inputElement) For instances where the viewer is displaying a tree containing a single 'root' element it is still necessary that the 'input' does not return itself from this method. This leads to recursion issues (see bug 9262).

(abstract) NavigableMap.lowerKey(key) Returns the greatest key strictly less than the given key, or null if there is no such key.

(abstract) NavigableSet.higher(e) Returns the least element in this set strictly greater than the given element, or null if there is no such element.

(abstract) NavigableMap.higherKey(key) Returns the least key strictly greater than the given key, or null if there is no such key.

(abstract) ThreadMXBean.getCurrentThreadCpuTime() The returned value is of nanoseconds precision but not necessarily nanoseconds accuracy.

(abstract) ResultSetMetaData.getPrecision(column) Get the designated column's specified column size. For numeric data, this is the maximum precision. For character data, this is the length in characters. For datetime datatypes, this is the length in characters of the String representation (assuming the maximum allowed precision of

the fractional seconds component). For binary data, this is the length in bytes. For the ROWID datatype, this is the length in bytes. 0 is returned for data types where the column size is not applicable.
(abstract) DatabaseMetaData.getFunctionColumns(catalog, schemaPattern, functionNamePattern, columnNamePattern) The PRECISION column represents the specified column size for the given parameter or column. For numeric data, this is the maximum precision. For character data, this is the length in characters. For datetime datatypes, this is the length in characters of the String representation (assuming the maximum allowed precision of the fractional seconds component). For binary data, this is the length in bytes. For the ROWID datatype, this is the length in bytes. Null is returned for data types where the column size is not applicable
(abstract) ThreadMXBean.getCurrentThreadCpuTime() If the implementation distinguishes between user mode time and system mode time, the returned CPU time is the amount of time that the current thread has executed in user mode or system mode.

1.9 Method Call Visibility Directive

LogRecord.setSequenceNumber(seq) it should not normally be necessary to use this method.
Logger.setParent(parent) It should not be called from application code.
AbstractOrderedMapDecorator.AbstractOrderedMapDecorator() Constructor only used in deserialization, do not use otherwise.
(abstract) IContributionManagerOverrides.getAccelerator(item) This is not intended to be called outside of the workbench.
(abstract) IContributionManagerOverrides.getAcceleratorText(item) This is not intended to be called outside of the workbench.
(abstract) IContributionManagerOverrides.getText(item) This is not intended to be called outside of the workbench.
(abstract) IContributionItem.dispose() Clients should not call this method directly, unless they have removed this contribution item from the containing IContributionManager before the contribution lifecycle has ended.
ListEditor.getShell() This method is internal to the framework; subclassers should not call this method.
AbstractInformationControlManager.fInformationControl This field should not be referenced by subclasses.
AbstractInformationControlManager.fInformationControlCreator This field should not be referenced by subclasses.
AbstractInformationControlManager.fInformationControlCloser This field should not be referenced by subclasses.
AbstractInformationControlManager.fDisposed This field should not be referenced by subclasses.
Activatable.exportObject(obj, location, data, restart, port) Note: Using this method (as well as the Activatable constructors that both register and export an activatable remote object) is strongly discouraged because the actions of registering and exporting the remote object are not guaranteed to be atomic.
Activatable.Activatable(location, data, restart, port) Note: Using the Activatable constructors that both register and export an activatable remote object is strongly discouraged because the actions of registering and exporting the remote object are not guaranteed to be atomic.
Activatable.Activatable(location, data, restart, port, csf, ssf) Note: Using the Activatable constructors that both register and export an activatable remote object is strongly discouraged because the actions of registering and exporting the remote object are not guaranteed to be atomic.
Activatable.exportObject(obj, location, data, restart, port, csf, ssf) Note: Using this method (as well as the Activatable constructors that both register and export an activatable remote object) is strongly discouraged because the actions of registering and exporting the remote object are not guaranteed to be atomic.
Policy.getPermissions(codesource) Applications are discouraged from calling this method since this operation may not be supported by all policy implementations.
IdentityScope.IdentityScope() This constructor is used for serialization only and should not be used by subclasses.
Signer.Signer() This constructor should only be used for serialization.
X509CertSelector.setSubject(subjectDN) Denigrated, use @linkplain #setSubject(X500Principal) or @linkplain #setSubject(byte[]) instead. This method should not be relied on as it can fail to match some certificates because of a loss of encoding information in the RFC 2253 String form of some distinguished names.
(abstract) X509Certificate.getSubjectDN() Denigrated, replaced by @linkplain #getSubjectX500Principal(). This method returns the subject as an implementation specific Principal object, which should not be relied upon by portable code.
X509CertSelector.getIssuerAsString() Denigrated, use @linkplain #getIssuer() or @linkplain #getIssuerAsBytes() instead. This method should not be relied on as it can fail to match some certificates because of a loss of encoding information in the RFC 2253 String form of some distinguished names.
X509CertSelector.getSubjectAsString() Denigrated, use @linkplain #getSubject() or @linkplain #getSubjectAsBytes() instead. This method should not be relied on as it can fail to match some certificates because of a loss of encoding information in the RFC 2253 String form of some distinguished names.
(abstract) X509Certificate.getIssuerDN() Denigrated, replaced by @linkplain #getIssuerX500Principal(). This method returns the issuer as an implementation specific Principal object, which should not be relied upon by portable code.
X509CRLSelector.addIssuerName(name) Denigrated, use @linkplain #addIssuer(X500Principal) or @linkplain #addIssuerName(byte[]) instead. This method should not be relied on as it can fail to match some CRLs because of a loss of encoding information in the RFC 2253 String form of some distinguished names.
(abstract) X509CRL.getIssuerDN() Denigrated, replaced by @linkplain #getIssuerX500Principal(). This method returns the issuer as an implementation specific Principal object, which should not be relied upon by portable code.
AbstractHashedMap.AbstractHashedMap() Constructor only used in deserialization, do not use otherwise.
AbstractLinkedMap.AbstractLinkedMap() Constructor only used in deserialization, do not use otherwise.
AbstractSortedBagDecorator.AbstractSortedBagDecorator() Constructor only used in deserialization, do not use otherwise.
AbstractBagDecorator.AbstractBagDecorator() Constructor only used in deserialization,
AbstractCollectionDecorator.AbstractCollectionDecorator() Constructor only used in deserialization, do not use otherwise.
AbstractBufferDecorator.AbstractBufferDecorator() Constructor only used in deserialization, do not use otherwise.
AbstractMapDecorator.AbstractMapDecorator() Constructor only used in deserialization, do not use otherwise.
AbstractSortedMapDecorator.AbstractSortedMapDecorator() Constructor only used in deserialization, do not use otherwise.
AbstractListDecorator.AbstractListDecorator() Constructor only used in deserialization, do not use otherwise.
AbstractSortedSetDecorator.AbstractSortedSetDecorator() Constructor only used in deserialization, do not use otherwise.
AbstractSetDecorator.AbstractSetDecorator() Constructor only used in deserialization, do not use otherwise.
abstract class MessageDigest Application developers should only take notice of the methods defined in this MessageDigest class; all the methods in the superclass are intended for cryptographic service providers who wish to supply their own implementations of message digest algorithms
Policy.getPermissions(domain) Applications are discouraged from calling this method since this operation may not be supported by all policy implementations. Applications should rely on the implies method to perform policy checks.
abstract class Signature Note that this class is abstract and extends from SignatureSpi for historical reasons. Application developers should only take notice of the methods defined in this Signature class; all the methods in the superclass are intended for cryptographic service providers who wish to supply their own implementations of digital signature algorithms.
abstract class KeyPairGenerator Note that this class is abstract and extends from KeyPairGeneratorSpi for historical reasons. Application developers should only take notice of the methods defined in this KeyPairGenerator class; all the methods in the superclass are intended for cryptographic service providers who wish to supply their own implementations of key pair generators.
MultiKey.MultiKey(keys, makeClone) This method is public for performance reasons only, to avoid a clone.
Time.getYear() This method is deprecated and should not be used because SQL TIME values do not have a year component.
Time.getMonth() This method is deprecated and should not be used because SQL TIME values do not have a month component.
Time.getDay() This method is deprecated and should not be used because SQL TIME values do not have a day component.
Time.getDate() This method is deprecated and should not be used because SQL TIME values do not have a date component.
Time.setYear(i) This method is deprecated and should not be used because SQL TIME values do not have a year component.
Time.setMonth(i) This method is deprecated and should not be used because SQL TIME values do not have a month component.
Time.setDate(i) This method is deprecated and should not be used because SQL TIME values do not have a date component.
Date.getHours() This method is deprecated and should not be used because SQL Date values do not have a time component.
Date.getMinutes() This method is deprecated and should not be used because SQL Date values do not have a time component.
Date.getSeconds() This method is deprecated and should not be used because SQL Date values do not have a time component.
Date.setHours(i) This method is deprecated and should not be used because SQL Date values do not have a time component.
Date.setMinutes(i) This method is deprecated and should not be used because SQL Date values do not have a time component.
Date.setSeconds(i) This method is deprecated and should not be used because SQL Date values do not have a time component.
(abstract) StructuredViewer.doFindItem(element) This method is internal to the framework; subclassers should not call this method.
StructuredViewer.handleOpen(event) This method is internal to the framework; subclassers should not call this method.
(abstract) StructuredViewer.doFindInputItem(element) This method is internal to the framework; subclassers should not call this method.
StructuredViewer.handleSelect(event) This method is internal to the framework; subclassers should not call this method.
StructuredViewer.handleDoubleSelect(event) This method is internal to the framework; subclassers should not call this method.
StructuredViewer.refreshItem(widget, element) This method is internal to the framework; subclassers should not call this method.
StructuredViewer.handlePostSelect(e) This method is internal to the framework; subclassers should not call this method.
StructuredViewer.unmapAllElements() This method is internal to the framework; subclassers should not call this method.
StructuredViewer.needsRefilter(element, property) This method is internal to the framework; subclassers should not call this method.

(abstract) StructuredViewer.doUpdateItem(item, element, fullMap) This method is internal to the framework; subclassers should not call this method.

StructuredViewer.unmapElement(element, item) This method is internal to the framework; subclassers should not call this method.

StructuredViewer.mapElement(element, item) This method is internal to the framework; subclassers should not call this method.

StructuredViewer.unmapElement(element) This method is internal to the framework; subclassers should not call this method.

StructuredViewer.updateItem(widget, element) This method is internal to the framework; subclassers should not call this method.

AbstractHoverInformationControlManager.getInternalAccessor() @noreference This method is not intended to be referenced by clients.

AnnotationBarHoverManager.getInternalAccessor() @noreference This method is not intended to be referenced by clients.

TemplateCompletionProcessor.createProposal(template, context, region, relevance) Do neither call nor override.

ViewerDropAdapter.handleException(exception, event) @deprecated this method should not be used.

Runtime.halt(status) This method should be used with extreme caution.

Thread.Thread(group, target, name, stackSize) Due to the platform-dependent nature of the behavior of this constructor, extreme care should be exercised in its use.

StructuredViewer.internalUpdate(widget, element, properties) This support will likely be removed in Eclipse 3.3 in favor of proper support for multiple equal elements (which was implemented for AbtractTreeViewer in 3.2).

Logger.global @deprecated Initialization of this field is prone to deadlocks.

(abstract) Handler.close() After close has been called this Handler should no longer be used.

SimpleTimeZone.getOffset(era, year, month, day, dayOfWeek, millis) Note: In general, clients should use Calendar.get(ZONE_OFFSET) + Calendar.get(DST_OFFSET) instead of calling this method.

SplitMapUtils.SplitMapUtils() SplitMapUtils should not normally be instantiated.

MapUtils.MapUtils() MapUtils should not normally be instantiated.

ListUtils.ListUtils() ListUtils should not normally be instantiated.

BufferUtils.BufferUtils() BufferUtils should not normally be instantiated.

ComparatorUtils.ComparatorUtils() ComparatorUtils should not normally be instantiated.

CollectionUtils.CollectionUtils() CollectionUtils should not normally be instantiated.

SetUtils.SetUtils() SetUtils should not normally be instantiated.

class Properties Because Properties inherits from Hashtable, the put and putAll methods can be applied to a Properties object. Their use is strongly discouraged as they allow the caller to insert entries whose keys or values are not Strings. The setProperty method should be used instead.

abstract class Preferences Ordinary applications should have no need to invoke any of these methods, which can be identified by the fact that they are declared to throw @link BackingStoreException.

ComparableComparator.ComparableComparator() Constructor whose use should be avoided.

AbstractMapBag.getMap() Not intended for interactive use outside of subclasses.

ContentAssistant.setProposalSelectorForeground(foreground) As of 3.4, you should only call this method if you want to override the @link JFacePreferences#CONTENT_ASSIST_FC

(abstract) IQuickAssistAssistant.setProposalSelectorBackground(background) As of 3.4, you should only call this method if you want to override the @link JFacePreferences#CONTENT_ASSIST_BACKGROUND_COLOR.

AbstractInformationControlManager.presentInformation() This method should only be called from overriding methods or from setInformation.

(abstract) ICompletionProposal.getAdditionalProposalInfo() If @link ICompletionProposalExtension5 is implemented, this method should not be called any longer.

Locale.setDefault(newLocale) Since changing the default locale may affect many different areas of functionality, this method should only be used if the caller is prepared to reinitialize locale-sensitive code running within the same Java Virtual Machine.

RevisionInformation.applyDiff(hunks) Note: This is an internal framework method and must not be called by clients.

class NodeChangeEvent Note, that although NodeChangeEvent inherits Serializable interface from java.util.EventObject, it is not intended to be Serializable.

class PreferenceChangeEvent Note, that although PreferenceChangeEvent inherits Serializable interface from EventObject, it is not intended to be Serializable.

class TableViewer Users of SWT.VIRTUAL should also avoid using getItems() from the Table within the TreeViewer as this does not necessarily generate a callback for the TreeViewer to populate the items. It also has the side effect of creating all of the items thereby eliminating the performance improvements of SWT.VIRTUAL.

interface SQLInput This interface, used only for custom mapping, is used by the driver behind the scenes, and a programmer never directly invokes SQLInput methods.

Binding.toString() The string representation of this binding – for debugging purposes only.

StructuredSelection.toString() Internal method which returns a string representation of this selection suitable for debug purposes only.

Scheme.toString() The string representation of this command – for debugging purposes only.

ContributionManager.dumpStatistics() Internal debug method for printing statistics about this manager to System.out.

Binding.string This string is for debugging purposes only, and is not meant to be displayed to the user.

WizardPage.toString() Returns a printable representation of this wizard page suitable only for debug purposes.

ContributionItem.toString() Returns a string representation of this contribution item suitable only for debugging.

AbstractInformationControlManager.getInternalAccessor() Note: This method is not intended to be referenced or overridden by clients.

(abstract) IWidgetTokenKeeper.requestWidgetToken(owner) Note, the keeper must not call releaseWidgetToken(IWidgetTokenKeeper) explicitly.

Date.setSeconds(i) @exception java.lang.IllegalArgumentException if this method is invoked

Calendar.toString() This method is intended to be used only for debugging purposes, and the format of the returned string may vary between implementations.

Thread.dumpStack() This method is used only for debugging.

interface SQLOutput The output stream for writing the attributes of a user-defined type back to the database. This interface, used only for custom mapping, is used by the driver, and its methods are never directly invoked by a programmer.

class LegacyActionTools Clients may neither instantiate this class nor extend.

abstract class AbstractAction Clients may neither instantiate nor extend this class.

class ExternalActionManager This class is not intended to be called or extended by any external clients.

class ModalContext @noinstantiate This class is not intended to be instantiated by clients.

class JFaceResources @noinstantiate This class is not intended to be instantiated by clients.

class ImageUtilities @noinstantiate This class is not intended to be instantiated by clients.

class JFaceTextUtil @noinstantiate This class is not intended to be instantiated by clients.

interface RegistryHandler It should never be accessed by application code.

interface LoaderHandler It should never be accessed by application code.

class RMISecurityException Application code should never directly reference this class

class RMISecurityManager Therefore an RMI application should use the SecurityManager class or another application-specific SecurityManager implementation instead of this class.

Thread.stop() An application should not normally try to catch ThreadDeath unless it must do some extraordinary cleanup operation (note that the throwing of ThreadDeath causes finally clauses of try statements to be executed before the thread officially dies). If a catch clause catches a ThreadDeath object, it is important to rethrow the object so that the thread actually dies.

class AllPermission Granting AllPermission should be done with extreme care, as it implies all other permissions. Thus, it grants code the ability to run with security disabled. Extreme caution should be taken before granting such a permission to code. This permission should be used only during testing, or in extremely rare cases where an application or applet is completely trusted and adding the necessary permissions to the policy is prohibitively cumbersome.

class StructuredViewerInternals @noinstantiate This class is not intended to be instantiated by clients.

abstract class Dictionary NOTE: This class is obsolete. New implementations should implement the Map interface, rather than extending this class.

interface PreferencesFactory This class is for Preferences implementers only. Normal users of the Preferences facility should have no need to consult this documentation.

class Stack A more complete and consistent set of LIFO stack operations is provided by the @link Deque interface and its implementations, which should be used in preference to this class.

abstract class AbstractList For sequential access data (such as a linked list), @link AbstractSequentialList should be used in preference to this class.

class LogRecord When a LogRecord is passed into the logging framework it logically belongs to the framework and should no longer be used or updated by the client application.

abstract class AbstractPreferences This class is for Preferences implementers only. Normal users of the Preferences facility should have no need to consult this documentation.

class ConcurrentModificationException Therefore, it would be wrong to write a program that depended on this exception for its correctness: ConcurrentModificationException should be used only to detect bugs.

class ArrayDeque Therefore, it would be wrong to write a program that depended on this exception for its correctness: the fail-fast behavior of iterators should be used only to detect bugs.

class HashMap Therefore, it would be wrong to write a program that depended on this exception for its correctness: the fail-fast behavior of iterators should be used only to detect bugs.

class Vector Therefore, it would be wrong to write a program that depended on this exception for its correctness: the fail-fast behavior of iterators should be used only to detect bugs.

class WeakHashMap Therefore, it would be wrong to write a program that depended on this exception for its correctness: the fail-fast behavior of iterators should be used only to detect bugs.
class LinkedHashSet Therefore, it would be wrong to write a program that depended on this exception for its correctness: the fail-fast behavior of iterators should be used only to detect bugs.
class LinkedHashMap Therefore, it would be wrong to write a program that depended on this exception for its correctness: the fail-fast behavior of iterators should be used only to detect bugs.
class IdentityHashMap Therefore, it would be wrong to write a program that depended on this exception for its correctness: fail-fast iterators should be used only to detect bugs.
class Hashtable Therefore, it would be wrong to write a program that depended on this exception for its correctness: the fail-fast behavior of iterators should be used only to detect bugs.
class TreeSet Therefore, it would be wrong to write a program that depended on this exception for its correctness: the fail-fast behavior of iterators should be used only to detect bugs.
class HashSet Therefore, it would be wrong to write a program that depended on this exception for its correctness: the fail-fast behavior of iterators should be used only to detect bugs.
class TreeMap Therefore, it would be wrong to write a program that depended on this exception for its correctness: the fail-fast behavior of iterators should be used only to detect bugs.
class LinkedList Therefore, it would be wrong to write a program that depended on this exception for its correctness: the fail-fast behavior of iterators should be used only to detect bugs.
class ArrayList Therefore, it would be wrong to write a program that depended on this exception for its correctness: the fail-fast behavior of iterators should be used only to detect bugs.
interface Skeleton @deprecated no replacement. Skeletons are no longer required for remote method calls in the Java 2 platform v1.2 and greater.
class SkeletonMismatchException @deprecated no replacement. Skeletons are no longer required for remote method calls in the Java 2 platform v1.2 and greater.
class FieldAssistColors @deprecated As of 3.3, this class is no longer necessary.
class ImageAndMessageArea @deprecated As of 3.3, this class is no longer necessary.
class LineNumberInputStream @deprecated This class incorrectly assumes that bytes adequately represent characters. As of JDK 1.1, the preferred way to operate on character streams is via the new character-stream classes, which include a class for counting line numbers.
interface LoaderHandler It should never be accessed by application code.
class ToolBarManager This class may be instantiated;
class ToolBarContributionItem This class may be instantiated;
class StatusLineManager This class may be instantiated;
class StatusLineContributionItem This class may be instantiated;
class Separator This class may be instantiated
class MenuManager This class may be instantiated;
interface ICoolBarManager @link CoolBarManager, which clients may instantiate or subclass.
class GroupMarker This class may be instantiated;
class CoolBarManager This class may be instantiated;
class ActionContributionItem This class may be instantiated;
class ProgressMonitorDialog This concrete dialog class can be instantiated as is
class MessageDialog This concrete dialog class can be instantiated as is
class InputDialog This concrete dialog class can be instantiated as is
class ControlDecoration This class is intended to be instantiated and used by clients.
class BrowserInformationControl This class may be instantiated;
class StringFieldEditor This class may be used as is
class ScaleFieldEditor This class may be used as is
class FontRegistry Clients may instantiate this class
class ColorRegistry Clients may instantiate this class
class TemplateReaderWriter Clients may instantiate this class
class TemplatePersistenceData Clients may use this class, although this is not usually needed except when implementing a custom template preference page or template store.
interface IInformationControl Clients can implement this interface and its extension interfaces, subclass @link AbstractInformationControl, or use the (text-based) default implementation @link DefaultInformationControl.
class TreePath Clients may instantiate this class.
class StyledString This class may be instantiated;
class ColumnWeightData This class may be instantiated;
class ComboBoxCellEditor This class may be instantiated;
class TextCellEditor This class may be instantiated;
class CheckboxCellEditor This class may be instantiated;
Dialog.getButtonBar() Clients may call this framework method
AbstractGroupMarker.AbstractGroupMarker() Constructor for use by subclasses.
ContentProposalAdapter.closeProposalPopup() This method is used by subclasses to explicitly close the popup based on additional logic.
AbstractInformationControlManager.DEBUG Subclasses may use this.
abstract class ViewerDropAdapter The setFeedbackEnabled method can be called to turn on and off visual insertion feedback (on by default).
Window.getConstrainedShellBounds(preferredSize) This is used for computing the initial window position, and subclasses can use this as a utility method if they want to limit the region in which the window may be moved.
OwnerDrawLabelProvider.setOwnerDrawEnabled(viewer, column, enabled) This method may be called from subclasses to enable or disable owner draw dynamically.
Dialog.convertHeightInCharsToPixels(chars) Clients may call this framework method
Dialog.getDialogArea() Clients may call this framework method
FieldEditor.convertHorizontalDLUsToPixels(control, dlus) Clients may call this framework method
FieldEditor.convertVerticalDLUsToPixels(control, dlus) Clients may call this framework method
CompositeImageDescriptor.drawImage(src, ox, oy) Call this internal framework method to superimpose another image atop this composite image.

1.10 Exception Raising Directive

class RandomAccessFile It is generally true of all the reading routines in this class that if end-of-file is reached before the desired number of bytes has been read, an EOFException (which is a kind of IOException) is thrown.
FileInputStream.finalize() @exception IOException if an I/O error occurs.
ZipOutputStream.closeEntry() @exception ZipException if a ZIP format error has occurred
ZipOutputStream.closeEntry() @exception IOException if an I/O error has occurred
(abstract) SocketImpl.bind(host, port) @exception IOException if an I/O error occurs when binding this socket.
(abstract) CallableStatement.getObject(parameterIndex, map) @exception SQLException if the parameterIndex is not valid; if a database access error occurs or this method is called on a closed CallableStatement
(abstract) CallableStatement.getObject(parameterIndex, map) @exception SQLFeatureNotSupportedException if the JDBC driver does not support this method
InflaterOutputStream.write(b, off, len) @throws IOException if an I/O error occurs or this stream is already closed
InflaterOutputStream.write(b, off, len) @throws ZipException if a compression (ZIP) format error occurs
PipedInputStream.close() @exception IOException if an I/O error occurs.
(abstract) Blob.free() @throws SQLException if an error occurs releasing the Blob's resources @exception SQLFeatureNotSupportedException if the JDBC driver does not support this method
AlgorithmParameters.getEncoded(format) @exception IOException on encoding errors, or if this parameter object has not been initialized.
(abstract) Statement.setEscapeProcessing(enable) @exception SQLException if a database access error occurs or this method is called on a closed Statement
(abstract) RemoteCall.getResultStream(success) @exception java.io.IOException if an I/O error occurs. @exception java.io.StreamCorruptedException If already been called.
ServerSocket.implAccept(s) @throws IOException if an I/O error occurs when waiting for a connection.
(abstract) DatabaseMetaData.getSuperTypes(catalog, schemaPattern, typeNamePattern) @throws SQLException if a database access error occurs

(abstract) ResultSet.updateObject(columnIndex, x, scaleOrLength) @exception SQLException if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR_READ_ONLY or this method is called on a closed result set @exception SQLFeatureNotSupportedException if the JDBC driver does not support this method

MulticastSocket.joinGroup(mcastaddr, netIf) @exception IOException if there is an error joining or when the address is not a multicast address.

NetworkInterface.getByInetAddress(addr) @throws SocketException If an I/O error occurs.

DatagramSocket.receive(p) @exception IOException if an I/O error occurs. @exception SocketTimeoutException if setSoTimeout was previously called and the timeout has expired. @exception PortUnreachableException may be thrown if the socket is connected to a currently unreachable destination. Note, there is no guarantee that the exception will be thrown. @exception java.nio.channels.IllegalBlockingModeException if this socket has an associated channel, and the channel is in non-blocking mode

(abstract) CallableStatement.setNString(parameterName, value) @throws SQLException if parameterName does not correspond to a named parameter; if the driver does not support national character sets; if the driver can detect that a data conversion error could occur; if a database access error occurs or this method is called on a closed CallableStatement @exception SQLFeatureNotSupportedException if the JDBC driver does not support this method

(abstract) CallableStatement.getBlob(parameterIndex) @exception SQLException if the parameterIndex is not valid; if a database access error occurs or this method is called on a closed CallableStatement @exception SQLFeatureNotSupportedException if the JDBC driver does not support this method

ServerSocket.getReuseAddress() @exception SocketException if there is an error in the underlying protocol, such as a TCP error.

(abstract) DataOutput.writeInt(v) @throws IOException if an I/O error occurs.

(abstract) DatabaseMetaData.ownInsertsAreVisible(type) @exception SQLException if a database access error occurs

(abstract) ParameterMetaData.getParameterClassName(param) @exception SQLException if a database access error occurs

LineNumberInputStream.reset() @exception IOException if an I/O error occurs.

RandomAccessFile.readFloat() @exception IOException if an I/O error occurs.

PipedInputStream.PipedInputStream(src, pipeSize) @exception IOException if an I/O error occurs

Socket.setOOBInline(on) @exception SocketException if there is an error in the underlying protocol, such as a TCP error.

ObjectInputStream.readShort() @throws EOFException If end of file is reached. @throws IOException If other I/O error has occurred.

PipedReader.PipedReader(src, pipeSize) @exception IOException if an I/O error occurs.

DatagramSocket.setSendBufferSize(size) @exception SocketException if there is an error in the underlying protocol, such as an UDP error.

AbstractQueue.addAll(c) @throws IllegalStateException if not all the elements can be added at this time due to insertion restrictions

AbstractQueue.add(e) @throws IllegalStateException if the element cannot be added at this time due to capacity restrictions @throws ClassCastException if the class of the specified element prevents it from being added to this queue @throws NullPointerException if the specified element is null and this queue does not permit null elements @throws IllegalArgumentException if some property of this element prevents it from being added to this queue

System.getenv(name) If a security manager exists, its @link SecurityManager#checkPermission checkPermission method is called with a @link RuntimePermission(”getenv.”+name) permission. This may result in a @link SecurityException being thrown. If no exception is thrown the value of the variable name is returned.

File.setWritable(writable) @throws SecurityException If a security manager exists and its @link java.lang.SecurityManager#checkWrite(java.lang.String) method denies write access to the file

System.load(filename) @param filename the file to load. @exception SecurityException if a security manager exists and its checkLink method doesn’t allow loading of the specified dynamic library @exception UnsatisfiedLinkError if the file does not exist.

class PrintStream Unlike other output streams, a PrintStream never throws an IOException; instead, exceptional situations merely set an internal flag that can be tested via the checkError method.

LinkedBlockingDeque.iterator() The returned Iterator is a ”weakly consistent” iterator that will never throw @link ConcurrentModificationException, and guarantees to traverse elements as they existed upon construction of the iterator, and may (but is not guaranteed to) reflect any modifications subsequent to construction.

ArrayBlockingQueue.iterator() The returned Iterator is a ”weakly consistent” iterator that will never throw @link ConcurrentModificationException, and guarantees to traverse elements as they existed upon construction of the iterator, and may (but is not guaranteed to) reflect any modifications subsequent to construction.

ConcurrentSkipListMap.values() The view’s iterator is a ”weakly consistent” iterator that will never throw @link ConcurrentModificationException, and guarantees to traverse elements as they existed upon construction of the iterator, and may (but is not guaranteed to) reflect any modifications subsequent to construction.

ConcurrentHashMap.entrySet() The view’s iterator is a ”weakly consistent” iterator that will never throw @link ConcurrentModificationException, and guarantees to traverse elements as they existed upon construction of the iterator, and may (but is not guaranteed to) reflect any modifications subsequent to construction.

ConcurrentHashMap.keySet() The view’s iterator is a ”weakly consistent” iterator that will never throw @link ConcurrentModificationException, and guarantees to traverse elements as they existed upon construction of the iterator, and may (but is not guaranteed to) reflect any modifications subsequent to construction.

(abstract) DataInput.skipBytes(n) This method never throws an EOFException.

Console.format(fmt, args) @throws IllegalFormatException If a format string contains an illegal syntax, a format specifier that is incompatible with the given arguments, insufficient arguments given the format string, or other illegal conditions. For specification of all possible formatting errors, see the Details section of the formatter class specification.

Class.newInstance() @exception SecurityException If a security manager, s, is present and any of the following conditions is met

AbstractQueuedLongSynchronizer.getWaitQueueLength(condition) @throws IllegalMonitorStateException if exclusive synchronization is not held @throws IllegalArgumentException if the given condition is not associated with this synchronizer @throws NullPointerException if the condition is null

PrintWriter.format(format, args) @throws IllegalFormatException If a format string contains an illegal syntax, a format specifier that is incompatible with the given arguments, insufficient arguments given the format string, or other illegal conditions. For specification of all possible formatting errors, see the Details section of the Formatter class specification.

PrintWriter.format(format, args) @throws NullPointerException If the format is null

PrintWriter.format(l, format, args) @throws IllegalFormatException If a format string contains an illegal syntax, a format specifier that is incompatible with the given arguments, insufficient arguments given the format string, or other illegal conditions. For specification of all possible formatting errors, see the Details section of the formatter class specification.

ReentrantReadWriteLock.getWaitQueueLength(condition) @throws IllegalMonitorStateException if this lock is not held @throws IllegalArgumentException if the given condition is not associated with this lock @throws NullPointerException if the condition is null

Class.getMethod(name, parameterTypes) @exception SecurityException If a security manager, s, is present and any of the following conditions is met

Class.getClasses() @exception SecurityException If a security manager, s, is present and any of the following conditions is met

PrintStream.printf(l, format, args) @throws IllegalFormatException If a format string contains an illegal syntax, a format specifier that is incompatible with the given arguments, insufficient arguments given the format string, or other illegal conditions. For specification of all possible formatting errors, see the Details section of the formatter class specification.

Class.getConstructors() @exception SecurityException If a security manager, s, is present and any of the following conditions is met:

invocation of @link SecurityManager#checkMemberAccess s.checkMemberAccess(this, Member.PUBLIC) denies access to the constructors within this class the caller’s class loader is not the same as or an ancestor of the class loader for the current class and invocation of @link SecurityManager#checkPackageAccess s.checkPackageAccess() denies access to the package of this class

Enum.clone() Throws CloneNotSupportedException. This guarantees that enums are never cloned, which is necessary to preserve their ”singleton” status. @return (never returns)

AbstractSortedMapDecorator.AbstractSortedMapDecorator(map) @throws IllegalArgumentException if the collection is null

FixedOrderComparator.compare(obj1, obj2) l @throws IllegalArgumentException if obj1 or obj2 are not known to this Comparator and an alternative behavior has not been set via @link #setUnknownObjectBehavior(UnknownObjectBehavior).

AbstractLinkedList.LinkedListIterator.checkModCount() @throws ConcurrentModificationException If the list’s modification count isn’t the value that was expected.

(abstract) Closure.execute(input) @throws ClassCastException (runtime) if the input is the wrong class @throws IllegalArgumentException (runtime) if the input is invalid @throws FunctorException (runtime) if any other error occurs

ExtendedProperties.load(input, enc) @throws IOException if an IO error occurs

ExtendedProperties.save(output, header) @throws IOException if an IO error occurs

ExtendedProperties.load(input) @throws IOException if an IO error occurs

ExtendedProperties.ExtendedProperties(file) @throws IOException if a file error occurs

ExtendedProperties.ExtendedProperties(file, defaultFile) @throws IOException if a file error occurs

IteratorUtils.arrayListIterator(array, start, end) @throws IllegalArgumentException if end is before start @throws NullPointerException if array is null

ClassLoader.getSystemClassLoader() @throws SecurityException If a security manager exists and its checkPermission method doesn’t allow access to the system class loader. @throws IllegalStateException If invoked recursively during the construction of the class loader specified by the ”java.system.class.loader” property. @throws Error If the system property ”java.system.class.loader” is defined but the named class could not be loaded, the provider class does not define the required constructor, or an exception is thrown by that constructor when it is invoked. The underlying cause of the error can be retrieved via the @link Throwable#getCause() method.

File.getAbsolutePath() @throws SecurityException If a required system property value cannot be accessed. @see java.io.File#isAbsolute()

File.getAbsoluteFile() @throws SecurityException If a required system property value cannot be accessed.

File.getCanonicalPath() @throws IOException If an I/O error occurs, which is possible because the construction of the canonical pathname may require filesystem queries @throws SecurityException If a required system property value cannot be accessed, or if a security manager exists and its @link java.lang.SecurityManager#checkRead method denies read access to the file

File.getCanonicalFile() @throws IOException If an I/O error occurs, which is possible because the construction of the canonical pathname may require filesystem queries @throws SecurityException If a required system property value cannot be accessed, or if a security manager exists and its @link java.lang.SecurityManager#checkRead method denies read access to the file

File.toURI() @throws SecurityException If a required system property value cannot be accessed.

ObjectOutputStream.writeUTF(str) @throws IOException if I/O errors occur while writing to the underlying stream

Formatter.format(l, format, args) @throws IllegalFormatException If a format string contains an illegal syntax, a format specifier that is incompatible with the given arguments, insufficient arguments given the format string, or other illegal conditions. For specification of all possible formatting errors, see the Details section of the formatter class specification. @throws FormatterClosedException If this formatter has been closed by invoking its @link #close() method

(abstract) Instrumentation.redefineClasses(definitions) @throws java.lang.instrument.UnmodifiableClassException if a specified class cannot be modified (@link #isModifiableClass would return false) @throws java.lang.UnsupportedOperationException if the current configuration of the JVM does not allow redefinition (@link #isRedefineClassesSupported is false) or the redefinition attempted to make unsupported changes @throws java.lang.ClassFormatError if the data did not contain a valid class @throws java.lang.NoClassDefFoundError if the name in the class file is not equal to the name of the class @throws java.lang.UnsupportedClassVersionError if the class file version numbers are not supported @throws java.lang.ClassCircularityError if the new classes contain a circularity @throws java.lang.LinkageError if a linkage error occurs @throws java.lang.NullPointerException if the supplied definitions array or any of its components is null @throws java.lang.ClassNotFoundException Can never be thrown (present for compatibility reasons only)

DataOutputStream.writeUTF(str) . @exception IOException if an I/O error occurs.

RandomAccessFile.read(b, off, len) @exception NullPointerException If b is null. @exception IndexOutOfBoundsException If off is negative, len is negative, or len is greater than b.length - off

PushbackReader.read(cbuf, off, len) @exception IOException If an I/O error occurs

PushbackReader.read() @exception IOException If an I/O error occurs

(abstract) Pack200.Packer.pack(in, out) @exception IOException if an error is encountered.

OutputStream.close() @exception IOException if an I/O error occurs.

(abstract) Externalizable.readExternal(in) @exception IOException if I/O errors occur @exception ClassNotFoundException If the class for an object being restored cannot be found.

(abstract) Struct.getAttributes(map) @exception SQLException if a database access error occurs @exception SQLFeatureNotSupportedException if the JDBC driver does not support this method

(abstract) ResultSet.insertRow() @exception SQLException if a database access error occurs; the result set concurrency is CONCUR_READ_ONLY, this method is called on a closed result set, if this method is called when the cursor is not on the insert row, or if not all of non-nullable columns in the insert row have been given a non-null value @exception SQLFeatureNotSupportedException if the JDBC driver does not support this method

OutputStream.flush() @exception IOException if an I/O error occurs.

SequenceInputStream.close() @exception IOException if an I/O error occurs.

(abstract) CallableStatement.getShort(parameterIndex) @exception SQLException if the parameterIndex is not valid; if a database access error occurs or this method is called on a closed CallableStatement

FilterWriter.flush() @exception IOException If an I/O error occurs

(abstract) ResultSet.updateBinaryStream(columnLabel, x, length) @exception SQLException if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR_READ_ONLY or this method is called on a closed result set @exception SQLFeatureNotSupportedException if the JDBC driver does not support this method

Reader.reset() @exception IOException If the stream has not been marked, or if the mark has been invalidated, or if the stream does not support reset(), or if some other I/O error occurs

(abstract) DatabaseMetaData.supportsTransactions() @exception SQLException if a database access error occurs

(abstract) ResultSet.updateCharacterStream(columnIndex, x, length) @exception SQLException if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR_READ_ONLY or this method is called on a closed result set @exception SQLFeatureNotSupportedException if the JDBC driver does not support this method

ModalContext.run(operation, fork, monitor, display) @exception InvocationTargetException if the run method must propagate a checked exception, it should wrap it inside an InvocationTargetException; runtime exceptions and errors are automatically wrapped in an InvocationTargetException by this method @exception InterruptedException if the operation detects a request to cancel, using IProgressMonitor.isCanceled(), it should exit by throwing InterruptedException; this method propagates the exception

Class.getField(name) @exception NoSuchFieldException if a field with the specified name is not found. @exception NullPointerException if name is null @exception SecurityException If a security manager, s, is present and any of the following conditions is met:

invocation of @link SecurityManager#checkMemberAccess s.checkMemberAccess(this, Member.PUBLIC) denies access to the field the caller's class loader is not the same as or an ancestor of the class loader for the current class and invocation of @link SecurityManager#checkPackageAccess s.checkPackageAccess() denies access to the package of this class

Class.getFields() @exception SecurityException If a security manager, s, is present and any of the following conditions is met:

invocation of @link SecurityManager#checkMemberAccess s.checkMemberAccess(this, Member.PUBLIC) denies access to the fields within this class the caller's class loader is not the same as or an ancestor of the class loader for the current class and invocation of @link SecurityManager#checkPackageAccess s.checkPackageAccess() denies access to the package of this class

DataInputStream.readByte() @exception EOFException if this input stream has reached the end. @exception IOException the stream has been closed and the contained input stream does not support reading after close, or another I/O error occurs.

DataInputStream.readChar() @exception EOFException if this input stream reaches the end before reading two bytes. @exception IOException the stream has been closed and the contained input stream does not support reading after close, or another I/O error occurs. @see java.io.FilterInputStream#in

(abstract) AbstractTableViewer.doRemove(indices) @exception IllegalArgumentException - if the array is null, or if any of the indices is not between 0 and the number of elements in the list minus 1 (inclusive)

(abstract) AbstractTableViewer.doGetItem(index) @exception IllegalArgumentException - if the index is not between 0 and the number of elements in the list minus 1 (inclusive)

DataInputStream.readFully(b, off, len) @exception EOFException if this input stream reaches the end before reading all the bytes. @exception IOException the stream has been closed and the contained input stream does not support reading after close, or another I/O error occurs.

DataInputStream.readUTF() @exception EOFException if this input stream reaches the end before reading all the bytes. @exception IOException the stream has been closed and the contained input stream does not support reading after close, or another I/O error occurs. @exception UTFDataFormatException if the bytes do not represent a valid modified UTF-8 encoding of a string.

(abstract) Statement.setQueryTimeout(seconds) @exception SQLException if a database access error occurs, this method is called on a closed Statement or the condition seconds ¿= 0 is not satisfied

class RemoteException Each method of a remote interface, an interface that extends java.rmi.Remote, must list RemoteException in its throws clause.

(abstract) Map.Entry.getKey() @throws IllegalStateException implementations may, but are not required to, throw this exception if the entry has been removed from the backing map.

abstract class PolicySpi This constructor also must throw an IllegalArgumentException if it does not understand the Policy.Parameters input.

TreeSet.TreeSet() all such elements must be mutually comparable: e1.compareTo(e2) must not throw a ClassCastException for any elements e1 and e2 in the set

TreeSet.TreeSet(c) all such elements must be mutually comparable: e1.compareTo(e2) must not throw a ClassCastException for any elements e1 and e2 in the set.

class ScriptException Checked exception types thrown by underlying scripting implementations must be wrapped in instances of ScriptException.

AbstractQueuedSynchronizer.tryAcquire(arg) This exception must be thrown in a consistent fashion for synchronization to work correctly.

AbstractQueuedSynchronizer.tryAcquireShared(arg) This exception must be thrown in a consistent fashion for synchronization to work correctly.

AbstractQueuedLongSynchronizer.tryAcquireShared(arg) @throws IllegalMonitorStateException if acquiring would place this synchronizer in an illegal state. This exception must be thrown in a consistent fashion for synchronization to work correctly.

(abstract) Comparator.compare(o1, o2) This implies that compare(x, y) must throw an exception if and only if compare(y, x) throws an exception.

AbstractQueuedSynchronizer.tryRelease(arg) This exception must be thrown in a consistent fashion for synchronization to work correctly.

AbstractQueuedSynchronizer.tryReleaseShared(arg) This exception must be thrown in a consistent fashion for synchronization to work correctly.

(abstract) IContentAssistSubjectControl.getLineHeight() @throws org.eclipse.swt.SWTException in these cases:

* @link org.eclipse.swt.SWT#ERROR_WIDGET_DISPOSED - if the receiver has been disposed * @link org.eclipse.swt.SWT#ERROR_THREAD_INVALID_ACCESS - if not called from the thread that created the receiver

(abstract) IContentAssistSubjectControl.getCaretOffset() @throws org.eclipse.swt.SWTException in these cases:

* @link org.eclipse.swt.SWT#ERROR_WIDGET_DISPOSED - if the receiver has been disposed * @link org.eclipse.swt.SWT#ERROR_THREAD_INVALID_ACCESS - if not called from the thread that created the receiver

(abstract) IContentAssistSubjectControl.getLocationAtOffset(offset) @throws org.eclipse.swt.SWTException in these cases:

* @link org.eclipse.swt.SWT#ERROR_WIDGET_DISPOSED - if the receiver has been disposed * @link org.eclipse.swt.SWT#ERROR_THREAD_INVALID_ACCESS - if not called from the thread that created the receiver

(abstract) IContentAssistSubjectControl.getLineDelimiter() @throws org.eclipse.swt.SWTException in these cases:

* @link org.eclipse.swt.SWT#ERROR_WIDGET_DISPOSED - if the receiver has been disposed * @link org.eclipse.swt.SWT#ERROR_THREAD_INVALID_ACCESS - if not called from the thread that created the receiver

(abstract) IContentAssistSubjectControl.getWidgetSelectionRange() @throws org.eclipse.swt.SWTException in these cases:

* @link org.eclipse.swt.SWT#ERROR_WIDGET_DISPOSED - if the receiver has been disposed * @link org.eclipse.swt.SWT#ERROR_THREAD_INVALID_ACCESS - if not called from the thread that created the receiver

(abstract) IContentAssistSubjectControl.addKeyListener(keyListener) @throws org.eclipse.swt.SWTException in these cases:

* @link org.eclipse.swt.SWT#ERROR_WIDGET_DISPOSED - if the receiver has been disposed * @link org.eclipse.swt.SWT#ERROR_THREAD_INVALID_ACCESS - if not called from the thread that created the receiver

(abstract) IContentAssistSubjectControl.removeKeyListener(keyListener) @throws org.eclipse.swt.SWTException in these cases:

* @link org.eclipse.swt.SWT#ERROR_WIDGET_DISPOSED - if the receiver has been disposed * @link org.eclipse.swt.SWT#ERROR_THREAD_INVALID_ACCESS - if not called from the thread that created the receiver

(abstract) IContentAssistSubjectControl.removeSelectionListener(selectionListener) @exception org.eclipse.swt.SWTException

o @link org.eclipse.swt.SWT#ERROR_WIDGET_DISPOSED - if the receiver has been disposed o @link org.eclipse.swt.SWT#ERROR_THREAD_INVALID_ACCESS - if not called from the thread that created the receiver

(abstract) IContentAssistSubjectControl.addSelectionListener(selectionListener) # @link org.eclipse.swt.SWT#ERROR_WIDGET_DISPOSED - if the receiver has been disposed # @link org.eclipse.swt.SWT#ERROR_THREAD_INVALID_ACCESS - if not called from the thread that created the receiver

(abstract) InvocationHandler.invoke(proxy, method, args) @throws Throwable the exception to throw from the method invocation on the proxy instance. The exception's type must be assignable either to any of the exception types declared in the throws clause of the interface method or to the unchecked exception types java.lang.RuntimeException or java.lang.Error.

(abstract) RemoteRef.invoke(call) If any exception is raised during the remote invocation, invoke should take care of cleaning up the connection before raising the "user" or remote exception.

(abstract) PolicyNode.getChildren() Any attempts to modify the children of this node through the Iterator's remove method must throw an UnsupportedOperationException.

abstract class Format Any parse method that does not take a ParsePosition argument should throw ParseException when no text in the required format is at the beginning of the input text.

(abstract) PKIXCertPathChecker.init(forward) @throws CertPathValidatorException if this PKIXCertPathChecker is unable to check certificates in the specified order; it should never be thrown if the forward flag is false since reverse checking must be supported

KeyStoreSpi.engineLoad(param) @exception IllegalArgumentException if the given KeyStore.LoadStoreParameter input is not recognized @exception IOException if there is an I/O or format problem with the keystore data. If the error is due to an incorrect ProtectionParameter (e.g. wrong password) the @link Throwable#getCause cause of the IOException should be an UnrecoverableKeyException

KeyStore.load(param) @exception IllegalArgumentException if the given LoadStoreParameter input is not recognized @exception IOException if there is an I/O or format problem with the keystore data. If the error is due to an incorrect ProtectionParameter (e.g. wrong password) the @link Throwable#getCause cause of the IOException should be an UnrecoverableKeyException

(abstract) KeyStoreSpi.engineLoad(stream, password) @exception IOException if there is an I/O or format problem with the keystore data, if a password is required but not given, or if the given password was incorrect. If the error is due to a wrong password, the @link Throwable#getCause cause of the IOException should be an UnrecoverableKeyException

KeyStore.load(stream, password) @exception IOException if there is an I/O or format problem with the keystore data, if a password is required but not given, or if the given password was incorrect. If the error is due to a wrong password, the @link Throwable#getCause cause of the IOException should be an UnrecoverableKeyException

abstract class Provider Some provider implementations may encounter unrecoverable internal errors during their operation, for example a failure to communicate with a security token. A @link ProviderException should be used to indicate such errors.

