SI External Interfaces

Package
com.ibm.retail.AEF.action

Provides `Action`classes to perform POS functions required by the `POSAutomationProvider`.

com.ibm.retail.AEF.action

Interface AEFAction

All Known Implementing Classes:
SimpleKeySequenceActionImpl

public interface **AEFAction**

AEFAction is an interface representing actions which may be performed within the AEF.

Method Summary

`java.lang.Object`	`performAction()` Perform the action represented by the ActionRequest and return an ActionResult.

Methods

performAction

```
public java.lang.Object performAction()
                                throws AEFException
```

Perform the action represented by the ActionRequest and return an ActionResult.

Returns:

Object Any required return value (may be null).

Exceptions:

`AEFException` -
Because of the nature of this method, just about any error code could be returned.

com.ibm.retail.AEF.action

Class SimpleKeySequenceActionImpl

```
java.lang.Object
   |
   +-com.ibm.retail.AEF.action.SimpleKeySequenceActionImpl
```

All Implemented interfaces:
AEFAction

---

public class **SimpleKeySequenceActionImpl**
extends java.lang.Object
implements AEFAction

SimpleKeySequenceActionImpl is a class which causes a simple key sequence (data and fcodes only) to be sent to the POS application.

Field Summary

java.lang.String	sessionID
com.ibm.retail.AEF.util.AEFMessage	tempAEFMessage

Constructor Summary

SimpleKeySequenceActionImpl(ActionRequest request) Constructor

Method Summary

java.lang.Object	performAction() Perform the action represented by the ActionRequest and return an ActionResult.
java.util.ArrayList	retrieveSubstitutionArguments(ActionRequest request) Retrieves the variable substitution arguments which will be substituted into the key sequence string.
java.util.ArrayList	retrieveSubstitutionArguments(java.lang.String sequenceID,ActionRequest sequenceID) Retrieves the variable substitution arguments which will be substituted into the key sequence string.

Methods inherited from : class `java.lang.Object`
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

---

Fields

(continued from last page)

sessionID

`protected java.lang.String` **`sessionID`**

tempAEFMessage

`protected com.ibm.retail.AEF.util.AEFMessage` **`tempAEFMessage`**

Constructors

SimpleKeySequenceActionImpl

`public` **`SimpleKeySequenceActionImpl`**`(ActionRequest request)`

Constructor

Parameters:

`request -`
The ActionRequest which contains a Map of arguments.

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.INVALID_ACTION_REQUEST

Methods

retrieveSubstitutionArguments

`public java.util.ArrayList` **`retrieveSubstitutionArguments`**`(ActionRequest request)`

Retrieves the variable substitution arguments which will be substituted into the key sequence string. The substitutions will occur in sequence wherever {0}, {1}, etc. appear.

Parameters:

`request -`
The ActionRequest which contains the classname and arguments.

Returns:

ArrayList An array of string substitution arguments.

retrieveSubstitutionArguments

`public java.util.ArrayList` **`retrieveSubstitutionArguments`**`(java.lang.String sequenceID, ActionRequest request)`

Retrieves the variable substitution arguments which will be substituted into the key sequence string. The substitutions will occur in sequence wherever {0}, {1}, etc. appear.

Parameters:

`String -`
The id of the sequence. The arguments must appear in the ActionRequest arguments as "%sequenceid.0", "%sequenceid.1" ...
`request -`
The ActionRequest which contains the classname and arguments.

Returns:

(continued from last page)

ArrayList An array of string substitution arguments.

performAction

```
public java.lang.Object performAction()
                                throws AEFException
```

Perform the action represented by the ActionRequest and return an ActionResult.

Parameters:

`request -`
The ActionRequest which contains the classname and arguments.

Returns:

Object Null is always returned.

Exceptions:

`AEFException`
`com.ibm.retail.AEF.util.AEFException -`
Among the possible AEFException error codes are:
AEFConst.CONFIG_ERROR, AEFConst.INVALID_KEYSEQUENCE_EXPRESSION
AEFConst.CONFIG_ERROR, AEFConst.NONNUMERIC_FUNCTIONCODE
SYSTEM_BUSY
INPUT_NOT_ALLOWED

Package
com.ibm.retail.AEF.automation

Provides interface for performing functions on real and virtual terminal POS terminals.

The automation package includes the `POSAutomationProvider`interface which allows applications to perform functions on a real or virtual POS terminal. The interface provided by `POSAutomationProvider`includes the commonly used functions associated with transactions at a POS terminal:
- Logging on to a terminal
- Starting a transaction
- Scanning a customer loyalty card
- Adding or removing items from the transaction
- Tendering
- Logging off the terminal

com.ibm.retail.AEF.automation

Class AbstractCondition

```
java.lang.Object
  |
  +-com.ibm.retail.AEF.automation.AbstractCondition
```

All Implemented interfaces:
Condition

Direct Known Subclasses:
OrCondition, AndThenCondition, AndCondition, AbstractPropertyCondition

---

public abstract class **AbstractCondition**
extends java.lang.Object
implements Condition

AbstractCondition is a base class for objects which are used in conjuction with a ConditionLock to block a calling thread based on a set of conditions.

Field Summary

static java.lang.Object	counterLock
com.ibm.retail.AEF.data.POSDataProvider	dataProvider
boolean	eligible
int	id
static int	idCounter
int	index
java.util.Collection	listeners
com.ibm.retail.AEF.thread.ConditionLock	lock
com.ibm.retail.AEF.thread.ConditionLockManager	manager
com.ibm.retail.AEF.session.AEFSession	session

Constructor Summary

AbstractCondition()

Method Summary

void	`addEvaluateListener(EvaluateConditionListener listener)` Add a EvaluateConditionListener which will be notified whenever the condition is evaluated.
int	`getID()` Gets the unique id of this condition.
int	`getIndex()` Gets the vector index of this condition.
int	`getNextID()` Gets the next available id.
POSDataProvider	`getPOSDataProvider()` Returns the POSDataProvider associated with this condition.
AEFSession	`getSession()` Gets the AEFSession associated with this condition.
java.lang.String	`getTerminalNumber()` Gets the terminal number associated with the session for this condition.
boolean	`isEligible()` Indicates whether the condition is eligible for evaluation.
void	`notifyEvaluateConditionListeners(Condition cond,boolean cond)` Notifies EvaluateConditionListeners
void	`removeEvaluateListener(EvaluateConditionListener listener)` Remove a EvaluateConditionListener
void	`setConditionLock(ConditionLock lock)` Sets the condition lock associated with this condition.
void	`setEligible(boolean flag)` Sets the condition eligible for evaluation.
void	`setID(int id)` Sets the unique id of this condition (for use in trace &amp.
void	`setIndex(int index)` Sets the index of this condition (for use within a vector).
void	`setLockManager(ConditionLockManager manager)` Sets the condition lock manager associated with this condition.
void	`setPOSDataProvider(POSDataProvider dataProvider)` Sets the POSDataProvider associated with this condition.
void	`setSession(AEFSession session)` Sets the AEFSession associated with this condition.

Methods inherited from : class `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
wait
```

Fields

index

protected int **index**

lock

protected com.ibm.retail.AEF.thread.ConditionLock **lock**

eligible

protected boolean **eligible**

dataProvider

protected com.ibm.retail.AEF.data.POSDataProvider **dataProvider**

idCounter

public static int **idCounter**

counterLock

public static java.lang.Object **counterLock**

id

public int **id**

session

public com.ibm.retail.AEF.session.AEFSession **session**

manager

public com.ibm.retail.AEF.thread.ConditionLockManager **manager**

(continued from last page)

listeners

`protected java.util.Collection listeners`

Constructors

AbstractCondition

`public AbstractCondition()`

Methods

setEligible

`public void setEligible(boolean flag)`

Sets the condition eligible for evaluation.

Parameters:

`flag -`
Set true to set the condition eligible for evaluation.

---

isEligible

`public boolean isEligible()`

Indicates whether the condition is eligible for evaluation.

Returns:

boolean True if the condition is currently eligible for evaluation.

---

setConditionLock

`public void setConditionLock(ConditionLock lock)`

Sets the condition lock associated with this condition.

Parameters:

`lock -`
An optional condition lock which will be signalled when the condition evaluates to true.

---

setLockManager

`public void setLockManager(ConditionLockManager manager)`

Sets the condition lock manager associated with this condition.

Parameters:

`manager -`
The condition lock manager which can notify the blocked thread.

---

setPOSDataProvider

`public void setPOSDataProvider(POSDataProvider dataProvider)`

Sets the POSDataProvider associated with this condition.

Parameters:

(continued from last page)

`dataProvider -`
The POSDataProvider which will provide property changes to this condition.

getPOSDataProvider

`public POSDataProvider **getPOSDataProvider**()`

Returns the POSDataProvider associated with this condition.

Returns:

The POSDataProvider associated with this condition.

setIndex

`public void **setIndex**(int index)`

Sets the index of this condition (for use within a vector).

Parameters:

`index -`
This value is returned from a condition lock if this particular condition caused the condition lock to be signalled.

getIndex

`public int **getIndex**()`

Gets the vector index of this condition.

Returns:

int This value is returned from a condition lock if this particular condition caused the condition lock to be signalled.

setSession

`public void **setSession**(AEFSession session)`

Sets the AEFSession associated with this condition.

Parameters:

`session`

getSession

`public AEFSession **getSession**()`

Gets the AEFSession associated with this condition.

Returns:

AEFSession

setID

`public void **setID**(int id)`

Sets the unique id of this condition (for use in trace &amp. debug).

Parameters:

`id -`
A unique id which is used in debug trace.

(continued from last page)

getNextID

`public int **getNextID**()`

Gets the next available id.

Returns:

int The next available unique id.

getID

`public int **getID**()`

Gets the unique id of this condition.

Returns:

int The unique id of this condition. Used in debug trace.

addEvaluateListener

`public void **addEvaluateListener**(EvaluateConditionListener listener)`

Add a EvaluateConditionListener which will be notified whenever the condition is evaluated.

Parameters:

`EvaluateConditionListener`

removeEvaluateListener

`public void **removeEvaluateListener**(EvaluateConditionListener listener)`

Remove a EvaluateConditionListener

Parameters:

`EvaluateConditionListener`

notifyEvaluateConditionListeners

```
public void notifyEvaluateConditionListeners(Condition cond,
                                             boolean value)
```

Notifies EvaluateConditionListeners

Parameters:

`Condition`
`boolean`

getTerminalNumber

`public java.lang.String **getTerminalNumber**()`

Gets the terminal number associated with the session for this condition.

Returns:

String terminal number

com.ibm.retail.AEF.automation

Class AbstractPropertyCondition

```
java.lang.Object
  |-com.ibm.retail.AEF.automation.AbstractCondition
    |-com.ibm.retail.AEF.automation.AbstractPropertyCondition
```

All Implemented interfaces:
AEFPropertyChangeListener, PropertyCondition, Condition

Direct Known Subclasses:
PropertyRegexNotMatchCondition, PropertyRegexMatchCondition, PropertyNotEqualsCondition, PropertyNotContainsCondition, PropertyNotContainsAtIndexCondition, PropertyLessThanCondition, PropertyLessOrEqualCondition, PropertyGreaterThanCondition, PropertyGreaterOrEqualCondition, PropertyEqualsCondition, PropertyContainsCondition, PropertyContainsAtIndexCondition

---

public abstract class **AbstractPropertyCondition**
extends AbstractCondition
implements Condition, PropertyCondition, AEFPropertyChangeListener

AbstractPropertyCondition is a base class for objects which detect conditions within POSDataProvider properties.

Field Summary

java.lang.String	category
java.lang.String	name
boolean	nextOnly
java.lang.String	value

Fields inherited from : class com.ibm.retail.AEF.automation.AbstractCondition
counterLock, dataProvider, eligible, id, idCounter, index, listeners, lock, manager, session

Constructor Summary

AbstractPropertyCondition()

Method Summary

void	currentValues(java.util.HashMap values) Populate the HashMap with property name/current value pairs included in the condition.
boolean	evaluate() Evaluates the condition given the current value of the property.

boolean	`evaluate(java.lang.Object newVal)` Evaluates the condition given a value object.
java.lang.String	`getCategory()` Gets the category of the property this condition is monitoring.
boolean	`getNextOnly()` Indicates whether only the next property change will be considered, or all future propety changes will be considered until the condition is true or deactivated.
java.lang.String	`getPropertyName()` Gets the name of the property this condition is monitoring.
java.lang.Double	`parseDouble(java.lang.String doubleStr)` Takes a string as input, and returns a Double representing the value of the string.
void	`setEligible(boolean flag)` Sets the condition eligible for evaluation.

Methods inherited from : class `com.ibm.retail.AEF.automation.AbstractCondition`
`addEvaluateListener, getID, getIndex, getNextID, getPOSDataProvider, getSession, getTerminalNumber, isEligible, notifyEvaluateConditionListeners, removeEvaluateListener, setConditionLock, setEligible, setID, setIndex, setLockManager, setPOSDataProvider, setSession`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

category

`protected java.lang.String` **`category`**

name

`protected java.lang.String` **`name`**

value

`protected java.lang.String` **`value`**

nextOnly

`protected boolean` **`nextOnly`**

Constructors

AbstractPropertyCondition

`public AbstractPropertyCondition()`

Methods

setEligible

`public void setEligible(boolean flag)`

Sets the condition eligible for evaluation.

---

evaluate

`public boolean evaluate()`

Evaluates the condition given the current value of the property.

Returns:

boolean True if the condition evaluates to true.

---

evaluate

`public boolean evaluate(java.lang.Object newVal)`

Evaluates the condition given a value object. This version always returns false. Concrete subclasses must override this method to return a meaninful value.

Parameters:

`newVal` -
The Object value.

Returns:

boolean True if the condition evaluates to true.

---

getCategory

`public java.lang.String getCategory()`

Gets the category of the property this condition is monitoring.

---

getPropertyName

`public java.lang.String getPropertyName()`

Gets the name of the property this condition is monitoring.

---

currentValues

`public void currentValues(java.util.HashMap values)`

Populate the HashMap with property name/current value pairs included in the condition. If a key is already in the HashMap, the new value will not override the value already in the map for the key.

Parameters:

(continued from last page)

`values`

getNextOnly

`public boolean` **`getNextOnly`**`()`

Indicates whether only the next property change will be considered, or all future propety changes will be considered until the condition is true or deactivated.

parseDouble

```
public java.lang.Double parseDouble(java.lang.String doubleStr)
                                throws java.lang.NumberFormatException
```

Takes a string as input, and returns a Double representing the value of the string.

Parameters:

`doubleStr -`
The string to convert to a double.

Returns:

Double The new double representation of the string.

Exceptions:

`NumberFormatException`

com.ibm.retail.AEF.automation

Class ActionRequest

```
java.lang.Object
   |
   +-com.ibm.retail.AEF.automation.ActionRequest
```

All Implemented interfaces:
java.io.Serializable

---

public class **ActionRequest**
extends java.lang.Object
implements java.io.Serializable

ActionRequest is sent to the ActionProcessor to cause an action to be performed. The request includes the action type, as well as a hashtable of arguments.

Constructor Summary

`ActionRequest(java.lang.String className,java.util.Map className)`
Constructs a ActionRequest Instance

Method Summary

`java.util.Map`	`getArguments()` Returns the Map (usually a HashMap) containing the action arguments.
`java.lang.String`	`getClassName()` Returns the classname for the requested action.

Methods inherited from : class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

ActionRequest

```
public ActionRequest(java.lang.String className,
                     java.util.Map args)
```

Constructs a ActionRequest Instance

Parameters:
`String -`
The classname of the action to use.
`Map -`
A collection of key/value arguments.

(continued from last page)

Methods

getClassName

`public java.lang.String` **`getClassName`**`()`

Returns the classname for the requested action.

Returns:

String

getArguments

`public java.util.Map` **`getArguments`**`()`

Returns the Map (usually a HashMap) containing the action arguments.

Returns:

Map

com.ibm.retail.AEF.automation

Class AndCondition

```
java.lang.Object
  |-com.ibm.retail.AEF.automation.AbstractCondition
    |-com.ibm.retail.AEF.automation.AndCondition
```

All Implemented interfaces:
EvaluateConditionListener, Condition, Condition

---

public class **AndCondition**
extends AbstractCondition
implements Condition, Condition, EvaluateConditionListener

AndCondition uses an array of PropertyConditions. When the entire collection of conditions evaluates to true, then the lock associated with this condition is notified.

Field Summary

com.ibm.retail.AEF.automation.Condition[]	conditions

Fields inherited from : class com.ibm.retail.AEF.automation.AbstractCondition
counterLock, dataProvider, eligible, id, idCounter, index, listeners, lock, manager, session

Constructor Summary

AndCondition(Condition[] conditions) Constructs the condition

Method Summary

void	conditionEvaluated(Condition condition,boolean condition) Called by a condition when the condition is evaluated.
void	currentValues(java.util.HashMap values) Populate the HashMap with property name/current value pairs included in the condition.
boolean	evaluate() Examines each of the child conditions immediately and determines whether they are all true.
boolean	evaluate(Condition truecond) Examines each of the child conditions immediately and determines whether they are all true, with the exception of the argument condition.

boolean	`evaluateValues(Condition truecond)` Determines whether all the conditions as observed are currently true, and signals the lock to release any blocked threads if they are all true.
java.lang.String	`explain()` Returns a string explaining this condition.
Condition	`makeOpposite()` Makes a new condition which is the logical opposite of this condition.
void	`setConditionLock(ConditionLock lock)` Sets the condition lock associated with this object.
void	`setEligible(boolean flag)` Sets the condition eligible for evaluation.
void	`setPOSDataProvider(POSDataProvider dataProvider)` Sets the POSDataProvider associated with this condition.

Methods inherited from : class `com.ibm.retail.AEF.automation.AbstractCondition`
`addEvaluateListener, getID, getIndex, getNextID, getPOSDataProvider, getSession, getTerminalNumber, isEligible, notifyEvaluateConditionListeners, removeEvaluateListener, setConditionLock, setEligible, setID, setIndex, setLockManager, setPOSDataProvider, setSession`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

conditions

`protected com.ibm.retail.AEF.automation.Condition` **`conditions`**

Constructors

AndCondition

`public` **`AndCondition`**`(Condition[] conditions)`

Constructs the condition

Parameters:

`conditions -`
An array of Condition objects. All of the conditions must evaluate to true for this AndCondition to evaluate to true.

Methods

(continued from last page)

setPOSDataProvider

```
public void setPOSDataProvider(POSDataProvider dataProvider)
```

Sets the POSDataProvider associated with this condition.

Parameters:

`dataProvider -`
The POSDataProvider which will provide property change events to this condition.

conditionEvaluated

```
public void conditionEvaluated(Condition condition,
                               boolean flag)
```

Called by a condition when the condition is evaluated.

Parameters:

`condition -`
Called by one of the child conditions when it is evaluated.
`flag -`
Indicates whether the child condition evaluated to true or false.

setEligible

```
public void setEligible(boolean flag)
```

Sets the condition eligible for evaluation.

Parameters:

`flag -`
Set to true to make this condition eligible for evaluation.

evaluate

```
public boolean evaluate()
```

Examines each of the child conditions immediately and determines whether they are all true.

Returns:

boolean True if the condition evaluates to true.

evaluate

```
public boolean evaluate(Condition truecond)
```

Examines each of the child conditions immediately and determines whether they are all true, with the exception of the argument condition.

Parameters:

`truecond`

Returns:

boolean True if the condition evaluates to true.

evaluateValues

```
public boolean evaluateValues(Condition truecond)
```

Determines whether all the conditions as observed are currently true, and signals the lock to release any blocked threads if they are all true.

(continued from last page)

Parameters:

`truecond -`
The original condition of the group which evaluated true.

Returns:

boolean True if the condition evaluates to true.

makeOpposite

`public Condition` **`makeOpposite`**`()`

Makes a new condition which is the logical opposite of this condition.

Returns:

Condition

explain

`public java.lang.String` **`explain`**`()`

Returns a string explaining this condition.

Returns:

String

currentValues

`public void` **`currentValues`**`(java.util.HashMap values)`

Populate the HashMap with property name/current value pairs included in the condition. If a key is already in the HashMap, the new value will not override the value already in the map for the key.

Parameters:

`values`

setConditionLock

`public void` **`setConditionLock`**`(ConditionLock lock)`

Sets the condition lock associated with this object.

Parameters:

`ConditionLock`

com.ibm.retail.AEF.automation

Class AndThenCondition

```
java.lang.Object
  |-com.ibm.retail.AEF.automation.AbstractCondition
    |-com.ibm.retail.AEF.automation.AndThenCondition
```

All Implemented interfaces:
EvaluateConditionListener, Condition, Condition

---

public class **AndThenCondition**
extends AbstractCondition
implements Condition, Condition, EvaluateConditionListener

AndThenCondition uses an array of PropertyConditions. When the first condition evaluates true, the second is elligible for evaluation and so on. When the entire collection of conditions evaluates to true, then the lock associated with this condition is notified.

Field Summary

int	childIndex
com.ibm.retail.AEF.automation.Condition[]	conditions

Fields inherited from : class com.ibm.retail.AEF.automation.AbstractCondition
counterLock, dataProvider, eligible, id, idCounter, index, listeners, lock, manager, session

Constructor Summary

AndThenCondition(Condition[] conditions) Constructs the condition

Method Summary

void	conditionEvaluated(Condition condition,boolean condition) Called by a condition when the condition is evaluated.
void	currentValues(java.util.HashMap values) Populate the HashMap with property name/current value pairs included in the condition.
boolean	evaluate() Examines each of the child conditions immediately and determines whether they are all true.
boolean	evaluate(Condition truecond) Examines each of the child conditions immediately and determines whether they are all true, with the exception of the argument condition.

boolean	`evaluateValues(Condition truecond)` Determines whether all the conditions as observed are currently true, and signals the lock to release any blocked threads if they are all true.
java.lang.String	`explain()` Returns a string explaining this condition.
Condition	`makeOpposite()` Makes a new condition which is the logical opposite of this condition.
void	`setConditionLock(ConditionLock lock)` Sets the condition lock associated with this object.
void	`setEligible(boolean flag)` Sets the condition eligible for evaluation.
void	`setPOSDataProvider(POSDataProvider dataProvider)` Sets the POSDataProvider associated with this condition.

Methods inherited from : class `com.ibm.retail.AEF.automation.AbstractCondition`
`addEvaluateListener, getID, getIndex, getNextID, getPOSDataProvider, getSession, getTerminalNumber, isEligible, notifyEvaluateConditionListeners, removeEvaluateListener, setConditionLock, setEligible, setID, setIndex, setLockManager, setPOSDataProvider, setSession`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

conditions

`protected com.ibm.retail.AEF.automation.Condition` **`conditions`**

childIndex

`protected int` **`childIndex`**

Constructors

AndThenCondition

`public` **`AndThenCondition`**`(Condition[] conditions)`

Constructs the condition

Parameters:

(continued from last page)

`conditions -`
An array of Condition objects. All of the conditions must evaluate to true for this AndThenCondition to evaluate to true.

Methods

setPOSDataProvider

```
public void setPOSDataProvider(POSDataProvider dataProvider)
```

Sets the POSDataProvider associated with this condition.

Parameters:

`dataProvider -`
The POSDataProvider which will provide property change events to this condition.

conditionEvaluated

```
public void conditionEvaluated(Condition condition,
                               boolean flag)
```

Called by a condition when the condition is evaluated.

Parameters:

`condition -`
Called by one of the child conditions when it is evaluated.
`flag -`
Indicates whether the child condition evaluated to true or false.

setEligible

```
public void setEligible(boolean flag)
```

Sets the condition eligible for evaluation.

Parameters:

`flag -`
Set to true to make this condition eligible for evaluation.

evaluate

```
public boolean evaluate()
```

Examines each of the child conditions immediately and determines whether they are all true.

Returns:

boolean True if the condition evaluates to true.

evaluate

```
public boolean evaluate(Condition truecond)
```

Examines each of the child conditions immediately and determines whether they are all true, with the exception of the argument condition.

Parameters:

`truecond`

Returns:

boolean True if the condition evaluates to true.

evaluateValues

`public boolean **evaluateValues**(Condition truecond)`

Determines whether all the conditions as observed are currently true, and signals the lock to release any blocked threads if they are all true.

Parameters:

`truecond -`
The original condition of the group which evaluated true.

Returns:

boolean True if the condition evaluates to true.

makeOpposite

`public Condition **makeOpposite**()`

Makes a new condition which is the logical opposite of this condition.

Returns:

Condition

explain

`public java.lang.String **explain**()`

Returns a string explaining this condition.

Returns:

String

currentValues

`public void **currentValues**(java.util.HashMap values)`

Populate the HashMap with property name/current value pairs included in the condition. If a key is already in the HashMap, the new value will not override the value already in the map for the key.

Parameters:

`values`

setConditionLock

`public void **setConditionLock**(ConditionLock lock)`

Sets the condition lock associated with this object.

Parameters:

`ConditionLock`

com.ibm.retail.AEF.automation

Interface BaseInfo

All Subinterfaces:

TransactionTotals, TransactionInfo, SalesTransactionInfo, OperatorInfo, OperatorAuthorization, LineItemInfo, TenderInfo, PointsInfo, ItemInfo, DiscountInfo, CreditInfo, CouponInfo, CustomerInfo

---

public interface **BaseInfo**
extends java.io.Serializable

BaseInfo is an interface which provides a base interface for all Info objects. BaseInfo objects have the capability to receive data and accept updates from a POSAppEvent object through the *update()*method.

Field Summary

static java.lang.String	CLASS_KEY

Method Summary

boolean	getBooleanPropertyValue(java.lang.String propertyName) get a boolean property value
int	getIntegerPropertyValue(java.lang.String propertyName) get an int property value
java.lang.Object	getProperty(java.lang.String name) Gets a property value for a specified property name.
java.util.Iterator	getPropertyEntries() Get an iterator of property name/value entries for this Info object.
void	setProperty(java.lang.String name,java.lang.Object name) Set a property associated with this Info object .
void	update(POSAppEvent evt) Update the contents of this info object using a POSAppEvent.

Fields

CLASS_KEY

public static final java.lang.String **CLASS_KEY**

Methods

(continued from last page)

update

```
public void update(POSAppEvent evt)
```

Update the contents of this info object using a POSAppEvent.

Parameters:

`evt -`
The POSAppEvent.

setProperty

```
public void setProperty(java.lang.String name,
                        java.lang.Object value)
```

Set a property associated with this Info object .

Parameters:

`propName -`
The property name.
`object -`
The property value.

getProperty

```
public java.lang.Object getProperty(java.lang.String name)
```

Gets a property value for a specified property name. Returns a null value if no value exists.

Parameters:

`propName -`
The property name.

Returns:

Object The property value.

getPropertyEntries

```
public java.util.Iterator getPropertyEntries()
```

Get an iterator of property name/value entries for this Info object.

Returns:

iterator of property name/value pairs as Map.Entry objects

getBooleanPropertyValue

```
public boolean getBooleanPropertyValue(java.lang.String propertyName)
                                       throws AEFException
```

get a boolean property value

Returns:

boolean

Exceptions:

`AEFException -`
Possible return codes are:
AEFConst.INVALID_PROPERTY_VALUE
AEFConst.NO_SUCH_PROPERTY

getIntegerPropertyValue

```
public int getIntegerPropertyValue(java.lang.String propertyName)
                                   throws AEFException
```

get an int property value

Returns:

int

Exceptions:

`AEFException -`
Possible return codes are:
AEFConst.INVALID_PROPERTY_VALUE
AEFConst.NO_SUCH_PROPERTY

com.ibm.retail.AEF.automation

Interface Condition

All Known Implementing Classes:
OrCondition, AndThenCondition, AndCondition, AbstractCondition

public interface **Condition**

Condition is an interface for classes which listen for events which may unblock a thread which is waiting on a ThreadManager.wait() call.

Method Summary

void	`addEvaluateListener(EvaluateConditionListener listener)` Add a EvaluateConditionListener which will be notified whenever the condition is evaluated.
void	`currentValues(java.util.HashMap values)` Populate the HashMap with property name/current value pairs included in the condition.
boolean	`evaluate()` Evaluates the condition with the current value of the property without waiting for the next property value change.
java.lang.String	`explain()` Returns a string which describes the condition being monitored.
int	`getID()` Gets the unique id of this condition.
int	`getIndex()` Gets the vector index of this condition.
POSDataProvider	`getPOSDataProvider()` Returns the POSDataProvider associated with this condition.
AEFSession	`getSession()` Gets the AEFSession associated with this condition.
java.lang.String	`getTerminalNumber()` Gets the terminal number associated with the session for this condition.
boolean	`isEligible()` Indicates whether the condition is eligible for evaluation.
Condition	`makeOpposite()` Makes a new condition which is the logical opposite of this condition.
void	`removeEvaluateListener(EvaluateConditionListener listener)` Remove a EvaluateConditionListener

void	`setConditionLock(ConditionLock lock)` Sets the ConditionLock associated with this condition.
void	`setEligible(boolean flag)` Sets the condition eligible for evaluation.
void	`setID(int id)` Sets the unique id of this condition (for use in trace & debug).
void	`setIndex(int index)` Sets the index of this condition (for use within a vector).
void	`setLockManager(ConditionLockManager manager)` Sets the condition lock manager associated with this condition.
void	`setPOSDataProvider(POSDataProvider dataProvider)` Sets the POSDataProvider associated with this condition.
void	`setSession(AEFSession session)` Sets the AEFSession associated with this condition.

Methods

setEligible

`public void **setEligible**(boolean flag)`

Sets the condition eligible for evaluation.

Parameters:

`flag -`
Set to true to make this condition eligible for evaluation.

isEligible

`public boolean **isEligible**()`

Indicates whether the condition is eligible for evaluation.

Returns:

boolean Indicates whether this condition is eligible for evaluation.

setLockManager

`public void **setLockManager**(ConditionLockManager manager)`

Sets the condition lock manager associated with this condition.

Parameters:

`manager -`
The condition lock manager which can notify the blocked thread.

setPOSDataProvider

`public void **setPOSDataProvider**(POSDataProvider dataProvider)`

(continued from last page)

Sets the POSDataProvider associated with this condition.

Parameters:

`dataProvider -`
The POSDataProvider which will provide property change events to this condition.

getPOSDataProvider

`public POSDataProvider **getPOSDataProvider**()`

Returns the POSDataProvider associated with this condition.

Returns:

POSDataProvider The POSDataProvider associated with this condition.

setIndex

`public void **setIndex**(int index)`

Sets the index of this condition (for use within a vector).

Parameters:

`index -`
This integer value will be returned by the ConditionLock to indicate which condition signalled the lock.

getIndex

`public int **getIndex**()`

Gets the vector index of this condition.

Returns:

int The integer which tells the ConditionLock which condition evaluated to true to cause it to be signalled.

setID

`public void **setID**(int id)`

Sets the unique id of this condition (for use in trace & debug).

Parameters:

`id -`
A unique id used for trace and debug.

getID

`public int **getID**()`

Gets the unique id of this condition.

Returns:

int The unique id of this condition used for trace and debug.

evaluate

`public boolean **evaluate**()`

Evaluates the condition with the current value of the property without waiting for the next property value change.

Returns:

(continued from last page)

boolean

makeOpposite

`public Condition **makeOpposite**()`

Makes a new condition which is the logical opposite of this condition.

Returns:

Condition

addEvaluateListener

`public void **addEvaluateListener**(EvaluateConditionListener listener)`

Add a EvaluateConditionListener which will be notified whenever the condition is evaluated.

Parameters:

`EvaluateConditionListener`

removeEvaluateListener

`public void **removeEvaluateListener**(EvaluateConditionListener listener)`

Remove a EvaluateConditionListener

Parameters:

`EvaluateConditionListener`

setSession

`public void **setSession**(AEFSession session)`

Sets the AEFSession associated with this condition.

Parameters:

`session`

getSession

`public AEFSession **getSession**()`

Gets the AEFSession associated with this condition.

Returns:

AEFSession

getTerminalNumber

`public java.lang.String **getTerminalNumber**()`

Gets the terminal number associated with the session for this condition.

Returns:

String terminal number

(continued from last page)

explain

`public java.lang.String **explain**()`

Returns a string which describes the condition being monitored.

Returns:

String

currentValues

`public void **currentValues**(java.util.HashMap values)`

Populate the HashMap with property name/current value pairs included in the condition. If a key is already in the HashMap, the new value will not override the value already in the map for the key.

Parameters:

`values`

setConditionLock

`public void **setConditionLock**(ConditionLock lock)`

Sets the ConditionLock associated with this condition.

Parameters:

`lock`

com.ibm.retail.AEF.automation

Interface Coupon

All Superinterfaces:
LineItem

public interface **Coupon**
extends LineItem

Coupon is an interface which encapsulates a coupon in a transaction.

Field Summary

static java.lang.String	CLASS_KEY

Fields

CLASS_KEY

public static final java.lang.String **CLASS_KEY**

com.ibm.retail.AEF.automation

Interface CouponInfo

All Superinterfaces:
LineItemInfo, BaseInfo

public interface **CouponInfo**
extends LineItemInfo

CouponInfo is an interface which encapsulates the information for a coupon in a transaction.

Field Summary

static java.lang.String	CLASS_KEY

Method Summary

void	addRestrictedPeriod(TimeInterval period) Adds a restricted period for the coupon.
int	getAgeRestriction() Returns the age restriction (in years) for this item.
java.lang.String	getCouponType() Get the coupon type.
java.lang.String	getDealPrice() Returns the deal price for this coupon if the pricing method utilizes any kind of deal price.
int	getDealQuantity() Returns the deal quantity for this coupon if the pricing method utilizes any kind of deal price.
ItemIdentifier	getItemID() Gets the coupon identifier.
java.lang.String	getItemModifier() Gets the coupon modifier.
java.lang.String	getManufacturerNumber() Returns the manufacturer number for this coupon.
java.lang.String	getMultiPricingGroup() Returns the multi-pricing group for this coupon if the pricing method utilizes multi-pricing.
java.lang.String	getPricingMethod() Returns the pricing method for this coupon.

int	`getQuantity()` Get the quantity
java.lang.String	`getReducedPrice()` Returns the reduced price for this coupon if the pricing method utilizes a reduced price.
java.util.Collection	`getRestrictedPeriods()` Returns the restricted time periods associated with this coupon.
java.lang.String	`getUnitPrice()` Get the unit price
java.lang.String	`getValue()` Get the coupon value.
java.lang.String	`getWeight()` Get the weight
boolean	`isItemRepeatAllowed()` Returns whether redeeming another coupon by using item repeat is allowed for this coupon.
boolean	`isTimeRestricted()` Is this coupon time restricted
boolean	`reducesFoodstampBalanceDue()` Indicates whether the coupon reduces the foodstamp balance.
boolean	`reducesTaxDue()` Indicates whether the coupon reduces the transaction tax due.
void	`setAgeRestriction(int years)` Sets the age restriction (in years) for this item.
void	`setCouponType(java.lang.String type)` Set the coupon type
void	`setDealPrice(java.lang.String dealPrice)` Set the deal price for this coupon.
void	`setDealQuantity(int dealQuantity)` Set the deal quantity for this coupon.
void	`setItemID(ItemIdentifier itemID)` Sets the coupon identifier.
void	`setItemModifier(java.lang.String itemModifier)` Sets the coupon modifier.
void	`setItemRepeatAllowed(boolean repeatAllowed)` Set whether redeeming another item by using item repeat is allowed for this coupon.
void	`setManufacturerNumber(java.lang.String manufacturerNumber)` Set the manufacturer number for this coupon.

void	`setMultiPricingGroup(java.lang.String multiPricingGroup)` Set the multi-pricing group for this coupon.
void	`setPricingMethod(java.lang.String pricingMethod)` Set the pricing method for this coupon.
void	`setQuantity(int qty)` Set the quantity
void	`setReducedPrice(java.lang.String reducedPrice)` Set the reduced price for this coupon.
void	`setReducesFoodstampBalanceDue(boolean reduces)` Sets whether the coupon reduces the foodstamp balance.
void	`setReducesTaxDue(boolean reduces)` Set whether the coupon reduces the transaction tax due.
void	`setRestrictedPeriods(java.util.Collection periods)` Sets the restricted time periods associated with this coupon.
void	`setUnitPrice(java.lang.String value)` Set the unit price
void	`setValue(java.lang.String netValue)` Set the coupon net value.
void	`setWeight(java.lang.String weight)` Set the weight

Fields

CLASS_KEY

`public static final java.lang.String` **`CLASS_KEY`**

Methods

setItemID

`public void` **`setItemID`**`(ItemIdentifier itemID)`

Sets the coupon identifier. This should be the scan label, velocity number, sku number, or department number.

Parameters:

`itemID -`
The unique item identifier.

getItemID

`public ItemIdentifier` **`getItemID`**`()`

(continued from last page)

Gets the coupon identifier. This should be the upc code, velocity number, sku number, or department number.

Returns:

ItemIdentifier The item id.

setItemModifier

`public void **setItemModifier**(java.lang.String itemModifier)`

Sets the coupon modifier. The modifier is optional application specific information.

Parameters:

`itemModifier`

getItemModifier

`public java.lang.String **getItemModifier**()`

Gets the coupon modifier. THe modifier is optional application specific information.

Returns:

String

getCouponType

`public java.lang.String **getCouponType**()`

Get the coupon type.

Returns:

String Coupon type = "Store", "Manufacturer", "Electronic", etc.

setCouponType

`public void **setCouponType**(java.lang.String type)`

Set the coupon type

Parameters:

`type`

getUnitPrice

`public java.lang.String **getUnitPrice**()`

Get the unit price

Returns:

unit price

setUnitPrice

`public void **setUnitPrice**(java.lang.String value)`

Set the unit price

Parameters:

`unit -`
price

getQuantity

`public int **getQuantity**()`

Get the quantity

Returns:

quantity

setQuantity

`public void **setQuantity**(int qty)`

Set the quantity

Parameters:

`quantity`

getWeight

`public java.lang.String **getWeight**()`

Get the weight

Returns:

weight

setWeight

`public void **setWeight**(java.lang.String weight)`

Set the weight

Parameters:

`weight`

Exceptions:

`NumberFormatException`

getValue

`public java.lang.String **getValue**()`

Get the coupon value. This is the net value.

Returns:

coupon net value.

setValue

`public void **setValue**(java.lang.String netValue)`

Set the coupon net value.

Parameters:

`netValue`

(continued from last page)

getAgeRestriction

`public int getAgeRestriction()`

Returns the age restriction (in years) for this item.

Returns:

int The number of years old the customer must be to redeem the coupon. Returns zero if there is no age restriction.

setAgeRestriction

`public void setAgeRestriction(int years)`

Sets the age restriction (in years) for this item.

Parameters:

`years -`
The number of years old the customer must be to redeem the item. Use zero for no age restriction.

reducesFoodstampBalanceDue

`public boolean reducesFoodstampBalanceDue()`

Indicates whether the coupon reduces the foodstamp balance.

Returns:

boolean True if coupon reduces foodstamp balance.

setReducesFoodstampBalanceDue

`public void setReducesFoodstampBalanceDue(boolean reduces)`

Sets whether the coupon reduces the foodstamp balance.

Parameters:

`boolean -`
True if coupon reduces foodstamp balance.

isTimeRestricted

`public boolean isTimeRestricted()`

Is this coupon time restricted

Returns:

true if coupon is time restricted

getRestrictedPeriods

`public java.util.Collection getRestrictedPeriods()`

Returns the restricted time periods associated with this coupon.

Returns:

Collection A collection of com.ibm.retail.AEF.event.TimeInterval objects, each representing a restricted time period for the coupon.

(continued from last page)

setRestrictedPeriods

`public void **setRestrictedPeriods**(java.util.Collection periods)`

Sets the restricted time periods associated with this coupon.

Parameters:

`Collection -`
A collection of com.ibm.retail.AEF.event.TimeInterval objects, each representing a restricted time period for the coupon.

addRestrictedPeriod

`public void **addRestrictedPeriod**(TimeInterval period)`

Adds a restricted period for the coupon.

Parameters:

`com.ibm.retail.AEF.event.TimeInterval`

getDealQuantity

`public int **getDealQuantity**()`

Returns the deal quantity for this coupon if the pricing method utilizes any kind of deal price.

Returns:

int The deal quantity.

setDealQuantity

`public void **setDealQuantity**(int dealQuantity)`

Set the deal quantity for this coupon.

Parameters:

`dealQuantity -`
The quantity required for the deal pricing method.

getDealPrice

`public java.lang.String **getDealPrice**()`

Returns the deal price for this coupon if the pricing method utilizes any kind of deal price.

Returns:

String The deal price.

setDealPrice

`public void **setDealPrice**(java.lang.String dealPrice)`

Set the deal price for this coupon.

Parameters:

`dealPrice -`
The deal price for the item if a deal pricing method applies.

getReducedPrice

`public java.lang.String **getReducedPrice**()`

(continued from last page)

Returns the reduced price for this coupon if the pricing method utilizes a reduced price.

Returns:

String The reduced price.

setReducedPrice

`public void` **`setReducedPrice`**`(java.lang.String reducedPrice)`

Set the reduced price for this coupon.

Parameters:

`reducedPrice -`
The reduced price for the coupon if a reduced price pricing method applies.

getMultiPricingGroup

`public java.lang.String` **`getMultiPricingGroup`**`()`

Returns the multi-pricing group for this coupon if the pricing method utilizes multi-pricing.

Returns:

String The multi-pricing group.

setMultiPricingGroup

`public void` **`setMultiPricingGroup`**`(java.lang.String multiPricingGroup)`

Set the multi-pricing group for this coupon.

Parameters:

`multiPricingGroup -`
The multi-pricing group for the coupon if the pricing method utilizes a multi-group pricing method.

isItemRepeatAllowed

`public boolean` **`isItemRepeatAllowed`**`()`

Returns whether redeeming another coupon by using item repeat is allowed for this coupon.

Returns:

boolean

setItemRepeatAllowed

`public void` **`setItemRepeatAllowed`**`(boolean repeatAllowed)`

Set whether redeeming another item by using item repeat is allowed for this coupon.

Parameters:

`repeatAllowed`

reducesTaxDue

`public boolean` **`reducesTaxDue`**`()`

Indicates whether the coupon reduces the transaction tax due.

Returns:

(continued from last page)

boolean

setReducesTaxDue

`public void setReducesTaxDue(boolean reduces)`

Set whether the coupon reduces the transaction tax due.

Parameters:

`reduces`

getPricingMethod

`public java.lang.String getPricingMethod()`

Returns the pricing method for this coupon.

Returns:

String The pricing method.

setPricingMethod

`public void setPricingMethod(java.lang.String pricingMethod)`

Set the pricing method for this coupon.

Parameters:

`pricingMethod`

getManufacturerNumber

`public java.lang.String getManufacturerNumber()`

Returns the manufacturer number for this coupon.

Returns:

String

setManufacturerNumber

`public void setManufacturerNumber(java.lang.String manufacturerNumber)`

Set the manufacturer number for this coupon.

Parameters:

`manufacturerNumber`

com.ibm.retail.AEF.automation

Interface CreditIdentifier

All Superinterfaces:
TenderIdentifier, Identifier

All Known Implementing Classes:
CreditIdentifierImpl

public interface **CreditIdentifier**
extends TenderIdentifier

CreditIdentifier is an interface for holding arguments required for a credit tender.

Field Summary

static java.lang.String	ACCOUNT_NUMBER
static java.lang.String	AMEX
static java.lang.String	AUTHORIZATION_CODE
static java.lang.String	CARD_ID
static java.lang.String	CARD_TYPE
static java.lang.String	CARTE_BLANCHE
static java.lang.String	CLASS_KEY
static java.lang.String	DINERS_CLUB
static java.lang.String	DISCOVER
static java.lang.String	ENROUTE
static java.lang.String	EXPIRATION_DATE
static java.lang.String	JCB
static java.lang.String	MASTERCARD
static java.lang.String	TENDER_KEY
static java.lang.String	UNKNOWN

static java.lang.String	VISA
static java.lang.String	VOUCHER_NUMBER

Method Summary

java.lang.String	getAccountNumber() Gets the account number.
java.lang.String	getAuthorizationCode() Gets the authorization code.
java.lang.String	getCardID() Gets the card id.
java.lang.String	getCardType() Gets the card type.
java.lang.String	getExpirationDate() Gets the expiration date.
java.lang.String	getVoucherNumber() Gets the voucher number.
void	setAccountNumber(java.lang.String accountNumber) Sets the account number.
void	setAuthorizationCode(java.lang.String authCode) Sets the authorization code.
void	setCardID(java.lang.String cardID) Sets the card id.
void	setCardType(java.lang.String cardType) Sets the card type.
void	setExpirationDate(java.lang.String expDate) Sets the expiration date.
void	setVoucherNumber(java.lang.String voucherNum) Sets the voucher number.

Fields

CLASS_KEY

public static final java.lang.String **CLASS_KEY**

(continued from last page)

TENDER_KEY

public static final java.lang.String **TENDER_KEY**

ACCOUNT_NUMBER

public static final java.lang.String **ACCOUNT_NUMBER**

EXPIRATION_DATE

public static final java.lang.String **EXPIRATION_DATE**

CARD_ID

public static final java.lang.String **CARD_ID**

CARD_TYPE

public static final java.lang.String **CARD_TYPE**

VOUCHER_NUMBER

public static final java.lang.String **VOUCHER_NUMBER**

AUTHORIZATION_CODE

public static final java.lang.String **AUTHORIZATION_CODE**

MASTERCARD

public static final java.lang.String **MASTERCARD**

VISA

public static final java.lang.String **VISA**

AMEX

public static final java.lang.String **AMEX**

DINERS_CLUB

public static final java.lang.String **DINERS_CLUB**

DISCOVER

`public static final java.lang.String` **`DISCOVER`**

ENROUTE

`public static final java.lang.String` **`ENROUTE`**

JCB

`public static final java.lang.String` **`JCB`**

CARTE_BLANCHE

`public static final java.lang.String` **`CARTE_BLANCHE`**

UNKNOWN

`public static final java.lang.String` **`UNKNOWN`**

Methods

setCardType

`public void` **`setCardType`**`(java.lang.String cardType)`

Sets the card type.

Parameters:

`cardType`

setAccountNumber

`public void` **`setAccountNumber`**`(java.lang.String accountNumber)`

Sets the account number.

Parameters:

`accountNumber`

setExpirationDate

`public void` **`setExpirationDate`**`(java.lang.String expDate)`

Sets the expiration date.

Parameters:

`expDate`

(continued from last page)

setCardID

`public void **setCardID**(java.lang.String cardID)`

Sets the card id.

Parameters:

`cardID`

setVoucherNumber

`public void **setVoucherNumber**(java.lang.String voucherNum)`

Sets the voucher number.

Parameters:

`voucherNum`

setAuthorizationCode

`public void **setAuthorizationCode**(java.lang.String authCode)`

Sets the authorization code.

Parameters:

`authCode`

getCardType

`public java.lang.String **getCardType**()`

Gets the card type.

Returns:

String (or null if unknown)

getAccountNumber

`public java.lang.String **getAccountNumber**()`

Gets the account number.

Returns:

String

getExpirationDate

`public java.lang.String **getExpirationDate**()`

Gets the expiration date.

Returns:

String

getCardID

`public java.lang.String **getCardID**()`

Gets the card id.

(continued from last page)

Returns:

String

getVoucherNumber

`public java.lang.String` **`getVoucherNumber`**`()`

Gets the voucher number.

Returns:

String

getAuthorizationCode

`public java.lang.String` **`getAuthorizationCode`**`()`

Gets the authorization code.

Returns:

String

com.ibm.retail.AEF.automation

Class CreditIdentifierImpl

```
java.lang.Object
  +-java.util.AbstractMap
    +-java.util.HashMap
      +-com.ibm.retail.AEF.automation.IdentifierImpl
        +-com.ibm.retail.AEF.automation.TenderIdentifierImpl
          +-com.ibm.retail.AEF.automation.CreditIdentifierImpl
```

All Implemented interfaces:
CreditIdentifier, java.util.Map, java.io.Serializable, java.lang.Cloneable, java.util.Map, java.io.Serializable, Identifier, TenderIdentifier

---

public class **CreditIdentifierImpl**
extends TenderIdentifierImpl
implements TenderIdentifier, Identifier, java.io.Serializable, java.util.Map, java.lang.Cloneable, java.io.Serializable, java.util.Map, CreditIdentifier

CreditIdentifierImpl is a class which contains information required to perform a credit tender.

Constructor Summary

`CreditIdentifierImpl()` Constructor.
`CreditIdentifierImpl(java.lang.String amount,java.lang.String amount,java.lang.String amount)` Constructor.
`CreditIdentifierImpl(java.lang.String amount,java.lang.String amount,java.lang.String amount,java.lang.String amount)` Constructor.
`CreditIdentifierImpl(java.lang.String amount,java.lang.String amount,java.lang.String amount,java.lang.String amount,java.lang.String amount)` Constructor.

Method Summary

`java.lang.String`	`getAccountNumber()` Gets the account number.
`java.lang.String`	`getAuthorizationCode()` Gets the authorization code
`java.lang.String`	`getCardID()` Gets the card id.

java.lang.String	getCardType() Gets the card type.
java.lang.String	getExpirationDate() Gets the expiration date.
java.lang.String	getKey() Gets the tender key.
java.lang.String	getVoucherNumber() Gets the voucher number
void	setAccountNumber(java.lang.String accountNumber) Sets the account number.
void	setAuthorizationCode(java.lang.String authCode) Sets the authorization code
void	setCardID(java.lang.String cardID) Sets the card id.
void	setCardType(java.lang.String cardType) Sets the card type.
void	setExpirationDate(java.lang.String expDate) Sets the expiration date.
void	setVoucherNumber(java.lang.String voucherNum) Sets the voucher number

Methods inherited from : class com.ibm.retail.AEF.automation.TenderIdentifierImpl
getAmount, setAmount

Methods inherited from : class com.ibm.retail.AEF.automation.IdentifierImpl
toString

Methods inherited from : class java.util.HashMap
clear, clone, containsKey, containsValue, entrySet, get, isEmpty, keySet, put, putAll, remove, size, values

Methods inherited from : class java.util.AbstractMap
clear, clone, containsKey, containsValue, entrySet, equals, get, hashCode, isEmpty, keySet, put, putAll, remove, size, toString, values

Methods inherited from : class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

CreditIdentifierImpl

```
public CreditIdentifierImpl()
```

Constructor.

CreditIdentifierImpl

```
public CreditIdentifierImpl(java.lang.String amount,
                            java.lang.String cardType,
                            java.lang.String accountNumber)
```

Constructor.

Parameters:

`amount` -
The amount of the tender.
`cardType` -
(pass CreditIdentifier.UNKNOWN for applications which do not require the card type to be identified ahead of time)
`accountNumber`

CreditIdentifierImpl

```
public CreditIdentifierImpl(java.lang.String amount,
                            java.lang.String cardType,
                            java.lang.String accountNumber,
                            java.lang.String expirationDate)
```

Constructor.

Parameters:

`amount` -
The amount of the tender.
`cardType` -
(pass CreditIdentifier.UNKNOWN for applications which do not require the card type to be identified ahead of time)
`accountNumber`
`expirationDate` -
MMYY format

CreditIdentifierImpl