(abstract) Driver.connect(url, info) The driver should throw an SQLException if it is the right driver to connect to the given URL but has trouble connecting to the database.

class NullPointerException Applications should throw instances of this class to indicate other illegal uses of the null object.

(abstract) Collection.add(e) If a collection refuses to add a particular element for any reason other than that it already contains the element, it must throw an exception (rather than returning false).

interface Comparable Note that null is not an instance of any class, and e.compareTo(null) should throw a NullPointerException even though e.equals(null) returns false.

Collections.sort(list) that is, e1.compareTo(e2) must not throw a ClassCastException for any elements e1 and e2 in the list

AbstractQueuedLongSynchronizer.tryRelease(arg) @throws IllegalMonitorStateException if releasing would place this synchronizer in an illegal state. This exception must be thrown in a consistent fashion for synchronization to work correctly.

AbstractQueuedLongSynchronizer.tryReleaseShared(arg) @throws IllegalMonitorStateException if releasing would place this synchronizer in an illegal state. This exception must be thrown in a consistent fashion for synchronization to work correctly.

ResourceBundle.Control.newBundle(baseName, locale, format, loader, reload) If a resource bundle can't be instantiated due to an unexpected error, the error must be reported by throwing an Error or Exception rather than simply returning null.

AbstractQueuedLongSynchronizer.tryAcquire(arg) This exception must be thrown in a consistent fashion for synchronization to work correctly.

class IllegalFormatException Only explicit subtypes of this exception which correspond to specific errors should be instantiated.

class TimeoutException For many such operations it is possible to return a value that indicates timeout; when that is not possible or desirable then TimeoutException should be declared and thrown.

(abstract) AbstractPreferences.getSpi(key) Generally speaking, this method should not throw an exception under any circumstances.

abstract class AbstractPreferences Generally speaking, these SPI methods should not throw exceptions.

abstract class AbstractPreferences There is one circumstance under which putSpi, removeSpi and childSpi should throw an exception: if the caller lacks sufficient privileges on the underlying operating system to perform the requested operation.

PredicatedCollection.validate(object) The predicate itself should not throw an exception, but return false to indicate that the object cannot be added.

PredicateUtils.invokerPredicate(methodName) If the input object is null, a PredicateException is thrown.

ExceptionTransformer.transform(input) @throws FunctorException always

(abstract) MultiMap.remove(key, item) @throws UnsupportedOperationException if the map is unmodifiable

Preferences.importPreferences(is) @throws InvalidPreferencesFormatException Data on input stream does not constitute a valid XML document with the mandated document type.

Properties.loadFromXML(in) @throws InvalidPropertiesFormatException Data on input stream does not constitute a valid XML document with the mandated document type.

Signature.initVerify(certificate) @exception InvalidKeyException if the public key in the certificate is not encoded properly or does not include required parameter information or cannot be used for digital signature purposes.

(abstract) Map.Entry.getValue() @throws IllegalStateException implementations may, but are not required to, throw this exception if the entry has been removed from the backing map.

(abstract) Map.Entry.setValue(value) @throws IllegalStateException implementations may, but are not required to, throw this exception if the entry has been removed from the backing map.

ObjectArrayListIterator.add(obj)

This iterator does not support modification of its backing array's size, and so will always throw an @link UnsupportedOperationException when this method is invoked.

ArrayListIterator.add(o) This iterator does not support modification of its backing collection, and so will always throw an @link UnsupportedOperationException when this method is invoked.

(abstract) ISafeRunnableRunner.run(code) Runs the runnable. All ISafeRunnableRunners must catch any exception thrown by the ISafeRunnable and pass the exception to ISafeRunnable.handleException().

interface Registry If a Registry method chooses to deny access for a given invocation, its implementation may throw @link java.rmi.AccessException, which (because it extends @link java.rmi.RemoteException) will be wrapped in a @link java.rmi.ServerException when caught by a remote client.

Socket.connect(endpoint) @throws IllegalArgumentException if endpoint is null or is a SocketAddress subclass not supported by this socket

DatagramPacket.setSocketAddress(address) @throws IllegalArgumentException if address is null or is a SocketAddress subclass not supported by this socket

MulticastSocket.leaveGroup(mcastaddr, netIf) @throws IllegalArgumentException if mcastaddr is null or is a SocketAddress subclass not supported by this socket

Field.get(obj) @exception IllegalArgumentException if the specified object is not an instance of the class or interface declaring the underlying field (or a subclass or implementor thereof)

(abstract) Collection.addAll(c) @throws IllegalStateException if not all the elements can be added at this time due to insertion restrictions

(abstract) NavigableMap.headMap(toKey, inclusive) @throws IllegalArgumentException if this map itself has a restricted range, and toKey lies outside the bounds of the range

(abstract) SortedSet.headSet(toElement) @throws IllegalArgumentException if this set itself has a restricted range, and toElement lies outside the bounds of the range

abstract class AbstractPreferences Even under these circumstances it is generally better to simply ignore the invocation and return, rather than throwing an exception

(abstract) ResultSet.updateShort(columnLabel, x) @param x the new column value @exception SQLException if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR_READ_ONLY or this method is called on a closed result set @exception SQLFeatureNotSupportedException if the JDBC driver does not support this method

(abstract) Blob.length() @exception SQLException if there is an error accessing the length of the BLOB @exception SQLFeatureNotSupportedException if the JDBC driver does not support this method

(abstract) Array.getResultSet(index, count) @exception SQLException if an error occurs while attempting to access the array @exception SQLFeatureNotSupportedException if the JDBC driver does not support this method

(abstract) ResultSet.updateBinaryStream(columnIndex, x, length) @exception SQLException if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR_READ_ONLY or this method is called on a closed result set @exception SQLFeatureNotSupportedException if the JDBC driver does not support this method

(abstract) DatabaseMetaData.getCatalogSeparator() @exception SQLException if a database access error occurs

(abstract) ParameterMetaData.isSigned(param) @exception SQLException if a database access error occurs

(abstract) CallableStatement.setDouble(parameterName, x) @exception SQLException if parameterName does not correspond to a named parameter; if a database access error occurs or this method is called on a closed CallableStatement @exception SQLFeatureNotSupportedException if the JDBC driver does not support this method

(abstract) ResultSet.updateString(columnIndex, x) @exception SQLException if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR_READ_ONLY or this method is called on a closed result set @exception SQLFeatureNotSupportedException if the JDBC driver does not support this method

(abstract) ResultSet.updateNCharacterStream(columnLabel, reader, length) @exception SQLException if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR_READ_ONLY or this method is called on a closed result set @exception SQLFeatureNotSupportedException if the JDBC driver does not support this method

(abstract) ResultSet.updateNCharacterStream(columnLabel, reader) @exception SQLException if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR_READ_ONLY or this method is called on a closed result set @exception SQLFeatureNotSupportedException if the JDBC driver does not support this method

(abstract) ResultSet.updateObject(columnLabel, x, scaleOrLength) @exception SQLException if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR_READ_ONLY or this method is called on a closed result set @exception SQLFeatureNotSupportedException if the JDBC driver does not support this method

(abstract) NavigableMap.tailMap(fromKey, inclusive) @throws IllegalArgumentException if this map itself has a restricted range, and fromKey lies outside the bounds of the range

(abstract) SortedMap.headMap(toKey) @throws NullPointerException if toKey is null and this map does not permit null keys @throws IllegalArgumentException if this map itself has a restricted range, and toKey lies outside the bounds of the range

(abstract) NavigableSet.subSet(fromElement, fromInclusive, toElement, toInclusive) @throws NullPointerException if fromElement or toElement is null and this set does not permit null elements @throws IllegalArgumentException if fromElement is greater than toElement; or if this set itself has a restricted range, and fromElement or toElement lies outside the bounds of the range.

(abstract) BlockingDeque.addFirst(e) Inserts the specified element at the front of this deque if it is possible to do so immediately without violating capacity restrictions, throwing an IllegalStateException if no space is currently available

(abstract) BlockingQueue.add(e) @throws IllegalStateException if the element cannot be added at this time due to capacity restrictions @throws ClassCastException if the class of the specified element prevents it from being added to this queue @throws NullPointerException if the specified element is null @throws IllegalArgumentException if some property of the specified element prevents it from being added to this queue

1.11 Miscellaneous Method Call Directive

TransformerUtils.nopTransformer() The input object should be immutable to maintain the contract of Transformer (although this is not checked).

PortableRemoteObject.toStub(obj) @param obj Must have been previously the target of a call to @link #exportObject.

(abstract) AppletContext.getImage(url) The url argument that is passed as an argument must specify an absolute URL.

Applet.getImage(url) The url that is passed as an argument must specify an absolute URL.

Properties.loadFromXML(in) The XML document must have the following DOCTYPE declaration

Properties.loadFromXML(in) Furthermore, the document must satisfy the properties DTD described above

Applet.getImage(url, name) The url argument must specify an absolute URL.

Array.newInstance(componentType, dimensions) The number of dimensions of the new array must not exceed the number of array dimensions supported by the implementation (typically 255).

(abstract) DatabaseMetaData.getColumnPrivileges(catalog, schema, table, columnNamePattern) @param catalog a catalog name; must match the catalog name as it is stored in the database

(abstract) DatabaseMetaData.getColumnPrivileges(catalog, schema, table, columnNamePattern) @param schema a schema name; must match the schema name as it is stored in the database; ”” retrieves those without a schema;

(abstract) DatabaseMetaData.getColumnPrivileges(catalog, schema, table, columnNamePattern) @param table a table name; must match the table name as it is stored in the database

(abstract) DatabaseMetaData.getColumnPrivileges(catalog, schema, table, columnNamePattern) @param columnNamePattern a column name pattern; must match the column name as it is stored in the database

InetAddress.getByAddress(host, addr) IPv4 address byte array must be 4 bytes long and IPv6 byte array must be 16 bytes long

class URI A URL must be absolute, that is, it must always specify a scheme

(abstract) DGC.dirty(ids, sequenceNum, lease) A client must use the VMID the garbage collector uses in order to make corresponding clean calls when the client drops remote object references

class SetUniqueList Bear in mind that Sets require immutable objects to function correctly.

Collections.min(coll) All elements in the collection must implement the Comparable interface.

Collections.binarySearch(list, key) The list must be sorted into ascending order according to the @linkplain Comparable natural ordering of its elements (as by the @link #sort(List) method) prior to making this call.

class URL The file component must be specified.

Proxy.getProxyClass(loader, interfaces) There are several restrictions on the parameters that may be passed to Proxy.getProxyClass: * All of the Class objects in the interfaces array must represent interfaces, not classes or primitive types. * No two elements in the interfaces array may refer to identical Class objects. * All of the interface types must be visible by name through the specified class loader. In other words, for class loader cl and every interface i, the following expression must be true: Class.forName(i.getName(), false, cl) == i * All non-public interfaces must be in the same package; otherwise, it would not be possible for the proxy class to implement all of the interfaces, regardless of what package it is defined in. * For any set of member methods of the specified interfaces that have the same signature: o If the return type of any of the methods is a primitive type or void, then all of the methods must have that same return type. o Otherwise, one of the methods must have a return type that is assignable to all of the return types of the rest of the methods.

MonitorInfo.from(cd) The given CompositeData must contain the following attributes as well as the attributes specified in the mapped type for the @link LockInfo class

MemoryUsage.from(cd) The given CompositeData must contain the following attributes:

MemoryNotificationInfo.from(cd) The given CompositeData must contain the following attributes:

SerializablePermission.SerializablePermission(name, actions) @param actions currently unused and must be set to null

DateFormatSymbols.setWeekdays(newWeekdays) @param newWeekdays the new weekday strings. The array should be indexed by Calendar.SUNDAY, Calendar.MONDAY, etc.

DateFormatSymbols.setShortWeekdays(newShortWeekdays) @param newShortWeekdays the new short weekday strings. The array should be indexed by Calendar.SUNDAY, Calendar.MONDAY, etc.

class ServiceLoader The file must be encoded in UTF-8.

abstract class LocaleServiceProvider The file must be encoded in UTF-8.

TransformerUtils.invokerTransformer(methodName) Gets a Transformer that invokes a method on the input object. The method must have no parameters.

SingletonMap.SingletonMap(map) @param map the map to copy, must be size 1

PredicateUtils.asPredicate(transformer) The Transformer must return either Boolean.TRUE or Boolean.FALSE otherwise a PredicateException will be thrown.

DateFormatSymbols.setZoneStrings(newZoneStrings) The argument must be a two-dimensional array of strings of size n by m, where m is at least 5. Each of the n rows is an entry containing the localized names for a single TimeZone.
URLStreamHandler.openConnection(u, p) @param p the proxy through which the connection will be made. If direct connection is desired, Proxy.NO_PROXY should be specified
URL.openConnection(proxy) @param proxy the Proxy through which this connection will be made. If direct connection is desired, Proxy.NO_PROXY should be specified.
class MulticastSocket The address 224.0.0.0 is reserved and should not be used.
MulticastSocket.send(p, ttl) @param p is the packet to be sent. The packet should contain the destination multicast ip address and the data to be sent.
ClassLoader.defineClass(name, b, off, len, protectionDomain) @param b The bytes that make up the class data. The bytes in positions off through off+len-1 should have the format of a valid class file as defined by the Java Virtual Machine Specification.
ProcessBuilder.start() Which commands are valid is system-dependent, but at the very least the command must be a non-empty list of non-null strings
Runtime.exec(cmdarray, envp, dir)
**** deprecated tag (Value Directive - Parameter: too vague): Which commands are valid is system-dependent, but at the very least the command must be a non-empty list of non-null strings.
X509CertSelector.setSubject(subjectDN) If subjectDN is not null, it should contain a single DER encoded distinguished name, as defined in X.501.
X509CRLSelector.addIssuerName(name) The name is provided as a byte array. This byte array should contain a single DER encoded distinguished name, as defined in X.501.
X509CertSelector.setIssuer(issuerDN) If issuerDN is not null, it should contain a single DER encoded distinguished name, as defined in X.501.
MultiKeyMap.MultiKeyMap(map) The map must not be null and should be empty or only contain valid keys.
X509CertSelector.setNameConstraints(bytes) The name constraints are specified as a byte array. This byte array should contain the DER encoded form of the name constraints, as they would appear in the NameConstraints structure defined in RFC 2459 and X.509.
MultiKey.MultiKey(keys, makeClone) The keys should be immutable If they are not then they must not be changed after adding to the MultiKey.
(abstract) Driver.connect(url, info) The java.util.Properties argument can be used to pass arbitrary string tag/value pairs as connection arguments. Normally at least ”user” and ”password” properties should be included in the Properties object.
Arrays.binarySearch(a, fromIndex, toIndex, key) The range must be sorted (as by the @link #sort(int[], int, int) method) prior to making this call
Arrays.binarySearch(a, fromIndex, toIndex, key) The range must be sorted (as by the @link #sort(short[], int, int) method) prior to making this call
Arrays.binarySearch(a, fromIndex, toIndex, key) The range must be sorted (as by the @link #sort(char[], int, int) method) prior to making this call.
Arrays.binarySearch(a, fromIndex, toIndex, key) The range must be sorted (as by the @link #sort(byte[], int, int) method) prior to making this call.
Arrays.binarySearch(a, key) The array must be sorted (as by the @link #sort(double[]) method) prior to making this call.
Arrays.binarySearch(a, fromIndex, toIndex, key) The range must be sorted (as by the @link #sort(double[], int, int) method) prior to making this call.
Arrays.binarySearch(a, key) The array must be sorted (as by the @link #sort(float[]) method) prior to making this call
Arrays.binarySearch(a, fromIndex, toIndex, key) The range must be sorted (as by the @link #sort(float[], int, int) method) prior to making this call.
Arrays.binarySearch(a, key) The array must be sorted into ascending order according to the @linkplain Comparable natural ordering of its elements (as by the @link #sort(Object[]) method) prior to making this call.
Arrays.binarySearch(a, key, c) The array must be sorted into ascending order according to the specified comparator (as by the @link #sort(Object[], Comparator) sort(T[], Comparator) method) prior to making this call.
Collections.binarySearch(list, key, c) The list must be sorted into ascending order according to the specified comparator (as by the @link #sort(List, Comparator) sort(List, Comparator) method), prior to making this call
ThreadInfo.from(cd) The given CompositeData must contain the following attributes
(abstract) DatabaseMetaData.getPrimaryKeys(catalog, schema, table) @param catalog a catalog name; must match the catalog name as it is stored in the database; ”” retrieves those without a catalog;
(abstract) DatabaseMetaData.getPrimaryKeys(catalog, schema, table) @param schema a schema name; must match the schema name as it is stored in the database; ”” retrieves those without a schema
(abstract) DatabaseMetaData.getPrimaryKeys(catalog, schema, table) @param table a table name; must match the table name as it is stored in the database
(abstract) DatabaseMetaData.getSchemas(catalog, schemaPattern) @param catalog a catalog name; must match the catalog name as it is stored in the database;”” retrieves those without a catalog; null means catalog name should not be used to narrow down the search.
(abstract) DatabaseMetaData.getTables(catalog, schemaPattern, tableNamePattern, types) @param catalog a catalog name; must match the catalog name as it is stored in the database; ”” retrieves those without a catalog; null means that the catalog name should not be used to narrow the search
(abstract) DatabaseMetaData.getTables(catalog, schemaPattern, tableNamePattern, types) @param tableNamePattern a table name pattern; must match the table name as it is stored in the database
(abstract) DatabaseMetaData.getCrossReference(parentCatalog, parentSchema, parentTable, foreignCatalog, foreignSchema, foreignTable) @param parentTable the name of the table that exports the key; must match the table name as it is stored in the database
(abstract) Statement.setCursorName(name) @param name the new cursor name, which must be unique within a connection
PropertyResourceBundle.PropertyResourceBundle(stream) The property file read with this constructor must be encoded in ISO-8859-1.
Collections.sort(list) The specified list must be modifiable, but need not be resizable.
Arrays.binarySearch(a, fromIndex, toIndex, key) The range must be sorted into ascending order according to the @linkplain Comparable natural ordering of its elements (as by the @link #sort(Object[], int, int) method) prior to making this call
EnumMap.EnumMap(m) If the specified map is an EnumMap instance, this constructor behaves identically to @link #EnumMap(EnumMap). Otherwise, the specified map must contain at least one mapping (in order to determine the new enum map's key type).
SingletonMap.putAll(map) @param map the map to add, must be size 0 or 1, and the key must match
(abstract) DatabaseMetaData.getCrossReference(parentCatalog, parentSchema, parentTable, foreignCatalog, foreignSchema, foreignTable) @param parentCatalog a catalog name; must match the catalog name as it is stored in the database;
(abstract) DatabaseMetaData.getCrossReference(parentCatalog, parentSchema, parentTable, foreignCatalog, foreignSchema, foreignTable) @param foreignCatalog a catalog name; must match the catalog name as it is stored in the database; ”” retrieves those without a catalog;
(abstract) DatabaseMetaData.getCrossReference(parentCatalog, parentSchema, parentTable, foreignCatalog, foreignSchema, foreignTable) @param foreignSchema a schema name; must match the schema name as it is stored in the database;
(abstract) DatabaseMetaData.getTablePrivileges(catalog, schemaPattern, tableNamePattern) @param catalog a catalog name; must match the catalog name as it is stored in the database; ”” retrieves those without a catalog; null means that the catalog name should not be used to narrow the search
(abstract) DatabaseMetaData.getTablePrivileges(catalog, schemaPattern, tableNamePattern) @param tableNamePattern a table name pattern; must match the table name as it is stored in the database @return ResultSet - each row is a table privilege description
(abstract) DatabaseMetaData.getTablePrivileges(catalog, schemaPattern, tableNamePattern) @param schemaPattern a schema name pattern; must match the schema name as it is stored in the database; ”” retrieves those without a schema; null means that the schema name should not be used to narrow the search
(abstract) DatabaseMetaData.getProcedureColumns(catalog, schemaPattern, procedureNamePattern, columnNamePattern) @param catalog a catalog name; must match the catalog name as it is stored in the database; ”” retrieves those without a catalog; null means that the catalog name should not be used to narrow the search
(abstract) DatabaseMetaData.getProcedureColumns(catalog, schemaPattern, procedureNamePattern, columnNamePattern) @param schemaPattern a schema name pattern; must match the schema name as it is stored in the database; ”” retrieves those without a schema; null means that the schema name should not be used to narrow the search
(abstract) DatabaseMetaData.getProcedureColumns(catalog, schemaPattern, procedureNamePattern, columnNamePattern) @param procedureNamePattern a procedure name pattern; must match the procedure name as it is stored in the database
(abstract) DatabaseMetaData.getProcedures(catalog, schemaPattern, procedureNamePattern) @param catalog a catalog name; must match the catalog name as it is stored in the database; ”” retrieves those without a catalog; null means that the catalog name should not be used to narrow the search
(abstract) DatabaseMetaData.getProcedures(catalog, schemaPattern, procedureNamePattern) @param schemaPattern a schema name pattern; must match the schema name as it is stored in the database; ”” retrieves those without a schema; null means that the schema name should not be used to narrow the search
(abstract) DatabaseMetaData.getProcedures(catalog, schemaPattern, procedureNamePattern) @param procedureNamePattern a procedure name pattern; must match the procedure name as it is stored in the database
(abstract) Acl.addEntry(caller, entry) @param caller the principal invoking this method. It must be an owner of this ACL.
Preferences.importPreferences(is) The XML document must have the following DOCTYPE declaration:
PredicateUtils.invokerPredicate(methodName, paramTypes, args) The method must return either a boolean or a non-null Boolean, and have no parameters.
MultiValueMap.MultiValueMap(map, collectionFactory) @param collectionFactory the collection factory which must return a Collection instance
MultiValueMap.decorate(map, collectionFactory) @param collectionFactory the collection factory (must return a Collection object).
MapUtils.putAll(map, array) If the first entry in the object array implements @link java.util.Map.Entry or @link KeyValue then the key and value are added from that object. If the first entry in the object array is an object array itself, then it is assumed that index 0 in the sub-array is the key and index 1 is the value. Otherwise, the array is treated as keys and values in alternate indices.

MapUtils.putAll(map, array) @throws ClassCastException if the array contents is mixed

MapUtils.putAll(map, array) @throws IllegalArgumentException if sub-array or entry matching used and an entry is invalid

AbstractMapBag.AbstractMapBag(map) The map must be empty and non-null.

MultiKeyMap.decorate(map) The map must not be null and must be empty.

(abstract) Handler.publish(record) The formatting should include localization.

(abstract) Instrumentation.appendToBootstrapClassLoaderSearch(jarfile) The agent should take care to ensure that the JAR does not contain any classes or resources other than those to be defined by the bootstrap class loader for the purpose of instrumentation. Failure to observe this warning could result in unexpected behaviour that is difficult to diagnose.

Arrays.binarySearch(a, key) The array must be sorted (as by the @link #sort(byte[]) method) prior to making this call

Arrays.binarySearch(a, key) The array must be sorted (as by the @link #sort(long[]) method) prior to making this call.

DriverManager.getConnection(url, info) @param info a list of arbitrary string tag/value pairs as connection arguments; normally at least a ”user” and ”password” property should be included

class PKIXBuilderParameters To instantiate a PKIXBuilderParameters object, an application must specify one or more most-trusted CAs as defined by the PKIX certification path validation algorithm.

class PKIXBuilderParameters In addition, an application must specify constraints on the target certificate that the CertPathBuilder will attempt to build a path to. The constraints are specified as a CertSelector object. These constraints should provide the CertPathBuilder with enough search criteria to find the target certificate. Minimal criteria for an X509Certificate usually include the subject name and/or one or more subject alternative names. If enough criteria is not specified, the CertPathBuilder may throw a CertPathBuilderException.

(abstract) DatabaseMetaData.getVersionColumns(catalog, schema, table) @param catalog a catalog name; must match the catalog name as it is stored in the database; ”” retrieves those without a catalog; null means that the catalog name should not be used to narrow the search

(abstract) DatabaseMetaData.getVersionColumns(catalog, schema, table) @param schema a schema name; must match the schema name as it is stored in the database; ”” retrieves those without a schema; null means that the schema name should not be used to narrow the search

(abstract) DatabaseMetaData.getVersionColumns(catalog, schema, table) @param table a table name; must match the table name as it is stored in the database

(abstract) Driver.getPropertyInfo(url, info) @param info a proposed list of tag/value pairs that will be sent on connect open @return an array of DriverPropertyInfo objects describing possible properties. This array may be an empty array if no properties are required.

(abstract) DatabaseMetaData.getFunctions(catalog, schemaPattern, functionNamePattern) @param catalog a catalog name; must match the catalog name as it is stored in the database; ”” retrieves those without a catalog; null means that the catalog name should not be used to narrow the search

(abstract) DatabaseMetaData.getFunctions(catalog, schemaPattern, functionNamePattern) @param schemaPattern a schema name pattern; must match the schema name as it is stored in the database; ”” retrieves those without a schema; null means that the schema name should not be used to narrow the search

(abstract) DatabaseMetaData.getFunctions(catalog, schemaPattern, functionNamePattern) @param functionNamePattern a function name pattern; must match the function name as it is stored in the database

(abstract) DatabaseMetaData.getBestRowIdentifier(catalog, schema, table, scope, nullable) @param catalog a catalog name; must match the catalog name as it is stored in the database; ”” retrieves those without a catalog; null means that the catalog name should not be used to narrow the search

(abstract) DatabaseMetaData.getBestRowIdentifier(catalog, schema, table, scope, nullable) @param schema a schema name; must match the schema name as it is stored in the database; ”” retrieves those without a schema; null means that the schema name should not be used to narrow the search

(abstract) DatabaseMetaData.getBestRowIdentifier(catalog, schema, table, scope, nullable) @param table a table name; must match the table name as it is stored in the database

(abstract) DatabaseMetaData.getIndexInfo(catalog, schema, table, unique, approximate) @param catalog a catalog name; must match the catalog name as it is stored in this database; ”” retrieves those without a catalog; null means that the catalog name should not be used to narrow the search

(abstract) DatabaseMetaData.getIndexInfo(catalog, schema, table, unique, approximate) @param schema a schema name; must match the schema name as it is stored in this database; ”” retrieves those without a schema; null means that the schema name should not be used to narrow the search

(abstract) DatabaseMetaData.getIndexInfo(catalog, schema, table, unique, approximate) @param table a table name; must match the table name as it is stored in this database

(abstract) DatabaseMetaData.getColumns(catalog, schemaPattern, tableNamePattern, columnNamePattern) @param catalog a catalog name; must match the catalog name as it is stored in the database; ”” retrieves those without a catalog; null means that the catalog name should not be used to narrow the search

(abstract) DatabaseMetaData.getColumns(catalog, schemaPattern, tableNamePattern, columnNamePattern) @param schemaPattern a schema name pattern; must match the schema name as it is stored in the database; ”” retrieves those without a schema; null means that the schema name should not be used to narrow the search

(abstract) DatabaseMetaData.getColumns(catalog, schemaPattern, tableNamePattern, columnNamePattern) @param tableNamePattern a table name pattern; must match the table name as it is stored in the database

(abstract) DatabaseMetaData.getColumns(catalog, schemaPattern, tableNamePattern, columnNamePattern) @param columnNamePattern a column name pattern; must match the column name as it is stored in the database

(abstract) DatabaseMetaData.getAttributes(catalog, schemaPattern, typeNamePattern, attributeNamePattern) @param catalog a catalog name; must match the catalog name as it is stored in the database; ”” retrieves those without a schema; null means that the catalog name should not be used to narrow the search

(abstract) DatabaseMetaData.getAttributes(catalog, schemaPattern, typeNamePattern, attributeNamePattern) @param schemaPattern a schema name pattern; must match the schema name as it is stored in the database; ”” retrieves those without a schema; null means that the schema name should not be used to narrow the search

(abstract) DatabaseMetaData.getAttributes(catalog, schemaPattern, typeNamePattern, attributeNamePattern) @param typeNamePattern a type name pattern; must match the type name as it is stored in the database

(abstract) DatabaseMetaData.getAttributes(catalog, schemaPattern, typeNamePattern, attributeNamePattern) @param attributeNamePattern an attribute name pattern; must match the attribute name as it is declared in the database

(abstract) DatabaseMetaData.getImportedKeys(catalog, schema, table) @param catalog a catalog name; must match the catalog name as it is stored in the database; ”” retrieves those without a catalog; null means that the catalog name should not be used to narrow the search

(abstract) DatabaseMetaData.getImportedKeys(catalog, schema, table) @param schema a schema name; must match the schema name as it is stored in the database; ”” retrieves those without a schema; null means that the schema name should not be used to narrow the search

(abstract) DatabaseMetaData.getImportedKeys(catalog, schema, table) @param table a table name; must match the table name as it is stored in the database

KeyStore.PrivateKeyEntry.KeyStore.PrivateKeyEntry(privateKey, chain) @param chain an array of Certificates representing the certificate chain. The chain must be ordered and contain a Certificate at index 0 corresponding to the private key.

Arrays.binarySearch(a, fromIndex, toIndex, key, c) The range must be sorted into ascending order according to the specified comparator (as by the @link #sort(Object[], int, int, Comparator) sort(T[], int, int, Comparator) method) prior to making this call

Arrays.binarySearch(a, key) The array must be sorted (as by the @link #sort(short[]) method) prior to making this call

ClassLoader.defineClass(name, b, off, len) @param b The bytes that make up the class data. The bytes in positions off through off+len-1 should have the format of a valid class file as defined by the Java Virtual Machine Specification.

EnumSet.copyOf(c) Otherwise, the specified collection must contain at least one element (in order to determine the new enum set’s element type)

(abstract) DatabaseMetaData.getUDTs(catalog, schemaPattern, typeNamePattern, types) @param catalog a catalog name; must match the catalog name as it is stored in the database; ”” retrieves those without a catalog; null means that the catalog name should not be used to narrow the search

(abstract) DatabaseMetaData.getUDTs(catalog, schemaPattern, typeNamePattern, types) @param schemaPattern a schema pattern name; must match the schema name as it is stored in the database; ”” retrieves those without a schema; null means that the schema name should not be used to narrow the search

(abstract) DatabaseMetaData.getUDTs(catalog, schemaPattern, typeNamePattern, types) @param typeNamePattern a type name pattern; must match the type name as it is stored in the database; may be a fully qualified name

(abstract) DatabaseMetaData.getFunctionColumns(catalog, schemaPattern, functionNamePattern, columnNamePattern) @param catalog a catalog name; must match the catalog name as it is stored in the database; ”” retrieves those without a catalog; null means that the catalog name should not be used to narrow the search

(abstract) DatabaseMetaData.getFunctionColumns(catalog, schemaPattern, functionNamePattern, columnNamePattern) @param schemaPattern a schema name pattern; must match the schema name as it is stored in the database; ”” retrieves those without a schema; null means that the schema name should not be used to narrow the search

(abstract) DatabaseMetaData.getFunctionColumns(catalog, schemaPattern, functionNamePattern, columnNamePattern) @param functionNamePattern a procedure name pattern; must match the function name as it is stored in the database

(abstract) DatabaseMetaData.getFunctionColumns(catalog, schemaPattern, functionNamePattern, columnNamePattern) @param columnNamePattern a parameter name pattern; must match the parameter or column name as it is stored in the database

(abstract) DatabaseMetaData.getExportedKeys(catalog, schema, table) @param catalog a catalog name; must match the catalog name as it is stored in this database; ”” retrieves those without a catalog; null means that the catalog name should not be used to narrow the search

(abstract) DatabaseMetaData.getExportedKeys(catalog, schema, table) @param schema a schema name; must match the schema name as it is stored in the database; ”” retrieves those without a schema; null means that the schema name should not be used to narrow the search

(abstract) DatabaseMetaData.getExportedKeys(catalog, schema, table) @param table a table name; must match the table name as it is stored in this database

Arrays.binarySearch(a, key) The array must be sorted (as by the @link #sort(int[]) method) prior to making this call
Arrays.binarySearch(a, fromIndex, toIndex, key) The range must be sorted (as by the @link #sort(long[], int, int) method) prior to making this call
class Level Clients should normally use the predefined Level constants such as Level.SEVERE.
MapUtils.transformedSortedMap(map, keyTransformer, valueTransformer) @param map the map to transform, must not be null, typically empty
(abstract) Acl.removeEntry(caller, entry) @param caller the principal invoking this method. It must be an owner of this ACL.
ChoiceFormat.setChoices(limits, formats) @param limits contains the top value that you want parsed with that format,and should be in ascending sorted order.
(abstract) Acl.setName(caller, name) @param caller the principal invoking this method. It must be an owner of this ACL.
Arrays.binarySearch(a, key) The array must be sorted (as by the @link #sort(char[]) method) prior to making this call
CertificateFactory.generateCertPath(certificates) The certificates supplied must be of a type supported by the CertificateFactory
Collections.newSetFromMap(map) The specified map must be empty at the time this method is invoked, and should not be accessed directly after this method returns.
Timer.schedule(task, delay, period) Fixed-delay execution is appropriate for recurring activities that require "smoothness." In other words, it is appropriate for activities where it is more important to keep the frequency accurate in the short run than in the long run.
Timer.schedule(task, firstTime, period) Fixed-delay execution is appropriate for recurring activities that require "smoothness." In other words, it is appropriate for activities where it is more important to keep the frequency accurate in the short run than in the lon
class WeakHashMap The value objects in a WeakHashMap are held by ordinary strong references. Thus care should be taken to ensure that value objects do not strongly refer to their own keys, either directly or indirectly, since that will prevent the keys from being discarded.
(abstract) IPageChangingListener.handlePageChanging(event) The doit field of the PageChangingEvent must be set to false to prevent the page from changing.
DeferredContentProvider.setSortOrder(sortOrder) @param sortOrder new sort order. The comparator must be able to support being used in a background thread.
HyperlinkManager.install(textViewer, hyperlinkPresenter, hyperlinkDetectors, eventStateMask) @param hyperlinkDetectors the array of hyperlink detectors, must not be empty
FontDescriptor.createFrom(font, originalDevice) @param originalDevice must be the same Device that was passed into the font's constructor when it was first created.
class DeferredContentProvider Content provider that performs sorting and filtering in a background thread. Requires a TableViewer created with the SWT.VIRTUAL flag and an IConcurrentModel as input.
ApplicationWindow.run(fork, cancelable, runnable) This implementation of IRunnableContext#run(boolean, boolean, IRunnableWithProgress) blocks until the runnable has been run, regardless of the value of fork. It is recommended that fork is set to true in most cases. If fork is set to false, the runnable will run in the UI thread and it is the runnable's responsibility to call Display.readAndDispatch() to ensure UI responsiveness.
WizardDialog.run(fork, cancelable, runnable) This implementation of IRunnableContext#run(boolean, boolean, IRunnableWithProgress) blocks until the runnable has been run, regardless of the value of fork. It is recommended that fork is set to true in most cases. If fork is set to false, the runnable will run in the UI thread and it is the runnable's responsibility to call Display.readAndDispatch() to ensure UI responsiveness.
PreferenceDialog.createSash(composite, rightControl) Note that this method assumes GridData for the layout data of the rightControl
RuleBasedPartitioner.getTokenContentType(token) If the token's data is not null and a string it is assumed that it is the encoded content type.
CellLabelProvider.getToolTipShift(object) Be aware of the fact that you should at least position the tool tip 1px right to your mouse cursor else click events may not get propagated properly.
ImageDescriptor.createImage(returnMissingImageOnError) In the event of an error, a default image is returned if returnMissingImageOnError is true, otherwise null is returned.

Note: Even if returnMissingImageOnError is true, it is still possible for this method to return null in extreme cases, for example if SWT runs out of image handles.
ImageDescriptor.createImage(returnMissingImageOnError, device) In the even of an error, a default image is returned if returnMissingImageOnError is true, otherwise null is returned.
System.arraycopy(src, srcPos, dest, destPos, length) Otherwise, if any of the following is true, an ArrayStoreException is thrown and the destination is not modified:

The src argument refers to an object that is not an array. The dest argument refers to an object that is not an array. The src argument and dest argument refer to arrays whose component types are different primitive types. The src argument refers to an array with a primitive component type and the dest argument refers to an array with a reference component type. The src argument refers to an array with a reference component type and the dest argument refers to an array with a primitive component type.

Otherwise, if any of the following is true, an IndexOutOfBoundsException is thrown and the destination is not modified:

The srcPos argument is negative. The destPos argument is negative. The length argument is negative. srcPos+length is greater than src.length, the length of the source array. destPos+length is greater than dest.length, the length of the destination array.

Because of the restrictions already itemized, this paragraph effectively applies only to the situation where both arrays have component types that are reference types.
AbstractQueuedLongSynchronizer.tryAcquire(arg) @param arg the acquire argument. This value is always the one passed to an acquire method, or is the value saved on entry to a condition wait. The value is otherwise uninterpreted and can represent anything you like.
BigInteger.BigInteger(val) The input array is assumed to be in big-endian byte-order: the most significant byte is in the zeroth element.
Dialog.createButton(parent, id, label, defaultButton) Note that the parent's layout is assumed to be a GridLayout and the number of columns in this layout is incremented.
AbstractTreeViewer.createChildren(widget) The default implementation of this framework method assumes that widget.getData() returns the element corresponding to the node.
(abstract) ISourceViewer.setAnnotationHover(annotationHover) The annotation hover provides the information to be displayed in a hover popup window if requested over the annotation presentation area. The annotation hover is assumed to be line oriented.
IconAndMessageDialog.createMessageArea(composite) The parent composite is assumed to use GridLayout as its layout manager, since the parent is typically the composite created in @link Dialog#createDialogArea.
BufferedInputStream.markpos If there is a marked position in the input stream, then buf[markpos] is the first byte to be supplied as input after a reset operation.
Preferences.userNodeForPackage(c) If the class name is misspelled, it will typically result in a compile-time error.
DefaultDamagerRepairer.getTokenTextAttribute(token) If the token's data is not null and a text attribute it is assumed that it is the encoded text attribute.
DefaultDamagerRepairer.DefaultDamagerRepairer(scanner) The scanner is assumed to return only token that carry text attributes.
SourceViewer.setOverviewRulerAnnotationHover(annotationHover) The annotation hover is assumed to be line oriented.
DelayQueue.offer(e, timeout, unit) @param timeout This parameter is ignored as the method never blocks
DelayQueue.offer(e, timeout, unit) @param unit This parameter is ignored as the method never blocks
class FileReader The constructors of this class assume that the default character encoding and the default byte-buffer size are appropriate.
(abstract) Acl.checkPermission(principal, permission) @param principal the principal, assumed to be a valid authenticated Principal.

2 Subclassing Directive

2.1 Extensible Class Identification Directive

interface ExternalActionManager.IExecuteCallback Clients must not implement this interface and must not extend.
class LegacyActionTools Clients may neither instantiate this class nor extend.
abstract class AbstractAction Clients may neither instantiate nor extend this class.
class ParseException Clients are not permitted to extend this class.
class KeySequence Clients are not permitted to extend this class.
class KeyStroke Clients are not permitted to extend this class.
class TextState Clients may instantiate this class, but must not extend.
interface IWidget Clients may implement, but must not extend.
class ToggleState Clients may instantiate this class, but must not extend.
interface ExternalActionManager.IBindingManagerCallback Clients may implement this interface, but must not extend.
class AnnotationRulerColumn Do not subclass.
class ExternalActionManager This class is not intended to be called or extended by any external clients.
class SubContributionItem This class is not intended to be subclassed.
class StatusLineContributionItem it is not intended to be subclassed.
class Separator it is not intended to be subclassed outside the framework.
class GroupMarker @noextend This class is not intended to be subclassed by clients.
class ExternalActionManager.CommandCallback Note: this class is not intended to be subclassed by clients.
class ActionContributionItem @noextend This class is not intended to be subclassed by clients.
abstract class AbstractGroupMarker @noextend This class is not intended to be subclassed by clients.

class DialogSettings @noextend This class is not intended to be subclassed by clients.
class DecoratedField This class is not intended to be subclassed.
class ControlDecoration It is not intended to be subclassed by clients.
class BrowserInformationControl it is not intended to be subclassed.
class ModalContext This class is not intended to be subclassed.
class PreferenceStore @noextend This class is not intended to be subclassed by clients.
class StringConverter @noextend This class is not intended to be subclassed by clients.
class JFaceResources @noextend This class is not intended to be subclassed by clients.
class ImageRegistry @noextend This class is not intended to be subclassed by clients.
class FontRegistry @noextend This class is not intended to be subclassed by clients.
class ColorRegistry @noextend This class is not intended to be subclassed by clients.
class ImageUtilities @noextend This class is not intended to be subclassed by clients.
class TemplateReaderWriter @noextend This class is not intended to be subclassed by clients.
class TemplatePersistenceData @noextend This class is not intended to be subclassed by clients.
class TextViewerUndoManager @noextend This class is not intended to be subclassed by clients.
class TextViewer @noextend This class is not intended to be subclassed by clients.
class MarginPainter @noextend This class is not intended to be subclassed by clients.
class JFaceTextUtil This class is neither intended to be instantiated nor subclassed.
class DefaultUndoManager This class is not intended to be subclassed.
class DefaultIndentLineAutoEditStrategy This class is not intended to be subclassed.
class DefaultAutoIndentStrategy @noextend This class is not intended to be subclassed by clients.
class CursorLinePainter This class is not intended to be subclassed.
class LocalSelectionTransfer This class is not intended to be subclassed.
interface IConcurrentModel Not intended to be implemented by clients.
class StructuredViewerInternals @noextend This class is not intended to be subclassed by clients.
class TableTreeViewer @noextend This class is not intended to be subclassed by clients.
class TableViewer This class is not intended to be subclassed outside the viewer framework.
class TreePath Not intended to be subclassed.
class StyledString it is not intended to be subclassed.
class TreeSelection This class is not intended to be subclassed.
class StructuredSelection @noextend This class is not intended to be subclassed by clients.
class ListViewer @noextend This class is not intended to be subclassed by clients.
class DecorationContext This class is not intended to be subclassed.
class ColumnWeightData @noextend This class is not intended to be subclassed by clients.
class ColumnViewerEditorDeactivationEvent @noextend This class is not intended to be subclassed by clients.
abstract class ColumnViewer This class is not intended to be subclassed outside of the JFace viewers framework.
class ColumnPixelData @noextend This class is not intended to be subclassed by clients.
abstract class ColumnLayoutData @noextend This class is not intended to be subclassed by clients.
class CellEditor.LayoutData @noextend This class is not intended to be subclassed by clients.
class ArrayContentProvider @noextend This class is not intended to be subclassed by clients.
class CheckboxTableViewer @noextend This class is not intended to be subclassed by clients.
class ComboBoxCellEditor @noextend This class is not intended to be subclassed by clients.
class TextCellEditor @noextend This class is not intended to be subclassed by clients.
class TreeViewer @noextend This class is not intended to be subclassed by clients.
class ColorCellEditor @noextend This class is not intended to be subclassed by clients.
class CheckboxTreeViewer @noextend This class is not intended to be subclassed by clients.
class CheckboxCellEditor @noextend This class is not intended to be subclassed by clients.
class ToolBarContributionItem @noextend This class is not intended to be subclassed by clients.
class SchemeEvent This class is not intended to be extended by clients.
class Scheme This class is not intended to be extended by clients.
class BindingManagerEvent This class is not intended to be extended by clients.
interface IContextIds This interface should not be implemented or extended by clients.
class ImageAndMessageArea This class is not intended to be extended by clients.
interface StructuredViewerInternals.AssociateListener @noextend This interface is not intended to be extended by clients.
interface StructuredViewerInternals.AssociateListener @noimplement This interface is not intended to be implemented by clients.
interface IEditingSupportRegistry @noimplement This interface is not intended to be implemented by clients
interface IEditingSupportRegistry @noextend This interface is not intended to be extended by clients.
interface IContributionManagerOverrides This interface is internal to the framework; it should not be implemented outside the framework.
interface IContributionItem This interface is internal to the framework; it should not be implemented outside the framework.
interface IDecorationContext This interface is not intended to be implemented by clients
interface IDecoration @noimplement This interface is not intended to be implemented by clients.
interface IBaseLabelProvider This interface should never be directly implemented.
abstract class Reference Because reference objects are implemented in close cooperation with the garbage collector, this class may not be subclassed directly.
class TextInvocationContext Clients may extend this class
abstract class PersistentState Clients may extend this class.
abstract class LinkedModeUI.LinkedModeUITarget Clients may extend this class.
class ProposalPosition Clients may instantiate or extend this class.
class TabsToSpacesConverter Clients can also extend it in their own subclass.
class ColumnViewerEditorActivationStrategy Implementors can extend
interface IMenuStateIds Clients may implement or extend this class.
class RadioState Clients may instantiate or extend this interface.
class LinkedModeUI Clients may instantiate and extend this class.
class TemplateProposal Clients may subclass.
abstract class AbstractConcurrentModel Clients should subclass this class instead of implementing IConcurrentModel directly.
abstract class AbstractHyperlinkDetector Clients may subclass.
abstract class Revision Clients may subclass.
abstract class AbstractKeyFormatter It is recommended that implementations of IKeyFormatter subclass from here, rather than implementing IKeyFormatter directly.
abstract class TrayDialog It is recommended to subclass this class instead of Dialog in all cases except where the dialog should never show a tray.
abstract class TrayDialog For example, dialogs which are very short, simple, and quick to dismiss (e.g. a message dialog with an OK button) should subclass Dialog.
class FindReplaceDocumentAdapterContentProposalProvider Clients can subclass to provide additional proposals in case they are supported by their own find/replace mechanism.
ThreadLocal.initialValue() if the programmer desires thread-local variables to have an initial value other than null, ThreadLocal must be subclassed, and this method overridden.
class ToolBarManager it may also be subclassed
class StatusLineManager it may also be subclassed if a more sophisticated layout is required.
class MenuManager it may also be subclassed.
interface ICoolBarManager @link CoolBarManager, which clients may instantiate or subclass.
class CoolBarManager it may also be subclassed.
class TitleAreaDialog This dialog class may be subclassed.
class ProgressMonitorDialog This concrete dialog class can be further subclassed as required.
class MessageDialog This concrete dialog class can be further subclassed as required.
class InputDialog This concrete dialog class can be further subclassed as required.
class StringFieldEditor This class may be
subclassed as required.

class ScaleFieldEditor This class may be subclassed as required

class SourceViewerConfiguration Clients should subclass

class SourceViewer Clients may subclass this class but should expect some breakage by future releases.

interface IInformationControlExtension5 Clients are encouraged to subclass @link AbstractInformationControl, which does this for free.

interface IInformationControl Clients can implement this interface and its extension interfaces, subclass @link AbstractInformationControl, or use the (text-based) default implementation @link DefaultInformationControl.

(abstract) IInformationControl.addFocusListener(listener) Clients are encouraged to subclass @link AbstractInformationControl, which does this for free.

interface Unreferenced A remote object implementation should implement the Unreferenced interface to receive notification when there are no more clients that reference that remote object.

interface RMIClientSocketFactory An implementation of this interface should be serializable and should implement @link Object#equals to return true when passed an instance that represents the same (functionally equivalent) client socket factory, and false otherwise (and it should also implement @link Object#hashCode consistently with its Object.equals implementation).