```
public CreditIdentifierImpl(java.lang.String amount,
                            java.lang.String cardType,
                            java.lang.String accountNumber,
                            java.lang.String expirationDate,
                            java.lang.String cardID)
```

Constructor.

Parameters:

`amount` -
The amount of the tender.
`accountNumber`
`cardType` -
(pass CreditIdentifer.UNKNOWN for applications which do not require the card type to be identified ahead of time)
`expirationDate` -
MMYY format
`cardID` -
The card security ID - usually an account number extension printed on the back of the card.

(continued from last page)

Methods

setAccountNumber

`public void **setAccountNumber**(java.lang.String accountNumber)`

Sets the account number.

Parameters:
`accountNumber`

setCardType

`public void **setCardType**(java.lang.String cardType)`

Sets the card type.

Parameters:
`cardType`

setExpirationDate

`public void **setExpirationDate**(java.lang.String expDate)`

Sets the expiration date.

Parameters:
`expDate`

setCardID

`public void **setCardID**(java.lang.String cardID)`

Sets the card id.

Parameters:
`cardID`

setVoucherNumber

`public void **setVoucherNumber**(java.lang.String voucherNum)`

Sets the voucher number

Parameters:
`voucherNum`

setAuthorizationCode

`public void **setAuthorizationCode**(java.lang.String authCode)`

Sets the authorization code

Parameters:
`authCode`

getAccountNumber

`public java.lang.String **getAccountNumber**()`

Gets the account number.

(continued from last page)

Returns:

String

getCardType

`public java.lang.String` **`getCardType`**`()`

Gets the card type.

Returns:

String (or null if unknown)

getExpirationDate

`public java.lang.String` **`getExpirationDate`**`()`

Gets the expiration date.

Returns:

String

getCardID

`public java.lang.String` **`getCardID`**`()`

Gets the card id.

Returns:

String

getVoucherNumber

`public java.lang.String` **`getVoucherNumber`**`()`

Gets the voucher number

Returns:

String

getAuthorizationCode

`public java.lang.String` **`getAuthorizationCode`**`()`

Gets the authorization code

Returns:

String

getKey

`public java.lang.String` **`getKey`**`()`

Gets the tender key. The tender key is used to determine the keys in classes.properties chain for the classes used to add a tender or void a tender. For example, if the method returns "Cash", then "AddCashTenderAction" is the key used to determine the action class for adding the tender to a transaction.

Returns:

(continued from last page)

String

com.ibm.retail.AEF.automation

Interface CreditInfo

All Superinterfaces:
LineItemInfo, BaseInfo

public interface **CreditInfo**
extends LineItemInfo

CreditInfo is an interface which represents data for a credit tender in a transaction.

Field Summary

static java.lang.String	CLASS_KEY

Method Summary

java.lang.String	getAccountNumber() Returns the account number.
java.lang.String	getReferenceNumber() Returns the reference number.
void	setAccountNumber(java.lang.String number) Sets the account amount.
void	setReferenceNumber(java.lang.String number) Sets the reference number.

Fields

CLASS_KEY

public static final java.lang.String **CLASS_KEY**

Methods

getAccountNumber

public java.lang.String **getAccountNumber**()

Returns the account number.

Returns:

String The account number.

setAccountNumber

`public void **setAccountNumber**(java.lang.String number)`

Sets the account amount.

Parameters:

`String -`
The account amount.

getReferenceNumber

`public java.lang.String **getReferenceNumber**()`

Returns the reference number.

Returns:

String The reference number.

setReferenceNumber

`public void **setReferenceNumber**(java.lang.String number)`

Sets the reference number.

Parameters:

`String -`
The reference number.

com.ibm.retail.AEF.automation

Interface CreditTender

All Superinterfaces:
Tender, LineItem

public interface **CreditTender**
extends Tender

CreditTender is an interface which encapsulates transaction credit currency.

com.ibm.retail.AEF.automation

Interface Customer

public interface **Customer**
extends java.rmi.Remote

Customer is an interface for objects representing a customer identified by loyalty number.

Method Summary

`Identifier`	`getIdentifier()` Returns the Identifier for this line item.
`CustomerInfo`	`getInfo()` Returns the information for the customer as determined by the POS application.
`void`	`setIdentifier(Identifier identifier)` Sets the Identifier for this line item.

Methods

getInfo

```
public CustomerInfo getInfo()
                    throws java.rmi.RemoteException
```

Returns the information for the customer as determined by the POS application.

Returns:

CustomerInfo

Exceptions:
`RemoteException`

getIdentifier

```
public Identifier getIdentifier()
                        throws java.rmi.RemoteException
```

Returns the Identifier for this line item.

Returns:

Identifier

Exceptions:
`RemoteException`

setIdentifier

```
public void setIdentifier(Identifier identifier)
                  throws java.rmi.RemoteException
```

(continued from last page)

Sets the Identifier for this line item.

Parameters:

`Identifier`

Exceptions:

`RemoteException`

com.ibm.retail.AEF.automation

Interface CustomerInfo

All Superinterfaces:
BaseInfo

public interface **CustomerInfo**
extends BaseInfo

CustomerInfo is an interface for objects representing a customer identified by loyalty number.

Field Summary

static java.lang.String	CLASS_KEY

Method Summary

java.lang.String	getEmail() Get the customer email
java.lang.String	getID() Get the customer ID
java.util.Collection	getLoyaltyMessages() Gets the loyalty messages
java.lang.String	getName() Get the customer full name
java.lang.String	getPoints() Gets the customer points
java.lang.String	getPointsBalance(java.lang.String type) Gets the points balance for a specific category GrossPoints etc.
java.util.Collection	getPointsBalances() Get the Points Balances
java.lang.String	getPointsTotal(java.lang.String type) Gets the points total for a specific type of points.
java.util.Collection	getPointsTotals() Get the Points Totals
java.util.Collection	getTargetedCouponIDs() Gets the targeted coupon ids
void	setEmail(java.lang.String value) Set the customer email

void	`setID(java.lang.String value)` Set the customer ID
void	`setName(java.lang.String value)` Set the customer full name
void	`setPoints(java.lang.String points)` Set the customer points

Fields

CLASS_KEY

`public static final java.lang.String` **`CLASS_KEY`**

Methods

getID

`public java.lang.String` **`getID`**`()`

Get the customer ID

Returns:

customer ID

setID

`public void` **`setID`**`(java.lang.String value)`

Set the customer ID

Parameters:

`customer -`
ID

getName

`public java.lang.String` **`getName`**`()`

Get the customer full name

Returns:

customer full name

setName

`public void` **`setName`**`(java.lang.String value)`

Set the customer full name

Parameters:

`customer -`
full name

getEmail

`public java.lang.String **getEmail**()`

Get the customer email

Returns:

customer email

setEmail

`public void **setEmail**(java.lang.String value)`

Set the customer email

Parameters:

`customer -` email

setPoints

`public void **setPoints**(java.lang.String points)`

Set the customer points

Parameters:

`customer -` points

getPoints

`public java.lang.String **getPoints**()`

Gets the customer points

Returns:

customer points

getLoyaltyMessages

`public java.util.Collection **getLoyaltyMessages**()`

Gets the loyalty messages

Returns:

loyalty messages

getTargetedCouponIDs

`public java.util.Collection **getTargetedCouponIDs**()`

Gets the targeted coupon ids

Returns:

targeted coupon ids

(continued from last page)

getPointsTotals

`public java.util.Collection` **`getPointsTotals`**`()`

Get the Points Totals

Returns:

Collection of PointsTotal objects (null if no totals)

See Also:

com.ibm.retail.AEF.event.PointsTotal

---

getPointsBalances

`public java.util.Collection` **`getPointsBalances`**`()`

Get the Points Balances

Returns:

Collection of PointsTotal objects (null if no totals)

See Also:

com.ibm.retail.AEF.event.PointsTotal

---

getPointsBalance

`public java.lang.String` **`getPointsBalance`**`(java.lang.String type)`

Gets the points balance for a specific category GrossPoints etc...

Returns:

points balance

---

getPointsTotal

`public java.lang.String` **`getPointsTotal`**`(java.lang.String type)`

Gets the points total for a specific type of points. GrossPoints etc...

Returns:

points total

com.ibm.retail.AEF.automation

Interface Discount

All Superinterfaces:
LineItem

public interface **Discount**
extends LineItem

Points is an interface which encapsulates a line item or transaction discount.

Field Summary

static java.lang.String	CLASS_KEY

Fields

CLASS_KEY

public static final java.lang.String **CLASS_KEY**

com.ibm.retail.AEF.automation

Interface DiscountInfo

All Superinterfaces:
LineItemInfo, BaseInfo

public interface **DiscountInfo**
extends LineItemInfo

DiscountInfo is an interface which encapsulates the item information for a line item or transaction discount.

Field Summary

static java.lang.String	CLASS_KEY

Method Summary

java.lang.String	getAmount() Gets the discount amount.
java.lang.String	getAppliesTo() For line item discounts, get a value indicating the line item the discount applies to.
java.lang.String	getMethod() Gets the discount method.
java.lang.String	getRate() Gets the discount rate if the discount method is "percent".
java.lang.String	getReason() Gets the discount reason.
java.lang.String	getType() Gets the discount type.
boolean	isTransactionDiscount() Indicates whether the discount is a transaction discount.
boolean	reducesTaxBalanceDue() Indicates whether the discount reduces the tax balance due.
void	setAmount(java.lang.String amount) Sets the discount amount.
void	setAppliesTo(java.lang.String appliesTo) For line item discounts, sets a value indicating the line item the discount applies to.
void	setMethod(java.lang.String method) Sets the discount method.

void	`setRate(java.lang.String rate)` Sets the discount rate if the discount method is "percent".
void	`setReason(java.lang.String reason)` Sets the discount reason.
void	`setReducesTaxBalanceDue(boolean flag)` Sets whether the discount reduces the tax balance due.
void	`setTransactionDiscount(boolean flag)` Sets whether the discount is a transaction discount.
void	`setType(java.lang.String type)` Sets the discount type.

Fields

CLASS_KEY

`public static final java.lang.String` **`CLASS_KEY`**

Methods

getAppliesTo

`public java.lang.String` **`getAppliesTo`**`()`

For line item discounts, get a value indicating the line item the discount applies to. For example, may be "previous", "next", "1", "10".

Returns:

String

setAppliesTo

`public void` **`setAppliesTo`**`(java.lang.String appliesTo)`

For line item discounts, sets a value indicating the line item the discount applies to. For example, may be "previous", "next", "1", "10".

Returns:

param

getMethod

`public java.lang.String` **`getMethod`**`()`

Gets the discount method. For example, "percent" or "allowance".

Returns:

String

setMethod

`public void **setMethod**(java.lang.String method)`

Sets the discount method. For example, "percent" or "allowance".

Parameters:

`String`

getRate

`public java.lang.String **getRate**()`

Gets the discount rate if the discount method is "percent".

Returns:

String

setRate

`public void **setRate**(java.lang.String rate)`

Sets the discount rate if the discount method is "percent".

Parameters:

`String`

getAmount

`public java.lang.String **getAmount**()`

Gets the discount amount.

Returns:

String

setAmount

`public void **setAmount**(java.lang.String amount)`

Sets the discount amount.

Parameters:

`String`

getType

`public java.lang.String **getType**()`

Gets the discount type. This value is application specific. For IBM Supermarket, it corresponds to the discount group.

Returns:

String

setType

`public void **setType**(java.lang.String type)`

Sets the discount type.

(continued from last page)

Parameters:
`String`

getReason

`public java.lang.String getReason()`

Gets the discount reason. This value is application specific. For IBM GSA, it corresponds to the discount reason code.

Returns:

String

setReason

`public void setReason(java.lang.String reason)`

Sets the discount reason.

Parameters:
`String`

reducesTaxBalanceDue

`public boolean reducesTaxBalanceDue()`

Indicates whether the discount reduces the tax balance due.

Returns:

boolean

setReducesTaxBalanceDue

`public void setReducesTaxBalanceDue(boolean flag)`

Sets whether the discount reduces the tax balance due.

Parameters:
`boolean`

isTransactionDiscount

`public boolean isTransactionDiscount()`

Indicates whether the discount is a transaction discount.

Returns:

boolean

setTransactionDiscount

`public void setTransactionDiscount(boolean flag)`

Sets whether the discount is a transaction discount.

Parameters:
`boolean`

com.ibm.retail.AEF.automation

Interface Identifier

All Subinterfaces:

TransactionIdentifier, TenderIdentifier, MSRCreditIdentifier, CreditIdentifier, OperatorIdentifier, LoyaltyIdentifier, ItemIdentifier

---

public interface **Identifier**
extends java.util.Map

Identifier is a base interface for various identifiers such as ItemIdentifier.

Field Summary

static java.lang.String	ACCOUNT_NUMBER
static java.lang.String	PRICE

Method Summary

java.lang.String	toString() Returns a string representation of the object.

Fields

ACCOUNT_NUMBER

`public static final java.lang.String` **`ACCOUNT_NUMBER`**

PRICE

`public static final java.lang.String` **`PRICE`**

Methods

toString

`public java.lang.String` **`toString`**`()`

Returns a string representation of the object.

Returns:

String

com.ibm.retail.AEF.automation

Interface Item

All Superinterfaces:
LineItem

public interface **Item**
extends LineItem

Item is an interface which encapsulates an item which is sold in a transaction.

Field Summary

static java.lang.String	CLASS_KEY

Fields

CLASS_KEY

public static final java.lang.String **CLASS_KEY**

com.ibm.retail.AEF.automation

Interface ItemIdentifier

All Superinterfaces:
Identifier

All Known Implementing Classes:
ItemIdentifierImpl

public interface **ItemIdentifier**
extends Identifier

ItemIdentifier is an interface which encapsulates an item code, and the type of the item code. For example, an item may be identified as a upc code, a velocity code, a sku code, or a department.

Field Summary

static java.lang.String	CLASS
static java.lang.String	CLASS_KEY
static java.lang.String	DEAL_PRICE
static java.lang.String	DEAL_QUANTITY
static java.lang.String	DEPARTMENT
static java.lang.String	ITEM_CODE
static java.lang.String	ITEM_TYPE
static java.lang.String	NONMERCHANDISE
static java.lang.String	ORIGINAL_SALESPERSON
static java.lang.String	PRICE
static java.lang.String	QUANTITY
static java.lang.String	TAXABLE
static java.lang.String	WEIGHT

Method Summary

java.lang.String	getDealPrice() Gets the item's deal price.
java.lang.String	getDealQuantity() Gets the item's deal quantity.
java.lang.String	getDepartment() Gets the item's department.
java.lang.String	getItemClass() Gets the item's class.
java.lang.String	getItemCode() Gets the item code.
java.lang.String	getItemCodeType() Gets the item code type.
java.lang.String	getOriginalSalesperson() Gets the Original Salesperson for this item.
java.lang.String	getPrice() Gets the item's price.
java.lang.String	getQuantity() Gets the item's quantity.
java.lang.String	getWeight() Gets the item's weight.
boolean	isNonMerchandise() Gets the item's non-merchandise status.
void	setDealPrice(java.lang.String dealPrice) Sets the deal price for the item.
void	setDealQuantity(java.lang.String dealQuantity) Sets the deal quantity for the item.
void	setDepartment(java.lang.String itemdepartment) Sets the department for the item.
void	setItemClass(java.lang.String itemClass) Sets the class for the item.
void	setItemCode(java.lang.String itemCode) Sets the item code.
void	setItemCodeType(java.lang.String itemCodeType) Sets the item code type.
void	setNonMerchandise(boolean itemNonMerchandiseStatus) Sets the non-merchandise flag for this item.

void	`setOriginalSalesperson(java.lang.String originalSalesperson)` Sets the Original Salesperson for this item.
void	`setPrice(java.lang.String itemPrice)` Sets the price for the item.
void	`setQuantity(java.lang.String itemQuantity)` Sets the quantity.
void	`setWeight(java.lang.String itemWeight)` Sets the weight of the item.

Fields

CLASS_KEY

`public static final java.lang.String` **`CLASS_KEY`**

ITEM_CODE

`public static final java.lang.String` **`ITEM_CODE`**

QUANTITY

`public static final java.lang.String` **`QUANTITY`**

PRICE

`public static final java.lang.String` **`PRICE`**

WEIGHT

`public static final java.lang.String` **`WEIGHT`**

DEPARTMENT

`public static final java.lang.String` **`DEPARTMENT`**

CLASS

`public static final java.lang.String` **`CLASS`**

(continued from last page)

TAXABLE

`public static final java.lang.String` **`TAXABLE`**

NONMERCHANDISE

`public static final java.lang.String` **`NONMERCHANDISE`**

ITEM_TYPE

`public static final java.lang.String` **`ITEM_TYPE`**

DEAL_PRICE

`public static final java.lang.String` **`DEAL_PRICE`**

DEAL_QUANTITY

`public static final java.lang.String` **`DEAL_QUANTITY`**

ORIGINAL_SALESPERSON

`public static final java.lang.String` **`ORIGINAL_SALESPERSON`**

Methods

setItemCode

`public void` **`setItemCode`**`(java.lang.String itemCode)`

Sets the item code. This should be the upc code, velocity number, sku number, or department number.

Parameters:

`itemCode -`
The unique item identifier.

setItemCodeType

`public void` **`setItemCodeType`**`(java.lang.String itemCodeType)`

Sets the item code type. This should be UPC, VELOCITY, SKU, or DEPT as defined in AEFConst.java.

Parameters:

`itemCodeType -`
The type of identifier (sku, velocity, upc, dept, etc)

See Also:

com.ibm.retail.si.util.AEFConst for the valid id types.

(continued from last page)

setQuantity

`public void **setQuantity**(java.lang.String itemQuantity)`

Sets the quantity.

Parameters:

`itemQuantity -`
The quantity for the item.

setPrice

`public void **setPrice**(java.lang.String itemPrice)`

Sets the price for the item.

Parameters:

`itemPrice -`
The price for the item.

setDealPrice

`public void **setDealPrice**(java.lang.String dealPrice)`

Sets the deal price for the item.

Parameters:

`dealPrice -`
The price for a number of items, for example 3 for $1.00.

setDealQuantity

`public void **setDealQuantity**(java.lang.String dealQuantity)`

Sets the deal quantity for the item.

Parameters:

`dealQuantity -`
The number of items selling for a price, for example 3 for $1.00.

setWeight

`public void **setWeight**(java.lang.String itemWeight)`

Sets the weight of the item.

Parameters:

`itemWeight -`
The weight of the item.

setDepartment

`public void **setDepartment**(java.lang.String itemdepartment)`

Sets the department for the item.

Parameters:

`itemdepartment -`
The department for the item.

setItemClass

`public void **setItemClass**(java.lang.String itemClass)`

Sets the class for the item.

(continued from last page)

Parameters:

`itemClass -`
The item's class.

setNonMerchandise

`public void` **`setNonMerchandise`**`(boolean itemNonMerchandiseStatus)`

Sets the non-merchandise flag for this item.

Parameters:

`itemNonMerchandiseStatus -`
The item's non-merchandise status.

setOriginalSalesperson

`public void` **`setOriginalSalesperson`**`(java.lang.String originalSalesperson)`

Sets the Original Salesperson for this item. This is used for returns on some systems.

Parameters:

`originalSalesperson -`
The original salesperson for this item.

getItemCode

`public java.lang.String` **`getItemCode`**`()`

Gets the item code.

Returns:

String The item code.

getItemCodeType

`public java.lang.String` **`getItemCodeType`**`()`

Gets the item code type. This should be UPC, VELOCITY, SKU, or DEPT as defined in AEFConst.java.

Returns:

String The type of identifier (sku, velocity, upc, dept, etc)

See Also:

com.ibm.retail.si.util.AEFConst for the valid id types.

getQuantity

`public java.lang.String` **`getQuantity`**`()`

Gets the item's quantity.

Returns:

String The item's quantity.

getPrice

`public java.lang.String` **`getPrice`**`()`

Gets the item's price.

(continued from last page)

Returns:

String The item's price.

getDealPrice

`public java.lang.String` **`getDealPrice`**`()`

Gets the item's deal price.

Returns:

String

getDealQuantity

`public java.lang.String` **`getDealQuantity`**`()`

Gets the item's deal quantity.

Returns:

String

getWeight

`public java.lang.String` **`getWeight`**`()`

Gets the item's weight.

Returns:

String The item's weight.

getDepartment

`public java.lang.String` **`getDepartment`**`()`

Gets the item's department.

Returns:

String The item's department.

getItemClass

`public java.lang.String` **`getItemClass`**`()`

Gets the item's class.

Returns:

String The item's class.

getOriginalSalesperson

`public java.lang.String` **`getOriginalSalesperson`**`()`

Gets the Original Salesperson for this item. This is used for returns on some systems.

Returns:

(continued from last page)

The original salesperson for this item.

isNonMerchandise

`public boolean` **`isNonMerchandise`**`()`

Gets the item's non-merchandise status.

Returns:

boolean The item's non-merchandise status.

com.ibm.retail.AEF.automation

Class ItemIdentifierImpl

```
java.lang.Object
  +-java.util.AbstractMap
    +-java.util.HashMap
      +-com.ibm.retail.AEF.automation.IdentifierImpl
        +-com.ibm.retail.AEF.automation.ItemIdentifierImpl
```

All Implemented interfaces:
ItemIdentifier, java.util.Map, java.io.Serializable, java.lang.Cloneable, java.util.Map, java.io.Serializable, Identifier

---

public class **ItemIdentifierImpl**
extends IdentifierImpl
implements Identifier, java.io.Serializable, java.util.Map, java.lang.Cloneable, java.io.Serializable, java.util.Map, ItemIdentifier

ItemIdentifierImpl is a class which contains item code information. It contains the item code data, as well as indicating the type of item code.

Constructor Summary

`ItemIdentifierImpl()` Constructor.
`ItemIdentifierImpl(java.lang.String itemCode,java.lang.String itemCode)` Constructor.

Method Summary

`java.lang.String`	`getDealPrice()` Gets the item's deal price.
`java.lang.String`	`getDealQuantity()` Gets the item's deal quantity.
`java.lang.String`	`getDepartment()` Gets the item's department.
`java.lang.String`	`getItemClass()` Gets the item's class.
`java.lang.String`	`getItemCode()` Gets the item code.
`java.lang.String`	`getItemCodeType()` Gets the item code type.

java.lang.String	getOriginalSalesperson() Gets the Original Salesperson for this item.
java.lang.String	getPrice() Gets the item's price.
java.lang.String	getQuantity() Gets the item's quantity.
java.lang.String	getWeight() Gets the item's weight.
boolean	isNonMerchandise() Gets the item's non-merchandise status.
void	setDealPrice(java.lang.String dealPrice) Sets the deal price for the item.
void	setDealQuantity(java.lang.String dealQuantity) Sets the deal quantity for the item.
void	setDepartment(java.lang.String itemdepartment) Sets the department for the item.
void	setItemClass(java.lang.String itemClass) Sets the class for the item.
void	setItemCode(java.lang.String itemCode) Sets the item code.
void	setItemCodeType(java.lang.String itemCodeType) Sets the item code type.
void	setNonMerchandise(boolean itemNonMerchandiseStatus) Sets the non-merchandise flag for this item.
void	setOriginalSalesperson(java.lang.String originalSalesperson) Sets the Original Salesperson for this item.
void	setPrice(java.lang.String itemPrice) Sets the price for the item.
void	setQuantity(java.lang.String itemQuantity) Sets the quantity.
void	setWeight(java.lang.String itemWeight) Sets the weight of the item.

Methods inherited from : class com.ibm.retail.AEF.automation.IdentifierImpl
toString

Methods inherited from : class java.util.HashMap

`clear, clone, containsKey, containsValue, entrySet, get, isEmpty, keySet, put, putAll, remove, size, values`

Methods inherited from : class `java.util.AbstractMap`
`clear, clone, containsKey, containsValue, entrySet, equals, get, hashCode, isEmpty, keySet, put, putAll, remove, size, toString, values`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

ItemIdentifierImpl

```
public ItemIdentifierImpl()
```

Constructor.

ItemIdentifierImpl

```
public ItemIdentifierImpl(java.lang.String itemCode,
                          java.lang.String itemCodeType)
```

Constructor.

Parameters:

`itemCode -`
The unique item identifier.
`itemCodeType -`
The type of identifier (sku, velocity, upc, dept, etc) See AEFConst.java for the valid id types.

Methods

setItemCode

```
public void setItemCode(java.lang.String itemCode)
```

Sets the item code. This should be the upc code, velocity number, sku number, or department number.

Parameters:

`itemCode -`
The unique item identifier.

setItemCodeType

```
public void setItemCodeType(java.lang.String itemCodeType)
```

Sets the item code type. This should be UPC, VELOCITY, SKU, or DEPT as defined in AEFConst.java.

Parameters:

`itemCodeType -`
The type of identifier (sku, velocity, upc, dept, etc)

See Also:

(continued from last page)

com.ibm.retail.si.util.AEFConst for the valid id types.

setQuantity

`public void **setQuantity**(java.lang.String itemQuantity)`

Sets the quantity.

Parameters:

`itemQuantity -`
The quantity for the item.

setPrice

`public void **setPrice**(java.lang.String itemPrice)`

Sets the price for the item.

Parameters:

`itemPrice -`
The price for the item.

setDealPrice

`public void **setDealPrice**(java.lang.String dealPrice)`

Sets the deal price for the item.

Parameters:

`dealPrice -`
The price for a number of items, for example 3 for $1.00.

setDealQuantity

`public void **setDealQuantity**(java.lang.String dealQuantity)`

Sets the deal quantity for the item.

Parameters:

`dealQuantity -`
The number of items selling for a price, for example 3 for $1.00.

setWeight

`public void **setWeight**(java.lang.String itemWeight)`

Sets the weight of the item.

Parameters:

`itemWeight -`
The weight of the item.

setDepartment

`public void **setDepartment**(java.lang.String itemdepartment)`

Sets the department for the item.

Parameters:

`itemdepartment -`
The department for the item.

(continued from last page)

setItemClass

`public void **setItemClass**(java.lang.String itemClass)`

Sets the class for the item.

Parameters:

`itemClass -`
The item's class.

setNonMerchandise

`public void **setNonMerchandise**(boolean itemNonMerchandiseStatus)`

Sets the non-merchandise flag for this item.

Parameters:

`itemNonMerchandiseStatus -`
The item's non-merchandise status.

setOriginalSalesperson

`public void **setOriginalSalesperson**(java.lang.String originalSalesperson)`

Sets the Original Salesperson for this item. This is used for returns on some systems.

Parameters:

`originalSalesperson -`
The original salesperson for this item.

getItemCode

`public java.lang.String **getItemCode**()`

Gets the item code.

Returns:

String The item code.

getItemCodeType

`public java.lang.String **getItemCodeType**()`

Gets the item code type. This should be UPC, VELOCITY, SKU, or DEPT as defined in AEFConst.java.

Returns:

String The type of identifier (sku, velocity, upc, dept, etc)

See Also:

com.ibm.retail.si.util.AEFConst for the valid id types.

getQuantity

`public java.lang.String **getQuantity**()`

Gets the item's quantity.

Returns:

String The item's quantity.

getPrice

`public java.lang.String` **`getPrice`**`()`

Gets the item's price.

Returns:

String The item's price.

getDealPrice

`public java.lang.String` **`getDealPrice`**`()`

Gets the item's deal price.

Returns:

String

getDealQuantity

`public java.lang.String` **`getDealQuantity`**`()`

Gets the item's deal quantity.

Returns:

String

getWeight

`public java.lang.String` **`getWeight`**`()`

Gets the item's weight.

Returns:

String The item's weight.

getDepartment

`public java.lang.String` **`getDepartment`**`()`

Gets the item's department.

Returns:

String The item's department.

getItemClass

`public java.lang.String` **`getItemClass`**`()`

Gets the item's class.

Returns:

String The item's class.

(continued from last page)

getOriginalSalesperson

`public java.lang.String` **`getOriginalSalesperson`**`()`

Gets the Original Salesperson for this item. This is used for returns on some systems.

Returns:

The original salesperson for this item.

isNonMerchandise

`public boolean` **`isNonMerchandise`**`()`

Gets the item's non-merchandise status.

Returns:

boolean The item's non-merchandise status.

com.ibm.retail.AEF.automation

Interface ItemInfo

All Superinterfaces:
LineItemInfo, BaseInfo

public interface **ItemInfo**
extends LineItemInfo

ItemInfo is an interface which encapsulates the item information for an item which is sold in a transaction.

Field Summary

static java.lang.String	CLASS_KEY

Method Summary

void	addRestrictedPeriod(TimeInterval period) Adds a restricted period for the item.
int	getAgeRestriction() Returns the age restriction (in years) for this item.
java.lang.String	getDealPrice() Returns the deal price for this item if the pricing method utilizes any kind of deal price.
int	getDealQuantity() Returns the deal quantity for this item if the pricing method utilizes any kind of deal price.
java.lang.String	getExtendedPrice() Get the extendedPrice
ItemIdentifier	getItemID() Gets the item identifier.
java.lang.String	getItemModifier() Gets the item modifier.
java.lang.String	getMultiPricingGroup() Returns the multi-pricing group for this item if the pricing method utilizes multi-pricing.
java.lang.String	getPricingMethod() Returns the pricing method for this item.
int	getQuantity() Get the quantity
java.lang.String	getReducedPrice() Returns the reduced price for this item if the pricing method utilizes a reduced price.

java.lang.String	`getRegularPrice()` Returns the regular price for this item.
java.util.Collection	`getRestrictedPeriods()` Returns the restricted time periods associated with this item.
java.lang.String	`getUnitPrice()` Get the unit price
java.lang.String	`getWeight()` Get the weight
boolean	`isFoodstampEligible()` Indicates whether the item is eligible for foodstamps.
boolean	`isItemRepeatAllowed()` Returns whether selling another item by using item repeat is allowed for this item.
boolean	`isTaxable()` Indicates whether the item is taxable.
boolean	`isTimeRestricted()` Is this item time restricted
boolean	`isWICEligible()` Indicates whether the item is eligible for WIC.
void	`setAgeRestriction(int years)` Sets the age restriction (in years) for this item.
void	`setDealPrice(java.lang.String dealPrice)` Set the deal price for this item.
void	`setDealQuantity(int dealQuantity)` Set the deal quantity for this item.
void	`setExtendedPrice(java.lang.String value)` Set the extendedPrice
void	`setFoodstampEligible(boolean eligible)` Sets whether the item is eligible for foodstamps.
void	`setItemID(ItemIdentifier itemID)` Sets the item identifier.
void	`setItemModifier(java.lang.String itemModifier)` Sets the item modifier.
void	`setItemRepeatAllowed(boolean repeatAllowed)` Set whether selling another item by using item repeat is allowed for this item.
void	`setMultiPricingGroup(java.lang.String multiPricingGroup)` Set the multi-pricing group for this item.

void	`setPricingMethod(java.lang.String pricingMethod)` Set the pricing method for this item.
void	`setQuantity(int qty)` Set the quantity
void	`setReducedPrice(java.lang.String reducedPrice)` Set the reduced price for this item.
void	`setRegularPrice(java.lang.String price)` Set the regular price for this item.
void	`setRestrictedPeriods(java.util.Collection periods)` Sets the restricted time periods associated with this item.
void	`setTaxable(boolean taxable)` Set whether the item is taxable.
void	`setUnitPrice(java.lang.String value)` Set the unit price
void	`setWeight(java.lang.String weight)` Set the weight
void	`setWICEligible(boolean eligible)` Sets whether the item is eligible for WIC.

Fields

CLASS_KEY

`public static final java.lang.String` **`CLASS_KEY`**

Methods

setItemID

`public void` **`setItemID`**`(ItemIdentifier itemID)`

Sets the item identifier. This should be the scan label, velocity number, sku number, or department number.

Parameters:

`itemID -`
The unique item identifier.

getItemID

`public ItemIdentifier` **`getItemID`**`()`

Gets the item identifier. This should be the upc code, velocity number, sku number, or department number.

Returns:

(continued from last page)

ItemIdentifier The item id.

setItemModifier

`public void **setItemModifier**(java.lang.String itemModifier)`

Sets the item modifier. The modifier is optional application specific information.

Parameters:

`itemModifier`

getItemModifier

`public java.lang.String **getItemModifier**()`

Gets the item modifier. THe modifier is optional application specific information.

Returns:

String

getUnitPrice

`public java.lang.String **getUnitPrice**()`

Get the unit price

Returns:

unit price

setUnitPrice

`public void **setUnitPrice**(java.lang.String value)`

Set the unit price

Parameters:

`unit -`
price

getQuantity

`public int **getQuantity**()`

Get the quantity

Returns:

quantity

setQuantity

`public void **setQuantity**(int qty)`

Set the quantity

Parameters:

`quantity`

(continued from last page)

getWeight

`public java.lang.String **getWeight**()`

Get the weight

Returns:

weight

setWeight

`public void **setWeight**(java.lang.String weight)`

Set the weight

Parameters:

`weight`

Exceptions:

`NumberFormatException`

getExtendedPrice

`public java.lang.String **getExtendedPrice**()`

Get the extendedPrice

Returns:

extendedPrice

setExtendedPrice

`public void **setExtendedPrice**(java.lang.String value)`

Set the extendedPrice

Parameters:

`extendedPrice`

getAgeRestriction

`public int **getAgeRestriction**()`

Returns the age restriction (in years) for this item.

Returns:

int The number of years old the customer must be to purchase the item. Returns zero if there is no age restriction.

setAgeRestriction

`public void **setAgeRestriction**(int years)`

Sets the age restriction (in years) for this item.

Parameters:

`years -`
The number of years old the customer must be to purchase the item. Use zero for no age restriction.

(continued from last page)

isFoodstampEligible

`public boolean **isFoodstampEligible**()`

Indicates whether the item is eligible for foodstamps.

Returns:

boolean True if eligible for food stamps.

setFoodstampEligible

`public void **setFoodstampEligible**(boolean eligible)`

Sets whether the item is eligible for foodstamps.

Parameters:

`boolean -`
True if eligible for food stamps.

isWICEligible

`public boolean **isWICEligible**()`

Indicates whether the item is eligible for WIC.

Returns:

boolean True if eligible for WIC.

setWICEligible

`public void **setWICEligible**(boolean eligible)`

Sets whether the item is eligible for WIC.

Parameters:

`boolean -`
True if eligible for WIC.

isTimeRestricted

`public boolean **isTimeRestricted**()`

Is this item time restricted

Returns:

true if item is time restricted

getRestrictedPeriods

`public java.util.Collection **getRestrictedPeriods**()`

Returns the restricted time periods associated with this item.

Returns:

Collection A collection of com.ibm.retail.AEF.event.TimeInterval objects, each representing a restricted time period for the item.

(continued from last page)

setRestrictedPeriods

`public void **setRestrictedPeriods**(java.util.Collection periods)`

Sets the restricted time periods associated with this item.

Parameters:

`Collection -`
A collection of com.ibm.retail.AEF.event.TimeInterval objects, each representing a restricted time period for the item.

addRestrictedPeriod

`public void **addRestrictedPeriod**(TimeInterval period)`

Adds a restricted period for the item.

Parameters:

`com.ibm.retail.AEF.event.TimeInterval`

getDealQuantity

`public int **getDealQuantity**()`

Returns the deal quantity for this item if the pricing method utilizes any kind of deal price.

Returns:

int The deal quantity.

setDealQuantity

`public void **setDealQuantity**(int dealQuantity)`

Set the deal quantity for this item.

Parameters:

`dealQuantity -`
The quantity required for the deal pricing method.

getDealPrice

`public java.lang.String **getDealPrice**()`

Returns the deal price for this item if the pricing method utilizes any kind of deal price.

Returns:

String The deal price.

setDealPrice

`public void **setDealPrice**(java.lang.String dealPrice)`

Set the deal price for this item.

Parameters:

`dealPrice -`
The deal price for the item if a deal pricing method applies.

getReducedPrice

`public java.lang.String **getReducedPrice**()`

Returns the reduced price for this item if the pricing method utilizes a reduced price.

(continued from last page)

Returns:

String The reduced price.

setReducedPrice

`public void setReducedPrice(java.lang.String reducedPrice)`

Set the reduced price for this item.

Parameters:

`reducedPrice -`
The reduced price for the item if a reduced price pricing method applies.

getMultiPricingGroup

`public java.lang.String getMultiPricingGroup()`

Returns the multi-pricing group for this item if the pricing method utilizes multi-pricing.

Returns:

String The multi-pricing group.

setMultiPricingGroup

`public void setMultiPricingGroup(java.lang.String multiPricingGroup)`

Set the multi-pricing group for this item.

Parameters:

`multiPricingGroup -`
The multi-pricing group for the item if the pricing method utilizes a multi-group pricing method.

isItemRepeatAllowed

`public boolean isItemRepeatAllowed()`

Returns whether selling another item by using item repeat is allowed for this item.

Returns:

boolean

setItemRepeatAllowed

`public void setItemRepeatAllowed(boolean repeatAllowed)`

Set whether selling another item by using item repeat is allowed for this item.

Parameters:

`repeatAllowed`

isTaxable

`public boolean isTaxable()`

Indicates whether the item is taxable.

Returns:

boolean

setTaxable

`public void **setTaxable**(boolean taxable)`

Set whether the item is taxable.

Parameters:

`taxable`

getPricingMethod

`public java.lang.String **getPricingMethod**()`

Returns the pricing method for this item.

Returns:

String The pricing method.

setPricingMethod

`public void **setPricingMethod**(java.lang.String pricingMethod)`

Set the pricing method for this item.

Parameters:

`pricingMethod`

getRegularPrice

`public java.lang.String **getRegularPrice**()`

Returns the regular price for this item.

Returns:

String The regular price for the item.

setRegularPrice

`public void **setRegularPrice**(java.lang.String price)`

Set the regular price for this item.

Parameters:

`price -`
- The regular price for the item.

com.ibm.retail.AEF.automation

Interface LineItem

All Subinterfaces:
Tender, CreditTender, Points, Item, Discount, Coupon

public interface **LineItem**
extends java.rmi.Remote

LineItem is an interface which represents any of the line items in a transaction (i.e., item, discount, tender, etc.).

Method Summary

Identifier	`getIdentifier()` Returns the Identifier for this line item.
LineItemInfo	`getInfo()` Returns the information for this line item.
AEFSession	`getSession()` Get the AEFSession object.
void	`setIdentifier(Identifier identifier)` Sets the Identifier for this line item.
void	`setInfo(LineItemInfo info)` Sets the information for this line item.
void	`setSession(AEFSession session)` Set the AEFSession object.
java.util.ArrayList	`voidLineItem()` Voids the line item.

Methods

voidLineItem

```
public java.util.ArrayList voidLineItem()
                                 throws AEFException,
                                        java.rmi.RemoteException
```

Voids the line item. Use to remove a tender or item from a transaction. Note that this method will attempt to void a line item which matches the characteristics of this line item. There is no guarantee that this actual line item is the one which will be voided. Note that this method will always throw an AEFException with an error code of AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.NOT_IMPLEMENTED. It is up to subclasses to provide working implementations of this method. Currently, only LineItems which are Items may be voided.

Returns:

ArrayList An array of one or more line items which were voided as a result of the call. Actual items will implement the "LineItem" interface.

(continued from last page)

Exceptions:

`java.rmi.RemoteException`

`AEFException -`

AEFException Error Codes:
AEFConst.APPLICATION_LIMIT_EXCEEDED,
AEFConst.FOODSTAMP_TENDER_IN_EXCESS_OF_FOODSTAMP_BALANCE
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.ITEM_LIMIT
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.NEGATIVE_TRANSACTION_BALANCE
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.TRANSACTION_LIMIT
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.TRANSACTION_TOTAL_TOO_LARGE
AEFConst.CONFIG_ERROR, AEFConst.FACTORY_ERROR
AEFConst.INVALID_ARGUMENT, AEFConst.EXTENDED_PRICE_TOO_LARGE
AEFConst.INVALID_MANAGER_OVERRIDE_NUMBER, AEFConst.NONE
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_NOT_REMOVED
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_REQUIRED
AEFConst.MANAGER_OVERRIDE_REQUIRED, AEFConst.NONE
AEFConst.POS_APP_FAILURE, AEFConst.INVALID_APPLICATION_DATA
AEFConst.POS_APP_FAILURE, AEFConst.FILE_IO_ERROR
AEFConst.POS_APP_FAILURE, AEFConst.OUT_OF_MEMORY
AEFConst.POS_APP_FAILURE, AEFConst.UNRECOGNIZED_PRINT_CHARACTERS
AEFConst.PRINTER_ERROR, AEFConst.COVER_OPEN
AEFConst.PRINTER_ERROR, AEFConst.NONE
AEFConst.PRINTER_ERROR, AEFConst.PAPER_LOW
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.NOT_APPROVED_BY_PAYMENT_SYSTEM
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.PAYMENT_SYSTEM_OFFLINE
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.TRANSACTION_NOT_ACTIVE
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.NOT_IMPLEMENTED
AEFConst.RETURN_COUPON_BEFORE_ITEM_VOID, AEFConst.NONE
AEFConst.SCALE_ERROR, AEFConst.NONE
AEFConst.VOID_MUST_MATCH_PREVIOUS, AEFConst.NONE
Common Errors

getInfo

```
public LineItemInfo getInfo()
                    throws java.rmi.RemoteException
```

Returns the information for this line item.

Returns:

LineItemInfo

Exceptions:

`RemoteException`

setInfo

```
public void setInfo(LineItemInfo info)
            throws java.rmi.RemoteException
```

Sets the information for this line item.

Parameters:

`LineItemInfo`

Exceptions:

`RemoteException`

setSession

```
public void setSession(AEFSession session)
               throws java.rmi.RemoteException
```

Set the AEFSession object.

(continued from last page)

Parameters:
`AEFSession`

Exceptions:
`RemoteException`

getSession

```
public AEFSession getSession()
                      throws java.rmi.RemoteException
```

Get the AEFSession object.

Parameters:
`AEFSession`

Exceptions:
`RemoteException`

getIdentifier

```
public Identifier getIdentifier()
                          throws java.rmi.RemoteException
```

Returns the Identifier for this line item.

Returns:

Identifier

Exceptions:
`RemoteException`

setIdentifier

```
public void setIdentifier(Identifier identifier)
                    throws java.rmi.RemoteException
```

Sets the Identifier for this line item.

Parameters:
`Identifier`

Exceptions:
`RemoteException`

com.ibm.retail.AEF.automation

Interface LineItemInfo

All Superinterfaces:
BaseInfo

All Subinterfaces:
TenderInfo, PointsInfo, ItemInfo, DiscountInfo, CreditInfo, CouponInfo

---

public interface **LineItemInfo**
extends BaseInfo

LineItemInfo is an interface which represents data for any of the line items in a transaction (i.e., item, discount, tender, etc.).

Field Summary

static java.lang.String	CLASS_KEY

Method Summary

void	addRawReceiptLine(java.lang.String line) Adds a raw print line for the line item.
void	addReceiptLine(java.lang.String line) Adds a print line for the line item.
java.lang.String	getDescription() Get the description
java.util.Collection	getRawReceiptLines() Returns the raw receipt print lines associated with this line item.
java.util.Collection	getReceiptLines() Returns the receipt print lines associated with this line item.
Transaction	getTransaction() Get the transaction
boolean	isDeposit() Indicates whether the line item is a deposit.
boolean	isRefunded() Indicates whether the line item has been refunded.
boolean	isVoided() Indicates whether the line item has been voided.
void	setDeposit(boolean flag) Sets whether the line item is a deposit.

void	`setDescription(java.lang.String value)` Set the description
void	`setRawReceiptLines(java.util.Collection lines)` Sets the raw receipt print lines associated with this line item.
void	`setReceiptLines(java.util.Collection lines)` Sets the receipt print lines associated with this line item.
void	`setRefunded(boolean flag)` Sets whether the line item has been refunded.
void	`setTransaction(Transaction trans)` Set the description
void	`setVoided(boolean flag)` Sets whether the line item has been voided.

Fields

CLASS_KEY

`public static final java.lang.String` **`CLASS_KEY`**

Methods

isVoided

`public boolean` **`isVoided`**`()`

Indicates whether the line item has been voided.

Returns:

boolean True if the line item has been voided.

setVoided

`public void` **`setVoided`**`(boolean flag)`

Sets whether the line item has been voided.

Parameters:

`boolean -`
True if the line item has been voided.

getReceiptLines

`public java.util.Collection` **`getReceiptLines`**`()`

Returns the receipt print lines associated with this line item.

Returns:

(continued from last page)

Collection A collection of Strings, each representing a print line.

setReceiptLines

`public void **setReceiptLines**(java.util.Collection lines)`

Sets the receipt print lines associated with this line item.

Parameters:

`Collection -`
A collection of strings representing the print lines for the line item.

addReceiptLine

`public void **addReceiptLine**(java.lang.String line)`

Adds a print line for the line item.

Parameters:

`String -`
The print line to be added.

getRawReceiptLines

`public java.util.Collection **getRawReceiptLines**()`

Returns the raw receipt print lines associated with this line item.

Returns:

Collection A collection of Strings, each representing a raw print line.

setRawReceiptLines

`public void **setRawReceiptLines**(java.util.Collection lines)`

Sets the raw receipt print lines associated with this line item.

Parameters:

`Collection -`
A collection of strings representing the print lines for the line item.

addRawReceiptLine

`public void **addRawReceiptLine**(java.lang.String line)`

Adds a raw print line for the line item.

Parameters:

`String -`
The raw print line to be added.

getDescription

`public java.lang.String **getDescription**()`

Get the description

Returns:

description

(continued from last page)

setDescription

`public void **setDescription**(java.lang.String value)`

Set the description

Parameters:

`description`

getTransaction

`public Transaction **getTransaction**()`

Get the transaction

Returns:

Transaction

setTransaction

`public void **setTransaction**(Transaction trans)`

Set the description

Parameters:

`Transaction`

isRefunded

`public boolean **isRefunded**()`

Indicates whether the line item has been refunded.

Returns:

boolean True if the line item has been refunded.

setRefunded

`public void **setRefunded**(boolean flag)`

Sets whether the line item has been refunded.

Parameters:

`boolean -`
True if the line item has been refunded.

isDeposit

`public boolean **isDeposit**()`

Indicates whether the line item is a deposit.

Returns:

boolean True if the line item is a deposit.

setDeposit

`public void **setDeposit**(boolean flag)`

Sets whether the line item is a deposit.

(continued from last page)

Parameters:

`boolean -`
True if the line item is a deposit.

com.ibm.retail.AEF.automation

Interface LoyaltyIdentifier

All Superinterfaces:
Identifier

All Known Implementing Classes:
LoyaltyIdentifierImpl

public interface **LoyaltyIdentifier**
extends Identifier

LoyaltyIdentifier is an interface for classes which contain the information to identify a loyalty customer.

Field Summary

static java.lang.String	ALTERNATE_TYPE
static java.lang.String	CLASS_KEY
static java.lang.String	EXPIRATION_DATE
static java.lang.String	ID
static java.lang.String	PRIMARY_TYPE
static java.lang.String	TYPE

Method Summary

java.lang.String	getExpirationDate() Gets the expiration date.
java.lang.String	getID() Gets the loyalty ID.
java.lang.String	getType() Gets the loyalty ID type.
void	setExpirationDate(java.lang.String date) Sets the expiration date.
void	setID(java.lang.String ID) Sets the loyalty ID.
void	setType(java.lang.String type) Sets the loyalty ID type.

Fields

CLASS_KEY

`public static final java.lang.String` **`CLASS_KEY`**

ID

`public static final java.lang.String` **`ID`**

TYPE

`public static final java.lang.String` **`TYPE`**

EXPIRATION_DATE

`public static final java.lang.String` **`EXPIRATION_DATE`**

PRIMARY_TYPE

`public static final java.lang.String` **`PRIMARY_TYPE`**

ALTERNATE_TYPE

`public static final java.lang.String` **`ALTERNATE_TYPE`**

Methods

setID

`public void` **`setID`**`(java.lang.String ID)`

Sets the loyalty ID.

Parameters:

`ID`

setType

`public void` **`setType`**`(java.lang.String type)`

Sets the loyalty ID type.

Parameters:

`type` -
(usually 'primary' or 'alternate').

setExpirationDate

`public void **setExpirationDate**(java.lang.String date)`

Sets the expiration date.

Parameters:

`expDate`

getID

`public java.lang.String **getID**()`

Gets the loyalty ID.

Returns:

String

getType

`public java.lang.String **getType**()`

Gets the loyalty ID type.

Returns:

String (usually 'primary' or 'alternate').

getExpirationDate

`public java.lang.String **getExpirationDate**()`

Gets the expiration date.

Returns:

String

com.ibm.retail.AEF.automation

Class LoyaltyIdentifierImpl

```
java.lang.Object
  |-java.util.AbstractMap
    |-java.util.HashMap
      |-com.ibm.retail.AEF.automation.IdentifierImpl
        |-com.ibm.retail.AEF.automation.LoyaltyIdentifierImpl
```

All Implemented interfaces:
LoyaltyIdentifier, java.util.Map, java.io.Serializable, java.lang.Cloneable, java.util.Map, java.io.Serializable, Identifier

---

public class **LoyaltyIdentifierImpl**
extends IdentifierImpl
implements Identifier, java.io.Serializable, java.util.Map, java.lang.Cloneable, java.io.Serializable, java.util.Map, LoyaltyIdentifier

LoyaltyIdentifierImpl is a class which contains tender information.

Constructor Summary

`LoyaltyIdentifierImpl()` Constructor.
`LoyaltyIdentifierImpl(java.lang.String ID,java.lang.String ID)` Constructor.
`LoyaltyIdentifierImpl(java.lang.String ID)` Constructor.
`LoyaltyIdentifierImpl(java.lang.String ID,java.lang.String ID,java.lang.String ID)` Constructor.

Method Summary

`java.lang.String`	`getExpirationDate()` Gets the expiration date.
`java.lang.String`	`getID()` Gets the loyalty ID.
`java.lang.String`	`getType()` Gets the loyalty ID type.
`void`	`setExpirationDate(java.lang.String expDate)` Sets the expiration date.

void	`setID(java.lang.String ID)` Sets the loyalty ID.
void	`setType(java.lang.String type)` Sets the loyalty ID type.

Methods inherited from : class `com.ibm.retail.AEF.automation.IdentifierImpl`
`toString`

Methods inherited from : class `java.util.HashMap`
`clear, clone, containsKey, containsValue, entrySet, get, isEmpty, keySet, put, putAll, remove, size, values`

Methods inherited from : class `java.util.AbstractMap`
`clear, clone, containsKey, containsValue, entrySet, equals, get, hashCode, isEmpty, keySet, put, putAll, remove, size, toString, values`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

LoyaltyIdentifierImpl

`public` **`LoyaltyIdentifierImpl`**`()`

Constructor.

LoyaltyIdentifierImpl

```
public LoyaltyIdentifierImpl(java.lang.String ID,
                             java.lang.String type)
```

Constructor.

Parameters:

`ID -`
The customer's loyalty ID.
`type -`
The ID type (usually LoyaltyIdentifier.PRIMARY_TYPE or LoyaltyIdentifier.ALTERNATE_TYPE)

LoyaltyIdentifierImpl

`public` **`LoyaltyIdentifierImpl`**`(java.lang.String ID)`

Constructor.

Parameters:

`ID -`
The customer's loyalty ID (the type defaults to LoyaltyIdentifier.PRIMARY_TYPE)

LoyaltyIdentifierImpl

```
public LoyaltyIdentifierImpl(java.lang.String ID,
                             java.lang.String type,
                             java.lang.String expDate)
```

Constructor.

Parameters:

`ID -`
The customer's loyalty ID.
`type -`
The ID type (usually LoyaltyIdentifier.PRIMARY_TYPE or LoyaltyIdentifier.ALTERNATE_TYPE)
`expDate -`
The loyalty number expiration date.

Methods

setID