(abstract) Wizard.performFinish() Subclasses must implement this IWizard method to perform any special finish processing for their wizard.

abstract class StyledCellLabelProvider For providing the label's styles, create a subclass and overwrite @link StyledCellLabelProvider#update(ViewerCell) to return set all information needed to render a element.

class PropertyChangeEvent This concrete class was designed to be instantiated, but may also be subclassed if required.

abstract class OwnerDrawLabelProvider This class is intended to be subclassed by implementors.

class LabelProvider This class may be used as is, or subclassed to provide richer labels.

class FocusCellOwnerDrawHighlighter This class can be subclassed to configure how the coloring of the selected cell.

class ColumnLabelProvider This classes is intended to be subclassed

class WizardDialog Clients may subclass WizardDialog, although this is rarely required.

abstract class CellLabelProvider This class is intended to be subclassed

interface IHyperlinkDetectorExtension2 Clients may implement this interface.

interface ITextViewer Clients may implement this interface and its extension interfaces

class TreeNodeContentProvider This class and all of its methods may be overridden or extended.

interface IHyperlinkDetectorExtension Clients may implement this interface.

class ContentProposalAdapter This class provides some overridable methods to allow clients to manually control the popup.

interface ISchemeListener This interface may be implemented by clients.

interface IBindingManagerListener This interface may be implemented by clients.

interface IControlCreator Clients are expected to implement this interface in order to create a particular kind of control for decoration.

interface IControlContentAdapter Clients are expected to implement this interface when defining a @link ContentProposalAdapter, in order to specify how to retrieve and set the contents of the control being adapted.

class Thread There are two ways to create a new thread of execution. One is to declare a class to be a subclass of Thread. This subclass should override the run method of class Thread.

class Thread The other way to create a thread is to declare a class that implements the Runnable interface. That class then implements the run method. An instance of the class can then be allocated, passed as an argument when creating Thread, and started.

interface IContextInformationValidator The interface can be implemented by clients.

interface IContextInformationPresenter The interface can be implemented by clients.

interface IContextInformation The interface can be implemented by clients.

interface IContentAssistant The interface can be implemented by clients.

interface IContentAssistProcessor This interface must be implemented by clients.

interface ICompletionProposal This interface can be implemented by clients.

interface ICompletionListenerExtension Clients may implement this interface.

interface ICompletionListener Clients may implement this interface.

interface IFormattingStrategyExtension This interface must be implemented by clients. Implementers should be registered with a content formatter in order get involved in the formatting process.

interface IFormattingStrategy This interface must be implemented by clients. Implementers should be registered with a content formatter in order get involved in the formatting process.

interface IContentFormatter The interface can be implemented by clients.

interface IHyperlinkPresenterExtension Clients may implement this interface.

interface IHyperlinkDetector Clients may implement this interface.

interface IHyperlink Clients may implement this interface.

interface IInformationProvider Clients may implement this interface.

interface LinkedModeUI.ILinkedModeUIFocusListener Clients may implement this interface.

interface LinkedModeUI.IExitPolicy Clients may implement this interface.

interface IQuickFixableAnnotation This interface can be implemented by clients.

interface IQuickAssistProcessor This interface can be implemented by clients.

interface IQuickAssistInvocationContext This interface can be implemented by clients.

interface IQuickAssistAssistant The interface can be implemented by clients. By default, clients use @link QuickAssistAssistant as the standard implementer of this interface.

interface IReconcilingStrategy This interface must be implemented by clients. Implementers should be registered with a reconciler in order get involved in the reconciling process.

interface IReconciler The interface can be implemented by clients. By default, clients use @link org.eclipse.jface.text.reconciler.MonoReconciler or @link org.eclipse.jface.text.reconcile as the standard implementers of this interface.

interface IReconcileStep This interface must be implemented by clients.

interface IReconcileResult This interface must be implemented by clients that want to let one of their model elements be part of a reconcile step result.

interface IReconcilableModel This interface must be implemented by clients that want to use one of their models as a reconcile step's input model.

interface IInformationPresenter The interface can be implemented by clients. By default, clients use @link org.eclipse.jface.text.information.InformationPresenter as the standard implementer of this interface.

interface IVerticalRulerListener This interface may be implemented by clients.

interface ISharedTextColors This interface may be implemented by clients.

interface ILineDiffer This interface may be implemented by clients.

interface ILineDiffInfo This interface may be implemented by clients.

(abstract) IAnnotationPresentation.paint(gc, canvas, bounds) Implement this method to draw a graphical representation of this annotation within the given bounds.

interface IAnnotationHover Clients may implement this interface.

interface IAnnotationAccess Clients usually implement this interface and its extension interfaces.

interface IViewportListener Clients may implement this interface.

interface ITextListener Clients may implement this interface.

interface ITextInputListener Clients may implement this interface.

interface ITextHover Clients may implement this interface.

interface ISelectionValidator Clients may implement and use this interface.

interface IPainter Clients may implement this interface.

interface IInputChangedListener Clients can implement that interface and its extension interfaces.

interface IEventConsumer Clients may implement this interface.

interface IEditingSupport Clients may implement this interface.

interface IDelayedInputChangeProvider Clients can implement that interface and its extension interfaces.

interface IAutoEditStrategy Clients may implement this interface.

interface DefaultInformationControl.IInformationPresenter The interface can be implemented by clients.

interface DefaultInformationControl.IInformationPresenterExtension The interface can be implemented by clients.

interface AbstractInformationControlManager.IInformationControlCloser Clients must implement this interface in order to equip an information control manager accordingly.

class TableTreeViewer Content providers for table tree viewers must implement the ITreeContentProvider interface.

class TableTreeViewer Label providers for table tree viewers must implement either the ITableLabelProvider or the ILabelProvider interface (see TableTreeViewer.setLabelProvider for more details).

interface ICellModifier This interface should be implemented by classes that wish to act as cell modifiers.
interface ICellEditorValidator This interface should be implemented by classes that wish to act as cell editor validators.
interface ICellEditorListener This interface should be implemented by classes that wish to react to cell editor activity.
abstract class AbstractLinkedList An abstract implementation of a linked list which provides numerous points for subclasses to override.
Overridable methods are provided to change the storage node and to change how nodes are added to and removed.
abstract class AbstractReferenceMap Key-value entries are stored in instances of the ReferenceEntry class which can be overridden and replaced.
class AbstractHashedMap An abstract implementation of a hash-based map which provides numerous points for subclasses to override.
class AbstractHashedMap Key-value entries are stored in instances of the HashEntry class, which can be overridden and replaced. The iterators can similarly be replaced, without the need to replace the KeySet, EntrySet and Values view classes.
abstract class AbstractLinkedMap Key-value entries are stored in instances of the LinkEntry class which can be overridden and replaced.
abstract class ClassLoader Applications implement subclasses of ClassLoader in order to extend the manner in which the Java virtual machine dynamically loads classes.
class CompositeRuler Such columns must implement the @link org.eclipse.jface.text.source.IVerticalRulerColumn. interface.
class DateFormatSymbols New DateFormatSymbols subclasses may be added to support SimpleDateFormat for date-time formatting for additional locales.
abstract class BasicPermission Subclasses may implement actions on top of BasicPermission, if desired.
class ProviderException A runtime exception for Provider exceptions (such as misconfiguration errors or unrecoverable internal errors), which may be subclassed by Providers to throw specialized, provider-specific runtime errors.

2.2 Method Overriding Directive

CellEditor.activate() Subclasses may reimplement.
ActivationGroup.inactiveObject(id) The subclass of ActivationGroup must override this method and unexport the object.
PopupDialog.fillDialogMenu(dialogMenu) Subclasses may extend or override.
AbstractReconciler.forceReconciling() Clients may extend this method.
FocusCellHighlighter.init() Subclasses may extend.
CellNavigationStrategy.init() Subclasses may extend.
WizardSelectionPage.dispose() Subclasses may extend.
CoolBarManager.update(force) Subclasses may extend
DialogPage.dispose() Subclasses may extend.
DialogPage.setTitle(title) Subclasses may extend.
DialogPage.setVisible(visible) Subclasses may extend.
CoolBarManager.itemAdded(item) Subclasses may extend
CoolBarManager.itemRemoved(item) Subclasses may extend
PreferencePage.setTitle(title) Subclasses may extend
AbstractInformationControlManager.handleSubjectControlDisposed() Subclasses may extend this method.
ViewerColumn.setLabelProvider(labelProvider) Subclasses may extend
FocusCellHighlighter.focusCellChanged(cell) Subclasses may extend
BaseLabelProvider.dispose() Subclasses may extend
WizardPage.isPageComplete() Subclasses may extend.
WizardPage.setDescription(description) Subclasses may extend.
WizardPage.setTitle(title) Subclasses may extend.
SubContributionManager.disposeManager() Subclasses may extend.
PopupDialog.getForegroundColorExclusions() Subclasses may extend this method
PopupDialog.getBackgroundColorExclusions() Subclasses may extend this method
AbstractRulerColumn.dispose() Subclasses may extend this method.
ViewerColumn.handleDispose() Subclasses may extend
PreferenceDialog.handleShellCloseEvent() Subclasses may extend or reimplement.
ViewerColumn.setEditingSupport(editingSupport) Subclasses may extend
TextCellEditor.editOccured(e) Subclasses may extend or reimplement.
FocusCellHighlighter.focusCellChanged(newCell, oldCell) Subclasses may extend.
DialogCellEditor.createButton(parent) Subclasses may extend or reimplement.
ContentViewer.setInput(input) Subclassers may extend this method if required.
CellLabelProvider.initialize(viewer, column) Subclasses may extend
CellLabelProvider.dispose(viewer, column) Subclasses may extend
CellEditor.getLayoutData() Subclasses may extend or reimplement.
Window.canHandleShellCloseEvent() Subclasses may extend or reimplement.
Window.configureShell(newShell) Subclasses may extend or reimplement.
Window.handleShellCloseEvent() Subclasses may extend or reimplement.
Window.initializeBounds() Subclasses may extend or reimplement.
PopupDialog.saveDialogBounds(shell) Subclasses may extend it
Window.getShellListener() Subclasses may extend or reimplement.
Dialog.createDialogArea(parent) Subclasses must override this method
PreferencePage.performDefaults() Subclasses may override
Dialog.buttonPressed(buttonId) Subclasses may override to handle other buttons
OwnerDrawLabelProvider.initialize(viewer, column) Subclasses may override this method
ErrorDialog.createDialogArea(parent) Subclasses that require a different dialog area may either override this method
OwnerDrawLabelProvider.erase(event, element) Subclasses should override this method
FieldEditorPreferencePage.dispose() Subclasses may override
PreferencePage.doComputeSize() All pages should override this method
PopupDialog.createDialogArea(parent) Subclasses must override this method
AbstractInformationControl.getInformationPresenterControlCreator() Subclasses may override.
SubMenuManager.dispose() Subclasses may override.
SubContributionItem.dispose() Subclasses may override.
ContributionItem.dispose() Subclasses may override.
ContributionItem.fill(parent) Subclasses may override.
ContributionItem.fill(menu, index) Subclasses may override.
MouseDragGestureRecognizer.unregisterListeners() subclasses must override this method
(abstract) DragGestureRecognizer.registerListeners() subclasses must override this method
(abstract) DragGestureRecognizer.unregisterListeners() subclasses must override this method
URLStreamHandler.parseURL(u, spec, start, limit) A stream protocol handler for a protocol that has a different syntax must override this routine.
abstract class CertStoreSpi All CertStore implementations must include a class (the SPI class) that extends this class (CertStoreSpi)
AbstractReconciler.startReconciling() Clients may extend this method.
TextViewer.disableRedrawing() Subclasses may extend.
AbstractInformationControl.setFocus() Subclasses can override or extend.
AbstractInformationControl.computeSizeConstraints(widthInChars, heightInChars) Subclasses can override or extend.
ContentViewer.labelProviderChanged() Subclasses may reimplement or extend.
abstract class AbstractInformationControl They should also extend @link #computeTrim() if they create a content area with additional trim (e.g. scrollbars)
abstract class AbstractInformationControl They should override @link #getInformationPresenterControlCreator().
AbstractRulerColumn.paintLine(gc, modelLine, widgetLine, linePixel, lineHeight) Subclasses may replace or extend.
abstract class PreferencePage Subclasses must implement the createContents framework method to supply the page's main control.
abstract class PreferencePage Subclasses should extend the doComputeSize framework method to compute the size of the page's control.
AbstractRulerColumn.paint(gc, lines) Subclasses may replace or extend.
abstract class Window close- extend to free other SWT resources
abstract class Window configureShell- extend or reimplement to set shell properties before window opens

abstract class Window createContents- extend or reimplement to create controls before window opens

abstract class Window getInitialSize- reimplement to give the initial size for the shell

abstract class Window getInitialLocation- reimplement to give the initial location for the shell

abstract class Window getShellListener- extend or reimplement to receive shell events

abstract class Window handleFontChange- reimplement to respond to font changes

abstract class Window handleShellCloseEvent- extend or reimplement to handle shell closings

CellEditor.focusLost() Subclasses may also extend or reimplement.

CellEditor.keyReleaseOccured(keyEvent) Subclasses may also extend or reimplement.

AbstractRulerColumn.computeForeground(line) Subclasses may replace or extend.

AbstractRulerColumn.computeBackground(line) Subclasses may replace or extend.

AbstractRulerColumn.computeText(line) Subclasses may replace or extend.

abstract class Wizard reimplement createPageControls

abstract class Wizard reimplement performCancel

abstract class Wizard extend addPages

abstract class Wizard reimplement performFinish

abstract class Wizard extend dispose

Wizard.dispose() Subclasses should extend this method if the wizard instance maintains addition SWT resource that need to be disposed.

Wizard.addPages() Subclasses should extend if extra pages need to be added before the wizard opens.

WizardPage.canFlipToNextPage() Subclasses may override (extend or reimplement)

ContentViewer.handleLabelProviderChanged(event) Subclasses may reimplement or extend.

StructuredViewer.handleLabelProviderChanged(event) Subclasses may reimplement or extend.

AbstractInformationControl.computeTrim() Subclasses can extend this method to add additional trim

abstract class Format Concrete subclasses must implement three methods:

1. format(Object obj, StringBuffer toAppendTo, FieldPosition pos) 2. formatToCharacterIterator(Object obj) 3. parseObject(String source, ParsePosition pos)

ContentViewer.hookControl(control) Subclasses may override if they need to add other control hooks

abstract class Authenticator Applications use this class by overriding @link #getPasswordAuthentication() in a sub-class.

(abstract) Collection.hashCode() programmers should take note that any class that overrides the Object.equals method must also override the Object.hashCode method in order to satisfy the general contract for the Object.hashCodemethod

MenuManager.isEnabled() Subclasses may reimplement.

FieldEditor.setFocus() Subclasses may reimplement.

AbstractReconciler.initialProcess() Clients may reimplement this method.

SourceViewer.updateSlaveDocuments(masterDocument) Subclasses may reimplement.

AbstractRulerColumn.getCanvasStyle() Clients may reimplement this method to create a canvas with their desired style bits.

TextViewer.validateSelectionRange(selectionRange) Subclasses may reimplement this method.

class ViewerSorter Subclasses may implement the isSorterProperty method; they may reimplement the category method to provide categorization; and they may override the compare methods to provide a totally different way of sorting elements.

ViewerFilter.isFilterProperty(element, property) Subclasses should reimplement.

class ViewerComparator Subclasses may implement the isSorterProperty method; they may reimplement the category method to provide categorization; and they may override the compare methods to provide a totally different way of sorting elements.

ViewerComparator.category(element) Subclasses may reimplement this method to provide non-trivial categorization.

ViewerComparator.isSorterProperty(element, property) Subclasses may reimplement.

ViewerComparator.sort(viewer, elements) Subclasses may reimplement this method to provide a more optimized implementation.

TreePathViewerSorter.isSorterProperty(parentPath, element, property) Subclasses may reimplement.

TreePathViewerSorter.sort(viewer, parentPath, elements) Subclasses may reimplement this method to provide a more optimized implementation.

DialogCellEditor.createContents(cell) Subclasses may reimplement.

DialogCellEditor.updateContents(value) Subclasses may reimplement.

Window.handleFontChange(event) Subclasses may reimplement.

abstract class WizardPage performHelp - may be reimplemented to display help for the page

abstract class WizardPage canFlipToNextPage - may be extended or reimplemented

abstract class WizardPage isPageComplete - may be extended

abstract class WizardPage setDescription - may be extended

abstract class WizardPage setTitle - may be extended

abstract class WizardPage dispose - may be extended to dispose additional allocated SWT resources

Wizard.createPageControls(pageContainer) Subclasses should reimplement this method if they want to delay creating one or more of the pages lazily.

Wizard.performCancel() Subclasses should reimplement this method if they need to perform any special cancel processing for their wizard.

abstract class AbstractMap To implement a modifiable map, the programmer must additionally override this class's put method (which otherwise throws an UnsupportedOperationException)

abstract class AbstractMap the iterator returned by entrySet().iterator() must additionally implement its remove method.

Applet.getParameterInfo() An applet should override this method to return an array of Strings describing these parameters.

ControlContribution.fill(parent) Subclasses must implement createControl rather than overriding this method.

ControlContribution.fill(parent, index) Subclasses must implement createControl rather than overriding this method.

Action.runWithEvent(event) Subclasses should override this method if they need information from the triggering event, or override run() if not.

PopupDialog.createInfoTextArea(parent) Subclasses typically need not override this method, but may do so.

PopupDialog.close() This method may be extended (super.close must be called).

Dialog.createButtonBar(parent) Subclasses may override.

ThreadLocal.initialValue() if the programmer desires thread-local variables to have an initial value other than null, ThreadLocal must be subclassed, and this method overridden.

SubContributionItem.dispose() Subclasses may override.

abstract class ControlContribution The createControl framework method must be implemented by concrete subclasses.

ControlContribution.computeWidth(control) Subclasses may override if required.

(abstract) ControlContribution.createControl(parent) This framework method must be implemented by concrete subclasses.

abstract class ContributionManager the changes are propagated to the control via the update method, which subclasses must implement.

ContributionItem.fill(parent, index) Subclasses may override.

ContributionItem.fill(parent, index) Subclasses may override.

ContributionItem.saveWidgetState() Subclasses may override.

ContributionItem.isDirty() Subclasses may override.

ContributionItem.isEnabled() Subclasses may override.

ContributionItem.isDynamic() Subclasses may override.

ContributionItem.isGroupMarker() Subclasses may override.

ContributionItem.isSeparator() Subclasses may override.

ContributionItem.update() Subclasses may override.

ContributionItem.update(id) Subclasses should override to update their state.

abstract class Action Subclasses must implement the IAction.run method to carry out the action's semantics.

Action.run() . Subclasses should override this method if they do not need information from the triggering event, or override runWithEvent(Event) if they do.

AbstractGroupMarker.isGroupMarker() Subclasses may override.

Binding.toString() This should be overridden by subclasses that add properties.

TrayDialog.createHelpControl(parent) Subclasses may override.

TitleAreaDialog.createDialogArea(parent) Subclasses should override.

StatusDialog.updateButtonsEnableState(status) Subclasses may override this method to update additional buttons.

PopupDialog.getFocusControl() Subclasses may override this method.

PopupDialog.hasTitleArea() Subclasses may override.

PopupDialog.hasInfoArea() Subclasses may override.

PopupDialog.createTitleControl(parent) Subclasses may override if a different control is desired for representing the title text, or if something different than the title should be displayed in location where the title text typically is shown.

PopupDialog.setInfoText(text) Subclasses may override this method.
PopupDialog.setTitleText(text) Subclasses may override this method.
PopupDialog.getPersistBounds() Subclasses may override this method.
PopupDialog.getPersistLocation() Subclasses may override this method.
PopupDialog.getPersistSize() Subclasses may override this method.
PopupDialog.open() It provides framework hooks that allow subclasses to set the focus and tab order
PopupDialog.getDialogSettings() Subclasses should override this method when they wish to persist the bounds of the dialog.
PopupDialog.getDefaultSize() Subclasses should override this method when an alternate default size is desired, rather than overriding @link #getInitialSize().
PopupDialog.getDefaultLocation(initialSize) Subclasses should override this method when an alternate default location is desired, rather than overriding @link #getInitialLocation(Point)
PopupDialog.adjustBounds() Subclasses should override this method when there are bounds computations that must be checked each time the dialog is opened.
PopupDialog.getForeground() Subclasses may override.
PopupDialog.getBackground() Subclasses may override.
PopupDialog.applyForegroundColor(color, control) Subclasses may override this method, but typically do not.
PopupDialog.applyBackgroundColor(color, control) Subclasses may override this method, but typically do not.
MessageDialog.createCustomArea(parent) Subclasses may override.
MessageDialog.getMinimumMessageWidth() Subclasses may override.
InputDialog.getInputTextStyle() Subclasses may override.
(abstract) IconAndMessageDialog.getImage() Subclasses may override.
DialogPage.getToolTipText(widgetId) Subclasses may override.
DialogPage.performHelp() Subclasses should override to take some action in response to a help request.
Dialog.cancelPressed() Subclasses may override if desired.
Dialog.createButton(parent, id, label, defaultButton) Subclasses may override.
Dialog.createButtonsForButtonBar(parent) Subclasses may override.
Dialog.okPressed() Subclasses may override.
ControlAnimator.setVisible(visible) Subclasses should override this method.
AnimatorFactory.createAnimator(control) Subclasses should override this method.
StringFieldEditor.doCheckState() Subclasses should override this method to specific state checks.
StringFieldEditor.doFillIntoGrid(parent, numColumns) Subclasses may override but must call super.doFillIntoGrid.
(abstract) StringButtonFieldEditor.changePressed() Subclasses must implement this method to provide a corresponding new string for the text field.
abstract class PreferencePage Subclasses may override the performOk, performApply, performDefaults, performCancel, and performHelp framework methods to react to the standard button events.
PreferencePage.contributeButtons(parent) Subclasses should override this method to contribute buttons to this page's button bar.
(abstract) PreferencePage.createContents(parent) This framework method must be implemented by concrete subclasses.
PreferencePage.applyDialogFont(composite) Subclasses may override if they wish to set the font themselves.
PreferencePage.doGetPreferenceStore() This is a framework hook method for subclasses to return a page-specific preference store.
PreferencePage.performCancel() This is a framework hook method for subclasses to do special things when the Cancel button has been pressed.
PreferenceDialog.handleSave() Subclasses may override.
abstract class ListEditor Subclasses must implement the parseString, createList, and getNewInputObject framework methods.
(abstract) ListEditor.createList(items) Subclasses must implement this method.
(abstract) ListEditor.getNewInputObject() Subclasses must implement this method.
(abstract) ListEditor.parseString(stringList) Subclasses must implement this method.
abstract class FieldEditorPreferencePage Subclasses must implement the createFieldEditors method and should override createLayout if a special layout of the field editors is needed.
(abstract) FieldEditorPreferencePage.createFieldEditors() Subclass must implement this method to create the field editors.
(abstract) FieldEditor.adjustForNumColumns(numColumns) Subclasses must implement this method to adjust the horizontal span of controls so they appear correct in the given number of columns.
FieldEditor.applyFont() Subclasses should override this method if they want to change the font of the SWT control to a value different than the standard dialog font.
(abstract) FieldEditor.doFillIntoGrid(parent, numColumns) Subclasses must implement this method to create the controls for this field editor.
(abstract) FieldEditor.doLoad() Subclasses must implement this method to properly initialize the field editor.
(abstract) FieldEditor.doLoadDefault() Subclasses must implement this method to properly initialize the field editor.
(abstract) FieldEditor.doStore() Subclasses must implement this method to save the entered value into the preference store.
FieldEditor.isValid() Subclasses wishing to perform validation should override both this method and refreshValidState.
FieldEditor.refreshValidState() Subclasses wishing to perform validation should override both this method and isValid.
abstract class CompositeImageDescriptor Subclasses must implement the getSize and fill methods.
(abstract) CompositeImageDescriptor.drawCompositeImage(width, height) Subclasses must implement this framework method
(abstract) CompositeImageDescriptor.getSize() Subclasses must implement this framework method.
(abstract) AbstractReconciler.reconcilerDocumentChanged(newDocument) Usually, subclasses use this hook to inform all their reconciling strategies about the change.
AbstractReconciler.aboutToBeReconciled() Hook for subclasses which want to perform some action as soon as reconciliation is needed.
Revision.getAuthor() Subclasses should replace - the default implementation returns the empty string.
FastPartitioner.connect(document, delayInitialization) May be extended by subclasses.
FastPartitioner.initialize() May be extended by subclasses.
FastPartitioner.disconnect() May be extended by subclasses.
FastPartitioner.documentAboutToBeChanged(e) May be extended by subclasses.
FastPartitioner.documentChanged2(e) May be extended by subclasses.
FastPartitioner.findClosestPosition(offset) May be extended or replaced by subclasses.
FastPartitioner.getContentType(offset) May be replaced or extended by subclasses.
FastPartitioner.getPartition(offset) May be replaced or extended by subclasses.
FastPartitioner.getLegalContentTypes() May be replaced or extended by subclasses.
FastPartitioner.isSupportedContentType(contentType) May be extended by subclasses.
FastPartitioner.getTokenContentType(token) May be replaced or extended by subclasses.
FastPartitioner.getContentType(offset, preferOpenPartitions) May be replaced or extended by subclasses.
FastPartitioner.getPartition(offset, preferOpenPartitions) May be replaced or extended by subclasses.
FastPartitioner.computePartitioning(offset, length, includeZeroLengthPartitions) May be replaced or extended by subclasses.
FastPartitioner.stopRewriteSession(session) May be extended by subclasses.
FastPartitioner.getActiveRewriteSession() May be extended by subclasses.
SourceViewer.createLayout() Subclasses may override this method.
LineNumberRulerColumn.computeNumberOfDigits() Subclasses may override this method if they need extra space on the line number ruler.
LineChangeHover.formatSource(content) May be overridden by subclasses.
LineChangeHover.getTabReplacement() subclasses may override to specify a number of spaces.
DefaultAnnotationHover.formatMultipleMessages(messages) Hook method to formats the given messages.
Subclasses can change this to create a different format like HTML.
DefaultAnnotationHover.formatSingleMessage(message) Hook method to format the given single message.
Subclasses can change this to create a different format like HTML.
abstract class AbstractRulerColumn Subclasses can hook into the paint loop at three levels:
* Override @link #paint(GC, ILineRange) to control the entire painting of the ruler. * Override @link #paintLine(GC, int, int, int, int) to control the painting of a line. * Leave the painting to the default implementation, but override @link #computeBackground(int), @link #computeForeground(int) and @link #computeText(int) to specify the ruler appearance for a line.
abstract class TemplateCompletionProcessor Subclasses need to provide implementations for @link #getTemplates(String), @link #getContextType(ITextViewer, IRegion) and @link #getImage(Template).
TemplateCompletionProcessor.createProposal(template, context, region, relevance) Subclasses may replace this method to provide their own implementations.
TextViewer.handleVisibleDocumentAboutToBeChanged(event) Subclasses may override.
TextViewer.handleVisibleDocumentChanged(event) Subclasses may override.

abstract class AbstractInformationControl Subclasses must either override @link IInformationControl#setInformation(String) or implement @link IInformationControlExtension2.

Authenticator.getPasswordAuthentication() Subclasses should override the default implementation, which returns null.

URLConnection.getPermission() Subclasses should override this method and return the permission that best represents the permission required to make a a connection to the URL.

(abstract) URLStreamHandler.openConnection(u) This method should be overridden by a subclass.

interface SocketOptions This interface is implemented by: SocketImpl and DatagramSocketImpl. Subclasses of these should override the methods of this interface in order to support their own options.

interface RMIServerSocketFactory An implementation of this interface should implement @link Object#equals to return true when passed an instance that represents the same (functionally equivalent) server socket factory, and false otherwise (and it should also implement @link Object#hashCode consistently with its Object.equals implementation).

interface RMIClientSocketFactory An implementation of this interface should be serializable and should implement @link Object#equals to return true when passed an instance that represents the same (functionally equivalent) client socket factory, and false otherwise (and it should also implement @link Object#hashCode consistently with its Object.equals implementation).

Reader.markSupported() Subclasses should override this method.

Writer.write(c) Subclasses that intend to support efficient single-character output should override this method.

abstract class FilterWriter Subclasses of FilterWriter should override some of these methods

abstract class FilterReader Subclasses of FilterReader should override some of these methods

Reader.read() Subclasses that intend to support efficient single-character input should override this method.

FilterOutputStream.write(b, off, len) Subclasses of FilterOutputStream should provide a more efficient implementation of this method.

ObjectOutputStream.writeClassDescriptor(desc) The corresponding method in ObjectInputStream, readClassDescriptor, should then be overridden to reconstitute the class descriptor from its custom stream representation.

InputStream.available() This method should be overridden by subclasses.

Thread.run() Subclasses of Thread should override this method. @see #start() @see #stop() @see #Thread(ThreadGroup, Runnable, String)

SecurityManager.checkAccess(t) Applications that want a stricter policy should override this method.

SecurityManager.getThreadGroup() . This should be overridden by a specific security manager to return the appropriate thread group.

abstract class Wizard Subclasses may override these methods if required:

ApplicationWindow.getCoolBarControl() Subclasses may override this method to customize the cool bar manager.

ApplicationWindow.getToolBarControl() Subclasses may override this method to customize the tool bar manager.

ApplicationWindow.getFont() Subclasses may override to use a different registry, etc.

ApplicationWindow.createToolBarControl(parent) Subclasses may override this method to customize the tool bar manager.

ApplicationWindow.createCoolBarControl(composite) Subclasses may override this method to customize the cool bar manager.

ApplicationWindow.createCoolBarManager2(style) Subclasses may override this method to provide an alternative implementation for the cool bar manager.

ApplicationWindow.createCoolBarManager(style) Subclasses may override this method to customize the cool bar manager.

ApplicationWindow.createToolBarManager2(style) Subclasses may override this method to provide an alternative implementation for the tool bar manager.

ApplicationWindow.createToolBarManager(style) Subclasses may override this method to customize the tool bar manager.

ApplicationWindow.createMenuManager() Subclasses may override this method to customize the menu manager.

ApplicationWindow.createStatusLineManager() Subclasses may override this method to customize the status line manager.

DefaultToolTip.createToolTipContentArea(event, parent) To customize the text Subclasses may override the following methods

* @link #getStyle(Event) * @link #getBackgroundColor(Event) * @link #getForegroundColor(Event) * @link #getFont(Event) * @link #getImage(Event) * @link #getText(Event) * @link #getBackgroundImage(Event)

ToolTip.afterHideToolTip(event) Subclasses may override to clean up requested system resources

Window.createShell() Subclasses should override configureShell if the shell needs to be customized.

Window.createContents(parent) Subclasses may override.

Window.getLayout() Subclasses that change the layout type by overriding this method should also override createContents.

ToolTip.getLocation(tipSize, event) Subclasses may overwrite to implement custom positioning.

(abstract) CellEditor.doSetValue(value) This framework method must be implemented by concrete subclasses.

(abstract) CellEditor.doSetFocus() This framework method must be implemented by concrete subclasses.

(abstract) CellEditor.doGetValue() This framework method must be implemented by concrete subclasses.

(abstract) CellEditor.createControl(parent) This framework method must be implemented by concrete subclasses.

ViewerRow.getVisualIndex(creationIndex) Implementators of @link ColumnViewer should overwrite this method if their widget supports reordered columns

ViewerRow.getCreationIndex(visualIndex) Implementators of @link ColumnViewer should overwrite this method if their widget supports reordered columns

ViewerRow.scrollCellIntoView(columnIndex) Implementators of @link ColumnViewer should overwrite this method if their widget supports reordered columns

abstract class StyledCellLabelProvider For providing the label's styles, create a subclass and overwrite @link StyledCellLabelProvider#update(ViewerCell) to return set all information needed to render a element.

(abstract) EditingSupport.setValue(element, value) Subclasses should overwrite.

ColumnViewerEditor.processTraverseEvent(columnIndex, row, event) Subclasses may overwrite to implement their custom logic to edit the next cell

AbstractTreeViewer.doGetColumnCount() Subclasses should overwrite this method, which has a default implementation (returning 0) for API backwards compatility reasons

ToolTip.shouldCreateToolTip(event) Subclasses may overwrite this to get custom behavior

abstract class ClassLoader The network class loader subclass must define the methods @link #findClass findClass and loadClassData to load a class from the network.

PKIXCertPathChecker.clone() All subclasses which maintain state must support and override this method, if necessary.

(abstract) Permission.implies(permission) This must be implemented by subclasses of Permission, as they are the only ones that can impose semantics on a Permission object.

abstract class PKIXCertPathChecker A stateful implementation must override the @link #clone clone method if necessary in order to allow a PKIX CertPathBuilder to efficiently backtrack and try other paths.

abstract class Permission An important method that must be implemented by each subclass is the implies method to compare Permissions.

abstract class PermissionCollection

When it is desirable to group together a number of Permission objects of the same type, the newPermissionCollection method on that particular type of Permission object should first be called. The default behavior (from the Permission class) is to simply return null. Subclasses of class Permission override the method if they need to store their permissions in a particular PermissionCollection object in order to provide the correct semantics when the PermissionCollection.implies method is called. If a non-null value is returned, that PermissionCollection must be used. If null is returned, then the caller of newPermissionCollection is free to store permissions of the given type in any PermissionCollection they choose (one that uses a Hashtable, one that uses a Vector, etc).

NumberFormat.getRoundingMode() Subclasses which handle different rounding modes should override this method.

NumberFormat.setRoundingMode(roundingMode) Subclasses which handle different rounding modes should override this method.

Format.formatToCharacterIterator(obj) Subclasses that support fields should override this and create an AttributedCharacterIterator with meaningful attributes.

PolicySpi.engineRefresh() This method should be overridden if a refresh operation is supported by the policy implementation.

Policy.refresh() This method should be overridden if a refresh operation is supported by the policy implementation.

Identity.identityEquals(identity) This method should be overriden by subclasses to test for equality.

X509CRLEntry.getCertificateIssuer() Subclasses that wish to support indirect CRLs should override it.

abstract class Signer The management of a signer's private keys is an important and sensitive issue that should be handled by subclasses as appropriate to their intended use.

MessageDigestSpi.engineDigest(buf, offset, len) Knowledgeable providers should override this method.

X509CRL.getRevokedCertificate(certificate) Subclasses that wish to support indirect CRLs should override this method.

SignatureSpi.engineVerify(sigBytes, offset, length) Subclasses should overwrite the default implementation

Permission.newPermissionCollection() Subclasses of class Permission should override this if they need to store their permissions in a particular PermissionCollection object in order to provide the correct semantics when the PermissionCollection.implies method is called.

Identity.equals(identity) Otherwise, it calls @link #identityEquals(Identity) identityEquals, which subclasses should override.

SignatureSpi.engineSign(outbuf, offset, len) Knowledgeable providers should override this method.

X509Certificate.getExtendedKeyUsage() Subclasses should override this method with a correct implementation.

X509Certificate.getIssuerAlternativeNames() Subclasses should override this method with a correct implementation.

X509Certificate.getSubjectAlternativeNames() Subclasses should override this method with a correct implementation.

ViewerFilter.filter(viewer, parentPath, elements) Subclasses may override

abstract class ViewerFilter Subclasses must implement the select method and may implement the isFilterProperty method.

abstract class ViewerDropAdapter Subclasses must implement the following methods:

* validateDrop - identifies valid drop targets in viewer * performDrop - carries out a drop into a viewer

(abstract) ViewerDropAdapter.performDrop(data) Subclasses must implement this method to provide drop behavior.

(abstract) ViewerDropAdapter.validateDrop(target, operation, transferType) Subclasses must implement this method to define which drops make sense.

TextCellEditor.isSelectAllEnabled() Subclasses may override

TreePathViewerSorter.category(parentPath, element) . Subclasses may override.

LabelProvider.getText(element) Subclasses may override.

class LabelProvider Subclasses may override any of the following methods:

* isLabelProperty * getImage * getText * dispose

(abstract) DialogCellEditor.openDialogBox(cellEditorWindow) This framework method must be implemented by concrete subclasses.

abstract class DialogCellEditor Subclasses may override the following methods:

* createButton: creates the cell editor's button control * createContents: creates the cell editor's 'display value' control * updateContents: updates the cell editor's 'display value' control after its value has changed * openDialogBox: opens the dialog box when the end user presses the button

StructuredViewer.handleInvalidSelection(invalidSelection, newSelection) Subclasses may override it to implement a different strategy for picking a new selection when the old selection becomes invalid.

(abstract) StructuredViewer.setSelectionToWidget(l, reveal) Subclasses should override to set their selection based on the given list of elements.

ColumnViewerToolTipSupport.createViewerToolTipContentArea(event, cell, parent) Subclasses can overload this method to implement their own tool tip design.

ColumnViewer.hookEditingSupport(control) Subclasses may override.

(abstract) ColumnViewer.createViewerEditor() To be implemented by subclasses.

AbstractTreeViewer.createViewerEditor() Subclasses should override.

CellEditor.isCopyEnabled() Subclasses may override

CellEditor.isCutEnabled() Subclasses may override

CellEditor.isDeleteEnabled() Subclasses may override

CellEditor.isFindEnabled() Subclasses may override

CellEditor.isPasteEnabled() Subclasses may override

CellEditor.isRedoEnabled() Subclasses may override

CellEditor.isSelectAllEnabled() Subclasses may override

CellEditor.isUndoEnabled() Subclasses may override

CellEditor.performCopy() Subclasses may override

CellEditor.performCut() Subclasses may override

CellEditor.performDelete() Subclasses may override

CellEditor.performFind() Subclasses may override

CellEditor.performPaste() Subclasses may override

CellEditor.performRedo() Subclasses may override

CellEditor.performSelectAll() Subclasses may override

CellEditor.performUndo() Subclasses may override

CellEditor.dependsOnExternalFocusListener() Subclasses that hook their own focus listener should override this method and return false.

BaseLabelProvider.isLabelProperty(element, property) Subclasses may override.

AbstractTreeViewer.getItemAt(point) Subclasses should override.

ContentViewer.setLabelProvider(labelProvider) Overriding this method is generally not required; however, if overriding in a subclass, super.setLabelProvider must be invoked.

ContentViewer.setContentProvider(contentProvider) Overriding this method is generally not required; however, if overriding in a subclass, super.setContentProvider must be invoked.

ContentViewer.handleDispose(event) Subclasses should override this method to perform any additional cleanup of resources;

ContentViewer.getLabelProvider() Overriding this method is generally not required; however, if overriding in a subclass, super.getLabelProvider must be invoked.

LabelProvider.getImage(element) Subclasses may override.

StructuredViewer.updateSelection(selection) Overriding this method is generally not required; however, if overriding in a subclass, super.updateSelection must be invoked.

TextCellEditor.dependsOnExternalFocusListener() Subclasses that hook their own focus listener should override this method and return false.

TreePathViewerSorter.compare(viewer, parentPath, e1, e2) Subclasses may override.

(abstract) Viewer.setSelection(selection, reveal) Subclasses must implement this method.

Viewer.inputChanged(input, oldInput) Subclassers may override this method to do something when a viewer's input is set.

ViewerComparator.compare(viewer, e1, e2) Subclasses may override.

Enum.toString() An enum type should override this method when a more "programmer-friendly" string form exists.

ClassLoader.findResource(name) Class loader implementations should override this method to specify where to find resources.

ClassLoader.findResources(name) lass loader implementations should override this method to specify where to load resources from.

SecurityManager.checkSecurityAccess(target) If you override this method, then you should make a call to super.checkSecurityAccess at the point the overridden method would normally throw an exception.

SecurityManager.checkAccess(g) Applications that want a stricter policy should override this method.

InheritableThreadLocal.childValue(parentValue) This method merely returns its input argument, and should be overridden if a different behavior is desired.

ClassLoader.findClass(name) This method should be overridden by class loader implementations that follow the delegation model for loading classes, and will be invoked by the @link #loadClass loadClass method after checking the parent class loader for the requested class.

Window.close() This framework method may be extended (super.close must be called).

class TreeNodeContentProvider This class and all of its methods may be overridden or extended.

TemplateProposal.apply(document) clients should overwrite @link #apply(ITextViewer, char, int, int) instead

TemplateProposal.apply(document, trigger, offset) clients should overwrite @link #apply(ITextViewer, char, int, int)

StatusLineManager.getProgressMonitorDelegate() Override this method to provide your own object used to handle progress.

ActionContributionItem.shortenText(textValue, item) Override if you need a different strategy.

class PopupDialog Clients are expected to override the creation of the main dialog area, and may optionally override the creation of the title area and info area in order to add content.

PopupDialog.setTabOrder(composite) Clients should override to introduce specific tab ordering.

PopupDialog.saveDialogBounds(shell) Subclasses typically need not override this method, but may extend it (calling super.saveDialogBounds if additional bounds information should be stored.

ListEditor.selectionChanged() Sublcasses may override.

(abstract) AbstractInformationControl.createContent(parent) Implementors must either use the dialog font or override @link #computeSizeConstraints(int, int).

StyledCellLabelProvider.prepareStyleRange(styleRange, applyColors) Clients can override.

StructuredViewer.getRoot() Override if the root element is different from the viewer's input element.

FocusCellHighlighter.focusCellChanged(newCell, oldCell) Subclasses should override this method rather than @link #focusCellChanged(ViewerCell)

DecoratingStyledCellLabelProvider.getDecorationStyle(element) Clients can override.

CellEditor.getDoubleClickTimeout() Clients may override, in particular, clients can return 0 to denote that two subsequent mouse clicks in a cell should not be interpreted as a double click.

AbstractTreeViewer.getChild(widget, index) Note that the default implementation is not very efficient and should be overridden if this class is implemented.

AbstractTreeViewer.isExpandable(elementOrTreePath) It may be overridden if necessary.

class Thread There are two ways to create a new thread of execution. One is to declare a class to be a subclass of Thread. This subclass should override the run method of class Thread.

abstract class AbstractList To implement a modifiable list, the programmer must additionally override the @link #set(int, Object) set(int, E) method (which otherwise throws an UnsupportedOperationException).

abstract class AbstractList If the list is variable-size the programmer must additionally override the @link #add(int, Object) add(int, E) and @link #remove(int) methods.

ComparableComparator.equals(object) Subclasses may want to override this behavior to remain consistent with the @link Comparator#equals(Object) contract.

ComparatorChain.equals(object) Subclasses may want to override this behavior to remain consistent with the @link Comparator#equals(Object) contract.

ReverseComparator.equals(object) Subclasses may want to override this behavior to remain consistent with the @link Comparator#equals(Object) equals contract.

PredicatedCollection.add(object) Override to validate the object being added to ensure it matches the predicate.

PredicatedCollection.addAll(coll) Override to validate the objects being added to ensure they match the predicate.

AbstractLinkedList.isEqualValue(value1, value2) Subclasses can override this to match differently.
AbstractLinkedList.updateNode(node, value) Subclasses can override this to record the change.
AbstractLinkedList.createHeaderNode() Subclasses can override this to create a different class.
AbstractLinkedList.createNode(value) Subclasses can override this to create a different class.
CursorableLinkedList.updateNode(node, value) Subclasses can override this to record the change.
TransformedMap.checkSetValue(value) Override to transform the value when using setValue.
AbstractReferenceMap.hashEntry(key, value) Subclasses can override this, for example to use the identityHashCode.
AbstractReferenceMap.doWriteObject(out) Subclasses may override if they have a specific field that must be present on read before this implementation will work.
AbstractReferenceMap.doReadObject(in) Subclasses may override if the subclass has a specific field that must be present before put() or calculateThreshold() will work correctly.
AbstractHashedMap.convertKey(key) Subclasses can override this to perform alternate key conversions.
AbstractHashedMap.hash(key) Subclasses can override this to return alternate hash codes.
AbstractHashedMap.isEqualKey(key1, key2) Subclasses can override this to match differently. @param key1 the first key to compare passed in from outside
AbstractHashedMap.isEqualValue(value1, value2) Subclasses can override this to match differently.
AbstractHashedMap.hashIndex(hashCode, dataSize) Subclasses can override this to return alternate bucketing
AbstractHashedMap.updateEntry(entry, newValue) Subclasses could override to handle changes to the map.
AbstractHashedMap.addMapping(hashIndex, hashCode, key, value) Subclasses could override to fully control adds to the map.
AbstractHashedMap.createEntry(next, hashCode, key, value) Subclasses can override this to return a different storage class, or implement caching.
AbstractHashedMap.addEntry(entry, hashIndex) Subclasses could override to handle changes to the map.
AbstractHashedMap.removeMapping(entry, hashIndex, previous) Subclasses could override to fully control removals from the map.
AbstractHashedMap.removeEntry(entry, hashIndex, previous) Subclasses could override to handle changes to the map.
AbstractHashedMap.destroyEntry(entry) Subclasses can override this to implement caching (override clear as well).
AbstractHashedMap.createEntrySetIterator() Subclasses can override this to return iterators with different properties.
AbstractHashedMap.createKeySetIterator() Subclasses can override this to return iterators with different properties.
AbstractHashedMap.createValuesIterator() Subclasses can override this to return iterators with different properties.
AbstractHashedMap.doWriteObject(out) This method must be overridden if a subclass must be setup before put() is used.
AbstractHashedMap.doReadObject(in) This method must be overridden if a subclass must be setup before put() is used.
AbstractHashedMap.doReadObject(in) Subclasses may override if the subclass has a specific field that must be present before put() or calculateThreshold() will work correctly.
AbstractHashedMap.doWriteObject(out) Subclasses may override if they have a specific field that must be present on read before this implementation will work.
AbstractLinkedMap.createEntrySetIterator() Subclasses can override this to return iterators with different properties
AbstractLinkedMap.createKeySetIterator() Subclasses can override this to return iterators with different properties.
AbstractLinkedMap.createValuesIterator() Subclasses can override this to return iterators with different properties.
MultiValueMap.createCollection(size) This method can be overridden to perform your own processing instead of using the factory.
Flat3Map.createDelegateMap() This can be overridden by subclasses to provide a different map implementation.
PredicatedMap.checkSetValue(value) Override to validate an object set into the map via setValue.
LRUMap.removeLRU(entry) This method exists for subclasses to override. A subclass may wish to provide cleanup of resources when an entry is removed.
DefaultKeyValue.hashCode() however subclasses may override this.
TransformedMap.checkSetValue(value) Override to transform the value when using setValue.
AbstractDualBidiMap.createKeySetIterator(iterator) Subclasses can override this to return iterators with different properties.
AbstractDualBidiMap.createValuesIterator(iterator) Subclasses can override this to return iterators with different properties.
AbstractDualBidiMap.createEntrySetIterator(iterator) Subclasses can override this to return iterators with different properties.
EnumerationUtils.toList(stringTokenizer) Override toList(Enumeration) for StringTokenizer as it implements Enumeration for the sake of backward compatibility.
abstract class ResourceBundle Your subclasses must override two methods: handleGetObject and getKeys().
abstract class AbstractPreferences Implementors must override the nine abstract service-provider interface (SPI) methods:
abstract class ListResourceBundle Subclasses must override getContents and provide an array, where each item in the array is a pair of objects.
abstract class AbstractCollection To implement a modifiable collection, the programmer must additionally override this class's add method (which otherwise throws an UnsupportedOperationException),
Random.next(bits) Subclasses should override this, as this is used by all other methods.
Calendar.roll(field, amount) Other subclasses should also provide overrides of this function that do the right thing.
Calendar.getActualMinimum(field) Subclasses should, if possible, override this with a more efficient implementation - in many cases, they can simply return getMinimum().
Calendar.getActualMaximum(field) Subclasses should, if possible, override this with a more efficient implementation.
AbstractMap.get(key) many implementations will override this method.
URLStreamHandler.equals(u1, u2) May be overidden by handlers for other protocols that have different requirements for equals().
URLStreamHandler.hashCode(u) May be overidden by handlers for other protocols that have different requirements for hashCode calculation.
abstract class AbstractQueuedSynchronizer Subclasses must define the protected methods that change this state, and which define what that state means in terms of this object being acquired or released.
abstract class AbstractQueue A Queue implementation that extends this class must minimally define a method @link Queue#offer which does not permit insertion of null elements, along with methods @link Queue#peek, @link Queue#poll, @link Collection#size, and a @link Collection#iterator supporting @link Iterator#remove.
X509Certificate.getSubjectX500Principal() It is recommended that subclasses override this method.
X509Certificate.getIssuerX500Principal() It is recommended that subclasses override this method.
X509CRL.getIssuerX500Principal() It is recommended that subclasses override this method.
AbstractCollection.clear() Most implementations will probably choose to override this method for efficiency.
URLConnection.getHeaderFieldDate(name, Default) Classes for that connection type can override this method and short-circuit the parsing.
RuleBasedCollator.compare(source, target) This can be overriden in a subclass.
URLStreamHandler.getDefaultPort() This method is meant to be overidden by handlers with default port numbers.
RuleBasedCollator.getCollationKey(source) It can be overriden in a subclass.
FutureTask.done() Subclasses may override this method to invoke completion callbacks or perform bookkeeping.
InputStream.read(b, off, len) Subclasses are encouraged to provide a more efficient implementation of this method.
AbstractList.addAll(index, c) Many implementations will override this method for efficiency.
OutputStream.write(b, off, len) Subclasses are encouraged to override this method and provide a more efficient implementation.
InputStream.skip(n) Subclasses are encouraged to provide a more efficient implementation of this method.
SourceViewerConfiguration.getIndentPrefixes(sourceViewer, contentType) Note: This default is incorrect but cannot be changed in order not to break any existing clients. Subclasses should overwrite this method and use @link #getIndentPrefixesForTab(int) if applicable.
ThreadGroup.uncaughtException(t, e) Applications can override this method in subclasses of ThreadGroup to provide alternative handling of uncaught exceptions.
class FilterInputStream The class FilterInputStream itself simply overrides all methods of InputStream with versions that pass all requests to the contained input stream. Subclasses of FilterInputStream may further override some of these methods and may also provide additional methods and fields.
Enum.toString() This method may be overridden, though it typically isn't necessary or desirable.
abstract class AbstractQueue Typically, additional methods will be overridden as well.
abstract class AbstractPreferences The implementor may, at his discretion, override one or more of the concrete methods if the default implementation is unsatisfactory for any reason, such as performance.
ResourceBundle.handleKeySet() Override this method in subclass implementations for faster handling.
PopupDialog.showDialogMenu() Subclasses typically do not override method.
DialogPage.convertHeightInCharsToPixels(chars) Clients may call this framework method, but should not override it.
DialogPage.convertHorizontalDLUsToPixels(dlus) Clients may call this framework method, but should not override it.
DialogPage.convertVerticalDLUsToPixels(dlus) Clients may call this framework method, but should not override it.
DialogPage.convertWidthInCharsToPixels(chars) Clients may call this framework method, but should not override it.
Dialog.convertHeightInCharsToPixels(chars) This method may only be called after initializeDialogUnits has been called.
Dialog.convertHorizontalDLUsToPixels(dlus) Clients may call this framework method, but should not override it.
Dialog.convertVerticalDLUsToPixels(dlus) Clients may call this framework method, but should not override it.
Dialog.convertWidthInCharsToPixels(chars) Clients may call this framework method, but should not override it.
AbstractInformationControl.removeFocusListener(listener) This method is not intended to be overridden by subclasses.
AbstractInformationControl.addFocusListener(listener) This method is not intended to be overridden by subclasses.

Dialog.getButtonBar() Clients should not override it.
SubContributionManager.itemAdded(item, wrap) Subclasses are not expected to override this method.
SubContributionManager.itemRemoved(item) Subclasses are not expected to override this method.
PopupDialog.createTitleMenuArea(parent) Subclasses typically need not override this method,
AbstractInformationControl.isFocusControl() This method is not intended to be overridden by subclasses.
Window.getShellStyle() Subclassers should call setShellStyle to change this value, rather than overriding this method.
EditingSupport.initializeCellEditorValue(cellEditor, cell) Standard customers should not overwrite this method but @link #getValue(Object)
EditingSupport.saveCellEditorValue(cellEditor, cell) Standard customers should not overwrite this method but @link #setValue(Object, Object)
AbstractTreeViewer.isSameSelection(items, current) This method is not intended to be overridden by subclasses.
ColumnViewer.isBusy() This method is not intended to be overridden by subclasses.
ColumnViewer.setBusy(busy) This method is not intended to be overridden by subclasses.
ColumnViewer.checkBusy() This method is not intended to be overridden by subclasses.
AbstractHoverInformationControlManager.getInternalAccessor() @nooverride This method is not intended to be re-implemented or extended by clients.
AbstractInformationControlManager.getInternalAccessor() @nooverride This method is not intended to be re-implemented or extended by clients.
AnnotationBarHoverManager.getInternalAccessor() @nooverride This method is not intended to be re-implemented or extended by clients.
Dialog.convertHeightInCharsToPixels(chars) Clients should not override it.
Dialog.getDialogArea() Clients should not override it.
FieldEditor.convertHorizontalDLUsToPixels(control, dlus) Clients should not override it.
FieldEditor.convertVerticalDLUsToPixels(control, dlus) Clients should not override it.
AnnotationBarHoverManager.getInternalAccessor() This method is not intended to be referenced or overridden by clients.
TemplateCompletionProcessor.createProposal(template, context, region, relevance) Do neither call nor override.
AbstractHoverInformationControlManager.getInternalAccessor() Note: This method is not intended to be referenced or overridden by clients.
OpenStrategy.getOpenMethod() This method is internal to the framework; it should not be implemented outside the framework.
OpenStrategy.setOpenMethod(method) This method is internal to the framework; it should not be implemented outside the framework.
AbstractInformationControlManager.getInternalAccessor() Note: This method is not intended to be referenced or overridden by clients.

2.3 Method Extension Directive

CoolBarManager.update(force) Subclasses may extend this IContributionManager method, but must call super.update
CoolBarManager.itemAdded(item) Subclassesmust call super.itemAdded
CoolBarManager.itemRemoved(item) Subclasses must call super.itemRemoved
BaseLabelProvider.dispose() Subclasses should call the super implementation
PopupDialog.getForegroundColorExclusions() Subclasses should always call super.getForegroundColorExclusions to aggregate the list.
PopupDialog.getBackgroundColorExclusions() Subclasses should always call super.getBackgroundColorExclusions to aggregate the list.
ViewerColumn.handleDispose() Subclasses must call the super implementation.
ViewerColumn.setEditingSupport(editingSupport) Subclasses must call the super implementationcall
ViewerColumn.setEditingSupport(editingSupport) Subclasses must call the super implementation.
CellLabelProvider.initialize(viewer, column) Subclasses should call the super implementation (which at this time is empty but may be changed in the future)
CellLabelProvider.dispose(viewer, column) Subclasses should call the super implementation
Dialog.buttonPressed(buttonId) Subclasses should call super.buttonPressed if the default handling of the ok and cancel buttons is desired.
PreferencePage.performDefaults() Subclasses should call super.performDefaults.
OwnerDrawLabelProvider.initialize(viewer, column) Subclasses should either call the super implementation or, alternatively, @link #initialize(ColumnViewer, ViewerColumn, boolean)
OwnerDrawLabelProvider.erase(event, element) Subclasses should not call the super implementation.
AbstractColorChooserPanel.uninstallChooserPanel(enclosingChooser) f override this, be sure to call super.
AbstractLinkedList.init() Subclasses which override this method should make sure they call super
FieldEditorPreferencePage.dispose() Subclasses call super.dispose
AbstractColorChooserPanel.installChooserPanel(enclosingChooser) If you override this, be sure to call super.
PreferencePage.doComputeSize() All pages should call super.doComputeSize.
Component.removeNotify() Code overriding this method should call super.removeNotify as the first line of the overriding method.
RequiredModelMBean.postDeregister() Any subclass of RequiredModelMBean overloading or overriding this method should call super.postDeregister()
NavigationFilter.setDot(fb, dot, bias) Subclasses may wish to conditionally call super with a different location
NavigationFilter.moveDot(fb, dot, bias) Subclasses may wish to conditionally call super with a different location
Canvas.update(g) applications that override this method should either call super.update(g) or incorporate the functionality described above into their own code
RequiredModelMBean.postRegister(registrationDone) Any subclass of RequiredModelMBean overloading or overriding this method should call super.postRegister(registrationDone)
RequiredModelMBean.preDeregister() Any subclass of RequiredModelMBean overloading or overriding this method should call super.preDeregister()
PopupDialog.createDialogArea(parent) Subclasses may call super as in the following example
Component.paint(g) Subclasses of Component that override this method need not call super.paint(g).
Canvas.paint(g) Applications that override this method need not call super.paint(g)
class BasicSplitPaneDivider To conditionally drag from certain areas subclass mousePressed and call super when you wish the dragging to begin.
Component.update(g) Subclasses of Component that override this method should either call super.update(g), or call paint(g) directly from their update method.
RequiredModelMBean.preRegister(server, name) Any subclass of RequiredModelMBean overloading or overriding this method should call super.preRegister(server, name)
AbstractInformationControl.setFocus() Subclasses can override or extend.
AbstractInformationControl.computeSizeConstraints(widthInChars, heightInChars) Subclasses can override or extend.
abstract class AbstractInformationControl They should also extend @link #computeTrim() if they create a content area with additional trim (e.g. scrollbars)
abstract class PreferencePage Subclasses should extend the doComputeSize framework method to compute the size of the page's control.
abstract class Window close- extend to free other SWT resources
abstract class Wizard extend addPages
abstract class Wizard extend dispose
Wizard.dispose() Subclasses should extend this method if the wizard instance maintains addition SWT resource that need to be disposed.
Wizard.addPages() Subclasses should extend if extra pages need to be added before the wizard opens.
AbstractInformationControl.computeTrim() Subclasses can extend this method to add additional trim
ContentViewer.getContentProvider() if overriding in a subclass, super.getContentProvider must be invoked.
ContentViewer.hookControl(control) however, super.hookControl must be invoked.
ContentViewer.setContentProvider(contentProvider) super.setContentProvider must be invoked.
ContentViewer.setLabelProvider(labelProvider) super.setLabelProvider must be invoked.
StructuredViewer.updateSelection(selection) super.updateSelection must be invoked.
ContentViewer.getLabelProvider() super.getLabelProvider must be invoked.
ContentViewer.handleDispose(event) overriding methods must invoke super.handleDispose
abstract class WizardPage isPageComplete - may be extended
abstract class WizardPage setDescription - may be extended
abstract class WizardPage setTitle - may be extended
abstract class WizardPage dispose - may be extended to dispose additional allocated SWT resources
PopupDialog.close() This method may be extended (super.close must be called).
SecurityManager.checkAccess(t) If this method is overridden, then super.checkAccess should be called by the first statement in the overridden method, or the equivalent security check should be placed in the overridden method.
SecurityManager.checkSecurityAccess(target) If you override this method, then you should make a call to super.checkSecurityAccess at the point the overridden method would normally throw an exception.
SecurityManager.checkCreateClassLoader() If you override this method, then you should make a call to super.checkCreateClassLoader at the point the overridden method would normally throw an exception.
SecurityManager.checkAccess(g) If this method is overridden, then super.checkAccess should be called by the first statement in the overridden method, or the equivalent security check should be placed in the overridden method.

SecurityManager.checkPrintJobAccess() If you override this method, then you should make a call to super.checkPrintJobAccess at the point the overridden method would normally throw an exception.
SecurityManager.checkSystemClipboardAccess() If you override this method, then you should make a call to super.checkSystemClipboardAccess at the point the overridden method would normally throw an exception.
SecurityManager.checkAwtEventQueueAccess() If you override this method, then you should make a call to super.checkAwtEventQueueAccess at the point the overridden method would normally throw an exception.
SecurityManager.checkTopLevelWindow(window) If you override this method, then you should make a call to super.checkTopLevelWindow at the point the overridden method would normally return false, and the value of super.checkTopLevelWindow should be returned.
SecurityManager.checkRead(fd) If you override this method, then you should make a call to super.checkRead at the point the overridden method would normally throw an exception.
SecurityManager.checkRead(file) If you override this method, then you should make a call to super.checkRead at the point the overridden method would normally throw an exception.
SecurityManager.checkWrite(fd) If you override this method, then you should make a call to super.checkWrite at the point the overridden method would normally throw an exception.
SecurityManager.checkWrite(file) If you override this method, then you should make a call to super.checkWrite at the point the overridden method would normally throw an exception.
SecurityManager.checkDelete(file) If you override this method, then you should make a call to super.checkDelete at the point the overridden method would normally throw an exception
SecurityManager.checkListen(port) If you override this method, then you should make a call to super.checkListen at the point the overridden method would normally throw an exception.
SecurityManager.checkMulticast(maddr) If you override this method, then you should make a call to super.checkMulticast at the point the overridden method would normally throw an exception.
SecurityManager.checkPropertiesAccess() If you override this method, then you should make a call to super.checkPropertiesAccess at the point the overridden method would normally throw an exception.
Window.close() This framework method may be extended (super.close must be called).
SecurityManager.checkExit(status) If you override this method, then you should make a call to super.checkExit at the point the overridden method would normally throw an exception. @param status the exit status.
PopupDialog.saveDialogBounds(shell) Subclasses typically need not override this method, but may extend it (calling super.saveDialogBounds if additional bounds information should be stored.
SecurityManager.checkConnect(host, port) If you override this method, then you should make a call to super.checkConnect at the point the overridden method would normally throw an exception.
SecurityManager.checkAccept(host, port) If you override this method, then you should make a call to super.checkAccept at the point the overridden method would normally throw an exception.
SecurityManager.checkMulticast(maddr, ttl) If you override this method, then you should make a call to super.checkMulticast at the point the overridden method would normally throw an exception.
SecurityManager.checkPropertyAccess(key) If you override this method, then you should make a call to super.checkPropertyAccess at the point the overridden method would normally throw an exception.
SecurityManager.checkPackageAccess(pkg) If this method is overridden, then super.checkPackageAccess should be called as the first line in the overridden method. @param pkg the package name.
SecurityManager.checkPackageDefinition(pkg) If this method is overridden, then super.checkPackageDefinition should be called as the first line in the overridden method.
SecurityManager.checkSetFactory() If you override this method, then you should make a call to super.checkSetFactory at the point the overridden method would normally throw an exception.
Object.clone() By convention, the returned object should be obtained by calling super.clone. If a class and all of its superclasses (except Object) obey this convention, it will be the case that x.clone().getClass() == x.getClass().
SecurityManager.checkConnect(host, port, context) If you override this method, then you should make a call to super.checkConnect at the point the overridden method would normally throw an exception.
ThreadPoolExecutor.terminated() Note: To properly nest multiple overridings, subclasses should generally invoke super.terminated within this method.
class ScheduledThreadPoolExecutor To preserve functionality, any further overrides of these methods in subclasses must invoke superclass versions, which effectively disables additional task customization.
ThreadPoolExecutor.beforeExecute(t, r) Note: To properly nest multiple overridings, subclasses should generally invoke super.beforeExecute at the end of this method.
ThreadPoolExecutor.afterExecute(r, t) Note: To properly nest multiple overridings, subclasses should generally invoke super.afterExecute at the beginning of this method.
StringFieldEditor.doFillIntoGrid(parent, numColumns) Subclasses may override but must call super.doFillIntoGrid.
ViewerColumn.setLabelProvider(labelProvider) Subclasses may extend but must call the super implementation.
Dialog.createDialogArea(parent) Subclasses may call super as in the following example:
CertPath.equals(other) This algorithm is implemented by this method. If it is overridden, the behavior specified here must be maintained.
SecurityManager.checkMemberAccess(clazz, which) If this method is overridden, then a call to super.checkMemberAccess cannot be made, as the default implementation of checkMemberAccess relies on the code being checked being at a stack depth of 4.

2.4 Call Contract Subclassing Directive

AbstractInformationControl.create() Subclasses must call this method at the end of their constructor(s)
Activatable.Activatable(id, port) A concrete subclass of this class must call this constructor when it is activated via the two parameter constructor described above
Activatable.Activatable(id, port, csf, ssf) A concrete subclass of this class must call this constructor when it is activated via the two parameter constructor described above.
AbstractPreferences.newNode The field is initialized to false, but may be set to true by a subclass constructor (and should not be modified thereafter)
Component.update(g) Subclasses of Component that override this method should either call super.update(g), or call paint(g) directly from their update method.
Wizard.addPages() New pages should be added by calling addPage.
ErrorDialog.setStatus(status) The message, image and title should be updated by the subclass, if desired
ContentViewer.hookControl(control) All subclasses must call this method when their control is first established.
abstract class IconAndMessageDialog Clients are expected to call @link #createMessageArea(Composite), otherwise neither the icon nor the message will appear.
IconAndMessageDialog.createMessageArea(composite) Clients are expected to call this method, otherwise neither the icon nor the message will appear.
StringFieldEditor.setValidateStrategy(value) Thus this method is really only useful for subclasses to call in their constructor. However, it has public visibility for backward compatibility.
PreferencePage.contributeButtons(parent) For each button a subclass contributes, it must also increase the parent's grid layout number of columns by one; that is,
((GridLayout) parent.getLayout()).numColumns++);
PreferencePage.noDefaultAndApplyButton() Subclasses wishing a preference page without these buttons should call this framework method before the page's control has been created.
(abstract) FieldEditorPreferencePage.createFieldEditors() Subclasses should call getFieldEditorParent to obtain the parent control for each field editor.
(abstract) CompositeImageDescriptor.drawCompositeImage(width, height) Subclasses must implement this framework method to paint images within the given bounds using one or more calls to the drawImage framework method.
class FastPartitioner Subclasses need to call @link #clearPositionCache() after modifying the partitioner's positions.
SourceViewerConfiguration.getIndentPrefixes(sourceViewer, contentType) Subclasses should overwrite this method and use @link #getIndentPrefixesForTab(int) if applicable.
abstract class ProxySelector When a proxy selector is registered, for instance, a subclass of URLConnection class should call the @link #select select method for each URL request so that the proxy selector can decide if a direct, or proxied connection should be used.
abstract class ProxySelector If a connection cannot be established to a proxy (PROXY or SOCKS) servers then the caller should call the proxy selector's @link #connectFailed connectFailed method to notify the proxy selector that the proxy server is unavailable.
ContentHandler.getContent(urlc, classes) The default implementation of this method should call getContent() and screen the return type for a match of the suggested types.
SecurityManager.checkAccess(t) If this method is overridden, the method that overrides it should additionally check to see if the calling thread has the RuntimePermission("modifyThread") permission, and if so, return silently.

abstract class Wizard Subclasses may call the following methods to configure the wizard:
* addPage * setHelpAvailable * setDefaultPageImageDescriptor * setDialogSettings * setNeedsProgressMonitor * setTitleBarColor * setWindowTitle

abstract class WizardPage Subclasses may call the following methods to configure the wizard page:
* setDescription * setErrorMessage * setImageDescriptor * setMessage * setPageComplete * setPreviousPage * setTitle

Format.formatToCharacterIterator(obj) Subclasses that support fields should override this and create an AttributedCharacterIterator with meaningful attributes.

AbstractInformationControl.AbstractInformationControl(parentShell, statusFieldText) Subclasses are required to call @link #create() at the end of their constructor.

AbstractInformationControl.AbstractInformationControl(parentShell, toolBarManager) Subclasses are required to call @link #create() at the end of their constructor.

AbstractInformationControl.AbstractInformationControl(parentShell, isResizable) Subclasses are required to call @link #create() at the end of their constructor.

CellEditor.setValueValid(valid) Subclasses should call this method on construction.

ColumnViewer.setBusy(busy) Subclasses MUST use try ...finally as follows to ensure that the busy flag is reset to its original value:
boolean oldBusy = isBusy(); setBusy(true); try // do work finally setBusy(oldBusy);

ColumnViewer.checkBusy() It is recommended that this method be used by subclasses to determine whether the viewer is busy to return early from state-changing methods.

OwnerDrawLabelProvider.initialize(viewer, column, enableOwnerDraw) May be called from subclasses that override @link #initialize(ColumnViewer, ViewerColumn) but want to customize whether owner draw will be enabled.

StructuredViewer.internalRefresh(element, updateLabels) If this method is overridden to do the actual refresh, then internalRefresh(Object element) should simply call internalRefresh(element, true).

SecurityManager.checkExec(cmd) If you override this method, then you should make a call to super.checkExec at the point the overridden method would normally throw an exception.

SecurityManager.checkLink(lib) If you override this method, then you should make a call to super.checkLink at the point the overridden method would normally throw an exception.

SecurityManager.checkRead(file, context) If you override this method, then you should make a call to super.checkRead at the point the overridden method would normally throw an exception.

interface IInformationControlExtension5 Likewise, @link IInformationControl#addFocusListener(org.eclipse.swt.events.FocusListener) should install listeners for @link SWT#Activate and @link SWT#Deactivate on the shell and forward events to the focus listeners.

(abstract) IInformationControlExtension5.computeSizeConstraints(widthInChars, heightInChars) Implementors should use the main font of the information control to do the characters-to-pixels conversion. This is typically the @link JFaceResources#getDialogFont() dialog font.

TreeViewer.replace(parentElementOrTreePath, index, element) This method should be called by implementers of ILazyTreeContentProvider to populate this viewer.

(abstract) EditingSupport.setValue(element, value) Note that implementers need to ensure that getViewer().update(element, null) or similar methods are called, either directly or through some kind of listener mechanism on the implementer's model, to cause the new value to appear in the viewer.

AbstractHashedMap.reuseEntry(entry, hashIndex, hashCode, key, value) Subclasses could populate additional entry fields.

Executors.privilegedCallable(callable) This method should normally be invoked within an @link AccessController#doPrivileged action to create callables that will, if possible, execute under the selected permission settings holding within that action; or if not possible, throw an associated @link AccessControlException.

Executors.privilegedCallableUsingCurrentClassLoader(callable) This method should normally be invoked within an @link AccessController#doPrivileged action to create callables that will, if possible, execute under the selected permission settings holding within that action; or if not possible, throw an associated @link AccessControlException.

AbstractLinkedList.doWriteObject(outputStream) The first serializable subclass must call this method from writeObject.

AbstractLinkedList.doReadObject(inputStream) The first serializable subclass must call this method from readObject.

AbstractLinkedList.AbstractLinkedList() If this constructor is used by a serializable subclass then the init() method must be called.

class ObjectInputStream In this case it is the responsibility of the subclass to save and restore the state of the non-serializable class.

(abstract) SQLData.writeSQL(stream) The implementation of the method must follow this protocol: It must write each of the attributes of the SQL type to the given output stream. This is done by calling a method of the output stream to write each item, in the order that they appear in the SQL definition of the type. Specifically, it must call the appropriate SQLOutput writer method(s) (writeInt, writeString, and so on) to do the following: for a Distinct Type, write its single data element; for a Structured Type, write a value for each attribute of the SQL type.

DriverManager.registerDriver(driver) A newly-loaded driver class should call the method registerDriver to make itself known to the DriverManager.

interface Thread.UncaughtExceptionHandler If the ThreadGroup object has no special requirements for dealing with the exception, it can forward the invocation to the @linkplain #getDefaultUncaughtExceptionHandler default uncaught exception handler.

abstract class PermissionCollection Whenever a permission is added to a heterogeneous PermissionCollection such as Permissions, and the PermissionCollection doesn't yet contain a PermissionCollection of the specified permission's type, the PermissionCollection should call the newPermissionCollection method on the permission's class to see if it requires a special PermissionCollection.

interface IMenuCreator Note that it is the responsibility of the implementor to dispose of SWT menus it creates.

class SQLRecoverableException At a minimum, the recovery operation must include closing the current connection and getting a new connection.

(abstract) IDialogPage.createControl(parent) Implementors are responsible for ensuring that the created control can be accessed via getControl

class ErrorManager When processing logging output, if a Handler encounters problems then rather than throwing an Exception back to the issuer of the logging call (who is unlikely to be interested) the Handler should call its associated ErrorManager.

(abstract) ScriptContext.setBindings(bindings, scope) Calls to the getAttribute and setAttribute methods must map to the get and put methods of the Bindings for the specified scope.

(abstract) AbstractInformationControlManager.computeInformation() Implementation of this method must finish their computation by setting the computation results using setInformation.

(abstract) IQuickAssistAssistant.setProposalSelectorForeground(foreground) Note: As of 3.4, you should only call this method if you want to override the @link JFacePreferences#CONTENT_ASSIST_FOREGROUND_COLOR.

abstract class PreferencePage Subclasses may call the noDefaultAndApplyButton framework method before the page's control has been created to suppress the standard Apply and Defaults buttons.

2.5 Method Implementation Directive

BigDecimal.setScale(newScale, roundingMode) If the scale is reduced by the operation, the unscaled value must be divided (rather than multiplied), and the value may be changed; in this case, the specified rounding mode is applied to the division.

FilePermission.newPermissionCollection() FilePermission objects must be stored in a manner that allows them to be inserted into the collection in any order

PopupDialog.createDialogArea(parent) This method must not modify the parent's layout.

Dialog.createDialogArea(parent) This method must not modify the parent's layout.

LinkedBlockingQueue.toArray() In other words, this method must allocate a new array

PriorityBlockingQueue.toArray() In other words, this method must allocate a new array

LinkedBlockingDeque.toArray() In other words, this method must allocate a new array

CopyOnWriteArrayList.toArray() In other words, this method must allocate a new array

ConcurrentLinkedQueue.toArray() In other words, this method must allocate a new array

ArrayBlockingQueue.toArray() In other words, this method must allocate a new array

PriorityQueue.toArray() In other words, this method must allocate a new array

(abstract) Collection.equals(o) The general contract for the Object.equals method states that equals must be symmetric (in other words, a.equals(b) if and only if b.equals(a)).

(abstract) Condition.awaitNanos(nanosTimeout) In either case the implementation must ensure that the signal is redirected to another waiting thread, if there is one.

SocketPermission.newPermissionCollection() SocketPermission objects must be stored in a manner that allows them to be inserted into the collection in any orde

URI.parseServerAuthority() They therefore must either be parsed as such or treated as an error.

BigDecimal.setScale(newScale, roundingMode) If the scale is reduced by the operation, the unscaled value must be divided (rather than multiplied), and the value may be changed

BasicPermission.newPermissionCollection() BasicPermission objects must be stored in a manner that allows them to be inserted in any order,

SocketOptions.SO_BINDADDR This option must be specified in the constructor.

(abstract) Comparator.compare(o1, o2) The implementor must also ensure that the relation is transitive

(abstract) Comparator.compare(o1, o2) Finally, the implementor must ensure that compare(x, y)==0 implies that sgn(compare(x, z))==sgn(compare(y, z)) for all z.

interface Statement All execution methods in the Statement interface implicitly close a statment's current ResultSet object if an open one exists.

interface ParameterizedType Instances of classes that implement this interface must implement an equals() method that equates any two instances that share the same generic type declaration and have equal type parameters.

ECFieldF2m.ECFieldF2m(m, ks) A valid reduction polynomial is either a trinomial (Xm̂ + Xk̂ + 1 with m ¿ k ¿= 1) or a pentanomial (Xm̂ + Xk̂3 + Xk̂2 + Xk̂1 + 1 with m ¿ k3 ¿ k2 ¿ k1 ¿= 1), so ks should have length 1 or 3.

(abstract) Statement.setCursorName(name) Note: By definition, the execution of positioned updates and deletes must be done by a different Statement object than the one that generated the ResultSet object being used for positioning.

interface TypeVariable Creating a type variable must not cause the creation of its bounds. Repeated creation of a type variable has no effect.

interface TypeVariable However, all instances representing a type variable must be equal() to each other.

ObjectOutputStream.replaceObject(obj) When a subclass is replacing objects it must insure that either a complementary substitution must be made during deserialization or that the substituted object is compatible with every field where the reference will be stored.

ObjectInputStream.resolveObject(obj) When a subclass is replacing objects it must insure that the substituted object is compatible with every field where the reference will be stored.

interface BidiMap This map enforces the restriction that there is a 1:1 relation between keys and values, meaning that multiple keys cannot map to the same value. This is required so that "inverting" the map results in a map without duplicate keys.

OutputStream.flush() The general contract of flush is that calling it is an indication that, if any bytes previously written have been buffered by the implementation of the output stream, such bytes should immediately be written to their intended destination.

(abstract) Instrumentation.redefineClasses(definitions) The redefinition must not add, remove or rename fields or methods, change the signatures of methods, or change inheritance.

(abstract) Statement.setQueryTimeout(seconds) A JDBC driver must apply this limit to the execute, executeQuery and executeUpdate methods.

interface Set The additional stipulation on constructors is, not surprisingly, that all constructors must create a set that contains no duplicate elements (as defined above).

Assert.isNotNull(object, message) Implementations are encouraged to make regular use of Assert.isNotNull to ensure that null parameters are detected as early as possible.

URL.equals(obj) Two hosts are considered equivalent if both host names can be resolved into the same IP addresses; else if either host name can't be resolved, the host names must be equal without regard to case; or both host names equal to null.

URI.equals(ob) For two URIs to be considered equal requires that either both are opaque or both are hierarchical. Their schemes must either both be undefined or else be equal without regard to case. Their fragments must either both be undefined or else be equal. For two opaque URIs to be considered equal, their scheme-specific parts must be equal. For two hierarchical URIs to be considered equal, their paths must be equal and their queries must either both be undefined or else be equal. Their authorities must either both be undefined, or both be registry-based, or both be server-based. If their authorities are defined and are registry-based, then they must be equal. If their authorities are defined and are server-based, then their hosts must be equal without regard to case, their port numbers must be equal, and their user-information components must be equal.

(abstract) SQLData.writeSQL(stream) The implementation of the method must follow this protocol: It must write each of the attributes of the SQL type to the given output stream. This is done by calling a method of the output stream to write each item, in the order that they appear in the SQL definition of the type. Specifically, it must call the appropriate SQLOutput writer method(s) (writeInt, writeString, and so on) to do the following: for a Distinct Type, write its single data element; for a Structured Type, write a value for each attribute of the SQL type.

interface Serializable Classes that require special handling during the serialization and deserialization process must implement special methods with these exact signatures

interface Serializable A serializable class can declare its own serialVersionUID explicitly by declaring a field named "serialVersionUID" that must be static, final, and of type long

(abstract) Comparable.compareTo(o) The implementor must ensure sgn(x.compareTo(y)) == -sgn(y.compareTo(x)) for all x and y. (This implies that x.compareTo(y) must throw an exception iff y.compareTo(x) throws an exception.) The implementor must also ensure that the relation is transitive: (x.compareTo(y)¿0 && y.compareTo(z)¿0) implies x.compareTo(z)¿0. Finally, the implementor must ensure that x.compareTo(y)==0 implies that sgn(x.compareTo(z)) == sgn(y.compareTo(z)), for all z.

BufferedInputStream.markpos If markpos is not -1, then all bytes from positions buf[markpos] through buf[pos-1] must remain in the buffer array

(abstract) RemoteCall.getResultStream(success) . Should only succeed once per remote call.

(abstract) HttpURLConnection.disconnect() Calling disconnect() should not imply that this HttpURLConnection instance can be reused for other requests.

(abstract) SQLData.readSQL(stream, typeName) The implementation of the method must follow this protocol: * It must read each of the attributes or elements of the SQL type from the given input stream. This is done by calling a method of the input stream to read each item, in the order that they appear in the SQL definition of the type. * The method readSQL then assigns the data to appropriate fields or elements (of this or other objects). Specifically, it must call the appropriate reader method (SQLInput.readString, SQLInput.readBigDecimal, and so on) method(s) to do the following: for a distinct type, read its single data element; for a structured type, read a value for each attribute of the SQL type.

(abstract) PKIXCertPathChecker.init(forward) A PKIXCertPathChecker must support reverse checking and may support forward checking.

Object.hashCode() The general contract of hashCode is: * Whenever it is invoked on the same object more than once during an execution of a Java application, the hashCode method must consistently return the same integer, provided no information used in equals comparisons on the object is modified. This integer need not remain consistent from one execution of an application to another execution of the same application. * If two objects are equal according to the equals(Object) method, then calling the hashCode method on each of the two objects must produce the same integer result. * It is not required that if two objects are unequal according to the @link java.lang.Object#equals(java.lang.Object) method, then calling the hashCode method on each of the two objects must produce distinct integer results. However, the programmer should be aware that producing distinct integer results for unequal objects may improve the performance of hashtables.

InputStream.mark(readlimit) Marking a closed stream should not have any effect on the stream.

CodeSource.implies(codesource) # codesource must not be null. # If this object's certificates are not null, then all of this object's certificates must be present in codesource's certificates. # If this object's location (getLocation()) is not null, then the following checks are made against this object's location and codesource's: 1. codesource's location must not be null. 2. If this object's location equals codesource's location, then return true. 3. This object's protocol (getLocation().getProtocol()) must be equal to codesource's protocol. 4. If this object's host (getLocation().getHost()) is not null, then the SocketPermission constructed with this object's host must imply the SocketPermission constructed with codesource's host. 5. If this object's port (getLocation().getPort()) is not equal to -1 (that is, if a port is specified), it must equal codesource's port. 6. If this object's file (getLocation().getFile()) doesn't equal codesource's file, then the following checks are made: If this object's file ends with "/-", then codesource's file must start with this object's file (exclusive the trailing "-"). If this object's file ends with a "/*", then codesource's file must start with this object's file and must not have any further "/" separators. If this object's file doesn't end with a "/", then codesource's file must match this object's file with a '/' appended. 7. If this object's reference (getLocation().getRef()) is not null, it must equal codesource's reference.

FieldPosition.FieldPosition(attribute, fieldID) Format subclasses that are aware of Field should give precedence to attribute and ignore fieldID if attribute is not null.

interface Driver When a Driver class is loaded, it should create an instance of itself and register it with the DriverManager. This means that a user can load and register a driver by calling Class.forName("foo.bah.Driver")

Proxy.isProxyClass(cl) The reliability of this method is important for the ability to use it to make security decisions, so its implementation should not just test if the class in question extends Proxy.

interface Key A Key should use KeyRep as its serialized representation.

(abstract) Transformer.transform(input) @param input the object to be transformed, should be left unchanged

(abstract) BidiMap.inverseBidiMap() Implementations should seek to avoid creating a new object every time this method is called.

class Normalizer When you are searching or comparing text, you must ensure that these two sequences are treated as equivalent. In addition, you must handle characters with more than one accent.

(abstract) ClassFileTransformer.transform(loader, className, classBeingRedefined, protectionDomain, classfileBuffer) @param classfileBuffer the input byte buffer in class file format - must not be modified

abstract class PKIXCertPathChecker A PKIXCertPathChecker implementation must support reverse checking (the ability to perform its checks when it is presented with certificates in the reverse direction) and may support forward checking (the ability to perform its checks when it is presented with certificates in the forward direction).

class DecimalFormat For example, either the positive and negative prefixes or the suffixes must be distinct for DecimalFormat.parse() to be able to distinguish positive from negative values. (If they are identical, then DecimalFormat will behave as if no negative subpattern was specified.) Another example is that the decimal separator and thousands separator should be distinct characters, or parsing will be impossible.

(abstract) Instrumentation.retransformClasses(classes) The retransformation must not add, remove or rename fields or methods, change the signatures of methods, or change inheritance.

interface X509Extension A certificate/CRL-using system (an application validating a certificate/CRL) must reject the certificate/CRL if it encounters a critical extension it does not recognize. A non-critical extension may be ignored if it is not recognized.

PKIXParameters.setCertPathCheckers(checkers) Regardless of whether these additional PKIXCertPathCheckers are set, a PKIX CertPathValidator or CertPathBuilder must perform all of the required PKIX checks on each certificate

(abstract) Bag.remove(object) According to the @link Collection#remove(Object) method, this method should only remove the first occurrence of the given object, not all occurrences.

ListUtils.subtract(list1, list2) This differs from @link List#removeAll(Collection) in that cardinality is respected; if list1 contains two occurrences of null and list2 only contains one occurrence, then the returned list will still contain one occurrence.

(abstract) Connection.setClientInfo(name, value) The driver stores the value specified in a suitable location in the database. For example in a special register, session parameter, or system table column. For efficiency the driver may defer setting the value in the database until the next time a statement is executed or prepared. Other than storing the client information in the appropriate place in the database, these methods shall not alter the behavior of the connection in anyway. The values supplied to these methods are used for accounting, diagnostics and debugging purposes only.

The driver shall generate a warning if the client info name specified is not recognized by the driver.

(abstract) Connection.setClientInfo(name, value) Drivers are not required to support these properties however if the driver supports a client info property that can be described by one of the standard properties, the standard property name should be used.

(abstract) Instrumentation.appendToSystemClassLoaderSearch(jarfile) The system class loader supports adding a JAR file to be searched if it implements a method named appendToClassPathForInstrumentation which takes a single parameter of type java.lang.String. The method is not required to have public access.

(abstract) Instrumentation.appendToSystemClassLoaderSearch(jarfile) The agent should take care to ensure that the JAR does not contain any classes or resources other than those to be defined by the system class loader for the purpose of instrumentation. Failure to observe this warning could result in unexpected behaviour that is difficult to diagnose (see @link #appendToBootstrapClassLoaderSearch appendToBootstrapClassLoaderSearch.

Constructor.newInstance(initargs) If the required access and argument checks succeed and the instantiation will proceed, the constructor's declaring class is initialized if it has not already been initialized.

(abstract) Connection.isValid(timeout) The driver shall submit a query on the connection or use some other mechanism that positively verifies the connection is still valid when this method is called.

The query submitted by the driver to validate the connection shall be executed in the context of the current transaction.

IdentityHashMap.keySet() The hashCode method of the returned set returns the sum of the identity hashcodes of the elements in the set, rather than the sum of their hashcodes. This is mandated by the change in the semantics of the equals method, in order to enforce the general contract of the Object.hashCode method among sets returned by this method.

(abstract) Comparable.compareTo(o) It is strongly recommended, but not strictly required that (x.compareTo(y)==0) == (x.equals(y)).

Math.copySign(magnitude, sign) Note that unlike the @link StrictMath#copySign(float, float) StrictMath.copySign method, this method does not require NaN sign arguments to be treated as positive values; implementations are permitted to treat some NaN arguments as positive and other NaN arguments as negative to allow greater performance.

Math.copySign(magnitude, sign) Note that unlike the @link StrictMath#copySign(double, double) StrictMath.copySign method, this method does not require NaN sign arguments to be treated as positive values; implementations are permitted to treat some NaN arguments as positive and other NaN arguments as negative to allow greater performance.

class Math Unlike some of the numeric methods of class StrictMath, all implementations of the equivalent functions of class Math are not defined to return the bit-for-bit same results. This relaxation permits better-performing implementations where strict reproducibility is not required.

class Math Therefore, most methods with more than 0.5 ulp errors are required to be semi-monotonic: whenever the mathematical function is non-decreasing, so is the floating-point approximation, likewise, whenever the mathematical function is non-increasing, so is the floating-point approximation.

class StrictMath To help ensure portability of Java programs, the definitions of some of the numeric functions in this package require that they produce the same results as certain published algorithms. These algorithms are available from the well-known network library netlib as the package ”Freely Distributable Math Library,” fdlibm.

DriverPropertyInfo.DriverPropertyInfo(name, value) The description and choices are intialized to null and required is initialized to false.

ActivationGroup.inactiveObject(id) This method simply informs the group's monitor that the object is inactive. It is up to the concrete subclass of ActivationGroup to fulfill the additional requirement of unexporting the object.

Signature.getParameters() The returned parameters may be the same that were used to initialize this signature, or may contain a combination of default and randomly generated parameter values used by the underlying signature implementation if this signature requires algorithm parameters but was not initialized with any.

CodeSource.equals(obj) Two CodeSource objects are considered equal if their locations are of identical value and if their signer certificate chains are of identical value. It is not required that the certificate chains be in the same order.

BigInteger.toByteArray() The byte array will be in big-endian byte-order: the most significant byte is in the zeroth element. The array will contain the minimum number of bytes required to represent this BigInteger, including at least one sign bit, which is (ceil((this.bitLength() + 1)/8)).

Identity.addCertificate(certificate) If the identity has a public key, the public key in the certificate must be the same, and if the identity does not have a public key, the identity's public key is set to be that specified in the certificate.

interface Owner The initial owner Principal should be specified as an argument to the constructor of the class implementing this interface.

interface SortedMap Note that the ordering maintained by a sorted map (whether or not an explicit comparator is provided) must be consistent with equals if the sorted map is to correctly implement the Map interface.

interface SortedMap All general-purpose sorted map implementation classes should provide four ”standard” constructors:

Pack200.Packer.EFFORT The special value 0 instructs the packer to copy through the original JAR file directly, with no compression. The JSR 200 standard requires any unpacker to understand this special case as a pass-through of the entire archive.

(abstract) SortedSet.subSet(fromElement, toElement) Implementations may, but are not required to, throw this exception if fromElement or toElement cannot be compared to elements currently in the set

(abstract) SortedSet.headSet(toElement) Implementations may, but are not required to, throw this exception if toElement cannot be compared to elements currently in the set. @throws NullPointerException if toElement is null and this set does not permit null elements

(abstract) SortedSet.tailSet(fromElement) Implementations may, but are not required to, throw this exception if fromElement cannot be compared to elements currently in the set. @throws NullPointerException if fromElement is null and this set does not permit null elements

(abstract) SortedMap.subMap(fromKey, toKey) Implementations may, but are not required to, throw this exception if fromKey or toKey cannot be compared to keys currently in the map.

(abstract) SortedMap.headMap(toKey) Implementations may, but are not required to, throw this exception if toKey cannot be compared to keys currently in the map.

(abstract) SortedMap.tailMap(fromKey) Implementations may, but are not required to, throw this exception if fromKey cannot be compared to keys currently in the map

(abstract) NavigableSet.subSet(fromElement, fromInclusive, toElement, toInclusive) Implementations may, but are not required to, throw this exception if fromElement or toElement cannot be compared to elements currently in the set.

(abstract) NavigableSet.headSet(toElement, inclusive) Implementations may, but are not required to, throw this exception if toElement cannot be compared to elements currently in the set.

(abstract) NavigableSet.tailSet(fromElement, inclusive) Implementations may, but are not required to, throw this exception if fromElement cannot be compared to elements currently in the set.

(abstract) NavigableMap.subMap(fromKey, fromInclusive, toKey, toInclusive) Implementations may, but are not required to, throw this exception if fromKey or toKey cannot be compared to keys currently in the map.

(abstract) NavigableMap.headMap(toKey, inclusive) Implementations may, but are not required to, throw this exception if toKey cannot be compared to keys currently in the map.

(abstract) NavigableMap.tailMap(fromKey, inclusive) Implementations may, but are not required to, throw this exception if fromKey cannot be compared to keys currently in the map.

(abstract) List.hashCode() The hash code of a list is defined to be the result of the following calculation:

int hashCode = 1; Iterator¡E¿ i = list.iterator(); while (i.hasNext()) E obj = i.next(); hashCode = 31*hashCode + (obj==null ? 0 : obj.hashCode());

This ensures that list1.equals(list2) implies that list1.hashCode()==list2.hashCode() for any two lists, list1 and list2, as required by the general contract of @link Object#hashCode.

BufferedReader.read(cbuf, off, len) Subclasses of this class are encouraged, but not required, to attempt to read as many characters as possible in the same fashion.

BufferedInputStream.read(b, off, len) Subclasses of this class are encouraged, but not required, to attempt to read as many bytes as possible in the same fashion.

URLClassLoader.getPermissions(codesource) If the protocol of this URL is ”jar”, then the permission granted is based on the permission that is required by the URL of the Jar file.

(abstract) DialogTray.createContents(parent) Tray implementions must not set a layout on the parent composite, or assume a particular layout on the parent.

Thread.setDefaultUncaughtExceptionHandler(eh) Note that the default uncaught exception handler should not usually defer to the thread's ThreadGroup object, as that could cause infinite recursion.

abstract class AbstractSet all of the methods and constructors in subclasses of this class must obey the additional constraints imposed by the Set interface

interface SortedSet All general-purpose sorted set implementation classes should provide four ”standard” constructors:

(abstract) DefaultInformationControl.IInformationPresenterExtension.updatePresentation(drawable, hoverInfo, presentation, maxWidth, maxHeight) Implementations should use the font of the given drawable to calculate the size of the text to be presented.

class ServiceLoader The only requirement enforced by this facility is that provider classes must have a zero-argument constructor so that they can be instantiated during loading.

(abstract) OrderedBidiMap.inverseBidiMap() Implementations should seek to avoid creating a new object every time this method is called. See AbstractMap.values() etc. Calling this method on the inverse map should return the original.

abstract class AbstractMap The programmer should generally provide a void (no argument) and map constructor, as per the recommendation in the Map interface specification.

class ServiceLoader The only requirement enforced by this facility is that provider classes must have a zero-argument constructor so that they can be instantiated during loading.

class Random In order to guarantee this property, particular algorithms are specified for the class Random. Java implementations must use all the algorithms shown here for the class Random, for the sake of absolute portability of Java code.

class Throwable Further, those subclasses that might likely have a cause associated with them should have two more constructors, one that takes a Throwable (the cause), and one that takes a String (the detail message) and a Throwable (the cause).

class LogManager Each class name must be for a Handler class which has a default constructor.

Object.equals(obj) The equals method implements an equivalence relation on non-null object references: * It is reflexive: for any non-null reference value x, x.equals(x) should return true. * It is symmetric: for any non-null reference values x and y, x.equals(y) should return true if and only if y.equals(x) returns true. * It is transitive: for any non-null reference values x, y, and z, if x.equals(y) returns true and y.equals(z) returns true, then x.equals(z) should return true. * It is consistent: for any non-null reference values x and y, multiple invocations of x.equals(y) consistently return true or consistently return false, provided no information used in equals comparisons on the objects is modified. * For any non-null reference value x, x.equals(null) should return false.

class HashMap If no such object exists, the map should be ”wrapped” using the @link Collections#synchronizedMap Collections.synchronizedMap method.

interface Collection All general-purpose Collection implementation classes (which typically implement Collection indirectly through one of its subinterfaces) should provide two ”standard” constructors: a void (no arguments) constructor, which creates an empty collection, and a constructor with a single argument of type Collection, which creates a new collection with the same elements as its argument.

AbstractHashedMap.clone() To implement clone(), a subclass must implement the Cloneable interface and make this method public.

interface Map All general-purpose map implementation classes should provide two ”standard” constructors: a void (no arguments) constructor which creates an empty map, and a constructor with a single argument of type Map, which creates a new map with the same key-value mappings as its argument.

(abstract) SortedBidiMap.inverseBidiMap() Implementations should seek to avoid creating a new object every time this method is called. See AbstractMap.values() etc. Calling this method on the inverse map should return the original.

(abstract) Comparator.equals(obj) This method must obey the general contract of @link Object#equals(Object).

AbstractList.modCount A single call to add(int, E) or remove(int) must add no more than one to this field, or the iterators (and list iterators) will throw bogus ConcurrentModificationExceptions.

interface BoundedCollection The size of the collection can vary, but it can never exceed a preset maximum number of elements.

(abstract) Condition.await() In that case the implementation must ensure that the signal is redirected to another waiting thread, if there is one.

class IdentityHashMap If no such object exists, the map should be ”wrapped” using the @link Collections#synchronizedMap Collections.synchronizedMap method.

class EnumMap If no such object exists, the map should be ”wrapped” using the @link Collections#synchronizedMap method.

AbstractQueuedSynchronizer.tryAcquire(arg) This method should query if the state of the object permits it to be acquired in the exclusive mode, and if so to acquire it.

class LinkedList If no such object exists, the list should be ”wrapped” using the @link Collections#synchronizedList Collections.synchronizedList method.

AbstractQueuedLongSynchronizer.tryAcquireShared(arg) This method should query if the state of the object permits it to be acquired in the shared mode, and if so to acquire it.