```
public void setID(java.lang.String ID)
```

Sets the loyalty ID.

Parameters:

`ID`

setType

```
public void setType(java.lang.String type)
```

Sets the loyalty ID type.

Parameters:

`type -`
(usually 'primary' or 'alternate').

setExpirationDate

```
public void setExpirationDate(java.lang.String expDate)
```

Sets the expiration date.

Parameters:

`expDate`

getID

```
public java.lang.String getID()
```

Gets the loyalty ID.

Returns:

String

getType

```
public java.lang.String getType()
```

Gets the loyalty ID type.

(continued from last page)

Returns:

String (usually 'primary' or 'alternate').

getExpirationDate

`public java.lang.String` **`getExpirationDate`**`()`

Gets the expiration date.

Returns:

String

com.ibm.retail.AEF.automation

Interface MSRCreditIdentifier

All Superinterfaces:
TenderIdentifier, Identifier

All Known Implementing Classes:
MSRCreditIdentifierImpl

---

public interface **MSRCreditIdentifier**
extends TenderIdentifier

MSRCreditIdentifier is an interface for holding arguments required for a credit tender. This interface is used instead of CreditIdentifier whenever actual MSR track data for the credit card is available.

Field Summary

static java.lang.String	AMEX
static java.lang.String	CARD_TYPE
static java.lang.String	CARTE_BLANCHE
static java.lang.String	CLASS_KEY
static java.lang.String	DINERS_CLUB
static java.lang.String	DISCOVER
static java.lang.String	ENROUTE
static java.lang.String	JCB
static java.lang.String	MASTERCARD
static java.lang.String	TENDER_KEY
static java.lang.String	TRACK_1_DATA
static java.lang.String	TRACK_2_DATA
static java.lang.String	TRACK_3_DATA
static java.lang.String	UNKNOWN
static java.lang.String	VISA

Method Summary

java.lang.String	getCardType() Gets the card type.
byte[]	getTrack1Data() Gets the track 1 data.
byte[]	getTrack2Data() Gets the track 2 data.
byte[]	getTrack3Data() Gets the track 3 data.
void	setCardType(java.lang.String cardType) Sets the card type.
void	setTrack1Data(byte[] track1Data) Sets the track 1 data.
void	setTrack2Data(byte[] track2Data) Sets the track 2 data.
void	setTrack3Data(byte[] track1Data) Sets the track 3 data.

Fields

CLASS_KEY

public static final java.lang.String **CLASS_KEY**

TENDER_KEY

public static final java.lang.String **TENDER_KEY**

CARD_TYPE

public static final java.lang.String **CARD_TYPE**

TRACK_1_DATA

public static final java.lang.String **TRACK_1_DATA**

(continued from last page)

TRACK_2_DATA

public static final java.lang.String **TRACK_2_DATA**

TRACK_3_DATA

public static final java.lang.String **TRACK_3_DATA**

MASTERCARD

public static final java.lang.String **MASTERCARD**

VISA

public static final java.lang.String **VISA**

AMEX

public static final java.lang.String **AMEX**

DINERS_CLUB

public static final java.lang.String **DINERS_CLUB**

DISCOVER

public static final java.lang.String **DISCOVER**

ENROUTE

public static final java.lang.String **ENROUTE**

JCB

public static final java.lang.String **JCB**

CARTE_BLANCHE

public static final java.lang.String **CARTE_BLANCHE**

UNKNOWN

public static final java.lang.String **UNKNOWN**

Methods

setCardType

`public void **setCardType**(java.lang.String cardType)`

Sets the card type.

Parameters:

`cardType`

setTrack1Data

`public void **setTrack1Data**(byte[] track1Data)`

Sets the track 1 data.

Parameters:

`track1Data -`
A byte array containing the track 1 data (without sentinels). Set to null or a zero length byte array to indicate no track 1 data.

setTrack2Data

`public void **setTrack2Data**(byte[] track2Data)`

Sets the track 2 data.

Parameters:

`track2Data -`
A byte array containing the track 2 data (without sentinels). Set to null or a zero length byte array to indicate no track 2 data.

setTrack3Data

`public void **setTrack3Data**(byte[] track1Data)`

Sets the track 3 data.

Parameters:

`track3Data -`
A byte array containing the track 3 data (without sentinels). Set to null or a zero length byte array to indicate no track 3 data.

getCardType

`public java.lang.String **getCardType**()`

Gets the card type.

Returns:

String (or null if unknown)

getTrack1Data

`public byte[] **getTrack1Data**()`

Gets the track 1 data.

Returns:

(continued from last page)

byte[]

getTrack2Data

`public byte[] getTrack2Data()`

Gets the track 2 data.

Returns:

byte[]

getTrack3Data

`public byte[] getTrack3Data()`

Gets the track 3 data.

Returns:

byte[]

com.ibm.retail.AEF.automation

Class MSRCreditIdentifierImpl

```
java.lang.Object
  +-java.util.AbstractMap
    +-java.util.HashMap
      +-com.ibm.retail.AEF.automation.IdentifierImpl
        +-com.ibm.retail.AEF.automation.TenderIdentifierImpl
          +-com.ibm.retail.AEF.automation.MSRCreditIdentifierImpl
```

All Implemented interfaces:
MSRCreditIdentifier, java.util.Map, java.io.Serializable, java.lang.Cloneable, java.util.Map, java.io.Serializable, Identifier, TenderIdentifier

---

public class **MSRCreditIdentifierImpl**
extends TenderIdentifierImpl
implements TenderIdentifier, Identifier, java.io.Serializable, java.util.Map, java.lang.Cloneable, java.io.Serializable, java.util.Map, MSRCreditIdentifier

MSRCreditIdentifierImpl is a class for holding arguments required for a credit tender. This interface is used instead of CreditIdentifierImpl whenever actual MSR track data for the credit card is available.

Constructor Summary

`MSRCreditIdentifierImpl()` Constructor.
`MSRCreditIdentifierImpl(java.lang.String amount,java.lang.String amount,byte[] amount,byte[] amount,byte[] amount)` Constructor.
`MSRCreditIdentifierImpl(java.lang.String amount,java.lang.String amount,java.lang.String amount,java.lang.String amount,java.lang.String amount)` Constructor.

Method Summary

`java.lang.String`	`getCardType()` Gets the card type.
`java.lang.String`	`getKey()` Gets the tender key.
`byte[]`	`getTrack1Data()` Gets the track 1 data.
`byte[]`	`getTrack2Data()` Gets the track 2 data.

byte[]	`getTrack3Data()` Gets the track 3 data.
void	`setAccountNumber(java.lang.String accountNumber)` Sets the account number.
void	`setCardType(java.lang.String cardType)` Sets the card type.
void	`setTrack1Data(byte[] track1Data)` Sets the track 1 data.
void	`setTrack2Data(byte[] track2Data)` Sets the track 2 data.
void	`setTrack3Data(byte[] track3Data)` Sets the track 3 data.

Methods inherited from : class `com.ibm.retail.AEF.automation.TenderIdentifierImpl`
`getAmount, setAmount`

Methods inherited from : class `com.ibm.retail.AEF.automation.IdentifierImpl`
`toString`

Methods inherited from : class `java.util.HashMap`
`clear, clone, containsKey, containsValue, entrySet, get, isEmpty, keySet, put, putAll, remove, size, values`

Methods inherited from : class `java.util.AbstractMap`
`clear, clone, containsKey, containsValue, entrySet, equals, get, hashCode, isEmpty, keySet, put, putAll, remove, size, toString, values`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

MSRCreditIdentifierImpl

`public` **`MSRCreditIdentifierImpl`**`()`

Constructor.

(continued from last page)

MSRCreditIdentifierImpl

```
public MSRCreditIdentifierImpl(java.lang.String amount,
                               java.lang.String cardType,
                               byte[] track1Data,
                               byte[] track2Data,
                               byte[] track3Data)
```

Constructor.

Parameters:

`amount` -
The amount of the tender.
`cardType` -
(pass MSRCreditIdentifier.UNKNOWN for applications which do not require the card type to be identified ahead of time)
`track1Data` -
A byte array with the track 1 data from the card. Use null or a zero length array to indicate no track 1 data. Note that the data should not include the sentinels.
`track2Data` -
A byte array with the track 2 data from the card. Use null or a zero length array to indicate no track 2 data. Note that the data should not include the sentinels.
`track3Data` -
A byte array with the track 3 data from the card. Use null or a zero length array to indicate no track 3 data. Note that the data should not include the sentinels.

MSRCreditIdentifierImpl

```
public MSRCreditIdentifierImpl(java.lang.String amount,
                               java.lang.String cardType,
                               java.lang.String track1Data,
                               java.lang.String track2Data,
                               java.lang.String track3Data)
```

Constructor.

Parameters:

`amount` -
The amount of the tender.
`cardType` -
(pass MSRCreditIdentifier.UNKNOWN for applications which do not require the card type to be identified ahead of time)
`track1Data` -
A String with the track 1 data from the card. Use null or a zero length String to indicate no track 1 data. Note that the data should not include the sentinels.
`track2Data` -
A String with the track 2 data from the card. Use null or a zero length String to indicate no track 2 data. Note that the data should not include the sentinels.
`track3Data` -
A String with the track 3 data from the card. Use null or a zero length String to indicate no track 3 data. Note that the data should not include the sentinels.

Methods

setAccountNumber

```
public void setAccountNumber(java.lang.String accountNumber)
```

Sets the account number.

Parameters:

`accountNumber`

(continued from last page)

setCardType

`public void **setCardType**(java.lang.String cardType)`

Sets the card type.

Parameters:

`cardType`

setTrack1Data

`public void **setTrack1Data**(byte[] track1Data)`

Sets the track 1 data.

Parameters:

`track1Data -`
A byte array containing the track 1 data (without sentinels). Set to null or a zero length byte array to indicate no track 1 data.

setTrack2Data

`public void **setTrack2Data**(byte[] track2Data)`

Sets the track 2 data.

Parameters:

`track2Data -`
A byte array containing the track 2 data (without sentinels). Set to null or a zero length byte array to indicate no track 2 data.

setTrack3Data

`public void **setTrack3Data**(byte[] track3Data)`

Sets the track 3 data.

Parameters:

`track3Data -`
A byte array containing the track 3 data (without sentinels). Set to null or a zero length byte array to indicate no track 3 data.

getCardType

`public java.lang.String **getCardType**()`

Gets the card type.

Returns:

String (or null if unknown)

getTrack1Data

`public byte[] **getTrack1Data**()`

Gets the track 1 data.

Returns:

byte[]

(continued from last page)

getTrack2Data

`public byte[] getTrack2Data()`

Gets the track 2 data.

Returns:

byte[]

getTrack3Data

`public byte[] getTrack3Data()`

Gets the track 3 data.

Returns:

byte[]

getKey

`public java.lang.String getKey()`

Gets the tender key. The tender key is used to determine the keys in classes.properties chain for the classes used to add a tender or void a tender. For example, if the method returns "Cash", then "AddCashTenderAction" is the key used to determine the action class for adding the tender to a transaction.

Returns:

String

com.ibm.retail.AEF.automation

Interface Operator

public interface **Operator**
extends java.rmi.Remote

Represents an operator who is logged onto an Application.

Method Summary

`OperatorAuthorization`	`getAuthorization()` Returns the authorization object of the operator as determined by auhtorization settings in the POS application.
`OperatorInfo`	`getInfo()` Returns the information for the operator as determined by the POS application.

Methods

getInfo

```
public OperatorInfo getInfo()
                    throws java.rmi.RemoteException
```

Returns the information for the operator as determined by the POS application.

Returns:

OperatorInfo

Exceptions:

`java.rmi.RemoteException`

getAuthorization

```
public OperatorAuthorization getAuthorization()
                                 throws java.rmi.RemoteException
```

Returns the authorization object of the operator as determined by auhtorization settings in the POS application.

Returns:

OperatorAuthorization The authorization privileges of the operator

Exceptions:

`java.rmi.RemoteException`

com.ibm.retail.AEF.automation

Interface OperatorAuthorization

All Superinterfaces:
BaseInfo

public interface **OperatorAuthorization**
extends BaseInfo

Represents the application authorization of an operator.

Field Summary

static java.lang.String	CLASS_KEY

Method Summary

boolean	isAuthorized(java.lang.String AuthProperty) Indicates wheter the operator is authorized to perform the function represented by the property name.
void	setAuthorized(java.lang.String AuthProperty,boolean AuthProperty) Sets whether the operator is authorized with the privilege indicated by the property name.

Fields

CLASS_KEY

```
public static final java.lang.String CLASS_KEY
```

Methods

isAuthorized

```
public boolean isAuthorized(java.lang.String AuthProperty)
                      throws AEFException
```

Indicates wheter the operator is authorized to perform the function represented by the property name.

Parameters:
AuthProperty -
The name of the authorization property.

Returns:

boolean True if authorized, false otherwise.

Exceptions:

(continued from last page)

`AEFException -`
AEFException Error Codes:
AEFConst.INVALID_PROPERTY_VALUE
AEFConst.NO_SUCH_PROPERTY

setAuthorized

```
public void setAuthorized(java.lang.String AuthProperty,
                          boolean flag)
```

Sets whether the operator is authorized with the privilege indicated by the property name.

Parameters:

`String -`
The name of the authorization property.
`boolean -`
True if authorized, false otherwise.

com.ibm.retail.AEF.automation

Interface OperatorIdentifier

All Superinterfaces:
Identifier

All Known Implementing Classes:
OperatorIdentifierImpl

public interface **OperatorIdentifier**
extends Identifier

OperatorIdentifierImpl is a class which contains information for operator logon.

Field Summary

static java.lang.String	CLASS_KEY
static java.lang.String	ID
static java.lang.String	NEW_PASSWORD
static java.lang.String	PASSWORD

Method Summary

java.lang.String	getID() Gets the operator ID.
java.lang.String	getNewPassword() Gets the new password for a change password operation.
java.lang.String	getPassword() Gets the operator logon password.
void	setID(java.lang.String id) Sets the operator ID.
void	setNewPassword(java.lang.String newPassword) Sets the new operator password for a change password operation upon logon.
void	setPassword(java.lang.String password) Sets the operator password for the logon.

Fields

(continued from last page)

CLASS_KEY

`public static final java.lang.String **CLASS_KEY**`

ID

`public static final java.lang.String **ID**`

PASSWORD

`public static final java.lang.String **PASSWORD**`

NEW_PASSWORD

`public static final java.lang.String **NEW_PASSWORD**`

Methods

setID

`public void **setID**(java.lang.String id)`

Sets the operator ID.

Parameters:

`ID`

setPassword

`public void **setPassword**(java.lang.String password)`

Sets the operator password for the logon.

Parameters:

`password`

setNewPassword

`public void **setNewPassword**(java.lang.String newPassword)`

Sets the new operator password for a change password operation upon logon.

Parameters:

`newPassword`

getID

`public java.lang.String **getID**()`

Gets the operator ID.

Returns:

String

getPassword

`public java.lang.String **getPassword**()`

Gets the operator logon password.

Returns:

String

getNewPassword

`public java.lang.String **getNewPassword**()`

Gets the new password for a change password operation.

Returns:

String

com.ibm.retail.AEF.automation

Class OperatorIdentifierImpl

```
java.lang.Object
  +-java.util.AbstractMap
    +-java.util.HashMap
      +-com.ibm.retail.AEF.automation.IdentifierImpl
        +-com.ibm.retail.AEF.automation.OperatorIdentifierImpl
```

All Implemented interfaces:
OperatorIdentifier, java.util.Map, java.io.Serializable, java.lang.Cloneable, java.util.Map, java.io.Serializable, Identifier

---

public class **OperatorIdentifierImpl**
extends IdentifierImpl
implements Identifier, java.io.Serializable, java.util.Map, java.lang.Cloneable, java.io.Serializable, java.util.Map, OperatorIdentifier

OperatorIdentifierImpl is a class which contains information for operator logon.

Constructor Summary

`OperatorIdentifierImpl()` Constructor.
`OperatorIdentifierImpl(java.lang.String id,java.lang.String id)` Constructor.
`OperatorIdentifierImpl(java.lang.String id,java.lang.String id,java.lang.String id)` Constructor.

Method Summary

`java.lang.String`	`getID()` Gets the operator ID.
`java.lang.String`	`getNewPassword()` Gets the new password for a change password operation.
`java.lang.String`	`getPassword()` Gets the operator logon password.
`void`	`setID(java.lang.String id)` Sets the operator ID.
`void`	`setNewPassword(java.lang.String newPassword)` Sets the new operator password for a change password operation upon logon.

void	`setPassword(java.lang.String password)` Sets the operator password for the logon.

Methods inherited from : class `com.ibm.retail.AEF.automation.IdentifierImpl`
`toString`

Methods inherited from : class `java.util.HashMap`
`clear, clone, containsKey, containsValue, entrySet, get, isEmpty, keySet, put, putAll, remove, size, values`

Methods inherited from : class `java.util.AbstractMap`
`clear, clone, containsKey, containsValue, entrySet, equals, get, hashCode, isEmpty, keySet, put, putAll, remove, size, toString, values`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

OperatorIdentifierImpl

```
public OperatorIdentifierImpl()
```

Constructor.

OperatorIdentifierImpl

```
public OperatorIdentifierImpl(java.lang.String id,
                              java.lang.String password)
```

Constructor.

Parameters:

`id` -
The operator id.
`password` -
The operator's password.

OperatorIdentifierImpl

```
public OperatorIdentifierImpl(java.lang.String id,
                              java.lang.String password,
                              java.lang.String newPassword)
```

Constructor.

Parameters:

`id` -
The operator id.
`password` -
The operator's password.

(continued from last page)

`newPassword` -
The operator's new password (for a change password upon logon).

Methods

setID

`public void **setID**(java.lang.String id)`

Sets the operator ID.

Parameters:

`ID`

setPassword

`public void **setPassword**(java.lang.String password)`

Sets the operator password for the logon.

Parameters:

`password`

setNewPassword

`public void **setNewPassword**(java.lang.String newPassword)`

Sets the new operator password for a change password operation upon logon.

Parameters:

`newPassword`

getID

`public java.lang.String **getID**()`

Gets the operator ID.

Returns:

String

getPassword

`public java.lang.String **getPassword**()`

Gets the operator logon password.

Returns:

String

getNewPassword

`public java.lang.String **getNewPassword**()`

Gets the new password for a change password operation.

Returns:

String

com.ibm.retail.AEF.automation

Interface OperatorInfo

All Superinterfaces:
BaseInfo

public interface **OperatorInfo**
extends BaseInfo

OperatorInfo objects contain the data for an Operator of the POS application.

Field Summary

static java.lang.String	CLASS_KEY

Method Summary

java.lang.String	getID() Returns the id of the operator as determined by the POS application.
java.lang.String	getName() Returns the name of the operator as determined by the POS application.
java.lang.String	getPassword() Returns the password of the operator.
void	setID(java.lang.String id) Sets the id of the operator as determined by the POS application.
void	setName(java.lang.String name) Sets the name of the operator as determined by the POS application.
void	setPassword(java.lang.String password) Sets the password of the operator.

Fields

CLASS_KEY

public static final java.lang.String **CLASS_KEY**

Methods

(continued from last page)

getName

`public java.lang.String getName()`

Returns the name of the operator as determined by the POS application. For example, and operator with an id of "302" may have a name of "Amanda Smith".

Returns:

String The long name of the operator

setName

`public void setName(java.lang.String name)`

Sets the name of the operator as determined by the POS application. For example, and operator with an id of "302" may have a name of "Amanda Smith".

Parameters:

`name -`
The long name of the operator

getID

`public java.lang.String getID()`

Returns the id of the operator as determined by the POS application.

Returns:

String The id of the operator

setID

`public void setID(java.lang.String id)`

Sets the id of the operator as determined by the POS application.

Parameters:

`id -`
The id of the operator

getPassword

`public java.lang.String getPassword()`

Returns the password of the operator. Note that the password will only be set if one of the POSAutomationProvider.logon() methods is used to log the operator on. The password might not automatically be set in the case where a register recovers from a power failure.

Returns:

String The password of the operator

setPassword

`public void setPassword(java.lang.String password)`

Sets the password of the operator. Note that calling this method will not cause the operator password to be modified in the POS application. In other words, this is not a "change password" function.

Parameters:

`password -`
The password of the operator

com.ibm.retail.AEF.automation

Class OrCondition

```
java.lang.Object
  |-com.ibm.retail.AEF.automation.AbstractCondition
    |-com.ibm.retail.AEF.automation.OrCondition
```

All Implemented interfaces:
EvaluateConditionListener, Condition, Condition

---

public class **OrCondition**
extends AbstractCondition
implements Condition, Condition, EvaluateConditionListener

OrCondition uses an array of PropertyConditions. When any of the conditions evaluates to true, then the lock associated with this condition is notified.

Field Summary

com.ibm.retail.AEF.automation.Condition[]	conditions

Fields inherited from : class com.ibm.retail.AEF.automation.AbstractCondition
counterLock, dataProvider, eligible, id, idCounter, index, listeners, lock, manager, session

Constructor Summary

OrCondition(Condition[] conditions) Constructs the condition

Method Summary

void	conditionEvaluated(Condition condition,boolean condition) Called by a condition when the condition is evaluated.
void	currentValues(java.util.HashMap values) Populate the HashMap with property name/current value pairs included in the condition.
boolean	evaluate() Examines each of the child conditions immediately and determines whether they are all true.
java.lang.String	explain() Returns a string explaining this condition.
Condition	makeOpposite() Makes a new condition which is the logical opposite of this condition.

void	`setConditionLock(ConditionLock lock)` Sets the condition lock associated with this object.
void	`setEligible(boolean flag)` Sets the condition eligible for evaluation.
void	`setPOSDataProvider(POSDataProvider dataProvider)` Sets the POSDataProvider associated with this condition.

Methods inherited from : class `com.ibm.retail.AEF.automation.AbstractCondition`
`addEvaluateListener, getID, getIndex, getNextID, getPOSDataProvider, getSession, getTerminalNumber, isEligible, notifyEvaluateConditionListeners, removeEvaluateListener, setConditionLock, setEligible, setID, setIndex, setLockManager, setPOSDataProvider, setSession`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

conditions

`protected com.ibm.retail.AEF.automation.Condition` **`conditions`**

Constructors

OrCondition

`public` **`OrCondition`**`(Condition[] conditions)`

Constructs the condition

Methods

setPOSDataProvider

`public void` **`setPOSDataProvider`**`(POSDataProvider dataProvider)`

Sets the POSDataProvider associated with this condition.

conditionEvaluated

```
public void conditionEvaluated(Condition condition,
                               boolean flag)
```

Called by a condition when the condition is evaluated.

setEligible

`public void` **`setEligible`**`(boolean flag)`

(continued from last page)

Sets the condition eligible for evaluation.

evaluate

`public boolean **evaluate**()`

Examines each of the child conditions immediately and determines whether they are all true.

Returns:

boolean True if the condition evaluates to true.

makeOpposite

`public Condition **makeOpposite**()`

Makes a new condition which is the logical opposite of this condition.

Returns:

Condition

explain

`public java.lang.String **explain**()`

Returns a string explaining this condition.

Returns:

String

currentValues

`public void **currentValues**(java.util.HashMap values)`

Populate the HashMap with property name/current value pairs included in the condition. If a key is already in the HashMap, the new value will not override the value already in the map for the key.

Parameters:

`values`

setConditionLock

`public void **setConditionLock**(ConditionLock lock)`

Sets the condition lock associated with this object.

Parameters:

`ConditionLock`

com.ibm.retail.AEF.automation

Interface Points

All Superinterfaces:
LineItem

public interface **Points**
extends LineItem

Points is an interface which encapsulates points being awarded or redeemed.

Field Summary

static java.lang.String	CLASS_KEY

Fields

CLASS_KEY

public static final java.lang.String **CLASS_KEY**

com.ibm.retail.AEF.automation

Interface PointsInfo

All Superinterfaces:
LineItemInfo, BaseInfo

public interface **PointsInfo**
extends LineItemInfo

PointsInfo is an interface which encapsulates the item information for an item which is sold in a transaction.

Field Summary

static java.lang.String	CLASS_KEY

Method Summary

java.util.Collection	getAdditionalPointsTotals() Get any additional points represented by the event.
java.lang.String	getPoints() Get the number of points represented by the event.
java.lang.String	getType() Get the type of points.
boolean	isAwarded() Indicates if the points are awarded.
boolean	isRedeemed() Indicates if the points are redeemed.
void	setAdditionalPointsTotals(java.util.Collection points) Set any additional points represented by the event.
void	setAwarded(boolean awarded) Determines if the points are awarded.
void	setPoints(java.lang.String points) Sets the number of points represented by the object.
void	setRedeemed(boolean redeemed) Determines if the points are redeemed.
void	setType(java.lang.String type) Sets the type of points.

(continued from last page)

Fields

CLASS_KEY

`public static final java.lang.String` **`CLASS_KEY`**

Methods

getType

`public java.lang.String` **`getType`**`()`

Get the type of points. For example, a enterprise may have "primary" points and secondary points such as "baby club" or "senior club".

Returns:

String

setType

`public void` **`setType`**`(java.lang.String type)`

Sets the type of points. For example, a enterprise may have "primary" points and secondary points such as "baby club" or "senior club".

Parameters:

`String`

isRedeemed

`public boolean` **`isRedeemed`**`()`

Indicates if the points are redeemed. If not redeemed, then they are awarded.

Returns:

boolean

setRedeemed

`public void` **`setRedeemed`**`(boolean redeemed)`

Determines if the points are redeemed. If not redeemed, then they are awarded.

Parameters:

`boolean`

isAwarded

`public boolean` **`isAwarded`**`()`

Indicates if the points are awarded. If not awarded, then they are redeemed.

Returns:

boolean

(continued from last page)

setAwarded

`public void **setAwarded**(boolean awarded)`

Determines if the points are awarded. If not awarded, then they are redeemed.

Parameters:

`boolean`

getPoints

`public java.lang.String **getPoints**()`

Get the number of points represented by the event.

Returns:

String

setPoints

`public void **setPoints**(java.lang.String points)`

Sets the number of points represented by the object.

Parameters:

`String`

getAdditionalPointsTotals

`public java.util.Collection **getAdditionalPointsTotals**()`

Get any additional points represented by the event. Addtional points may apply to clubs.

Returns:

Collection of PointsTotal objects

setAdditionalPointsTotals

`public void **setAdditionalPointsTotals**(java.util.Collection points)`

Set any additional points represented by the event. Addtional points may apply to clubs.

Parameters:

`Collection -`
of PointsTotal objects

com.ibm.retail.AEF.automation

Interface POSAutomationProvider

public interface **POSAutomationProvider**
extends java.rmi.Remote

POSAutomationProvider is an interface which encapsulates the automation APIs for a POS Application and makes it available to remote clients.

Field Summary

static java.lang.String	AGE_RESTRICT A property which may be used to control whether the AEF will allow age restricted items to be added to the transaction.
static java.lang.String	AUTO_MGR_OVERRIDE A property which may be used to control whether the AEF automatically performs manager overrides.
static java.lang.String	AUTO_OPR_OVERRIDE A property which may be used to control whether the AEF automatically performs operator overrides.
static java.lang.String	AUTO_TILL_EXCHANGE A property which determines whether the error handler will automatically perform a till exchange when prompted by the POS application.
static java.lang.String	ERROR_HANDLING_MODE A property which determines how the AEF deals with application errors and prompts.
static int	HANDLE_AUTOMATIC
static int	HANDLE_CALLBACK
static int	HANDLE_DEFAULT
static java.lang.String	MAX_ERRORS_TO_HANDLE A property which determines how many sequential application errors should be handled by the AEF.
static java.lang.String	MGR_OVERRIDE_NUMBER A property which contains the manager override number used for automatic overrides.
static int	SIGNON
static int	START_TRANSACTION

Method Summary

void	`cancelOverride()` Cancel an override in the POS Application.
void	`forceLogoff()` Logs an operator off the POS Application even if a transaction is in progress.
AEFErrorCallback	`getCallback()` Get the Callback object for the automation provider.
int	`getErrorHandlingMode()` Get the Error Handling Mode for this application.
int	`getMaxErrorToHandle()` Gets the maximum number of errors that the error handler will try to clear.
Operator	`getOperator()` Returns the currently logged on operator.
java.lang.String	`getProperty(java.lang.String property)` Gets the value of a configuration property within the automation provider.
AEFSession	`getSession()` Gets the AEFSession for this instance.
java.lang.String	`getTerminalNumber()` Retrieves the terminal number of the application.
boolean	`getTrainingMode()` Determines if the terminal is in training mode.
Transaction	`getTransaction()` Retrieves a transaction that was already in progress.
void	`initialize(int initialState)` Initialize the application to a known state (either signon, or start of a regular sales transaction).
void	`logoff()` Logs an operator off the POS Application.
Operator	`logon()` Logs an operator onto the POS application using the default id and password defined in applogon.
Operator	`logon(OperatorIdentifier operatorId)` Logs an operator onto the POS Application.
Operator	`logon(java.lang.String operatorID,java.lang.String operatorID)` Logs an operator onto the POS Application.
void	`managerOverride()` Send a manager override to the POS Application using the default manager number specified by the default manager override number property POSAutomationProvider.

void	`managerOverride(java.lang.String password)` Send a manager override to the POS Application.
void	`operatorOverride(java.lang.String password)` Send an operator override to the POSAutomationProvider.
java.lang.Object	`performAction(ActionRequest request)` Perform an action specified by an ActionRequest.
void	`registerCallback(AEFErrorCallback callback)` Set the AEFErrorCallback object for the automation provider.
SuspendedTransactionInfo[]	`retrieveSuspendedTransactionList()` Retrieves a list of suspended transaction info objects for all terminals returned, sorted by terminal number.
SuspendedTransactionInfo[]	`retrieveSuspendedTransactionList(int termNumber)` Retrieves a list of suspended transaction info objects for a given terminal
Transaction	`retrieveTransaction(int termNumber,int termNumber)` Retrieves a suspended transaction by terminal and transaction number.
void	`setOperator(Operator operator)` Sets the currently logged on operator.
void	`setProperty(java.lang.String[] properties,java.lang.String[] properties)` Sets multiple configuration property values within the automation provider.
void	`setProperty(java.lang.String property,java.lang.String property)` Sets a configuration property within the automation provider.
void	`setTrainingMode(boolean trainingMode)` Sets the training mode based on the trainingMode parameter.
void	`setTransaction(Transaction trans)` Sets the transaction in progress.
SalesTransaction	`startTransaction()` Start a regular sale transaction.
Transaction	`startTransaction(TransactionIdentifier transID)` Start a transaction of the requested type.
Transaction	`suspendCurrentTransaction()` Suspends the transaction in progress.
Transaction	`suspendCurrentTransaction(java.lang.String reasonCode)` Suspends the transaction in progress.
void	`voidCurrentTransaction()` Voids the transaction currently in progress.

(continued from last page)

Fields

AUTO_MGR_OVERRIDE

`public static final java.lang.String` **`AUTO_MGR_OVERRIDE`**

A property which may be used to control whether the AEF automatically performs manager overrides. Use the setProperty method to set this property true or false. Default value is defined in automation.properties.

MGR_OVERRIDE_NUMBER

`public static final java.lang.String` **`MGR_OVERRIDE_NUMBER`**

A property which contains the manager override number used for automatic overrides. Use the setProperty method to set this property value. Default value is defined in appautomation.properties.

AUTO_OPR_OVERRIDE

`public static final java.lang.String` **`AUTO_OPR_OVERRIDE`**

A property which may be used to control whether the AEF automatically performs operator overrides. Use the setProperty method to set this property true or false. Default value is defined in appautomation.properties.

AGE_RESTRICT

`public static final java.lang.String` **`AGE_RESTRICT`**

A property which may be used to control whether the AEF will allow age restricted items to be added to the transaction. If true, the AEF will attempt to answer any age verification automatically. Use the setProperty method to set this property true or false. Default value is defined in appautomation.properties.

ERROR_HANDLING_MODE

`public static final java.lang.String` **`ERROR_HANDLING_MODE`**

A property which determines how the AEF deals with application errors and prompts. May be set to POSAutomationProvider.HANDLE_AUTOMATIC or POSAutomationProvider.HANDLE_DEFAULT. In AUTOMATIC mode, the AEF will attempt to complete the automation call by clearing any errors and handling any prompts where possible. This mode is intended for use in an environment where there is no cashier present. In DEFAULT mode, the AEF will pause at errors and prompts to allow the operator to clear or provide the required input. Default value is defined in appautomation.properties.

MAX_ERRORS_TO_HANDLE

`public static final java.lang.String` **`MAX_ERRORS_TO_HANDLE`**

A property which determines how many sequential application errors should be handled by the AEF. The maximum number of errors will keep the automation provider from waiting indefinitely to complete an automation API if the application is in an error loop. Default value is defined in appautomation.properties.

AUTO_TILL_EXCHANGE

`public static final java.lang.String` **`AUTO_TILL_EXCHANGE`**

A property which determines whether the error handler will automatically perform a till exchange when prompted by the POS application.

HANDLE_AUTOMATIC

`public static final int` **`HANDLE_AUTOMATIC`**

(continued from last page)

HANDLE_DEFAULT

```
public static final int HANDLE_DEFAULT
```

HANDLE_CALLBACK

```
public static final int HANDLE_CALLBACK
```

SIGNON

```
public static final int SIGNON
```

START_TRANSACTION

```
public static final int START_TRANSACTION
```

Methods

logon

```
public Operator logon(OperatorIdentifier operatorId)
               throws java.rmi.RemoteException,
                      AEFException
```

Logs an operator onto the POS Application. Action ID = LogonAction

Parameters:

`operatorId -`
The identifier containing the logon arguments.

Returns:

Operator

Exceptions:

`RemoteException`

(continued from last page)

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.CASH_DRAWER_ERROR, AEFConst.NONE_AVAILABLE
AEFConst.INVALID_ARGUMENT, AEFConst.ID_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.INVALID_ID
AEFConst.INVALID_ARGUMENT, AEFConst.INVALID_PASSWORD
AEFConst.INVALID_ARGUMENT, AEFConst.NEW_PASSWORD_PROHIBITED
AEFConst.INVALID_ARGUMENT, AEFConst.PASSWORD_REQUIRED
AEFConst.INVALID_MANAGER_OVERRIDE_NUMBER, AEFConst.NONE
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_NOT_REMOVED
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_REQUIRED
AEFConst.MANAGER_OVERRIDE_REQUIRED, AEFConst.NONE
AEFConst.PASSWORD_EXPIRED, AEFConst.NONE
AEFConst.POS_APP_FAILURE, AEFConst.FILE_IO_ERROR
AEFConst.POS_APP_FAILURE, AEFConst.UNRECOGNIZED_PRINT_CHARACTERS
AEFConst.PRINTER_ERROR, AEFConst.COVER_OPEN
AEFConst.PRINTER_ERROR, AEFConst.NONE
AEFConst.PRINTER_ERROR, AEFConst.PAPER_LOW
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.ANOTHER_OPERATOR_SIGNED_ON
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.CLOSING_ACCOUNTING_PERIOD
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.OPERATOR_ALREADY_ACTIVE
AEFConst.UNCHECKED_EXCEPTION, AEFConst.NONE
Common Errors

logon

```
public Operator logon(java.lang.String operatorID,
                      java.lang.String password)
               throws java.rmi.RemoteException,
                      AEFException
```

Logs an operator onto the POS Application. Action ID = LogonAction

Parameters:

`operatorID -`
The ID assigned to the operator in the POS application.
`password -`
The password corresponding to the operator. Use a null or empty string if the password is not required.

Returns:

Operator

Exceptions:

`RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.CASH_DRAWER_ERROR, AEFConst.NONE_AVAILABLE
AEFConst.INVALID_ARGUMENT, AEFConst.ID_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.INVALID_ID
AEFConst.INVALID_ARGUMENT, AEFConst.INVALID_PASSWORD
AEFConst.INVALID_ARGUMENT, AEFConst.PASSWORD_REQUIRED
AEFConst.INVALID_MANAGER_OVERRIDE_NUMBER, AEFConst.NONE
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_NOT_REMOVED
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_REQUIRED
AEFConst.MANAGER_OVERRIDE_REQUIRED, AEFConst.NONE
AEFConst.PASSWORD_EXPIRED, AEFConst.NONE
AEFConst.POS_APP_FAILURE, AEFConst.FILE_IO_ERROR
AEFConst.POS_APP_FAILURE, AEFConst.UNRECOGNIZED_PRINT_CHARACTERS
AEFConst.PRINTER_ERROR, AEFConst.COVER_OPEN
AEFConst.PRINTER_ERROR, AEFConst.NONE
AEFConst.PRINTER_ERROR, AEFConst.PAPER_LOW
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.ANOTHER_OPERATOR_SIGNED_ON
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.CLOSING_ACCOUNTING_PERIOD
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.OPERATOR_ALREADY_ACTIVE
AEFConst.UNCHECKED_EXCEPTION, AEFConst.NONE
Common Errors

logon

```
public Operator logon()
                throws java.rmi.RemoteException,
                       AEFException
```

Logs an operator onto the POS application using the default id and password defined in applogon.properties for this virtual terminal. Action ID = LogonAction

Returns:

Operator

Exceptions:

`RemoteException`

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.CASH_DRAWER_ERROR, AEFConst.NONE_AVAILABLE
AEFConst.CONFIG_ERROR, AEFConst.NO_DEFAULT_ID_CONFIGURED
AEFConst.INVALID_ARGUMENT, AEFConst.INVALID_ID
AEFConst.INVALID_ARGUMENT, AEFConst.INVALID_PASSWORD
AEFConst.INVALID_MANAGER_OVERRIDE_NUMBER, AEFConst.NONE
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_NOT_REMOVED
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_REQUIRED
AEFConst.MANAGER_OVERRIDE_REQUIRED, AEFConst.NONE
AEFConst.PASSWORD_EXPIRED, AEFConst.NONE
AEFConst.POS_APP_FAILURE, AEFConst.FILE_IO_ERROR
AEFConst.POS_APP_FAILURE, AEFConst.UNRECOGNIZED_PRINT_CHARACTERS
AEFConst.PRINTER_ERROR, AEFConst.COVER_OPEN
AEFConst.PRINTER_ERROR, AEFConst.NONE
AEFConst.PRINTER_ERROR, AEFConst.PAPER_LOW
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.ANOTHER_OPERATOR_SIGNED_ON
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.CLOSING_ACCOUNTING_PERIOD
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.OPERATOR_ALREADY_ACTIVE
AEFConst.UNCHECKED_EXCEPTION, AEFConst.NONE
Common Errors

getOperator

```
public Operator getOperator()
                     throws java.rmi.RemoteException
```

Returns the currently logged on operator.

Returns:

Operator

Exceptions:

`RemoteException`

setOperator

```
public void setOperator(Operator operator)
                  throws java.rmi.RemoteException
```

Sets the currently logged on operator.

Parameters:

`Operator`

Exceptions:

`RemoteException`

getTerminalNumber

```
public java.lang.String getTerminalNumber()
                                    throws java.rmi.RemoteException
```

Retrieves the terminal number of the application.

Returns:

String The terminal number of the application.

Exceptions:

`RemoteException`

initialize

```
public void initialize(int initialState)
                 throws java.rmi.RemoteException,
                        AEFException
```

Initialize the application to a known state (either signon, or start of a regular sales transaction). The framework will perform the following steps: 1. Clear any error conditions. 2. Void any transactions in progress. 3. Signoff if requested state is signon. 4. Start a regular transaction if requested state is start transaction. Action ID = InitializeAction

Parameters:

`int -`
Requested initial state (either POSAutomationProvider.SIGNON or POSAutomationProvider.START_TRANSACTION).

Exceptions:

`RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.CASH_DRAWER_ERROR, AEFConst.NONE_AVAILABLE
AEFConst.INVALID_ARGUMENT, AEFConst.INVALID_INITIALIZE_STATE
AEFConst.INVALID_MANAGER_OVERRIDE_NUMBER, AEFConst.NONE
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_NOT_REMOVED
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_REQUIRED
AEFConst.MANAGER_OVERRIDE_REQUIRED
AEFConst.POS_APP_FAILURE, AEFConst.FILE_IO_ERROR
AEFConst.POS_APP_FAILURE, AEFConst.UNRECOGNIZED_PRINT_CHARACTERS
AEFConst.PRINTER_ERROR, AEFConst.COVER_OPEN
AEFConst.PRINTER_ERROR, AEFConst.NONE
AEFConst.PRINTER_ERROR, AEFConst.PAPER_LOW
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.CLOSING_ACCOUNTING_PERIOD
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.NONE
AEFConst.UNCHECKED_EXCEPTION, AEFConst.NONE
Common Errors

setProperty

```
public void setProperty(java.lang.String property,
                        java.lang.String value)
                 throws java.rmi.RemoteException,
                        AEFException
```

Sets a configuration property within the automation provider. Supported properties are determined in the automation.properties file.

Parameters:

`property -`
The property name to set.
`value -`
The property value.

(continued from last page)

Exceptions:

`RemoteException`

`AEFException -`
AEFException error codes are:
AEFConst.PROPERTY_NOT_SUPPORTED
AEFConst.INVALID_PROPERTY_VALUE

setProperty

```
public void setProperty(java.lang.String[] properties,
                        java.lang.String[] values)
                 throws java.rmi.RemoteException,
                        AEFException
```

Sets multiple configuration property values within the automation provider. Supported properties are determined in the automation.properties file.

Parameters:

`properties -`
An array of property names.

`values -`
An array of property values.

Exceptions:

`RemoteException`

`AEFException -`
AEFException error codes are:
AEFConst.INVALID_ARGUMENT
AEFConst.PROPERTY_NOT_SUPPORTED
AEFConst.INVALID_PROPERTY_VALUE

getProperty

```
public java.lang.String getProperty(java.lang.String property)
                              throws java.rmi.RemoteException,
                                     AEFException
```

Gets the value of a configuration property within the automation provider. Supported properties are determined in automation.properties.

Parameters:

`property -`
The property name to get.

Returns:

String The property value.

Exceptions:

`RemoteException`

`AEFException -`
AEFException error codes are:
AEFConst.PROPERTY_NOT_SUPPORTED

logoff

```
public void logoff()
            throws java.rmi.RemoteException,
                   AEFException
```

Logs an operator off the POS Application. Will generate an exception if a transaction is in progress. Action ID = LogoffAction

Exceptions:

`RemoteException`

(continued from last page)

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.CASH_DRAWER_ERROR, AEFConst.NONE_AVAILABLE
AEFConst.INVALID_MANAGER_OVERRIDE_NUMBER, AEFConst.NONE
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_NOT_REMOVED
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_REQUIRED
AEFConst.MANAGER_OVERRIDE_REQUIRED
AEFConst.POS_APP_FAILURE, AEFConst.FILE_IO_ERROR
AEFConst.POS_APP_FAILURE, AEFConst.UNRECOGNIZED_PRINT_CHARACTERS
AEFConst.PRINTER_ERROR, AEFConst.COVER_OPEN
AEFConst.PRINTER_ERROR, AEFConst.NONE
AEFConst.PRINTER_ERROR, AEFConst.PAPER_LOW
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.CLOSING_ACCOUNTING_PERIOD
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.NONE
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.TRANSACTION_IN_PROGRESS
AEFConst.UNCHECKED_EXCEPTION, AEFConst.NONE
Common Errors

forceLogoff

```
public void forceLogoff()
                 throws java.rmi.RemoteException,
                        AEFException
```

Logs an operator off the POS Application even if a transaction is in progress. If transaction is in progress, it is voided prior to logging off. Action ID = ForcedLogoffAction

Exceptions:

`RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.CASH_DRAWER_ERROR, AEFConst.NONE_AVAILABLE
AEFConst.INVALID_MANAGER_OVERRIDE_NUMBER, AEFConst.NONE
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_NOT_REMOVED
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_REQUIRED
AEFConst.MANAGER_OVERRIDE_REQUIRED
AEFConst.POS_APP_FAILURE, AEFConst.FILE_IO_ERROR
AEFConst.POS_APP_FAILURE, AEFConst.UNRECOGNIZED_PRINT_CHARACTERS
AEFConst.PRINTER_ERROR, AEFConst.COVER_OPEN
AEFConst.PRINTER_ERROR, AEFConst.NONE
AEFConst.PRINTER_ERROR, AEFConst.PAPER_LOW
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.CLOSING_ACCOUNTING_PERIOD
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.NONE
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.PAYMENT_SYSTEM_OFFLINE
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.TRANSACTION_CANNOT_BE_VOIDED
AEFConst.UNCHECKED_EXCEPTION, AEFConst.NONE
Common Errors

startTransaction

```
public Transaction startTransaction(TransactionIdentifier transID)
                              throws java.rmi.RemoteException,
                                     AEFException
```

Start a transaction of the requested type. Action ID = StartTransaction Note that some POS applications (such as IBM's Supermarket Application) will not actually start a transaction until the first item is sold. This API will return a transaction object, but the transaction id and date/time in the TransactionInfo object will be null until the first item is sold.

Parameters:

`transID -`
A transaction Identifier that describes the type of transaction to be started.
`return -`
Transaction The Transaction object which was started.

Exceptions:

`RemoteException`

(continued from last page)

`AEFException. -`
Among the possible AEFException error codes are:
AEFConst.CASH_DRAWER_ERROR, AEFConst.CLOSE_DRAWER
AEFConst.INVALID_ARGUMENT, AEFConst.INVALID_TRANSACTION_TYPE
AEFConst.INVALID_ARGUMENT, AEFConst.NONE
AEFConst.INVALID_ARGUMENT, AEFConst.TRANSACTION_TYPE_REQUIRED
AEFConst.INVALID_MANAGER_OVERRIDE_NUMBER, AEFConst.NONE
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_NOT_REMOVED
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_REQUIRED
AEFConst.MANAGER_OVERRIDE_REQUIRED
AEFConst.POS_APP_FAILURE, AEFConst.FILE_IO_ERROR
AEFConst.POS_APP_FAILURE, AEFConst.UNRECOGNIZED_PRINT_CHARACTERS
AEFConst.PRINTER_ERROR, AEFConst.COVER_OPEN
AEFConst.PRINTER_ERROR, AEFConst.NONE
AEFConst.PRINTER_ERROR, AEFConst.PAPER_LOW
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.CLOSING_ACCOUNTING_PERIOD
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.NONE
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.NOT_ALLOWED_TRAINING
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.TILL_EXCHANGE_NEEDED
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.TRANSACTION_ALREADY_IN_PROGRESS
AEFConst.UNCHECKED_EXCEPTION, AEFConst.NONE
Common Errors

---

startTransaction

```
public SalesTransaction startTransaction()
                                  throws java.rmi.RemoteException,
                                         AEFException
```

Start a regular sale transaction. Action ID = StartTransaction Note that some POS applications (such as IBM's Supermarket Application) will not actually start a transaction until the first item is sold. This API will return a transaction object, but the transaction id and date/time in the TransactionInfo object will be null until the first item is sold.

Returns:

SalesTransaction The SalesTransaction object which was started.

Exceptions:

`RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.CASH_DRAWER_ERROR, AEFConst.CLOSE_DRAWER
AEFConst.INVALID_MANAGER_OVERRIDE_NUMBER, AEFConst.NONE
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_NOT_REMOVED
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_REQUIRED
AEFConst.MANAGER_OVERRIDE_REQUIRED
AEFConst.POS_APP_FAILURE, AEFConst.FILE_IO_ERROR
AEFConst.POS_APP_FAILURE, AEFConst.UNRECOGNIZED_PRINT_CHARACTERS
AEFConst.PRINTER_ERROR, AEFConst.COVER_OPEN
AEFConst.PRINTER_ERROR, AEFConst.NONE
AEFConst.PRINTER_ERROR, AEFConst.PAPER_LOW
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.CLOSING_ACCOUNTING_PERIOD
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.TILL_EXCHANGE_NEEDED
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.TRANSACTION_ALREADY_IN_PROGRESS
AEFConst.UNCHECKED_EXCEPTION, AEFConst.NONE
Common Errors

---

getTransaction

```
public Transaction getTransaction()
                            throws java.rmi.RemoteException
```

Retrieves a transaction that was already in progress.

Returns:

Transaction The Transaction already in progress. If no transaction in progress, returns a null.

(continued from last page)

Exceptions:

`RemoteException`

setTransaction

```
public void setTransaction(Transaction trans)
                    throws java.rmi.RemoteException
```

Sets the transaction in progress. This method is intended to be called by the TransactionDetector.

Parameters:

`Transaction -`
The Transaction in progress.

Exceptions:

`RemoteException`

retrieveTransaction

```
public Transaction retrieveTransaction(int termNumber,
                                       int transNumber)
                                 throws java.rmi.RemoteException,
                                       AEFException
```

Retrieves a suspended transaction by terminal and transaction number. Action ID = RetrieveTransaction

Parameters:

`termNumber -`
Terminal number
`transNumber -`
Transaction number to retrieve.

Returns:

Transaction The suspended transaction.

Exceptions:

`RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.CONFIG_ERROR, AEFConst.FACTORY_ERROR
AEFConst.INVALID_ARGUMENT, AEFConst.NONE
AEFConst.INVALID_MANAGER_OVERRIDE_NUMBER, AEFConst.NONE
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_NOT_REMOVED
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_REQUIRED
AEFConst.MANAGER_OVERRIDE_REQUIRED
AEFConst.POS_APP_FAILURE, AEFConst.FILE_IO_ERROR
AEFConst.POS_APP_FAILURE, AEFConst.UNRECOGNIZED_PRINT_CHARACTERS
AEFConst.PRINTER_ERROR, AEFConst.COVER_OPEN
AEFConst.PRINTER_ERROR, AEFConst.NONE
AEFConst.PRINTER_ERROR, AEFConst.PAPER_LOW
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.NONE
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.NO_SUCH_SUSPENDED_TRANSACTION
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.NOT_ALLOWED_TRAINING
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.PICKUP_NEEDED
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.RETRIEVE_TRANSACTION_WHERE_SUSPENDED
AEFConst.PROCEDURE_NOT_ALLOWED,
AEFConst.SAME_OPERATOR_MUST_RETRIEVE_TRANSACTION
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.TILL_EXCHANGE_NEEDED
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.TRANSACTION_ALREADY_IN_PROGRESS
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.TRANSACTION_ALREADY_RETRIEVED
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.UNABLE_TO_RETRIEVE_TRANSACTION
AEFConst.UNCHECKED_EXCEPTION, AEFConst.NONE
Common Errors

(continued from last page)

suspendCurrentTransaction

```
public Transaction suspendCurrentTransaction(java.lang.String reasonCode)
                                        throws AEFException,
                                               java.rmi.RemoteException
```

Suspends the transaction in progress. Action ID = SuspendTransaction

Parameters:

`reasonCode -`
The reason for the transaction being suspended.

Returns:

Transaction The retrieved transaction.

Exceptions:

`RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.APPLICATION_LIMIT_EXCEEDED, SUSPENDED_TRANSACTION_LIMIT
AEFConst.CONFIG_ERROR, AEFConst.FACTORY_ERROR
AEFConst.INVALID_ARGUMENT, AEFConst.REASON_CODE_PROHIBITED
AEFConst.INVALID_MANAGER_OVERRIDE_NUMBER, AEFConst.NONE
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_NOT_REMOVED
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_REQUIRED
AEFConst.MANAGER_OVERRIDE_REQUIRED
AEFConst.POS_APP_FAILURE, AEFConst.FILE_IO_ERROR
AEFConst.POS_APP_FAILURE, AEFConst.UNRECOGNIZED_PRINT_CHARACTERS
AEFConst.PRINTER_ERROR, AEFConst.COVER_OPEN
AEFConst.PRINTER_ERROR, AEFConst.NONE
AEFConst.PRINTER_ERROR, AEFConst.PAPER_LOW
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.HOST_REQUEST_PENDING
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.NONE
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.TRANSACTION_NOT_ACTIVE
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.VALUE_CARDS_MUST_BE_VOIDED
AEFConst.UNCHECKED_EXCEPTION, AEFConst.NONE
Common Errors

suspendCurrentTransaction

```
public Transaction suspendCurrentTransaction()
                                        throws AEFException,
                                               java.rmi.RemoteException
```

Suspends the transaction in progress. Action ID = SuspendTransaction

Returns:

Transaction The retrieved transaction.

Exceptions:

`RemoteException`

(continued from last page)

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.APPLICATION_LIMIT_EXCEEDED, SUSPENDED_TRANSACTION_LIMIT
AEFConst.CONFIG_ERROR, AEFConst.FACTORY_ERROR
AEFConst.INVALID_MANAGER_OVERRIDE_NUMBER, AEFConst.NONE
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_NOT_REMOVED
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_REQUIRED
AEFConst.MANAGER_OVERRIDE_REQUIRED
AEFConst.POS_APP_FAILURE, AEFConst.FILE_IO_ERROR
AEFConst.POS_APP_FAILURE, AEFConst.UNRECOGNIZED_PRINT_CHARACTERS
AEFConst.PRINTER_ERROR, AEFConst.COVER_OPEN
AEFConst.PRINTER_ERROR, AEFConst.NONE
AEFConst.PRINTER_ERROR, AEFConst.PAPER_LOW
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.HOST_REQUEST_PENDING
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.NONE
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.TRANSACTION_NOT_ACTIVE
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.VALUE_CARDS_MUST_BE_VOIDED
AEFConst.UNCHECKED_EXCEPTION, AEFConst.NONE
Common Errors

retrieveSuspendedTransactionList

```
public SuspendedTransactionInfo[] retrieveSuspendedTransactionList()
                                                                 throws
java.rmi.RemoteException,
                                                                        AEFException
```

Retrieves a list of suspended transaction info objects for all terminals returned, sorted by terminal number. Action ID = RetrieveSuspendedTransactionList

Returns:

SuspendedTransactionInfo[] An array of SuspendedTransactionInfo objects.

Exceptions:

`RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.CONFIG_ERROR, AEFConst.FACTORY_ERROR
AEFConst.UNCHECKED_EXCEPTION, AEFConst.NONE

retrieveSuspendedTransactionList

```
public SuspendedTransactionInfo[] retrieveSuspendedTransactionList(int termNumber)
                                                                 throws
java.rmi.RemoteException,
                                                                        AEFException
```

Retrieves a list of suspended transaction info objects for a given terminal

Parameters:

`termNumber -`
Terminal number

Returns:

SuspendedTransactionInfo[] An array of SuspendedTransactionInfo objects.

Exceptions:

`RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.CONFIG_ERROR, AEFConst.FACTORY_ERROR
AEFConst.UNCHECKED_EXCEPTION, AEFConst.NONE

(continued from last page)

operatorOverride

```
public void operatorOverride(java.lang.String password)
                      throws java.rmi.RemoteException,
                             AEFException
```

Send an operator override to the POSAutomationProvider. Operator overrides apply to the current operator. Action ID = OperatorOverrideAction

Parameters:

`password -`
The operator override password. Use null or an empty string if the password is not required by the POS application.

Exceptions:

`RemoteException`
`AEFException -`
AEFException error codes are:
AEFConst.APPLICATION_NOT_IN_PROPER_STATE,
AEFConst.APPLICATION_NOT_IN_PROPER_SUBSTATE
AEFConst.CONFIG_ERROR, AEFConst.FACTORY_ERROR
AEFConst.POS_APP_FAILURE, AEFConst.UNRECOGNIZED_PRINT_CHARACTERS
AEFConst.PRINTER_ERROR, AEFConst.COVER_OPEN
AEFConst.PRINTER_ERROR, AEFConst.NONE
AEFConst.PRINTER_ERROR, AEFConst.PAPER_LOW
AEFConst.UNCHECKED_EXCEPTION, AEFConst.NONE
Common Errors

managerOverride

```
public void managerOverride(java.lang.String password)
                     throws java.rmi.RemoteException,
                            AEFException
```

Send a manager override to the POS Application. Action ID = ManagerOverrideAction

Parameters:

`password -`
The manager override number.

Exceptions:

`RemoteException`
`AEFException -`
AEFException error codes are:
AEFConst.APPLICATION_NOT_IN_PROPER_STATE,
AEFConst.APPLICATION_NOT_IN_PROPER_SUBSTATE
AEFConst.CONFIG_ERROR, AEFConst.FACTORY_ERROR
AEFConst.DEVICE_HOOK_ERROR, AEFConst.NONE
AEFConst.INVALID_MANAGER_OVERRIDE_NUMBER, AEFConst.NONE
AEFConst.JAVA_POS_EXCEPTION, AEFConst.NONE
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_NOT_REMOVED
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_REQUIRED
AEFConst.POS_APP_FAILURE, AEFConst.UNRECOGNIZED_PRINT_CHARACTERS
AEFConst.PRINTER_ERROR, AEFConst.COVER_OPEN
AEFConst.PRINTER_ERROR, AEFConst.NONE
AEFConst.PRINTER_ERROR, AEFConst.PAPER_LOW
AEFConst.UNCHECKED_EXCEPTION, AEFConst.NONE
Common Errors

managerOverride

```
public void managerOverride()
                     throws java.rmi.RemoteException,
                            AEFException
```

Send a manager override to the POS Application using the default manager number specified by the default manager override number property POSAutomationProvider.MGR_OVERRIDE_NUMBER. Action ID = ManagerOverrideAction

(continued from last page)

Exceptions:

`RemoteException`

`AEFException -`
AEFException error codes are:
AEFConst.APPLICATION_NOT_IN_PROPER_STATE,
AEFConst.APPLICATION_NOT_IN_PROPER_SUBSTATE
AEFConst.CONFIG_ERROR, AEFConst.FACTORY_ERROR
AEFConst.CONFIG_ERROR, AEFConst.NO_DEFAULT_MANAGER_OVERRIDE_PASSWORD
AEFConst.DEVICE_HOOK_ERROR, AEFConst.NONE
AEFConst.INVALID_MANAGER_OVERRIDE_NUMBER, AEFConst.NONE
AEFConst.JAVA_POS_EXCEPTION, AEFConst.NONE
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_NOT_REMOVED
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_REQUIRED
AEFConst.POS_APP_FAILURE, AEFConst.UNRECOGNIZED_PRINT_CHARACTERS
AEFConst.PRINTER_ERROR, AEFConst.COVER_OPEN
AEFConst.PRINTER_ERROR, AEFConst.NONE
AEFConst.PRINTER_ERROR, AEFConst.PAPER_LOW
AEFConst.UNCHECKED_EXCEPTION, AEFConst.NONE
Common Errors

cancelOverride

```
public void cancelOverride()
                    throws java.rmi.RemoteException,
                           AEFException
```

Cancel an override in the POS Application.

Exceptions:

`RemoteException`

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.CONFIG_ERROR, AEFConst.FACTORY_ERROR
AEFConst.UNCHECKED_EXCEPTION, AEFConst.NONE
Common Errors

voidCurrentTransaction

```
public void voidCurrentTransaction()
                             throws java.rmi.RemoteException,
                                    AEFException
```

Voids the transaction currently in progress. Action ID = VoidTransaction

Exceptions:

`RemoteException`

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.CONFIG_ERROR, AEFConst.FACTORY_ERROR
AEFConst.INVALID_MANAGER_OVERRIDE_NUMBER, AEFConst.NONE
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_NOT_REMOVED
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_REQUIRED
AEFConst.POS_APP_FAILURE, AEFConst.FILE_IO_ERROR
AEFConst.POS_APP_FAILURE, AEFConst.UNRECOGNIZED_PRINT_CHARACTERS
AEFConst.PRINTER_ERROR, AEFConst.COVER_OPEN
AEFConst.PRINTER_ERROR, AEFConst.NONE
AEFConst.PRINTER_ERROR, AEFConst.PAPER_LOW
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.NONE
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.TRANSACTION_NOT_ACTIVE
AEFConst.UNCHECKED_EXCEPTION, AEFConst.NONE
Common Errors

(continued from last page)

performAction

```
public java.lang.Object performAction(ActionRequest request)
                                throws java.rmi.RemoteException,
                                       AEFException
```

Perform an action specified by an ActionRequest. This API is meant to be used in situation where the more specific APIs are not sufficient for the necessary function. In this case, the developer writes an action class and requests the action to be performed.

Parameters:

`ActionRequest -`
An object which contains the class key and arguments for the action to be executed.

Returns:

Object An object returned from the action.

Exceptions:

`RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.INVALID_ACTION_REQUEST

getSession

```
public AEFSession getSession()
                        throws java.rmi.RemoteException
```

Gets the AEFSession for this instance.

Returns:

AEFSession

Exceptions:

`RemoteException`

getErrorHandlingMode

```
public int getErrorHandlingMode()
                         throws java.rmi.RemoteException,
                                AEFException
```

Get the Error Handling Mode for this application. The options are: HANDLE_AUTOMATIC which means that the error helper will send all the key strokes needed to get aroun the errorHANDLE_DEFAULT which means that the error helper will monitor the system and wait for the application to get to the correct state/substate. This would more than likely be the result of the operator performing some action on the system.HANDLE_CALLBACK which means that the error helper will make a call back to the calling application and wait for data or instructions.

Returns:

int POSAutomationProvider.HANDLE_AUTOMATIC, POSAutomationProvider.HANDLE_DEFAULT, or POSAutomationProvider.HANDLE_CALLBACK.

Exceptions:

`RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.PROPERTY_NOT_SUPPORTED
Common Errors

(continued from last page)

registerCallback

```
public void registerCallback(AEFErrorCallback callback)
                        throws java.rmi.RemoteException
```

Set the AEFErrorCallback object for the automation provider. The AEFErrorCallback object will be used when the error handling mode is set to HANDLE_CALLBACK. In this mode, the callback will be used when additional data is needed to complete the POS operation. For example, when a SalesTransaction.addItem call requires the operator to enter a date of birth.

Parameters:

`callback -`
An object which implements the com.ibm.retail.AEF.util.AEFErrorCallback interface.

Exceptions:

`RemoteException`

---

getCallback

```
public AEFErrorCallback getCallback()
                              throws java.rmi.RemoteException
```

Get the Callback object for the automation provider.

Returns:

AEFErrorCallback An object which implements the com.ibm.retail.AEF.util.AEFErrorCallback interface.

Exceptions:

`RemoteException`

---

getMaxErrorToHandle

```
public int getMaxErrorToHandle()
                         throws java.rmi.RemoteException,
                                AEFException
```

Gets the maximum number of errors that the error handler will try to clear.

Returns:

int The maximum number of errors that the error handler should try to clear.

Exceptions:

`RemoteException`

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.PROPERTY_NOT_SUPPORTED

---

setTrainingMode

```
public void setTrainingMode(boolean trainingMode)
                      throws java.rmi.RemoteException,
                             AEFException
```

Sets the training mode based on the trainingMode parameter. Action ID = TrainingModeOn or TrainingModeOff

Parameters:

`trainingMode -`
If true the terminal is put into training mode. If false the terminal will be taken out of training mode.

Exceptions:

`RemoteException`

(continued from last page)

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.CONFIG_ERROR, AEFConst.FACTORY_ERROR
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_NOT_REMOVED
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_REQUIRED
AEFConst.POS_APP_FAILURE, AEFConst.FILE_IO_ERROR
AEFConst.POS_APP_FAILURE, AEFConst.UNRECOGNIZED_PRINT_CHARACTERS
AEFConst.PRINTER_ERROR, AEFConst.COVER_OPEN
AEFConst.PRINTER_ERROR, AEFConst.NONE
AEFConst.PRINTER_ERROR, AEFConst.PAPER_LOW
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.NOT_ALLOWED_TRAINING
AEFConst.UNCHECKED_EXCEPTION, AEFConst.NONE
Common Errors

getTrainingMode

```
public boolean getTrainingMode()
                        throws java.rmi.RemoteException
```

Determines if the terminal is in training mode.

Returns:

boolean True if the terminal is in training mode. False if the terminal is not in training mode.

Exceptions:

`RemoteException`

com.ibm.retail.AEF.automation

Class PropertyContainsAtIndexCondition

```
java.lang.Object
  |-com.ibm.retail.AEF.automation.AbstractCondition
    |-com.ibm.retail.AEF.automation.AbstractPropertyCondition
      |-com.ibm.retail.AEF.automation.PropertyContainsAtIndexCondition
```

public class **PropertyContainsAtIndexCondition**
extends AbstractPropertyCondition

PropertyContainsAtIndexCondition listens to a POSDataProvider property and indicates whether the property contains the specified value at the specified index. The property being monitored is first converted to a String if it is not already a string data type. If the property is detected to contain the value, the lock associated with this condition is notified, which will free a thread blocking on the lock.

Field Summary

int	offset

Fields inherited from : class com.ibm.retail.AEF.automation.AbstractPropertyCondition
category, name, nextOnly, value

Fields inherited from : class com.ibm.retail.AEF.automation.AbstractCondition
counterLock, dataProvider, eligible, id, idCounter, index, listeners, lock, manager, session

Constructor Summary

PropertyContainsAtIndexCondition(java.lang.String cat,java.lang.String cat,java.lang.String cat,int cat,boolean cat) Constructs the condition
PropertyContainsAtIndexCondition(java.lang.String cat,java.lang.String cat,java.lang.String cat,int cat) Constructs the condition
PropertyContainsAtIndexCondition(PropertyNotContainsAtIndexCondition condition) Constructs the condition which is the opposite of a PropertyNotContainsAtIndexCondtion.

Method Summary

boolean	evaluate(java.lang.Object newVal) Evaluates the condition given a value object.
java.lang.String	explain() Returns a string explaining this condition.

int	`getTargetOffset()` Returns the target value offset which the target value must ocurr within the property.
java.lang.String	`getTargetValue()` Returns the target value which this condition is looking for within the property.
Condition	`makeOpposite()` Makes a new condition which is the logical opposite of this condition.
void	`propertyChanged(AEFPropertyChangeEvent evt)` A AEF POSDataProvider property was updated

Methods inherited from : class `com.ibm.retail.AEF.automation.AbstractPropertyCondition`
`currentValues, evaluate, evaluate, getCategory, getNextOnly, getPropertyName, parseDouble, setEligible`

Methods inherited from : class `com.ibm.retail.AEF.automation.AbstractCondition`
`addEvaluateListener, getID, getIndex, getNextID, getPOSDataProvider, getSession, getTerminalNumber, isEligible, notifyEvaluateConditionListeners, removeEvaluateListener, setConditionLock, setEligible, setID, setIndex, setLockManager, setPOSDataProvider, setSession`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

offset

```
protected int offset
```

Constructors

PropertyContainsAtIndexCondition

```
public PropertyContainsAtIndexCondition(java.lang.String cat,
                                        java.lang.String name,
                                        java.lang.String value,
                                        int index,
                                        boolean nextValueOnly)
```

Constructs the condition

Parameters:

`cat` -
The category name of the property to be monitored.
`name` -
The property name to be monitored.
`value` -
The property value is compared to see if it contains this string.

(continued from last page)

`index -`
The zero based index within the property where the value must be observed for the condition to be true.
`nextValueOnly -`
Set to true to compare only the next value of the property. If false, all values of the property change will be compared.

PropertyContainsAtIndexCondition

```
public PropertyContainsAtIndexCondition(java.lang.String cat,
                                        java.lang.String name,
                                        java.lang.String value,
                                        int index)
```

Constructs the condition

Parameters:

`cat -`
The category name of the property to be monitored.
`name -`
The property name to be monitored.
`value -`
The property value is compared to see if it contains this string.
`index -`
The zero based index within the property where the value must be observed for the condition to be true.

PropertyContainsAtIndexCondition

```
public PropertyContainsAtIndexCondition(PropertyNotContainsAtIndexCondition condition)
```

Constructs the condition which is the opposite of a PropertyNotContainsAtIndexCondtion.

Parameters:

`condition -`
The PropertyNotContainsAtIndexCondition to construct the opposite of.

Methods

makeOpposite

```
public Condition makeOpposite()
```

Makes a new condition which is the logical opposite of this condition.

Returns:

Condition

propertyChanged

```
public void propertyChanged(AEFPropertyChangeEvent evt)
                     throws java.rmi.RemoteException
```

A AEF POSDataProvider property was updated

Parameters:

`evt -`
contains details of the event

evaluate

```
public boolean evaluate(java.lang.Object newVal)
```

Evaluates the condition given a value object.

Parameters:

(continued from last page)

`newVal` -
The Object value.

Returns:

boolean True if the condition evaluates to true.

getTargetValue

`public java.lang.String` **`getTargetValue`**`()`

Returns the target value which this condition is looking for within the property.

Returns:

String

getTargetOffset

`public int` **`getTargetOffset`**`()`

Returns the target value offset which the target value must ocurr within the property.

Returns:

int

explain

`public java.lang.String` **`explain`**`()`

Returns a string explaining this condition.

Returns:

String

com.ibm.retail.AEF.automation

Class PropertyContainsCondition

```
java.lang.Object
  |-com.ibm.retail.AEF.automation.AbstractCondition
    |-com.ibm.retail.AEF.automation.AbstractPropertyCondition
      |-com.ibm.retail.AEF.automation.PropertyContainsCondition
```

public class **PropertyContainsCondition**
extends AbstractPropertyCondition

PropertyContainsCondition listens to a POSDataProvider property and indicates whether the property contains the specified value. The property being monitored is first converted to a String if it is not already a string data type. If the property is detected to contain the value, the lock associated with this condition is notified, which will free a thread blocking on the lock.

Fields inherited from : class `com.ibm.retail.AEF.automation.AbstractPropertyCondition`
`category`, `name`, `nextOnly`, `value`

Fields inherited from : class `com.ibm.retail.AEF.automation.AbstractCondition`
`counterLock`, `dataProvider`, `eligible`, `id`, `idCounter`, `index`, `listeners`, `lock`, `manager`, `session`

Constructor Summary

`PropertyContainsCondition(java.lang.String cat,java.lang.String cat,java.lang.String cat,boolean cat)` Constructs the condition
`PropertyContainsCondition(java.lang.String cat,java.lang.String cat,java.lang.String cat)` Constructs the condition
`PropertyContainsCondition(PropertyNotContainsCondition condition)` Constructs a new condition which is the logical opposite of the give condition.

Method Summary

`boolean`	`evaluate(java.lang.Object newVal)` Evaluates the condition given a value object.
`java.lang.String`	`explain()` Returns a string explaining this condition.
`java.lang.String`	`getTargetValue()` Returns the target value which this condition is looking for within the property.
`Condition`	`makeOpposite()` Makes a new condition which is the logical opposite of this condition.

void	`propertyChanged(AEFPropertyChangeEvent evt)` A AEF POSDataProvider property was updated

Methods inherited from : class `com.ibm.retail.AEF.automation.AbstractPropertyCondition`
`currentValues, evaluate, evaluate, getCategory, getNextOnly, getPropertyName, parseDouble, setEligible`

Methods inherited from : class `com.ibm.retail.AEF.automation.AbstractCondition`
`addEvaluateListener, getID, getIndex, getNextID, getPOSDataProvider, getSession, getTerminalNumber, isEligible, notifyEvaluateConditionListeners, removeEvaluateListener, setConditionLock, setEligible, setID, setIndex, setLockManager, setPOSDataProvider, setSession`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

PropertyContainsCondition

```
public PropertyContainsCondition(java.lang.String cat,
                                 java.lang.String name,
                                 java.lang.String value,
                                 boolean nextValueOnly)
```

Constructs the condition

PropertyContainsCondition

```
public PropertyContainsCondition(java.lang.String cat,
                                 java.lang.String name,
                                 java.lang.String value)
```

Constructs the condition

PropertyContainsCondition

```
public PropertyContainsCondition(PropertyNotContainsCondition condition)
```

Constructs a new condition which is the logical opposite of the give condition.

Methods

makeOpposite

```
public Condition makeOpposite()
```

Makes a new condition which is the logical opposite of this condition.

Returns:

Condition

propertyChanged

```
public void propertyChanged(AEFPropertyChangeEvent evt)
                      throws java.rmi.RemoteException
```

A AEF POSDataProvider property was updated

Parameters:

`evt -`
contains details of the event

evaluate

```
public boolean evaluate(java.lang.Object newVal)
```

Evaluates the condition given a value object.

Parameters:

`newVal -`
The Object value.

Returns:

boolean True if the condition evaluates to true.

getTargetValue

```
public java.lang.String getTargetValue()
```

Returns the target value which this condition is looking for within the property.

Returns:

String

explain

```
public java.lang.String explain()
```

Returns a string explaining this condition.

Returns:

String

com.ibm.retail.AEF.automation

Class PropertyEqualsCondition

```
java.lang.Object
  |-com.ibm.retail.AEF.automation.AbstractCondition
    |-com.ibm.retail.AEF.automation.AbstractPropertyCondition
      |-com.ibm.retail.AEF.automation.PropertyEqualsCondition
```

public class **PropertyEqualsCondition**
extends AbstractPropertyCondition

PropertyEqualsCondition listens to a POSDataProvider property and indicates whether the property is equal to a specified value. If the property value is equal, then lock associated with this condition is notified.

Field Summary

`java.lang.Double`	`doubleValue`

Fields inherited from : class `com.ibm.retail.AEF.automation.AbstractPropertyCondition`
`category`, `name`, `nextOnly`, `value`

Fields inherited from : class `com.ibm.retail.AEF.automation.AbstractCondition`
`counterLock`, `dataProvider`, `eligible`, `id`, `idCounter`, `index`, `listeners`, `lock`, `manager`, `session`

Constructor Summary

`PropertyEqualsCondition(java.lang.String cat,java.lang.String cat,java.lang.String cat,boolean cat)` Constructs the condition
`PropertyEqualsCondition(java.lang.String cat,java.lang.String cat,java.lang.String cat)` Constructs the condition
`PropertyEqualsCondition(java.lang.String cat,java.lang.String cat,double cat,boolean cat)` Constructs the condition
`PropertyEqualsCondition(java.lang.String cat,java.lang.String cat,double cat)` Constructs the condition
`PropertyEqualsCondition(PropertyNotEqualsCondition condition)` Constructs the condition which is the logical opposite of the given condition.

Method Summary

`boolean`	`evaluate(java.lang.Object newVal)` Evaluates the condition given a value object.

java.lang.String	explain() Returns a string explaining this condition.
java.lang.Double	getTargetDoubleValue() Returns the target value which this condition is looking for within the property.
java.lang.String	getTargetValue() Returns the target value which this condition is looking for within the property.
Condition	makeOpposite() Makes a new condition which is the logical opposite of this condition.
void	propertyChanged(AEFPropertyChangeEvent evt) A AEF POSDataProvider property was updated

Methods inherited from : class com.ibm.retail.AEF.automation.AbstractPropertyCondition
currentValues, evaluate, evaluate, getCategory, getNextOnly, getPropertyName, parseDouble, setEligible

Methods inherited from : class com.ibm.retail.AEF.automation.AbstractCondition
addEvaluateListener, getID, getIndex, getNextID, getPOSDataProvider, getSession, getTerminalNumber, isEligible, notifyEvaluateConditionListeners, removeEvaluateListener, setConditionLock, setEligible, setID, setIndex, setLockManager, setPOSDataProvider, setSession

Methods inherited from : class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

doubleValue

protected java.lang.Double **doubleValue**

Constructors

PropertyEqualsCondition

```
public PropertyEqualsCondition(java.lang.String cat,
                               java.lang.String name,
                               java.lang.String value,
                               boolean nextValueOnly)
```

Constructs the condition

Parameters:

cat -
The POSDataProvider property category

(continued from last page)

`name` -
The POSDataProvider property name
`value` -
The string value to test the property for equality against
`nextValueOnly` -
Set true to evaluate this condition for the next property value only.

PropertyEqualsCondition

```
public PropertyEqualsCondition(java.lang.String cat,
                               java.lang.String name,
                               java.lang.String value)
```

Constructs the condition

Parameters:

`cat` -
The POSDataProvider property category
`name` -
The POSDataProvider property name
`value` -
The string value to test the property for equality against

PropertyEqualsCondition

```
public PropertyEqualsCondition(java.lang.String cat,
                               java.lang.String name,
                               double value,
                               boolean nextValueOnly)
```

Constructs the condition

Parameters:

`cat` -
The POSDataProvider property category
`name` -
The POSDataProvider property name
`value` -
A numeric value to test the property for equality against
`nextValueOnly` -
Set true to evaluate this condition for the next property value only.

PropertyEqualsCondition

```
public PropertyEqualsCondition(java.lang.String cat,
                               java.lang.String name,
                               double value)
```

Constructs the condition

Parameters:

`cat` -
The POSDataProvider property category
`name` -
The POSDataProvider property name
`value` -
The numeric value to test the property for equality against

PropertyEqualsCondition

```
public PropertyEqualsCondition(PropertyNotEqualsCondition condition)
```

Constructs the condition which is the logical opposite of the given condition.

Parameters:

`condition`

Methods

makeOpposite

```
public Condition makeOpposite()
```

Makes a new condition which is the logical opposite of this condition.

Returns:

Condition

propertyChanged

```
public void propertyChanged(AEFPropertyChangeEvent evt)
                    throws java.rmi.RemoteException
```

A AEF POSDataProvider property was updated

Parameters:

`evt -`
contains details of the event

evaluate

```
public boolean evaluate(java.lang.Object newVal)
```

Evaluates the condition given a value object.

Parameters:

`newVal -`
The Object value.

Returns:

boolean True if the condition evaluates to true.

explain

```
public java.lang.String explain()
```

Returns a string explaining this condition.

Returns:

String

getTargetValue

```
public java.lang.String getTargetValue()
```

Returns the target value which this condition is looking for within the property.

Returns:

String

getTargetDoubleValue

```
public java.lang.Double getTargetDoubleValue()
```

Returns the target value which this condition is looking for within the property.

(continued from last page)

Returns:

Double

com.ibm.retail.AEF.automation

Class PropertyGreaterOrEqualCondition

```
java.lang.Object
  +-com.ibm.retail.AEF.automation.AbstractCondition
    +-com.ibm.retail.AEF.automation.AbstractPropertyCondition
      +-com.ibm.retail.AEF.automation.PropertyGreaterOrEqualCondition
```

public class **PropertyGreaterOrEqualCondition**
extends AbstractPropertyCondition

PropertyGreaterOrEqualCondition listens to a POSDataProvider property and indicates whether the numeric property value is greater or equal to a specified value. If so, then lock associated with this condition is notified.

Field Summary

`java.lang.Double`	`doubleValue`

Fields inherited from : class `com.ibm.retail.AEF.automation.AbstractPropertyCondition`
`category`, `name`, `nextOnly`, `value`

Fields inherited from : class `com.ibm.retail.AEF.automation.AbstractCondition`
`counterLock`, `dataProvider`, `eligible`, `id`, `idCounter`, `index`, `listeners`, `lock`, `manager`, `session`

Constructor Summary

`PropertyGreaterOrEqualCondition(java.lang.String cat,java.lang.String cat,double cat,boolean cat)` Constructs the condition
`PropertyGreaterOrEqualCondition(java.lang.String cat,java.lang.String cat,double cat)` Constructs the condition
`PropertyGreaterOrEqualCondition(PropertyLessThanCondition condition)` Constructs the condition which is the logical opposite of the given condition.

Method Summary

`boolean`	`evaluate(java.lang.Object newVal)` Evaluates the condition given a value object.
`java.lang.String`	`explain()` Returns a string explaining this condition.
`java.lang.Double`	`getTargetDoubleValue()` Returns the target value which this condition is looking for within the property.

Condition	`makeOpposite()` Makes a new condition which is the logical opposite of this condition.
void	`propertyChanged(AEFPropertyChangeEvent evt)` A AEF POSDataProvider property was updated

Methods inherited from : class `com.ibm.retail.AEF.automation.AbstractPropertyCondition`
`currentValues, evaluate, evaluate, getCategory, getNextOnly, getPropertyName, parseDouble, setEligible`

Methods inherited from : class `com.ibm.retail.AEF.automation.AbstractCondition`
`addEvaluateListener, getID, getIndex, getNextID, getPOSDataProvider, getSession, getTerminalNumber, isEligible, notifyEvaluateConditionListeners, removeEvaluateListener, setConditionLock, setEligible, setID, setIndex, setLockManager, setPOSDataProvider, setSession`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

doubleValue

```
protected java.lang.Double doubleValue
```

Constructors

PropertyGreaterOrEqualCondition

```
public PropertyGreaterOrEqualCondition(java.lang.String cat,
                                       java.lang.String name,
                                       double value,
                                       boolean nextValueOnly)
```

Constructs the condition

Parameters:

`cat` -
The POSDataProvider property category
`name` -
The POSDataProvider property name
`value` -
A numeric value to test the property against
`nextValueOnly` -
Set true to evaluate this condition for the next property value only.

(continued from last page)

PropertyGreaterOrEqualCondition

```
public PropertyGreaterOrEqualCondition(java.lang.String cat,
                                       java.lang.String name,
                                       double value)
```

Constructs the condition

Parameters:

`cat -`
The POSDataProvider property category
`name -`
The POSDataProvider property name
`value -`
The numeric value to test the property against

PropertyGreaterOrEqualCondition

```
public PropertyGreaterOrEqualCondition(PropertyLessThanCondition condition)
```

Constructs the condition which is the logical opposite of the given condition.

Parameters:

`condition`

Methods

makeOpposite

```
public Condition makeOpposite()
```

Makes a new condition which is the logical opposite of this condition.

Returns:

Condition

propertyChanged

```
public void propertyChanged(AEFPropertyChangeEvent evt)
                     throws java.rmi.RemoteException
```

A AEF POSDataProvider property was updated

Parameters:

`evt -`
contains details of the event

evaluate

```
public boolean evaluate(java.lang.Object newVal)
```

Evaluates the condition given a value object.

Parameters:

`newVal -`
The Object value.

Returns:

boolean True if the condition evaluates to true.

(continued from last page)

explain

`public java.lang.String` **`explain`**`()`

Returns a string explaining this condition.

Returns:

String

getTargetDoubleValue

`public java.lang.Double` **`getTargetDoubleValue`**`()`

Returns the target value which this condition is looking for within the property.

Returns:

Double

com.ibm.retail.AEF.automation

Class PropertyGreaterThanCondition

```
java.lang.Object
  |-com.ibm.retail.AEF.automation.AbstractCondition
    |-com.ibm.retail.AEF.automation.AbstractPropertyCondition
      |-com.ibm.retail.AEF.automation.PropertyGreaterThanCondition
```

public class **PropertyGreaterThanCondition**
extends AbstractPropertyCondition

PropertyGreaterThanCondition listens to a POSDataProvider property and indicates whether the numeric property value is greater than a specified value. If so, then lock associated with this condition is notified.

Field Summary

`java.lang.Double`	`doubleValue`

Fields inherited from : class `com.ibm.retail.AEF.automation.AbstractPropertyCondition`
`category`, `name`, `nextOnly`, `value`

Fields inherited from : class `com.ibm.retail.AEF.automation.AbstractCondition`
`counterLock`, `dataProvider`, `eligible`, `id`, `idCounter`, `index`, `listeners`, `lock`, `manager`, `session`

Constructor Summary

`PropertyGreaterThanCondition(java.lang.String cat,java.lang.String cat,double cat,boolean cat)` Constructs the condition
`PropertyGreaterThanCondition(java.lang.String cat,java.lang.String cat,double cat)` Constructs the condition
`PropertyGreaterThanCondition(PropertyLessOrEqualCondition condition)` Constructs the condition which is the logical opposite of the given condition.

Method Summary

`boolean`	`evaluate(java.lang.Object newVal)` Evaluates the condition given a value object.
`java.lang.String`	`explain()` Returns a string explaining this condition.
`java.lang.Double`	`getTargetDoubleValue()` Returns the target value which this condition is looking for within the property.

Condition	`makeOpposite()` Makes a new condition which is the logical opposite of this condition.
void	`propertyChanged(AEFPropertyChangeEvent evt)` A AEF POSDataProvider property was updated

Methods inherited from : class `com.ibm.retail.AEF.automation.AbstractPropertyCondition`
`currentValues, evaluate, evaluate, getCategory, getNextOnly, getPropertyName, parseDouble, setEligible`

Methods inherited from : class `com.ibm.retail.AEF.automation.AbstractCondition`
`addEvaluateListener, getID, getIndex, getNextID, getPOSDataProvider, getSession, getTerminalNumber, isEligible, notifyEvaluateConditionListeners, removeEvaluateListener, setConditionLock, setEligible, setID, setIndex, setLockManager, setPOSDataProvider, setSession`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

doubleValue

```
protected java.lang.Double doubleValue
```

Constructors

PropertyGreaterThanCondition

```
public PropertyGreaterThanCondition(java.lang.String cat,
                                    java.lang.String name,
                                    double value,
                                    boolean nextValueOnly)
```

Constructs the condition

Parameters:

`cat` -
The POSDataProvider property category
`name` -
The POSDataProvider property name
`value` -
A numeric value to test the property against
`nextValueOnly` -
Set true to evaluate this condition for the next property value only.

(continued from last page)

PropertyGreaterThanCondition

```
public PropertyGreaterThanCondition(java.lang.String cat,
                                    java.lang.String name,
                                    double value)
```

Constructs the condition

Parameters:

`cat -`
The POSDataProvider property category
`name -`
The POSDataProvider property name
`value -`
The numeric value to test the property against

PropertyGreaterThanCondition

```
public PropertyGreaterThanCondition(PropertyLessOrEqualCondition condition)
```

Constructs the condition which is the logical opposite of the given condition.

Parameters:

`condition`

Methods

makeOpposite

```
public Condition makeOpposite()
```

Makes a new condition which is the logical opposite of this condition.

Returns:

Condition

propertyChanged

```
public void propertyChanged(AEFPropertyChangeEvent evt)
                     throws java.rmi.RemoteException
```

A AEF POSDataProvider property was updated

Parameters:

`evt -`
contains details of the event

evaluate

```
public boolean evaluate(java.lang.Object newVal)
```

Evaluates the condition given a value object.

Parameters:

`newVal -`
The Object value.

Returns:

boolean True if the condition evaluates to true.

(continued from last page)

explain

`public java.lang.String` **`explain`**`()`

Returns a string explaining this condition.

Returns:

String

getTargetDoubleValue

`public java.lang.Double` **`getTargetDoubleValue`**`()`

Returns the target value which this condition is looking for within the property.

Returns:

Double

com.ibm.retail.AEF.automation

Class PropertyLessOrEqualCondition

```
java.lang.Object
  +-com.ibm.retail.AEF.automation.AbstractCondition
    +-com.ibm.retail.AEF.automation.AbstractPropertyCondition
      +-com.ibm.retail.AEF.automation.PropertyLessOrEqualCondition
```

public class **PropertyLessOrEqualCondition**
extends AbstractPropertyCondition

PropertyLessOrEqualCondition listens to a POSDataProvider property and indicates whether the numeric property value is less than or equal to a specified value. If so, then lock associated with this condition is notified.

Field Summary

`java.lang.Double`	`doubleValue`

Fields inherited from : class `com.ibm.retail.AEF.automation.AbstractPropertyCondition`
`category`, `name`, `nextOnly`, `value`

Fields inherited from : class `com.ibm.retail.AEF.automation.AbstractCondition`
`counterLock`, `dataProvider`, `eligible`, `id`, `idCounter`, `index`, `listeners`, `lock`, `manager`, `session`

Constructor Summary

`PropertyLessOrEqualCondition(java.lang.String cat,java.lang.String cat,double cat,boolean cat)` Constructs the condition
`PropertyLessOrEqualCondition(java.lang.String cat,java.lang.String cat,double cat)` Constructs the condition
`PropertyLessOrEqualCondition(PropertyGreaterThanCondition condition)` Constructs the condition which is the logical opposite of the given condition.

Method Summary

`boolean`	`evaluate(java.lang.Object newVal)` Evaluates the condition given a value object.
`java.lang.String`	`explain()` Returns a string explaining this condition.
`java.lang.Double`	`getTargetDoubleValue()` Returns the target value which this condition is looking for within the property.

Condition	`makeOpposite()` Makes a new condition which is the logical opposite of this condition.
void	`propertyChanged(AEFPropertyChangeEvent evt)` A AEF POSDataProvider property was updated

Methods inherited from : class `com.ibm.retail.AEF.automation.AbstractPropertyCondition`
`currentValues, evaluate, evaluate, getCategory, getNextOnly, getPropertyName, parseDouble, setEligible`

Methods inherited from : class `com.ibm.retail.AEF.automation.AbstractCondition`
`addEvaluateListener, getID, getIndex, getNextID, getPOSDataProvider, getSession, getTerminalNumber, isEligible, notifyEvaluateConditionListeners, removeEvaluateListener, setConditionLock, setEligible, setID, setIndex, setLockManager, setPOSDataProvider, setSession`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

doubleValue

```
protected java.lang.Double doubleValue
```

Constructors

PropertyLessOrEqualCondition

```
public PropertyLessOrEqualCondition(java.lang.String cat,
                                    java.lang.String name,
                                    double value,
                                    boolean nextValueOnly)
```

Constructs the condition

Parameters:

`cat` -
The POSDataProvider property category
`name` -
The POSDataProvider property name
`value` -
A numeric value to test the property against
`nextValueOnly` -
Set true to evaluate this condition for the next property value only.

(continued from last page)

PropertyLessOrEqualCondition

```
public PropertyLessOrEqualCondition(java.lang.String cat,
                                    java.lang.String name,
                                    double value)
```

Constructs the condition

Parameters:

`cat -`
The POSDataProvider property category
`name -`
The POSDataProvider property name
`value -`
The numeric value to test the property against

PropertyLessOrEqualCondition

```
public PropertyLessOrEqualCondition(PropertyGreaterThanCondition condition)
```

Constructs the condition which is the logical opposite of the given condition.

Parameters:

`condition`

Methods

makeOpposite

```
public Condition makeOpposite()
```

Makes a new condition which is the logical opposite of this condition.

Returns:

Condition

propertyChanged

```
public void propertyChanged(AEFPropertyChangeEvent evt)
                      throws java.rmi.RemoteException
```

A AEF POSDataProvider property was updated

Parameters:

`evt -`
contains details of the event

evaluate

```
public boolean evaluate(java.lang.Object newVal)
```

Evaluates the condition given a value object.

Parameters:

`newVal -`
The Object value.

Returns:

boolean True if the condition evaluates to true.

(continued from last page)

explain

`public java.lang.String` **`explain`**`()`

Returns a string explaining this condition.

Returns:

String

getTargetDoubleValue

`public java.lang.Double` **`getTargetDoubleValue`**`()`

Returns the target value which this condition is looking for within the property.

Returns:

Double

com.ibm.retail.AEF.automation

Class PropertyLessThanCondition

```
java.lang.Object
  |-com.ibm.retail.AEF.automation.AbstractCondition
    |-com.ibm.retail.AEF.automation.AbstractPropertyCondition
      |-com.ibm.retail.AEF.automation.PropertyLessThanCondition
```

public class **PropertyLessThanCondition**
extends AbstractPropertyCondition

PropertyLessThanCondition listens to a POSDataProvider property and indicates whether the numeric property value is less than a specified value. If so, then lock associated with this condition is notified.

Field Summary

`java.lang.Double`	`doubleValue`

Fields inherited from : class `com.ibm.retail.AEF.automation.AbstractPropertyCondition`
`category`, `name`, `nextOnly`, `value`

Fields inherited from : class `com.ibm.retail.AEF.automation.AbstractCondition`
`counterLock`, `dataProvider`, `eligible`, `id`, `idCounter`, `index`, `listeners`, `lock`, `manager`, `session`

Constructor Summary

`PropertyLessThanCondition(java.lang.String cat,java.lang.String cat,double cat,boolean cat)` Constructs the condition
`PropertyLessThanCondition(java.lang.String cat,java.lang.String cat,double cat)` Constructs the condition
`PropertyLessThanCondition(PropertyGreaterOrEqualCondition condition)` Constructs the condition which is the logical opposite of the given condition.

Method Summary

`boolean`	`evaluate(java.lang.Object newVal)` Evaluates the condition given a value object.
`java.lang.String`	`explain()` Returns a string explaining this condition.
`java.lang.Double`	`getTargetDoubleValue()` Returns the target value which this condition is looking for within the property.

Condition	`makeOpposite()` Makes a new condition which is the logical opposite of this condition.
void	`propertyChanged(AEFPropertyChangeEvent evt)` A AEF POSDataProvider property was updated

Methods inherited from : class `com.ibm.retail.AEF.automation.AbstractPropertyCondition`
`currentValues, evaluate, evaluate, getCategory, getNextOnly, getPropertyName, parseDouble, setEligible`

Methods inherited from : class `com.ibm.retail.AEF.automation.AbstractCondition`
`addEvaluateListener, getID, getIndex, getNextID, getPOSDataProvider, getSession, getTerminalNumber, isEligible, notifyEvaluateConditionListeners, removeEvaluateListener, setConditionLock, setEligible, setID, setIndex, setLockManager, setPOSDataProvider, setSession`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

doubleValue

```
protected java.lang.Double doubleValue
```

Constructors

PropertyLessThanCondition

```
public PropertyLessThanCondition(java.lang.String cat,
                                 java.lang.String name,
                                 double value,
                                 boolean nextValueOnly)
```

Constructs the condition

Parameters:

`cat` -
The POSDataProvider property category
`name` -
The POSDataProvider property name
`value` -
A numeric value to test the property against
`nextValueOnly` -
Set true to evaluate this condition for the next property value only.

(continued from last page)

PropertyLessThanCondition

```
public PropertyLessThanCondition(java.lang.String cat,
                                 java.lang.String name,
                                 double value)
```

Constructs the condition

Parameters:

`cat -`
The POSDataProvider property category
`name -`
The POSDataProvider property name
`value -`
The numeric value to test the property against

PropertyLessThanCondition

```
public PropertyLessThanCondition(PropertyGreaterOrEqualCondition condition)
```

Constructs the condition which is the logical opposite of the given condition.

Parameters:

`condition`

Methods

makeOpposite

```
public Condition makeOpposite()
```

Makes a new condition which is the logical opposite of this condition.

Returns:

Condition

propertyChanged

```
public void propertyChanged(AEFPropertyChangeEvent evt)
                     throws java.rmi.RemoteException
```

A AEF POSDataProvider property was updated

Parameters:

`evt -`
contains details of the event

evaluate

```
public boolean evaluate(java.lang.Object newVal)
```

Evaluates the condition given a value object.

Parameters:

`newVal -`
The Object value.

Returns:

boolean True if the condition evaluates to true.

(continued from last page)

explain

`public java.lang.String` **`explain`**`()`

Returns a string explaining this condition.

Returns:

String

getTargetDoubleValue

`public java.lang.Double` **`getTargetDoubleValue`**`()`

Returns the target value which this condition is looking for within the property.

Returns:

Double

com.ibm.retail.AEF.automation

Class PropertyNotContainsAtIndexCondition

```
java.lang.Object
  |-com.ibm.retail.AEF.automation.AbstractCondition
    |-com.ibm.retail.AEF.automation.AbstractPropertyCondition
      |-com.ibm.retail.AEF.automation.PropertyNotContainsAtIndexCondition
```

public class **PropertyNotContainsAtIndexCondition**
extends AbstractPropertyCondition

PropertyNotContainsAtIndexCondition listens to a POSDataProvider property and indicates whether the property does not contain the specified value at the specified index. The property being monitored is first converted to a String if it is not already a string data type. If the property is detected to not contain the value, the lock associated with this condition is notified, which will free a thread blocking on the lock.

Field Summary

int	offset

Fields inherited from : class com.ibm.retail.AEF.automation.AbstractPropertyCondition
category, name, nextOnly, value

Fields inherited from : class com.ibm.retail.AEF.automation.AbstractCondition
counterLock, dataProvider, eligible, id, idCounter, index, listeners, lock, manager, session

Constructor Summary

PropertyNotContainsAtIndexCondition(java.lang.String cat,java.lang.String cat,java.lang.String cat,int cat,boolean cat) Constructs the condition
PropertyNotContainsAtIndexCondition(java.lang.String cat,java.lang.String cat,java.lang.String cat,int cat) Constructs the condition
PropertyNotContainsAtIndexCondition(PropertyContainsAtIndexCondition condition) Constructs the condition which is the opposite of a PropertyContainsAtIndexCondtion.

Method Summary

boolean	evaluate(java.lang.Object newVal) Evaluates the condition given a value object.
java.lang.String	explain() Returns a string explaining this condition.

int	`getTargetOffset()` Returns the target value offset which the target value must ocurr within the property.
java.lang.String	`getTargetValue()` Returns the target value which this condition is looking for within the property.
Condition	`makeOpposite()` Makes a new condition which is the logical opposite of this condition.
void	`propertyChanged(AEFPropertyChangeEvent evt)` A AEF POSDataProvider property was updated

Methods inherited from : class `com.ibm.retail.AEF.automation.AbstractPropertyCondition`
`currentValues, evaluate, evaluate, getCategory, getNextOnly, getPropertyName, parseDouble, setEligible`

Methods inherited from : class `com.ibm.retail.AEF.automation.AbstractCondition`
`addEvaluateListener, getID, getIndex, getNextID, getPOSDataProvider, getSession, getTerminalNumber, isEligible, notifyEvaluateConditionListeners, removeEvaluateListener, setConditionLock, setEligible, setID, setIndex, setLockManager, setPOSDataProvider, setSession`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

offset

```
protected int offset
```

Constructors

PropertyNotContainsAtIndexCondition

```
public PropertyNotContainsAtIndexCondition(java.lang.String cat,
                                           java.lang.String name,
                                           java.lang.String value,
                                           int index,
                                           boolean nextValueOnly)
```

Constructs the condition

Parameters:

`cat` -
The category name of the property to be monitored.
`name` -
The property name to be monitored.
`value` -
The property value is compared to see if does not contain this string.

(continued from last page)

`index -`
The zero based index within the property where the value must not be observed for the condition to be true.
`nextValueOnly -`
Set to true to compare only the next value of the property. If false, all values of the property change will be compared.

PropertyNotContainsAtIndexCondition

```
public PropertyNotContainsAtIndexCondition(java.lang.String cat,
                                           java.lang.String name,
                                           java.lang.String value,
                                           int index)
```

Constructs the condition

Parameters:

`cat -`
The category name of the property to be monitored.
`name -`
The property name to be monitored.
`value -`
The property value is compared to see if it does not contain this string.
`index -`
The zero based index within the property where the value must not be observed for the condition to be true.

PropertyNotContainsAtIndexCondition

```
public PropertyNotContainsAtIndexCondition(PropertyContainsAtIndexCondition condition)
```

Constructs the condition which is the opposite of a PropertyContainsAtIndexCondtion.

Parameters:

`condition -`
The PropertyContainsAtIndexCondition to construct the opposite of.

Methods

makeOpposite

```
public Condition makeOpposite()
```

Makes a new condition which is the logical opposite of this condition.

Returns:

Condition

propertyChanged

```
public void propertyChanged(AEFPropertyChangeEvent evt)
                     throws java.rmi.RemoteException
```

A AEF POSDataProvider property was updated

Parameters:

`evt -`
contains details of the event

evaluate

```
public boolean evaluate(java.lang.Object newVal)
```

Evaluates the condition given a value object.

Parameters:

(continued from last page)

`newVal` -
The Object value.

Returns:

boolean True if the condition evaluates to true.

explain

`public java.lang.String` **`explain`**`()`

Returns a string explaining this condition.

Returns:

String

getTargetValue

`public java.lang.String` **`getTargetValue`**`()`

Returns the target value which this condition is looking for within the property.

Returns:

String

getTargetOffset

`public int` **`getTargetOffset`**`()`

Returns the target value offset which the target value must ocurr within the property.

Returns:

int

com.ibm.retail.AEF.automation

Class PropertyNotContainsCondition

```
java.lang.Object
  |-com.ibm.retail.AEF.automation.AbstractCondition
    |-com.ibm.retail.AEF.automation.AbstractPropertyCondition
      |-com.ibm.retail.AEF.automation.PropertyNotContainsCondition
```

public class **PropertyNotContainsCondition**
extends AbstractPropertyCondition

PropertyNotContainsCondition listens to a POSDataProvider property and indicates whether the property does not contain the specified value. The property being monitored is first converted to a String if it is not already a string data type. If the property is detected to not contain the value, the lock associated with this condition is notified, which will free a thread blocking on the lock.

Fields inherited from : class `com.ibm.retail.AEF.automation.AbstractPropertyCondition`
`category`, `name`, `nextOnly`, `value`

Fields inherited from : class `com.ibm.retail.AEF.automation.AbstractCondition`
`counterLock`, `dataProvider`, `eligible`, `id`, `idCounter`, `index`, `listeners`, `lock`, `manager`, `session`

Constructor Summary

`PropertyNotContainsCondition(java.lang.String cat,java.lang.String cat,java.lang.String cat,boolean cat)` Constructs the condition
`PropertyNotContainsCondition(java.lang.String cat,java.lang.String cat,java.lang.String cat)` Constructs the condition
`PropertyNotContainsCondition(PropertyContainsCondition condition)` Constructs a new condition which is the logical opposite of the give condition.

Method Summary

`boolean`	`evaluate(java.lang.Object newVal)` Evaluates the condition given a value object.
`java.lang.String`	`explain()` Returns a string explaining this condition.
`java.lang.String`	`getTargetValue()` Returns the target value which this condition is looking for within the property.
`Condition`	`makeOpposite()` Makes a new condition which is the logical opposite of this condition.

void	`propertyChanged(AEFPropertyChangeEvent evt)` A AEF POSDataProvider property was updated

Methods inherited from : class `com.ibm.retail.AEF.automation.AbstractPropertyCondition`
`currentValues, evaluate, evaluate, getCategory, getNextOnly, getPropertyName, parseDouble, setEligible`

Methods inherited from : class `com.ibm.retail.AEF.automation.AbstractCondition`
`addEvaluateListener, getID, getIndex, getNextID, getPOSDataProvider, getSession, getTerminalNumber, isEligible, notifyEvaluateConditionListeners, removeEvaluateListener, setConditionLock, setEligible, setID, setIndex, setLockManager, setPOSDataProvider, setSession`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

PropertyNotContainsCondition

```
public PropertyNotContainsCondition(java.lang.String cat,
                                    java.lang.String name,
                                    java.lang.String value,
                                    boolean nextValueOnly)
```

Constructs the condition

PropertyNotContainsCondition

```
public PropertyNotContainsCondition(java.lang.String cat,
                                    java.lang.String name,
                                    java.lang.String value)
```

Constructs the condition

PropertyNotContainsCondition

```
public PropertyNotContainsCondition(PropertyContainsCondition condition)
```

Constructs a new condition which is the logical opposite of the give condition.

Methods

makeOpposite

```
public Condition makeOpposite()
```

Makes a new condition which is the logical opposite of this condition.

Returns:

Condition

propertyChanged

```
public void propertyChanged(AEFPropertyChangeEvent evt)
                      throws java.rmi.RemoteException
```

A AEF POSDataProvider property was updated

Parameters:

`evt -`
contains details of the event

evaluate