AbstractQueuedSynchronizer.tryAcquireShared(arg) This method should query if the state of the object permits it to be acquired in the shared mode, and if so to acquire it.

class LinkedHashMap If no such object exists, the map should be ”wrapped” using the @link Collections#synchronizedMap Collections.synchronizedMap method.

(abstract) Bag.retainAll(coll) Remove any members of the bag that are not in the given collection, respecting cardinality. That is, if the given collection coll contains n copies of a given object and the bag has m ¿ n copies, then delete m - n copies from the bag. In addition, if e is an object in the bag but !coll.contains(e), then remove e and any of its copies.

interface Lock It must also obey the interruption semantics as defined in this interface, to the extent that interruption of lock acquisition is supported: which is either totally, or only on method entry.

interface Lock All Lock implementations must enforce the same memory synchronization semantics as provided by the built-in monitor lock, as described in The Java Language Specification, Third Edition (17.4 Memory Model)

DelegatingDragAdapter.dragStart(event) A listener must set event.doit to false if it cannot handle the drag operation.

Arrays.deepEquals(a1, a2) Two possibly null elements e1 and e2 are deeply equal if any of the following conditions hold:

e1 and e2 are both arrays of object reference types, and Arrays.deepEquals(e1, e2) would return true e1 and e2 are arrays of the same primitive type, and the appropriate overloading of Arrays.equals(e1, e2) would return true. e1 == e2 e1.equals(e2) would return true.

Note that this definition permits null elements at any depth.

(abstract) IQuickAssistAssistant.canFix(annotation) This test must be fast and optimistic i.e. it is OK to return true even though there might be no quick fix.

(abstract) IQuickAssistProcessor.canFix(annotation) This test must be fast and optimistic i.e. it is OK to return true even though there might be no quick fix.

interface Deque While Deque implementations are not strictly required to prohibit the insertion of null elements, they are strongly encouraged to do so.

(abstract) AuthProvider.setCallbackHandler(handler) If the security property is not set, the provider is assumed to have alternative means for obtaining authentication information.

(abstract) ConcurrentNavigableMap.keySet() The view's iterator is a ”weakly consistent” iterator that will never throw @link ConcurrentModificationException, and guarantees to traverse elements as they existed upon construction of the iterator, and may (but is not guaranteed to) reflect any modifications subsequent to construction.

interface Acl The ACL Entries in each ACL observe the following rules:

Each principal can have at most one positive ACL entry and one negative entry; that is, multiple positive or negative ACL entries are not allowed for any principal. Each entry specifies the set of permissions that are to be granted (if positive) or denied (if negative).

If there is no entry for a particular principal, then the principal is considered to have a null (empty) permission set.

If there is a positive entry that grants a principal a particular permission, and a negative entry that denies the principal the same permission, the result is as though the permission was never granted or denied.

Individual permissions always override permissions of the group(s) to which the individual belongs. That is, individual negative permissions (specific denial of permissions) override the groups' positive permissions. And individual positive permissions override the groups' negative permissions.

(abstract) IQuickFixableAnnotation.isQuickFixableStateSet() Normally this means @link #setQuickFixable(boolean) has been called at least once but it can also be hard-coded, e.g. always return true.

interface RandomAccess Generic list algorithms are encouraged to check whether the given list is an instanceof this interface before applying an algorithm that would provide poor performance if it were applied to a sequential access list, and to alter their behavior if necessary to guarantee acceptable performance.

(abstract) ResultSet.rowDeleted() Note: Support for the rowDeleted method is optional with a result set concurrency of CONCUR_READ_ONLY

(abstract) IOverviewRuler.getAnnotationHeight() Assumes that all annotations are represented using the same height.

(abstract) ITextViewer.setVisibleRegion(offset, length) Every character inside the specified region is supposed to be visible in the viewer's widget after that call.

(abstract) IPresentationReconciler.install(viewer) After this method has been finished, the reconciler is operational. I.e., it works without requesting further client actions until uninstall is called.

(abstract) Set.remove(o) This set will not contain the element once the call returns

2.6 Non-local Consistency Subclassing Directive

MimeType.readExternal(in) The readExternal method must read the values in the same sequence and with the same types as were written by writeExternal

abstract class ViewerColumn Concrete subclasses of @link ColumnViewer should implement a matching concrete subclass of @link ViewerColumn.

abstract class Format Generally, a format's parseObject method must be able to parse any string formatted by its format method.

DialogCellEditor.createContents(cell) If you reimplement this method, you should also reimplement updateContents.

DialogCellEditor.updateContents(value) If you reimplement this method, you should also reimplement createContents.

(abstract) PersistentState.load(store, preferenceKey) This method must be symmetric with a call to @link #save(IPreferenceStore, String).

(abstract) PersistentState.save(store, preferenceKey) This method must be symmetric with a call to @link #load(IPreferenceStore, String).

(abstract) Instrumentation.isModifiableClass(theClass) For a class to be retransformed, @link #isRetransformClassesSupported must also be true. But the value of isRetransformClassesSupported() does not influence the value returned by this function. For a class to be redefined, @link #isRedefineClassesSupported must also be true. But the value of isRedefineClassesSupported() does not influence the value returned by this function.

ObjectInputStream.resolveProxyClass(interfaces) For a given subclass of ObjectInputStream that overrides this method, the annotateProxyClass method in the corresponding subclass of ObjectOutputStream must write any data or objects read by this method

class ObjectOutputStream Multiple objects or primitives can be written to the stream. The objects must be read back from the corresponding ObjectInputstream with the same types and in the same order as they were written.

FieldEditor.isValid() Subclasses wishing to perform validation should override both this method and refreshValidState.

FieldEditor.refreshValidState() Subclasses wishing to perform validation should override both this method and isValid.

interface RMIServerSocketFactory An implementation of this interface should implement @link Object#equals to return true when passed an instance that represents the same (functionally equivalent) server socket factory, and false otherwise (and it should also implement @link Object#hashCode consistently with its Object.equals implementation).

OutputStream.write(b) The general contract for write(b) is that it should have exactly the same effect as the call write(b, 0, b.length).

Window.getLayout() Subclasses that change the layout type by overriding this method should also override createContents.

Object.equals(obj) Note that it is generally necessary to override the hashCode method whenever this method is overridden, so as to maintain the general contract for the hashCode method, which states that equal objects must have equal hash codes.

interface PreparedStatement The setter methods (setShort, setString, and so on) for setting IN parameter values must specify types that are compatible with the defined SQL type of the input parameter. For instance, if the IN parameter has SQL type INTEGER, then the method setInt should be used.

(abstract) CallableStatement.registerOutParameter(parameterIndex, sqlType, typeName) When reading the value of an out parameter, you must use the getter method whose Java type corresponds to the parameter's registered SQL type.

(abstract) CallableStatement.registerOutParameter(parameterName, sqlType, typeName) When reading the value of an out parameter, you must use the getXXX method whose Java type XXX corresponds to the parameter's registered SQL type.

(abstract) Statement.executeBatch() However, the driver's behavior must be consistent with a particular DBMS, either always continuing to process commands or never continuing to process commands.

(abstract) Permission.hashCode() If two Permission objects are equal according to the equals method, then calling the hashCode method on each of the two Permission objects must produce the same integer result.

abstract class CertPathValidatorSpi All CertPathValidator implementations must include a class (the SPI class) that extends this class (CertPathValidatorSpi)

abstract class CertPathBuilderSpi All CertPathBuilder implementations must include a class (the SPI class) that extends this class (CertPathBuilderSpi) and implements all of its methods.

(abstract) Certificate.decode(stream) The format should be that returned by getFormat and produced by encode.

(abstract) Wrapper.isWrapperFor(iface) If this method returns true then calling unwrap with the same argument should succeed.

class CloneNotSupportedException Applications that override the clone method can also throw this exception to indicate that an object could not or should not be cloned.

interface Cloneable By convention, classes that implement this interface should override Object.clone (which is protected) with a public method.

Dialog.getDialogBoundsStrategy() This strategy is ignored if the implementer does not also specify the dialog settings for storing the bounds in Dialog.getDialogBoundsSettings().

interface Comparable The natural ordering for a class C is said to be consistent with equals if and only if e1.compareTo(e2) == 0 has the same boolean value as e1.equals(e2) for every e1 and e2 of class C.

class ResourceBundle.Control The formats returned by the @link Control#getFormats(String) getFormats method and candidate locales returned by the @link ResourceBundle.Control#getCandidateLocales(String, Locale) getCandidateLocales method must be consistent in all ResourceBundle.getBundle invocations for the same base bundle.

interface ReadWriteLock All ReadWriteLock implementations must guarantee that the memory synchronization effects of writeLock operations (as specified in the @link Lock interface) also hold with respect to the associated readLock

(abstract) ScriptEngine.get(key) Must have the same effect as getBindings(ScriptContext.ENGINE_SCOPE).get

(abstract) ScriptEngine.getBindings(scope) The Bindings instances that are returned must be identical to those returned by the getBindings method of ScriptContext called with corresponding arguments on the default ScriptContext of the ScriptEngine.

MathContext.MathContext(val) The string must be in the same format as that produced by the @link #toString method.

(abstract) DatabaseMetaData.supportsGetGeneratedKeys() If true is returned, the JDBC driver must support the returning of auto-generated keys for at least SQL INSERT statements

Window.createContents(parent) Subclasses may modify the parent's layout if they overload getLayout() to return null.

ErrorDialog.createDropDownList(parent) The visibility of the details button is controlled by @link #shouldShowDetailsButton(), which should also be overridden together with this method.

ContentAssistant.setProposalSelectorBackground(background) As of 3.4, you should only call this method if you want to override the @link JFacePreferences#CONTENT_ASSIST_B

interface IDocumentAdapter Clients may implement this interface and override TextViewer.createDocumentAdapter if they want to intercept the communication between the viewer's text widget and the viewer's document.

Viewer.getData(key) Overriding this method is generally not required if the number of different keys is small. If a more efficient representation of a viewer's properties is required, override both getData and setData.

(abstract) IContextInformation.equals(object) As specified in @link Object#equals(Object) clients will most likely also have to implement @link Object#hashCode().

(abstract) ICompletionProposalExtension6.getStyledDisplayString() @link ICompletionProposal#getDisplayString() still needs to be correctly implemented as this method might be ignored in case of uninstalled owner draw support.

interface IMarkSelection This interface is not intended to be implemented by clients other than @link org.eclipse.jface.text.ITextViewer implementers.

interface IInformationControlExtension5 Enriching this information control only works properly if @link IInformationControl#isFocusControl() is implemented like this (fShell is the control's shell): return fShell.getDisplay().getActiveShell() == fShell

abstract class ViewerRow Implementors of @link ColumnViewer have to provide a concrete implementation for the underlying widget

Viewer.setData(key, value) Overriding this method is generally not required if the number of different keys is small. If a more efficient representation of a viewer's properties is required, override both getData and setData.

AbstractHashedMap.destroyEntry(entry) Subclasses can override this to implement caching (override clear as well).

(abstract) IQuickAssistAssistant.install(sourceViewer) This quick assist assistant will only be able to query the invocation context if sourceViewer also implements @link ISourceViewerExtension3.

(abstract) Collection.hashCode() programmers should take note that any class that overrides the Object.equals method must also override the Object.hashCode method in order to satisfy the general contract for the Object.hashCodemethod

(abstract) Externalizable.readExternal(in) The readExternal method must read the values in the same sequence and with the same types as were written by writeExternal.

ManagementFactory.getPlatformMBeanServer() @exception SecurityException if there is a security manager and the caller does not have the permission required by @link javax.management.MBeanServerFactory#createMBeanServer.

interface Delayed An implementation of this interface must define a compareTo method that provides an ordering consistent with its getDelay method

interface ReadWriteLock All ReadWriteLock implementations must guarantee that the memory synchronization effects of writeLock operations (as specified in the @link Lock interface) also hold with respect to the associated readLock.

class TreeMap Note that the ordering maintained by a sorted map (whether or not an explicit comparator is provided) must be consistent with equals if this sorted map is to correctly implement the Map interface.

interface Comparator In order for the data structure to serialize successfully, the comparator (if provided) must implement Serializable.

FormattableFlags.UPPERCASE The output should be equivalent the following invocation of @link String#toUpperCase(java.util.Locale) out.toUpperCase()

abstract class AbstractSequentialList For a modifiable list the programmer should additionally implement the list iterator's set method.

abstract class AbstractSequentialList For a variable-size list the programmer should additionally implement the list iterator's remove and add methods.

(abstract) BidiMap.removeValue(value) Implementations should seek to make this method perform equally as well as remove(Object)

(abstract) BidiMap.getKey(value) Implementations should seek to make this method perform equally as well as get(Object).

CursorableLinkedList.init() The equivalent of a default constructor called by any constructor and by readObject.

abstract class Preferences Every Preferences implementation must have an associated @link PreferencesFactory implementation.

abstract class Preferences Every Java(TM) SE implementation must provide some means of specifying which PreferencesFactory implementation is used to generate the root preferences nodes.

class TreeSet Note that the ordering maintained by a set (whether or not an explicit comparator is provided) must be consistent with equals if it is to correctly implement the Set interface.

interface PreferencesFactory Providers of new @link Preferences implementations should provide corresponding PreferencesFactory implementations so that the new Preferences implementation can be installed in place of the platform-specific default implementation.

ContentAssistant.setRestoreCompletionProposalSize(dialogSettings) This API is only valid if the information control implements @link org.eclipse.jface.text.IInformationControlEx Not following this restriction will later result in an @link UnsupportedOperationException

AbstractInformationControlManager.setRestoreInformationControlBounds(dialogSettings, restoreLocation, restoreSize) This API is only valid if the information control implements @link IInformationControlExtension3. Not following this restriction will later result in an @link UnsupportedOperationException.

TableTreeViewer.getColumnProperties() Returns the column properties of this viewer. The properties must correspond with the columns of the table control.

TemplateReaderWriter.readSingle(reader, id) The reader must present a serialized form as produced by the save method.

ContentAssistEvent.isAutoActivated Note: This flag is only valid in @link ICompletionListener#assistSessionStarted(ContentAssistEvent).

class SignedObject However, it is crucial that the programmer writing the verifier code be aware what Signature engine is being used, as its own implementation of the verify method is invoked to verify a signature.

interface IReconcilingStrategy If a reconcile strategy consists of several steps between which model transformation is desired the each step should implement @link org.eclipse.jface.text.reconciler.IReconcileStep.
(abstract) IContentAssistSubjectControl.appendVerifyKeyListener(verifyKeyListener) Note: This content assist subject control may not support appending a verify listener, in which case false will be returned. If this content assist subject control only supports addVerifyKeyListener then this method can be used but prependVerifyKeyListener must return false.
(abstract) Connection.clearWarnings() After a call to this method, the method getWarnings returns null until a new warning is reported for this Connection object.
(abstract) ResultSet.getArray(columnLabel) @exception SQLException if the columnLabel is not valid; if a database access error occurs or this method is called on a closed result set @exception SQLFeatureNotSupportedException if the JDBC driver does not support this method
(abstract) ICompletionProposalExtension.apply(document, trigger, offset) This method assumes that @link #isValidFor(IDocument, int) returns true if called for offset.
(abstract) IContributionManager.insertBefore(id, action) Equivalent to insertBefore(id,new ActionContributionItem(action)).
interface Comparable It is strongly recommended (though not required) that natural orderings be consistent with equals.
Character.isUnicodeIdentifierStart(codePoint) A character may start a Unicode identifier if and only if one of the following conditions is true:
@link #isLetter(int) isLetter(codePoint) returns true @link #getType(int) getType(codePoint) returns LETTER_NUMBER.

2.7 Miscellaneous Subclassing Directive

abstract class ColumnViewerEditor ColumnViewer implementors have to subclass this class and implement the missing methods
class ObjectInputStream Subclasses of Objects that are not serializable can be serializable. In this case the non-serializable class must have a no-arg constructor to allow its fields to be initialized.
DialogCellEditor.createContents(cell) If you reimplement this method, you should also reimplement updateContents.
DialogCellEditor.updateContents(value) If you reimplement this method, you should also reimplement createContents.
Dialog.createButtonBar(parent) The returned control's layout data must be an instance of GridData.
abstract class Number Subclasses of Number must provide methods to convert the represented numeric value to byte, double, float, int, long, and short.
ObjectOutputStream.annotateProxyClass(cl) For a given subclass of ObjectOutputStream that overrides this method, the resolveProxyClass method in the corresponding subclass of ObjectInputStream must read any data or objects written by annotateProxyClass
ObjectOutputStream.annotateClass(cl) The resolveClass method in the corresponding subclass of ObjectInputStream must read and use any data or objects written by annotateClass
ObjectOutputStream.replaceObject(obj) When a subclass is replacing objects it must insure that either a complementary substitution must be made during deserialization or that the substituted object is compatible with every field where the reference will be stored.
interface Serializable A no-arg constructor must be accessible to the subclass that is serializable.
interface Externalizable These methods must explicitly coordinate with the supertype to save its state.
class ObjectOutputStream Classes that require special handling during the serialization and deserialization process must implement special methods with these exact signatures:
class ObjectOutputStream In this case the non-serializable class must have a no-arg constructor to allow its fields to be initialized.
abstract class ImageDescriptor Using this abstract class involves defining a concrete subclass and providing an implementation for the getImageData method.
abstract class AbstractReconciler It is subclass responsibility to specify how dirty regions are processed.
(abstract) AbstractInformationControl.createContent(parent) Subclasses may set a different layout.
DateFormat.calendar Subclasses should initialize this to a calendar appropriate for the locale associated with this DateFormat.
DateFormat.numberFormat Subclasses should initialize this to a number format appropriate for the locale associated with this DateFormat.
interface ContentHandlerFactory An implementation of this interface should map a MIME type into an instance of ContentHandler.
abstract class PolicySpi Subclass implementations of this abstract class must provide a public constructor that takes a Policy.Parameters object as an input parameter.
abstract class CertPathValidatorSpi All CertPathValidator implementations must include a class (the SPI class) that extends this class (CertPathValidatorSpi) and implements all of its methods.
abstract class CertPathBuilderSpi All CertPathBuilder implementations must include a class (the SPI class) that extends this class (CertPathBuilderSpi) and implements all of its methods.
abstract class Permission Subclasses should not provide methods that can change the state of a permission once it has been created.
TableViewer.refresh(element, updateLabels, reveal) Subclasses who can provide this feature can open this method for the public
StructuredViewer.getSelection() Subclasses do not typically override this method, but implement getSelectionFromWidget(List) instead.
StructuredViewer.setSelectionToWidget(selection, reveal) Subclasses do not typically override this method, but implement setSelectionToWidget(List, boolean) instead.
abstract class ColumnViewer Concrete subclasses of @link ColumnViewer should implement a matching concrete subclass of @link ViewerColumn.
TextCellEditor.dependsOnExternalFocusListener() Subclasses that hook their own focus listener should override this method and return false.
SecurityManager.checkAccess(g) If this method is overridden, the method that overrides it should additionally check to see if the calling thread has the RuntimePermission("modifyThreadGroup") permission, and if so, return silently.
class UnicastRemoteObject The stub class must extend @link RemoteStub and must have a public constructor that has one parameter, of type @link RemoteRef.
abstract class EventListenerProxy Subclasses must provide the storage and accessor methods for the additional arguments or parameters.
abstract class AbstractSet all of the methods and constructors in subclasses of this class must obey the additional constraints imposed by the Set interface
abstract class AbstractQueuedSynchronizer Subclasses must define the protected methods that change this state, and which define what that state means in terms of this object being acquired or released.
class Logger Therefore, any subclasses of Logger (unless they are implemented in conjunction with a new LogManager class) should take care to obtain a Logger instance from the LogManager class and should delegate operations such as "isLoggable" and "log(LogRecord)" to that instance.
InetAddress.getByAddress(addr) IPv4 address byte array must be 4 bytes long and IPv6 byte array must be 16 bytes long
Assert.isNotNull(object) Implementations are encouraged to make regular use of Assert.isNotNull to ensure that null parameters are detected as early as possible.
abstract class PKIXCertPathChecker Additional input parameters required for executing the check may be specified through constructors of concrete implementations of this class.
interface Collection Bags or multisets (unordered collections that may contain duplicate elements) should implement this interface directly.
class Observable The default implementation provided in the Observable class will notify Observers in the order in which they registered interest, but subclasses may change this order, use no guaranteed order, deliver notifications on separate threads, or may guarantee that their subclass follows this order, as they choose.
ObjectInputStream.readUnshared() ObjectInputStream subclasses which override this method can only be constructed in security contexts possessing the "enableSubclassImplementation" SerializablePermission; any attempt to instantiate such a subclass without this permission will cause a SecurityException to be thrown.
class MemoryHandler A subclass overrides the log method and scans each incoming LogRecord and calls push if a record matches some desired criteria.
interface Deque Most Deque implementations place no fixed limits on the number of elements they may contain, but this interface supports capacity-restricted deques as well as those with no fixed size limit.
class DecimalFormat Decimal formats are generally not synchronized. It is recommended to create separate format instances for each thread.
(abstract) ResultSet.rowUpdated() Note: Support for the rowUpdated method is optional with a result set concurrency of CONCUR_READ_ONLY
interface ITextHover Any implementer of this interface must be capable of operating in a non-UI thread.

3 State Directive

3.1 Method Call Sequence Directive

PipedReader.PipedReader() It must be @linkplain java.io.PipedWriter#connect(java.io.PipedReader) connected to a PipedWriter before being used.
ComparatorChain.ComparatorChain() You must add at least one Comparator before calling the compare(Object,Object) method
(abstract) ActivationSystem.registerGroup(desc) An activation group must be registered with the ActivationSystem before objects can be registered within that group
(abstract) ActivationGroup.activeObject(id, obj) The group must inform its ActivationMonitor that the object is active (via the monitor's activeObject method) if the group hasn't already done so.
ActivationGroup.inactiveObject(id) A remote object implementation must call Activatable's inactive method when that object deactivates (the object deems that it is no longer active)
ActivationGroup.inactiveObject(id) After removing the object from the RMI runtime, the group must inform its ActivationMonitor (via the monitor's inactiveObject method) that the remote object is not currently active so that the remote object will be re-activated by the activator upon a subsequent activation request.

ActivationGroup.createGroup(id, desc, incarnation) A group must first be registered with the ActivationSystem before it can be created via this method

(abstract) DGC.dirty(ids, sequenceNum, lease) The client must also make a dirty call to renew leases on remote references before such leases expire

(abstract) DGC.dirty(ids, sequenceNum, lease) When the client no longer has any references to a specific remote object, it must schedule a clean call for the object ID associated with the reference

interface DGC A failed dirty call must schedule a strong clean call so that the call's sequence number can be retained in order to detect future calls received out of order by the distributed garbage collector

ServerSocket.setReceiveBufferSize(size) The value actually set in the accepted socket must be determined by calling @link Socket#getReceiveBufferSize() after the socket is returned by @link #accept().

Inflater.setInput(b) Should be called whenever needsInput() returns true indicating that more input data is required.

Deflater.setInput(b) This should be called whenever needsInput() returns true indicating that more input data is required.

Inflater.inflate(b, off, len) A return value of 0 indicates that needsInput() or needsDictionary() should be called

Inflater.inflate(b) A return value of 0 indicates that needsInput() or needsDictionary() should be called

Inflater.setInput(b, off, len) Should be called whenever needsInput() returns true indicating that more input data is required

Inflater.setDictionary(b) Should be called when inflate() returns 0 and needsDictionary() returns true indicating that a preset dictionary is required.

Deflater.setInput(b, off, len) This should be called whenever needsInput() returns true indicating that more input data is required

ObjectOutputStream.close() This method must be called to release any resources associated with the stream.

(abstract) ObjectOutput.close() This method must be called to release any resources associated with the stream.

ObjectInputStream.close() Must be called to release any resources associated with the stream

(abstract) ObjectInput.close() Must be called to release any resources associated with the stream.

(abstract) ResultSet.getBinaryStream(columnIndex) All the data in the returned stream must be read prior to getting the value of any other column

Manifest.write(out) Attributes.Name.MANIFEST_VERSION must be set in MainAttributes prior to invoking this method.

(abstract) ResultSet.wasNull() Note that you must first call one of the getter methods on a column to try to read its value and then call the method wasNull to see if the value read was SQL NULL

DropTargetContext.rejectDrop() must be called during DropTargetListener.drop method invocation.

DropTargetContext.acceptDrop(dropOperation) must be called during DropTargetListener.drop method invocation.

class SQLRecoverableException At a minimum, the recovery operation must include closing the current connection and getting a new connection.

ClassLoader.definePackage(name, specTitle, specVersion, specVendor, implTitle, implVersion, implVendor, sealBase) Packages must be created before the class is defined

(abstract) ResultSet.moveToInsertRow() All of the columns in a result set must be given a value each time this method is called before calling insertRow.

(abstract) ResultSet.moveToInsertRow() An updater method must be called before a getter method can be called on a column value.

(abstract) PreparedStatement.execute() you must call getMoreResults to move to any subsequent result(s).

ErrorDialog.showDetailsArea() This method will only work when it is invoked after the control of the dialog has been set

PreferencePage.setPreferenceStore(store) f preferenceStore is set to null, getPreferenceStore will invoke doGetPreferenceStore the next time it is called

JFaceResources.setFontRegistry(registry) This method may only be called once; the call must occur before JFaceResources.getFontRegistry is invoked

Socket.setSoTimeout(timeout) The option must be enabled prior to entering the blocking operation to have effect.

ServerSocket.setSoTimeout(timeout) The option must be enabled prior to entering the blocking operation to have effect.

DatagramSocket.setSoTimeout(timeout) The option must be enabled prior to entering the blocking operation to have effect.

SocketOptions.SO_TIMEOUT The option must be set prior to entering a blocking operation to take effect

Socket.setReceiveBufferSize(size) For sockets accepted from a ServerSocket, this must be done by calling @link ServerSocket#setReceiveBufferSize(int) before the ServerSocket is bound to a local address.

Socket.setReceiveBufferSize(size) For client sockets, setReceiveBufferSize() must be called before connecting the socket to its remote peer.

HttpURLConnection.setFixedLengthStreamingMode(contentLength) This method must be called before the URLConnection is connected.

HttpURLConnection.setChunkedStreamingMode(chunklen) This method must be called before the URLConnection is connected.

ThreadPoolExecutor.allowCoreThreadTimeOut(value) To avoid continual thread replacement, the keep-alive time must be greater than zero when setting true.

Applet.Applet() applet should avoid calling methods in java.applet.Applet in the constructor.

Applet.start() Because isShowing returns false when the applet's start is first called, methods requiring isShowing to return true should be called from a ComponentListener.

CoolBarManager.refresh() This method should be called before requesting the order of the contribution items to ensure that the order is accurate.

AbstractAction.firePropertyChange(event) Only listeners registered at the time this method is called are notified.

AbstractAction.firePropertyChange(propertyName, oldValue, newValue) Only listeners registered at the time this method is called are notified.

TitleAreaDialog.setMessage(newMessage) This method should be called after the dialog has been opened as it updates the message label immediately

ProgressMonitorDialog.run(fork, cancelable, runnable) If fork is set to false, the runnable will run in the UI thread and it is the runnable's responsibility to call Display.readAndDispatch() to ensure UI responsiveness.

ImageAndMessageArea.ImageAndMessageArea(parent, style) Calls to setText(String text) and setImage(Image image) are required in order to fill the message area.

DialogPage.convertHeightInCharsToPixels(chars) This method may only be called after initializeDialogUnits has been called.

DialogPage.convertHorizontalDLUsToPixels(dlus) This method may only be called after initializeDialogUnits has been called.

DialogPage.convertVerticalDLUsToPixels(dlus) This method may only be called after initializeDialogUnits has been called.

DialogPage.convertWidthInCharsToPixels(chars) This method may only be called after initializeDialogUnits has been called.

DialogPage.getShell() This method may only be called after the page's control has been created.

DialogPage.initializeDialogUnits(testControl) This method must be called before any of the dialog unit based conversion methods are called.

DialogPage.setButtonLayoutData(button) The method initializeDialogUnits must be called once before calling this method for the first time.

Dialog.convertHeightInCharsToPixels(chars) This method may only be called after initializeDialogUnits has been called.

Dialog.convertHorizontalDLUsToPixels(dlus) This method may only be called after initializeDialogUnits has been called.

Dialog.convertVerticalDLUsToPixels(dlus) This method may only be called after initializeDialogUnits has been called.

Dialog.convertWidthInCharsToPixels(chars) This method may only be called after initializeDialogUnits has been called.

Character.toCodePoint(high, low) The caller must validate it using @link #isSurrogatePair(char, char) isSurrogatePair if necessary.

ThreadGroup.destroy() This thread group must be empty, indicating that all threads that had been in this thread group have since stopped.

Thread.setDaemon(on) This method must be called before the thread is started.

Character.charCount(codePoint) The caller must validate the character value using @link #isValidCodePoint(int) isValidCodePoint if necessary.

SecurityManager.checkRead(file, context) The context must be a security context returned by a previous call to getSecurityContext.

PipedWriter.PipedWriter() It must be connected to a piped reader, either by the receiver or the sender, before being used.

PipedReader.PipedReader(pipeSize) It must be @linkplain java.io.PipedWriter#connect(java.io.PipedReader) connected to a PipedWriter before being used

PipedOutputStream.PipedOutputStream() It must be connected to a piped input stream, either by the receiver or the sender, before being used.

PipedInputStream.PipedInputStream(pipeSize) It must be @linkplain java.io.PipedOutputStream#connect(java.io.PipedInputStream) connected to a PipedOutputStream before being used.

PersistentState.shouldPersist() Subclasses should check this method before loading or saving.

ParsePosition.setErrorIndex(ei) Formatters should set this before returning an error code from their parseObject method.

URLConnection.setDefaultRequestProperty(key, value) The instance specific setRequestProperty method should be used after an appropriate instance of URLConnection is obtained.

URLConnection.getDefaultRequestProperty(key) The instance specific getRequestProperty method should be used after an appropriate instance of URLConnection is obtained.

URLConnection.getPermission() In the first case, the permission should be obtained after the object has been obtained.

URLConnection.getPermission() In the second case, the permission should be obtained and tested before connecting.

(abstract) CookieHandler.get(uri, requestHeaders) HTTP protocol implementers should make sure that this method is called after all request headers related to choosing cookies are added, and before the request is sent.

Socket.setTrafficClass(tc) Applications should not assume that they can change the TOS field after the connection.

(abstract) RemoteRef.done(call) Done should only be called if the invoke returns successfully (non-exceptionally) to the stub.

PipedInputStream.PipedInputStream() It must be @linkplain java.io.PipedOutputStream#connect(java.io.PipedInputStream) connected to a PipedOutputStream before being used.

ClassLoader.setSigners(c, signers) This should be invoked after defining a class.

(abstract) CallableStatement.registerOutParameter(parameterIndex, sqlType, scale) All OUT parameters must be registered before a stored procedure is executed.

(abstract) CallableStatement.registerOutParameter(parameterName, sqlType, scale) All OUT parameters must be registered before a stored procedure is executed.

interface CallableStatement The type of all OUT parameters must be registered prior to executing the stored procedure; their values are retrieved after execution via the get methods provided here.

(abstract) CallableStatement.registerOutParameter(parameterIndex, sqlType) All OUT parameters must be registered before a stored procedure is executed.

(abstract) CallableStatement.registerOutParameter(parameterName, sqlType) All OUT parameters must be registered before a stored procedure is executed.

(abstract) Statement.execute(sql) The execute method executes an SQL statement and indicates the form of the first result. You must then use the methods getResultSet or getUpdateCount to retrieve the result, and getMoreResults to move to any subsequent result(s)

(abstract) SQLOutput.writeObject(x) The attributes must be read from an SQLInput input stream and written to an SQLOutput output stream in the same order in which they were listed in the SQL definition of the user-defined type.

(abstract) CallableStatement.registerOutParameter(parameterIndex, sqlType, typeName) All OUT parameters must be registered before a stored procedure is executed.

(abstract) CallableStatement.registerOutParameter(parameterName, sqlType, typeName) All OUT parameters must be registered before a stored procedure is executed.

interface SQLInput And as described in SQLData, calls to reader methods must be made in the order that their corresponding attributes appear in the SQL definition of the type.

(abstract) ResultSet.getAsciiStream(columnIndex) All the data in the returned stream must be read prior to getting the value of any other column.

(abstract) ResultSet.getUnicodeStream(columnIndex) All the data in the returned stream must be read prior to getting the value of any other column

(abstract) ResultSet.getAsciiStream(columnLabel) All the data in the returned stream must be read prior to getting the value of any other column.

(abstract) ResultSet.getUnicodeStream(columnLabel) All the data in the returned stream must be read prior to getting the value of any other column.

(abstract) ResultSet.getBinaryStream(columnLabel) All the data in the returned stream must be read prior to getting the value of any other column.

(abstract) Statement.execute(sql, autoGeneratedKeys) You must then use the methods getResultSet or getUpdateCount to retrieve the result, and getMoreResults to move to any subsequent result(s).

(abstract) Statement.execute(sql, columnIndexes) The execute method executes an SQL statement and indicates the form of the first result. You must then use the methods getResultSet or getUpdateCount to retrieve the result, and getMoreResults to move to any subsequent result(s).

(abstract) Statement.execute(sql, columnNames) The execute method executes an SQL statement and indicates the form of the first result. You must then use the methods getResultSet or getUpdateCount to retrieve the result, and getMoreResults to move to any subsequent result(s).

interface SQLXML All streams must be closed before execute() occurs or an SQLException will be thrown.

(abstract) KeyStore.Builder.getProtectionParameter(alias) The getKeyStore method must be invoked before this method may be called.

class PKIXParameters To instantiate a PKIXParameters object, an application must specify one or more most-trusted CAs as defined by the PKIX certification path validation algorithm.

class KeyStore Before a keystore can be accessed, it must be @link #load(java.io.InputStream, char[]) loaded.

(abstract) MessageDigestSpi.engineDigest() Once engineDigest has been called, the engine should be reset (see @link #engineReset() engineReset).

MessageDigestSpi.engineDigest(buf, offset, len) Once engineDigest has been called, the engine should be reset (see @link #engineReset() engineReset).

abstract class MessageDigest A MessageDigest object starts out initialized. The data is processed through it using the @link #update(byte) update methods. At any point @link #reset() reset can be called to reset the digest. Once all the data to be updated has been updated, one of the @link #digest() digest methods should be called to complete the hash computation.

IteratorChain.lastUsedIterator The "last used" Iterator is the Iterator upon which next() or hasNext() was most recently called used for the remove() operation only

LoopingIterator.remove() This method can only be called after at least one @link #next method call.

LoopingListIterator.remove() This method can only be called after at least one @link #next or @link #previous method call.

ObjectArrayIterator.setArray(array) This method may only be called once, otherwise an IllegalStateException will occur.

FilterIterator.remove() This method can only be called if next() was called, but not after hasNext(), because the hasNext() call changes the base iterator.

class PolicyQualifierInfo Applications that need to process policy qualifiers should explicitly set the policyQualifiersRejected flag to false (by calling the @link PKIXParameters#setPolicyQualifiersRejected PKIXParameters.setPolicyQualifiersRejected method) before validating a certification path.

ArrayListIterator.set(o) Note: @link ListIterator implementations that support add() and remove() only allow set() to be called once per call to next() or previous (see the @link ListIterator javadoc for more details).

(abstract) Statement.getResultSet() Retrieves the current result as a ResultSet object. This method should be called only once per result.

ObjectArrayListIterator.set(obj) @throws IllegalStateException if next() has not yet been called.

(abstract) Statement.getUpdateCount() This method should be called only once per result.

(abstract) CallableStatement.wasNull() Note that this method should be called only after calling a getter method; otherwise, there is no value to use in determining whether it is null or not.

interface CallableStatement For maximum portability, a call's ResultSet objects and update counts should be processed prior to getting the values of output parameters.

CopyOnWriteArraySet.toArray() this method must allocate a new array even if this set is backed by an array

(abstract) Set.toArray() In other words, this method must allocate a new array even if this set is backed by an array

(abstract) Collection.toArray() In other words, this method must allocate a new array even if this collection is backed by an array

abstract class ClassLoader Once it has downloaded the bytes that make up the class, it should use the method @link #defineClass defineClass to create a class instance.

interface IFindReplaceTarget Clients of a IFindReplaceTarget that also implements the IFindReplaceTargetExtension have to indicate the start of a find/replace session before using the target and to indicate the end of the session when the target is no longer used.

Dialog.initializeDialogUnits(control) This method must be called before any of the dialog unit based conversion methods are called.

LogRecord.getResourceBundle() This is the ResourceBundle that should be used to localize the message string before formatting it.

LogRecord.getResourceBundleName() This is the name for the ResourceBundle that should be used to localize the message string before formatting it.

Inflater.end() This method should be called when the decompressor is no longer being used, but will also be called automatically by the finalize() method.

Deflater.end() This method should be called when the compressor is no longer being used, but will also be called automatically by the finalize() method.

ThreadPoolExecutor.allowCoreThreadTimeOut(value) This method should in general be called before the pool is actively used.

Inflater.setDictionary(b, off, len) Should be called when inflate() returns 0 and needsDictionary() returns true indicating that a preset dictionary is required.

class Semaphore Before obtaining an item each thread must acquire a permit from the semaphore, guaranteeing that an item is available for use.

class LockSupport The park method may also return at any other time, for "no reason", so in general must be invoked within a loop that rechecks conditions upon return.

LoopingIterator.next() Use @link #hasNext to avoid this error.

class Date As of JDK 1.1, the Calendar class should be used to convert between dates and time fields and the DateFormat class should be used to format and parse date strings.

EntrySetMapIterator.remove() This method can be called once per call to next().

EntrySetMapIterator.remove() @throws IllegalStateException if next() has not yet been called

(abstract) MapIterator.remove() This method can be called once per call to next().

(abstract) MapIterator.remove() @throws IllegalStateException if next() has not yet been called

(abstract) MapIterator.remove() @throws IllegalStateException if remove() has already been called since the last call to next()

CursorableLinkedList.Cursor.remove() However, you can't call this method on the same iterator more than once without calling next() or previous().

ObjectArrayListIterator.set(obj) Note: @link ListIterator implementations that support add() and remove() only allow set() to be called once per call to next() or previous (see the @link ListIterator javadoc for more details).

class StaticBucketMap This means that you don't have to wrap instances of this class with @link java.util.Collections#synchronizedMap(Map); instances are already thread-safe.

class ServiceLoader Trusted system code should typically invoke the methods in this class, and the methods of the iterators which they return, from within a privileged security context.

class LogManager Thus level settings for child nodes in the tree should come after settings for their parents.

AbstractReferenceMap.purgeBeforeRead() Purges stale mappings from this map before read operations.

AbstractReferenceMap.purgeBeforeWrite() Purges stale mappings from this map before write operations.

interface ExecutorService An unused ExecutorService should be shut down to allow reclamation of its resources.

FilterIterator.remove() @throws IllegalStateException if hasNext() has already been called.

IteratorChain.IteratorChain() You will normally use @link #addIterator(Iterator) to add some iterators after using this constructor.

IteratorChain.IteratorChain(iterator) You will normally use @link #addIterator(Iterator) to add some more iterators after using this constructor.

SingletonListIterator.remove() @throws IllegalStateException if the next or previous method has not yet been called, or the remove method has already been called after the last call to next or previous.

SingletonIterator.remove() @throws IllegalStateException if the next method has not yet been called, or the remove method has already been called after the last call to the next method.

IteratorChain.remove() @throws IllegalStateException if the next method has not yet been called, or the remove method has already been called after the last call to the next method.

ObjectGraphIterator.remove() @throws IllegalStateException if the next method has not yet been called, or the remove method has already been called after the last call to the next method.

(abstract) DGC.dirty(ids, sequenceNum, lease)

ClassLoader.definePackage(name, specTitle, specVersion, specVendor, implTitle, implVersion, implVendor, sealBase) package names must be unique within a class loader and cannot be redefined or changed once created.

SecurityManager.checkPermission(perm, context) The context must be a security context returned by a previous call to getSecurityContext and the access control decision is based upon the configured security policy for that security context.

SecurityManager.checkRead(file, context) The context must be a security context returned by a previous call to getSecurityContext.

MultiKey.MultiKey(keys, makeClone) The keys should be immutable If they are not then they must not be changed after adding to the MultiKey.

Handler.setEncoding(encoding) The encoding should be set before any LogRecords are written to the Handler.

(abstract) ResultSet.next() When a call to the next method returns false, the cursor is positioned after the last row. Any invocation of a ResultSet method which requires a current row will result in a SQLException being thrown.

(abstract) ResultSet.previous() When a call to the previous method returns false, the cursor is positioned before the first row. Any invocation of a ResultSet method which requires a current row will result in a SQLException being thrown.

ActivationDesc.ActivationDesc(className, location, data) If an activatable object requires restart services, use one of the ActivationDesc constructors that takes a boolean parameter, restart.

ActivationDesc.ActivationDesc(groupID, className, location, data) If an activatable object requires restart services, use one of the ActivationDesc constructors that takes a boolean parameter, restart.

StreamHandler.setEncoding(encoding) The encoding should be set before any LogRecords are written to the Handler.

LogManager.removePropertyChangeListener(l) If the same listener instance has been added to the listener table through multiple invocations of addPropertyChangeListener, then an equivalent number of removePropertyChangeListener invocations are required to remove all instances of that listener from the listener table.

Deflater.setDictionary(b, off, len) When the data is later uncompressed with Inflater.inflate(), Inflater.getAdler() can be called in order to get the Adler-32 value of the dictionary required for decompression.

ObjectArrayListIterator.lastItemIndex This is set to -1 if neither method has yet been invoked.

FilterIterator.FilterIterator(iterator) Constructs a new FilterIterator that will not function until @link #setPredicate(Predicate) setPredicate is invoked.

FilterListIterator.FilterListIterator() Constructs a new FilterListIterator that will not function until @link #setListIterator(ListIterator) setListIterator and @link #setPredicate(Predicate) setPredicate are invoked.

FilterListIterator.FilterListIterator(iterator) Constructs a new FilterListIterator that will not function until @link #setPredicate(Predicate) setPredicate is invoked.

FilterListIterator.FilterListIterator(predicate) Constructs a new FilterListIterator that will not function until @link #setListIterator(ListIterator) setListIterator is invoked.

TransformIterator.TransformIterator() Constructs a new TransformIterator that will not function until the @link #setIterator(Iterator) setIterator and @link #setTransformer(Transformer) methods are invoked.

ArrayListIterator.lastItemIndex Holds the index of the last item returned by a call to next() or previous(). This is set to -1 if neither method has yet been invoked.

IteratorEnumeration.IteratorEnumeration()

Constructs a new IteratorEnumeration that will not function until @link #setIterator(Iterator) setIterator is invoked.

FilterIterator.FilterIterator() Constructs a new FilterIterator that will not function until @link #setIterator(Iterator) setIterator is invoked.

AbstractLinkedList.LinkedListIterator.getLastNodeReturned() @throws IllegalStateException If @link #next() or @link #previous() haven't been called, or if the node has been removed with @link #remove() or a new node added with @link #add(Object).

ObjectArrayListIterator.previousIndex() Gets the index of the item to be retrieved if @link #previous() is called.

ObjectArrayListIterator.ObjectArrayListIterator() Using this constructor, the iterator is equivalent to an empty iterator until @link #setArray is called to establish the array to iterate over.

SingletonListIterator.previousIndex() Returns the index of the element that would be returned by a subsequent call to previous.

LoopingListIterator.nextIndex() Returns the index of the element that would be returned by a subsequent call to @link #next.

LoopingListIterator.previousIndex() Returns the index of the element that would be returned by a subsequent call to @link #previous.

EnumerationIterator.EnumerationIterator() Constructs a new EnumerationIterator that will not function until @link #setEnumeration(Enumeration) is called.

(abstract) Preferences.addPreferenceChangeListener(pcl) Events are not generated when preferences are modified in descendants of this node; a caller desiring such events must register with each descendant.

Matcher.replaceFirst(replacement) If the matcher is to be used in further matching operations then it should first be reset.

interface Condition When waiting upon a Condition, a "spurious wakeup" is permitted to occur, in general, as a concession to the underlying platform semantics. This has little practical impact on most application programs as a Condition should always be waited upon in a loop, testing the state predicate that is being waited for.

Matcher.replaceAll(replacement) If the matcher is to be used in further matching operations then it should first be reset.

GridLayoutFactory.generateLayout(c) This must be called AFTER all of the child controls have been created and their layouts attached. This method will attach a layout to the given composite. If any new children are created after calling this method, their GridData must be created manually. The algorithm does not recurse into child composites. To generate all the layouts in a widget hierarchy, the method must be called bottom-up for each Composite.

(abstract) ListIterator.previous() This method may be called repeatedly to iterate through the list backwards, or intermixed with calls to next to go back and forth. (Note that alternating calls to next and previous will return the same element repeatedly.)

abstract class AbstractTreeViewer As of 3.2, AbstractTreeViewer supports multiple equal elements (each with a different parent chain) in the tree. This support requires that clients enable the element map by calling setUseHashLookup(true).

interface ILazyContentProvider As the ILazyContentProvider does not have API for determining the total item count any changes to the number of items for this object while require a call to #setItemCount on the viewer that uses it.

java/net The recommended usage is to use @link java.net.URI to identify resources, then convert it into a @link java.net.URL when it is time to access the resource. From that URL, you can either get the @link java.net.URLConnection for fine control, or get directly the InputStream.

Here is an example:

URI uri = new URI("http://java.sun.com/"); URL url = uri.toURL(); InputStream in = url.openStream();

ZipFile.finalize() Since the time when GC would invoke this method is undetermined, it is strongly recommended that applications invoke the close method as soon they have finished accessing this ZipFile.

(abstract) Connection.close() It is strongly recommended that an application explicitly commits or rolls back an active transaction prior to calling the close method.

AbstractTreeViewer.internalExpandToLevel(widget, level) Note that the default implementation of this method does not call setRedraw.

ErrorDialog.ErrorDialog(parentShell, dialogTitle, message, status, displayMask) Note that the dialog will have no visual representation (no widgets) until it is told to open.

AbstractTreeViewer.createChildren(widget) Note: the node is not visually expanded! You may have to call parent.setExpanded(true).

TrayDialog.TrayDialog(shell) Note that the window will have no visual representation (no widgets) until it is told to open.

InputDialog.InputDialog(parentShell, dialogTitle, dialogMessage, initialValue, validator) Note that the dialog will have no visual representation (no widgets) until it is told to open.

(abstract) ISourceViewer.configure(configuration) Prior to 3.0 this method can only be called once. Since 3.0 this method can be called again after a call to @link ISourceViewerExtension2#unconfigure().

LazySortedCollection.setCapacity(newSize) This can be used prior to a sequence of additions to avoid memory reallocation.

(abstract) IPainter.dispose() Prior to disposing, a painter should be deactivated.

CheckboxTreeViewer.setSubtreeChecked(element, state) Assumes that the element has been expanded before.

ByteArrayOutputStream.reset() The output stream can be used again, reusing the already allocated buffer space.

Signature.sign(outbuf, offset, len) This signature object is reset to its initial state (the state it was in after a call to one of the initSign methods) and can be reused to generate further signatures with the same private key.

(abstract) IMenuCreator.getMenu(parent) Returns the SWT menu, created as a pop up menu parented by the given control. In most cases, this menu can be created once, cached and reused when the pop-up/drop-down action occurs. If the menu must be dynamically created (i.e., each time it is popped up or dropped down), the old menu should be disposed of before replacing it with the new menu.

PixelConverter.PixelConverter(control) If the font of the specified control is changed after this PixelConverter is created, then the conversions from this instance will not produce the desired effect.