```
public boolean evaluate(java.lang.Object newVal)
```

Evaluates the condition given a value object.

Parameters:

`newVal -`
The Object value.

Returns:

boolean True if the condition evaluates to true.

explain

```
public java.lang.String explain()
```

Returns a string explaining this condition.

Returns:

String

getTargetValue

```
public java.lang.String getTargetValue()
```

Returns the target value which this condition is looking for within the property.

Returns:

String

com.ibm.retail.AEF.automation

Class PropertyNotEqualsCondition

```
java.lang.Object
  |-com.ibm.retail.AEF.automation.AbstractCondition
    |-com.ibm.retail.AEF.automation.AbstractPropertyCondition
      |-com.ibm.retail.AEF.automation.PropertyNotEqualsCondition
```

public class **PropertyNotEqualsCondition**
extends AbstractPropertyCondition

PropertyNotEqualsCondition listens to a POSDataProvider property and indicates whether the property is not equal to a specified value. If the property value is not equal, then lock associated with this condition is notified.

Field Summary

`java.lang.Double`	`doubleValue`

Fields inherited from : class `com.ibm.retail.AEF.automation.AbstractPropertyCondition`
`category`, `name`, `nextOnly`, `value`

Fields inherited from : class `com.ibm.retail.AEF.automation.AbstractCondition`
`counterLock`, `dataProvider`, `eligible`, `id`, `idCounter`, `index`, `listeners`, `lock`, `manager`, `session`

Constructor Summary

`PropertyNotEqualsCondition(java.lang.String cat,java.lang.String cat,java.lang.String cat,boolean cat)` Constructs the condition
`PropertyNotEqualsCondition(java.lang.String cat,java.lang.String cat,java.lang.String cat)` Constructs the condition
`PropertyNotEqualsCondition(java.lang.String cat,java.lang.String cat,double cat,boolean cat)` Constructs the condition
`PropertyNotEqualsCondition(java.lang.String cat,java.lang.String cat,double cat)` Constructs the condition
`PropertyNotEqualsCondition(PropertyEqualsCondition condition)` Constructs the logical opposite of a given condition

Method Summary

`boolean`	`evaluate(java.lang.Object newVal)` Evaluates the condition given a value object.

java.lang.String	explain() Returns a string explaining this condition.
java.lang.Double	getTargetDoubleValue() Returns the target value which this condition is looking for within the property.
java.lang.String	getTargetValue() Returns the target value which this condition is looking for within the property.
Condition	makeOpposite() Makes a new condition which is the logical opposite of this condition.
void	propertyChanged(AEFPropertyChangeEvent evt) A AEF POSDataProvider property was updated

Methods inherited from : class com.ibm.retail.AEF.automation.AbstractPropertyCondition
currentValues, evaluate, evaluate, getCategory, getNextOnly, getPropertyName, parseDouble, setEligible

Methods inherited from : class com.ibm.retail.AEF.automation.AbstractCondition
addEvaluateListener, getID, getIndex, getNextID, getPOSDataProvider, getSession, getTerminalNumber, isEligible, notifyEvaluateConditionListeners, removeEvaluateListener, setConditionLock, setEligible, setID, setIndex, setLockManager, setPOSDataProvider, setSession

Methods inherited from : class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

doubleValue

```
public java.lang.Double doubleValue
```

Constructors

PropertyNotEqualsCondition

```
public PropertyNotEqualsCondition(java.lang.String cat,
                                  java.lang.String name,
                                  java.lang.String value,
                                  boolean nextValueOnly)
```

Constructs the condition

Parameters:

cat -
The property category

(continued from last page)

`name` -
The property name
`value` -
A String value which the value should not equal to evaluate true.

PropertyNotEqualsCondition

```
public PropertyNotEqualsCondition(java.lang.String cat,
                                  java.lang.String name,
                                  java.lang.String value)
```

Constructs the condition

Parameters:

`cat` -
The property category
`name` -
The property name
`value` -
A String value which the value should not equal to evaluate true.

PropertyNotEqualsCondition

```
public PropertyNotEqualsCondition(java.lang.String cat,
                                  java.lang.String name,
                                  double value,
                                  boolean nextValueOnly)
```

Constructs the condition

Parameters:

`cat` -
The property category
`name` -
The property name
`value` -
A double value which the value should not equal to evaluate true.

PropertyNotEqualsCondition

```
public PropertyNotEqualsCondition(java.lang.String cat,
                                  java.lang.String name,
                                  double value)
```

Constructs the condition

Parameters:

`cat` -
The property category
`name` -
The property name
`value` -
A double value which the value should not equal to evaluate true.

PropertyNotEqualsCondition

```
public PropertyNotEqualsCondition(PropertyEqualsCondition condition)
```

Constructs the logical opposite of a given condition

Methods

(continued from last page)

makeOpposite

```
public Condition makeOpposite()
```

Makes a new condition which is the logical opposite of this condition.

Returns:

Condition

propertyChanged

```
public void propertyChanged(AEFPropertyChangeEvent evt)
                    throws java.rmi.RemoteException
```

A AEF POSDataProvider property was updated

Parameters:

`evt -`
contains details of the event

evaluate

```
public boolean evaluate(java.lang.Object newVal)
```

Evaluates the condition given a value object.

Parameters:

`newVal -`
The Object value.

Returns:

boolean True if the condition evaluates to true.

explain

```
public java.lang.String explain()
```

Returns a string explaining this condition.

Returns:

String

getTargetValue

```
public java.lang.String getTargetValue()
```

Returns the target value which this condition is looking for within the property.

Returns:

String

getTargetDoubleValue

```
public java.lang.Double getTargetDoubleValue()
```

Returns the target value which this condition is looking for within the property.

Returns:

(continued from last page)

Double

com.ibm.retail.AEF.automation

Class PropertyRegexMatchCondition

```
java.lang.Object
  +-com.ibm.retail.AEF.automation.AbstractCondition
    +-com.ibm.retail.AEF.automation.AbstractPropertyCondition
      +-com.ibm.retail.AEF.automation.PropertyRegexMatchCondition
```

public class **PropertyRegexMatchCondition**
extends AbstractPropertyCondition

PropertyRegexMatchCondition listens to a POSDataProvider property and indicates whether the property values matches a regular expression. The property being monitored is first converted to a String if it is not already a string data type. If the property is detected to match the regular expression, the lock associated with this condition is notified, which will free a thread blocking on the lock.

Field Summary

`java.lang.String`	`regex`

Fields inherited from : class `com.ibm.retail.AEF.automation.AbstractPropertyCondition`
`category`, `name`, `nextOnly`, `value`

Fields inherited from : class `com.ibm.retail.AEF.automation.AbstractCondition`
`counterLock`, `dataProvider`, `eligible`, `id`, `idCounter`, `index`, `listeners`, `lock`, `manager`, `session`

Constructor Summary

`PropertyRegexMatchCondition(java.lang.String cat,java.lang.String cat,java.lang.String cat,boolean cat)` Constructs the condition
`PropertyRegexMatchCondition(java.lang.String cat,java.lang.String cat,java.lang.String cat)` Constructs the condition
`PropertyRegexMatchCondition(PropertyRegexNotMatchCondition condition)` Constructs the opposite of a PropertyRegexNotMatchCondition

Method Summary

`boolean`	`evaluate(java.lang.Object newVal)` Evaluates the condition given a value object.
`java.lang.String`	`explain()` Returns a string explaining this condition.

java.lang.String	getRegex() Returns the regular expression for this condition.
Condition	makeOpposite() Makes a new condition which is the logical opposite of this condition.
void	propertyChanged(AEFPropertyChangeEvent evt) A AEF POSDataProvider property was updated

Methods inherited from : class com.ibm.retail.AEF.automation.AbstractPropertyCondition
currentValues, evaluate, evaluate, getCategory, getNextOnly, getPropertyName, parseDouble, setEligible

Methods inherited from : class com.ibm.retail.AEF.automation.AbstractCondition
addEvaluateListener, getID, getIndex, getNextID, getPOSDataProvider, getSession, getTerminalNumber, isEligible, notifyEvaluateConditionListeners, removeEvaluateListener, setConditionLock, setEligible, setID, setIndex, setLockManager, setPOSDataProvider, setSession

Methods inherited from : class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

regex

```
protected java.lang.String regex
```

Constructors

PropertyRegexMatchCondition

```
public PropertyRegexMatchCondition(java.lang.String cat,
                                   java.lang.String name,
                                   java.lang.String regex,
                                   boolean nextValueOnly)
```

Constructs the condition

Parameters:

cat -
The category name of the property to be monitored.
name -
The property name to be monitored.
value -
The property value is compared to see if it contains this string.
regex -
The regular expression which will be used to compare the value of the property.
nextValueOnly -
Set to true to compare only the next value of the property. If false, all values of the property change will be compared.

PropertyRegexMatchCondition

```
public PropertyRegexMatchCondition(java.lang.String cat,
                                   java.lang.String name,
                                   java.lang.String regex)
```

Constructs the condition

Parameters:

`cat -`
The category name of the property to be monitored.
`name -`
The property name to be monitored.
`regex -`
The regular expression which will be used to compare the value of the property.

PropertyRegexMatchCondition

```
public PropertyRegexMatchCondition(PropertyRegexNotMatchCondition condition)
```

Constructs the opposite of a PropertyRegexNotMatchCondition

Parameters:

`condition -`
The PropertyRegexNotMatchCondition to construct the opposite of.

Methods

makeOpposite

```
public Condition makeOpposite()
```

Makes a new condition which is the logical opposite of this condition.

Returns:

Condition

propertyChanged

```
public void propertyChanged(AEFPropertyChangeEvent evt)
                     throws java.rmi.RemoteException
```

A AEF POSDataProvider property was updated

Parameters:

`evt -`
contains details of the event

evaluate

```
public boolean evaluate(java.lang.Object newVal)
```

Evaluates the condition given a value object.

Parameters:

`newVal -`
The Object value.

Returns:

boolean True if the condition evaluates to true.

explain

`public java.lang.String **explain**()`

Returns a string explaining this condition.

Returns:

String

getRegex

`public java.lang.String **getRegex**()`

Returns the regular expression for this condition.

Returns:

String The regular expression.

com.ibm.retail.AEF.automation

Class PropertyRegexNotMatchCondition

```
java.lang.Object
  |-com.ibm.retail.AEF.automation.AbstractCondition
    |-com.ibm.retail.AEF.automation.AbstractPropertyCondition
      |-com.ibm.retail.AEF.automation.PropertyRegexNotMatchCondition
```

public class **PropertyRegexNotMatchCondition**
extends AbstractPropertyCondition

PropertyRegexNotMatchCondition listens to a POSDataProvider property and indicates whether the property value does not match a regular expression. The property being monitored is first converted to a String if it is not already a string data type. If the property is detected to not match the regular expression, the lock associated with this condition is notified, which will free a thread blocking on the lock.

Field Summary

`java.lang.String`	`regex`

Fields inherited from : class `com.ibm.retail.AEF.automation.AbstractPropertyCondition`
`category`, `name`, `nextOnly`, `value`

Fields inherited from : class `com.ibm.retail.AEF.automation.AbstractCondition`
`counterLock`, `dataProvider`, `eligible`, `id`, `idCounter`, `index`, `listeners`, `lock`, `manager`, `session`

Constructor Summary

`PropertyRegexNotMatchCondition(java.lang.String cat,java.lang.String cat,java.lang.String cat,boolean cat)` Constructs the condition
`PropertyRegexNotMatchCondition(PropertyRegexMatchCondition condition)` Constructs the opposite of a PropertyRegexMatchCondition
`PropertyRegexNotMatchCondition(java.lang.String cat,java.lang.String cat,java.lang.String cat,java.lang.String cat)` Constructs the condition

Method Summary

`boolean`	`evaluate(java.lang.Object newVal)` Evaluates the condition given a value object.
`java.lang.String`	`explain()` Returns a string explaining this condition.

java.lang.String	getRegex() Returns the regular expression for this condition.
Condition	makeOpposite() Makes a new condition which is the logical opposite of this condition.
void	propertyChanged(AEFPropertyChangeEvent evt) A AEF POSDataProvider property was updated

Methods inherited from : class com.ibm.retail.AEF.automation.AbstractPropertyCondition
currentValues, evaluate, evaluate, getCategory, getNextOnly, getPropertyName, parseDouble, setEligible

Methods inherited from : class com.ibm.retail.AEF.automation.AbstractCondition
addEvaluateListener, getID, getIndex, getNextID, getPOSDataProvider, getSession, getTerminalNumber, isEligible, notifyEvaluateConditionListeners, removeEvaluateListener, setConditionLock, setEligible, setID, setIndex, setLockManager, setPOSDataProvider, setSession

Methods inherited from : class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

regex

protected java.lang.String **regex**

Constructors

PropertyRegexNotMatchCondition

```
public PropertyRegexNotMatchCondition(java.lang.String cat,
                                      java.lang.String name,
                                      java.lang.String regex,
                                      boolean nextValueOnly)
```

Constructs the condition

Parameters:

cat -
The category name of the property to be monitored.
name -
The property name to be monitored.
regex -
The regular expression which will be used to compare the value of the property.
nextValueOnly -
Set to true to compare only the next value of the property. If false, all values of the property change will be compared.

(continued from last page)

PropertyRegexNotMatchCondition

```
public PropertyRegexNotMatchCondition(PropertyRegexMatchCondition condition)
```

Constructs the opposite of a PropertyRegexMatchCondition

Parameters:

`condition -`
The PropertyRegexMatchCondition to construct the opposite of.

PropertyRegexNotMatchCondition

```
public PropertyRegexNotMatchCondition(java.lang.String cat,
                                      java.lang.String name,
                                      java.lang.String value,
                                      java.lang.String regex)
```

Constructs the condition

Parameters:

`cat -`
The category name of the property to be monitored.
`name -`
The property name to be monitored.
`value -`
The property value is compared to see if it contains this string.
`regex -`
The regular expression which will be used to compare the value of the property.

Methods

makeOpposite

```
public Condition makeOpposite()
```

Makes a new condition which is the logical opposite of this condition.

Returns:

Condition

propertyChanged

```
public void propertyChanged(AEFPropertyChangeEvent evt)
                    throws java.rmi.RemoteException
```

A AEF POSDataProvider property was updated

Parameters:

`evt -`
contains details of the event

evaluate

```
public boolean evaluate(java.lang.Object newVal)
```

Evaluates the condition given a value object.

Parameters:

`newVal -`
The Object value.

Returns:

(continued from last page)

boolean True if the condition evaluates to true.

getRegex

`public java.lang.String` **`getRegex`**`()`

Returns the regular expression for this condition.

Returns:

String The regular expression.

explain

`public java.lang.String` **`explain`**`()`

Returns a string explaining this condition.

Returns:

String

com.ibm.retail.AEF.automation

Interface SalesTransaction

All Superinterfaces:
Transaction

public interface **SalesTransaction**
extends Transaction

SalesTransaction is an interface representing a customer sales transaction on the POS application.

Method Summary

Customer	`addAlternateCustomerID(java.lang.String custID)` Add a loyalty customer (using alternate id) to the transaction.
void	`addCustomerBirthdate(java.util.Date birthdate)` Add the customer birthdate.
Customer	`addCustomerLoyaltyID(java.lang.String custID)` Add a loyalty customer to the transaction.
java.util.ArrayList	`addItem(ItemIdentifier itemID)` Add an item to a sales transaction.
java.util.ArrayList	`addItem(java.lang.String itemCode)` Add an item to a sales transaction.
java.util.ArrayList	`addTender(TenderIdentifier tenderIdentifier)` Adds a tender to the transaction.
java.util.ArrayList	`applyDelayedCoupons()` Signals that any pending delayed coupons should be applied.
Customer	`getCustomer()` Get the customer object.
TransactionTotals	`getTransactionTotals()` Returns the TransactionTotals object.
boolean	`hasItems()` Indicates whether the transaction contains any unvoided items.
java.util.ArrayList	`returnItem(ItemIdentifier itemID)` Return an item in a sales transaction.
java.util.ArrayList	`returnItem(java.lang.String itemCode)` Return an item in a sales transaction.

void	`setCustomer(Customer value)` Set the customer object.
Transaction	`suspend()` Suspends the transaction.
Transaction	`suspend(java.lang.String reasonCode)` Suspends the transaction.
java.util.ArrayList	`voidItem(ItemIdentifier itemID)` Void an item in a sales transaction.
java.util.ArrayList	`voidItem(java.lang.String itemCode)` Void an item in a sales transaction.
java.util.ArrayList	`voidLineItem(LineItem lineItem)` Voids a line item (item or tender) from the transaction.
java.util.ArrayList	`voidPreviousLineItem()` Voids the previous line item (item or tender).

Methods

getTransactionTotals

```
public TransactionTotals getTransactionTotals()
                                        throws java.rmi.RemoteException
```

Returns the TransactionTotals object.

Returns:

TransactionTotals

Exceptions:
`RemoteException`

addCustomerBirthdate

```
public void addCustomerBirthdate(java.util.Date birthdate)
                            throws java.rmi.RemoteException
```

Add the customer birthdate.

Exceptions:
`RemoteException`

addCustomerLoyaltyID

```
public Customer addCustomerLoyaltyID(java.lang.String custID)
                                throws AEFException,
                                       java.rmi.RemoteException
```

Add a loyalty customer to the transaction. Action ID = CustomerEntryAction

Parameters:

(continued from last page)

`Customer -`
id

Exceptions:

`RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.DAILY_LOYALTY_CARD_USAGE_LIMIT
AEFConst.CONFIG_ERROR, AEFConst.FACTORY_ERROR
AEFConst.INVALID_ARGUMENT, AEFConst.NONE
AEFConst.INVALID_MANAGER_OVERRIDE_NUMBER, AEFConst.NONE
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_REQUIRED
AEFConst.LOYALTY_CARD_EXPIRED, AEFConst.NONE
AEFConst.MANAGER_OVERRIDE_REQUIRED, AEFConst.NONE
AEFConst.POS_APP_FAILURE, AEFConst.FILE_IO_ERROR
AEFConst.POS_APP_FAILURE, AEFConst.UNRECOGNIZED_PRINT_CHARACTERS
AEFConst.PRINTER_ERROR, AEFConst.COVER_OPEN
AEFConst.PRINTER_ERROR, AEFConst.NONE
AEFConst.PRINTER_ERROR, AEFConst.PAPER_LOW
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.NONE
AEFConst.UNCHECKED_EXCEPTION, AEFConst.NONE
Common Errors

addAlternateCustomerID

```
public Customer addAlternateCustomerID(java.lang.String custID)
                                 throws AEFException,
                                        java.rmi.RemoteException
```

Add a loyalty customer (using alternate id) to the transaction. Action ID = CustomerEntryAction

Parameters:

`Alternate -`
customer id

Exceptions:

`RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.DAILY_LOYALTY_CARD_USAGE_LIMIT
AEFConst.CONFIG_ERROR, AEFConst.FACTORY_ERROR
AEFConst.INVALID_ARGUMENT, AEFConst.NONE
AEFConst.INVALID_MANAGER_OVERRIDE_NUMBER, AEFConst.NONE
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_REQUIRED
AEFConst.LOYALTY_CARD_EXPIRED, AEFConst.NONE
AEFConst.MANAGER_OVERRIDE_REQUIRED, AEFConst.NONE
AEFConst.POS_APP_FAILURE, AEFConst.FILE_IO_ERROR
AEFConst.POS_APP_FAILURE, AEFConst.UNRECOGNIZED_PRINT_CHARACTERS
AEFConst.PRINTER_ERROR, AEFConst.COVER_OPEN
AEFConst.PRINTER_ERROR, AEFConst.NONE
AEFConst.PRINTER_ERROR, AEFConst.PAPER_LOW
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.NONE
AEFConst.UNCHECKED_EXCEPTION, AEFConst.NONE
AEFConst.UNSUPPORTED_OPERATION, AEFConst.NONE
Common Errors

addItem

```
public java.util.ArrayList addItem(java.lang.String itemCode)
                            throws AEFException,
                                   java.rmi.RemoteException
```

Add an item to a sales transaction. Action ID = ItemEntryAction

Parameters:

`itemCode -`
The item code as an operator would key it.

(continued from last page)

Returns:

ArrayList An array of one or more line items which were added as a result of the call. Actual items will implement the "Item" interface. Additional Items or Coupons may be included in the array because of linked items.

Exceptions:

`RemoteException`

`AEFException -`

Among the possible AEFException error codes are:
AEFConst.AGE_RESTRICTED_ITEM, AEFConst.NONE
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.ITEM_LIMIT
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.LOYALTY_POINTS_LIMIT
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.MAX_NUMBER_OF_ITEMS
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.NEGATIVE_TRANSACTION_BALANCE
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.NUMBER_OF_COUPONS_LIMIT
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.TRANSACTION_LIMIT
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.TRANSACTION_TOTALS_TOO_LARGE
AEFConst.COUPON_EXPIRED, AEFConst.NONE
AEFConst.COUPON_VALUE_EXCEEDS_ITEM_VALUE, AEFConst.NONE
AEFConst.CONFIG_ERROR, AEFConst.FACTORY_ERROR
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_WEIGHT_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_PRICE_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_QUANTITY_REQUIRED
AEFConst.ITEM_NOT_FOR_SALE, AEFConst.NONE
AEFConst.ITEM_NOT_FOUND, AEFConst.NONE
AEFConst.INVALID_MANAGER_OVERRIDE_NUMBER, AEFConst.NONE
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_REQUIRED
AEFConst.MANAGER_OVERRIDE_REQUIRED, AEFConst.NONE
AEFConst.NO_ITEM_MATCH_FOR_COUPON, AEFConst.NONE
AEFConst.POS_APP_FAILURE, AEFConst.INVALID_APPLICATION_DATA
AEFConst.POS_APP_FAILURE, AEFConst.FILE_IO_ERROR
AEFConst.POS_APP_FAILURE, AEFConst.UNRECOGNIZED_PRINT_CHARACTERS
AEFConst.PRINTER_ERROR, AEFConst.COVER_OPEN
AEFConst.PRINTER_ERROR, AEFConst.NONE
AEFConst.PRINTER_ERROR, AEFConst.PAPER_LOW
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.HOST_REQUEST_PENDING
AEFConst.PROCEDURE_NOT_ALLOWED,
AEFConst.LOYALTY_COUPON_NOT_APPLICABLE_TO_CUSTOMER_STATUS_LEVEL
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.LOYALTY_NUMBER_REQUIRED
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.MINIMUM_SALE_NOT_SATISFIED
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.NONE
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.PAYMENT_SYSTEM_OFFLINE
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.TRANSACTION_NOT_ACTIVE
AEFConst.SCALE_ERROR, AEFConst.NONE
AEFConst.SCALE_ERROR, AEFConst.REWEIGH_ITEM
AEFConst.UNCHECKED_EXCEPTION, AEFConst.NONE
Common Errors

addItem

```
public java.util.ArrayList addItem(ItemIdentifier itemID)
                            throws AEFException,
                                   java.rmi.RemoteException
```

Add an item to a sales transaction. Action ID = ItemEntryAction

Parameters:

`itemID -`

A unique identifier for the item to be added.

Returns:

ArrayList An array of one or more line items which were added as a result of the call. Actual items will implement the "Item" interface. Additional Items or Coupons may be included in the array because of linked items.

Exceptions:

(continued from last page)

`RemoteException`

`AEFException -`

Among the possible AEFException error codes are:
AEFConst.AGE_RESTRICTED_ITEM, AEFConst.NONE
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.ITEM_LIMIT
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.LOYALTY_POINTS_LIMIT
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.MAX_NUMBER_OF_ITEMS
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.NEGATIVE_TRANSACTION_BALANCE
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.NUMBER_OF_COUPONS_LIMIT
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.TRANSACTION_LIMIT
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.TRANSACTION_TOTALS_TOO_LARGE
AEFConst.COUPON_EXPIRED, AEFConst.NONE
AEFConst.COUPON_VALUE_EXCEEDS_ITEM_VALUE, AEFConst.NONE
AEFConst.CONFIG_ERROR, AEFConst.FACTORY_ERROR
AEFConst.INVALID_ARGUMENT,
AEFConst.DEAL_PRICE_AND_DEAL_QUANTITY_MUTUALLY_EXCLUSIVE
AEFConst.INVALID_ARGUMENT, AEFConst.DEAL_PRICE_AND_DEAL_QUANTITY_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.EXTENDED_PRICE_TOO_LARGE
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_WEIGHT_PROHIBITED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_WEIGHT_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.INVALID_AMOUNT
AEFConst.INVALID_ARGUMENT, AEFConst.INVALID_DEAL_PRICE
AEFConst.INVALID_ARGUMENT, AEFConst.INVALID_DEAL_QUANTITY
AEFConst.INVALID_ARGUMENT, AEFConst.INVALID_DEPARTMENT
AEFConst.INVALID_ARGUMENT, AEFConst.INVALID_ITEM_IDENTIFIER_TYPE
AEFConst.INVALID_ARGUMENT, AEFConst.INVALID_PRICE
AEFConst.INVALID_ARGUMENT, AEFConst.INVALID_QUANTITY
AEFConst.INVALID_ARGUMENT, AEFConst.INVALID_WEIGHT
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_CLASS_PROHIBITED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_CLASS_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_DEPARTMENT_PROHIBITED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_DEPARTMENT_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.PRICE_AND_DEAL_PRICE_MUTUALLY_EXCLUSIVE
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_PRICE_OR_QUANTITY_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_PRICE_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_QUANTITY_PROHIBITED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_QUANTITY_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_STOCK_PROHIBITED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_STOCK_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.WEIGHT_AND_QUANTITY_MUTUALLY_EXCLUSIVE
AEFConst.INVALID_MANAGER_OVERRIDE_NUMBER, AEFConst.NONE
AEFConst.ITEM_NOT_FOR_SALE, AEFConst.NONE
AEFConst.ITEM_NOT_FOUND, AEFConst.NONE
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_REQUIRED
AEFConst.MANAGER_OVERRIDE_REQUIRED, AEFConst.NONE
AEFConst.NO_ITEM_MATCH_FOR_COUPON, AEFConst.NONE
AEFConst.POS_APP_FAILURE, AEFConst.INVALID_APPLICATION_DATA
AEFConst.POS_APP_FAILURE, AEFConst.FILE_IO_ERROR
AEFConst.POS_APP_FAILURE, AEFConst.UNRECOGNIZED_PRINT_CHARACTERS
AEFConst.PRINTER_ERROR, AEFConst.COVER_OPEN
AEFConst.PRINTER_ERROR, AEFConst.NONE
AEFConst.PRINTER_ERROR, AEFConst.PAPER_LOW
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.HOST_REQUEST_PENDING
AEFConst.PROCEDURE_NOT_ALLOWED,
AEFConst.LOYALTY_COUPON_NOT_APPLICABLE_TO_CUSTOMER_STATUS_LEVEL
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.LOYALTY_NUMBER_REQUIRED
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.MINIMUM_SALE_NOT_SATISFIED
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.NONE
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.PAYMENT_SYSTEM_OFFLINE
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.TRANSACTION_NOT_ACTIVE
AEFConst.SCALE_ERROR, AEFConst.ITEM_IS_ON_THE_SCALE
AEFConst.SCALE_ERROR, AEFConst.NONE
AEFConst.SCALE_ERROR, AEFConst.REWEIGH_ITEM
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.TRANSACTION_NOT_ACTIVE
AEFConst.UNCHECKED_EXCEPTION, AEFConst.NONE
Common Errors

(continued from last page)

voidItem

```
public java.util.ArrayList voidItem(ItemIdentifier itemID)
                            throws AEFException,
                                   java.rmi.RemoteException
```

Void an item in a sales transaction. Action ID = ItemVoidAction

Parameters:

`itemID -`
A unique identifier for the item to be voided.

Returns:

ArrayList An array of one or more line items which were voided as a result of the call. Actual items will implement the "Item" interface. Additional Items or Coupons may be included in the array because of linked items.

Exceptions:

`java.rmi.RemoteException`

(continued from last page)

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.APPLICATION_LIMIT_EXCEEDED,
AEFConst.FOODSTAMP_TENDER_IN_EXCESS_OF_FOODSTAMP_BALANCE
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.ITEM_LIMIT
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.NEGATIVE_TRANSACTION_BALANCE
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.TRANSACTION_LIMIT
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.TRANSACTION_TOTALS_TOO_LARGE
AEFConst.CONFIG_ERROR, AEFConst.FACTORY_ERROR
AEFConst.INVALID_ARGUMENT,
AEFConst.DEAL_PRICE_AND_DEAL_QUANTITY_MUTUALLY_EXCLUSIVE
AEFConst.INVALID_ARGUMENT, AEFConst.DEAL_PRICE_AND_DEAL_QUANTITY_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.EXTENDED_PRICE_TOO_LARGE
AEFConst.INVALID_ARGUMENT, AEFConst.INVALID_AMOUNT
AEFConst.INVALID_ARGUMENT, AEFConst.INVALID_DEAL_PRICE
AEFConst.INVALID_ARGUMENT, AEFConst.INVALID_DEAL_QUANTITY
AEFConst.INVALID_ARGUMENT, AEFConst.INVALID_DEPARTMENT
AEFConst.INVALID_ARGUMENT, AEFConst.INVALID_ITEM_IDENTIFIER_TYPE
AEFConst.INVALID_ARGUMENT, AEFConst.INVALID_PRICE
AEFConst.INVALID_ARGUMENT, AEFConst.INVALID_QUANTITY
AEFConst.INVALID_ARGUMENT, AEFConst.INVALID_WEIGHT
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_CLASS_PROHIBITED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_CLASS_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_DEPARTMENT_PROHIBITED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_DEPARTMENT_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_PRICE_OR_QUANTITY_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_PRICE_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_QUANTITY_PROHIBITED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_QUANTITY_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_STOCK_PROHIBITED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_STOCK_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_WEIGHT_PROHIBITED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_WEIGHT_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.PRICE_AND_DEAL_PRICE_MUTUALLY_EXCLUSIVE
AEFConst.INVALID_ARGUMENT, AEFConst.WEIGHT_AND_QUANTITY_MUTUALLY_EXCLUSIVE
AEFConst.INVALID_MANAGER_OVERRIDE_NUMBER, AEFConst.NONE
AEFConst.ITEM_NOT_FOUND, AEFConst.NONE
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_REQUIRED
AEFConst.MANAGER_OVERRIDE_REQUIRED, AEFConst.NONE
AEFConst.POS_APP_FAILURE, AEFConst.INVALID_APPLICATION_DATA
AEFConst.POS_APP_FAILURE, AEFConst.FILE_IO_ERROR
AEFConst.POS_APP_FAILURE, AEFConst.UNRECOGNIZED_PRINT_CHARACTERS
AEFConst.PRINTER_ERROR, AEFConst.COVER_OPEN
AEFConst.PRINTER_ERROR, AEFConst.NONE
AEFConst.PRINTER_ERROR, AEFConst.PAPER_LOW
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.NOT_APPROVED_BY_PAYMENT_SYSTEM
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.PAYMENT_SYSTEM_OFFLINE
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.TRANSACTION_NOT_ACTIVE
AEFConst.RETURN_COUPON_BEFORE_ITEM_VOID, AEFConst.NONE
AEFConst.SCALE_ERROR, AEFConst.ITEM_IS_ON_THE_SCALE
AEFConst.SCALE_ERROR, AEFConst.NONE
AEFConst.UNCHECKED_EXCEPTION, AEFConst.NONE
AEFConst.VOID_MUST_MATCH_PREVIOUS, AEFConst.NONE
Common Errors

voidItem

```
public java.util.ArrayList voidItem(java.lang.String itemCode)
                            throws AEFException,
                                   java.rmi.RemoteException
```

Void an item in a sales transaction. Action ID = ItemVoidAction

Parameters:

`itemCode -`
The item code as an operator would key it.

Returns:

(continued from last page)

ArrayList An array of one or more line items which were voided as a result of the call. Actual items will implement the "Item" interface. Additional Items or Coupons may be included in the array because of linked items.

Exceptions:

`java.rmi.RemoteException`

`AEFException -`

Among the possible AEFException error codes are:
AEFConst.APPLICATION_LIMIT_EXCEEDED,
AEFConst.FOODSTAMP_TENDER_IN_EXCESS_OF_FOODSTAMP_BALANCE
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.ITEM_LIMIT
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.NEGATIVE_TRANSACTION_BALANCE
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.TRANSACTION_LIMIT
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.TRANSACTION_TOTALS_TOO_LARGE
AEFConst.CONFIG_ERROR, AEFConst.FACTORY_ERROR
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_PRICE_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_QUANTITY_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_STOCK_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_WEIGHT_REQUIRED
AEFConst.INVALID_MANAGER_OVERRIDE_NUMBER, AEFConst.NONE
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_REQUIRED
AEFConst.MANAGER_OVERRIDE_REQUIRED, AEFConst.NONE
AEFConst.POS_APP_FAILURE, AEFConst.INVALID_APPLICATION_DATA
AEFConst.POS_APP_FAILURE, AEFConst.FILE_IO_ERROR
AEFConst.POS_APP_FAILURE, AEFConst.UNRECOGNIZED_PRINT_CHARACTERS
AEFConst.PRINTER_ERROR, AEFConst.COVER_OPEN
AEFConst.PRINTER_ERROR, AEFConst.NONE
AEFConst.PRINTER_ERROR, AEFConst.PAPER_LOW
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.NOT_APPROVED_BY_PAYMENT_SYSTEM
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.PAYMENT_SYSTEM_OFFLINE
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.TRANSACTION_NOT_ACTIVE
AEFConst.RETURN_COUPON_BEFORE_ITEM_VOID, AEFConst.NONE
AEFConst.SCALE_ERROR, AEFConst.NONE
AEFConst.UNCHECKED_EXCEPTION, AEFConst.NONE
AEFConst.VOID_MUST_MATCH_PREVIOUS, AEFConst.NONE
Common Errors

returnItem

```
public java.util.ArrayList returnItem(ItemIdentifier itemID)
                                 throws AEFException,
                                        java.rmi.RemoteException
```

Return an item in a sales transaction. Action ID = ItemReturnAction

Parameters:

`itemID -`

A unique identifier for the item to be returned.

Returns:

ArrayList An array of one or more line items which were returned as a result of the call. Actual items will implement the "Item" interface. Additional Items or Coupons may be included in the array because of linked items.

Exceptions:

`RemoteException`

(continued from last page)

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.DEPARTMENT_RETURN_LIMIT
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.ITEM_LIMIT
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.NEGATIVE_TRANSACTION_BALANCE
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.TRANSACTION_LIMIT
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.TRANSACTION_TOTALS_TOO_LARGE
AEFConst.CONFIG_ERROR, AEFConst.FACTORY_ERROR
AEFConst.COUPON_VALUE_EXCEEDS_ITEM_VALUE, AEFConst.NONE
AEFConst.INVALID_ARGUMENT,
AEFConst.DEAL_PRICE_AND_DEAL_QUANTITY_MUTUALLY_EXCLUSIVE
AEFConst.INVALID_ARGUMENT, AEFConst.DEAL_PRICE_AND_DEAL_QUANTITY_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.EXTENDED_PRICE_TOO_LARGE
AEFConst.INVALID_ARGUMENT, AEFConst.INVALID_AMOUNT
AEFConst.INVALID_ARGUMENT, AEFConst.INVALID_DEAL_PRICE
AEFConst.INVALID_ARGUMENT, AEFConst.INVALID_DEAL_QUANTITY
AEFConst.INVALID_ARGUMENT, AEFConst.INVALID_DEPARTMENT
AEFConst.INVALID_ARGUMENT, AEFConst.INVALID_ITEM_IDENTIFIER_TYPE
AEFConst.INVALID_ARGUMENT, AEFConst.INVALID_PRICE
AEFConst.INVALID_ARGUMENT, AEFConst.INVALID_QUANTITY
AEFConst.INVALID_ARGUMENT, AEFConst.INVALID_WEIGHT
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_CLASS_PROHIBITED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_CLASS_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_DEPARTMENT_PROHIBITED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_DEPARTMENT_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_PRICE_OR_QUANTITY_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_PRICE_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_QUANTITY_PROHIBITED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_QUANTITY_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_STOCK_PROHIBITED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_STOCK_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_WEIGHT_PROHIBITED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_WEIGHT_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.PRICE_AND_DEAL_PRICE_MUTUALLY_EXCLUSIVE
AEFConst.INVALID_ARGUMENT, AEFConst.WEIGHT_AND_QUANTITY_MUTUALLY_EXCLUSIVE
AEFConst.INVALID_MANAGER_OVERRIDE_NUMBER, AEFConst.NONE
AEFConst.ITEM_NOT_FOR_SALE, AEFConst.NONE
AEFConst.ITEM_NOT_FOUND, AEFConst.NONE
AEFConst.ITEM_NOT_RETURNABLE, AEFConst.NONE
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_REQUIRED
AEFConst.MANAGER_OVERRIDE_REQUIRED, AEFConst.NONE
AEFConst.NO_ITEM_MATCH_FOR_COUPON, AEFConst.NONE
AEFConst.POS_APP_FAILURE, AEFConst.INVALID_APPLICATION_DATA
AEFConst.POS_APP_FAILURE, AEFConst.FILE_IO_ERROR
AEFConst.POS_APP_FAILURE, AEFConst.UNRECOGNIZED_PRINT_CHARACTERS
AEFConst.PRINTER_ERROR, AEFConst.COVER_OPEN
AEFConst.PRINTER_ERROR, AEFConst.NONE
AEFConst.PRINTER_ERROR, AEFConst.PAPER_LOW
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.NONE
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.PAYMENT_SYSTEM_OFFLINE
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.TRANSACTION_NOT_ACTIVE
AEFConst.SCALE_ERROR, AEFConst.ITEM_IS_ON_THE_SCALE
AEFConst.SCALE_ERROR, AEFConst.NONE
AEFConst.SCALE_ERROR, AEFConst.REWEIGH_ITEM
AEFConst.UNCHECKED_EXCEPTION, AEFConst.NONE
Common Errors

returnItem

```
public java.util.ArrayList returnItem(java.lang.String itemCode)
                                 throws AEFException,
                                        java.rmi.RemoteException
```

Return an item in a sales transaction. Action ID = ItemReturnAction

Parameters:

`itemCode -`
The item code as an operator would key it.

(continued from last page)

Returns:

ArrayList An array of one or more line items which were returned as a result of the call. Actual items will implement the "Item" interface. Additional Items or Coupons may be included in the array because of linked items.

Exceptions:

`RemoteException`

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.DEPARTMENT_RETURN_LIMIT
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.ITEM_LIMIT
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.NEGATIVE_TRANSACTION_BALANCE
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.TRANSACTION_LIMIT
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.TRANSACTION_TOTALS_TOO_LARGE
AEFConst.CONFIG_ERROR, AEFConst.FACTORY_ERROR
AEFConst.COUPON_VALUE_EXCEEDS_ITEM_VALUE, AEFConst.NONE
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_CLASS_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_DEPARTMENT_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_PRICE_OR_QUANTITY_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_PRICE_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_QUANTITY_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_STOCK_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_WEIGHT_REQUIRED
AEFConst.INVALID_MANAGER_OVERRIDE_NUMBER, AEFConst.NONE
AEFConst.ITEM_NOT_FOR_SALE, AEFConst.NONE
AEFConst.ITEM_NOT_FOUND, AEFConst.NONE
AEFConst.ITEM_NOT_RETURNABLE, AEFConst.NONE
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_REQUIRED
AEFConst.MANAGER_OVERRIDE_REQUIRED, AEFConst.NONE
AEFConst.NO_ITEM_MATCH_FOR_COUPON, AEFConst.NONE
AEFConst.POS_APP_FAILURE, AEFConst.INVALID_APPLICATION_DATA
AEFConst.POS_APP_FAILURE, AEFConst.FILE_IO_ERROR
AEFConst.POS_APP_FAILURE, AEFConst.UNRECOGNIZED_PRINT_CHARACTERS
AEFConst.PRINTER_ERROR, AEFConst.COVER_OPEN
AEFConst.PRINTER_ERROR, AEFConst.NONE
AEFConst.PRINTER_ERROR, AEFConst.PAPER_LOW
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.NONE
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.PAYMENT_SYSTEM_OFFLINE
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.TRANSACTION_NOT_ACTIVE
AEFConst.SCALE_ERROR, AEFConst.NONE
AEFConst.UNCHECKED_EXCEPTION, AEFConst.NONE
Common Errors

voidLineItem

```
public java.util.ArrayList voidLineItem(LineItem lineItem)
                                  throws AEFException,
                                         java.rmi.RemoteException
```

Voids a line item (item or tender) from the transaction. Action ID = ItemVoidAction Note: This method currently only supports voiding "items". All other type of LineItems will result in an AEFException being thrown with error codes of AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.NOT_IMPLEMENTED.

Parameters:

`LineItem -`
The item to be removed from the transaction.

Returns:

ArrayList An array of one or more line items which were voided as a result of the call. Actual items will implement the "LineItem" interface.

Exceptions:

`RemoteException`

(continued from last page)

`AEFException -`
Among the possible error codes are:
AEFConst.APPLICATION_LIMIT_EXCEEDED,
AEFConst.FOODSTAMP_TENDER_IN_EXCESS_OF_FOODSTAMP_BALANCE
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.ITEM_LIMIT
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.NEGATIVE_TRANSACTION_BALANCE
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.TRANSACTION_LIMIT
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.TRANSACTION_TOTALS_TOO_LARGE
AEFConst.CONFIG_ERROR, AEFConst.FACTORY_ERROR
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_CLASS_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_DEPARTMENT_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_PRICE_OR_QUANTITY_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_PRICE_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_QUANTITY_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_STOCK_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.ITEM_WEIGHT_REQUIRED
AEFConst.INVALID_MANAGER_OVERRIDE_NUMBER, AEFConst.NONE
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_REQUIRED
AEFConst.MANAGER_OVERRIDE_REQUIRED, AEFConst.NONE
AEFConst.POS_APP_FAILURE, AEFConst.INVALID_APPLICATION_DATA
AEFConst.POS_APP_FAILURE, AEFConst.FILE_IO_ERROR
AEFConst.POS_APP_FAILURE, AEFConst.UNRECOGNIZED_PRINT_CHARACTERS
AEFConst.PRINTER_ERROR, AEFConst.COVER_OPEN
AEFConst.PRINTER_ERROR, AEFConst.NONE
AEFConst.PRINTER_ERROR, AEFConst.PAPER_LOW
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.NOT_APPROVED_BY_PAYMENT_SYSTEM
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.NOT_IMPLEMENTED
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.PAYMENT_SYSTEM_OFFLINE
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.TRANSACTION_NOT_ACTIVE
AEFConst.SCALE_ERROR, AEFConst.NONE
AEFConst.UNCHECKED_EXCEPTION, AEFConst.NONE
AEFConst.VOID_MUST_MATCH_PREVIOUS, AEFConst.NONE
Common Errors

addTender

```
public java.util.ArrayList addTender(TenderIdentifier tenderIdentifier)
                                 throws AEFException,
                                        java.rmi.RemoteException
```

Adds a tender to the transaction. The TenderIdentifier determines the type of tender and provides the necessary arguments. Action ID = AddCreditTenderAction

Parameters:

`tenderIdentifier -`
An object which determines the tender type, amount, and any additional required tender information. The TenderIdentifier must implement CreditIdentifer or MSRCreditIdentifier.

Exceptions:

`RemoteException`

(continued from last page)

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.ACCOUNT_TENDER_LIMIT
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.CHANGE_AMOUNT_LIMIT
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.LOYALTY_POINTS_LIMIT
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.NUMBER_OF_COUPONS_LIMIT
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.NUMBER_OF_TENDERS_LIMIT
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.STAND_IN_COUNT_LIMIT
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.TENDER_AMOUNT_LIMIT
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.TENDER_FLOOR_LIMIT
AEFConst.CONFIG_ERROR, AEFConst.NONE
AEFConst.CONFIG_ERROR, AEFConst.TAX_TABLE_NOT_FOUND
AEFConst.CONFIG_ERROR, AEFConst.UNDEFINED_TENDER_VARIETY
AEFConst.ENABLEMENT_ERROR, AEFConst.AUTHORIZATION_NUMBER_MISMATCH
AEFConst.INVALID_ARGUMENT, AEFConst.ACCOUNT_NUMBER_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.AMOUNT_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.AUTHORIZATION_CODE_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.CREDIT_CARD_TYPE_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.EXPIRATION_DATE_REQUIRED
AEFConst.INVALID_ARGUMENT, AEFConst.INVALID_ACCOUNT_NUMBER
AEFConst.INVALID_ARGUMENT, AEFConst.INVALID_AUTHORIZATION_CODE
AEFConst.INVALID_ARGUMENT, AEFConst.INVALID_EXPIRY_DATE
AEFConst.INVALID_ARGUMENT, AEFConst.INVALID_TENDER_TYPE
AEFConst.INVALID_ARGUMENT, AEFConst.NONE
AEFConst.INVALID_ARGUMENT, AEFConst.TRACK_DATA_PROHIBITED
AEFConst.INVALID_ARGUMENT, AEFConst.TRACK_DATA_REQUIRED
AEFConst.INVALID_MANAGER_OVERRIDE_NUMBER, AEFConst.NONE
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_REQUIRED
AEFConst.MANAGER_OVERRIDE_REQUIRED, AEFConst.NONE
AEFConst.MSR_HOOK_SWIPE_ERROR, AEFConst.MSR_SET_TO_DECODE
AEFConst.MSR_HOOK_SWIPE_ERROR, AEFConst.MSR_NOT_ENABLED
AEFConst.JAVA_POS_EXCEPTION, see exception cause() for actual javaPos exception
AEFConst.POS_APP_FAILURE, AEFConst.FILE_IO_ERROR
AEFConst.POS_APP_FAILURE, AEFConst.PAYMENT_SYSTEM_ERROR
AEFConst.POS_APP_FAILURE, AEFConst.UNRECOGNIZED_PRINT_CHARACTERS
AEFConst.PRINTER_ERROR, AEFConst.COVER_OPEN
AEFConst.PRINTER_ERROR, AEFConst.NONE
AEFConst.PRINTER_ERROR, AEFConst.PAPER_LOW
AEFConst.PROCEDURE_NOT_ALLOWED,
AEFConst.EXTERNAL_TENDER_AUTHORIZATION_SUSPENDED
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.HOST_REQUEST_PENDING
AEFConst.PROCEDURE_NOT_ALLOWED,
AEFConst.LOYTALTY_COUPON_NOT_APPLICABLE_TO_CUSTOMER_STATUS_LEVEL
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.NONE
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.NOT_IMPLEMENTED
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.TENDER_TYPE_NOT_ALLOWED
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.TRANSACTION_NOT_ACTIVE
AEFConst.TENDER_NOT_ACCEPTED, AEFConst.CALL_FOR_AUTHORIZATION
AEFConst.TENDER_NOT_ACCEPTED, AEFConst.CNCELLED_BY_OPERATOR
AEFConst.TENDER_NOT_ACCEPTED, AEFConst.CARD_EXPIRED
AEFConst.TENDER_NOT_ACCEPTED, AEFConst.CARD_TYPE_UNKNOWN
AEFConst.TENDER_NOT_ACCEPTED, AEFConst.CREDIT_NOT_AVAILABLE
AEFConst.TENDER_NOT_ACCEPTED, AEFConst.INVALID_CARD_DATA
AEFConst.TENDER_NOT_ACCEPTED, AEFConst.LOYALTY_NUMBER_REQUIRED
AEFConst.TENDER_NOT_ACCEPTED, AEFConst.PAYMENT_SYSTEM_OFFLINE
AEFConst.TENDER_NOT_ACCEPTED, AEFConst.PIN_PAD_REQUIRED
AEFConst.TENDER_NOT_ACCEPTED, AEFConst.RISK_1
AEFConst.TENDER_NOT_ACCEPTED, AEFConst.RISK_2
AEFConst.TENDER_NOT_ACCEPTED, AEFConst.RISK_3
AEFConst.TENDER_NOT_ACCEPTED, AEFConst.RISK_4
AEFConst.TENDER_NOT_ACCEPTED, AEFConst.SERVICER_CLOSED
AEFConst.TENDER_NOT_ACCEPTED, AEFConst.TENDER_EXPIRED
AEFConst.TENDER_NOT_ACCEPTED, AEFConst.TENDER_NOT_AUTHORIZED
AEFConst.TENDER_NOT_ACCEPTED, AEFConst.TOO_LONG_IN_STAND_IN
AEFConst.TENDER_NOT_ACCEPTED, AEFConst.UNKNOWN_SERVICER
AEFConst.TENDER_NOT_ACCEPTED, AEFConst.VERIFICATION_TIMEOUT
AEFConst.UNCHECKED_EXCEPTION, AEFConst.NONE
AEFConst.VERIFY_SIGNATURE, AEFConst.NONE
Common Errors

voidPreviousLineItem

```
public java.util.ArrayList voidPreviousLineItem()
                                          throws AEFException,
                                                 java.rmi.RemoteException
```

Voids the previous line item (item or tender). Action ID = CancelPreviousEntryAction

Returns:

ArrayList An array of one or more line items which were voided as a result of the call. Actual items will implement the "Item" interface. Additional Items or Coupons may be included in the array because of linked items.

Exceptions:

`RemoteException`

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.ITEM_LIMIT
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.NEGATIVE_TRANSACTION_BALANCE
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.TRANSACTION_LIMIT
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.TRANSACTION_TOTALS_TOO_LARGE
AEFConst.CONFIG_ERROR, AEFConst.FACTORY_ERROR
AEFConst.INVALID_MANAGER_OVERRIDE_NUMBER, AEFConst.NONE
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_REQUIRED
AEFConst.MANAGER_OVERRIDE_REQUIRED, AEFConst.NONE
AEFConst.POS_APP_FAILURE, AEFConst.INVALID_APPLICATION_DATA
AEFConst.POS_APP_FAILURE, AEFConst.FILE_IO_ERROR
AEFConst.POS_APP_FAILURE, AEFConst.UNRECOGNIZED_PRINT_CHARACTERS
AEFConst.PRINTER_ERROR, AEFConst.COVER_OPEN
AEFConst.PRINTER_ERROR, AEFConst.NONE
AEFConst.PRINTER_ERROR, AEFConst.PAPER_LOW
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.TRANSACTION_NOT_ACTIVE
AEFConst.SCALE_ERROR, AEFConst.NONE
AEFConst.UNCHECKED_EXCEPTION, AEFConst.NONE
AEFConst.VOID_MUST_MATCH_PREVIOUS, AEFConst.NONE
Common Errors

suspend

```
public Transaction suspend(java.lang.String reasonCode)
                     throws AEFException,
                            java.rmi.RemoteException
```

Suspends the transaction. Action ID = SuspendTransaction

Parameters:

`String -`
The reason for the suspend.

Returns:

Transaction The suspended transaction.

Exceptions:

`RemoteException`

(continued from last page)

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.CONFIG_ERROR, AEFConst.FACTORY_ERROR
AEFConst.INVALID_ARGUMENT, AEFConst.REASON_CODE_PROHIBITED
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_REQUIRED
AEFConst.MANAGER_OVERRIDE_REQUIRED, AEFConst.NONE
AEFConst.POS_APP_FAILURE, AEFConst.FILE_IO_ERROR
AEFConst.POS_APP_FAILURE, AEFConst.UNRECOGNIZED_PRINT_CHARACTERS
AEFConst.PRINTER_ERROR, AEFConst.COVER_OPEN
AEFConst.PRINTER_ERROR, AEFConst.NONE
AEFConst.PRINTER_ERROR, AEFConst.PAPER_LOW
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.HOST_REQUEST_PENDING
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.NONE
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.TRANSACTION_NOT_ACTIVE
AEFConst.UNCHECKED_EXCEPTION, AEFConst.NONE
Common Errors

suspend