ImageDescriptor.createFromImage(img, theDevice) This method takes the Device that created the Image as an argument, allowing the original Image to be reused if the descriptor is asked for another Image on the same device.

(abstract) IMenuCreator.getMenu(parent) Returns an SWT menu created as a drop down menu parented by the given menu. In most cases, this menu can be created once, cached and reused when the pop-up/drop-down action occurs.

interface IPreferenceNode Note that all preference nodes must be dispose their resources. The node must dispose the page managed by this node, and any SWT resources allocated by this node (Images, Fonts, etc). However the node itself may be reused.

Calendar.set(year, month, date, hourOfDay, minute) Previous values of other fields are retained. If this is not desired, call @link #clear() first.

abstract class ImageDescriptor When the caller is done with an image obtained from createResource, they must call destroyResource() rather than disposing the Image directly. The result of createResource() can be safely cast to an Image.

(abstract) Driver.getPropertyInfo(url, info) Note that depending on the values the human has supplied so far, additional values may become necessary, so it may be necessary to iterate though several calls to the getPropertyInfo method.

Socket.setReceiveBufferSize(size) There are two cases to be aware of:

For sockets accepted from a ServerSocket, this must be done by calling @link ServerSocket#setReceiveBufferSize(int) before the ServerSocket is bound to a local address.

For client sockets, setReceiveBufferSize() must be called before connecting the socket to its remote peer.

FontRegistry.getItalic(symbolicName) This method should only be called from the UI thread. If you are not on the UI thread then wrap the call with a PlatformUI.getWorkbench().getDisplay().synchExec() in order to guarantee the correct result.

interface ITextViewerExtension2 Painters are informed about changes to the viewer content and state and can take the necessary action in responds to the notification.

(abstract) ResourceManager.create(descriptor) If the resource is intended to live for entire lifetime of the resource manager, a subsequent call to @link #destroy(DeviceResourceDescriptor) may be omitted and the resource will be cleaned up when the resource manager is disposed. This pattern is useful for short-lived @link LocalResourceManagers, but should never be used with the global resource manager since doing so effectively leaks the resource.

(abstract) Preferences.flush() Once this method returns successfully, it is safe to assume that all changes made in the subtree rooted at this node prior to the method invocation have become permanent.

PopupDialog.adjustBounds() Adjust the bounds of the popup as necessary prior to opening the dialog.

WizardDialog.run(fork, cancelable, runnable) If fork is set to false, the runnable will run in the UI thread and it is the runnable's responsibility to call Display.readAndDispatch() to ensure UI responsiveness.

class InformationPresenter Usually, clients instantiate this class and configure it before using it. The configuration must be consistent: This means the used @link org.eclipse.jface.text.IInformationControlCreator must create an information control expecting information in the same format the configured @link org.eclipse.jface.text.information.II use to encode the information they provide.

PreferenceNode.PreferenceNode(id, label, image, className) The preference node assumes (sole) responsibility for disposing of the image; this will happen when the node is disposed.

ImageRegistry.ImageRegistry() There must be an SWT Display created in the current thread before calling this method.

(abstract) IReconcilingStrategy.setDocument(document) This method will be called before any other method and can be called multiple times.

(abstract) IReconcileStep.setInputModel(inputModel) This method will be called before any other method and can be called multiple times.

MarginPainter.setMarginRulerColor(color) Must be called before paint is called the first time.

ApplicationWindow.addStatusLine() This method must be called before this window's shell is created.

Thread.interrupted() In other words, if this method were to be called twice in succession, the second call would return false (unless the current thread were interrupted again, after the first call had cleared its interrupted status and before the second call had examined it).

(abstract) IPredicateRule.evaluate(scanner, resume) If this rules relies on a text pattern comprising a opening and a closing character sequence this method can also be called when the scanner is positioned already between the opening and the closing sequence. In this case, resume must be set to true.

(abstract) Clob.free() The object is invalid once the free method is called.

StringReader.close() Once the stream has been closed, further read(), ready(), mark(), or reset() invocations will throw an IOException

JarURLConnection.getCertificates() This method can only be called once the connection has been completely verified by reading from the input stream until the end of the stream has been reached.

Socket.close() Once a socket has been closed, it is not available for further networking use (i.e. can't be reconnected or rebound). A new socket needs to be created.

(abstract) Writer.close() Once the stream has been closed, further write() or flush() invocations will cause an IOException to be thrown.

Deflater.end() Once this method is called, the behavior of the Deflater object is undefined.

FieldAssistColors.computeErrorFieldBackgroundRGB(control) lients should typically call this method once, create a color from the RGB provided, and dispose of the color when finished using it.

AbstractRulerColumn.dispose() Clients who created this column are responsible to call this method once the column is no longer used.

(abstract) IReconcileStep.setPreviousStep(step) Note: This method must be called at most once per reconcile step.

interface IWizardNode Once the client is done with a wizard node, its disposemethod must be called to free up the wizard; once disposes, the node should no longer be used.

(abstract) ISourceViewerExtension2.unconfigure() Unlike @link ISourceViewer#configure(SourceViewerConfiguration) this method can be called more than once without interleaving calls to @link ISourceViewer#configure(SourceViewerConfiguration).

(abstract) IAction.setId(id) It should be set when the action is created. It should not be modified once the action is part of an action contribution item.

(abstract) IWizardPage.setWizard(newWizard) Once established, a page's wizard cannot be changed to a different wizard.

AbstractInformationControlManager.getNextFallbackAnchor(anchor) Note: It is the callers responsibility to prevent an endless loop i.e. to test whether a given anchor has already been used once.

AbstractHyperlinkDetector.setContext(context) @throws IllegalStateException if this method is called more than once

(abstract) AbstractTableViewer.doGetColumn(index) If no TableColumns were created by the programmer, this method will throw ERROR_INVALID_RANGE despite the fact that a single column of data may be visible in the table.

FontRegistry.FontRegistry(location, loader) There must be an SWT Display created in the current thread before calling this method.

JFaceResources.getResources() May only be called from a UI thread.

SetUtils.transformedSortedSet(set, transformer) It is important not to use the original set after invoking this method, as it is a backdoor for adding untransformed objects.

SetUtils.transformedSet(set, transformer) It is important not to use the original set after invoking this method, as it is a backdoor for adding untransformed objects.

ListUtils.predicatedList(list, predicate) It is important not to use the original list after invoking this method, as it is a backdoor for adding invalid objects.

SetUtils.predicatedSet(set, predicate) It is important not to use the original set after invoking this method, as it is a backdoor for adding invalid objects

BagUtils.predicatedSortedBag(bag, predicate) It is important not to use the original bag after invoking this method, as it is a backdoor for adding invalid objects.

ListUtils.transformedList(list, transformer) It is important not to use the original list after invoking this method, as it is a backdoor for adding untransformed objects.

BagUtils.transformedSortedBag(bag, transformer) It is important not to use the original bag after invoking this method, as it is a backdoor for adding untransformed objects.

Window.setParentShell(newParentShell) Once the shell is created, it must be disposed (i.e., closed) before this method can be called.

(abstract) AbstractInformationControlManager.IInformationControlCloser.setSubjectControl(subject) Must be called before start. May again be called between start and stop.

ErrorDialog.showDetailsArea() Invoking the method before the content area has been set or after the dialog has been disposed will have no effect.

class ContentFormatter Usually, clients instantiate this class and configure it before using it.

FontRegistry.FontRegistry() There must be an SWT Display created in the current thread before calling this method.

ApplicationWindow.addToolBar(style) This method must be called before this window's shell is created.

interface IFormattingStrategyExtension Note that multiple calls to formatterStarts can be issued to a strategy before launching the formatting process with format.

(abstract) AbstractInformationControlManager.IInformationControlCloser.setInformationControl(control) Must be called before start. May again be called between start and stop.

(abstract) ILazyContentProvider.updateElement(index) TableViewer#replace(Object, int) is not called before returning from this method selections may have missing or stale elements. In this situation it is suggested that the selection is asked for again after he update.

StringFieldEditor.setValidateStrategy(value) Calling this method has no effect after createPartControl is called.

HyperlinkManager.setHyperlinkDetectors(hyperlinkDetectors) It is allowed to call this method after this hyperlink manger has been installed.

ByteArrayInputStream.close() Closing a ByteArrayInputStream has no effect. The methods in this class can be called after the stream has been closed without generating an IOException.

ServerSocket.setPerformancePreferences(connectionTime, latency, bandwidth) Invoking this method after this socket has been bound will have no effect.

(abstract) Clob.free() After free has been called, any attempt to invoke a method other than free will result in a SQLException being thrown.

(abstract) SQLXML.free() . After free has been called, any attempt to invoke a method other than free will result in a SQLException being thrown.

ObjectInputStream.registerValidation(obj, prio) Register an object to be validated before the graph is returned. While similar to resolveObject these validations are called after the entire graph has been reconstituted.

ScheduledThreadPoolExecutor.setExecuteExistingDelayedTasksAfterShutdownPolicy(value) In this case, these tasks will only terminate upon shutdownNow, or after setting the policy to false when already shutdown.

ScheduledThreadPoolExecutor.setExecuteExistingDelayedTasksAfterShutdownPolicy(value) @param value if true, execute after shutdown, else don't. @see #getExecuteExistingDelayedTasksAfterShutdownPolicy

class PrintStream this means that the flush method is automatically invoked after a byte array is written, one of the println methods is invoked, or a newline character or byte ('\n') is written.

ActivationGroup.createGroup(id, desc, incarnation) After the group is created, the ActivationSystem is informed that the group is active by calling the activeGroup method which returns the ActivationMonitor for the group.

class ByteArrayOutputStream The methods in this class can be called after the stream has been closed without generating an IOException.

Preferences.systemNodeForPackage(c) Invoking this method will result in the creation of the returned node and its ancestors if they do not already exist. If the returned node did not exist prior to this call, this node and any ancestors that were created by this call are not guaranteed to become permanent until the flush method is called on the returned node (or one of its ancestors or descendants).

(abstract) Preferences.node(pathName) If the returned node did not exist prior to this call, this node and any ancestors that were created by this call are not guaranteed to become permanent until the flush method is called on the returned node (or one of its ancestors or descendants).

URLConnection.setContentHandlerFactory(fac) It can be called at most once by an application.

interface SQLXML Reading and writing XML values to or from an SQLXML object can happen at most once. The conceptual states of readable and not readable determine if one of the reading APIs will return a value or throw an exception. The conceptual states of writable and not writable determine if one of the writing APIs will set a value or throw an exception.

The state moves from readable to not readable once free() or any of the reading APIs are called: getBinaryStream(), getCharacterStream(), getSource(), and getString(). Implementations may also change the state to not writable when this occurs.

The state moves from writable to not writeable once free() or any of the writing APIs are called: setBinaryStream(), setCharacterStream(), setResult(), and setString(). Implementations may also change the state to not readable when this occurs.

Runtime.addShutdownHook(hook) Once the shutdown sequence has begun it can be stopped only by invoking the @link #halt halt method, which forcibly terminates the virtual machine.

Once the shutdown sequence has begun it is impossible to register a new shutdown hook or de-register a previously-registered hook. Attempting either of these operations will cause an @link IllegalStateException to be thrown.

(abstract) Preferences.removeNode() Once a node has been removed, attempting any method other than @link #name(), @link #absolutePath(), @link #isUserNode(), @link #flush() or @link #node(String) nodeExists("") on the corresponding Preferences instance will fail with an IllegalStateException

Formatter.format(l, format, args) @throws FormatterClosedException If this formatter has been closed by invoking its @link #close() method

3.2 Non Call-based State Directive

DriverPropertyInfo.required The required field is true if a value must be supplied for this property during Driver.connect and false otherwise.

ThreadGroup.destroy() This thread group must be empty, indicating that all threads that had been in this thread group have since stopped.

File.delete() If this pathname denotes a directory, then the directory must be empty in order to be deleted.

(abstract) ObjectOutputStream.PutField.write(out) Write the data and fields to the specified ObjectOutput stream, which must be the same stream that produced this PutField object.

MultiKey.getKeys() The keys should be immutable If they are not then they must not be changed

ClassLoader.defineClass(name, b, off, len, protectionDomain) The first class defined in a package determines the exact set of certificates that all subsequent classes defined in that package must contain. The set of certificates for a class is obtained from the @link java.security.CodeSource CodeSource within the ProtectionDomain of the class. Any classes added to that package must contain the same set of certificates or a SecurityException will be thrown.

Object.wait() The current thread must own this object's monitor.

Object.wait(timeout) The current thread must own this object's monitor.

Object.wait(timeout, nanos) The current thread must own this object's monitor.

(abstract) ResultSet.insertRow() The cursor must be on the insert row when this method is called.

(abstract) Owner.addOwner(caller, owner) The caller principal must be an owner of the ACL in order to invoke this method.

(abstract) Owner.deleteOwner(caller, owner) The caller principal must be an owner of the ACL in order to invoke this method.

(abstract) Acl.setName(caller, name) @param caller the principal invoking this method. It must be an owner of this ACL.

(abstract) Acl.removeEntry(caller, entry) @param caller the principal invoking this method. It must be an owner of this ACL.

(abstract) Acl.addEntry(caller, entry) @param caller the principal invoking this method. It must be an owner of this ACL.

abstract class CertPath For each certificate cert in a certification path certPath, cert.getType().equals(certPath.getType()) must be true.

Identity.addCertificate(certificate) If the identity has a public key, the public key in the certificate must be the same, and if the identity does not have a public key, the identity's public key is set to be that specified in the certificate.

LoopingIterator.remove() This feature is only supported if the underlying collection's @link Collection#iterator iterator method returns an implementation that supports it.

LoopingListIterator.remove() This feature is only supported if the underlying list's @link List#iterator iterator method returns an implementation that supports it.

LoopingListIterator.set(obj) This feature is only supported if the underlying list's @link List#listIterator method returns an implementation that supports it.

FactoryUtils.constantFactory(constantToReturn) In general, only immutable objects should use the constant factory.

MultiKey.MultiKey(keys, makeClone) The keys should be immutable If they are not then they must not be changed after adding to the MultiKey. @

LoopingListIterator.add(obj) This feature is only supported if the underlying list's @link List#listIterator method returns an implementation that supports it

LoopingListIterator.add(obj) @throws UnsupportedOperationException if the add method is not supported by the iterator implementation of the underlying list

class DualTreeBidiMap The setValue() method on iterators will succeed only if the new value being set is not already in the bidimap.

(abstract) Connection.commit() This method should be used only when auto-commit mode has been disabled.

(abstract) Connection.rollback() This method should be used only when auto-commit mode has been disabled.

(abstract) Connection.rollback(savepoint) This method should be used only when auto-commit has been disabled.

(abstract) CallableStatement.registerOutParameter(parameterIndex, sqlType, scale) This version of registerOutParameter should be used when the parameter is of JDBC type NUMERIC or DECIMAL.

(abstract) CallableStatement.registerOutParameter(parameterName, sqlType, scale) This version of registerOutParameter should be used when the parameter is of JDBC type NUMERIC or DECIMAL.

Object.notifyAll() This method should only be called by a thread that is the owner of this object's monitor.

Object.wait() This method should only be called by a thread that is the owner of this object's monitor.

MessageDialogWithToggle.openInformation(parent, title, message, toggleMessage, toggleState, store, key) @return the dialog, after being closed by the user, which the client can only call getReturnCode() or getToggleState()

MessageDialogWithToggle.openError(parent, title, message, toggleMessage, toggleState, store, key) @return the dialog, after being closed by the user, which the client can only call getReturnCode() or getToggleState()

MessageDialogWithToggle.open(kind, parent, title, message, toggleMessage, toggleState, store, key, style) @return the dialog, after being closed by the user, which the client can only call getReturnCode() or getToggleState()

RMIClassLoader.getDefaultProviderInstance() If there is a security manager, then the calling context must have permission to connect to all of the URLs in the codebase URL path; otherwise, the current thread's context class loader will be used instead of the codebase loader.

Object.wait(timeout) This method should only be called by a thread that is the owner of this object's monitor.

Object.wait(timeout, nanos) This method should only be called by a thread that is the owner of this object's monitor.

Object.notify() This method should only be called by a thread that is the owner of this object's monitor.

class DefaultKeyValue Note that a DefaultKeyValue instance may not contain itself as a key or value.

MultiKey.hashCode() This value is computed once and then cached, so elements should not change their hash codes once created

class GrowthList Note that IndexOutOfBoundsException IS returned for invalid negative indices.

class GrowthList Take care when using this list with null values, as null is the value added when growing the list.

ComparatorChain.isLocked() ComparatorChains cannot be modified once they have performed a comparison.

FixedOrderComparator.isLocked() FixedOrderComparators cannot be modified once they have performed a comparison.

IteratorChain.isLocked() IteratorChains cannot be modified once they have executed a method from the Iterator interface.

EntrySetMapIterator.remove() @throws IllegalStateException if remove() has already been called since the last call to next()

class FixedOrderComparator Once compare has been called, the FixedOrderComparator is locked and attempts to modify it yield an UnsupportedOperationException.

MultiKey.MultiKey(keys, makeClone) The hashcode is calculated once here in this method. Therefore, changing the array passed in would not change the hashcode but would change the equals method, which is a bug.

CompositeMap.remove(key) the composited map does not not permit null keys (optional).

FixedOrderComparator.checkLocked() @throws UnsupportedOperationException if the comparator is locked

MapUtils.invertMap(map) This operation assumes that the inverse mapping is well defined.

(abstract) Predicate.evaluate(object) @param object the object to evaluate, should not be changed

CompositeMap.put(key, value) @throws UnsupportedOperationException if no MapMutator has been specified

FixedOrderComparator.add(obj) @throws UnsupportedOperationException if a comparison has already been made

Vector.insertElementAt(obj, index) The index must be a value greater than or equal to 0 and less than or equal to the current size of the vector.

PopupMenu.show(origin, x, y) The origin component must be contained within the component hierarchy of the popup menu's parent.

NumberFormat.setMaximumFractionDigits(newValue) maximumFractionDigits must be ¿= minimumFractionDigits.

Vector.removeElementAt(index) The index must be a value greater than or equal to 0 and less than the current size of the vector.
NumberFormat.setMinimumFractionDigits(newValue) minimumFractionDigits must be ¡= maximumFractionDigits.
DateFormat.setLenient(lenient) With strict parsing, inputs must match this object's format.
Vector.setElementAt(obj, index) The index must be a value greater than or equal to 0 and less than the current size of the vector.
ObjectArrayIterator.setArray(array) @throws IllegalStateException if the array was set in the constructor
CursorableLinkedList.cursor(fromIndex) @throws IndexOutOfBoundsException if the index is out of range (index ¡ 0 —— index ¿ size()).
interface Statement Therefore, if the reading of one ResultSet object is interleaved with the reading of another, each must have been generated by different Statement objects.
abstract class CertPath For each certificate cert in a certification path certPath, cert.getType().equals(certPath.getType()) must be true.
abstract class CertPath The certificate representing the @link TrustAnchor TrustAnchor should not be included in the certification path.
SingletonMap.put(key, value) @param key the key to set, must be the key of the map
abstract class LocaleServiceProvider The provider must be accessible from the same class loader that was initially queried to locate the configuration file; this is not necessarily the class loader that loaded the file.
AbstractPreferences.newNode This field should be true if this node did not exist in the backing store prior to the creation of this object.
abstract class BreakIterator You must use a different iterator for each unit boundary analysis you wish to perform.
PKIXBuilderParameters.setMaxPathLength(maxPathLength) A PKIX CertPathBuilder instance must not build paths longer than the length specified.
BigDecimal.round(mc) @throws ArithmeticException if the rounding mode is UNNECESSARY and the BigDecimal operation would require rounding.
BigDecimal.setScale(newScale, roundingMode) @throws ArithmeticException if roundingMode==UNNECESSARY and the specified scaling operation would require rounding.
BigDecimal.setScale(newScale, roundingMode) @throws ArithmeticException if roundingMode==ROUND_UNNECESSARY and the specified scaling operation would require rounding.
BigDecimal.setScale(newScale) @throws ArithmeticException if the specified scaling operation would require rounding.
LogManager.checkAccess() This requires LoggingPermission("control").
LogManager.addLogger(logger) The application should retain its own reference to the Logger object to avoid it being garbage collec
class ServiceLoader The provider must be accessible from the same class loader that was initially queried to locate the configuration file; note that this is not necessarily the class loader from which the file was actually loaded.
(abstract) IFindReplaceTarget.replaceSelection(text) This target must be editable.
ImageRegistry.put(key, image) Note that an image registry owns all of the image objects registered with it, and automatically disposes of them when the SWT Display is disposed. Because of this, clients must not register an image object that is managed by another object.
TextPresentation.mergeStyleRange(range) Merges the given range into this presentation. The range must be a subrange of the presentation's default range.
TextPresentation.replaceStyleRanges(ranges) Replaces the given ranges in this presentation. Each range must be a subrange of the presentation's default range. The ranges must be ordered by increasing offset and must not overlap (but may be adjacent).
(abstract) Revision.getHoverInfo() The hover information control which is used to display the information must be able process the given object.
TextPresentation.replaceStyleRange(range) Replaces the given range in this presentation. The range must be a subrange of the presentation's default range.
FontDescriptor.createFrom(font, originalDevice) @param originalDevice must be the same Device that was passed into the font's constructor when it was first created.
DelegatingDropAdapter.removeDropTargetListener(listener) Removes the given TransferDropTargetListener. Listeners should not be removed while a drag and drop operation is in progress.
ViewerCell.setStyleRanges(styleRanges) Requires @link StyledCellLabelProvider with owner draw enabled.
ViewerRow.setStyleRanges(columnIndex, styleRanges) Set the style ranges to be applied on the text label at the column index Note: Requires @link StyledCellLabelProvider with owner draw enabled.
ObjectOutputStream.writeUnshared(obj) ObjectOutputStream subclasses which override this method can only be constructed in security contexts possessing the "enableSubclassImplementation" SerializablePermission; any attempt to instantiate such a subclass without this permission will cause a SecurityException to be thrown.
class StatusLineLayoutData NOTE: Do not reuse StatusLineLayoutData objects. Every control in the status line must have a unique StatusLineLayoutData instance or null.
StreamHandler.close() @exception SecurityException if a security manager exists and if the caller does not have LoggingPermission("control").
MulticastSocket.joinGroup(mcastaddr, netIf) @exception SecurityException if a security manager exists and its checkMulticast method doesn't allow the join.
Handler.setErrorManager(em) @exception SecurityException if a security manager exists and if the caller does not have LoggingPermission("control").
ReentrantReadWriteLock.WriteLock.unlock() If the current thread is the holder of this lock then the hold count is decremented. If the hold count is now zero then the lock is released. If the current thread is not the holder of this lock then @link IllegalMonitorStateException is thrown. @throws IllegalMonitorStateException if the current thread does not hold this lock.
FontRegistry.defaultFont() This method should only be called from the UI thread. If you are not on the UI thread then wrap the call with a PlatformUI.getWorkbench().getDisplay().syn in order to guarantee the correct result.
(abstract) SortedMap.tailMap(fromKey) @throws IllegalArgumentException if this map itself has a restricted range, and fromKey lies outside the bounds of the range
abstract class Calendar However, a call to set(Calendar.DAY_OF_MONTH, 30) before the call to getTime() sets the date to September 30, 1999, since no recomputation occurs after set() itself.
File.setReadOnly() After invoking this method the file or directory is guaranteed not to change until it is either deleted or marked to allow write access.
PushbackReader.unread(cbuf, off, len) After this method returns, the next character to be read will have the value cbuf[off], the character after that will have the value cbuf[off+1], and so forth.
PushbackReader.unread(cbuf) After this method returns, the next character to be read will have the value cbuf[0], the character after that will have the value cbuf[1], and so forth.
TreeMap.clear() The map will be empty after this call returns.
CopyOnWriteArrayList.clear() The list will be empty after this call returns.
interface BlockingQueue However, the bulk Collection operations addAll, containsAll, retainAll and removeAll are not necessarily performed atomically unless specified otherwise in an implementation. So it is possible, for example, for addAll(c) to fail (throwing an exception) after adding only some of the elements in c.
SecureRandom.setSeed(seed) Thus, repeated calls are guaranteed never to reduce randomness.
StringReader.reset() Resets the stream to the most recent mark, or to the beginning of the string if it has never been marked.
ActivationGroup.inactiveObject(id) If the object does not call Activatable.inactive when it deactivates, the object will never be garbage collected since the group keeps strong references to the objects it creates.
RMIClassLoader.getDefaultProviderInstance() The table does not maintain strong references to its contained loaders, in order to allow them and their defined classes to be garbage collected when not otherwise reachable.
(abstract) Condition.awaitNanos(nanosTimeout) The current thread is assumed to hold the lock associated with this Condition when this method is called.
(abstract) Condition.await() The current thread is assumed to hold the lock associated with this Condition when this method is called.

4 Alternative Directive

4.1 Alternative Directive

ContentViewer.setInput(input) Subclassers are advised to override inputChanged rather than this method
OwnerDrawLabelProvider.initialize(viewer, column) Subclasses should either call the super implementation or, alternatively, @link #initialize(ColumnViewer, ViewerColumn, boolean).
ErrorDialog.createDialogArea(parent) Subclasses that require a different dialog area may either override this method, or call the super implementation and add controls to the created composite.
Canvas.update(g) applications that override this method should either call super.update(g) or incorporate the functionality described
Component.update(g) Subclasses of Component that override this method should either call super.update(g), or call paint(g) directly from their update method.
interface PrivilegedAction This interface is used only for computations that do not throw checked exceptions; computations that throw checked exceptions must use PrivilegedExceptionAction instead.
(abstract) PreparedStatement.execute() You must call either the method getResultSet or getUpdateCount to retrieve the result
KeyboardFocusManager.redispatchEvent(target, e) Client implementations of dispatchEvent and client-defined KeyEventDispatchers must call redispatchEvent(target, e) instead of target.dispatchEvent(e) to dispatch an event.
BigDecimal.divide(divisor, scale, roundingMode) The new @link #divide(BigDecimal, int, RoundingMode) method should be used in preference to this legacy method.
BigDecimal.divide(divisor, roundingMode) The new @link #divide(BigDecimal, RoundingMode) method should be used in preference to this legacy method.

class BigDecimal Using the integer fields in this class (such as @link #ROUND_HALF_UP) to represent rounding mode is largely obsolete; the enumeration values of the RoundingMode enum, (such as @link RoundingMode#HALF_UP) should be used instead.

BigDecimal.setScale(newScale, roundingMode) The new @link #setScale(int, RoundingMode) method should be used in preference to this legacy method.

BigDecimal.BigDecimal(val) When a double must be used as a source for a BigDecimal, note that this constructor provides an exact conversion; it does not give the same result as converting the double to a String using the @link Double#toString(double) method and then using the @link #BigDecimal(String) constructor. To get that result, use the static @link #valueOf(double) method.

abstract class AbstractKeyFormatter It is recommended that implementations of IKeyFormatter subclass from here, rather than implementing IKeyFormatter directly.

abstract class TrayDialog It is recommended to subclass this class instead of Dialog in all cases except where the dialog should never show a tray.

abstract class ContentHandler An application does not generally call the getContent method in this class directly. Instead, an application calls the getContent method in class URL or in URLConnection.

class ViewerSorter It is recommended to use ViewerComparator instead.

ToolBarContributionItem.setCurrentHeight(currentHeight) Update(SIZE) should be called to adjust the widget.

ToolBarContributionItem.setCurrentWidth(currentWidth) Update(SIZE) should be called to adjust the widget.

ControlContribution.fill(parent) Subclasses must implement createControl rather than overriding this method.

ControlContribution.fill(parent, index) Subclasses must implement createControl rather than overriding this method.

BindingManager.setBindings(bindings) This method clears the existing bindings, but does not trigger a recomputation (other method calls are required to do that).

MessageDialog.createDialogArea(parent) Subclasses should override createCustomArea to add contents below the message.

Dialog.createContents(parent) Overriding createDialogArea and createButtonBar are recommended rather than overriding this method.

interface IAction Clients should subclass the abstract base class Action to define concrete actions rather than implementing IAction from scratch.

Action.run() Subclasses should override this method if they do not need information from the triggering event, or override runWithEvent(Event) if they do.

PopupDialog.createTitleMenuArea(parent) Subclasses typically need not override this method, but instead should use the constructor parameters showDialogMenu and showPersistAction to indicate whether a menu should be shown, and createTitleControl to to customize the presentation of the title.

PopupDialog.getDefaultSize() Subclasses should override this method when an alternate default size is desired, rather than overriding @link #getInitialSize().

PopupDialog.getDefaultLocation(initialSize) Subclasses should override this method when an alternate default location is desired, rather than overriding @link #getInitialLocation(Point)

interface IInformationControl Clients can implement this interface and its extension interfaces, subclass @link AbstractInformationControl, or use the (text-based) default implementation @link DefaultInformationControl.

class DefaultAutoIndentStrategy @deprecated since 3.1 use @link org.eclipse.jface.text.DefaultIndentLineAutoEditStrategy instead

AbstractInformationControl.setInformation(information) Subclasses must either override this method or implement @link IInformationControlExtension2.

DatagramSocket.setReceiveBufferSize(size) Because SO_RCVBUF is a hint, applications that want to verify what size the buffers were set to should call @link #getReceiveBufferSize().

DatagramSocket.setSendBufferSize(size) As SO_SNDBUF is a hint, applications that want to verify what size the buffer is should call @link #getSendBufferSize().

Socket.setReceiveBufferSize(size) Because SO_RCVBUF is a hint, applications that want to verify what size the buffers were set to should call @link #getReceiveBufferSize().

class RMISecurityManager Therefore an RMI application should use the SecurityManager class or another application-specific SecurityManager implementation instead of this class.

Activatable.exportObject(obj, location, data, restart, port) Instead, an application should register an activation descriptor and export a remote object separately, so that exceptions can be handled properly.

Activatable.Activatable(location, data, restart, port) Instead, an application should register an activation descriptor and export a remote object separately, so that exceptions can be handled properly.

Activatable.Activatable(location, data, restart, port, csf, ssf) Instead, an application should register an activation descriptor and export a remote object separately, so that exceptions can be handled properly.

Activatable.exportObject(obj, location, data, restart, port, csf, ssf) Instead, an application should register an activation descriptor and export a remote object separately, so that exceptions can be handled properly.

File.deleteOnExit() Note: this method should not be used for file-locking, as the resulting protocol cannot be made to work reliably. The @link java.nio.channels.FileLock FileLock facility should be used instead.

Writer.lock A subclass should therefore use the object in this field rather than this or a synchronized method.

Reader.lock A subclass should therefore use the object in this field rather than this or a synchronized method.

File.createNewFile() Note: this method should not be used for file-locking, as the resulting protocol cannot be made to work reliably. The @link java.nio.channels.FileLock FileLock facility should be used instead.

ByteArrayOutputStream.toString() The @linkplain java.nio.charset.CharsetDecoder class should be used when more control over the decoding process is required.

class PrintStream The @link PrintWriter class should be used in situations that require writing characters rather than bytes.

ByteArrayOutputStream.toString(charsetName) The @link java.nio.charset.CharsetDecoder class should be used when more control over the decoding process is required.

class OutputStreamWriter . The @linkplain java.nio.charset.CharsetEncoder class should be used when more control over the encoding process is required.

class Console Security note: If an application needs to read a password or other secure data, it should use @link #readPassword() or @link #readPassword(String, Object...) and manually zero the returned character array after processing to minimize the lifetime of sensitive data in memory.

Window.getShellStyle() Subclassers should call setShellStyle to change this value, rather than overriding this method.

EditingSupport.initializeCellEditorValue(cellEditor, cell) Standard customers should not overwrite this method but @link #getValue(Object)

EditingSupport.saveCellEditorValue(cellEditor, cell) Standard customers should not overwrite this method but @link #setValue(Object, Object)

ColumnViewerToolTipSupport.createToolTipContentArea(event, parent) Instead of overwriting this method subclasses should overwrite @link #createViewerToolTipContentArea(Event, ViewerCell, Composite)

class InheritableThreadLocal Inheritable thread-local variables are used in preference to ordinary thread-local variables when the per-thread-attribute being maintained in the variable (e.g., User ID, Transaction ID) must be automatically transmitted to any child threads that are created.

Policy.getPermissions(codesource) Applications should solely rely on the implies method to perform policy checks.

Policy.getPermissions(codesource) If an application absolutely must call a getPermissions method, it should call getPermissions(ProtectionDomain).

class PKIXParameters An application can call @link #PKIXParameters(Set) PKIXParameters(Set), specifying a Set of TrustAnchor objects, each of which identify a most-trusted CA. Alternatively, an application can call @link #PKIXParameters(KeyStore) PKIXParameters(KeyStore), specifying a KeyStore instance containing trusted certificate entries, each of which will be considered as a most-trusted CA.

DateFormatSymbols.getZoneStrings() Use of this method is discouraged; use @link java.util.TimeZone#getDisplayName() TimeZone.getDisplayName() instead

CertPath.CertPath(type) This constructor is protected because most users should use a CertificateFactory to create CertPaths.

X509CertSelector.setSubject(subjectDN) Denigrated, use @linkplain #setSubject(X500Principal) or @linkplain #setSubject(byte[]) instead. This method should not be relied on as it can fail to match some certificates because of a loss of encoding information in the RFC 2253 String form of some distinguished names.

interface PrivilegedExceptionAction This interface is used only for computations that throw checked exceptions; computations that do not throw checked exceptions should use PrivilegedAction instead.

(abstract) X509Certificate.getSubjectDN() Denigrated, replaced by @linkplain #getSubjectX500Principal(). This method returns the subject as an implementation specific Principal object, which should not be relied upon by portable code.

X509CertSelector.getIssuerAsString() Denigrated, use @linkplain #getIssuer() or @linkplain #getIssuerAsBytes() instead. This method should not be relied on as it can fail to match some certificates because of a loss of encoding information in the RFC 2253 String form of some distinguished names.

X509CertSelector.getSubjectAsString() Denigrated, use @linkplain #getSubject() or @linkplain #getSubjectAsBytes() instead. This method should not be relied on as it can fail to match some certificates because of a loss of encoding information in the RFC 2253 String form of some distinguished names.

X509CertSelector.addSubjectAlternativeName(type, name) No standard string format is defined for otherNames, X.400 names, EDI party names, IPv6 address names, or any other type of names. They should be specified using the @link #addSubjectAlternativeName(int type, byte [] name) addSubjectAlternativeName(int type, byte [] name) method.

X509CertSelector.addPathToName(type, name) No standard string format is defined for otherNames, X.400 names, EDI party names, IPv6 address names, or any other type of names. They should be specified using the @link #addPathToName(int type, byte [] name) addPathToName(int type, byte [] name) method.

X509CertSelector.addPathToName(type, name) for distinguished names, use @linkplain #addPathToName(int, byte[]) instead. This method should not be relied on as it can fail to match some certificates because of a loss of encoding information in the RFC 2253 String form of some distinguished names.

(abstract) X509Certificate.getIssuerDN() Denigrated, replaced by @linkplain #getIssuerX500Principal(). This method returns the issuer as an implementation specific Principal object, which should not be relied upon by portable code.

X509CRLSelector.addIssuerName(name) Denigrated, use @linkplain #addIssuer(X500Principal) or @linkplain #addIssuerName(byte[]) instead. This method should not be relied on as it can fail to match some CRLs because of a loss of encoding information in the RFC 2253 String form of some distinguished names.

(abstract) X509CRL.getIssuerDN() Denigrated, replaced by @linkplain #getIssuerX500Principal(). This method returns the issuer as an implementation specific Principal object, which should not be relied upon by portable code.

abstract class CertStoreSpi In general, instances of this class should only be accessed through the CertStore class.

DriverManager.getLogWriter() The getLogWriter and setLogWriter methods should be used instead of the get/setlogStream methods, which are deprecated.

DatabaseMetaData.sqlStateSQL99 This constant remains only for compatibility reasons. Developers should use the constant sqlStateSQL instead.

interface PreparedStatement If arbitrary parameter type conversions are required, the method setObject should be used with a target SQL type.

(abstract) PreparedStatement.setObject(parameterIndex, x) Not all databases allow for a non-typed Null to be sent to the backend. For maximum portability, the setNull or the setObject(int parameterIndex, Object x, int sqlType) method should be used instead of setObject(int parameterIndex, Object x).

(abstract) CallableStatement.setObject(parameterName, x) Not all databases allow for a non-typed Null to be sent to the backend. For maximum portability, the setNull or the setObject(String parameterName, Object x, int sqlType) method should be used instead of setObject(String parameterName, Object x).

interface Connection When configuring a Connection, JDBC applications should use the appropritate Connection method such as setAutoCommit or setTransactionIsolation. Applications should not invoke SQL commands directly to change the connection's configuration when there is a JDBC method available.

interface IConcurrentModel Clients should subclass AbstractConcurrentModel instead.

StructuredViewer.getSelection() Subclasses do not typically override this method, but implement getSelectionFromWidget(List) instead.

StructuredViewer.getItem(x, y) Viewers who are not subclasses of @link ColumnViewer should consider using a widget relative implementation like @link ColumnViewer#getItemAt(org.eclipse.swt.graphics.Point).

StructuredViewer.setSelectionToWidget(selection, reveal) Subclasses do not typically override this method, but implement setSelectionToWidget(List, boolean) instead.

String.getBytes() The @link java.nio.charset.CharsetEncoder class should be used when more control over the encoding process is required.

Short.valueOf(s) If a new Short instance is not required, this method should generally be used in preference to the constructor @link #Short(short), as this method is likely to yield significantly better space and time performance by caching frequently requested values.

Long.valueOf(l) If a new Long instance is not required, this method should generally be used in preference to the constructor @link #Long(long), as this method is likely to yield significantly better space and time performance by caching frequently requested values.

Integer.valueOf(i) If a new Integer instance is not required, this method should generally be used in preference to the constructor @link #Integer(int), as this method is likely to yield significantly better space and time performance by caching frequently requested values.

Float.valueOf(f) If a new Float instance is not required, this method should generally be used in preference to the constructor @link #Float(float), as this method is likely to yield significantly better space and time performance by caching frequently requested values.

Enum.name() Most programmers should use the @link #toString method in preference to this one, as the toString method may return a more user-friendly name.

Double.valueOf(d) If a new Double instance is not required, this method should generally be used in preference to the constructor @link #Double(double), as this method is likely to yield significantly better space and time performance by caching frequently requested values.

Character.valueOf(c) If a new Character instance is not required, this method should generally be used in preference to the constructor @link #Character(char), as this method is likely to yield significantly better space and time performance by caching frequently requested values.

Byte.valueOf(b) If a new Byte instance is not required, this method should generally be used in preference to the constructor @link #Byte(byte), as this method is likely to yield significantly better space and time performance by caching frequently requested values.

interface Runnable In most cases, the Runnable interface should be used if you are only planning to override the run() method and no other Thread methods.

String.getBytes(charsetName) The @link java.nio.charset.CharsetEncoder class should be used when more control over the encoding process is required.

String.getBytes(charset) The @link java.nio.charset.CharsetEncoder class should be used when more control over the encoding process is required.

String.String(bytes, charset) The @link java.nio.charset.CharsetDecoder class should be used when more control over the decoding process is required.

String.String(bytes) The @link java.nio.charset.CharsetDecoder class should be used when more control over the decoding process is required.

Boolean.valueOf(b) If a new Boolean instance is not required, this method should generally be used in preference to the constructor @link #Boolean(boolean), as this method is likely to yield significantly better space and time performance.

ThreadGroup.enumerate(list) An application might use the activeCount method to get an estimate of how big the array should be,

ThreadGroup.enumerate(list, recurse) An application might use the activeCount method to get an estimate of how big the array should be

interface ITextViewer In order to provide backward compatibility for clients of ITextViewer, extension interfaces are used as a means of evolution. The following extension interfaces exist:

* @link org.eclipse.jface.text.ITextViewerExtension since version 2.0 replacing the event consumer mechanism and introducing the concept of rewrite targets and means to manage the viewer's redraw behavior * @link org.eclipse.jface.text.ITextViewerExtension2since version 2.1 adding a way to invalidate a viewer's presentation and setters for hovers. * @link org.eclipse.jface.text.ITextViewerExtension3 since version 2.1 which itself was replaced by @link org.eclipse.jface.text.ITextViewerExtension5 in version 3.0 * @link org.eclipse.jface.text.ITextViewerExtension4 since version 3.0 introducing focus handling for widget token keepers and the concept of text presentation listeners. * @link org.eclipse.jface.text.ITextViewerExtension5 since version 3.0 extending the visible region concept with explicit handling and conversion of widget and model coordinates. * @link org.eclipse.jface.text.ITextViewerExtension6 since version 3.1 extending the text viewer with the ability to detect hyperlinks and access the undo manager. * @link org.eclipse.jface.text.ITextViewerExtension7 since version 3.3 extending the text viewer with the ability to install tabs to spaces conversion. * @link org.eclipse.jface.text.ITextViewerExtension8 since version 3.4 extending the text viewer with the ability to print and rich hover support.

interface ITextViewer Clients may use the standard implementation @link org.eclipse.jface.text.TextViewer.

ThreadGroup.enumerate(list) An application might use the activeGroupCount method to get an estimate of how big the array should be, however if the array is too short to hold all the thread groups, the extra thread groups are silently ignored.

ThreadGroup.enumerate(list, recurse) An application might use the activeGroupCount method to get an estimate of how big the array should be, however if the array is too short to hold all the thread groups, the extra thread groups are silently ignored.

String.String(bytes, offset, length, charsetName) The @link java.nio.charset.CharsetDecoder class should be used when more control over the decoding process is required.

String.String(bytes, offset, length, charset) The @link java.nio.charset.CharsetDecoder class should be used when more control over the decoding process is required.

String.String(bytes, charsetName) The @link java.nio.charset.CharsetDecoder class should be used when more control over the decoding process is required.

String.String(bytes, offset, length) The @link java.nio.charset.CharsetDecoder class should be used when more control over the decoding process is required.

TextViewer.deleteText() @deprecated use StyledText.invokeAction instead

TemplateProposal.apply(document) This method is no longer called by the framework and clients should overwrite @link #apply(ITextViewer, char, int, int) instead

TemplateProposal.apply(document, trigger, offset) This method is no longer called by the framework and clients should overwrite @link #apply(ITextViewer, char, int, int) instead

PopupDialog.createContents(parent) Overriding createDialogArea and (optionally) createTitleMenuArea and createTitleMenuArea are recommended rather than overriding this method.

(abstract) AbstractInformationControl.createContent(parent) Implementors must either use the dialog font or override @link #computeSizeConstraints(int, int).

FocusCellHighlighter.focusCellChanged(newCell, oldCell) Subclasses should override this method rather than @link #focusCellChanged(ViewerCell)

Class.getModifiers() . The modifiers consist of the Java Virtual Machine's constants for public, protected, private, final, static, abstract and interface; they should be decoded using the methods of class Modifier.

Character.toLowerCase(ch) In general, @link java.lang.String#toLowerCase() should be used to map characters to lowercase.

Character.toLowerCase(codePoint) In general, @link java.lang.String#toLowerCase() should be used to map characters to lowercase.

Character.toUpperCase(ch) In general, @link java.lang.String#toUpperCase() should be used to map characters to uppercase.

Character.toUpperCase(codePoint)

(abstract) ThreadMXBean.findMonitorDeadlockedThreads() To find deadlocks involving both object monitors and ownable synchronizers, the @link #findDeadlockedThreads findDeadlockedThreads method should be used.

Class.getFields() User code should use the methods of class Array to manipulate arrays.

Character.toUpperCase(codePoint) In general, @link java.lang.String#toUpperCase() should be used to map characters to uppercase.

(abstract) Instrumentation.redefineClasses(definitions) Where the existing class file bytes are to be transformed (for example in bytecode instrumentation) @link #retransformClasses retransformClasses should be used.

class StringBuffer . The StringBuilder class should generally be used in preference to this one, as it supports all of the same operations but it is faster, as it performs no synchronization.

class SecurityManager Sometimes a security check that should be made within a given context will actually need to be done from within a different context (for example, from within a worker thread). The @link SecurityManager#getSecurityContext getSecurityContext method and the @link SecurityManager#checkPermission(java.security.Permission, java.lang.Object) checkPermission method that includes a context argument are provided for this situation.

ContentAssistant2.enableColoredLabels(isEnabled) Completion proposals can implement @link ICompletionProposalExtension6 to provide colored proposal labels.

ContentAssistant.enableColoredLabels(isEnabled) Completion proposals can implement @link ICompletionProposalExtension6 to provide colored proposal labels.

interface IContentFormatter By default, clients use ContentFormatter or MultiPassContentFormatter as the standard implementers of this interface.

class ListenerList Please use @link org.eclipse.core.runtime.ListenerList instead.

class ListenerList Also note that the equivalent of @link #ListenerList() is actually @link org.eclipse.core.runtime.ListenerList#ListenerList(int) with @link org.eclipse.core.runtime.Lis as the argument.

class TreeViewer Content providers for tree viewers must implement either the @link ITreeContentProvider interface, (as of 3.2) the @link ILazyTreeContentProvider interface, or (as of 3.3) the @link ILazyTreePathContentProvider.

class TableViewer Label providers for table viewers must implement either the ITableLabelProvider or the ILabelProvider interface (see TableViewer.setLabelProvider for more details).

class TableTreeViewer Label providers for table tree viewers must implement either the ITableLabelProvider or the ILabelProvider interface (see TableTreeViewer.setLabelProvider for more details).

ObjectArrayListIterator.set(obj) Since this implementation does not support add() or remove(), set() may be called as often as desired.

ArrayListIterator.set(o) Since this implementation does not support add() or remove(), set() may be called as often as desired.

class SetUniqueList The @link org.apache.commons.collections.set.ListOrderedSet ListOrderedSet class provides an alternative approach, by wrapping an existing Set and retaining insertion order in the iterator.

ListOrderedMap.asList() An alternative to this method is to use the better named @link #keyList() or @link #keySet().

class HashedMap A Map implementation that is a general purpose alternative to HashMap.

This implementation improves on the JDK1.4 HashMap by adding the @link org.apache.commons.collections.MapIterator MapIterator functionality and many methods for subclassing.

LinkedMap.asList() An alternative to this method is to use @link #keySet().

MapUtils.putAll(map, array) This method is an alternative to the @link java.util.Map#putAll(java.util.Map) method and constructors. It allows you to build a map from an object array of various possible styles.

class Flat3Map This map is designed for performance and can outstrip HashMap. It also has good garbage collection characteristics.

* Optimised for operation at size 3 or less. * Still works well once size 3 exceeded. * Gets at size 3 or less are about 0-10% faster than HashMap, * Puts at size 3 or less are over 4 times faster than HashMap. * Performance 5% slower than HashMap once size 3 exceeded once.

class FixedSizeMap The simplest approach is to wrap this map using @link java.util.Collections#synchronizedMap(Map).

class PredicatedSortedMap Note that PredicatedSortedMap is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization. The simplest approach is to wrap this map using @link java.util.Collections#synchronizedSortedMap.

class TransformedMap Note that TransformedMap is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization. The simplest approach is to wrap this map using @link java.util.Collections#synchronizedMap(Map).

class LazyMap Note that LazyMap is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization. The simplest approach is to wrap this map using @link java.util.Collections#synchronizedMap(Map).

class FixedSizeSortedMap Note that FixedSizeSortedMap is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization. The simplest approach is to wrap this map using @link java.util.Collections#synchronizedSortedMap.

class ReferenceIdentityMap Note that ReferenceIdentityMap is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization. The simplest approach is to wrap this map using @link java.util.Collections#synchronizedMap.

class TransformedSortedMap Note that TransformedSortedMap is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization. The simplest approach is to wrap this map using @link java.util.Collections#synchronizedSortedMap.

class LazySortedMap Note that LazySortedMap is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization. The simplest approach is to wrap this map using @link java.util.Collections#synchronizedSortedMap.

class ReferenceMap Note that ReferenceMap is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization. The simplest approach is to wrap this map using @link java.util.Collections#synchronizedMap.

class HashedMap Note that HashedMap is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization. The simplest approach is to wrap this map using @link java.util.Collections#synchronizedMap(Map).

class CaseInsensitiveMap Note that CaseInsensitiveMap is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization. The simplest approach is to wrap this map using @link java.util.Collections#synchronizedMap(Map).

class IdentityMap Note that IdentityMap is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization. The simplest approach is to wrap this map using @link java.util.Collections#synchronizedMap(Map).

class ListOrderedMap Note that ListOrderedMap is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization. The simplest approach is to wrap this map using @link java.util.Collections#synchronizedMap(Map).

class Flat3Map Note that Flat3Map is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization. The simplest approach is to wrap this map using @link java.util.Collections#synchronizedMap(Map).

class LinkedMap Note that LinkedMap is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization. The simplest approach is to wrap this map using @link java.util.Collections#synchronizedMap(Map).

class PredicatedMap Note that PredicatedMap is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization. The simplest approach is to wrap this map using @link java.util.Collections#synchronizedMap(Map).

class CompositeMap Note that CompositeMap is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization. The simplest approach is to wrap this map using @link java.util.Collections#synchronizedMap(Map).

class LRUMap Note that LRUMap is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization. The simplest approach is to wrap this map using @link java.util.Collections#synchronizedMap(Map).

class TransformedMap Note that TransformedMap is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization. The simplest approach is to wrap this map using @link java.util.Collections#synchronizedMap(Map).

class Flat3Map slow

class TreeList This list implementation utilises a tree structure internally to ensure that all insertions and removals are O(log n). This provides much faster performance than both an ArrayList and a LinkedList where elements are inserted and removed repeatedly from anywhere in the list.

class TreeList ArrayList is a good general purpose list implementation. It is faster than TreeList for most operations except inserting and removing in the middle of the list. ArrayList also uses less memory as TreeList uses one object per entry.

class ArrayIterator If you know that you have an object array, the @link org.apache.commons.collections.iterators.ObjectArrayIterator ObjectArrayIterator class is a better choice, as it will perform better.

class ArrayListIterator If you know that you have an object array, the @link ObjectArrayListIterator class is a better choice, as it will perform better.

class IteratorChain Under many circumstances, linking Iterators together in this manner is more efficient (and convenient) than reading out the contents of each Iterator into a List and creating a new Iterator.

class ObjectGraphIterator Under many circumstances, linking Iterators together in this manner is more efficient (and convenient) than using nested for loops to extract a list.

interface MapIterator This iterator is a special version designed for maps. It can be more efficient to use this rather than an entry set iterator where the option is available, and it is certainly more convenient.

(abstract) IterableGet.mapIterator() A map iterator is an efficient way of iterating over maps. There is no need to access the entry set or use Map Entry objects.

interface IterableMap A map iterator is an efficient way of iterating over maps. There is no need to access the entry set or use Map Entry objects.

IteratorUtils.objectGraphIterator(root, transformer) Under many circumstances, linking Iterators together in this manner is more efficient (and convenient) than using nested for loops to extract a list.

class Locale Because a Locale object is just an identifier for a region, no validity check is performed when you construct a Locale. If you want to see whether particular resources are available for the Locale you construct, you must query those resources.

class BoundedFifoBuffer The following can be used to provide synchronized access to your BoundedFifoBuffer:

Buffer fifo = BufferUtils.synchronizedBuffer(new BoundedFifoBuffer());

class CircularFifoBuffer The following can be used to provide synchronized access to your CircularFifoBuffer:

Buffer fifo = BufferUtils.synchronizedBuffer(new CircularFifoBuffer());

MapUtils.safeAddToMap(map, key, value) Note that this method can be used to circumvent the map's value type at runtime.

class MultiKeyMap Note that IdentityMap and ReferenceIdentityMap are unsuitable for use as the key comparison would work on the whole MultiKey, not the elements within.

class PriorityBuffer Use @link org.apache.commons.collections.BufferUtils#synchronizedBuffer(Buffer) or @link org.apache.commons.collections.buffer.SynchronizedBuffer#decorat to provide synchronized access to a PriorityBuffer:

class ReferenceIdentityMap . You can use @link java.util.Collections#synchronizedMap to provide synchronized access to a ReferenceIdentityMap.

IteratorUtils.emptyIterator() WARNING: This method is binary incompatible with Commons Collections 2.1 and 2.1.1. Use EmptyIterator.INSTANCE for compatability with Commons Collections 2.1.1.

IteratorUtils.emptyListIterator() WARNING: This method is binary incompatible with Commons Collections 2.1 and 2.1.1. Use EmptyListIterator.INSTANCE for compatability with Commons Collections 2.1.1.

IteratorUtils.singletonIterator(object) WARNING: This method is binary incompatible with Commons Collections 2.1 and 2.1.1. Use new SingletonIterator(object) for compatability.

IteratorUtils.arrayIterator(array) WARNING: This method is binary incompatible with Commons Collections 2.1 and 2.1.1. Use new ArrayIterator(array) for compatability.

IteratorUtils.arrayIterator(array, start) WARNING: This method is binary incompatible with Commons Collections 2.1 and 2.1.1. Use new ArrayIterator(array,start) for compatability.

IteratorUtils.arrayIterator(array, start, end) WARNING: This method is binary incompatible with Commons Collections 2.1 and 2.1.1. Use new ArrayIterator(array,start,end) for compatability.

IteratorUtils.EMPTY_ITERATOR WARNING: This constant is binary incompatible with Commons Collections 2.1 and 2.1.1. Use EmptyIterator.INSTANCE for compatability with Commons Collections 2.1.1.

IteratorUtils.EMPTY_LIST_ITERATOR WARNING: This constant is binary incompatible with Commons Collections 2.1 and 2.1.1. Use EmptyListIterator.INSTANCE for compatability with Commons Collections 2.1.1.

CursorableLinkedList.Cursor.close() In previous versions of this class, it was mandatory to close all cursor objects to avoid memory leaks. It is no longer necessary to call this close method; an instance of this class can now be treated exactly like a normal iterator.

class AbstractLinkedMap.LinkEntry If you subclass AbstractLinkedMap but not LinkEntry then you will not be able to access the protected fields. The entryXxx() methods on AbstractLinkedMap exist to provide the necessary access.

class AbstractHashedMap.HashEntry If you subclass AbstractHashedMap but not HashEntry then you will not be able to access the protected fields. The entryXxx() methods on AbstractHashedMap exist to provide the necessary access.

class MultiValueMap In addition, this implementation allows the type of collection used for the values to be controlled. By default, an ArrayList is used, however a Class to instantiate may be specified, or a factory that returns a Collection instance.

ArrayIterator.setArray(array) Although this can be used to reset iteration, the @link #reset() method is a more effective choice.

(abstract) MultiMap.remove(key, item) If the last value for a key is removed, implementations typically return null from a subsequant get(Object), however they may choose to return an empty collection.

(abstract) MultiMap.get(key) Implementations typically return null if no values have been mapped to the key, however the implementation may choose to return an empty collection.

(abstract) MultiMap.get(key) @return the Collection of values, implementations should return null for no mapping, but may return an empty collection

(abstract) MultiMap.remove(key) Implementations typically return null from a subsequant get(Object), however they may choose to return an empty collection.

(abstract) MultiMap.remove(key) @return the Collection of values removed, implementations should return null for no mapping found, but may return an empty collection

abstract class AbstractSequentialList For random access data (such as an array), AbstractList should be used in preference to this class.

abstract class EnumSet Like most collection implementations, EnumSet is not synchronized. If multiple threads access an enum set concurrently, and at least one of the threads modifies the set, it should be synchronized externally. This is typically accomplished by synchronizing on some object that naturally encapsulates the enum set. If no such object exists, the set should be ”wrapped” using the @link Collections#synchronizedSet method.

class TreeSet If multiple threads access a tree set concurrently, and at least one of the threads modifies the set, it must be synchronized externally. This is typically accomplished by synchronizing on some object that naturally encapsulates the set. If no such object exists, the set should be ”wrapped” using the @link Collections#synchronizedSortedSet Collections.synchronizedSortedSet method.

class BoundedBuffer Note: This class should only be used if you need to add bounded behaviour to another buffer. If you just want a bounded buffer then you should use @link BoundedFifoBuffer or @link CircularFifoBuffer

interface IPresentationReconciler The following extension interface exists:

@link org.eclipse.jface.text.presentation.IPresentationReconcilerExtension since version 3.0 adding support for documents with multiple partitionings.

interface IVerticalRulerInfo The following extension interfaces exist:

@link org.eclipse.jface.text.source.IVerticalRulerInfoExtension since version 3.0 allowing custom annotation hovers and specific annotation models.

interface IUndoManager The following extension interfaces exist:

@link org.eclipse.jface.text.IUndoManagerExtension since version 3.1 introducing access to the undo context.

interface IRevisionRulerColumn The following extension interfaces exist:

@link IRevisionRulerColumnExtension since version 3.3 allowing to register a selection listener on revisions and a configurable rendering mode.

interface IVerticalRuler The following extension interfaces exist:

@link org.eclipse.jface.text.source.IVerticalRulerExtension since version 2.0 introducing setters for font and mouse button activity location.

interface ISourceViewer The following extension interfaces exist:

@link org.eclipse.jface.text.source.ISourceViewerExtension since version 2.1 introducing the concept of an annotation overview. @link org.eclipse.jface.text.source.ISourceViewerE since version 3.0 allowing source viewers to roll back a previously performed configuration and allows access to the viewer’s visual annotation model. @link org.eclipse.jface.text.source.ISourceViewerExtension3 since version 3.2 introducing the concept of a quick assist assistant and providing access to the quick assist invocation context as well as the current annotation hover. @link org.eclipse.jface.text.source.ISourceViewerExtension4 since version 3.4 introducing API to access a minimal set of content assistant APIs.

interface IWidgetTokenKeeper The following extension interfaces exist:

@link org.eclipse.jface.text.IWidgetTokenKeeperExtension since version 3.0 introducing priorities when requesting a widget token and thus replacing the non-prioritized scheme. It also allows a client to force a widget token keeper to accept focus.

interface ITextOperationTarget The following extension interfaces exist:

@link org.eclipse.jface.text.ITextOperationTargetExtension since version 2.0 introducing text operation enabling/disabling.

(abstract) ITextViewer.setAutoIndentStrategy(strategy, contentType) This method has been replaced by @link ITextViewerExtension2#prependAutoEditStrategy(IAutoEditStrategy, String) and @link ITextViewerExtension2#removeAutoEditStrategy(IAutoEditStrategy, String).

(abstract) ITextViewer.setTextHover(textViewerHover, contentType) This method has been replaced by @link ITextViewerExtension2#setTextHover(ITextHover, String, int).

class TemplateStore In order to load templates contributed using the org.eclipse.ui.editors.templates extension point, use a ContributionTemplateStore.

interface IWidgetTokenOwner The following extension interfaces exist:

@link org.eclipse.jface.text.IWidgetTokenOwnerExtension since version 3.0 introducing priorities when requesting a widget token and thus replacing the non-prioritized scheme.

interface IInformationControlCreator The following extension interfaces exist:

@link org.eclipse.jface.text.IInformationControlCreatorExtension since version 3.0 introducing checks of whether existing information control can be reused and whether information control creators can replace each other.

ArrayStack.empty() . New users of this class should use isEmpty instead.

AbstractLinkedList.LinkedListIterator.current Should be accessed through @link #getLastNodeReturned() to enforce this behaviour.

class ManagementPermission Programmers do not normally create ManagementPermission objects directly. Instead they are created by the security policy code based on reading the security policy file.

AllPredicate.getInstance(predicates) Use @link #allPredicateinstead

class StringBuilder Instances of StringBuilder are not safe for use by multiple threads. If such synchronization is required then it is recommended that @link java.lang.StringBuffer be used.

Boolean.Boolean(value) Unless a new instance is required, the static factory @link #valueOf(boolean) is generally a better choice.

(abstract) PreparedStatement.setObject(parameterIndex, x, targetSqlType, scaleOrLength) If the object has a custom mapping (is of a class implementing the interface SQLData), the JDBC driver should call the method SQLData.writeSQL to write it to the SQL data stream. If, on the other hand, the object is of a class implementing Ref, Blob, Clob, NClob, Struct, java.net.URL, or Array, the driver should pass it to the database as a value of the corresponding SQL type.

(abstract) CallableStatement.setObject(parameterName, x, targetSqlType, scale) If the object has a custom mapping (is of a class implementing the interface SQLData), the JDBC driver should call the method SQLData.writeSQL to write it to the SQL data stream. If, on the other hand, the object is of a class implementing Ref, Blob, Clob, NClob, Struct, java.net.URL, or Array, the driver should pass it to the database as a value of the corresponding SQL type.

(abstract) CallableStatement.setObject(parameterName, x) If the object is of a class implementing the interface SQLData, the JDBC driver should call the method SQLData.writeSQL to write it to the SQL data stream. If, on the other hand, the object is of a class implementing Ref, Blob, Clob, NClob, Struct, java.net.URL, or Array, the driver should pass it to the database as a value of the corresponding SQL type.

MemoryHandler.flush() Note that the current contents of the MemoryHandler buffer are not written out. That requires a ”push”.

Matcher.lookingAt() Like the @link #matches matches method, this method always starts at the beginning of the region; unlike that method, it does not require that the entire region be matched.

class BufferedWriter Unless prompt output is required, it is advisable to wrap a BufferedWriter around any Writer whose write() operations may be costly, such as FileWriters and OutputStreamWriters.

InetAddress.getHostName() If a lookup of the name service is required, call @link #getCanonicalHostName() getCanonicalHostName.

interface Closure Standard implementations of common closures are provided by @link ClosureUtils. These include method invokation and for/while loops.

interface Predicate Standard implementations of common predicates are provided by @link PredicateUtils. These include true, false, instanceof, equals, and, or, not, method invokation and null testing.

class TreeMap If no such object exists, the map should be ”wrapped” using the @link Collections#synchronizedSortedMap Collections.synchronizedSortedMap method.

class ArrayList If no such object exists, the list should be ”wrapped” using the @link Collections#synchronizedList Collections.synchronizedList method.

class CountDownLatch When threads must repeatedly count down in this way, instead use a @link CyclicBarrier.
(abstract) Preferences.addNodeChangeListener(ncl) Events are not generated when indirect descendants of this node are added or removed; a caller desiring such events must register with each descendant.
abstract class AbstractQueue If these requirements cannot be met, consider instead subclassing @link AbstractCollection.
class ThreadPoolExecutor If you would like to ensure that unreferenced pools are reclaimed even if users forget to call @link ThreadPoolExecutor#shutdown, then you must arrange that unused threads eventually die, by setting appropriate keep-alive times, using a lower bound of zero core threads and/or setting @link ThreadPoolExecutor#allowCoreThreadTimeOut(boolean).
(abstract) ICompletionProposal.getAdditionalProposalInfo() If @link ICompletionProposalExtension5 is implemented, this method should not be called any longer.
abstract class AlgorithmParameterGeneratorSpi In case the client does not explicitly initialize the AlgorithmParameterGenerator (via a call to an engineInit method), each provider must supply (and document) a default initialization.
class AlgorithmParameterGenerator In case the client does not explicitly initialize the AlgorithmParameterGenerator (via a call to an init method), each provider must supply (and document) a default initialization.
abstract class KeyPairGenerator In case the client does not explicitly initialize the KeyPairGenerator (via a call to an initialize method), each provider must supply (and document) a default initialization.
ResourceManager.createImage(descriptor) Images allocated in this manner must be disposed by @link #destroyImage(ImageDescriptor), and never by calling @link Image#dispose().
ResourceManager.createColor(descriptor) Any color allocated in this manner must be disposed by calling @link #destroyColor(RGB), or by an eventual call to @link #dispose(). @link Color#dispose() must never been called directly on the returned color.
(abstract) CallableStatement.setAsciiStream(parameterName, x, length) When a very large ASCII value is input to a LONGVARCHAR parameter, it may be more practical to send it via a java.io.InputStream.
TreeSelection.EMPTY This selection should be used instead of null.
(abstract) ILabelDecorator.decorateImage(image, element) Clients should handle labelProviderChangedEvents for the given element to get the complete decoration.
class CompositeSet Add operations require the use of a pluggable strategy. If no strategy is provided then add is unsupported.
class CompositeCollection Add and remove operations require the use of a pluggable strategy. If no strategy is provided then add and remove are unsupported.
org/eclipse/jface/text/formatter These formatting strategies are required to handle the position updating of the document. To facilitate the work with these text edit based formatting strategies, a default implementation called ContextBasedFormattingStrategy is provided.
(abstract) IAction.setAccelerator(keycode) This method should no longer be used for actions in the Eclipse workbench. IWorkbenchCommandSupport and IWorkbenchContextSupport provide all the functionality required for key bindings.
Dialog.convertVerticalDLUsToPixels(fontMetrics, dlus) The required FontMetrics parameter may be created in the following way: GC gc = new GC(control); gc.setFont(control.getFont()); fontMetrics = gc.getFontMetrics(); gc.dispose();
Dialog.convertHorizontalDLUsToPixels(fontMetrics, dlus) The required FontMetrics parameter may be created in the following way: GC gc = new GC(control); gc.setFont(control.getFont()); fontMetrics = gc.getFontMetrics(); gc.dispose();
Dialog.convertWidthInCharsToPixels(fontMetrics, chars) The required FontMetrics parameter may be created in the following way: GC gc = new GC(control); gc.setFont(control.getFont()); fontMetrics = gc.getFontMetrics(); gc.dispose();
Dialog.convertHeightInCharsToPixels(fontMetrics, chars) The required FontMetrics parameter may be created in the following way: GC gc = new GC(control); gc.setFont(control.getFont()); fontMetrics = gc.getFontMetrics(); gc.dispose();
StructuredViewer.update(element, properties) Specifying which properties are affected may allow the viewer to optimize the update. For example, if the label provider is not affected by changes to any of these properties, an update may not actually be required.
interface ITreeSelection It is recommended that clients do not implement this interface but instead use the standard implementation of this interface, @link TreeSelection.
ColumnPixelData.addTrim The default is false for backwards compatibility, but the recommended practice is to specify true, and specify the desired width for the content of the column, rather than adding a fudge factor to the specified width.
StructuredViewer.setSorter(sorter) It is recommended to use setComparator() instead.
StructuredViewer.getSorter() It is recommended to use getComparator() instead.
PreferenceConverter.getFontData(store, name) It is recommended that getFontDataArray is used instead.
ApplicationWindow.getSymbolicFontName() This is not recommended and is included for backwards compatability. It is recommended to use the default font provided by SWT (that is, do not set the font).
CheckboxTreeViewer.setAllChecked(state) @deprecated as this method only checks or unchecks visible items is is recommended that @link #setSubtreeChecked(Object, boolean) is used instead.
CheckboxTableViewer.CheckboxTableViewer(parent, style) If the caller adds its own columns, uses Table.setHeadersVisible(true), or needs to handle dynamic resizing of the table, it is recommended to create the Table itself, specifying the SWT.CHECK style bit (along with any other style bits needed), and use new CheckboxTableViewer(Table) rather than this constructor.
PreferenceConverter.getDefaultFontData(store, name) This method is provided for backwards compatibility. It is recommended that getDefaultFontDataArray is used instead.
SecurityManager.inCheck @deprecated This type of security checking is not recommended. It is recommended that the checkPermission call be used instead.
BigDecimal.BigDecimal(val, mc) The results of this constructor can be somewhat unpredictable and its use is generally not recommended; see the notes under the @link #BigDecimal(double) constructor.
SecurityManager.currentLoadedClass() This type of security checking is not recommended. It is recommended that the checkPermission call be used instead.
BigInteger.BigInteger(bitLength, certainty, rnd) It is recommended that the @link #probablePrime probablePrime method be used in preference to this constructor unless there is a compelling need to specify a certainty.
SecurityManager.classDepth(name) @deprecated This type of security checking is not recommended. It is recommended that the checkPermission call be used instead.
SecurityManager.inClass(name) @deprecated This type of security checking is not recommended. It is recommended that the checkPermission call be used instead.
SecurityManager.classLoaderDepth() @deprecated This type of security checking is not recommended. It is recommended that the checkPermission call be used instead.
SecurityManager.inClassLoader() @deprecated This type of security checking is not recommended. It is recommended that the checkPermission call be used instead.
SecurityManager.currentClassLoader() @deprecated This type of security checking is not recommended. It is recommended that the checkPermission call be used instead.
File.toURL() @deprecated This method does not automatically escape characters that are illegal in URLs. It is recommended that new code convert an abstract pathname into a URL by first converting it into a URI, via the @link #toURI() toURI method, and then converting the URI into a URL via the @link java.net.URI#toURL() URI.toURL method.
class StringTokenizer StringTokenizer is a legacy class that is retained for compatibility reasons although its use is discouraged in new code. It is recommended that anyone seeking this functionality use the split method of String or the java.util.regex package instead.
EnumSet.of(e1, e2, e3, e4, e5) Overloadings of this method exist to initialize an enum set with one through five elements. A sixth overloading is provided that uses the varargs feature. This overloading may be used to create an enum set initially containing an arbitrary number of elements, but is likely to run slower than the overloadings that do not use varargs.
EnumSet.of(e1, e2, e3) Overloadings of this method exist to initialize an enum set with one through five elements. A sixth overloading is provided that uses the varargs feature. This overloading may be used to create an enum set initially containing an arbitrary number of elements, but is likely to run slower than the overloadings that do not use varargs.
(abstract) Annotation.hashCode() The hash code of an array member-value is computed by calling the appropriate overloading of @link java.util.Arrays#hashCode(long[]) Arrays.hashCode on the value. (There is one overloading for each primitive type, and one for object reference types.)
EnumSet.of(e1, e2, e3, e4) Overloadings of this method exist to initialize an enum set with one through five elements. A sixth overloading is provided that uses the varargs feature. This overloading may be used to create an enum set initially containing an arbitrary number of elements, but is likely to run slower than the overloadings that do not use varargs.
EnumSet.of(e1, e2) Overloadings of this method exist to initialize an enum set with one through five elements. A sixth overloading is provided that uses the varargs feature. This overloading may be used to create an enum set initially containing an arbitrary number of elements, but is likely to run slower than the overloadings that do not use varargs.
EnumSet.of(first, rest) This factory, whose parameter list uses the varargs feature, may be used to create an enum set initially containing an arbitrary number of elements, but it is likely to run slower than the overloadings that do not use varargs.
EnumSet.of(e) Overloadings of this method exist to initialize an enum set with one through five elements. A sixth overloading is provided that uses the varargs feature. This overloading may be used to create an enum set initially containing an arbitrary number of elements, but is likely to run slower than the overloadings that do not use varargs.
(abstract) ResultSet.updateNCharacterStream(columnIndex, x) Consult your JDBC driver documentation to determine if it might be more efficient to use a version of updateNCharacterStream which takes a length parameter.

(abstract) ResultSet.updateAsciiStream(columnLabel, x) Consult your JDBC driver documentation to determine if it might be more efficient to use a version of updateAsciiStream which takes a length parameter.

(abstract) CallableStatement.setBinaryStream(parameterName, x) Consult your JDBC driver documentation to determine if it might be more efficient to use a version of setBinaryStream which takes a length parameter.

(abstract) PreparedStatement.setNCharacterStream(parameterIndex, value) Consult your JDBC driver documentation to determine if it might be more efficient to use a version of setNCharacterStream which takes a length parameter.

(abstract) ResultSet.updateCharacterStream(columnLabel, reader) Consult your JDBC driver documentation to determine if it might be more efficient to use a version of updateCharacterStream which takes a length parameter.

Pattern.matches(regex, input) If a pattern is to be used multiple times, compiling it once and reusing it will be more efficient than invoking this method each time.

(abstract) ResultSet.updateBinaryStream(columnIndex, x) Consult your JDBC driver documentation to determine if it might be more efficient to use a version of updateBinaryStream which takes a length parameter.

(abstract) ResultSet.updateAsciiStream(columnIndex, x) Consult your JDBC driver documentation to determine if it might be more efficient to use a version of updateAsciiStream which takes a length parameter.

(abstract) ResultSet.updateBlob(columnIndex, inputStream) Consult your JDBC driver documentation to determine if it might be more efficient to use a version of updateBlob which takes a length parameter.

(abstract) Pack200.Unpacker.unpack(in, out) It may be more efficient to read the Pack200 archive to a file and pass the File object, using the alternate method described below.

(abstract) ResultSet.updateBinaryStream(columnLabel, x) Consult your JDBC driver documentation to determine if it might be more efficient to use a version of updateBinaryStream which takes a length parameter.

(abstract) ResultSet.updateClob(columnLabel, reader) Consult your JDBC driver documentation to determine if it might be more efficient to use a version of updateClob which takes a length parameter.

(abstract) PreparedStatement.setCharacterStream(parameterIndex, reader) Consult your JDBC driver documentation to determine if it might be more efficient to use a version of setCharacterStream which takes a length parameter.

(abstract) CallableStatement.setAsciiStream(parameterName, x) Consult your JDBC driver documentation to determine if it might be more efficient to use a version of setAsciiStream which takes a length parameter.

(abstract) ResultSet.updateBlob(columnLabel, inputStream) Consult your JDBC driver documentation to determine if it might be more efficient to use a version of updateBlob which takes a length parameter.

(abstract) ResultSet.updateNClob(columnIndex, reader) Consult your JDBC driver documentation to determine if it might be more efficient to use a version of updateNClob which takes a length parameter.

AbstractListViewer.add(element) Note that there is another method for efficiently processing the simultaneous addition of multiple elements.

(abstract) IPreferenceStore.setValue(name, value) Note that the preferred way of re-initializing a preference to its default value is to call setToDefault.

AbstractTreeViewer.remove(elementsOrTreePaths) Note that there is another method for efficiently processing the simultaneous removal of multiple elements.

(abstract) IAnnotationAccessExtension.isPaintable(annotation) Note that this method is not used when drawing annotations on the editor's text widget. This is handled trough a @link org.eclipse.jface.text.source.AnnotationPainter.IDrawingStrategy.

AllPredicate.getInstance(predicates) @deprecated Use @link #allPredicate(Predicate¡? super T¿...) instead

SecureRandom.getSeed(numBytes) This method is only included for backwards compatibility. The caller is encouraged to use one of the alternative getInstance methods to obtain a SecureRandom object, and then call the generateSeed method to obtain seed bytes from that object.

ClassLoader.loadClass(name, resolve) Subclasses of ClassLoader are encouraged to override @link #findClass(String), rather than this method.

class DateFormatSymbols Typically you shouldn't use DateFormatSymbols directly. Rather, you are encouraged to create a date-time formatter with the DateFormat class's factory methods: getTimeInstance, getDateInstance, or getDateTimeInstance.

ImageDescriptor.createWithFlags(originalImage, swtFlags) Clients are encouraged to supply their own disabled/grayed/etc. images rather than using a default image and transforming it.

ImageDescriptor.createImage() Clients are encouraged to use create/destroyResource and downcast the result to Image rather than using createImage.

ComparableComparator.getInstance() Developers are encouraged to use the comparator returned from this method instead of constructing a new instance to reduce allocation and GC overhead when multiple comparable comparators may be used in the same VM.

BooleanComparator.getTrueFirstComparator() Clients are encouraged to use the value returned from this method instead of constructing a new instance to reduce allocation and garbage collection overhead when multiple BooleanComparators may be used in the same virtual machine.

BooleanComparator.getFalseFirstComparator() Clients are encouraged to use the value returned from this method instead of constructing a new instance to reduce allocation and garbage collection overhead when multiple BooleanComparators may be used in the same virtual machine.

BooleanComparator.getBooleanComparator(trueFirst) Clients are encouraged to use the value returned from this method instead of constructing a new instance to reduce allocation and garbage collection overhead when multiple BooleanComparators may be used in the same virtual machine.

(abstract) ILazyContentProvider.updateElement(index) NOTE #updateElement(int index) can be used to determine selection values.

AbstractTableViewer.remove(element) Note that there is another method for efficiently processing the simultaneous removal of multiple elements.

AbstractTreeViewer.add(parentElementOrTreePath, childElement) Note that there is another method for efficiently processing the simultaneous addition of multiple elements.

BigDecimal.valueOf(unscaledVal, scale) This @literal "static factory method" is provided in preference to a (long, int) constructor because it allows for reuse of frequently used BigDecimal values.

class LockSupport The three forms of park each also support a blocker object parameter. This object is recorded while the thread is blocked to permit monitoring and diagnostic tools to identify the reasons that threads are blocked. (Such tools may access blockers using method @link #getBlocker.) The use of these forms rather than the original forms without this parameter is strongly encouraged. The normal argument to supply as a blocker within a lock implementation is this.

class SimpleDateFormat SimpleDateFormat allows you to start by choosing any user-defined patterns for date-time formatting. However, you are encouraged to create a date-time formatter with either getTimeInstance, getDateInstance, or getDateTimeInstance in DateFormat.

StructuredViewer.refresh(element, updateLabels) If only the given element needs updating, it is more efficient to use the update methods.

AbstractListViewer.remove(element) . Note that there is another method for efficiently processing the simultaneous removal of multiple elements.

class Pattern The statement

```
boolean b = Pattern.matches("a*b", "aaaaab");
```

is equivalent to the three statements above, though for repeated matches it is less efficient since it does not allow the compiled pattern to be reused.

abstract class Policy The default Policy implementation can be changed by setting the value of the "policy.provider" security property (in the Java security properties file) to the fully qualified name of the desired Policy subclass implementation.

ImageAndMessageArea.ImageAndMessageArea(parent, style) The style bit SWT.WRAP should be used if a larger message area is desired.

(abstract) AtomicReferenceFieldUpdater.weakCompareAndSet(obj, expect, update) May fail spuriously and does not provide ordering guarantees, so is only rarely an appropriate alternative to compareAndSet.

AtomicMarkableReference.weakCompareAndSet(expectedReference, newReference, expectedMark, newMark) May fail spuriously and does not provide ordering guarantees, so is only rarely an appropriate alternative to compareAndSet.

AtomicReferenceArray.weakCompareAndSet(i, expect, update) May fail spuriously and does not provide ordering guarantees, so is only rarely an appropriate alternative to compareAndSet.

AtomicLongArray.weakCompareAndSet(i, expect, update) May fail spuriously and does not provide ordering guarantees, so is only rarely an appropriate alternative to compareAndSet.

class ScheduledThreadPoolExecutor However, this class provides alternative protected extension method decorateTask (one version each for Runnable and Callable) that can be used to customize the concrete task types used to execute commands entered via execute, submit, schedule, scheduleAtFixedRate, and scheduleWithFixed-Delay.

ColumnViewer.setCellEditors(editors) Since 3.3, an alternative API is available, see @link ViewerColumn#setEditingSupport(EditingSupport) for a more flexible way of editing values in a column viewer.

ColumnViewer.setColumnProperties(columnProperties) Since 3.3, an alternative API is available, see @link ViewerColumn#setEditingSupport(EditingSupport) for a more flexible way of editing values in a column viewer.

ColumnViewer.isCellEditorActive() Since 3.3, an alternative API is available, see @link ViewerColumn#setEditingSupport(EditingSupport) for a more flexible way of editing values in a column viewer.

abstract class ImageDescriptor There are two ways to get an Image from an ImageDescriptor. The method createImage will always return a new Image which must be disposed by the caller. Alternatively, createResource() returns a shared Image.

interface IPresentationRepairer This interface may be implemented by clients. Alternatively, clients may use the rule-based default implementation @link org.eclipse.jface.text.rules.D

ColumnViewer.getColumnProperties() Since 3.3, an alternative API is available, see @link ViewerColumn#setEditingSupport(EditingSupport) for a more flexible way of editing values in a column viewer.

class Logger Logger names can be arbitrary strings, but they should normally be based on the package name or class name of the logged component, such as java.net or javax.swing. In addition it is possible to create "anonymous" Loggers that are not stored in the Logger namespace.

Pattern.CASE_INSENSITIVE Unicode-aware case-insensitive matching can be enabled by specifying the @link #UNICODE_CASE flag in conjunction with this flag.

interface IReconciler @link org.eclipse.jface.text.reconciler.IReconcilerExtension since version 3.0 introducing the ability to be aware of documents with multiple partitionings.

ColumnViewer.getCellEditors() Since 3.3, an alternative API is available, see @link ViewerColumn#setEditingSupport(EditingSupport) for a more flexible way of editing values in a column viewer.

Dialog.getImage(key) Dialog does not use the following images in the registry DLG_IMG_ERROR DLG_IMG_INFO DLG_IMG_QUESTION DLG_IMG_WARNING They are now coming directly from SWT, see ImageRegistry. For backwards compatibility they are still supported, however new code should use SWT for these.

interface IAnnotationAccess The following extension interfaces exist:

@link org.eclipse.jface.text.source.IAnnotationAccessExtension since version 3.0 replacing all methods in that interface @link IAnnotationAccessExtension2 since version 3.2 allowing to set a quick assist assistant to an annotation access.

abstract class ResourceBundle You do not have to restrict yourself to using a single family of ResourceBundles. For example, you could have a set of bundles for exception messages, ExceptionResources (ExceptionResources_fr, ExceptionResources_de, ...), and one for widgets, WidgetResource (WidgetResources_fr, WidgetResources_de, ...); breaking up the resources however you like.

File.getUsableSpace() When possible, this method checks for write permissions and other operating system restrictions and will therefore usually provide a more accurate estimate of how much new data can actually be written than @link #getFreeSpace.

interface IAnnotationHover The following extension interfaces exist:

@link org.eclipse.jface.text.source.IAnnotationHoverExtension since version 3.0 allowing a text hover to provide a creator for the hover control. This allows for sophisticated hovers in a way that information computed by the hover can be displayed in the best possible form.

interface ITextHover The following extension interfaces exist:

@link org.eclipse.jface.text.ITextHoverExtension since version 3.0 allowing a text hover to provide a creator for the hover control. This allows for sophisticated hovers in a way that information computed by the hover can be displayed in the best possible form. @link org.eclipse.jface.text.ITextHoverExtension2 since version 3.4 allowing a text hover to return hover-specific information objects.

interface ITextViewerExtension2 It replaces @link org.eclipse.jface.text.ITextViewer#setTextHover(ITextHover, String) with a new method that allows to specify state masks for a better control of the hover behavior.

(abstract) Collator.getCollationKey(source) CollationKeys provide better performance than Collator.compare when Strings are involved in multiple comparisons.

MessageFormat.setFormat(formatElementIndex, newFormat) Since the order of format elements in a pattern string often changes during localization, it is generally better to use the @link #setFormatByArgumentIndex setFormatByArgumentIndex method, which accesses format elements based on the argument index they specify.

abstract class Collator For comparing Strings exactly once, the compare method provides the best performance. When sorting a list of Strings however, it is generally necessary to compare each String multiple times. In this case, CollationKeys provide better performance.

MessageFormat.setFormats(newFormats) Since the order of format elements in a pattern string often changes during localization, it is generally better to use the @link #setFormatsByArgumentIndex setFormatsByArgumentIndex method, which assumes an order of formats corresponding to the order of elements in the arguments array passed to the format methods or the result array returned by the parse methods.

MessageFormat.getFormats() Since the order of format elements in a pattern string often changes during localization, it's generally better to use the @link #getFormatsByArgumentIndex getFormatsByArgumentIndex method, which assumes an order of formats corresponding to the order of elements in the arguments array passed to the format methods or the result array returned by the parse methods.

OwnerDrawLabelProvider.setUpOwnerDraw(viewer) @deprecated Since 3.4, the default implementation of @link CellLabelProvider#initialize(ColumnViewer, ViewerColumn) in this class will set up the necessary owner draw callbacks automatically. Calls to this method can be removed.

ToolBarManager.dispose() Note that this method does not clean up references between this tool bar manager and its associated contribution items. Use removeAll for that purpose.

CoolBarManager.dispose() Note that this method does not clean up references between this cool bar manager and its associated contribution items. Use removeAll for that purpose.

MenuManager.dispose() Note that this method does not clean up references between this menu manager and its associated contribution items. Use removeAll for that purpose.

ContributionItem.setId(itemId) It should be set once either in the constructor or using this method.

class HashSet If multiple threads access a hash set concurrently, and at least one of the threads modifies the set, it must be synchronized externally. This is typically accomplished by synchronizing on some object that naturally encapsulates the set. If no such object exists, the set should be "wrapped" using the @link Collections#synchronizedSet Collections.synchronizedSet method. This is best done at creation time, to prevent accidental unsynchronized access to the set:

Set s = Collections.synchronizedSet(new HashSet(...));

class LinkedHashSet If multiple threads access a linked hash set concurrently, and at least one of the threads modifies the set, it must be synchronized externally. This is typically accomplished by synchronizing on some object that naturally encapsulates the set. If no such object exists, the set should be "wrapped" using the @link Collections#synchronizedSet Collections.synchronizedSet method. This is best done at creation time, to prevent accidental unsynchronized access to the set:

Set s = Collections.synchronizedSet(new LinkedHashSet(...));

class IdentityHashMap If multiple threads access an identity hash map concurrently, and at least one of the threads modifies the map structurally, it must be synchronized externally. (A structural modification is any operation that adds or deletes one or more mappings; merely changing the value associated with a key that an instance already contains is not a structural modification.) This is typically accomplished by synchronizing on some object that naturally encapsulates the map. If no such object exists, the map should be "wrapped" using the @link Collections#synchronizedMap Collections.synchronizedMap method. This is best done at creation time, to prevent accidental unsynchronized access to the map:

Map m = Collections.synchronizedMap(new IdentityHashMap(...));

class TreeMap If multiple threads access a map concurrently, and at least one of the threads modifies the map structurally, it must be synchronized externally. (A structural modification is any operation that adds or deletes one or more mappings; merely changing the value associated with an existing key is not a structural modification.) This is typically accomplished by synchronizing on some object that naturally encapsulates the map. If no such object exists, the map should be "wrapped" using the @link Collections#synchronizedSortedMap Collections.synchronizedSortedMap method. This is best done at creation time, to prevent accidental unsynchronized access to the map:

SortedMap m = Collections.synchronizedSortedMap(new TreeMap(...));

System.getenv(name) For these reasons, environment variables are more likely to have unintended side effects. It is best to use system properties where possible.

(abstract) AbstractPreferences.flushSpi() If the backing store naturally flushes an entire subtree at once, the implementer is encouraged to override flush(), rather than merely overriding this method.

(abstract) AbstractPreferences.syncSpi() If the backing store naturally syncs an entire subtree at once, the implementer is encouraged to override sync(), rather than merely overriding this method.

(abstract) ResultSet.updateNCharacterStream(columnLabel, reader) Consult your JDBC driver documentation to determine if it might be more efficient to use a version of updateNCharacterStream which takes a length parameter.

(abstract) CallableStatement.setNCharacterStream(parameterName, value) Consult your JDBC driver documentation to determine if it might be more efficient to use a version of setNCharacterStream which takes a length parameter.

(abstract) PreparedStatement.setClob(parameterIndex, reader) Consult your JDBC driver documentation to determine if it might be more efficient to use a version of setClob which takes a length parameter.

Reader.lock For efficiency, a character-stream object may use an object other than itself to protect critical sections.

Writer.lock For efficiency, a character-stream object may use an object other than itself to protect critical sections.

StatusLineManager.dispose() Note that this method does not clean up references between this status line manager and its associated contribution items. Use removeAll for that purpose.

(abstract) IPreferenceStore.setValue(name, value) Note that the preferred way of re-initializing a preference to its default value is to call setToDefault.

JFaceResources.getImageRegistry() Note that the static convenience method getImage is also provided on this class.

AbstractTableViewer.add(element) Note that there is another method for efficiently processing the simultaneous addition of multiple elements.

StructuredViewer.refresh(element, updateLabels) If only the given element needs updating, it is more efficient to use the update methods.

ErrorDialog.createDialogArea(parent) If the old behavior is desired by subclasses, get the returned composite's layout data and set grabExcessVerticalSpace to true.

CertPathBuilder.getDefaultType() The default CertPathBuilder type can be changed by setting the value of the "certpathbuilder.type" security property (in the Java security properties file) to the desired type.

class Naming Once a remote object is registered (bound) with the RMI registry on the local host, callers on a remote (or local) host can lookup the remote object by name, obtain its reference, and then invoke remote methods on the object. A registry may be shared by all servers running on a host or an individual server process may create and use its own registry if desired (see java.rmi.registry.LocateRegistry.createRegistry method for details).

CertPathValidator.getDefaultType() The default CertPathValidator type can be changed by setting the value of the "certpathvalidator.type" security property (in the Java security properties file) to the desired type.

ColumnViewer.setCellModifier(modifier) Since 3.3, an alternative API is available, see @link ViewerColumn#setEditingSupport(EditingSupport) for a more flexible way of editing values in a column viewer.

class ComboViewer This class is intended as an alternative to the JFace ListViewer, which displays its content in a combo box rather than a list.

AtomicLong.weakCompareAndSet(expect, update) May fail spuriously and does not provide ordering guarantees, so is only rarely an appropriate alternative to compareAndSet.
AtomicInteger.weakCompareAndSet(expect, update) May fail spuriously and does not provide ordering guarantees, so is only rarely an appropriate alternative to compareAndSet.
(abstract) AtomicLongFieldUpdater.weakCompareAndSet(obj, expect, update) May fail spuriously and does not provide ordering guarantees, so is only rarely an appropriate alternative to compareAndSet.
abstract class ContentHandler By default it looks in sun.net.www.content, but users can define a vertical-bar delimited set of class prefixes to search through in addition by defining the java.content.handler.pkgs property.
class InetAddress If the default behavior is not desired, then a Java security property can be set to a different Time-to-live (TTL) value for positive caching.
StructuredViewer.update(elements, properties) This does not handle structural changes (e.g. addition or removal of elements), and does not update any other related elements (e.g. child elements). To handle structural changes, use the refresh methods instead.
class LockSupport These methods are designed to be used as tools for creating higher-level synchronization utilities, and are not in themselves useful for most concurrency control applications.
FontRegistry.defaultFont() This method should only be called from the UI thread. If you are not on the UI thread then wrap the call with a PlatformUI.getWorkbench().getDisplay().syn
in order to guarantee the correct result.
interface SQLXML Because databases may use an optimized representation for the XML, accessing the value through getSource() and setResult() can lead to improved processing performance without serializing to a stream representation and parsing the XML.
LinkedBlockingQueue.offer(e) When using a capacity-restricted queue, this method is generally preferable to method @link BlockingQueue#add add, which can fail to insert an element only by throwing an exception.
(abstract) Deque.addLast(e) When using a capacity-restricted deque, it is generally preferable to use method @link #offerLast.
(abstract) Queue.offer(e) When using a capacity-restricted queue, this method is generally preferable to @link #add, which can fail to insert an element only by throwing an exception.
(abstract) BlockingDeque.addFirst(e) When using a capacity-restricted deque, it is generally preferable to use @link #offerFirst(Object) offerFirst.
(abstract) BlockingQueue.offer(e) When using a capacity-restricted queue, this method is generally preferable to @link #add, which can fail to insert an element only by throwing an exception.
(abstract) BlockingDeque.addLast(e) When using a capacity-restricted deque, it is generally preferable to use @link #offerLast(Object) offerLast.
(abstract) BlockingQueue.add(e) When using a capacity-restricted queue, it is generally preferable to use @link #offer(Object) offer.
abstract class EnumSet This is best done at creation time, to prevent accidental unsynchronized access:
 Set¡MyEnum¿ s = Collections.synchronizedSet(EnumSet.noneOf(MyEnum.class));
(abstract) Collator.compare(source, target) For a one time comparison, this method has the best performance. If a given String will be involved in multiple comparisons, CollationKey.compareTo has the best performance.
BigDecimal.divideAndRemainder(divisor, mc) Note that if both the integer quotient and remainder are needed, this method is faster than using the divideToIntegralValue and remainder methods separately because the division need only be carried out once.
ObjectOutputStream.writeUTF(str) Note that there is a significant difference between writing a String into the stream as primitive data or as an Object.

5 Synchronization Directive

5.1 Synchronization Directive

BeanContextSupport.addAll(c) implementations must synchronized on the hierarchy lock
BeanContextSupport.removeAll(c) implementations must synchronized on the hierarchy lock
BeanContextSupport.retainAll(c) implementations must synchronized on the hierarchy lock and "children" protected field
BeanContextSupport.clear() implementations must synchronized on the hierarchy lock
(abstract) Condition.signal() thread must then re-acquire the lock before returning from await.
(abstract) Condition.signalAll() Each thread must re-acquire the lock before it can return from await.
(abstract) Condition.awaitNanos(nanosTimeout) In all cases, before this method can return the current thread must re-acquire the lock associated with this condition.
(abstract) Condition.awaitUninterruptibly() In all cases, before this method can return the current thread must re-acquire the lock associated with this condition.
(abstract) Condition.awaitUntil(deadline) In all cases, before this method can return the current thread must re-acquire the lock associated with this condition.
abstract class NumberFormat If multiple threads access a format concurrently, it must be synchronized externally.
abstract class DateFormat If multiple threads access a format concurrently, it must be synchronized externally.
class DecimalFormat f multiple threads access a format concurrently, it must be synchronized externally.
class MessageFormat If multiple threads access a format concurrently, it must be synchronized externally.
MapUtils.verbosePrint(out, label, map) This method is NOT thread-safe in any special way. You must manually synchronize on either this class or the stream as required.
MapUtils.debugPrint(out, label, map) This method is NOT thread-safe in any special way. You must manually synchronize on either this class or the stream as required.
Writer.lock A subclass should therefore use the object in this field rather than this or a synchronized method.
Reader.lock A subclass should therefore use the object in this field rather than this or a synchronized method.
(abstract) Condition.await() In all cases, before this method can return the current thread must re-acquire the lock associated with this condition.
AbstractReferenceMap.purge() Note that this method is not synchronized!
class BoundedFifoBuffer Note that this implementation is not synchronized.
class CircularFifoBuffer Note that this implementation is not synchronized.
class NodeCachingLinkedList Note that this implementation is not synchronized.
class PriorityBuffer Note that this implementation is not synchronized.
class UnboundedFifoBuffer Note that this implementation is not synchronized.
class ComparatorChain Instances of ComparatorChain are not synchronized. The class is not thread-safe at construction time, but it is thread-safe to perform multiple comparisons after all the setup operations are complete.
class FixedOrderComparator Instances of FixedOrderComparator are not synchronized. The class is not thread-safe at construction time, but it is thread-safe to perform multiple comparisons after all the setup operations are complete.
CursorableLinkedList.listIterator() The cursor enables iteration and list changes to occur in any order without invalidating the iterator (from one thread).
CursorableLinkedList.listIterator(fromIndex) The cursor enables iteration and list changes to occur in any order without invalidating the iterator (from one thread).
CursorableLinkedList.cursor() The cursor enables iteration and list changes to occur in any order without invalidating the iterator (from one thread).
class StaticBucketMap Each bucket in the hash table has its own monitor, so two threads can safely operate on the map at the same time, often without incurring any monitor contention.
class SynchronizedSortedBag Methods are synchronized, then forwarded to the decorated bag. Iterators must be separately synchronized around the loop.
class SynchronizedBag Iterators must be separately synchronized around the loop.
class SynchronizedCollection Iterators must be manually synchronized:
 synchronized (coll) Iterator it = coll.iterator(); // do stuff with iterator
SynchronizedCollection.iterator() Iterators must be manually synchronized.
 synchronized (coll) Iterator it = coll.iterator(); // do stuff with iterator
 @return an iterator that must be manually synchronized on the collection
SynchronizedList.listIterator() Iterators must be manually synchronized.
 synchronized (coll) ListIterator it = coll.listIterator(); // do stuff with iterator
 @return an iterator that must be manually synchronized on the collection
SynchronizedList.listIterator(index) Iterators must be manually synchronized.
 synchronized (coll) ListIterator it = coll.listIterator(3); // do stuff with iterator
 @return an iterator that must be manually synchronized on the collection
abstract class AbstractReferenceMap This implementation is not synchronized. You can use @link java.util.Collections#synchronizedMap to provide synchronized access to a ReferenceMap.
class ReferenceIdentityMap This implementation is not synchronized. You can use @link java.util.Collections#synchronizedMap to provide synchronized access to a ReferenceIdentityMap. Remember that synchronization will not stop the garbage collecter removing entries.
class ReferenceMap This implementation is not synchronized. You can use @link java.util.Collections#synchronizedMap to provide synchronized access to a ReferenceMap. Remember that synchronization will not stop the garbage collecter removing entries.