```
public Transaction suspend()
                    throws AEFException,
                           java.rmi.RemoteException
```

Suspends the transaction. Find out if this is the current transaction. If so then tell the automation provider to suspend it. Otherwise throw an exception. Action ID = SuspendTransaction

Returns:

Transaction The suspended transaction.

Exceptions:

`RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.TRANSACTION_NOT_ACTIVE
AEFConst.CONFIG_ERROR, AEFConst.FACTORY_ERROR
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_REQUIRED
AEFConst.MANAGER_OVERRIDE_REQUIRED, AEFConst.NONE
AEFConst.POS_APP_FAILURE, AEFConst.FILE_IO_ERROR
AEFConst.POS_APP_FAILURE, AEFConst.UNRECOGNIZED_PRINT_CHARACTERS
AEFConst.PRINTER_ERROR, AEFConst.COVER_OPEN
AEFConst.PRINTER_ERROR, AEFConst.NONE
AEFConst.PRINTER_ERROR, AEFConst.PAPER_LOW
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.HOST_REQUEST_PENDING
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.NONE
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.TRANSACTION_NOT_ACTIVE
AEFConst.UNCHECKED_EXCEPTION, AEFConst.NONE
Common Errors

setCustomer

```
public void setCustomer(Customer value)
                 throws java.rmi.RemoteException
```

Set the customer object.

Parameters:

`Customer`

Exceptions:

`RemoteException`

getCustomer

```
public Customer getCustomer()
                     throws java.rmi.RemoteException
```

(continued from last page)

Get the customer object.

Parameters:

`Customer`

Exceptions:

`RemoteException`

applyDelayedCoupons

```
public java.util.ArrayList applyDelayedCoupons()
                                        throws AEFException,
                                               java.rmi.RemoteException
```

Signals that any pending delayed coupons should be applied. Note that this action is typically only allowed once during any transaction. Action ID = ApplyDelayedCouponsAction

Returns:

An ArrayList of LineItems. The line items will contain any delayed coupons that are applied as a result of this call.

Exceptions:

`RemoteException`

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.ITEM_LIMIT
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.LOYALTY_POINTS_LIMIT
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.NUMBER_OF_COUPONS_LIMIT
AEFConst.APPLICATION_LIMIT_EXCEEDED, AEFConst.TRANSACTION_LIMIT
AEFConst.CONFIG_ERROR, AEFConst.FACTORY_ERROR
AEFConst.INVALID_MANAGER_OVERRIDE_NUMBER, AEFConst.NONE
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_REQUIRED
AEFConst.MANAGER_OVERRIDE_REQUIRED, AEFConst.NONE
AEFConst.POS_APP_FAILURE, AEFConst.UNRECOGNIZED_PRINT_CHARACTERS
AEFConst.PRINTER_ERROR, AEFConst.COVER_OPEN
AEFConst.PRINTER_ERROR, AEFConst.NONE
AEFConst.PRINTER_ERROR, AEFConst.PAPER_LOW
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.HOST_REQUEST_PENDING
AEFConst.PROCEDURE_NOT_ALLOWED,
AEFConst.LOYALTY_COUPON_NOT_APPLICABLE_TO_CUSTOMER_STATUS_LEVEL
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.TRANSACTION_NOT_ACTIVE
AEFConst.UNCHECKED_EXCEPTION, AEFConst.NONE
Common Errors

hasItems

```
public boolean hasItems()
                  throws java.rmi.RemoteException
```

Indicates whether the transaction contains any unvoided items.

Returns:

boolean

Exceptions:

`RemoteException`

com.ibm.retail.AEF.automation

Interface SalesTransactionInfo

All Superinterfaces:
TransactionInfo, BaseInfo

public interface **SalesTransactionInfo**
extends TransactionInfo

A SalesTransactionInfo contains details of a sales transaction in progress.

Field Summary

static java.lang.String	CLASS_KEY

Method Summary

boolean	getIsTaxExempt() Indicates whether the transaction is tax exempt.
void	setIsTaxExempt(boolean flag) Sets whether the transaction is tax exempt.

Fields

CLASS_KEY

`public static final java.lang.String` **`CLASS_KEY`**

Methods

getIsTaxExempt

`public boolean` **`getIsTaxExempt`**`()`

Indicates whether the transaction is tax exempt.

Returns:

boolean True if the transaction is tax exempt.

setIsTaxExempt

`public void` **`setIsTaxExempt`**`(boolean flag)`

Sets whether the transaction is tax exempt.

Parameters:

(continued from last page)

`boolean -`
True if the transaction is tax exempt.

com.ibm.retail.AEF.automation

Interface Tender

All Superinterfaces:
LineItem

All Subinterfaces:
CreditTender

---

public interface **Tender**
extends LineItem

Tender is an interface which encapsulates transaction currency.

---

com.ibm.retail.AEF.automation

Interface TenderIdentifier

All Superinterfaces:
Identifier

All Subinterfaces:
MSRCreditIdentifier, CreditIdentifier

public interface **TenderIdentifier**
extends Identifier

TenderIdentifier is an interface which encapsulates an item code, and the type of the item code. For example, an item may be identified as a upc code, a velocity code, a sku code, or a department.

Field Summary

static java.lang.String	AMOUNT
static java.lang.String	BALANCE_DUE

Method Summary

java.lang.String	getAmount() Gets the tender amount.
java.lang.String	getKey() Gets the tender key.
void	setAmount(java.lang.String amount) Sets the tender amount.

Fields

AMOUNT

public static final java.lang.String **AMOUNT**

BALANCE_DUE

public static final java.lang.String **BALANCE_DUE**

Methods

(continued from last page)

setAmount

`public void **setAmount**(java.lang.String amount)`

Sets the tender amount.

Parameters:

`amount`

getAmount

`public java.lang.String **getAmount**()`

Gets the tender amount.

Returns:

String The amount of the tender.

getKey

`public java.lang.String **getKey**()`

Gets the tender key. The tender key is used to determine the keys in classes.properties chain for the classes used to add a tender or void a tender. For example, if the method returns "Cash", then "AddCashTenderAction" is the key used to determine the action class for adding the tender to a transaction.

Returns:

String

com.ibm.retail.AEF.automation

Interface TenderInfo

All Superinterfaces:
LineItemInfo, BaseInfo

public interface **TenderInfo**
extends LineItemInfo

TenderInfo is an interface which represents data for a tender in a transaction.

Field Summary

static java.lang.String	CLASS_KEY

Method Summary

java.lang.String	getAmount() Returns the tender amount.
void	setAmount(java.lang.String amount) Sets the tender amount.

Fields

CLASS_KEY

public static final java.lang.String **CLASS_KEY**

Methods

getAmount

public java.lang.String **getAmount**()

Returns the tender amount.

Returns:

String The tender amount.

setAmount

public void **setAmount**(java.lang.String amount)

Sets the tender amount.

Parameters:

(continued from last page)

`String` -
The tender amount.

com.ibm.retail.AEF.automation

Interface Transaction

All Subinterfaces:
SalesTransaction

public interface **Transaction**
extends java.rmi.Remote

Transaction is an interface for conducting a transaction on the POS application.

Method Summary

void	`addLineItem(LineItem lineItem)` Adds a LineItem instance to the transaction's collection.
void	`addLineItemList(java.util.ArrayList list)` Adds a collection of LineItem instances to the transaction's collection.
boolean	`getIsActive()` Indicates if the transaction is currently active (in progress).
java.util.ArrayList	`getLineItemArrays()` Gets an array.
java.util.Iterator	`getLineItems()` Gets the LineItems in the transaction.
AEFSession	`getSession()` Get the AEFSession object.
TransactionInfo	`getTransactionInfo()` Returns the transaction information
void	`setIsActive(boolean isActive)` Sets the indicator determining if the transaciton is currently active (in progress).
void	`setSession(AEFSession session)` Set the AEFSession object.
void	`voidTransaction()` Voids the transaction

Methods

(continued from last page)

getTransactionInfo

```
public TransactionInfo getTransactionInfo()
                                    throws java.rmi.RemoteException
```

Returns the transaction information

Returns:

TransactionInfo The transaction information.

Exceptions:

`RemoteException`

voidTransaction

```
public void voidTransaction()
                     throws AEFException,
                            java.rmi.RemoteException
```

Voids the transaction

Exceptions:

`AEFException -`
Among the AEFException error codes are:
AEFConst.CONFIG_ERROR, AEFConst.FACTORY_ERROR
AEFConst.INVALID_MANAGER_OVERRIDE_NUMBER, AEFConst.NONE
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_NOT_REMOVED
AEFConst.KEYLOCK_ERROR, AEFConst.MANAGER_KEY_REQUIRED
AEFConst.POS_APP_FAILURE, AEFConst.FILE_IO_ERROR
AEFConst.POS_APP_FAILURE, AEFConst.UNRECOGNIZED_PRINT_CHARACTERS
AEFConst.PRINTER_ERROR, AEFConst.COVER_OPEN
AEFConst.PRINTER_ERROR, AEFConst.NONE
AEFConst.PRINTER_ERROR, AEFConst.PAPER_LOW
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.NONE
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.PAYMENT_SYSTEM_OFFLINE
AEFConst.PROCEDURE_NOT_ALLOWED, AEFConst.TRANSACTION_NOT_ACTIVE
AEFConst.UNCHECKED_EXCEPTION, AEFConst.NONE
Common Errors
`RemoteException`

getLineItems

```
public java.util.Iterator getLineItems()
                                  throws java.rmi.RemoteException
```

Gets the LineItems in the transaction.

Returns:

Iterator An iterator over the LineItems.

Exceptions:

`RemoteException`

getLineItemArrays

```
public java.util.ArrayList getLineItemArrays()
                                     throws java.rmi.RemoteException
```

Gets an array. Each element in the array is an array of linked LineItems in the transaction.

Returns:

(continued from last page)

ArrayList An array of LineItem arrays.

Exceptions:
`RemoteException`

addLineItem

```
public void addLineItem(LineItem lineItem)
                 throws java.rmi.RemoteException
```

Adds a LineItem instance to the transaction's collection.

Parameters:
`LineItem`

Exceptions:
`RemoteException`

addLineItemList

```
public void addLineItemList(java.util.ArrayList list)
                     throws java.rmi.RemoteException
```

Adds a collection of LineItem instances to the transaction's collection.

Parameters:
`list -`
An ArrayList of LineItems

Exceptions:
`RemoteException`

setSession

```
public void setSession(AEFSession session)
                throws java.rmi.RemoteException
```

Set the AEFSession object.

Parameters:
`AEFSession`

Exceptions:
`RemoteException`

getSession

```
public AEFSession getSession()
                     throws java.rmi.RemoteException
```

Get the AEFSession object.

Parameters:
`AEFSession`

Exceptions:
`RemoteException`

(continued from last page)

getIsActive

```
public boolean getIsActive()
                    throws java.rmi.RemoteException
```

Indicates if the transaction is currently active (in progress).

Returns:

boolean True if transaction is active.

Exceptions:

`RemoteException`

setIsActive

```
public void setIsActive(boolean isActive)
                 throws java.rmi.RemoteException
```

Sets the indicator determining if the transaciton is currently active (in progress).

Parameters:

`isActive -`
True if transaction is currently active.

Exceptions:

`RemoteException`

com.ibm.retail.AEF.automation

Interface TransactionIdentifier

All Superinterfaces:
Identifier

All Known Implementing Classes:
TransactionIdentifierImpl

public interface **TransactionIdentifier**
extends Identifier

TransactionIdentifier is an interface which encapsulates a transaction. For example, a transaction may be identified as NO_SALE, REGULAR_SALE, or any other type listed in AEFConst.java.

Field Summary

static java.lang.String	CLASS_KEY
static java.lang.String	DATA
static java.lang.String	MODIFIER
static java.lang.String	TRANSACTION_TYPE

Method Summary

java.lang.String	getData() Gets the data to be included with the transaction.
java.lang.String	getModifier() Gets the transaction modifier.
java.lang.String	getTransactionType() Gets the transaction type.
void	setData(java.lang.String transactionData) Sets the data to be included with the transaction.
void	setModifier(java.lang.String modifier) Sets the transaction modifier.
void	setTransactionType(java.lang.String transactionType) Sets the transaction type.

Fields

(continued from last page)

CLASS_KEY

`public static final java.lang.String` **`CLASS_KEY`**

TRANSACTION_TYPE

`public static final java.lang.String` **`TRANSACTION_TYPE`**

DATA

`public static final java.lang.String` **`DATA`**

MODIFIER

`public static final java.lang.String` **`MODIFIER`**

Methods

setTransactionType

`public void` **`setTransactionType`**`(java.lang.String transactionType)`

Sets the transaction type. This should be one of the types listed in AEFConst.java.

Parameters:

`transactionType -`
The transaction type from AEFConst.java.

See Also:

com.ibm.retail.si.util.AEFConst for the valid transaction types & modifiers.

setModifier

`public void` **`setModifier`**`(java.lang.String modifier)`

Sets the transaction modifier. This should be one of the modifiers listed in AEFConst.java.

Parameters:

`modifier -`
The transaction modifier from AEFConst.java.

See Also:

com.ibm.retail.si.util.AEFConst for the valid transaction types & modifiers.

setData

`public void` **`setData`**`(java.lang.String transactionData)`

Sets the data to be included with the transaction.

Parameters:

(continued from last page)

`transactionData -`
A String of data to be included in the transaction.

getTransactionType

`public java.lang.String` **`getTransactionType`**`()`

Gets the transaction type. This should be one of the types listed in AEFConst.java.

Returns:

String The transaction type from AEFConst.java.

See Also:

com.ibm.retail.si.util.AEFConst for the valid transaction types.

getData

`public java.lang.String` **`getData`**`()`

Gets the data to be included with the transaction.

Returns:

String A String of data to be included in the transaction.

getModifier

`public java.lang.String` **`getModifier`**`()`

Gets the transaction modifier. This should be one of the modifiers listed in AEFConst.java.

Parameters:

`modifier -`
The transaction modifier from AEFConst.java.

See Also:

com.ibm.retail.si.util.AEFConst for the valid transaction types & modifiers.

com.ibm.retail.AEF.automation

Class TransactionIdentifierImpl

```
java.lang.Object
  |-java.util.AbstractMap
    |-java.util.HashMap
      |-com.ibm.retail.AEF.automation.IdentifierImpl
        |-com.ibm.retail.AEF.automation.TransactionIdentifierImpl
```

All Implemented interfaces:
TransactionIdentifier, java.util.Map, java.io.Serializable, java.lang.Cloneable, java.util.Map, java.io.Serializable, Identifier

---

public class **TransactionIdentifierImpl**
extends IdentifierImpl
implements Identifier, java.io.Serializable, java.util.Map, java.lang.Cloneable, java.io.Serializable, java.util.Map, TransactionIdentifier

TransactionIdentifier is a class which contains information about a transaction.

Constructor Summary

`TransactionIdentifierImpl()` Constructor.
`TransactionIdentifierImpl(java.lang.String transactionType,java.lang.String transactionType)` Constructor.
`TransactionIdentifierImpl(java.lang.String transactionType,java.lang.String transactionType,java.lang.String transactionType)` Constructor.

Method Summary

`java.lang.String`	`getData()` Gets the data to be included with the transaction.
`java.lang.String`	`getModifier()` Gets the transaction modifier.
`java.lang.String`	`getTransactionType()` Gets the transaction type.
`void`	`setData(java.lang.String transactionData)` Sets the data to be included with the transaction.
`void`	`setModifier(java.lang.String modifier)` Sets the transaction modifier.

void	`setTransactionType(java.lang.String transactionType)` Sets the transaction type.

Methods inherited from : class `com.ibm.retail.AEF.automation.IdentifierImpl`
`toString`

Methods inherited from : class `java.util.HashMap`
`clear, clone, containsKey, containsValue, entrySet, get, isEmpty, keySet, put, putAll, remove, size, values`

Methods inherited from : class `java.util.AbstractMap`
`clear, clone, containsKey, containsValue, entrySet, equals, get, hashCode, isEmpty, keySet, put, putAll, remove, size, toString, values`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

TransactionIdentifierImpl

```
public TransactionIdentifierImpl()
```

Constructor.

TransactionIdentifierImpl

```
public TransactionIdentifierImpl(java.lang.String transactionType,
                                 java.lang.String modifier)
```

Constructor.

Parameters:

`transactionType -`
The transaction type from AEFConst.java.
`modifier -`
A modifier such as "return", "send", "vat". Use constants from AEFConst.java

See Also:

com.ibm.retail.si.util.AEFConst for modifier constants.

TransactionIdentifierImpl

```
public TransactionIdentifierImpl(java.lang.String transactionType,
                                 java.lang.String modifier,
                                 java.lang.String data)
```

Constructor.

Parameters:

(continued from last page)

`transactionType -`
The transaction type from AEFConst.java.
`modifier -`
A modifier such as "return", "send", "vat". Use constants from AEFConst.java
`data -`
Any additional data to be included in the transaction.

See Also:

com.ibm.retail.si.util.AEFConst for modifier constants.

Methods

setTransactionType

`public void` **`setTransactionType`**`(java.lang.String transactionType)`

Sets the transaction type. This should be one of the types listed in AEFConst.java.

Parameters:

`transactionType -`
The transaction type from AEFConst.java.

See Also:

com.ibm.retail.si.util.AEFConst for the valid transaction types & modifiers.

setModifier

`public void` **`setModifier`**`(java.lang.String modifier)`

Sets the transaction modifier. This should be one of the modifiers listed in AEFConst.java.

Parameters:

`modifier -`
The transaction modifier from AEFConst.java.

See Also:

com.ibm.retail.si.util.AEFConst for the valid transaction types & modifiers.

setData

`public void` **`setData`**`(java.lang.String transactionData)`

Sets the data to be included with the transaction.

Parameters:

`transactionData -`
A String of data to be included in the transaction.

getTransactionType

`public java.lang.String` **`getTransactionType`**`()`

Gets the transaction type. This should be one of the types listed in AEFConst.java.

Returns:

String The transaction type from AEFConst.java.

See Also:

(continued from last page)

com.ibm.retail.si.util.AEFConst for the valid transaction types.

getData

`public java.lang.String **getData**()`

Gets the data to be included with the transaction.

Returns:

String A String of data to be included in the transaction.

getModifier

`public java.lang.String **getModifier**()`

Gets the transaction modifier. This should be one of the modifiers listed in AEFConst.java.

Parameters:

`modifier -`
The transaction modifier from AEFConst.java.

See Also:

com.ibm.retail.si.util.AEFConst for the valid transaction types & modifiers.

com.ibm.retail.AEF.automation

Interface TransactionInfo

All Superinterfaces:
BaseInfo

All Subinterfaces:
SalesTransactionInfo

public interface **TransactionInfo**
extends BaseInfo

A TransactionInfo contains details of a transaction in progress.

Field Summary

static java.lang.String	CLASS_KEY

Method Summary

boolean	getIsVoided() Indicates whether the transaction as been voided.
java.util.Date	getTransactionDate() Returns the transaction date & time.
java.lang.String	getTransactionID() Returns the transaction id.
void	setIsVoided(boolean flag) Sets whether the transaction as been voided.
void	setTransactionDate(java.util.Date date) Sets the transaction date & time.
void	setTransactionID(java.lang.String id) Sets the transaction id.

Fields

CLASS_KEY

public static final java.lang.String **CLASS_KEY**

Methods

(continued from last page)

getTransactionID

`public java.lang.String **getTransactionID**()`

Returns the transaction id.

Returns:

String The transaction id.

setTransactionID

`public void **setTransactionID**(java.lang.String id)`

Sets the transaction id.

Parameters:

`String -`
The transaction id.

getTransactionDate

`public java.util.Date **getTransactionDate**()`

Returns the transaction date & time.

Returns:

Date The transaction date & time.

setTransactionDate

`public void **setTransactionDate**(java.util.Date date)`

Sets the transaction date & time.

Parameters:

`Date -`
The transaction date & time.

getIsVoided

`public boolean **getIsVoided**()`

Indicates whether the transaction as been voided.

Returns:

boolean True if the transaction is voided.

setIsVoided

`public void **setIsVoided**(boolean flag)`

Sets whether the transaction as been voided.

Parameters:

`boolean -`
True if the transaction is voided.

com.ibm.retail.AEF.automation

Interface TransactionTotals

All Superinterfaces:
BaseInfo

public interface **TransactionTotals**
extends BaseInfo

TransactionTotals encapsulates the totals associated with a sales transaction. Properties in this object are typically changed whenever an item is added to or removed from a transaction, or if a discount is applied.

Method Summary

`java.lang.String`	`getBalanceDue()` Get the Balance Due
`java.lang.String`	`getChangeDue()` Get the Change Due
`java.lang.String`	`getCouponTotal()` Get the Coupon Total
`java.lang.String`	`getFoodstampBalanceDue()` Get the Foodstamp Balance Due
`java.lang.String`	`getFoodstampChangeDue()` Get the Foodstamp Change Due
`java.lang.String`	`getFoodstampTotal()` Get the Foodstamp Total
`java.lang.String`	`getSubTotal()` Get the SubTotal Amount Due
`java.lang.String`	`getTax()` Get the Tax Amount Due
`java.lang.String`	`getTotal()` Get the Transaction Total Amount
`java.lang.String`	`getTotalCoupons()` Get the Total Coupons
`java.lang.String`	`getTotalItems()` Get the Total Items
`java.lang.String`	`getTotalSavings()` Get the Total Savings

void	`setBalanceDue(java.lang.String amt)` Set the Balance Due
void	`setChangeDue(java.lang.String amt)` Set the Change Due
void	`setCouponTotal(java.lang.String amt)` Set the Coupon Total
void	`setFoodstampBalanceDue(java.lang.String amt)` Set the Foodstamp Balance Due
void	`setFoodstampChangeDue(java.lang.String amt)` Set the Foodstamp Change Due
void	`setFoodstampTotal(java.lang.String foodstampTotal)` Set the Foodstamp Total
void	`setSubTotal(java.lang.String amt)` Set the SubTotal Amount Due
void	`setTax(java.lang.String amt)` Set the Tax Amount Due
void	`setTotal(java.lang.String amt)` Set the Transaction Total
void	`setTotalCoupons(java.lang.String amt)` Set the Total Coupons
void	`setTotalItems(java.lang.String amt)` Set the Total Items
void	`setTotalSavings(java.lang.String amt)` Set the Total Savings

Methods

getTotal

`public java.lang.String` **`getTotal`**`()`

Get the Transaction Total Amount

Returns:

total

setTotal

`public void` **`setTotal`**`(java.lang.String amt)`

Set the Transaction Total

(continued from last page)

Parameters:
`total`

getSubTotal

`public java.lang.String **getSubTotal**()`

Get the SubTotal Amount Due

Returns:

subtotal

setSubTotal

`public void **setSubTotal**(java.lang.String amt)`

Set the SubTotal Amount Due

Parameters:
`subtotal`

getTax

`public java.lang.String **getTax**()`

Get the Tax Amount Due

Returns:

tax

setTax

`public void **setTax**(java.lang.String amt)`

Set the Tax Amount Due

Parameters:
`tax`

getBalanceDue

`public java.lang.String **getBalanceDue**()`

Get the Balance Due

Returns:

amt due

setBalanceDue

`public void **setBalanceDue**(java.lang.String amt)`

Set the Balance Due

Parameters:
`amt` - due

(continued from last page)

getFoodstampBalanceDue

`public java.lang.String` **`getFoodstampBalanceDue`**`()`

Get the Foodstamp Balance Due

Returns:

amt due

setFoodstampBalanceDue

`public void` **`setFoodstampBalanceDue`**`(java.lang.String amt)`

Set the Foodstamp Balance Due

Parameters:

`amt` - due

getFoodstampTotal

`public java.lang.String` **`getFoodstampTotal`**`()`

Get the Foodstamp Total

Returns:

foodstamp total

setFoodstampTotal

`public void` **`setFoodstampTotal`**`(java.lang.String foodstampTotal)`

Set the Foodstamp Total

Parameters:

`foodstampTotal`

setChangeDue

`public void` **`setChangeDue`**`(java.lang.String amt)`

Set the Change Due

Parameters:

`change` - due

getChangeDue

`public java.lang.String` **`getChangeDue`**`()`

Get the Change Due

Returns:

change due

setFoodstampChangeDue

`public void` **`setFoodstampChangeDue`**`(java.lang.String amt)`

Set the Foodstamp Change Due

(continued from last page)

Parameters:
`change -`
due

getFoodstampChangeDue

`public java.lang.String` **`getFoodstampChangeDue`**`()`

Get the Foodstamp Change Due

Returns:

change due

getCouponTotal

`public java.lang.String` **`getCouponTotal`**`()`

Get the Coupon Total

Returns:

coupon total

setCouponTotal

`public void` **`setCouponTotal`**`(java.lang.String amt)`

Set the Coupon Total

Parameters:
`coupon -`
total

getTotalItems

`public java.lang.String` **`getTotalItems`**`()`

Get the Total Items

Returns:

total items

setTotalItems

`public void` **`setTotalItems`**`(java.lang.String amt)`

Set the Total Items

Parameters:
`total -`
Items

getTotalCoupons

`public java.lang.String` **`getTotalCoupons`**`()`

Get the Total Coupons

Returns:

total coupons

setTotalCoupons

`public void **setTotalCoupons**(java.lang.String amt)`

Set the Total Coupons

Parameters:

`total` - coupons

getTotalSavings

`public java.lang.String **getTotalSavings**()`

Get the Total Savings

Returns:

total savings

setTotalSavings

`public void **setTotalSavings**(java.lang.String amt)`

Set the Total Savings

Parameters:

`total` - savings

Package
com.ibm.retail.AEF.client

Provides proxy objects to facilitate remote listener clients of the POSDataProvider interface.

The client package provides listener proxy objects to help facilitate remote clients using the POSDataProvider event listener interfaces. The listener proxy objects provide two essential functions:

1. **Remote listener (RMI) capabability**
   Listener proxy objects extend the appropriate RMI server classes which handle the RMI duties for the client (so the client does not need to deal with rmi server issues).

2. **Event dispatching**
   The proxy objects also perform the role of event dispatching. Since event notification calls from the AEF POSDataProvider are synchronous, clients should perform their event processing on a separate thread so as not to affect the performance of the AEF. By default, the listener proxy object will perform event queuing on a separate AEF event dispatcher thread. This relieves the client of any "thread swapping" responsibilities and insures that the AEF event dispatching performance is not affected by client processing.

Sample Usage:

```
import com.ibm.retail.AEF.event.DiscountListener;
import com.ibm.retail.AEF.event.DiscountEvent;
import com.ibm.retail.AEF.client.DiscountListenerProxy;
import com.ibm.retail.AEF.data.POSDataProvider;

// Usage Note: The client does not need to extend java.rmi.server classes, this is handled
// by the proxy object.
public class MyClient implements DiscountListener
{
  /**
   *  Constructor.
   *
   *  @param   dataProvider POSDataProvider for an AEFSession
   */
  public MyClient(POSDataProvider dataProvider)
  {
    this.dataProvider = dataProvider;
    try
    {
       // Use proxy to register as a listener to discount events
       DiscountListenerProxy proxy = new DiscountListenerProxy(this,dataProvider);
    }
    catch (RemoteException re)
    {
       System.err.println("RemoteException: Failed to register as listener" + re);
    }
    catch (AEFException ae)
    {
       System.err.println("AEFException: Failed to register as listener" + ae);
    }
  }

  /**
   *  A discount was applied to an item or transaction.
   *
   *  @param   evt contains details of the discount event
   *  @throws  RemoteException  if a listener can not be notified
   */
  public void discountApplied(DiscountEvent evt) throws java.rmi.RemoteException
  {
     // Usage Note: The client does not need to dispatch the event on another
     // thread before calling updateGUI().  Dispatching has been handled by the proxy object.
     updateGUI(evt);
  }
```

(continued from last page)

```
{
```

com.ibm.retail.AEF.client

Class AEFEventListenerProxy

```
java.lang.Object
  +-java.rmi.server.RemoteObject
    +-java.rmi.server.RemoteServer
      +-java.rmi.server.UnicastRemoteObject
        +-com.ibm.retail.AEF.util.AEFRemoteObject
          +-com.ibm.retail.AEF.client.AEFEventListenerProxy
```

Direct Known Subclasses:
WorkstationStatusListenerProxy, TransactionTotalsListenerProxy, TransactionStatusListenerProxy, TenderListenerProxy, StateChangeListenerProxy, SessionStatusListenerProxy, ScaleListenerProxy, ReportListenerProxy, PointsListenerProxy, OptionsListenerProxy, OperatorListenerProxy, ItemSalesListenerProxy, GenericListenerProxy, DiscountListenerProxy, CustomerListenerProxy, CouponListenerProxy, CashReceiptListenerProxy, AEFPropertyListenerProxyImpl

---

public class **AEFEventListenerProxy**
extends AEFRemoteObject

The AEFEventListenerProxy class provides a base class for the client listener proxy objects used to monitor the events of an associated terminal session.

Remote proxy objects extend the appropriate RMI server classes which handle the RMI duties for the client (so the client does not need to deal with rmi server issues). The proxy objects also perform the role of event dispatching.

By default, the listener proxy object will perform event queuing on an AEF event thread. This relieves the client of any "thread swapping" responsibilities and insures that the AEF event dispatching performance is not affected by client processing. To override this default behavior, use the *setDispatchQueue()*method. The proxy utilizes the EventDispatcher for queueing event listener notification.

To further ensure that the AEF event dispatching thread is not blocked by non-responsive clients, proxies use a default RMI timeout value as specified in config.properties. This value may be overridden using a timeout value on the proxy constructor.

To use a proxy object, the client must implement the listener interface and get the POSDataProvider object from the AEFSession to monitor. The proxy object is then constructed passing a reference to the client and the data provider. The proxy object performs the listener registration and forwards all events from the data provider to the client.

See Also:
Sample Usage, com.ibm.retail.AEF.client.EventDispatcher EventDispatcher

---

Field Summary

java.lang.String	dispatchQueueName
static int	SOCKET_CONNECT_TIMEOUT
static int	SOCKET_READ_TIMEOUT

Fields inherited from : class com.ibm.retail.AEF.util.AEFRemoteObject
DEFAULT_CONNECT_TIMEOUT, DEFAULT_READ_TIMEOUT

Fields inherited from : class `java.rmi.server.RemoteObject`
`ref`

Constructor Summary

`AEFEventListenerProxy()` Construct an AEFEventListenerProxy
`AEFEventListenerProxy(int soTimeout,int soTimeout)` Construct an AEFEventListenerProxy.

Method Summary

`void`	`dispatchEvent(java.lang.Object evt,java.lang.Object evt,java.lang.reflect.Method evt)` Dispatch an event.
`java.lang.String`	`getDispatchQueueName()` Get the dispatch queue name
`static java.lang.reflect.Method`	`getListenerMethod(java.lang.Class eventClass,java.lang.Class eventClass,java.lang.String eventClass)` Determine a listener method
`void`	`setDispatchQueueName(java.lang.String queue)` Set the dispatch queue name.

Methods inherited from : class `com.ibm.retail.AEF.util.AEFRemoteObject`
`getClientSocketConnectTimeout, getClientSocketReadTimeout`

Methods inherited from : class `java.rmi.server.UnicastRemoteObject`
`clone, exportObject, exportObject, exportObject, unexportObject`

Methods inherited from : class `java.rmi.server.RemoteServer`
`getClientHost, getLog, setLog`

Methods inherited from : class `java.rmi.server.RemoteObject`
`equals, getRef, hashCode, toString, toStub`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

SOCKET_READ_TIMEOUT

`public static int` **`SOCKET_READ_TIMEOUT`**

SOCKET_CONNECT_TIMEOUT

`public static int` **`SOCKET_CONNECT_TIMEOUT`**

dispatchQueueName

`protected java.lang.String` **`dispatchQueueName`**

Constructors

AEFEventListenerProxy

`public` **`AEFEventListenerProxy`**`()`

Construct an AEFEventListenerProxy

AEFEventListenerProxy

```
public AEFEventListenerProxy(int soTimeout,
                             int connectTimeout)
```

Construct an AEFEventListenerProxy. This constructor overrides the default timeout value.

Parameters:

`RMI -`
read timeout value
`RMI -`
connect timeout value

Methods

getDispatchQueueName

`public java.lang.String` **`getDispatchQueueName`**`()`

Get the dispatch queue name

Returns:

dispatch queue name

See Also:

com.ibm.retail.AEF.client.EventDispatcher#AEF_QUEUE AEF_QUEUE

com.ibm.retail.AEF.client.EventDispatcher#AWT_QUEUE AWT_QUEUE

com.ibm.retail.AEF.client.EventDispatcher#NO_QUEUE NO_QUEUE

setDispatchQueueName

```
public void setDispatchQueueName(java.lang.String queue)
```

Set the dispatch queue name. Defaults to the AEF_QUEUE.

Parameters:

`queue -`
the dispatch queue name identifier

See Also:

com.ibm.retail.AEF.client.EventDispatcher#AEF_QUEUE AEF_QUEUE

com.ibm.retail.AEF.client.EventDispatcher#AWT_QUEUE AWT_QUEUE

com.ibm.retail.AEF.client.EventDispatcher#NO_QUEUE NO_QUEUE

dispatchEvent

```
public void dispatchEvent(java.lang.Object evt,
                          java.lang.Object listener,
                          java.lang.reflect.Method listenerMethod)
```

Dispatch an event.

Parameters:

`evt -`
the event object to be dispatched
`listener -`
the listener object to receive the event
`listenerMethod -`
the listener method to be invoked for notification

getListenerMethod

```
protected static java.lang.reflect.Method getListenerMethod(java.lang.Class
eventClass,
                                                            java.lang.Class
listenerClass,
                                                            java.lang.String
methodName)
```

Determine a listener method

Parameters:

`eventClass -`
the event class
`listenerClass -`
the listener class
`methodName -`
the listener method name

Returns:

the method for listener notification

com.ibm.retail.AEF.client

Interface AEFListenerProxyInterface

All Known Implementing Classes:

WorkstationStatusListenerProxy, TransactionTotalsListenerProxy, TransactionStatusListenerProxy, TenderListenerProxy, StateChangeListenerProxy, SessionStatusListenerProxy, ScaleListenerProxy, ReportListenerProxy, PointsListenerProxy, OptionsListenerProxy, OperatorListenerProxy, ItemSalesListenerProxy, GenericListenerProxy, DiscountListenerProxy, CustomerListenerProxy, CouponListenerProxy, CashReceiptListenerProxy

---

public interface **AEFListenerProxyInterface**
extends java.rmi.Remote

The AEFListenerProxy interface is the base interface for client proxy classes.

Method Summary

void	`removeListener()` Remove proxy listener.

Methods

removeListener

```
public void removeListener()
                    throws java.rmi.RemoteException,
                           AEFException
```

Remove proxy listener.

Exceptions:

`java.rmi.RemoteException`

`AEFException` -
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

com.ibm.retail.AEF.client

Interface AEFPropertyListenerProxy

All Known Implementing Classes:
AEFPropertyListenerProxyImpl

public interface **AEFPropertyListenerProxy**
extends java.rmi.Remote

The AEFPropertyListenerProxy interface defines a proxy interface for remote listeners to monitor the AEFPropertyChanges of its associated terminal session.

Remote proxy objects extend the appropriate RMI server classes which handle the RMI duties for the client (so the client does not need to deal with rmi server issues). The proxy objects also perform the role of event dispatching.

By default, the listener proxy object will perform event queuing on an AEF event thread. This relieves the client of any "thread swapping" responsibilities and insures that the AEF event dispatching performance is not affected by client processing. To override this default behavior, use the *setDispatchQueue()*method. The proxy utilizes the EventDispatcher for queueing event listener notification.

To use a proxy object, the client must implement the listener interface and get the POSDataProvider object from the AEFSession to monitor. The proxy object is then constructed passing a reference to the client and the data provider. The proxy object performs the listener registration and forwards all events from the data provider to the client.

See Also:
Sample Usage, com.ibm.retail.AEF.client.EventDispatcher EventDispatcher

Method Summary

void	`addAEFPropertyChangeListener()` Add a property change listener to listen to all data provider events.
void	`addAEFPropertyChangeListener(java.lang.String category)` Add a filtered property change listener to listen to data provider events for a specific category.
void	`addAEFPropertyChangeListener(java.lang.String category,java.lang.String category)` Add a filtered property change listener to listen to data provider events for a specific property.
void	`removeAEFPropertyChangeListener()` Remove a property change listener.
void	`removeAEFPropertyChangeListener(java.lang.String category)` Remove a filtered property change listener.
void	`removeAEFPropertyChangeListener(java.lang.String category,java.lang.String category)` Remove a filtered property change listener.

(continued from last page)

Methods

addAEFPropertyChangeListener

```
public void addAEFPropertyChangeListener()
                                    throws java.rmi.RemoteException
```

Add a property change listener to listen to all data provider events.

Exceptions:

`RemoteException` -
if remote access fails

removeAEFPropertyChangeListener

```
public void removeAEFPropertyChangeListener()
                                       throws java.rmi.RemoteException
```

Remove a property change listener.

Exceptions:

`RemoteException` -
if remote access fails

addAEFPropertyChangeListener

```
public void addAEFPropertyChangeListener(java.lang.String category)
                                    throws java.rmi.RemoteException
```

Add a filtered property change listener to listen to data provider events for a specific category.

Parameters:

`category` -
identifies the category of data

Exceptions:

`RemoteException` -
if remote access fails

removeAEFPropertyChangeListener

```
public void removeAEFPropertyChangeListener(java.lang.String category)
                                       throws java.rmi.RemoteException
```

Remove a filtered property change listener.

Parameters:

`category` -
identifies the category of data

Exceptions:

`RemoteException` -
if remote access fails

addAEFPropertyChangeListener

```
public void addAEFPropertyChangeListener(java.lang.String category,
                                         java.lang.String property)
                                    throws java.rmi.RemoteException
```

Add a filtered property change listener to listen to data provider events for a specific property.

Parameters:

(continued from last page)

`category` -
identifies the category of data
`property` -
identifies the property data

Exceptions:
`RemoteException` -
if remote access fails

removeAEFPropertyChangeListener

```
public void removeAEFPropertyChangeListener(java.lang.String category,
                                            java.lang.String property)
                                     throws java.rmi.RemoteException
```

Remove a filtered property change listener.

Parameters:
`category` -
identifies the category of data
`property` -
identifies the property data

Exceptions:
`RemoteException` -
if remote access fails

com.ibm.retail.AEF.client

Class AEFPropertyListenerProxyImpl

```
java.lang.Object
  |-java.rmi.server.RemoteObject
    |-java.rmi.server.RemoteServer
      |-java.rmi.server.UnicastRemoteObject
        |-com.ibm.retail.AEF.util.AEFRemoteObject
          |-com.ibm.retail.AEF.client.AEFEventListenerProxy
            |-com.ibm.retail.AEF.client.AEFPropertyListenerProxyImpl
```

All Implemented interfaces:
AEFPropertyChangeListener, AEFPropertyListenerProxy, java.io.Serializable, java.rmi.Remote

---

public class **AEFPropertyListenerProxyImpl**
extends AEFEventListenerProxy
implements java.rmi.Remote, java.io.Serializable, AEFPropertyListenerProxy, AEFPropertyChangeListener

The AEFPropertyListenerProxy provides a proxy object for remote listeners to monitor the AEFPropertyChanges of its associated terminal session.

Remote proxy objects extend the appropriate RMI server classes which handle the RMI duties for the client (so the client does not need to deal with rmi server issues). The proxy objects also perform the role of event dispatching.

By default, the listener proxy object will perform event queuing on an AEF event thread. This relieves the client of any "thread swapping" responsibilities and insures that the AEF event dispatching performance is not affected by client processing. To override this default behavior, use the *setDispatchQueue()* method. The proxy utilizes the EventDispatcher for queueing event listener notification.

To use a proxy object, the client must implement the listener interface and get the POSDataProvider object from the AEFSession to monitor. The proxy object is then constructed passing a reference to the client and the data provider. The proxy object performs the listener registration and forwards all events from the data provider to the client.

See Also:
Sample Usage, com.ibm.retail.AEF.client.EventDispatcher EventDispatcher

---

Field Summary

com.ibm.retail.AEF.data.POSDataProvider	dataProvider
com.ibm.retail.AEF.event.AEFPropertyChangeListener	listener
static java.lang.reflect.Method	listenerMethod

Fields inherited from : class com.ibm.retail.AEF.client.AEFEventListenerProxy
dispatchQueueName, SOCKET_CONNECT_TIMEOUT, SOCKET_READ_TIMEOUT

Fields inherited from : class `com.ibm.retail.AEF.util.AEFRemoteObject`
`DEFAULT_CONNECT_TIMEOUT`, `DEFAULT_READ_TIMEOUT`

Fields inherited from : class `java.rmi.server.RemoteObject`
`ref`

Constructor Summary

`AEFPropertyListenerProxyImpl(POSDataProvider dataProvider,AEFPropertyChangeListener dataProvider)`

Construct an AEFPropertyListenerProxy.

Method Summary

`void`	`addAEFPropertyChangeListener()` Add a property change listener to listen to all data provider events.
`void`	`addAEFPropertyChangeListener(java.lang.String category)` Add a filtered property change listener to listen to data provider events for a specific category
`void`	`addAEFPropertyChangeListener(java.lang.String category,java.lang.String category)` Add a filtered property change listener to listen to data provider events for a specific property
`void`	`propertyChanged(AEFPropertyChangeEvent evt)` An AEF POSDataProvider property was updated.
`void`	`removeAEFPropertyChangeListener()` Remove a property change listener.
`void`	`removeAEFPropertyChangeListener(java.lang.String category)` Remove a filtered property change listener
`void`	`removeAEFPropertyChangeListener(java.lang.String category,java.lang.String category)` Remove a filtered property change listener.

Methods inherited from : class `com.ibm.retail.AEF.client.AEFEventListenerProxy`
`dispatchEvent, getDispatchQueueName, getListenerMethod, setDispatchQueueName`

Methods inherited from : class `com.ibm.retail.AEF.util.AEFRemoteObject`
`getClientSocketConnectTimeout, getClientSocketReadTimeout`

Methods inherited from : class `java.rmi.server.UnicastRemoteObject`

clone, exportObject, exportObject, exportObject, unexportObject

Methods inherited from : class `java.rmi.server.RemoteServer`
getClientHost, getLog, setLog

Methods inherited from : class `java.rmi.server.RemoteObject`
equals, getRef, hashCode, toString, toStub

Methods inherited from : class `java.lang.Object`
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

listenerMethod

```
protected static java.lang.reflect.Method listenerMethod
```

dataProvider

```
protected com.ibm.retail.AEF.data.POSDataProvider dataProvider
```

listener

```
protected com.ibm.retail.AEF.event.AEFPropertyChangeListener listener
```

Constructors

AEFPropertyListenerProxyImpl

```
public AEFPropertyListenerProxyImpl(POSDataProvider dataProvider,
                                    AEFPropertyChangeListener listener)
```

Construct an AEFPropertyListenerProxy. Use the *addPropertyChangeListener()*methods to register the client listener with the POSDataProvider.

Parameters:

`dataProvider -`
an instance of POSDataProvider is the source of event data
`listener -`
client listener for this proxy is the recepient of event data

Exceptions:

`java.rmi.RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

Methods

addAEFPropertyChangeListener

```
public void addAEFPropertyChangeListener()
                                    throws java.rmi.RemoteException
```

Add a property change listener to listen to all data provider events.

Exceptions:

`RemoteException` -
if remote access fails

removeAEFPropertyChangeListener

```
public void removeAEFPropertyChangeListener()
                                       throws java.rmi.RemoteException
```

Remove a property change listener.

Exceptions:

`RemoteException` -
if remote access fails

addAEFPropertyChangeListener

```
public void addAEFPropertyChangeListener(java.lang.String category)
                                    throws java.rmi.RemoteException
```

Add a filtered property change listener to listen to data provider events for a specific category

Parameters:

`category` -
identifies the category of data

Exceptions:

`RemoteException` -
if remote access fails

removeAEFPropertyChangeListener

```
public void removeAEFPropertyChangeListener(java.lang.String category)
                                       throws java.rmi.RemoteException
```

Remove a filtered property change listener

Parameters:

`category` -
identifies the category of data

Exceptions:

`RemoteException` -
if remote access fails

addAEFPropertyChangeListener

```
public void addAEFPropertyChangeListener(java.lang.String category,
                                         java.lang.String property)
                                    throws java.rmi.RemoteException
```

Add a filtered property change listener to listen to data provider events for a specific property

Parameters:

(continued from last page)

`category` -
identifies the category of data
`property` -
identifies the property data

Exceptions:
`RemoteException` -
if remote access fails

removeAEFPropertyChangeListener

```
public void removeAEFPropertyChangeListener(java.lang.String category,
                                            java.lang.String property)
                                     throws java.rmi.RemoteException
```

Remove a filtered property change listener.

Parameters:
`category` -
identifies the category of data
`property` -
identifies the property data

Exceptions:
`RemoteException` -
if remote access fails

propertyChanged

```
public void propertyChanged(AEFPropertyChangeEvent evt)
                     throws java.rmi.RemoteException
```

An AEF POSDataProvider property was updated.

Parameters:
`evt` -
contains details of the event

Exceptions:
`RemoteException` -
if remote access fails

com.ibm.retail.AEF.client

Class CashReceiptListenerProxy

```
java.lang.Object
  |-java.rmi.server.RemoteObject
    |-java.rmi.server.RemoteServer
      |-java.rmi.server.UnicastRemoteObject
        |-com.ibm.retail.AEF.util.AEFRemoteObject
          |-com.ibm.retail.AEF.client.AEFEventListenerProxy
            |-com.ibm.retail.AEF.client.CashReceiptListenerProxy
```

All Implemented interfaces:
CashReceiptListener, AEFListenerProxyInterface, java.io.Serializable, java.rmi.Remote

---

public class **CashReceiptListenerProxy**
extends AEFEventListenerProxy
implements java.rmi.Remote, java.io.Serializable, AEFListenerProxyInterface, CashReceiptListener

The CashReceiptListenerProxy provides a proxy object for remote listeners to monitor the CashReceipt events of its associated terminal session.

Remote proxy objects extend the appropriate RMI server classes which handle the RMI duties for the client (so the client does not need to deal with rmi server issues). The proxy objects also perform the role of event dispatching.

By default, the listener proxy object will perform event queuing on an AEF event thread. This relieves the client of any "thread swapping" responsibilities and insures that the AEF event dispatching performance is not affected by client processing. To override this default behavior, use the *setDispatchQueue()*method. The proxy utilizes the EventDispatcher for queueing event listener notification.

To use a proxy object, the client must implement the listener interface and get the POSDataProvider object from the AEFSession to monitor. The proxy object is then constructed passing a reference to the client and the data provider. The proxy object performs the listener registration and forwards all events from the data provider to the client.

See Also:
Sample Usage, com.ibm.retail.AEF.client.EventDispatcher EventDispatcher

---

Field Summary

com.ibm.retail.AEF.data.POSDataProvider	dataProvider
com.ibm.retail.AEF.event.CashReceiptListener	listener
static java.lang.reflect.Method	listenerMethod

Fields inherited from : class com.ibm.retail.AEF.client.AEFEventListenerProxy
dispatchQueueName, SOCKET_CONNECT_TIMEOUT, SOCKET_READ_TIMEOUT

Fields inherited from : class `com.ibm.retail.AEF.util.AEFRemoteObject`
`DEFAULT_CONNECT_TIMEOUT`, `DEFAULT_READ_TIMEOUT`

Fields inherited from : class `java.rmi.server.RemoteObject`
`ref`

Constructor Summary

`CashReceiptListenerProxy(POSDataProvider dataProvider,CashReceiptListener dataProvider)` Construct a CashReceiptListenerProxy.

Method Summary

`void`	`linePrinted(CashReceiptEvent evt)` A line was printed on the receipt.
`void`	`removeListener()` Remove the proxy listener from the POSDataProvider.

Methods inherited from : class `com.ibm.retail.AEF.client.AEFEventListenerProxy`
`dispatchEvent, getDispatchQueueName, getListenerMethod, setDispatchQueueName`

Methods inherited from : class `com.ibm.retail.AEF.util.AEFRemoteObject`
`getClientSocketConnectTimeout, getClientSocketReadTimeout`

Methods inherited from : class `java.rmi.server.UnicastRemoteObject`
`clone, exportObject, exportObject, exportObject, unexportObject`

Methods inherited from : class `java.rmi.server.RemoteServer`
`getClientHost, getLog, setLog`

Methods inherited from : class `java.rmi.server.RemoteObject`
`equals, getRef, hashCode, toString, toStub`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

(continued from last page)

listenerMethod

```
protected static java.lang.reflect.Method listenerMethod
```

dataProvider

```
protected com.ibm.retail.AEF.data.POSDataProvider dataProvider
```

listener

```
protected com.ibm.retail.AEF.event.CashReceiptListener listener
```

Constructors

CashReceiptListenerProxy

```
public CashReceiptListenerProxy(POSDataProvider dataProvider,
                                CashReceiptListener listener)
```

Construct a CashReceiptListenerProxy. This method registers the proxy as a listener with the POSDataProvider and begins event notification to the proxy client listener.

Parameters:

`dataProvider` -
instance of POSDataProvider which is the source of events
`listener` -
client listener which receives the event notification via this proxy

Exceptions:

`java.rmi.RemoteException`
`AEFException` -
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

Methods

removeListener

```
public void removeListener()
                    throws java.rmi.RemoteException,
                           AEFException
```

Remove the proxy listener from the POSDataProvider. Ends notification of events.

Exceptions:

`java.rmi.RemoteException`
`AEFException` -
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

linePrinted

```
public void linePrinted(CashReceiptEvent evt)
                  throws java.rmi.RemoteException
```

A line was printed on the receipt. Dispatches event notification to the proxy client listener.

(continued from last page)

Parameters:

`evt -`
contains details of the print line

Exceptions:

`RemoteException -`
if a listener can not be notified

com.ibm.retail.AEF.client

Class CouponListenerProxy