MapUtils.synchronizedMap(map) You must manually synchronize on the returned buffer's iterator to avoid non-deterministic behavior:

Map m = MapUtils.synchronizedMap(myMap); Set s = m.keySet(); // outside synchronized block synchronized (m) // synchronized on MAP! Iterator i = s.iterator(); while (i.hasNext()) process (i.next());

MapUtils.synchronizedSortedMap(map) You must manually synchronize on the returned buffer's iterator to avoid non-deterministic behavior:

Map m = MapUtils.synchronizedSortedMap(myMap); Set s = m.keySet(); // outside synchronized block synchronized (m) // synchronized on MAP! Iterator i = s.iterator(); while (i.hasNext()) process (i.next());

ListUtils.synchronizedList(list) You must manually synchronize on the returned buffer's iterator to avoid non-deterministic behavior:

List list = ListUtils.synchronizedList(myList); synchronized (list) Iterator i = list.iterator(); while (i.hasNext()) process (i.next());

BufferUtils.synchronizedBuffer(buffer) Much like the synchronized collections returned by @link java.util.Collections, you must manually synchronize on the returned buffer's iterator to avoid non-deterministic behavior:

Buffer b = BufferUtils.synchronizedBuffer(myBuffer); synchronized (b) Iterator i = b.iterator(); while (i.hasNext()) process (i.next());

CollectionUtils.synchronizedCollection(collection) You must manually synchronize on the returned buffer's iterator to avoid non-deterministic behavior:

Collection c = CollectionUtils.synchronizedCollection(myCollection); synchronized (c) Iterator i = c.iterator(); while (i.hasNext()) process (i.next());

BagUtils.synchronizedBag(bag) It is imperative that the user manually synchronize on the returned bag when iterating over it:

Bag bag = BagUtils.synchronizedBag(new HashBag()); ... synchronized(bag) Iterator i = bag.iterator(); // Must be in synchronized block while (i.hasNext()) foo(i.next());

SetUtils.synchronizedSortedSet(set) You must manually synchronize on the returned buffer's iterator to avoid non-deterministic behavior:

Set s = SetUtils.synchronizedSet(mySet); synchronized (s) Iterator i = s.iterator(); while (i.hasNext()) process (i.next());

interface Lock It is recommended that to avoid confusion you never use Lock instances in this way, except within their own implementation.

AbstractQueuedSynchronizer.tryAcquireShared(arg) @return a negative value on failure; zero if acquisition in shared mode succeeded but no subsequent shared-mode acquire can succeed; and a positive value if acquisition in shared mode succeeded and subsequent shared-mode acquires might also succeed, in which case a subsequent waiting thread must check availability.

AbstractQueuedLongSynchronizer.tryAcquireShared(arg) @return a negative value on failure; zero if acquisition in shared mode succeeded but no subsequent shared-mode acquire can succeed; and a positive value if acquisition in shared mode succeeded and subsequent shared-mode acquires might also succeed, in which case a subsequent waiting thread must check availability.

abstract class Format If multiple threads access a format concurrently, it must be synchronized externally.

(abstract) Condition.awaitNanos(nanosTimeout) In either case the implementation must ensure that the signal is redirected to another waiting thread, if there is one.

(abstract) Lock.newCondition() Before waiting on the condition the lock must be held by the current thread.

UndoableEditSupport.endUpdate() DEADLOCK WARNING: Calling this method may call undoableEditHappened in all listeners. It is unwise to call this method from one of its listeners.

Collections.synchronizedCollection(c) It is imperative that the user manually synchronize on the returned collection when iterating over it:

Collection c = Collections.synchronizedCollection(myCollection); ... synchronized(c) Iterator i = c.iterator(); // Must be in the synchronized block while (i.hasNext()) foo(i.next());

Collections.synchronizedSet(s) It is imperative that the user manually synchronize on the returned set when iterating over it:

Set s = Collections.synchronizedSet(new HashSet()); ... synchronized(s) Iterator i = s.iterator(); // Must be in the synchronized block while (i.hasNext()) foo(i.next());

Collections.synchronizedSortedSet(s) It is imperative that the user manually synchronize on the returned sorted set when iterating over it or any of its subSet, headSet, or tailSet views.

SortedSet s = Collections.synchronizedSortedSet(new TreeSet()); ... synchronized(s) Iterator i = s.iterator(); // Must be in the synchronized block while (i.hasNext()) foo(i.next());

Collections.synchronizedList(list) It is imperative that the user manually synchronize on the returned list when iterating over it:

List list = Collections.synchronizedList(new ArrayList()); ... synchronized(list) Iterator i = list.iterator(); // Must be in synchronized block while (i.hasNext()) foo(i.next());

Collections.synchronizedMap(m) It is imperative that the user manually synchronize on the returned map when iterating over any of its collection views:

Map m = Collections.synchronizedMap(new HashMap()); ... Set s = m.keySet(); // Needn't be in synchronized block ... synchronized(m) // Synchronizing on m, not s! Iterator i = s.iterator(); // Must be in synchronized block while (i.hasNext()) foo(i.next());

Collections.synchronizedSortedMap(m) It is imperative that the user manually synchronize on the returned sorted map when iterating over any of its collection views, or the collections views of any of its subMap, headMap or tailMap views.

SortedMap m = Collections.synchronizedSortedMap(new TreeMap()); ... Set s = m.keySet(); // Needn't be in synchronized block ... synchronized(m) // Synchronizing on m, not s! Iterator i = s.iterator(); // Must be in synchronized block while (i.hasNext()) foo(i.next());

class ProcessBuilder Note that this class is not synchronized. If multiple threads access a ProcessBuilder instance concurrently, and at least one of the threads modifies one of the attributes structurally, it must be synchronized externally.

abstract class CertStoreSpi The public methods of all CertStoreSpi objects must be thread-safe.

class PolicyQualifierInfo All PolicyQualifierInfo objects must be immutable and thread-safe.

CollectionCertStoreParameters.CollectionCertStoreParameters(collection) If the Collection will be modified by one thread while another thread is calling a method of a Collection CertStore that has been initialized with this Collection, the Collection must have fail-fast iterators.

abstract class CertPath All CertPath objects must be thread-safe. That is, multiple threads may concurrently invoke the methods defined in this class on a single CertPath object (or more than one) with no ill effects.

interface PolicyNode All PolicyNode objects must be immutable and thread-safe.

class CertStore All public methods of CertStore objects must be thread-safe.

interface CertSelector Unless otherwise specified, the methods defined in this interface are not thread-safe. Multiple threads that need to access a single object concurrently should synchronize amongst themselves and provide the necessary locking.

interface CRLSelector Unless otherwise specified, the methods defined in this interface are not thread-safe. Multiple threads that need to access a single object concurrently should synchronize amongst themselves and provide the necessary locking.

abstract class CertPathValidatorSpi Threads that need to access a single CertPathValidatorSpi instance concurrently should synchronize amongst themselves and provide the necessary locking before calling the wrapping CertPathValidator object.

abstract class CertPathBuilderSpi Instances of this class need not be protected against concurrent access from multiple threads. Threads that need to access a single CertPathBuilderSpi instance concurrently should synchronize amongst themselves and provide the necessary locking before calling the wrapping CertPathBuilder object.

class LDAPCertStoreParameters Unless otherwise specified, the methods defined in this class are not thread-safe. Multiple threads that need to access a single object concurrently should synchronize amongst themselves and provide the necessary locking.

abstract class PermissionCollection Subclass implementations of PermissionCollection should assume that they may be called simultaneously from multiple threads, and therefore should be synchronized properly.

class CollectionCertStoreParameters Unless otherwise specified, the methods defined in this class are not thread-safe. Multiple threads that need to access a single object concurrently should synchronize amongst themselves and provide the necessary locking. Multiple threads each manipulating separate objects need not synchronize.

class CertStoreException Unless otherwise specified, the methods defined in this class are not thread-safe. Multiple threads that need to access a single object concurrently should synchronize amongst themselves and provide the necessary locking. Multiple threads each manipulating separate objects need not synchronize.

class CertPathBuilderException Unless otherwise specified, the methods defined in this class are not thread-safe. Multiple threads that need to access a single object concurrently should synchronize amongst themselves and provide the necessary locking. Multiple threads each manipulating separate objects need not synchronize.

class PKIXBuilderParameters Unless otherwise specified, the methods defined in this class are not thread-safe. Multiple threads that need to access a single object concurrently should synchronize amongst themselves and provide the necessary locking. Multiple threads each manipulating separate objects need not synchronize.

class PKIXCertPathValidatorResult Unless otherwise specified, the methods defined in this class are not thread-safe. Multiple threads that need to access a single object concurrently should synchronize amongst themselves and provide the necessary locking. Multiple threads each manipulating separate objects need not synchronize.

class PKIXCertPathBuilderResult Unless otherwise specified, the methods defined in this class are not thread-safe. Multiple threads that need to access a single object concurrently should synchronize amongst themselves and provide the necessary locking. Multiple threads each manipulating separate objects need not synchronize.

class CertPathValidatorException Unless otherwise specified, the methods defined in this class are not thread-safe. Multiple threads that need to access a single object concurrently should synchronize amongst themselves and provide the necessary locking. Multiple threads each manipulating separate objects need not synchronize.

interface CertPathBuilderResult Unless otherwise specified, the methods defined in this interface are not thread-safe. Multiple threads that need to access a single object concurrently should synchronize amongst themselves and provide the necessary locking. Multiple threads each manipulating separate objects need not synchronize.

class X509CertSelector Unless otherwise specified, the methods defined in this class are not thread-safe. Multiple threads that need to access a single object concurrently should synchronize amongst themselves and provide the necessary locking. Multiple threads each manipulating separate objects need not synchronize

class X509CRLSelector Unless otherwise specified, the methods defined in this class are not thread-safe. Multiple threads that need to access a single object concurrently should synchronize amongst themselves and provide the necessary locking. Multiple threads each manipulating separate objects need not synchronize.

class CertPathValidator Unless otherwise documented by a specific provider, threads that need to access a single CertPathValidator instance concurrently should synchronize amongst themselves and provide the necessary locking. Multiple threads each manipulating a different CertPathValidator instance need not synchronize.

class CertPathValidator The static methods of this class are guaranteed to be thread-safe. Multiple threads may concurrently invoke the static methods defined in this class with no ill effects.

class CertPathBuilder The static methods of this class are guaranteed to be thread-safe. Multiple threads may concurrently invoke the static methods defined in this class with no ill effects.

class CertPathBuilder Unless otherwise documented by a specific provider, threads that need to access a single CertPathBuilder instance concurrently should synchronize amongst themselves and provide the necessary locking. Multiple threads each manipulating a different CertPathBuilder instance need not synchronize.

class PKIXParameters Unless otherwise specified, the methods defined in this class are not thread-safe. Multiple threads that need to access a single object concurrently should synchronize amongst themselves and provide the necessary locking. Multiple threads each manipulating separate objects need not synchronize.

abstract class PKIXCertPathChecker Unless otherwise specified, the methods defined in this class are not thread-safe. Multiple threads that need to access a single object concurrently should synchronize amongst themselves and provide the necessary locking. Multiple threads each manipulating separate objects need not synchronize.

CursorableLinkedList.iterator() Returns an iterator that does not support concurrent modification.

If the underlying list is modified while iterating using this iterator a ConcurrentModificationException will occur.

class CursorableLinkedList.Cursor An extended ListIterator that allows concurrent changes to the underlying list.

class LoopingIterator Concurrent modifications are not directly supported, and for most collection implementations will throw a ConcurrentModificationException.

class LoopingListIterator Concurrent modifications are not directly supported, and for most collection implementations will throw a ConcurrentModificationException.

class DefaultedMap Note that DefaultedMap is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization.

class FixedSizeMap Note that FixedSizeMap is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization.

class StaticBucketMap The iterators returned by the collection views of this class are not fail-fast. They will never raise a @link java.util.ConcurrentModificationException.

class PredicatedSortedMap Note that PredicatedSortedMap is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization.

class TransformedMap Note that TransformedMap is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization.

class LazyMap Note that LazyMap is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization.

class FixedSizeSortedMap Note that FixedSizeSortedMap is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization.

class ReferenceIdentityMap Note that ReferenceIdentityMap is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization.

class TransformedSortedMap Note that TransformedSortedMap is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization.

class MultiKeyMap Note that MultiKeyMap is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization. This class may throw exceptions when accessed by concurrent threads without synchronization.

class LazySortedMap Note that LazySortedMap is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization. T

class ReferenceMap Note that ReferenceMap is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization.

class HashedMap Note that HashedMap is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization.

class CaseInsensitiveMap Note that CaseInsensitiveMap is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization.

class IdentityMap Note that IdentityMap is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization.

class MultiValueMap Note that MultiValueMap is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization. This class may throw exceptions when accessed by concurrent threads without synchronization.

class ListOrderedMap Note that ListOrderedMap is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization

class Flat3Map Note that Flat3Map is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization.

class LinkedMap Note that LinkedMap is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization.

class PredicatedMap Note that PredicatedMap is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization. T

class CompositeMap Note that CompositeMap is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization.

class LRUMap Note that LRUMap is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization.

class TransformedMap Note that TransformedMap is not synchronized and is not thread-safe. If you wish to use this map from multiple threads concurrently, you must use appropriate synchronization.

class HashMap If multiple threads access a hash map concurrently, and at least one of the threads modifies the map structurally, it must be synchronized externally.

class LinkedHashMap If multiple threads access a linked hash map concurrently, and at least one of the threads modifies the map structurally, it must be synchronized externally.

class LinkedHashSet If multiple threads access a linked hash set concurrently, and at least one of the threads modifies the set, it must be synchronized externally.

class ArrayList If multiple threads access an ArrayList instance concurrently, and at least one of the threads modifies the list structurally, it must be synchronized externally.

class LinkedList If multiple threads access a linked list concurrently, and at least one of the threads modifies the list structurally, it must be synchronized externally.

class HashSet If multiple threads access a hash set concurrently, and at least one of the threads modifies the set, it must be synchronized externally.

class IdentityHashMap If multiple threads access an identity hash map concurrently, and at least one of the threads modifies the map structurally, it must be synchronized externally.

class TreeMap If multiple threads access a map concurrently, and at least one of the threads modifies the map structurally, it must be synchronized externally.

class TreeSet If multiple threads access a tree set concurrently, and at least one of the threads modifies the set, it must be synchronized externally.

(abstract) AbstractPreferences.syncSpi() Note that this method should not synchronize the preferences in any subnodes of this node.

class CursorableLinkedList Note that this implementation is not synchronized.

StaticBucketMap.putAll(map) This operation is not atomic and may have undesired effects.

abstract class EnumSet If multiple threads access an enum set concurrently, and at least one of the threads modifies the set, it should be synchronized externally.

class EnumMap If multiple threads access an enum map concurrently, and at least one of the threads modifies the map, it should be synchronized externally.

Pack200.newPacker() A multi-threaded application should either allocate multiple packer engines, or else serialize use of one engine with a lock.

Pack200.newUnpacker() A multi-threaded application should either allocate multiple unpacker engines, or else serialize use of one engine with a lock.

class PriorityQueue Note that this implementation is not synchronized. Multiple threads should not access a PriorityQueue instance concurrently if any of the threads modifies the queue.

SetUtils.synchronizedSet(set) You must manually synchronize on the returned buffer's iterator to avoid non-deterministic behavior:

class Semaphore Generally, semaphores used to control resource access should be initialized as fair, to ensure that no thread is starved out from accessing a resource.

interface Appendable Appendables are not necessarily safe for multithreaded access. Thread safety is the responsibility of classes that extend and implement this interface.

HierarchyEvent.HierarchyEvent(source, id, changed, changedParent) Note that passing in an invalid id results in unspecified behavior.

interface Condition An implementation is required to clearly document the semantics and guarantees provided by each of the waiting methods, and when an implementation does support interruption of thread suspension then it must obey the interruption semantics as defined in this interface.

(abstract) Condition.awaitUntil(deadline) In either case the implementation must ensure that the signal is redirected to another waiting thread, if there is one.

Runtime.addShutdownHook(hook) Shutdown hooks run at a delicate time in the life cycle of a virtual machine and should therefore be coded defensively. They should, in particular, be written to be thread-safe and to avoid deadlocks insofar as possible. They should also not rely blindly upon services that may have registered their own shutdown hooks and therefore may themselves in the process of shutting down. Attempts to use other thread-based services such as the AWT event-dispatch thread, for example, may lead to deadlocks.

abstract class CertStoreSpi The public methods of all CertStoreSpi objects must be thread-safe.

class PolicyQualifierInfo All PolicyQualifierInfo objects must be immutable and thread-safe.

class TrustAnchor All TrustAnchor objects must be immutable and thread-safe.

Thread.Thread(group, target, name, stackSize) The thread stack size necessary to perform a given computation will likely vary from one JRE implementation to another. In light of this variation, careful tuning of the stack size parameter may be required, and the tuning may need to be repeated for each JRE implementation on which an application is to run.

MapUtils.transformedMap(map, keyTransformer, valueTransformer) @param map the map to transform, must not be null, typically empty

abstract class AbstractQueuedSynchronizer Implementations of these methods must be internally thread-safe, and should in general be short and not block.

interface Lock When locking and unlocking occur in different scopes, care must be taken to ensure that all code that is executed while the lock is held is protected by try-finally or try-catch to ensure that the lock is released when necessary.

(abstract) Instrumentation.removeTransformer(transformer) Due to the multi-threaded nature of class loading, it is possible for a transformer to receive calls after it has been removed. Transformers should be written defensively to expect this situation.

class CountDownLatch When threads must repeatedly count down in this way, instead use a @link CyclicBarrier.

class ResourceBundle.Control A ResourceBundle.Control instance must be thread-safe if it's simultaneously used by multiple threads.

AbstractQueuedLongSynchronizer.tryAcquire(arg) This method should query if the state of the object permits it to be acquired in the exclusive mode, and if so to acquire it.

interface Lock The absence of block-structured locking removes the automatic release of locks that occurs with synchronized methods and statements. In most cases, the following idiom should be used:

interface Lock All Lock implementations must enforce the same memory synchronization semantics as provided by the built-in monitor lock, as described in The Java Language Specification, Third Edition (17.4 Memory Model)

DeferredContentProvider.setSortOrder(sortOrder) @param sortOrder new sort order. The comparator must be able to support being used in a background thread.

FontRegistry.getBold(symbolicName) This method should only be called from the UI thread. If you are not on the UI thread then wrap the call with a PlatformUI.getWorkbench().getDisplay().synchExec() in order to guarantee the correct result. Failure to do this may result in an @link SWTException being thrown.

(abstract) IPreferenceStore.addPropertyChangeListener(listener) Likewise any application that updates an IPreferenceStore from a Thread other than the UI Thread should be aware of any listeners that require an update in the UI Thread.

class ReentrantLock It is recommended practice to always immediately follow a call to lock with a try block, most typically in a before/after construction

class PipedOutputStream ttempting to use both objects from a single thread is not recommended as it may deadlock the thread.

interface IContentFormatterExtension Note that in context based mode the content formatter is fully reentrant, but not thread-safe.

(abstract) ICompletionProposalExtension5.getAdditionalProposalInfo(monitor) This method may be called on a non-UI thread.

(abstract) Lock.tryLock(time, unit) The ability to interrupt a lock acquisition in some implementations may not be possible, and if possible may be an expensive operation. The programmer should be aware that this may be the case.

abstract class SafeRunnable This may be executed on any thread.

VerticalRuler.update() Thread-safe implementation. Can be called from any thread.

FontRegistry.getItalic(symbolicName) This method should only be called from the UI thread. If you are not on the UI thread then wrap the call with a PlatformUI.getWorkbench().getDisplay().synchExec() in order to guarantee the correct result.

interface Condition It is recommended that to avoid confusion you never use Condition instances in this way, except perhaps within their own implementation.

abstract class Format Formats are generally not synchronized. It is recommended to create separate format instances for each thread.

class PipedInputStream Attempting to use both objects from a single thread is not recommended, as it may deadlock the thread.

ConcurrentSkipListMap.size() Additionally, it is possible for the size to change during execution of this method, in which case the returned result will be inaccurate.

ReentrantReadWriteLock.WriteLock.newCondition() If this write lock is not held when any @link Condition method is called then an @link IllegalMonitorStateException is thrown. (Read locks are held independently of write locks, so are not checked or affected. However it is essentially always an error to invoke a condition waiting method when the current thread has also acquired read locks, since other threads that could unblock it will not be able to acquire the write lock.)

interface ConcurrentMap Memory consistency effects: As with other concurrent collections, actions in a thread prior to placing an object into a ConcurrentMap as a key or value happen-before actions subsequent to the access or removal of that object from the ConcurrentMap in another thread.

abstract class PermissionCollection Furthermore, Enumerations returned via the elements method are not fail-fast. Modifications to a collection should not be performed while enumerating over that collection.

abstract class AbstractQueuedSynchronizer Throughput and scalability are generally highest for the default barging (also known as greedy, renouncement, and convoy-avoidance) strategy. While this is not guaranteed to be fair or starvation-free, earlier queued threads are allowed to recontend before later queued threads, and each recontention has an unbiased chance to succeed against incoming threads. Also, while acquires do not "spin" in the usual sense, they may perform multiple invocations of tryAcquire interspersed with other computations before blocking. This gives most of the benefits of spins when exclusive synchronization is only briefly held, without most of the liabilities when it isn't. If so desired, you can augment this by preceding calls to acquire methods with "fast-path" checks, possibly prechecking @link #hasContended and/or @link #hasQueuedThreads to only do so if the synchronizer is likely not to be contended.

(abstract) IContentFormatterExtension.format(document, context) This method is fully reentrant, but not thread-safe.

(abstract) Preferences.exportSubtree(os) If the preferences or nodes in the subtree rooted at this node are modified concurrently with an invocation of this method, the exported preferences comprise a "fuzzy snapshot" of the subtree; some of the concurrent modifications may be reflected in the exported data while others may not.

(abstract) Preferences.exportNode(os) . If the preferences at this node are modified concurrently with an invocation of this method, the exported preferences comprise a "fuzzy snapshot" of the preferences contained in the node; some of the concurrent modifications may be reflected in the exported data while others may not.

interface IConcurrentModel The model is allowed to send back notifications to its listeners in any thread, and the listeners must not assume that the notifications will arrive in the UI thread.

class DataInputStream DataInputStream is not necessarily safe for multithreaded access.

6 Miscellaneous Directive

ActivationGroup.createGroup(id, desc, incarnation) Note that if your application creates its own custom activation group, a security manager must be set for that group

ActivationGroup.createGroup(id, desc, incarnation) If your application needs to set a different security manager, you must ensure that the policy file specified by the group's ActivationGroupDesc grants the group the necessary permissions to set a new security manager

interface WebMethod The associated method must be public and its parameters return value, and exceptions must follow the rules defined in JAX-RPC 1.1, section

(abstract) WebService.wsdlLocation() The JSR-181 tool must provide feedback if the service implementation bean is inconsistent with the portType and bindings declared in this WSDL.

class MailcapCommandMap Mailcap files must conform to the mailcap file specification

(abstract) Lock.lock() The circumstances and the exception type must be documented by that Lock implementation.

(abstract) Condition.awaitNanos(nanosTimeout) Typically, an exception will be thrown (such as @link IllegalMonitorStateException) and the implementation must document that fact.

interface Condition An implementation is required to clearly document the semantics and guarantees provided by each of the waiting methods

interface Condition If an implementation provides such specialized semantics then the implementation must document those semantics.

(abstract) Lock.newCondition() The exact operation of the @link Condition instance depends on the Lock implementation and must be documented by that implementation.

(abstract) Condition.awaitUninterruptibly() Typically, an exception will be thrown (such as @link IllegalMonitorStateException) and the implementation must document that fact.

(abstract) Condition.awaitUntil(deadline) Typically, an exception will be thrown (such as @link IllegalMonitorStateException) and the implementation must document that fact.

PopupDialog.showDialogMenu() Clients may call this method in order to trigger the menu via keystrokes or other gestures

class ProgressIndicator For indeterminate progress, we don't have to know the total amount of work in advance and no worked method needs to be called.

PopupDialog.saveDialogBounds(shell) Clients may also call this method to persist the bounds at times other than closing the dialog.

MessageDialogWithToggle.getToggleState() This can be called even after the dialog is closed.

class ThreadDeath An application should catch instances of this class only if it must clean up after being terminated asynchronously.

URLConnection.getContent(classes) The instanceof operator should be used to determine the specific kind of object returned.

abstract class JarURLConnection Users should cast the generic URLConnection to a JarURLConnection when they know that the URL they created is a JAR URL, and they need JAR-specific functionality. For example:

URL url = new URL("jar:file:/home/duke/duke.jar!/"); JarURLConnection jarConnection = (JarURLConnection)url.openConnection(); Manifest manifest = jarConnection.getManifest();

URLConnection.getContent() @return the object fetched. The instanceof operator should be used to determine the specific kind of object returned.

abstract class ActivationGroup If your application requires specific properties to be set when objects are activated in the group, the application should create a special Properties object containing these properties, then create an ActivationGroupDesc with the Properties object, and use ActivationGroup.createGroup before creating any ActivationDescs (before the default ActivationGroupDesc is created).

ObjectStreamField.ObjectStreamField(name, type) This field should be documented with a serialField tag.

(abstract) Externalizable.writeExternal(out) @serialData Overriding methods should use this tag to describe the data layout of this Externalizable object.

class PipedInputStream A piped input stream should be connected to a piped output stream; the piped input stream then provides whatever data bytes are written to the piped output stream.

class Console Security note: If an application needs to read a password or other secure data, it should use @link #readPassword() or @link #readPassword(String, Object...) and manually zero the returned character array after processing to minimize the lifetime of sensitive data in memory.

interface Serializable Classes that need to designate a replacement when an instance of it is read from the stream should implement this special method with the exact signature.

ANY-ACCESS-MODIFIER Object readResolve() throws ObjectStreamException;

interface Serializable Serializable classes that need to designate an alternative object to be used when writing an object to the stream should implement this special method with the exact signature:

ANY-ACCESS-MODIFIER Object writeReplace() throws ObjectStreamException;

class Error An Error is a subclass of Throwable that indicates serious problems that a reasonable application should not try to catch.

interface SuppressWarnings As a matter of style, programmers should always use this annotation on the most deeply nested element where it is effective.

interface Appendable The Appendable interface must be implemented by any class whose instances are intended to receive formatted output from a @link java.util.Formatter.

Thread.stop() An application should not normally try to catch ThreadDeath unless it must do some extraordinary cleanup operation (note that the throwing of ThreadDeath causes finally clauses of try statements to be executed before the thread officially dies). If a catch clause catches a ThreadDeath object, it is important to rethrow the object so that the thread actually dies.

interface Array To create a custom mapping, a programmer must do two things:

* create a class that implements the @link SQLData interface for the UDT to be custom mapped. * make an entry in a type map that contains o the fully-qualified SQL type name of the UDT o the Class object for the class implementing SQLData

interface DatabaseMetaData Additional columns beyond the columns defined to be returned by the ResultSet object for a given method can be defined by the JDBC driver vendor and must be accessed by their column label.

interface Connection If auto-commit mode has been disabled, the method commit must be called explicitly in order to commit changes; otherwise, database changes will not be saved.

class RuleBasedCollator First, if the strings to be collated contain combining sequences that may not be in canonical order, you should set the collator to CANONICAL_DECOMPOSITION or FULL_DECOMPOSITION to enable sorting of combining sequences.

Signer.Signer() This constructor should only be used for serialization.

IdentityScope.IdentityScope() This constructor is used for serialization only and should not be used by subclasses.

(abstract) PrivilegedExceptionAction.run() Each class that implements PrivilegedExceptionAction should document what (if anything) this value represents.

(abstract) PrivilegedAction.run() Each class that implements PrivilegedAction should document what (if anything) this value represents.

(abstract) CertStoreParameters.clone() Each subclass should state the precise behavior of this method so that users and developers know what to expect.

class DigestInputStream Note that digest objects can compute only one digest (see @link MessageDigest), so that in order to compute intermediate digests, a caller should retain a handle onto the digest object, and clone it for each digest to be computed, leaving the orginal digest untouched.

class ConstantFactory No check is made that the object is immutable. In general, only immutable objects should use the constant factory.

class ConstantTransformer No check is made that the object is immutable. In general, only immutable objects should use the constant factory.

class Permissions Enumerations returned via the elements method are not fail-fast. Modifications to a collection should not be performed while enumerating over that collection.

abstract class KeyFactorySpi A provider should document all the key specifications supported by its key factory

(abstract) PreparedStatement.setObject(parameterIndex, x) Not all databases allow for a non-typed Null to be sent to the backend. For maximum portability, the setNull or the setObject(int parameterIndex, Object x, int sqlType) method should be used instead of setObject(int parameterIndex, Object x).

(abstract) CallableStatement.setObject(parameterName, x) Not all databases allow for a non-typed Null to be sent to the backend. For maximum portability, the setNull or the setObject(String parameterName, Object x, int sqlType) method should be used instead of setObject(String parameterName, Object x).

(abstract) CallableStatement.registerOutParameter(parameterName, sqlType, typeName) To be portable, however, applications should always provide these values for user-named and REF parameters.

SecurityManager.checkTopLevelWindow(window) In this case, the caller can still decide to show the window, but the window should include some sort of visual warning.

interface Runnable This is important because classes should not be subclassed unless the programmer intends on modifying or enhancing the fundamental behavior of the class.

(abstract) SuppressWarnings.value() Compiler vendors should document the warning names they support in conjunction with this annotation type.

ThreadGroup.enumerate(list, recurse) If it is critical to obtain every active thread in this thread group, the caller should verify that the returned int value is strictly less than the length of list.

ThreadGroup.enumerate(list) f it is critical to obtain every active subgroup in this thread group, the caller should verify that the returned int value is strictly less than the length of list.

ThreadGroup.enumerate(list, recurse) If it is critical to obtain every active subgroup in this thread group, the caller should verify that the returned int value is strictly less than the length of list.

Object.wait() As in the one argument version, interrupts and spurious wakeups are possible, and this method should always be used in a loop:

synchronized (obj) while (¡condition does not hold¿) obj.wait(); ... // Perform action appropriate to condition

(abstract) MemoryPoolMXBean.getUsage() An implementation should document when this is the case.

(abstract) Lock.unlock() Any restrictions and the exception type must be documented by that Lock implementation.

(abstract) Lock.lockInterruptibly() The circumstances and the exception type must be documented by that Lock implementation

(abstract) Lock.tryLock(time, unit) The circumstances and the exception type must be documented by that Lock implementation.

Boolean.valueOf(b) If a new Boolean instance is not required, this method should generally be used in preference to the constructor @link #Boolean(boolean), as this method is likely to yield significantly better space and time performance.

AbstractTreeViewer.getChild(widget, index) Note that the default implementation is not very efficient and should be overridden if this class is implemented.

Object.wait(timeout) While this will rarely occur in practice, applications must guard against it by testing for the condition that should have caused the thread to be awakened, and continuing to wait if the condition is not satisfied. In other words, waits should always occur in loops, like this one:

synchronized (obj) while (¡condition does not hold¿) obj.wait(timeout); ... // Perform action appropriate to condition

Object.wait(timeout, nanos) As in the one argument version, interrupts and spurious wakeups are possible, and this method should always be used in a loop:

synchronized (obj) while (¡condition does not hold¿) obj.wait(timeout, nanos); ... // Perform action appropriate to condition

interface MemoryPoolMXBean For some memory pools, for example, when objects are not packed contiguously, this method may be an expensive operation that requires some computation to determine the current memory usage. An implementation should document when this is the case.

System.getenv(name) Environment variables should be used when a global effect is desired, or when an external system interface requires an environment variable (such as PATH).

(abstract) Comparable.compareTo(o) Generally speaking, any class that implements the Comparable interface and violates this condition should clearly indicate this fact.

ManagementFactory.newPlatformMXBeanProxy(connection, mxbeanName, mxbeanInterface) An application remotely accesses the platform MXBeans using proxy should prepare to catch IOException as if accessing with the MBeanServerConnector interface.

ManagementFactory.newPlatformMXBeanProxy(connection, mxbeanName, mxbeanInterface) When a client application is designed to remotely access MXBeans for a running virtual machine whose version is different than the version on which the application is running, it should prepare to catch @link java.io.InvalidObjectException InvalidObjectException which is thrown when an MXBean proxy receives a name of an enum constant which is missing in the enum class loaded in the client application.

class Throwable The specification for the persistent collection should indicate that it is capable of throwing such exceptions.

class StringBuffer The StringBuilder class should generally be used in preference to this one, as it supports all of the same operations but it is faster, as it performs no synchronization.

(abstract) IRunnableContext.run(fork, cancelable, runnable) Implementers should document whether the runnable is run synchronously (blocking) or asynchronously (non-blocking), or if no assumption can be made about the blocking behaviour.

(abstract) IInformationProviderExtension.getInformation2(textViewer, subject) Implementers should ignore the text returned by @link IInformationProvider#getInformation(ITex IRegion).

(abstract) IInformationControlExtension5.getInformationPresenterControlCreator() This method may be called frequently, so implementors should ensure it returns quickly, e.g. by caching the returned creator.

class Flat3Map This map is designed for performance and can outstrip HashMap. It also has good garbage collection characteristics.

* Optimised for operation at size 3 or less. * Still works well once size 3 exceeded. * Gets at size 3 or less are about 0-10% faster than HashMap, * Puts at size 3 or less are over 4 times faster than HashMap. * Performance 5% slower than HashMap once size 3 exceeded once.

class TreeList This list implementation utilises a tree structure internally to ensure that all insertions and removals are O(log n). This provides much faster performance than both an ArrayList and a LinkedList where elements are inserted and removed repeatedly from anywhere in the list.

class TreeList ArrayList is a good general purpose list implementation. It is faster than TreeList for most operations except inserting and removing in the middle of the list. ArrayList also uses less memory as TreeList uses one object per entry.

class IteratorChain Under many circumstances, linking Iterators together in this manner is more efficient (and convenient) than reading out the contents of each Iterator into a List and creating a new Iterator.

class ObjectGraphIterator Under many circumstances, linking Iterators together in this manner is more efficient (and convenient) than using nested for loops to extract a list.

IteratorUtils.objectGraphIterator(root, transformer) Under many circumstances, linking Iterators together in this manner is more efficient (and convenient) than using nested for loops to extract a list.

interface Lock If an implementation provides such specialized semantics then the implementation must document those semantics.

(abstract) Condition.await() Typically, an exception will be thrown (such as @link IllegalMonitorStateException) and the implementation must document that fact.

(abstract) List.add(e) List classes should clearly specify in their documentation any restrictions on what elements may be added.

(abstract) Lock.tryLock(time, unit) An implementation should document when this is the case.

interface Pack200.Packer In order both to sign and to pack a JAR, you must first pack and unpack the JAR to "normalize" it, then compute signatures on the unpacked JAR elements, and finally repack the signed JAR.

Level.SEVERE In general SEVERE messages should describe events that are of considerable importance and which will prevent normal program execution. They should be reasonably intelligible to end users and to system administrators.

class IteratorUtils he class remained source and test compatible, so if you can recompile all your classes and dependencies everything is OK.

LoopingListIterator.previous() Note that in this case, traversal to find that element takes linear time.

MultiKey.MultiKey(key1, key2) If they are not then they must not be changed after adding to the MultiKey.

MultiKey.MultiKey(key1, key2, key3) If they are not then they must not be changed after adding to the MultiKey.

StaticBucketMap.atomic(r) Implementation note: This method requires a lot of time and a ton of stack space. Essentially a recursive algorithm is used to enter each bucket's monitor. If you have twenty thousand buckets in your map, then the recursive method will be invoked twenty thousand times. You have been warned.

ArrayIterator.setArray(array) Although this can be used to reset iteration, the @link #reset() method is a more effective choice.

LockSupport.parkUntil(blocker, deadline) Callers should re-check the conditions which caused the thread to park in the first place.

(abstract) Lock.lockInterruptibly() An implementation should document when this is the case.

LockSupport.park(blocker) Callers should re-check the conditions which caused the thread to park in the first place.

LockSupport.park() Callers should re-check the conditions which caused the thread to park in the first place.

LockSupport.parkNanos(nanos) Callers should re-check the conditions which caused the thread to park in the first place.

LockSupport.parkUntil(deadline) Callers should re-check the conditions which caused the thread to park in the first place.

LockSupport.parkNanos(blocker, nanos) Callers should re-check the conditions which caused the thread to park in the first place

(abstract) Set.add(e) Individual set implementations should clearly document any restrictions on the elements that they may contain.

(abstract) Collection.add(e) Collection classes should clearly specify in their documentation any restrictions on what elements may be added.

AbstractList.add(e) List classes should clearly specify in their documentation any restrictions on what elements may be added.

(abstract) Comparator.compare(o1, o2) Generally speaking, any comparator that violates this condition should clearly indicate this fact.

interface OrderedBidiMap Implementations should allow a value to be looked up from a key and a key to be looked up from a value with equal performance.

interface SortedBidiMap Implementations should allow a value to be looked up from a key and a key to be looked up from a value with equal performance.

interface BidiMap Implementations should allow a value to be looked up from a key and a key to be looked up from a value with equal performance.

class StaticBucketMap Finally, unlike @link java.util.HashMap-style implementations, this class never rehashes the map. The number of buckets is fixed at construction time and never altered. Performance may degrade if you do not allocate enough buckets upfront.

StaticBucketMap.StaticBucketMap(numBuckets) he number of buckets is inversely proportional to the chances for thread contention. The fewer buckets, the more chances for thread contention. The more buckets the fewer chances for thread contention.

class StaticBucketMap The @link #atomic(Runnable) method is provided to allow atomic iterations and bulk operations; however, overuse of @link #atomic(Runnable) atomic will basically result in a map that's slower than an ordinary synchronized @link java.util.HashMap.

Level.WARNING In general WARNING messages should describe events that will be of interest to end users or system managers, or which indicate potential problems.

Level.INFO So the INFO level should only be used for reasonably significant messages that will make sense to end users and system admins.

Level.FINE In general the FINE level should be used for information that will be broadly interesting to developers who do not have a specialized interest in the specific subsystem.

interface Pack200.Packer Both packing steps should use precisely the same options, and the segment limit may also need to be set to "-1", to prevent accidental variation of segment boundaries as class file sizes change slightly.

Preferences.userNodeForPackage(c) Valuable data should not be stored at this node as it is shared by all programs that use it.

class TransformedSortedSet The add methods are affected by this class. Thus objects must be removed or searched for using their transformed form. For example, if the transformation converts Strings to Integers, you must use the Integer form to remove objects.

class TransformedSet The add methods are affected by this class. Thus objects must be removed or searched for using their transformed form. For example, if the transformation converts Strings to Integers, you must use the Integer form to remove objects.

interface List From a performance standpoint, these methods should be used with caution. In many implementations they will perform costly linear searches.

class TransformedBag The add methods are affected by this class. Thus objects must be removed or searched for using their transformed form. For example, if the transformation converts Strings to Integers, you must use the Integer form to remove objects.

class TransformedSortedBag The add methods are affected by this class. Thus objects must be removed or searched for using their transformed form. For example, if the transformation converts Strings to Integers, you must use the Integer form to remove objects.

class TransformedList The add and set methods are affected by this class. Thus objects must be removed or searched for using their transformed form. For example, if the transformation converts Strings to Integers, you must use the Integer form to remove objects.

class TransformedCollection The add methods are affected by this class. Thus objects must be removed or searched for using their transformed form. For example, if the transformation converts Strings to Integers, you must use the Integer form to remove objects.

class TransformedBuffer The add methods are affected by this class. Thus objects must be removed or searched for using their transformed form. For example, if the transformation converts Strings to Integers, you must use the Integer form to remove objects.

Preferences.systemNodeForPackage(c) Valuable data should not be stored at this node as it is shared by all programs that use it.

class LogRecord Therefore, if a logging Handler wants to pass off a LogRecord to another thread, or to transmit it over RMI, and if it wishes to subsequently obtain method name or class name information it should call one of getSourceClassName or getSourceMethodName to force the values to be filled in.

interface Queue Every Queue implementation must specify its ordering properties.

(abstract) CallableStatement.registerOutParameter(parameterIndex, sqlType, typeName) To be portable, however, applications should always provide these values for user-defined and REF parameters.

(abstract) SuppressWarnings.value() Compilers must ignore any warning names they do not recognize.

class DriverManager JDBC 4.0 Drivers must include the file META-INF/services/java.sql.Driver.

interface Driver It is strongly recommended that each Driver class should be small and standalone so that the Driver class can be loaded and queried without bringing in vast quantities of supporting code.

(abstract) MemoryPoolMXBean.getUsage() For some memory pools, this method may be an expensive operation that requires some computation to determine the estimate.

interface Override If a method is annotated with this annotation type but does not override a superclass method, compilers are required to generate an error message.

abstract class Process There is no requirement that a process represented by a Process object execute asynchronously or concurrently with respect to the Java process that owns the Process object.

Runtime.totalMemory() Note that the amount of memory required to hold an object of any given type may be implementation-dependent.

class RuntimeException A method is not required to declare in its throws clause any subclasses of RuntimeException that might be thrown during the execution of the method but not caught.

(abstract) Driver.jdbcCompliant() A driver may only report true here if it passes the JDBC compliance tests; otherwise it is required to return false.

JDBC compliance requires full support for the JDBC API and full support for SQL 92 Entry Level. It is expected that JDBC compliant drivers will be available for all the major commercial databases.

class ConcurrentSkipListSet Beware that, unlike in most collections, the size method is not a constant-time operation. Because of the asynchronous nature of these sets, determining the current number of elements requires a traversal of the elements.

ConcurrentSkipListSet.size() Beware that, unlike in most collections, this method is NOT a constant-time operation. Because of the asynchronous nature of these sets, determining the current number of elements requires traversing them all to count the

class ConcurrentSkipListMap Beware that, unlike in most collections, the size method is not a constant-time operation. Because of the asynchronous nature of these maps, determining the current number of elements requires a traversal of the element

ConcurrentSkipListMap.size() Beware that, unlike in most collections, this method is NOT a constant-time operation. Because of the asynchronous nature of these maps, determining the current number of elements requires traversing them all to count them.

ConcurrentLinkedQueue.size() Beware that, unlike in most collections, this method is NOT a constant-time operation. Because of the asynchronous nature of these queues, determining the current number of elements requires an O(n) traversal.

ConcurrentHashMap.containsValue(value) Note: This method requires a full internal traversal of the hash table, and so is much slower than method containsKey.

TreeMap.containsValue(value) This operation will probably require time linear in the map size for most implementations.

AbstractList.removeRange(fromIndex, toIndex) Note: if ListIterator.remove requires linear time, this implementation requires quadratic time.

(abstract) Map.containsValue(value) This operation will probably require time linear in the map size for most implementations of the Map interface.

AbstractMap.containsValue(value) Note that this implementation requires linear time in the size of the map.

AbstractMap.containsKey(key) Note that this implementation requires linear time in the size of the map;

AbstractMap.get(key) Note that this implementation requires linear time in the size of the map;

AbstractMap.remove(key) Note that this implementation requires linear time in the size of the map;

Socket.setReuseAddress(on) When a TCP connection is closed the connection may remain in a timeout state for a period of time after the connection is closed (typically known as the TIME_WAIT state or 2MSL wait state). For applications using a well known socket address or port it may not be possible to bind a socket to the required SocketAddress if there is a connection in the timeout state involving the socket address or

ServerSocket.setReuseAddress(on) When a TCP connection is closed the connection may remain in a timeout state for a period of time after the connection is closed (typically known as the TIME_WAIT state or 2MSL wait state). For applications using a well known socket address or port it may not be possible to bind a socket to the required SocketAddress if there is a connection in the timeout state involving the socket address or port.

interface RandomAccess For example, some List implementations provide asymptotically linear access times if they get huge, but constant access times in practice. Such a List implementation should generally implement this interface.

abstract class Dictionary As a rule, the equals method should be used by implementations of this class to decide if two keys are the same.

class ConcurrentLinkedQueue Beware that, unlike in most collections, the size method is NOT a constant-time operation. Because of the asynchronous nature of these queues, determining the current number of elements requires a traversal of the elements.

TableViewer.setSelection(selection, reveal) Sets a new selection for this viewer and optionally makes it visible. The TableViewer implementation of this method is inefficient for the ILazyContentProvider as lookup is done by indices rather than elements and may require population of the entire table in worse case.

Use Table#setSelection(int[] indices) and Table#showSelection() if you wish to set selection more efficiently when using a ILazyContentProvider.

class MessageDialogWithToggle It is strongly recommended that a cancel option be provided, so that the user has the option of making the decision at a later point in time.

interface IDialogConstants IDialogConstants is the interface for common dialog strings and ids used throughout JFace. It is recommended that you use these labels and ids whereever for consistency with the JFace dialogs.

ManagementFactory.getPlatformMBeanServer() It is recommended that this platform MBeanServer also be used to register other application managed beans besides the platform MXBeans.

ThreadGroup.activeCount() Due to the inherently imprecise nature of the result, it is recommended that this method only be used for informational purposes.

ThreadGroup.activeGroupCount() Due to the inherently imprecise nature of the result, it is recommended that this method only be used for informational purposes.

Thread.Thread(group, target, name, stackSize) Implementation note: Java platform implementers are encouraged to document their implementation's behavior with respect to the stackSize parameter.

class Math Code generators are encouraged to use platform-specific native libraries or microprocessor instructions, where available, to provide higher-performance implementations of Math methods. Such higher-performance implementations still must conform to the specification for Math.

interface Target It is a compile-time error for a single ElementType constant to appear more than once in a Target annotation.

interface IQuickFixableAnnotation Caching the state is important to improve overall performance as calling @link org.eclipse.jface.text.quickassist.IQuickAssistAssistant#canFix(Anno can be expensive.

LazySortedCollection.enableDebug Disables randomization and enables additional runtime error checking. Severely degrades performance if set to true. Intended for use in test suites only.

Socket.setReceiveBufferSize(size) Increasing the receive buffer size can increase the performance of network I/O for high-volume connection, while decreasing it can help reduce the backlog of incoming data.

Pattern.CANON_EQ Specifying this flag may impose a performance penalty.

RemoteObject.getRef() Note: The object returned from this method may be an instance of an implementation-specific class. The RemoteObject class ensures serialization portability of its instances' remote references through the behavior of its custom writeObject and readObject methods. An instance of RemoteRef should not be serialized outside of its RemoteObject wrapper instance or the result may be unportable.