```
java.lang.Object
  |-java.rmi.server.RemoteObject
    |-java.rmi.server.RemoteServer
      |-java.rmi.server.UnicastRemoteObject
        |-com.ibm.retail.AEF.util.AEFRemoteObject
          |-com.ibm.retail.AEF.client.AEFEventListenerProxy
            |-com.ibm.retail.AEF.client.CouponListenerProxy
```

All Implemented interfaces:
CouponListener, AEFListenerProxyInterface, java.io.Serializable, java.rmi.Remote

---

public class **CouponListenerProxy**
extends AEFEventListenerProxy
implements java.rmi.Remote, java.io.Serializable, AEFListenerProxyInterface, CouponListener

The CouponListenerProxy provides a proxy object for remote listeners to monitor the Coupon events of its associated terminal session.

Remote proxy objects extend the appropriate RMI server classes which handle the RMI duties for the client (so the client does not need to deal with rmi server issues). The proxy objects also perform the role of event dispatching.

By default, the listener proxy object will perform event queuing on an AEF event thread. This relieves the client of any "thread swapping" responsibilities and insures that the AEF event dispatching performance is not affected by client processing. To override this default behavior, use the *setDispatchQueue()*method. The proxy utilizes the EventDispatcher for queueing event listener notification.

To use a proxy object, the client must implement the listener interface and get the POSDataProvider object from the AEFSession to monitor. The proxy object is then constructed passing a reference to the client and the data provider. The proxy object performs the listener registration and forwards all events from the data provider to the client.

See Also:
Sample Usage, com.ibm.retail.AEF.client.EventDispatcher EventDispatcher

---

Field Summary

com.ibm.retail.AEF.data.POSDataProvider	dataProvider
com.ibm.retail.AEF.event.CouponListener	listener
static java.lang.reflect.Method	listenerMethod

Fields inherited from : class com.ibm.retail.AEF.client.AEFEventListenerProxy

dispatchQueueName, SOCKET_CONNECT_TIMEOUT, SOCKET_READ_TIMEOUT

Fields inherited from : class com.ibm.retail.AEF.util.AEFRemoteObject

DEFAULT_CONNECT_TIMEOUT, DEFAULT_READ_TIMEOUT

Fields inherited from : class `java.rmi.server.RemoteObject`
ref

Constructor Summary

`CouponListenerProxy(POSDataProvider dataProvider,CouponListener dataProvider)` Construct a CouponListenerProxy.

Method Summary

void	`couponApplied(CouponEvent evt)` A coupon was applied to an item or transaction.
void	`removeListener()` Remove the proxy listener from the POSDataProvider.

Methods inherited from : class `com.ibm.retail.AEF.client.AEFEventListenerProxy`
dispatchEvent, getDispatchQueueName, getListenerMethod, setDispatchQueueName

Methods inherited from : class `com.ibm.retail.AEF.util.AEFRemoteObject`
getClientSocketConnectTimeout, getClientSocketReadTimeout

Methods inherited from : class `java.rmi.server.UnicastRemoteObject`
clone, exportObject, exportObject, exportObject, unexportObject

Methods inherited from : class `java.rmi.server.RemoteServer`
getClientHost, getLog, setLog

Methods inherited from : class `java.rmi.server.RemoteObject`
equals, getRef, hashCode, toString, toStub

Methods inherited from : class `java.lang.Object`
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

(continued from last page)

listenerMethod

```
protected static java.lang.reflect.Method listenerMethod
```

dataProvider

```
protected com.ibm.retail.AEF.data.POSDataProvider dataProvider
```

listener

```
protected com.ibm.retail.AEF.event.CouponListener listener
```

Constructors

CouponListenerProxy

```
public CouponListenerProxy(POSDataProvider dataProvider,
                           CouponListener listener)
```

Construct a CouponListenerProxy. This method registers the proxy as a listener with the POSDataProvider.

Parameters:

`dataProvider` -
instance of POSDataProvider which is the source of events
`listener` -
client listener which receives the event notification via this proxy

Exceptions:

`java.rmi.RemoteException`
`AEFException` -
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

Methods

removeListener

```
public void removeListener()
                    throws java.rmi.RemoteException,
                           AEFException
```

Remove the proxy listener from the POSDataProvider. Ends notification of events.

Exceptions:

`java.rmi.RemoteException`
`AEFException` -
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

couponApplied

```
public void couponApplied(CouponEvent evt)
                   throws java.rmi.RemoteException
```

A coupon was applied to an item or transaction. Dispatches event notification to the proxy client listener.

(continued from last page)

Parameters:

`evt -`
contains coupon details

Exceptions:

`RemoteException -`
if remote access fails

com.ibm.retail.AEF.client

Class CustomerListenerProxy

```
java.lang.Object
  |-java.rmi.server.RemoteObject
    |-java.rmi.server.RemoteServer
      |-java.rmi.server.UnicastRemoteObject
        |-com.ibm.retail.AEF.util.AEFRemoteObject
          |-com.ibm.retail.AEF.client.AEFEventListenerProxy
            |-com.ibm.retail.AEF.client.CustomerListenerProxy
```

All Implemented interfaces:
CustomerListener, AEFListenerProxyInterface, java.io.Serializable, java.rmi.Remote

---

public class **CustomerListenerProxy**
extends AEFEventListenerProxy
implements java.rmi.Remote, java.io.Serializable, AEFListenerProxyInterface, CustomerListener

The CustomerListenerProxy provides a proxy object for remote listeners to monitor the Customer events of its associated terminal session.

Remote proxy objects extend the appropriate RMI server classes which handle the RMI duties for the client (so the client does not need to deal with rmi server issues). The proxy objects also perform the role of event dispatching.

By default, the listener proxy object will perform event queuing on an AEF event thread. This relieves the client of any "thread swapping" responsibilities and insures that the AEF event dispatching performance is not affected by client processing. To override this default behavior, use the *setDispatchQueue()*method. The proxy utilizes the EventDispatcher for queueing event listener notification.

To use a proxy object, the client must implement the listener interface and get the POSDataProvider object from the AEFSession to monitor. The proxy object is then constructed passing a reference to the client and the data provider. The proxy object performs the listener registration and forwards all events from the data provider to the client.

See Also:
Sample Usage, com.ibm.retail.AEF.client.EventDispatcher EventDispatcher

---

Field Summary

com.ibm.retail.AEF.data.POSDataProvider	dataProvider
com.ibm.retail.AEF.event.CustomerListener	listener
static java.lang.reflect.Method	listenerMethod

Fields inherited from : class com.ibm.retail.AEF.client.AEFEventListenerProxy
dispatchQueueName, SOCKET_CONNECT_TIMEOUT, SOCKET_READ_TIMEOUT

Fields inherited from : class com.ibm.retail.AEF.util.AEFRemoteObject

DEFAULT_CONNECT_TIMEOUT, DEFAULT_READ_TIMEOUT

Fields inherited from : class `java.rmi.server.RemoteObject`
ref

Constructor Summary

`CustomerListenerProxy(POSDataProvider dataProvider,CustomerListener dataProvider)` Construct a CustomerListenerProxy.

Method Summary

void	`customerCardEntered(CustomerEvent evt)` A customer card number was entered by scanning or swiping the card.
void	`removeListener()` Remove the proxy listener from the POSDataProvider.

Methods inherited from : class `com.ibm.retail.AEF.client.AEFEventListenerProxy`
dispatchEvent, getDispatchQueueName, getListenerMethod, setDispatchQueueName

Methods inherited from : class `com.ibm.retail.AEF.util.AEFRemoteObject`
getClientSocketConnectTimeout, getClientSocketReadTimeout

Methods inherited from : class `java.rmi.server.UnicastRemoteObject`
clone, exportObject, exportObject, exportObject, unexportObject

Methods inherited from : class `java.rmi.server.RemoteServer`
getClientHost, getLog, setLog

Methods inherited from : class `java.rmi.server.RemoteObject`
equals, getRef, hashCode, toString, toStub

Methods inherited from : class `java.lang.Object`
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

(continued from last page)

listenerMethod

```
protected static java.lang.reflect.Method listenerMethod
```

dataProvider

```
protected com.ibm.retail.AEF.data.POSDataProvider dataProvider
```

listener

```
protected com.ibm.retail.AEF.event.CustomerListener listener
```

Constructors

CustomerListenerProxy

```
public CustomerListenerProxy(POSDataProvider dataProvider,
                             CustomerListener listener)
```

Construct a CustomerListenerProxy. This method registers the proxy as a listener with the POSDataProvider and begins event notification to the proxy client listener.

Parameters:

`dataProvider` -
instance of POSDataProvider which is the source of events
`listener` -
client listener which receives the event notification via this proxy

Exceptions:

`java.rmi.RemoteException`
`AEFException` -
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

Methods

removeListener

```
public void removeListener()
                    throws java.rmi.RemoteException,
                           AEFException
```

Remove the proxy listener from the POSDataProvider. Ends notification of events.

Exceptions:

`java.rmi.RemoteException`
`AEFException` -
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

customerCardEntered

```
public void customerCardEntered(CustomerEvent evt)
                        throws java.rmi.RemoteException
```

A customer card number was entered by scanning or swiping the card. Dispatches event notification to the proxy client listener.

(continued from last page)

Parameters:

`evt` -
contains customer details

com.ibm.retail.AEF.client

Class DiscountListenerProxy

```
java.lang.Object
  |-java.rmi.server.RemoteObject
    |-java.rmi.server.RemoteServer
      |-java.rmi.server.UnicastRemoteObject
        |-com.ibm.retail.AEF.util.AEFRemoteObject
          |-com.ibm.retail.AEF.client.AEFEventListenerProxy
            |-com.ibm.retail.AEF.client.DiscountListenerProxy
```

All Implemented interfaces:
DiscountListener, AEFListenerProxyInterface, java.io.Serializable, java.rmi.Remote

---

public class **DiscountListenerProxy**
extends AEFEventListenerProxy
implements java.rmi.Remote, java.io.Serializable, AEFListenerProxyInterface, DiscountListener

The DiscountListenerProxy provides a proxy object for remote listeners to monitor the Discount events of its associated terminal session.

Remote proxy objects extend the appropriate RMI server classes which handle the RMI duties for the client (so the client does not need to deal with rmi server issues). The proxy objects also perform the role of event dispatching.

By default, the listener proxy object will perform event queuing on an AEF event thread. This relieves the client of any "thread swapping" responsibilities and insures that the AEF event dispatching performance is not affected by client processing. To override this default behavior, use the *setDispatchQueue()*method. The proxy utilizes the EventDispatcher for queueing event listener notification.

To use a proxy object, the client must implement the listener interface and get the POSDataProvider object from the AEFSession to monitor. The proxy object is then constructed passing a reference to the client and the data provider. The proxy object performs the listener registration and forwards all events from the data provider to the client.

See Also:
Sample Usage, com.ibm.retail.AEF.client.EventDispatcher EventDispatcher

---

Field Summary

com.ibm.retail.AEF.data.POSDataProvider	dataProvider
com.ibm.retail.AEF.event.DiscountListener	listener
static java.lang.reflect.Method	listenerMethod

Fields inherited from : class com.ibm.retail.AEF.client.AEFEventListenerProxy
dispatchQueueName, SOCKET_CONNECT_TIMEOUT, SOCKET_READ_TIMEOUT

Fields inherited from : class com.ibm.retail.AEF.util.AEFRemoteObject

DEFAULT_CONNECT_TIMEOUT, DEFAULT_READ_TIMEOUT

Fields inherited from : class `java.rmi.server.RemoteObject`
ref

Constructor Summary

`DiscountListenerProxy(POSDataProvider dataProvider,DiscountListener dataProvider)` Construct a DiscountListenerProxy.

Method Summary

void	`discountApplied(DiscountEvent evt)` A discount was applied to an item or transaction.
void	`removeListener()` Remove the proxy listener from the POSDataProvider.

Methods inherited from : class `com.ibm.retail.AEF.client.AEFEventListenerProxy`
dispatchEvent, getDispatchQueueName, getListenerMethod, setDispatchQueueName

Methods inherited from : class `com.ibm.retail.AEF.util.AEFRemoteObject`
getClientSocketConnectTimeout, getClientSocketReadTimeout

Methods inherited from : class `java.rmi.server.UnicastRemoteObject`
clone, exportObject, exportObject, exportObject, unexportObject

Methods inherited from : class `java.rmi.server.RemoteServer`
getClientHost, getLog, setLog

Methods inherited from : class `java.rmi.server.RemoteObject`
equals, getRef, hashCode, toString, toStub

Methods inherited from : class `java.lang.Object`
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

(continued from last page)

listenerMethod

```
protected static java.lang.reflect.Method listenerMethod
```

dataProvider

```
protected com.ibm.retail.AEF.data.POSDataProvider dataProvider
```

listener

```
protected com.ibm.retail.AEF.event.DiscountListener listener
```

Constructors

DiscountListenerProxy

```
public DiscountListenerProxy(POSDataProvider dataProvider,
                             DiscountListener listener)
```

Construct a DiscountListenerProxy. This method registers the proxy as a listener with the POSDataProvider and begins event notification to the proxy client listener.

Parameters:

`dataProvider` -
instance of POSDataProvider which is the source of events
`listener` -
client listener which receives the event notification via this proxy

Exceptions:

`java.rmi.RemoteException`
`AEFException` -
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

Methods

removeListener

```
public void removeListener()
                    throws java.rmi.RemoteException,
                           AEFException
```

Remove the proxy listener from the POSDataProvider. Ends notification of events.

Exceptions:

`java.rmi.RemoteException`
`AEFException` -
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

discountApplied

```
public void discountApplied(DiscountEvent evt)
                     throws java.rmi.RemoteException
```

A discount was applied to an item or transaction. Dispatches event notification to the proxy client listener.

(continued from last page)

Parameters:

`evt -`
contains details of the discount

Exceptions:

`RemoteException -`
if a listener can not be notified

com.ibm.retail.AEF.client

Class EventDispatcher

```
java.lang.Object
   |
   +-com.ibm.retail.AEF.client.EventDispatcher
```

public class **EventDispatcher**
extends java.lang.Object

The EventDispatcher provides an event dispatch mechanism for AEF clients. It is used primarily to perform a thread context swap to prevent client listeners from hanging the AEF remote event calls.

EventDispatcher is a singleton object that is accessed through the static *dispatch()* method.

Field Summary

static java.lang.String	AEF_QUEUE Dispatch event queue name identifier: AEF event queue (for non-GUI apps).
static java.lang.String	AWT_QUEUE Dispatch event queue name identifier: AWT event queue (for GUI apps).
static java.lang.String	DEFAULT_QUEUE Default dispatch event queue name identifier.
static com.ibm.retail.AEF.client.EventDispatcher	instance
static java.lang.String	NO_QUEUE Dispatch event queue name identifier: none.
com.ibm.retail.AEF.client.EventDispatcher.EventDispatchQueue	queue

Constructor Summary

EventDispatcher() Construct an EventDispatcher.

Method Summary

void	createEventQueues() Create the event queues.
static void	dispatch(java.lang.Runnable obj,java.lang.String obj) Handle a dispatch request.
EventDispatcher.EventDispatchQueue	getQueue() Get an event queue.

static void	setDefaultDispatchQueueName(java.lang.String queue) Set the dispatch queue name.

Methods inherited from : class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

instance

protected static com.ibm.retail.AEF.client.EventDispatcher **instance**

AWT_QUEUE

public static java.lang.String **AWT_QUEUE**

Dispatch event queue name identifier: AWT event queue (for GUI apps).

AEF_QUEUE

public static java.lang.String **AEF_QUEUE**

Dispatch event queue name identifier: AEF event queue (for non-GUI apps).

NO_QUEUE

public static java.lang.String **NO_QUEUE**

Dispatch event queue name identifier: none. (Performs the event dispatching synchronously.)

DEFAULT_QUEUE

public static java.lang.String **DEFAULT_QUEUE**

Default dispatch event queue name identifier.

queue

protected com.ibm.retail.AEF.client.EventDispatcher.EventDispatchQueue **queue**

Constructors

EventDispatcher

protected **EventDispatcher**()

Construct an EventDispatcher.

Methods

(continued from last page)

setDefaultDispatchQueueName

```
public static void setDefaultDispatchQueueName(java.lang.String queue)
```

Set the dispatch queue name. Defaults to the AEF_QUEUE.

Parameters:

`queue -`
the dispatch queue name identifier

See Also:

com.ibm.retail.AEF.client.EventDispatcher#AEF_QUEUE AEF_QUEUE

com.ibm.retail.AEF.client.EventDispatcher#AWT_QUEUE AWT_QUEUE

com.ibm.retail.AEF.client.EventDispatcher#NO_QUEUE NO_QUEUE

---

dispatch

```
public static void dispatch(java.lang.Runnable obj,
                            java.lang.String queueName)
                     throws AEFException
```

Handle a dispatch request.

Parameters:

`obj -`
dispatch request object
`queueName -`
dispatch queue name

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.INVALID_DISPATCH_QUEUE

See Also:

#AWT_QUEUE AWT_QUEUE queue name

#AEF_QUEUE AEF_QUEUE queue name

#NO_QUEUE NO_QUEUE no queueing

---

createEventQueues

```
protected void createEventQueues()
```

Create the event queues.

---

getQueue

```
protected EventDispatcher.EventDispatchQueue getQueue()
```

Get an event queue.

Returns:

EventDispatchQueue

com.ibm.retail.AEF.client

Class GenericListenerProxy

```
java.lang.Object
  |-java.rmi.server.RemoteObject
    |-java.rmi.server.RemoteServer
      |-java.rmi.server.UnicastRemoteObject
        |-com.ibm.retail.AEF.util.AEFRemoteObject
          |-com.ibm.retail.AEF.client.AEFEventListenerProxy
            |-com.ibm.retail.AEF.client.GenericListenerProxy
```

All Implemented interfaces:
GenericEventListener, AEFListenerProxyInterface, java.io.Serializable, java.rmi.Remote

---

public class **GenericListenerProxy**
extends AEFEventListenerProxy
implements java.rmi.Remote, java.io.Serializable, AEFListenerProxyInterface, GenericEventListener

GenericListenerProxy provides a proxy object for remote listeners to monitor the Generic (POSAppEvent) events of its associated terminal session.

Remote proxy objects extend the appropriate RMI server classes which handle the RMI duties for the client (so the client does not need to deal with rmi server issues). The proxy objects also perform the role of event dispatching.

By default, the listener proxy object will perform event queuing on an AEF event thread. This relieves the client of any "thread swapping" responsibilities and insures that the AEF event dispatching performance is not affected by client processing. To override this default behavior, use the *setDispatchQueue()*method. The proxy utilizes the EventDispatcher for queueing event listener notification.

To use a proxy object, the client must implement the listener interface and get the POSDataProvider object from the AEFSession to monitor. The proxy object is then constructed passing a reference to the client and the data provider. The proxy object performs the listener registration and forwards all events from the data provider to the client.

See Also:
Sample Usage, com.ibm.retail.AEF.client.EventDispatcher EventDispatcher

---

Field Summary

com.ibm.retail.AEF.data.POSDataProvider	dataProvider
com.ibm.retail.AEF.event.GenericEventListener	listener
static java.lang.reflect.Method	listenerMethod

Fields inherited from : class com.ibm.retail.AEF.client.AEFEventListenerProxy
dispatchQueueName, SOCKET_CONNECT_TIMEOUT, SOCKET_READ_TIMEOUT

Fields inherited from : class `com.ibm.retail.AEF.util.AEFRemoteObject`
`DEFAULT_CONNECT_TIMEOUT`, `DEFAULT_READ_TIMEOUT`

Fields inherited from : class `java.rmi.server.RemoteObject`
`ref`

Constructor Summary

`GenericListenerProxy(POSDataProvider dataProvider,GenericEventListener dataProvider)` Construct a GenericListenerProxy.

Method Summary

`void`	`eventOccurred(POSAppEvent evt)` A POSAppEvent occurred.
`void`	`removeListener()` Remove the proxy listener from the POSDataProvider.

Methods inherited from : class `com.ibm.retail.AEF.client.AEFEventListenerProxy`
`dispatchEvent, getDispatchQueueName, getListenerMethod, setDispatchQueueName`

Methods inherited from : class `com.ibm.retail.AEF.util.AEFRemoteObject`
`getClientSocketConnectTimeout, getClientSocketReadTimeout`

Methods inherited from : class `java.rmi.server.UnicastRemoteObject`
`clone, exportObject, exportObject, exportObject, unexportObject`

Methods inherited from : class `java.rmi.server.RemoteServer`
`getClientHost, getLog, setLog`

Methods inherited from : class `java.rmi.server.RemoteObject`
`equals, getRef, hashCode, toString, toStub`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

(continued from last page)

listenerMethod

```
protected static java.lang.reflect.Method listenerMethod
```

dataProvider

```
protected com.ibm.retail.AEF.data.POSDataProvider dataProvider
```

listener

```
protected com.ibm.retail.AEF.event.GenericEventListener listener
```

Constructors

GenericListenerProxy

```
public GenericListenerProxy(POSDataProvider dataProvider,
                            GenericEventListener listener)
```

Construct a GenericListenerProxy. This method registers the proxy as a listener with the POSDataProvider and begins event notification to the proxy client listener.

Parameters:

`dataProvider` -
instance of POSDataProvider which is the source of events
`listener` -
client listener which receives the event notification via this proxy

Exceptions:

`java.rmi.RemoteException`
`AEFException` -
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

Methods

removeListener

```
public void removeListener()
                    throws java.rmi.RemoteException,
                           AEFException
```

Remove the proxy listener from the POSDataProvider. Ends notification of events.

Exceptions:

`java.rmi.RemoteException`
`AEFException` -
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

eventOccurred

```
public void eventOccurred(POSAppEvent evt)
                   throws java.rmi.RemoteException
```

A POSAppEvent occurred. Dispatches event notification to the proxy client listener.

(continued from last page)

Parameters:

`evt -`
contains details of the event

Exceptions:

`RemoteException -`
if a listener can not be notified

com.ibm.retail.AEF.client

Class ItemSalesListenerProxy

```
java.lang.Object
  |-java.rmi.server.RemoteObject
    |-java.rmi.server.RemoteServer
      |-java.rmi.server.UnicastRemoteObject
        |-com.ibm.retail.AEF.util.AEFRemoteObject
          |-com.ibm.retail.AEF.client.AEFEventListenerProxy
            |-com.ibm.retail.AEF.client.ItemSalesListenerProxy
```

All Implemented interfaces:
ItemSalesListener, AEFListenerProxyInterface, java.io.Serializable, java.rmi.Remote

---

public class **ItemSalesListenerProxy**
extends AEFEventListenerProxy
implements java.rmi.Remote, java.io.Serializable, AEFListenerProxyInterface, ItemSalesListener

ItemSalesListenerProxy provides a proxy object for remote listeners to monitor the Item events of its associated terminal session.

Remote proxy objects extend the appropriate RMI server classes which handle the RMI duties for the client (so the client does not need to deal with rmi server issues). The proxy objects also perform the role of event dispatching.

By default, the listener proxy object will perform event queuing on an AEF event thread. This relieves the client of any "thread swapping" responsibilities and insures that the AEF event dispatching performance is not affected by client processing. To override this default behavior, use the *setDispatchQueue()*method. The proxy utilizes the EventDispatcher for queueing event listener notification.

To use a proxy object, the client must implement the listener interface and get the POSDataProvider object from the AEFSession to monitor. The proxy object is then constructed passing a reference to the client and the data provider. The proxy object performs the listener registration and forwards all events from the data provider to the client.

See Also:
Sample Usage, com.ibm.retail.AEF.client.EventDispatcher EventDispatcher

---

Field Summary

com.ibm.retail.AEF.data.POSDataProvider	dataProvider
static java.lang.reflect.Method	itemAddedMethod
static java.lang.reflect.Method	itemDepositAddedMethod
static java.lang.reflect.Method	itemDepositRemovedMethod
static java.lang.reflect.Method	itemRefundMethod

static java.lang.reflect.Method	itemRemovedMethod
com.ibm.retail.AEF.event.ItemSalesListener	listener

Fields inherited from : class com.ibm.retail.AEF.client.AEFEventListenerProxy
dispatchQueueName, SOCKET_CONNECT_TIMEOUT, SOCKET_READ_TIMEOUT

Fields inherited from : class com.ibm.retail.AEF.util.AEFRemoteObject
DEFAULT_CONNECT_TIMEOUT, DEFAULT_READ_TIMEOUT

Fields inherited from : class java.rmi.server.RemoteObject
ref

Constructor Summary

ItemSalesListenerProxy(POSDataProvider dataProvider,ItemSalesListener dataProvider) Construct a ItemSalesListenerProxy.

Method Summary

void	itemAdded(ItemSalesEvent evt) An item was added to the transaction.
void	itemDepositAdded(ItemSalesEvent evt) A deposit was added to the transaction.
void	itemDepositRemoved(ItemSalesEvent evt) An item deposit was removed from the transaction.
void	itemRefund(ItemSalesEvent evt) An item refund was performed in this transaction.
void	itemRemoved(ItemSalesEvent evt) An item was removed from the transaction.
void	removeListener() Remove the proxy listener from the POSDataProvider.

Methods inherited from : class com.ibm.retail.AEF.client.AEFEventListenerProxy
dispatchEvent, getDispatchQueueName, getListenerMethod, setDispatchQueueName

Methods inherited from : class com.ibm.retail.AEF.util.AEFRemoteObject
getClientSocketConnectTimeout, getClientSocketReadTimeout

Methods inherited from : class `java.rmi.server.UnicastRemoteObject`
`clone, exportObject, exportObject, exportObject, unexportObject`

Methods inherited from : class `java.rmi.server.RemoteServer`
`getClientHost, getLog, setLog`

Methods inherited from : class `java.rmi.server.RemoteObject`
`equals, getRef, hashCode, toString, toStub`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

itemAddedMethod

`protected static java.lang.reflect.Method` **`itemAddedMethod`**

itemRemovedMethod

`protected static java.lang.reflect.Method` **`itemRemovedMethod`**

itemDepositAddedMethod

`protected static java.lang.reflect.Method` **`itemDepositAddedMethod`**

itemDepositRemovedMethod

`protected static java.lang.reflect.Method` **`itemDepositRemovedMethod`**

itemRefundMethod

`protected static java.lang.reflect.Method` **`itemRefundMethod`**

dataProvider

`protected com.ibm.retail.AEF.data.POSDataProvider` **`dataProvider`**

(continued from last page)

listener

```
protected com.ibm.retail.AEF.event.ItemSalesListener listener
```

Constructors

ItemSalesListenerProxy

```
public ItemSalesListenerProxy(POSDataProvider dataProvider,
                              ItemSalesListener listener)
```

Construct a ItemSalesListenerProxy. This method registers the proxy as a listener with the POSDataProvider and begins event notification to the proxy client listener.

Parameters:

`dataProvider` -
instance of POSDataProvider which is the source of events
`listener` -
client listener which receives the event notification via this proxy

Exceptions:

`java.rmi.RemoteException`
`AEFException` -
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

Methods

removeListener

```
public void removeListener()
                    throws java.rmi.RemoteException,
                           AEFException
```

Remove the proxy listener from the POSDataProvider. Ends notification of events.

Exceptions:

`java.rmi.RemoteException`
`AEFException` -
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

---

itemAdded

```
public void itemAdded(ItemSalesEvent evt)
                throws java.rmi.RemoteException
```

An item was added to the transaction.

Parameters:

`evt` -
contains details of the item event

Exceptions:

`RemoteException` -
if a listener can not be notified

---

itemRemoved

```
public void itemRemoved(ItemSalesEvent evt)
                  throws java.rmi.RemoteException
```

(continued from last page)

An item was removed from the transaction.

Parameters:

`evt -`
contains details of the item event

Exceptions:

`RemoteException -`
if a listener can not be notified

itemDepositAdded

```
public void itemDepositAdded(ItemSalesEvent evt)
                     throws java.rmi.RemoteException
```

A deposit was added to the transaction.

Parameters:

`evt -`
contains details of the item event

Exceptions:

`RemoteException -`
if a listener can not be notified

itemDepositRemoved

```
public void itemDepositRemoved(ItemSalesEvent evt)
                       throws java.rmi.RemoteException
```

An item deposit was removed from the transaction.

Parameters:

`evt -`
contains details of the item event

Exceptions:

`RemoteException -`
if a listener can not be notified

itemRefund

```
public void itemRefund(ItemSalesEvent evt)
                 throws java.rmi.RemoteException
```

An item refund was performed in this transaction.

Parameters:

`evt -`
contains details of the item event

Exceptions:

`RemoteException -`
if a listener can not be notified

com.ibm.retail.AEF.client

Class KeyConsumerProxy

```
java.lang.Object
  +-java.rmi.server.RemoteObject
    +-java.rmi.server.RemoteServer
      +-java.rmi.server.UnicastRemoteObject
        +-com.ibm.retail.AEF.util.AEFRemoteObject
          +-com.ibm.retail.AEF.client.KeyConsumerProxy
```

All Implemented interfaces:
KeyConsumer, KeyConsumerProxyInterface, java.io.Serializable, java.rmi.Remote

---

public class **KeyConsumerProxy**
extends AEFRemoteObject
implements java.rmi.Remote, java.io.Serializable, KeyConsumerProxyInterface, KeyConsumer

KeyConsumerProxy provides a proxy object for remote listeners/consumers of POS keyboard events.

Remote proxy objects extend the appropriate RMI server classes which handle the RMI duties for the client (so the client does not need to deal with rmi server issues).

To use this proxy object, the client must implement the KeyConsumer interface and aqquire the Workstation object from the AEFSession to monitor. The proxy object is then constructed passing a reference to the client and the workstation. The proxy object provides the listener registration interface for the client and forwards all events from the workstation to the client.

NOTE:*Since the KeyConsumer allows the client to consume events, this proxy does not use the event dispatching (thread swapping). All calls block between the AEF and the KeyConsumer client.*

Field Summary

com.ibm.retail.AEF.workstation.KeyConsumer	consumer
static int	SOCKET_CONNECT_TIMEOUT
static int	SOCKET_READ_TIMEOUT
com.ibm.retail.AEF.workstation.Workstation	workstation

Fields inherited from : class com.ibm.retail.AEF.util.AEFRemoteObject
DEFAULT_CONNECT_TIMEOUT, DEFAULT_READ_TIMEOUT

Fields inherited from : class java.rmi.server.RemoteObject
ref

Constructor Summary

`KeyConsumerProxy(Workstation workstation,KeyConsumer workstation)` Construct a KeyConsumerProxy.

Method Summary

boolean	`keyPress(KeyCode key)` A function or data key has been pressed.
void	`removeConsumer()` Remove the proxy listener from the Workstation.

Methods inherited from : class `com.ibm.retail.AEF.util.AEFRemoteObject`
`getClientSocketConnectTimeout, getClientSocketReadTimeout`

Methods inherited from : class `java.rmi.server.UnicastRemoteObject`
`clone, exportObject, exportObject, exportObject, unexportObject`

Methods inherited from : class `java.rmi.server.RemoteServer`
`getClientHost, getLog, setLog`

Methods inherited from : class `java.rmi.server.RemoteObject`
`equals, getRef, hashCode, toString, toStub`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

SOCKET_READ_TIMEOUT

`public static int` **`SOCKET_READ_TIMEOUT`**

SOCKET_CONNECT_TIMEOUT

`public static int` **`SOCKET_CONNECT_TIMEOUT`**

workstation

`protected com.ibm.retail.AEF.workstation.Workstation` **`workstation`**

consumer

```
protected com.ibm.retail.AEF.workstation.KeyConsumer consumer
```

Constructors

KeyConsumerProxy

```
public KeyConsumerProxy(Workstation workstation,
                        KeyConsumer consumer)
```

Construct a KeyConsumerProxy. This method registers the proxy as a listener with the Workstation and begins event notification to the proxy client listener.

Parameters:

`workstation -`
instance of Workstation which is the source of key events
`listener -`
client listener which receives the event notification via this proxy

Exceptions:

`java.rmi.RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

Methods

removeConsumer

```
public void removeConsumer()
                    throws java.rmi.RemoteException,
                           AEFException
```

Remove the proxy listener from the Workstation. Ends notification of events.

Exceptions:

`java.rmi.RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

keyPress

```
public boolean keyPress(KeyCode key)
                  throws java.rmi.RemoteException
```

A function or data key has been pressed.

Parameters:

`key -`
the KeyCode object contains information about the key press

com.ibm.retail.AEF.client

Interface KeyConsumerProxyInterface

All Known Implementing Classes:
KeyConsumerProxy

---

public interface **KeyConsumerProxyInterface**
extends java.rmi.Remote

The KeyConsumerProxyInterface API provides a proxy object for remote listeners/consumers of POS keyboard events.

Remote proxy objects extend the appropriate RMI server classes which handle the RMI duties for the client (so the client does not need to deal with rmi server issues).

To use this proxy object, the client must implement the KeyConsumer interface and aqquire the Workstation object from the AEFSession to monitor. The proxy object is then constructed passing a reference to the client and the workstation. The proxy object provides the listener registration interface for the client and forwards all events from the workstation to the client.

Method Summary

boolean	`keyPress(KeyCode key)` A function or data key has been pressed.
void	`removeConsumer()` Removes this proxy as a KeyConsumer of the Workstation.

Methods

removeConsumer

```
public void removeConsumer()
                    throws java.rmi.RemoteException,
                           AEFException
```

Removes this proxy as a KeyConsumer of the Workstation.

Exceptions:
`RemoteException` -
if remote access fails

---

keyPress

```
public boolean keyPress(KeyCode key)
                  throws java.rmi.RemoteException
```

A function or data key has been pressed.

Parameters:
`key` -
KeyCode object contains information about the key press

com.ibm.retail.AEF.client

Class OperatorListenerProxy

```
java.lang.Object
  |-java.rmi.server.RemoteObject
    |-java.rmi.server.RemoteServer
      |-java.rmi.server.UnicastRemoteObject
        |-com.ibm.retail.AEF.util.AEFRemoteObject
          |-com.ibm.retail.AEF.client.AEFEventListenerProxy
            |-com.ibm.retail.AEF.client.OperatorListenerProxy
```

All Implemented interfaces:
OperatorListener, AEFListenerProxyInterface, java.io.Serializable, java.rmi.Remote

---

public class **OperatorListenerProxy**
extends AEFEventListenerProxy
implements java.rmi.Remote, java.io.Serializable, AEFListenerProxyInterface, OperatorListener

The OperatorListenerProxy provides a proxy object for remote listeners to monitor the Operator events of its associated terminal session.

Remote proxy objects extend the appropriate RMI server classes which handle the RMI duties for the client (so the client does not need to deal with rmi server issues). The proxy objects also perform the role of event dispatching.

By default, the listener proxy object will perform event queuing on an AEF event thread. This relieves the client of any "thread swapping" responsibilities and insures that the AEF event dispatching performance is not affected by client processing. To override this default behavior, use the *setDispatchQueue()*method. The proxy utilizes the EventDispatcher for queueing event listener notification.

To use a proxy object, the client must implement the listener interface and get the POSDataProvider object from the AEFSession to monitor. The proxy object is then constructed passing a reference to the client and the data provider. The proxy object performs the listener registration and forwards all events from the data provider to the client.

See Also:
Sample Usage, com.ibm.retail.AEF.client.EventDispatcher EventDispatcher

---

Field Summary

com.ibm.retail.AEF.data.POSDataProvider	dataProvider
com.ibm.retail.AEF.event.OperatorListener	listener
static java.lang.reflect.Method	listenerMethod

Fields inherited from : class com.ibm.retail.AEF.client.AEFEventListenerProxy

dispatchQueueName, SOCKET_CONNECT_TIMEOUT, SOCKET_READ_TIMEOUT

Fields inherited from : class com.ibm.retail.AEF.util.AEFRemoteObject

DEFAULT_CONNECT_TIMEOUT, DEFAULT_READ_TIMEOUT

Fields inherited from : class `java.rmi.server.RemoteObject`
ref

Constructor Summary

`OperatorListenerProxy(POSDataProvider dataProvider,OperatorListener dataProvider)` Construct an OperatorListenerProxy.

Method Summary

void	`operatorEventOccurred(OperatorEvent evt)` An operator action has been performed on the POS terminal session.
void	`removeListener()` Remove the proxy listener from the POSDataProvider.

Methods inherited from : class `com.ibm.retail.AEF.client.AEFEventListenerProxy`
dispatchEvent, getDispatchQueueName, getListenerMethod, setDispatchQueueName

Methods inherited from : class `com.ibm.retail.AEF.util.AEFRemoteObject`
getClientSocketConnectTimeout, getClientSocketReadTimeout

Methods inherited from : class `java.rmi.server.UnicastRemoteObject`
clone, exportObject, exportObject, exportObject, unexportObject

Methods inherited from : class `java.rmi.server.RemoteServer`
getClientHost, getLog, setLog

Methods inherited from : class `java.rmi.server.RemoteObject`
equals, getRef, hashCode, toString, toStub

Methods inherited from : class `java.lang.Object`
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

(continued from last page)

listenerMethod

```
protected static java.lang.reflect.Method listenerMethod
```

dataProvider

```
protected com.ibm.retail.AEF.data.POSDataProvider dataProvider
```

listener

```
protected com.ibm.retail.AEF.event.OperatorListener listener
```

Constructors

OperatorListenerProxy

```
public OperatorListenerProxy(POSDataProvider dataProvider,
                             OperatorListener listener)
```

Construct an OperatorListenerProxy. This method registers the proxy as a listener with the POSDataProvider and begins event notification to the proxy client listener.

Parameters:

`dataProvider` -
instance of POSDataProvider which is the source of events
`listener` -
client listener which receives the event notification via this proxy

Exceptions:

`java.rmi.RemoteException`
`AEFException` -
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

Methods

removeListener

```
public void removeListener()
                    throws java.rmi.RemoteException,
                           AEFException
```

Remove the proxy listener from the POSDataProvider. Ends notification of events.

Exceptions:

`java.rmi.RemoteException`
`AEFException` -
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

operatorEventOccurred

```
public void operatorEventOccurred(OperatorEvent evt)
                           throws java.rmi.RemoteException
```

An operator action has been performed on the POS terminal session. Dispatches event notification to the proxy client listener.

(continued from last page)

Parameters:

`evt` -
contains operator details

Exceptions:

`RemoteException` -
if remote access fails

com.ibm.retail.AEF.client

Class OptionsListenerProxy

```
java.lang.Object
  |-java.rmi.server.RemoteObject
    |-java.rmi.server.RemoteServer
      |-java.rmi.server.UnicastRemoteObject
        |-com.ibm.retail.AEF.util.AEFRemoteObject
          |-com.ibm.retail.AEF.client.AEFEventListenerProxy
            |-com.ibm.retail.AEF.client.OptionsListenerProxy
```

All Implemented interfaces:
OptionsListener, AEFListenerProxyInterface, java.io.Serializable, java.rmi.Remote

---

public class **OptionsListenerProxy**
extends AEFEventListenerProxy
implements java.rmi.Remote, java.io.Serializable, AEFListenerProxyInterface, OptionsListener

The OptionsListenerProxy API provides a proxy object for remote listeners to monitor the Options events of its associated terminal session.

Remote proxy objects extend the appropriate RMI server classes which handle the RMI duties for the client (so the client does not need to deal with rmi server issues). The proxy objects also perform the role of event dispatching.

By default, the listener proxy object will perform event queuing on an AEF event thread. This relieves the client of any "thread swapping" responsibilities and insures that the AEF event dispatching performance is not affected by client processing. To override this default behavior, use the *setDispatchQueue()*method. The proxy utilizes the EventDispatcher for queueing event listener notification.

To use a proxy object, the client must implement the listener interface and get the POSDataProvider object from the AEFSession to monitor. The proxy object is then constructed passing a reference to the client and the data provider. The proxy object performs the listener registration and forwards all events from the data provider to the client.

See Also:
Sample Usage, com.ibm.retail.AEF.client.EventDispatcher EventDispatcher

---

Field Summary

com.ibm.retail.AEF.data.POSDataProvider	dataProvider
com.ibm.retail.AEF.event.OptionsListener	listener
static java.lang.reflect.Method	listenerMethod

Fields inherited from : class com.ibm.retail.AEF.client.AEFEventListenerProxy
dispatchQueueName, SOCKET_CONNECT_TIMEOUT, SOCKET_READ_TIMEOUT

Fields inherited from : class com.ibm.retail.AEF.util.AEFRemoteObject

DEFAULT_CONNECT_TIMEOUT, DEFAULT_READ_TIMEOUT

Fields inherited from : class `java.rmi.server.RemoteObject`
ref

Constructor Summary

`OptionsListenerProxy(POSDataProvider dataProvider,OptionsListener dataProvider)` Construct an OptionsListenerProxy.

Method Summary

void	`optionsLoaded(OptionsEvent evt)` Options were loaded for this terminal sessions.
void	`removeListener()` Remove the proxy listener from the POSDataProvider.

Methods inherited from : class `com.ibm.retail.AEF.client.AEFEventListenerProxy`
dispatchEvent, getDispatchQueueName, getListenerMethod, setDispatchQueueName

Methods inherited from : class `com.ibm.retail.AEF.util.AEFRemoteObject`
getClientSocketConnectTimeout, getClientSocketReadTimeout

Methods inherited from : class `java.rmi.server.UnicastRemoteObject`
clone, exportObject, exportObject, exportObject, unexportObject

Methods inherited from : class `java.rmi.server.RemoteServer`
getClientHost, getLog, setLog

Methods inherited from : class `java.rmi.server.RemoteObject`
equals, getRef, hashCode, toString, toStub

Methods inherited from : class `java.lang.Object`
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

(continued from last page)

listenerMethod

```
protected static java.lang.reflect.Method listenerMethod
```

dataProvider

```
protected com.ibm.retail.AEF.data.POSDataProvider dataProvider
```

listener

```
protected com.ibm.retail.AEF.event.OptionsListener listener
```

Constructors

OptionsListenerProxy

```
public OptionsListenerProxy(POSDataProvider dataProvider,
                            OptionsListener listener)
```

Construct an OptionsListenerProxy. This method registers the proxy as a listener with the POSDataProvider and begins event notification to the proxy client listener.

Parameters:

`dataProvider` -
instance of POSDataProvider which is the source of events
`listener` -
client listener which receives the event notification via this proxy

Exceptions:

`java.rmi.RemoteException`
`AEFException` -
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

Methods

removeListener

```
public void removeListener()
                    throws java.rmi.RemoteException,
                           AEFException
```

Remove the proxy listener from the POSDataProvider. Ends notification of events.

Exceptions:

`java.rmi.RemoteException`
`AEFException` -
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

optionsLoaded

```
public void optionsLoaded(OptionsEvent evt)
                    throws java.rmi.RemoteException
```

Options were loaded for this terminal sessions. Dispatches event notification to the proxy client listener.

(continued from last page)

Parameters:

`evt -`
contains options details

Exceptions:

`RemoteException -`
if remote access fails

com.ibm.retail.AEF.client

Class PointsListenerProxy

```
java.lang.Object
  |-java.rmi.server.RemoteObject
    |-java.rmi.server.RemoteServer
      |-java.rmi.server.UnicastRemoteObject
        |-com.ibm.retail.AEF.util.AEFRemoteObject
          |-com.ibm.retail.AEF.client.AEFEventListenerProxy
            |-com.ibm.retail.AEF.client.PointsListenerProxy
```

All Implemented interfaces:
PointsListener, AEFListenerProxyInterface, java.io.Serializable, java.rmi.Remote

---

public class **PointsListenerProxy**
extends AEFEventListenerProxy
implements java.rmi.Remote, java.io.Serializable, AEFListenerProxyInterface, PointsListener

PointsListenerProxy provides a proxy object for remote listeners to monitor the Points events of its associated terminal session.

Remote proxy objects extend the appropriate RMI server classes which handle the RMI duties for the client (so the client does not need to deal with rmi server issues). The proxy objects also perform the role of event dispatching.

By default, the listener proxy object will perform event queuing on an AEF event thread. This relieves the client of any "thread swapping" responsibilities and insures that the AEF event dispatching performance is not affected by client processing. To override this default behavior, use the *setDispatchQueue()*method. The proxy utilizes the EventDispatcher for queueing event listener notification.

To use a proxy object, the client must implement the listener interface and get the POSDataProvider object from the AEFSession to monitor. The proxy object is then constructed passing a reference to the client and the data provider. The proxy object performs the listener registration and forwards all events from the data provider to the client.

See Also:
Sample Usage, com.ibm.retail.AEF.client.EventDispatcher EventDispatcher

---

Field Summary

com.ibm.retail.AEF.data.POSDataProvider	dataProvider
com.ibm.retail.AEF.event.PointsListener	listener
static java.lang.reflect.Method	listenerMethod

Fields inherited from : class com.ibm.retail.AEF.client.AEFEventListenerProxy

dispatchQueueName, SOCKET_CONNECT_TIMEOUT, SOCKET_READ_TIMEOUT

Fields inherited from : class com.ibm.retail.AEF.util.AEFRemoteObject

DEFAULT_CONNECT_TIMEOUT, DEFAULT_READ_TIMEOUT

Fields inherited from : class `java.rmi.server.RemoteObject`
ref

Constructor Summary

`PointsListenerProxy(POSDataProvider dataProvider,PointsListener dataProvider)` Construct a PointsListenerProxy.

Method Summary

void	`pointsOccurred(PointsEvent evt)` Points were occurred (awarded or redeemed).
void	`removeListener()` Remove the proxy listener from the POSDataProvider.

Methods inherited from : class `com.ibm.retail.AEF.client.AEFEventListenerProxy`
dispatchEvent, getDispatchQueueName, getListenerMethod, setDispatchQueueName

Methods inherited from : class `com.ibm.retail.AEF.util.AEFRemoteObject`
getClientSocketConnectTimeout, getClientSocketReadTimeout

Methods inherited from : class `java.rmi.server.UnicastRemoteObject`
clone, exportObject, exportObject, exportObject, unexportObject

Methods inherited from : class `java.rmi.server.RemoteServer`
getClientHost, getLog, setLog

Methods inherited from : class `java.rmi.server.RemoteObject`
equals, getRef, hashCode, toString, toStub

Methods inherited from : class `java.lang.Object`
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

(continued from last page)

listenerMethod

```
protected static java.lang.reflect.Method listenerMethod
```

dataProvider

```
protected com.ibm.retail.AEF.data.POSDataProvider dataProvider
```

listener

```
protected com.ibm.retail.AEF.event.PointsListener listener
```

Constructors

PointsListenerProxy

```
public PointsListenerProxy(POSDataProvider dataProvider,
                           PointsListener listener)
```

Construct a PointsListenerProxy. This method registers the proxy as a listener with the POSDataProvider and begins event notification to the proxy client listener.

Parameters:

`dataProvider` -
instance of POSDataProvider which is the source of events
`listener` -
client listener which receives the event notification via this proxy

Exceptions:

`java.rmi.RemoteException`
`AEFException` -
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

Methods

removeListener

```
public void removeListener()
                    throws java.rmi.RemoteException,
                           AEFException
```

Remove the proxy listener from the POSDataProvider. Ends notification of events.

Exceptions:

`java.rmi.RemoteException`
`AEFException` -
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

pointsOccurred

```
public void pointsOccurred(PointsEvent evt)
                    throws java.rmi.RemoteException
```

Points were occurred (awarded or redeemed). Dispatches event notification to the proxy client listener.

(continued from last page)

Parameters:

`evt -`
contains points details

Exceptions:

`RemoteException -`
if remote access fails

com.ibm.retail.AEF.client

Class ReportListenerProxy

```
java.lang.Object
  |-java.rmi.server.RemoteObject
    |-java.rmi.server.RemoteServer
      |-java.rmi.server.UnicastRemoteObject
        |-com.ibm.retail.AEF.util.AEFRemoteObject
          |-com.ibm.retail.AEF.client.AEFEventListenerProxy
            |-com.ibm.retail.AEF.client.ReportListenerProxy
```

All Implemented interfaces:
ReportListener, AEFListenerProxyInterface, java.io.Serializable, java.rmi.Remote

---

public class **ReportListenerProxy**
extends AEFEventListenerProxy
implements java.rmi.Remote, java.io.Serializable, AEFListenerProxyInterface, ReportListener

ReportListenerProxy provides a proxy object for remote listeners to monitor the report events of its associated terminal session.

Remote proxy objects extend the appropriate RMI server classes which handle the RMI duties for the client (so the client does not need to deal with rmi server issues). The proxy objects also perform the role of event dispatching.

By default, the listener proxy object will perform event queuing on an AEF event thread. This relieves the client of any "thread swapping" responsibilities and insures that the AEF event dispatching performance is not affected by client processing. To override this default behavior, use the *setDispatchQueue()*method. The proxy utilizes the EventDispatcher for queueing event listener notification.

To use a proxy object, the client must implement the listener interface and get the POSDataProvider object from the AEFSession to monitor. The proxy object is then constructed passing a reference to the client and the data provider. The proxy object performs the listener registration and forwards all events from the data provider to the client.

See Also:
Sample Usage, com.ibm.retail.AEF.client.EventDispatcher EventDispatcher

---

Field Summary

com.ibm.retail.AEF.data.POSDataProvider	dataProvider
com.ibm.retail.AEF.event.ReportListener	listener
static java.lang.reflect.Method	listenerMethod

Fields inherited from : class com.ibm.retail.AEF.client.AEFEventListenerProxy

dispatchQueueName, SOCKET_CONNECT_TIMEOUT, SOCKET_READ_TIMEOUT

Fields inherited from : class com.ibm.retail.AEF.util.AEFRemoteObject

DEFAULT_CONNECT_TIMEOUT, DEFAULT_READ_TIMEOUT

Fields inherited from : class `java.rmi.server.RemoteObject`
ref

Constructor Summary

`ReportListenerProxy(POSDataProvider dataProvider,ReportListener dataProvider)` Construct a ReportListenerProxy.

Method Summary

void	`removeListener()` Remove the proxy listener from the POSDataProvider.
void	`reportDataAvailable(ReportEvent evt)` Report event occurred.

Methods inherited from : class `com.ibm.retail.AEF.client.AEFEventListenerProxy`
dispatchEvent, getDispatchQueueName, getListenerMethod, setDispatchQueueName

Methods inherited from : class `com.ibm.retail.AEF.util.AEFRemoteObject`
getClientSocketConnectTimeout, getClientSocketReadTimeout

Methods inherited from : class `java.rmi.server.UnicastRemoteObject`
clone, exportObject, exportObject, exportObject, unexportObject

Methods inherited from : class `java.rmi.server.RemoteServer`
getClientHost, getLog, setLog

Methods inherited from : class `java.rmi.server.RemoteObject`
equals, getRef, hashCode, toString, toStub

Methods inherited from : class `java.lang.Object`
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

(continued from last page)

listenerMethod

```
protected static java.lang.reflect.Method listenerMethod
```

dataProvider

```
protected com.ibm.retail.AEF.data.POSDataProvider dataProvider
```

listener

```
protected com.ibm.retail.AEF.event.ReportListener listener
```

Constructors

ReportListenerProxy

```
public ReportListenerProxy(POSDataProvider dataProvider,
                           ReportListener listener)
```

Construct a ReportListenerProxy. This method registers the proxy as a listener with the POSDataProvider and begins event notification to the proxy client listener.

Parameters:

`dataProvider` -
instance of POSDataProvider which is the source of events
`listener` -
client listener which receives the event notification via this proxy

Exceptions:

`java.rmi.RemoteException`
`AEFException` -
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

Methods

removeListener

```
public void removeListener()
                    throws java.rmi.RemoteException,
                           AEFException
```

Remove the proxy listener from the POSDataProvider. Ends notification of events.

Exceptions:

`java.rmi.RemoteException`
`AEFException` -
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

reportDataAvailable

```
public void reportDataAvailable(ReportEvent evt)
                        throws java.rmi.RemoteException
```

Report event occurred.

(continued from last page)

Parameters:

`evt` -
contains report information

Exceptions:

`RemoteException` -
if remote access fails

com.ibm.retail.AEF.client

Class ScaleListenerProxy

```
java.lang.Object
  +-java.rmi.server.RemoteObject
    +-java.rmi.server.RemoteServer
      +-java.rmi.server.UnicastRemoteObject
        +-com.ibm.retail.AEF.util.AEFRemoteObject
          +-com.ibm.retail.AEF.client.AEFEventListenerProxy
            +-com.ibm.retail.AEF.client.ScaleListenerProxy
```

All Implemented interfaces:
ScaleListener, AEFListenerProxyInterface, java.io.Serializable, java.rmi.Remote

---

public class **ScaleListenerProxy**
extends AEFEventListenerProxy
implements java.rmi.Remote, java.io.Serializable, AEFListenerProxyInterface, ScaleListener

ScaleListenerProxy provides a proxy object for remote listeners to monitor the Scale events of its associated terminal session.

Remote proxy objects extend the appropriate RMI server classes which handle the RMI duties for the client (so the client does not need to deal with rmi server issues). The proxy objects also perform the role of event dispatching.

By default, the listener proxy object will perform event queuing on an AEF event thread. This relieves the client of any "thread swapping" responsibilities and insures that the AEF event dispatching performance is not affected by client processing. To override this default behavior, use the *setDispatchQueue()*method. The proxy utilizes the EventDispatcher for queueing event listener notification.

To use a proxy object, the client must implement the listener interface and get the POSDataProvider object from the AEFSession to monitor. The proxy object is then constructed passing a reference to the client and the data provider. The proxy object performs the listener registration and forwards all events from the data provider to the client.

See Also:
Sample Usage, com.ibm.retail.AEF.client.EventDispatcher EventDispatcher

---

Field Summary

com.ibm.retail.AEF.data.POSDataProvider	dataProvider
com.ibm.retail.AEF.event.ScaleListener	listener
static java.lang.reflect.Method	listenerMethod

Fields inherited from : class com.ibm.retail.AEF.client.AEFEventListenerProxy
dispatchQueueName, SOCKET_CONNECT_TIMEOUT, SOCKET_READ_TIMEOUT

Fields inherited from : class com.ibm.retail.AEF.util.AEFRemoteObject

DEFAULT_CONNECT_TIMEOUT, DEFAULT_READ_TIMEOUT

Fields inherited from : class `java.rmi.server.RemoteObject`
ref

Constructor Summary

`ScaleListenerProxy(POSDataProvider dataProvider,ScaleListener dataProvider)` Construct a ScaleListenerProxy.

Method Summary

void	`itemWeighed(ScaleEvent evt)` An item was weighed.
void	`removeListener()` Remove the proxy listener from the POSDataProvider.

Methods inherited from : class `com.ibm.retail.AEF.client.AEFEventListenerProxy`
dispatchEvent, getDispatchQueueName, getListenerMethod, setDispatchQueueName

Methods inherited from : class `com.ibm.retail.AEF.util.AEFRemoteObject`
getClientSocketConnectTimeout, getClientSocketReadTimeout

Methods inherited from : class `java.rmi.server.UnicastRemoteObject`
clone, exportObject, exportObject, exportObject, unexportObject

Methods inherited from : class `java.rmi.server.RemoteServer`
getClientHost, getLog, setLog

Methods inherited from : class `java.rmi.server.RemoteObject`
equals, getRef, hashCode, toString, toStub

Methods inherited from : class `java.lang.Object`
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

(continued from last page)

listenerMethod

```
protected static java.lang.reflect.Method listenerMethod
```

dataProvider

```
protected com.ibm.retail.AEF.data.POSDataProvider dataProvider
```

listener

```
protected com.ibm.retail.AEF.event.ScaleListener listener
```

Constructors

ScaleListenerProxy

```
public ScaleListenerProxy(POSDataProvider dataProvider,
                          ScaleListener listener)
```

Construct a ScaleListenerProxy. This method registers the proxy as a listener with the POSDataProvider and begins event notification to the proxy client listener.

Parameters:

`dataProvider` -
instance of POSDataProvider which is the source of events
`listener` -
client listener which receives the event notification via this proxy

Exceptions:

`java.rmi.RemoteException`
`AEFException` -
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

Methods

removeListener

```
public void removeListener()
                    throws java.rmi.RemoteException,
                           AEFException
```

Remove the proxy listener from the POSDataProvider. Ends notification of events.

Exceptions:

`java.rmi.RemoteException`
`AEFException` -
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

itemWeighed

```
public void itemWeighed(ScaleEvent evt)
                 throws java.rmi.RemoteException
```

An item was weighed. Dispatches event notification to the proxy client listener.

(continued from last page)

Parameters:

`evt` -
contains points details

Exceptions:

`RemoteException` -
if remote access fails

com.ibm.retail.AEF.client

Class SessionStatusListenerProxy

```
java.lang.Object
  |-java.rmi.server.RemoteObject
    |-java.rmi.server.RemoteServer
      |-java.rmi.server.UnicastRemoteObject
        |-com.ibm.retail.AEF.util.AEFRemoteObject
          |-com.ibm.retail.AEF.client.AEFEventListenerProxy
            |-com.ibm.retail.AEF.client.SessionStatusListenerProxy
```

All Implemented interfaces:
SessionStatusListener, AEFListenerProxyInterface, java.io.Serializable, java.rmi.Remote

---

public class **SessionStatusListenerProxy**
extends AEFEventListenerProxy
implements java.rmi.Remote, java.io.Serializable, AEFListenerProxyInterface, SessionStatusListener

SessionStatusListenerProxy provides a proxy object for remote listeners to monitor the SessionStatus events of its associated terminal session.

Remote proxy objects extend the appropriate RMI server classes which handle the RMI duties for the client (so the client does not need to deal with rmi server issues). The proxy objects also perform the role of event dispatching.

By default, the listener proxy object will perform event queuing on an AEF event thread. This relieves the client of any "thread swapping" responsibilities and insures that the AEF event dispatching performance is not affected by client processing. To override this default behavior, use the *setDispatchQueue()*method. The proxy utilizes the EventDispatcher for queueing event listener notification.

To use a proxy object, the client must implement the listener interface and get the POSDataProvider object from the AEFSession to monitor. The proxy object is then constructed passing a reference to the client and the data provider. The proxy object performs the listener registration and forwards all events from the data provider to the client.

See Also:
Sample Usage, com.ibm.retail.AEF.client.EventDispatcher EventDispatcher

---

Field Summary

com.ibm.retail.AEF.session.SessionStatusListener	listener
com.ibm.retail.AEF.session.AEFSession	session
static java.lang.reflect.Method	sessionEndedMethod
static java.lang.reflect.Method	sessionReadyMethod

static java.lang.reflect.Method	sessionStatusChangedMethod

Fields inherited from : class com.ibm.retail.AEF.client.AEFEventListenerProxy
dispatchQueueName, SOCKET_CONNECT_TIMEOUT, SOCKET_READ_TIMEOUT

Fields inherited from : class com.ibm.retail.AEF.util.AEFRemoteObject
DEFAULT_CONNECT_TIMEOUT, DEFAULT_READ_TIMEOUT

Fields inherited from : class java.rmi.server.RemoteObject
ref

Constructor Summary

SessionStatusListenerProxy(AEFSession session,SessionStatusListener session) Construct a SessionStatusListenerProxy.
SessionStatusListenerProxy(AEFSession session,SessionStatusListener session,int session,int session) Construct a SessionStatusListenerProxy.

Method Summary

void	removeListener() Remove the proxy listener from AEFSession.
void	sessionEnded(SessionStatusEvent evt) The session ended.
void	sessionReady(SessionStatusEvent evt) The session is ready.
void	sessionStatusChanged(SessionStatusEvent evt) A session status event has occurred.

Methods inherited from : class com.ibm.retail.AEF.client.AEFEventListenerProxy
dispatchEvent, getDispatchQueueName, getListenerMethod, setDispatchQueueName

Methods inherited from : class com.ibm.retail.AEF.util.AEFRemoteObject
getClientSocketConnectTimeout, getClientSocketReadTimeout

Methods inherited from : class java.rmi.server.UnicastRemoteObject
clone, exportObject, exportObject, exportObject, unexportObject

Methods inherited from : class java.rmi.server.RemoteServer

`getClientHost, getLog, setLog`

Methods inherited from : class `java.rmi.server.RemoteObject`
`equals, getRef, hashCode, toString, toStub`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

sessionStatusChangedMethod

`protected static java.lang.reflect.Method` **`sessionStatusChangedMethod`**

sessionReadyMethod

`protected static java.lang.reflect.Method` **`sessionReadyMethod`**

sessionEndedMethod

`protected static java.lang.reflect.Method` **`sessionEndedMethod`**

session

`protected com.ibm.retail.AEF.session.AEFSession` **`session`**

listener

`protected com.ibm.retail.AEF.session.SessionStatusListener` **`listener`**

Constructors

SessionStatusListenerProxy

```
public SessionStatusListenerProxy(AEFSession session,
                                  SessionStatusListener listener)
```

Construct a SessionStatusListenerProxy. This method registers the proxy as a listener with the AEFSession and begins event notification to the proxy client listener.

Parameters:

`session` -
instance of AEFSession which is the source of events

(continued from last page)

`listener -`
client listener which receives the event notification via this proxy

Exceptions:

`java.rmi.RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

SessionStatusListenerProxy

```
public SessionStatusListenerProxy(AEFSession session,
                                  SessionStatusListener listener,
                                  int soTimeout,
                                  int connectTimeout)
```

Construct a SessionStatusListenerProxy. This method registers the proxy as a listener with the AEFSession and begins event notification to the proxy client listener.

Parameters:

`session -`
instance of AEFSession which is the source of events
`listener -`
client listener which receives the event notification via this proxy

Exceptions:

`java.rmi.RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

Methods

removeListener

```
public void removeListener()
                    throws java.rmi.RemoteException,
                           AEFException
```

Remove the proxy listener from AEFSession. Ends notification of events.

Exceptions:

`java.rmi.RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

---

sessionStatusChanged

```
public void sessionStatusChanged(SessionStatusEvent evt)
                           throws java.rmi.RemoteException
```

A session status event has occurred. Dispatches event notification to the proxy client listener.

Parameters:

`evt -`
SessionStatusEvent

---

sessionReady

```
public void sessionReady(SessionStatusEvent evt)
                    throws java.rmi.RemoteException
```

The session is ready. Dispatches event notification to the proxy client listener.

(continued from last page)

Parameters:

`evt -` SessionStatusEvent

sessionEnded

```
public void sessionEnded(SessionStatusEvent evt)
                  throws java.rmi.RemoteException
```

The session ended. Dispatches event notification to the proxy client listener.

Parameters:

`evt -` SessionStatusEvent

com.ibm.retail.AEF.client

Class StateChangeListenerProxy

```
java.lang.Object
  |-java.rmi.server.RemoteObject
    |-java.rmi.server.RemoteServer
      |-java.rmi.server.UnicastRemoteObject
        |-com.ibm.retail.AEF.util.AEFRemoteObject
          |-com.ibm.retail.AEF.client.AEFEventListenerProxy
            |-com.ibm.retail.AEF.client.StateChangeListenerProxy
```

All Implemented interfaces:
StateChangeListener, AEFListenerProxyInterface, java.io.Serializable, java.rmi.Remote

---

public class **StateChangeListenerProxy**
extends AEFEventListenerProxy
implements java.rmi.Remote, java.io.Serializable, AEFListenerProxyInterface, StateChangeListener

StateChangeListenerProxy provides a proxy object for remote listeners to monitor the StateChange events of its associated terminal session.

Remote proxy objects extend the appropriate RMI server classes which handle the RMI duties for the client (so the client does not need to deal with rmi server issues). The proxy objects also perform the role of event dispatching.

By default, the listener proxy object will perform event queuing on an AEF event thread. This relieves the client of any "thread swapping" responsibilities and insures that the AEF event dispatching performance is not affected by client processing. To override this default behavior, use the *setDispatchQueue()*method. The proxy utilizes the EventDispatcher for queueing event listener notification.

To use a proxy object, the client must implement the listener interface and get the POSDataProvider object from the AEFSession to monitor. The proxy object is then constructed passing a reference to the client and the data provider. The proxy object performs the listener registration and forwards all events from the data provider to the client.

See Also:
Sample Usage, com.ibm.retail.AEF.client.EventDispatcher EventDispatcher

---

Field Summary

com.ibm.retail.AEF.data.POSDataProvider	dataProvider
com.ibm.retail.AEF.event.StateChangeListener	listener
static java.lang.reflect.Method	listenerMethod

Fields inherited from : class com.ibm.retail.AEF.client.AEFEventListenerProxy
dispatchQueueName, SOCKET_CONNECT_TIMEOUT, SOCKET_READ_TIMEOUT

Fields inherited from : class `com.ibm.retail.AEF.util.AEFRemoteObject`
`DEFAULT_CONNECT_TIMEOUT`, `DEFAULT_READ_TIMEOUT`

Fields inherited from : class `java.rmi.server.RemoteObject`
`ref`

Constructor Summary

`StateChangeListenerProxy(POSDataProvider dataProvider,StateChangeListener dataProvider)`

Construct a StateChangeListenerProxy This method registers the proxy as a listener with the POSDataProvider and begins event notification to the proxy client listener.

Method Summary

`void`	`removeListener()` Remove the proxy listener from the POSDataProvider.
`void`	`stateChanged(StateChangeEvent evt)` A state change event occurred on the POS terminal session.

Methods inherited from : class `com.ibm.retail.AEF.client.AEFEventListenerProxy`
`dispatchEvent, getDispatchQueueName, getListenerMethod, setDispatchQueueName`

Methods inherited from : class `com.ibm.retail.AEF.util.AEFRemoteObject`
`getClientSocketConnectTimeout, getClientSocketReadTimeout`

Methods inherited from : class `java.rmi.server.UnicastRemoteObject`
`clone, exportObject, exportObject, exportObject, unexportObject`

Methods inherited from : class `java.rmi.server.RemoteServer`
`getClientHost, getLog, setLog`

Methods inherited from : class `java.rmi.server.RemoteObject`
`equals, getRef, hashCode, toString, toStub`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

(continued from last page)

Fields

listenerMethod

```
protected static java.lang.reflect.Method listenerMethod
```

dataProvider

```
protected com.ibm.retail.AEF.data.POSDataProvider dataProvider
```

listener

```
protected com.ibm.retail.AEF.event.StateChangeListener listener
```

Constructors

StateChangeListenerProxy

```
public StateChangeListenerProxy(POSDataProvider dataProvider,
                                StateChangeListener listener)
```

Construct a StateChangeListenerProxy This method registers the proxy as a listener with the POSDataProvider and begins event notification to the proxy client listener.

Parameters:

`dataProvider` -
instance of POSDataProvider which is the source of events
`listener` -
client listener which receives the event notification via this proxy

Exceptions:

`java.rmi.RemoteException`
`AEFException` -
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

Methods

removeListener

```
public void removeListener()
                    throws java.rmi.RemoteException,
                           AEFException
```

Remove the proxy listener from the POSDataProvider. Ends notification of events.

Exceptions:

`java.rmi.RemoteException`
`AEFException` -
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

(continued from last page)

stateChanged

```
public void stateChanged(StateChangeEvent evt)
                  throws java.rmi.RemoteException
```

A state change event occurred on the POS terminal session. Dispatches event notification to the proxy client listener.

Parameters:

`evt -`
contains state details

Exceptions:

`RemoteException -`
if remote access fails

com.ibm.retail.AEF.client

Class TenderListenerProxy

```
java.lang.Object
  +-java.rmi.server.RemoteObject
    +-java.rmi.server.RemoteServer
      +-java.rmi.server.UnicastRemoteObject
        +-com.ibm.retail.AEF.util.AEFRemoteObject
          +-com.ibm.retail.AEF.client.AEFEventListenerProxy
            +-com.ibm.retail.AEF.client.TenderListenerProxy
```

All Implemented interfaces:
TenderListener, AEFListenerProxyInterface, java.io.Serializable, java.rmi.Remote

---

public class **TenderListenerProxy**
extends AEFEventListenerProxy
implements java.rmi.Remote, java.io.Serializable, AEFListenerProxyInterface, TenderListener

TenderListenerProxy provides a proxy object for remote listeners to monitor the Tender events of its associated terminal session.

Remote proxy objects extend the appropriate RMI server classes which handle the RMI duties for the client (so the client does not need to deal with rmi server issues). The proxy objects also perform the role of event dispatching.

By default, the listener proxy object will perform event queuing on an AEF event thread. This relieves the client of any "thread swapping" responsibilities and insures that the AEF event dispatching performance is not affected by client processing. To override this default behavior, use the *setDispatchQueue()*method. The proxy utilizes the EventDispatcher for queueing event listener notification.

To use a proxy object, the client must implement the listener interface and get the POSDataProvider object from the AEFSession to monitor. The proxy object is then constructed passing a reference to the client and the data provider. The proxy object performs the listener registration and forwards all events from the data provider to the client.

See Also:
Sample Usage, com.ibm.retail.AEF.client.EventDispatcher EventDispatcher

---

Field Summary

com.ibm.retail.AEF.data.POSDataProvider	dataProvider
com.ibm.retail.AEF.event.TenderListener	listener
static java.lang.reflect.Method	listenerMethod

Fields inherited from : class com.ibm.retail.AEF.client.AEFEventListenerProxy
dispatchQueueName, SOCKET_CONNECT_TIMEOUT, SOCKET_READ_TIMEOUT

Fields inherited from : class com.ibm.retail.AEF.util.AEFRemoteObject

DEFAULT_CONNECT_TIMEOUT, DEFAULT_READ_TIMEOUT

Fields inherited from : class `java.rmi.server.RemoteObject`
ref

Constructor Summary

`TenderListenerProxy(POSDataProvider dataProvider,TenderListener dataProvider)` Construct a TenderListenerProxy.

Method Summary

void	`removeListener()` Remove the proxy listener from the POSDataProvider.
void	`tenderAccepted(TenderEvent evt)` A tender was aaccepted in the transaction.

Methods inherited from : class `com.ibm.retail.AEF.client.AEFEventListenerProxy`
dispatchEvent, getDispatchQueueName, getListenerMethod, setDispatchQueueName

Methods inherited from : class `com.ibm.retail.AEF.util.AEFRemoteObject`
getClientSocketConnectTimeout, getClientSocketReadTimeout

Methods inherited from : class `java.rmi.server.UnicastRemoteObject`
clone, exportObject, exportObject, exportObject, unexportObject

Methods inherited from : class `java.rmi.server.RemoteServer`
getClientHost, getLog, setLog

Methods inherited from : class `java.rmi.server.RemoteObject`
equals, getRef, hashCode, toString, toStub

Methods inherited from : class `java.lang.Object`
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

(continued from last page)

listenerMethod

```
protected static java.lang.reflect.Method listenerMethod
```

dataProvider

```
protected com.ibm.retail.AEF.data.POSDataProvider dataProvider
```

listener

```
protected com.ibm.retail.AEF.event.TenderListener listener
```

Constructors

TenderListenerProxy

```
public TenderListenerProxy(POSDataProvider dataProvider,
                           TenderListener listener)
```

Construct a TenderListenerProxy. This method registers the proxy as a listener with the POSDataProvider and begins event notification to the proxy client listener.

Parameters:

`dataProvider` -
instance of POSDataProvider which is the source of events
`listener` -
client listener which receives the event notification via this proxy

Exceptions:

`java.rmi.RemoteException`
`AEFException` -
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

Methods

removeListener

```
public void removeListener()
                    throws java.rmi.RemoteException,
                           AEFException
```

Remove the proxy listener from the POSDataProvider. Ends notification of events.

Exceptions:

`java.rmi.RemoteException`
`AEFException` -
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

tenderAccepted

```
public void tenderAccepted(TenderEvent evt)
                    throws java.rmi.RemoteException
```

A tender was aaccepted in the transaction. Dispatches event notification to the proxy client listener.

(continued from last page)

Parameters:

`evt` -
contains tender details

Exceptions:

`RemoteException` -
if remote access fails

com.ibm.retail.AEF.client

Class TransactionStatusListenerProxy

```
java.lang.Object
  |-java.rmi.server.RemoteObject
    |-java.rmi.server.RemoteServer
      |-java.rmi.server.UnicastRemoteObject
        |-com.ibm.retail.AEF.util.AEFRemoteObject
          |-com.ibm.retail.AEF.client.AEFEventListenerProxy
            |-com.ibm.retail.AEF.client.TransactionStatusListenerProxy
```

All Implemented interfaces:
TransactionStatusListener, AEFListenerProxyInterface, java.io.Serializable, java.rmi.Remote

---

public class **TransactionStatusListenerProxy**
extends AEFEventListenerProxy
implements java.rmi.Remote, java.io.Serializable, AEFListenerProxyInterface, TransactionStatusListener

TransactionStatusListenerProxy provides a proxy object for remote listeners to monitor the TransactionStatus events of its associated terminal session.

Remote proxy objects extend the appropriate RMI server classes which handle the RMI duties for the client (so the client does not need to deal with rmi server issues). The proxy objects also perform the role of event dispatching.

By default, the listener proxy object will perform event queuing on an AEF event thread. This relieves the client of any "thread swapping" responsibilities and insures that the AEF event dispatching performance is not affected by client processing. To override this default behavior, use the *setDispatchQueue()*method. The proxy utilizes the EventDispatcher for queueing event listener notification.

To use a proxy object, the client must implement the listener interface and get the POSDataProvider object from the AEFSession to monitor. The proxy object is then constructed passing a reference to the client and the data provider. The proxy object performs the listener registration and forwards all events from the data provider to the client.

See Also:
Sample Usage, com.ibm.retail.AEF.client.EventDispatcher EventDispatcher

---

Field Summary

com.ibm.retail.AEF.data.POSDataProvider	dataProvider
com.ibm.retail.AEF.event.TransactionStatusListener	listener
static java.lang.reflect.Method	listenerMethod

Fields inherited from : class com.ibm.retail.AEF.client.AEFEventListenerProxy
dispatchQueueName, SOCKET_CONNECT_TIMEOUT, SOCKET_READ_TIMEOUT

Fields inherited from : class `com.ibm.retail.AEF.util.AEFRemoteObject`
`DEFAULT_CONNECT_TIMEOUT`, `DEFAULT_READ_TIMEOUT`

Fields inherited from : class `java.rmi.server.RemoteObject`
`ref`

Constructor Summary

`TransactionStatusListenerProxy(POSDataProvider dataProvider,TransactionStatusListener dataProvider)`

Construct a TransactionStatusListenerProxy.

Method Summary

`void`	`removeListener()` Remove the proxy listener from the POSDataProvider.
`void`	`transactionStatusEventOccurred(TransactionStatusEvent evt)` A transaction status event occurred.

Methods inherited from : class `com.ibm.retail.AEF.client.AEFEventListenerProxy`
`dispatchEvent, getDispatchQueueName, getListenerMethod, setDispatchQueueName`

Methods inherited from : class `com.ibm.retail.AEF.util.AEFRemoteObject`
`getClientSocketConnectTimeout, getClientSocketReadTimeout`

Methods inherited from : class `java.rmi.server.UnicastRemoteObject`
`clone, exportObject, exportObject, exportObject, unexportObject`

Methods inherited from : class `java.rmi.server.RemoteServer`
`getClientHost, getLog, setLog`

Methods inherited from : class `java.rmi.server.RemoteObject`
`equals, getRef, hashCode, toString, toStub`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

(continued from last page)

Fields

listenerMethod

```
protected static java.lang.reflect.Method listenerMethod
```

dataProvider

```
protected com.ibm.retail.AEF.data.POSDataProvider dataProvider
```

listener

```
protected com.ibm.retail.AEF.event.TransactionStatusListener listener
```

Constructors

TransactionStatusListenerProxy

```
public TransactionStatusListenerProxy(POSDataProvider dataProvider,
                                      TransactionStatusListener listener)
```

Construct a TransactionStatusListenerProxy. This method registers the proxy as a listener with the POSDataProvider and begins event notification to the proxy client listener.

Parameters:

`dataProvider -`
instance of POSDataProvider which is the source of events
`listener -`
client listener which receives the event notification via this proxy

Exceptions:

`java.rmi.RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

Methods

removeListener

```
public void removeListener()
                    throws java.rmi.RemoteException,
                           AEFException
```

Remove the proxy listener from the POSDataProvider. Ends notification of events.

Exceptions:

`java.rmi.RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

(continued from last page)

transactionStatusEventOccurred

```
public void transactionStatusEventOccurred(TransactionStatusEvent evt)
                                           throws java.rmi.RemoteException
```

A transaction status event occurred. Dispatches event notification to the proxy client listener.

Parameters:

`evt -`
contains details of the event

Exceptions:

`RemoteException -`
if remote access fails

com.ibm.retail.AEF.client

Class TransactionTotalsListenerProxy

```
java.lang.Object
  |-java.rmi.server.RemoteObject
    |-java.rmi.server.RemoteServer
      |-java.rmi.server.UnicastRemoteObject
        |-com.ibm.retail.AEF.util.AEFRemoteObject
          |-com.ibm.retail.AEF.client.AEFEventListenerProxy
            |-com.ibm.retail.AEF.client.TransactionTotalsListenerProxy
```

All Implemented interfaces:
TransactionTotalsListener, AEFListenerProxyInterface, java.io.Serializable, java.rmi.Remote

---

public class **TransactionTotalsListenerProxy**
extends AEFEventListenerProxy
implements java.rmi.Remote, java.io.Serializable, AEFListenerProxyInterface, TransactionTotalsListener

The TransactionTotalsListenerProxy API provides a proxy object for remote listeners to monitor the TransactionTotals events of its associated terminal session.

Remote proxy objects extend the appropriate RMI server classes which handle the RMI duties for the client (so the client does not need to deal with rmi server issues). The proxy objects also perform the role of event dispatching.

By default, the listener proxy object will perform event queuing on an AEF event thread. This relieves the client of any "thread swapping" responsibilities and insures that the AEF event dispatching performance is not affected by client processing. To override this default behavior, use the *setDispatchQueue()*method. The proxy utilizes the EventDispatcher for queueing event listener notification.

To use a proxy object, the client must implement the listener interface and get the POSDataProvider object from the AEFSession to monitor. The proxy object is then constructed passing a reference to the client and the data provider. The proxy object performs the listener registration and forwards all events from the data provider to the client.

See Also:
Sample Usage, com.ibm.retail.AEF.client.EventDispatcher EventDispatcher

---

Field Summary

com.ibm.retail.AEF.data.POSDataProvider	dataProvider
com.ibm.retail.AEF.event.TransactionTotalsListener	listener
static java.lang.reflect.Method	listenerMethod

Fields inherited from : class com.ibm.retail.AEF.client.AEFEventListenerProxy
dispatchQueueName, SOCKET_CONNECT_TIMEOUT, SOCKET_READ_TIMEOUT

Fields inherited from : class `com.ibm.retail.AEF.util.AEFRemoteObject`
`DEFAULT_CONNECT_TIMEOUT`, `DEFAULT_READ_TIMEOUT`

Fields inherited from : class `java.rmi.server.RemoteObject`
`ref`

Constructor Summary

`TransactionTotalsListenerProxy(POSDataProvider dataProvider,TransactionTotalsListener dataProvider)`

Construct a TransactionTotalsListenerProxy.

Method Summary

`void`	`removeListener()` Remove the proxy listener from the POSDataProvider.
`void`	`totalsChanged(TransactionTotalsEvent evt)` The transaction totals were updated.

Methods inherited from : class `com.ibm.retail.AEF.client.AEFEventListenerProxy`
`dispatchEvent, getDispatchQueueName, getListenerMethod, setDispatchQueueName`

Methods inherited from : class `com.ibm.retail.AEF.util.AEFRemoteObject`
`getClientSocketConnectTimeout, getClientSocketReadTimeout`

Methods inherited from : class `java.rmi.server.UnicastRemoteObject`
`clone, exportObject, exportObject, exportObject, unexportObject`

Methods inherited from : class `java.rmi.server.RemoteServer`
`getClientHost, getLog, setLog`

Methods inherited from : class `java.rmi.server.RemoteObject`
`equals, getRef, hashCode, toString, toStub`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

(continued from last page)

Fields

listenerMethod

```
protected static java.lang.reflect.Method listenerMethod
```

dataProvider

```
protected com.ibm.retail.AEF.data.POSDataProvider dataProvider
```

listener

```
protected com.ibm.retail.AEF.event.TransactionTotalsListener listener
```

Constructors

TransactionTotalsListenerProxy

```
public TransactionTotalsListenerProxy(POSDataProvider dataProvider,
                                      TransactionTotalsListener listener)
```

Construct a TransactionTotalsListenerProxy. This method registers the proxy as a listener with the POSDataProvider and begins event notification to the proxy client listener.

Parameters:

`dataProvider` -
instance of POSDataProvider which is the source of events
`listener` -
client listener which receives the event notification via this proxy

Exceptions:

`java.rmi.RemoteException`
`AEFException` -
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

Methods

removeListener

```
public void removeListener()
                    throws java.rmi.RemoteException,
                           AEFException
```

Remove the proxy listener from the POSDataProvider. Ends notification of events.

Exceptions:

`java.rmi.RemoteException`
`AEFException` -
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

(continued from last page)

totalsChanged

```
public void totalsChanged(TransactionTotalsEvent evt)
                    throws java.rmi.RemoteException
```

The transaction totals were updated. Dispatches event notification to the proxy client listener.

Parameters:

`evt -`
contains details of the totals event

Exceptions:

`RemoteException -`
if a listener can not be notified

com.ibm.retail.AEF.client

Class WorkstationStatusListenerProxy

```
java.lang.Object
  |-java.rmi.server.RemoteObject
    |-java.rmi.server.RemoteServer
      |-java.rmi.server.UnicastRemoteObject
        |-com.ibm.retail.AEF.util.AEFRemoteObject
          |-com.ibm.retail.AEF.client.AEFEventListenerProxy
            |-com.ibm.retail.AEF.client.WorkstationStatusListenerProxy
```

All Implemented interfaces:
WorkstationStatusListener, AEFListenerProxyInterface, java.io.Serializable, java.rmi.Remote

---

public class **WorkstationStatusListenerProxy**
extends AEFEventListenerProxy
implements java.rmi.Remote, java.io.Serializable, AEFListenerProxyInterface, WorkstationStatusListener

WorkstationStatusListenerProxy provides a proxy object for remote listeners to monitor the WorkstationStatus events of its associated terminal session.

Remote proxy objects extend the appropriate RMI server classes which handle the RMI duties for the client (so the client does not need to deal with rmi server issues). The proxy objects also perform the role of event dispatching.

By default, the listener proxy object will perform event queuing on an AEF event thread. This relieves the client of any "thread swapping" responsibilities and insures that the AEF event dispatching performance is not affected by client processing. To override this default behavior, use the *setDispatchQueue()* method. The proxy utilizes the EventDispatcher for queueing event listener notification.

To use a proxy object, the client must implement the listener interface and get the POSDataProvider object from the AEFSession to monitor. The proxy object is then constructed passing a reference to the client and the data provider. The proxy object performs the listener registration and forwards all events from the data provider to the client.

See Also:
Sample Usage, com.ibm.retail.AEF.client.EventDispatcher EventDispatcher

---

Field Summary

com.ibm.retail.AEF.data.POSDataProvider	dataProvider
com.ibm.retail.AEF.event.WorkstationStatusListener	listener
static java.lang.reflect.Method	listenerMethod

Fields inherited from : class com.ibm.retail.AEF.client.AEFEventListenerProxy
dispatchQueueName, SOCKET_CONNECT_TIMEOUT, SOCKET_READ_TIMEOUT

Fields inherited from : class `com.ibm.retail.AEF.util.AEFRemoteObject`
`DEFAULT_CONNECT_TIMEOUT`, `DEFAULT_READ_TIMEOUT`

Fields inherited from : class `java.rmi.server.RemoteObject`
`ref`

Constructor Summary

`WorkstationStatusListenerProxy(POSDataProvider dataProvider,WorkstationStatusListener dataProvider)`

Construct a WorkstationStatusListenerProxy.

Method Summary

`void`	`removeListener()` Remove the proxy listener from the POSDataProvider.
`void`	`workstationStatusChanged(WorkstationStatusEvent evt)` A change occurred to the workstation status

Methods inherited from : class `com.ibm.retail.AEF.client.AEFEventListenerProxy`
`dispatchEvent, getDispatchQueueName, getListenerMethod, setDispatchQueueName`

Methods inherited from : class `com.ibm.retail.AEF.util.AEFRemoteObject`
`getClientSocketConnectTimeout, getClientSocketReadTimeout`

Methods inherited from : class `java.rmi.server.UnicastRemoteObject`
`clone, exportObject, exportObject, exportObject, unexportObject`

Methods inherited from : class `java.rmi.server.RemoteServer`
`getClientHost, getLog, setLog`

Methods inherited from : class `java.rmi.server.RemoteObject`
`equals, getRef, hashCode, toString, toStub`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

(continued from last page)

Fields

listenerMethod

```
protected static java.lang.reflect.Method listenerMethod
```

dataProvider

```
protected com.ibm.retail.AEF.data.POSDataProvider dataProvider
```

listener

```
protected com.ibm.retail.AEF.event.WorkstationStatusListener listener
```

Constructors

WorkstationStatusListenerProxy

```
public WorkstationStatusListenerProxy(POSDataProvider dataProvider,
                                      WorkstationStatusListener listener)
```

Construct a WorkstationStatusListenerProxy. This method registers the proxy as a listener with the POSDataProvider and begins event notification to the proxy client listener.

Parameters:

`dataProvider` -
instance of POSDataProvider which is the source of events
`listener` -
client listener which receives the event notification via this proxy

Exceptions:

`java.rmi.RemoteException`
`AEFException` -
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

Methods

removeListener

```
public void removeListener()
                    throws java.rmi.RemoteException,
                           AEFException
```

Remove the proxy listener from the POSDataProvider. Ends notification of events.

Exceptions:

`java.rmi.RemoteException`
`AEFException` -
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

(continued from last page)

workstationStatusChanged

```
public void workstationStatusChanged(WorkstationStatusEvent evt)
                                     throws java.rmi.RemoteException
```

A change occurred to the workstation status

Parameters:

`evt -`
contains status details

Exceptions:

`RemoteException -`
if listener can not be notified

Package
com.ibm.retail.AEF.data

Provides interface for monitoring data events associated with the POS application.

The data package includes the POSDataProviderinterface which allows applications to monitor the data events associated with a POS application. Data events provide information which reflects the current state of the POS application including transaction totals, line items, and customer loyalty information.

Event information is obtained through a standard event listener registration and notification process. Listeners must implement the appropriate listener interface and register with the POSDataProvider through an `addListener`method. When an event occurs (e.g., the transaction totals change), the client listener is notified by a call through the listener interface. An event object is passed containing the details of the event. The appropriate `removeListener`method of POSDataProvider should be invoked when the client no longer requires event notification. See the event packagefor information about the listener interfaces and event classes.

The POSDataProvider supports a number of event listener interfaces which can be divided into two main categories:

1. **AEFPropertyChangeListeners**
   are fine-grained property change events which allow a specific, individual property to be observed.
2. **POSAppEventListeners**
   are coarse-grained events which aggregate related data. This allows fewer events but may provide the listener with more data than required.

Usage Notes:

Client Proxy Classes for Listeners

The listener interfaces provided by the AEF extend the `java.rmi.Remote`interface and therefore allow clients to register as "remote" listeners (through RMI) to the POSDataProvider. Since listener registration requires a reference to the client listener for event notification, this implies that the remote client extend the appropriate RMI server classes in order to allow a remote reference to be passed during listener registration (otherwise, an attempt will be made to serialize the listener object). The AEF provides a client packagewhich provides client proxy classes to handle the RMI duties and provide an event dispatching mechanism for AEF listeners.

Category and Property Names

The AEFPropertyPropertyChangeListener interface allows listeners to register for notification of change to a specific property in the POSDataProvider. A **category**defines a logical grouping of properties such as transaction totals or item data. String identifiers for the property names are provided in corresponding interfaces. For example, the property name for the current transaction total is accessible as `TransactionTotalsProperties.TOTAL`.

Accessing Property Values

The POSDataProvider contains a data repository of the current property values. These values are accessible through the `getPropertyValue`methods.

Sample Usage:

```
sample code here...
```

com.ibm.retail.AEF.data

Interface ApplicationDataConnector

public interface **ApplicationDataConnector**
extends java.rmi.Remote

ApplicationDataConnector is the interface which used by the POS application to provide application data to the AEF. Data is sent from the POS application to the AEF in the form of XML strings which are parsed to generate AEF events.

Method Summary

`void`	`processEvent(java.lang.String xmlEvent)` Process the XML event.

Methods

processEvent

```
public void processEvent(java.lang.String xmlEvent)
                  throws AEFException,
                         java.rmi.RemoteException
```

Process the XML event.

Parameters:

`xmlEvent` -
String containing event data in XML format

Exceptions:

`java.rmi.RemoteException`
`AEFException`

com.ibm.retail.AEF.data

Interface CouponProperties

All Superinterfaces:
LineItemProperties, POSDataProperties

public interface **CouponProperties**
extends LineItemProperties

CouponProperties contains the data provider property names for coupon attributes. These properties are communicated to client listeners through the `CouponListener`interface. Individual property values for the current coupon are also accessible through the `POSDataProvider`interface.

See Also:
com.ibm.retail.AEF.event.CouponListener CouponListener, com.ibm.retail.AEF.data.POSDataProvider#getPropertyValue getPropertyValue

Field Summary

static java.lang.String	`AGE_RESTRICTION` Coupon age restriction.
static java.lang.String	`CATEGORY` Category identifier
static java.lang.String	`COUPON_REPEAT_ALLOWED` Coupon repeat allowed.
static java.lang.String	`COUPON_TYPE` Coupon type.
static java.lang.String	`DEAL_PRICE` Coupon deal price.
static java.lang.String	`DEAL_QUANTITY` Coupon deal quantity.
static java.lang.String	`IS_DEPOSIT` Coupon is a deposit.
static java.lang.String	`IS_REFUNDED` Coupon was refunded.
static java.lang.String	`IS_VOIDED` Coupon is voided.
static java.lang.String	`ITEM_MODIFIER` Coupon modifier (provides addtional item type information)
static java.lang.String	`ITEM_TAXABLE` Coupon is taxable.

static java.lang.String	MANUFACTURER_NUMBER Coupon manufacturer number.
static java.lang.String	MULTI_PRICING_GROUP Coupon multi pricing group id.
static java.lang.String	PRICING_METHOD Coupon pricing method.
static java.lang.String	REDUCED_PRICE Coupon reduced price.
static java.lang.String	REDUCES_FOODSTAMP_BALANCE Coupon reduces foodstamp balance.
static java.lang.String	REDUCES_TAX_DUE Coupon reduces taxable balance.
static java.lang.String	RESTRICTED_PERIODS Restricted periods for this coupon.
static java.lang.String	UNIT_PRICE Coupon unit price.
static java.lang.String	VALUE Coupon value.
static java.lang.String	WEIGHT Coupon weight field.
static java.lang.String	WIC_ELIGIBLE Is this coupon eligible for WIC?

Fields

CATEGORY

`public static final java.lang.String` **`CATEGORY`**

Category identifier

COUPON_TYPE

`public static final java.lang.String` **`COUPON_TYPE`**

Coupon type.

UNIT_PRICE

`public static final java.lang.String` **`UNIT_PRICE`**

Coupon unit price.

(continued from last page)

DEAL_QUANTITY

`public static final java.lang.String` **`DEAL_QUANTITY`**

Coupon deal quantity.

WEIGHT

`public static final java.lang.String` **`WEIGHT`**

Coupon weight field.

VALUE

`public static final java.lang.String` **`VALUE`**

Coupon value.

DEAL_PRICE

`public static final java.lang.String` **`DEAL_PRICE`**

Coupon deal price.

REDUCED_PRICE

`public static final java.lang.String` **`REDUCED_PRICE`**

Coupon reduced price.

MULTI_PRICING_GROUP

`public static final java.lang.String` **`MULTI_PRICING_GROUP`**

Coupon multi pricing group id.

AGE_RESTRICTION

`public static final java.lang.String` **`AGE_RESTRICTION`**

Coupon age restriction.

REDUCES_FOODSTAMP_BALANCE

`public static final java.lang.String` **`REDUCES_FOODSTAMP_BALANCE`**

Coupon reduces foodstamp balance.

COUPON_REPEAT_ALLOWED

`public static final java.lang.String` **`COUPON_REPEAT_ALLOWED`**

Coupon repeat allowed.

REDUCES_TAX_DUE

`public static final java.lang.String` **`REDUCES_TAX_DUE`**

Coupon reduces taxable balance.

PRICING_METHOD

`public static final java.lang.String` **`PRICING_METHOD`**

Coupon pricing method.

MANUFACTURER_NUMBER

`public static final java.lang.String` **`MANUFACTURER_NUMBER`**

Coupon manufacturer number.

IS_VOIDED

`public static final java.lang.String` **`IS_VOIDED`**

Coupon is voided.

ITEM_TAXABLE

`public static final java.lang.String` **`ITEM_TAXABLE`**

Coupon is taxable.

IS_DEPOSIT

`public static final java.lang.String` **`IS_DEPOSIT`**

Coupon is a deposit.

IS_REFUNDED

`public static final java.lang.String` **`IS_REFUNDED`**

Coupon was refunded.

ITEM_MODIFIER

`public static final java.lang.String` **`ITEM_MODIFIER`**

Coupon modifier (provides addtional item type information)

WIC_ELIGIBLE

`public static final java.lang.String` **`WIC_ELIGIBLE`**

Is this coupon eligible for WIC?

RESTRICTED_PERIODS

`public static final java.lang.String` **`RESTRICTED_PERIODS`**

Restricted periods for this coupon.

com.ibm.retail.AEF.data

Interface CustomerProperties

All Superinterfaces:
POSDataProperties

public interface **CustomerProperties**
extends POSDataProperties

CustomerProperties contains the data provider property names for customer attributes. These properties are communicated to client listeners through the `CustomerListener`interface. Individual property values for the current customer are also accessible through the `POSDataProvider`interface.

See Also:
com.ibm.retail.AEF.event.CustomerListener CustomerListener,

com.ibm.retail.AEF.data.POSDataProvider#getPropertyValue getPropertyValue

Field Summary

static java.lang.String	`CATEGORY` Category identifier.
static java.lang.String	`COUPON_CODE` Coupon code.
static java.lang.String	`CUSTOMER_ADDRESS1` Customer address line 1.
static java.lang.String	`CUSTOMER_ADDRESS2` Customer address line 2.
static java.lang.String	`CUSTOMER_CITY` Customer city.
static java.lang.String	`CUSTOMER_CONTACT` Customer contact name.
static java.lang.String	`CUSTOMER_EMAIL` Customer email address
static java.lang.String	`CUSTOMER_FAX` Customer fax number.
static java.lang.String	`CUSTOMER_ID` Customer ID or loyalty card number.
static java.lang.String	`CUSTOMER_NAME` Customer full name
static java.lang.String	`CUSTOMER_PHONE` Customer phone number.

static java.lang.String	CUSTOMER_STATE Customer state.
static java.lang.String	CUSTOMER_YTD_PTS Customer loyalty YTD points.
static java.lang.String	CUSTOMER_YTD_SAVED Customer loyalty YTD savings.
static java.lang.String	CUSTOMER_ZIP Customer zipcode.
static java.lang.String	ID_EXPIRATION_DATE Customer ID card expiration date
static java.lang.String	ID_TYPE Customer ID number type ('primary' or 'alternate')
static java.lang.String	MESSAGE Loyalty message.
static java.lang.String	MESSAGES Loyalty messages.
static java.lang.String	POINTS_BALANCE Balance of points earned after redemptions.
static java.lang.String	POINTS_BALANCES Collection of points category balances.
static java.lang.String	POINTS_DESCRIPTION Description of points.
static java.lang.String	POINTS_TOTAL Total points earned.
static java.lang.String	POINTS_TOTALS Collection of points earned categories.
static java.lang.String	POINTS_TYPE Type of points.
static java.lang.String	POINTS_VALUE Number of points .
static java.lang.String	TARGETED_COUPON Idetifier for targeted coupon for customer.
static java.lang.String	TARGETED_COUPONS Targeted coupons for customer.

(continued from last page)

Fields

CATEGORY

`public static final java.lang.String` **`CATEGORY`**

Category identifier.

CUSTOMER_ID

`public static final java.lang.String` **`CUSTOMER_ID`**

Customer ID or loyalty card number.

ID_TYPE

`public static final java.lang.String` **`ID_TYPE`**

Customer ID number type ('primary' or 'alternate')

ID_EXPIRATION_DATE

`public static final java.lang.String` **`ID_EXPIRATION_DATE`**

Customer ID card expiration date

CUSTOMER_NAME

`public static final java.lang.String` **`CUSTOMER_NAME`**

Customer full name

CUSTOMER_EMAIL

`public static final java.lang.String` **`CUSTOMER_EMAIL`**

Customer email address

TARGETED_COUPONS

`public static final java.lang.String` **`TARGETED_COUPONS`**

Targeted coupons for customer. Collection of `String`values, where each `String`is a coupon code.

TARGETED_COUPON

`public static final java.lang.String` **`TARGETED_COUPON`**

Idetifier for targeted coupon for customer.

COUPON_CODE

`public static final java.lang.String` **`COUPON_CODE`**

Coupon code.

POINTS_TOTAL

`public static final java.lang.String` **`POINTS_TOTAL`**

Total points earned.

(continued from last page)

POINTS_BALANCE

`public static final java.lang.String` **`POINTS_BALANCE`**

Balance of points earned after redemptions.

POINTS_TOTALS

`public static final java.lang.String` **`POINTS_TOTALS`**

Collection of points earned categories. Collection of `PointsTotal`objects.

POINTS_BALANCES

`public static final java.lang.String` **`POINTS_BALANCES`**

Collection of points category balances. Collection of `PointsTotal`objects.

MESSAGES

`public static final java.lang.String` **`MESSAGES`**

Loyalty messages. Collection of `String`objects.

MESSAGE

`public static final java.lang.String` **`MESSAGE`**

Loyalty message.

POINTS_TYPE

`public static final java.lang.String` **`POINTS_TYPE`**

Type of points. Attribute of `PointsTotal`

POINTS_DESCRIPTION

`public static final java.lang.String` **`POINTS_DESCRIPTION`**

Description of points. Attibute of `PointsTotal`

POINTS_VALUE

`public static final java.lang.String` **`POINTS_VALUE`**

Number of points . Attribute of `PointsTotal`

CUSTOMER_ADDRESS1

`public static final java.lang.String` **`CUSTOMER_ADDRESS1`**

Customer address line 1.

CUSTOMER_ADDRESS2

`public static final java.lang.String` **`CUSTOMER_ADDRESS2`**

Customer address line 2.

CUSTOMER_CITY

`public static final java.lang.String` **`CUSTOMER_CITY`**

Customer city.

CUSTOMER_STATE

`public static final java.lang.String` **`CUSTOMER_STATE`**

Customer state.

CUSTOMER_ZIP

`public static final java.lang.String` **`CUSTOMER_ZIP`**

Customer zipcode.

CUSTOMER_PHONE

`public static final java.lang.String` **`CUSTOMER_PHONE`**

Customer phone number.

CUSTOMER_CONTACT

`public static final java.lang.String` **`CUSTOMER_CONTACT`**

Customer contact name.

CUSTOMER_FAX

`public static final java.lang.String` **`CUSTOMER_FAX`**

Customer fax number.

CUSTOMER_YTD_PTS

`public static final java.lang.String` **`CUSTOMER_YTD_PTS`**

Customer loyalty YTD points.

CUSTOMER_YTD_SAVED

`public static final java.lang.String` **`CUSTOMER_YTD_SAVED`**

Customer loyalty YTD savings.

com.ibm.retail.AEF.data

Interface DiscountProperties

All Superinterfaces:
LineItemProperties, POSDataProperties

public interface **DiscountProperties**
extends LineItemProperties

DiscountProperties contains the data provider property names for discount attributes. These properties are communicated to client listeners through the `DiscountListener`interface. Individual property values for the current discount are also accessible through the `POSDataProvider`interface.

See Also:
com.ibm.retail.AEF.event.DiscountListener DiscountListener,

com.ibm.retail.AEF.data.POSDataProvider#getPropertyValue getPropertyValue

Field Summary

static java.lang.String	`APPLIES_TO` Indicates what the discount applies to within a transaction.
static java.lang.String	`CATEGORY` Category identifier.
static java.lang.String	`DISCOUNT_AMOUNT` Amount of the discount.
static java.lang.String	`DISCOUNT_METHOD` Discount method ("percent" or "allowance")
static java.lang.String	`DISCOUNT_RATE` Value of discount percentage-off.
static java.lang.String	`DISCOUNT_REASON` Discount reason (application specific).
static java.lang.String	`DISCOUNT_REDUCES_TAX` Indicates whether the discount reduces the tax balance due.
static java.lang.String	`DISCOUNT_TYPE` Discount type (application specific).
static java.lang.String	`IS_TRANSACTION_DISCOUNT` Indicates whether the discount applies to the entire transaction.
static java.lang.String	`IS_VOIDED` Indicates whether the discount was voided.

(continued from last page)

Fields

CATEGORY

`public static final java.lang.String` **`CATEGORY`**

Category identifier.

APPLIES_TO

`public static final java.lang.String` **`APPLIES_TO`**

Indicates what the discount applies to within a transaction. Values include "next", "previous", or a line number within the transaction.

DISCOUNT_METHOD

`public static final java.lang.String` **`DISCOUNT_METHOD`**

Discount method ("percent" or "allowance")

DISCOUNT_AMOUNT

`public static final java.lang.String` **`DISCOUNT_AMOUNT`**

Amount of the discount.

DISCOUNT_RATE

`public static final java.lang.String` **`DISCOUNT_RATE`**

Value of discount percentage-off.

DISCOUNT_REDUCES_TAX

`public static final java.lang.String` **`DISCOUNT_REDUCES_TAX`**

Indicates whether the discount reduces the tax balance due.

IS_TRANSACTION_DISCOUNT

`public static final java.lang.String` **`IS_TRANSACTION_DISCOUNT`**

Indicates whether the discount applies to the entire transaction.

IS_VOIDED

`public static final java.lang.String` **`IS_VOIDED`**

Indicates whether the discount was voided.

DISCOUNT_TYPE

`public static final java.lang.String` **`DISCOUNT_TYPE`**

Discount type (application specific).

DISCOUNT_REASON

`public static final java.lang.String` **`DISCOUNT_REASON`**

Discount reason (application specific).

com.ibm.retail.AEF.data

Interface FunctionCodeDefinition

public interface **FunctionCodeDefinition**
extends java.io.Serializable

A `FunctionCodeDefinition`is a simple data object to contain information about a function code used by a POS Sales application. A function code is typically associated with a key on the POS keyboard and is used to indicate a specific function (e.g. TOTAL) or to initiate input (e.g. ENTER or CLEAR).

Method Summary

int	`getDataLengthMaximum()` Returns the maximum length for the data using information in the input state table
int	`getDataLengthMinimum()` Returns the minimum length for the data using information in the input state table
int	`getDataValueMaximum()` Returns the maximum value for the data using information in the input state table
int	`getDataValueMinimum()` Returns the minimum value for the data using information in the input state table
int	`getHighRange()` If this function code defines a range of values, gets the high end value.
int	`getId()` Returns the function code identifier.
int	`getLowRange()` If this function code defines a range of values, gets the low end value.
int	`getStateID()` Returns id of POS State this function code belongs to, or zero if none.
boolean	`hasDataLengthSet()` Indicates if data length is available in the table.
boolean	`hasDataValueSet()` Indicates if data value is set in the table.
boolean	`isClearKey()` Indicates if this function code is a clear key.
boolean	`isDataAllowed()` Indicates if data is allowed for this function code.
boolean	`isDataDisplayed()` Indicates if the data for this function code should be displayed on the 2x20 display.

boolean	`isDataOptional()` Indicates if data is optional for this function code.
boolean	`isDataPrecedesFunctionCode()` Indicates if data should precede this function code.
boolean	`isDataRequired()` Indicates if data is required for this function code.
boolean	`isEligibleForSave()` Indicates whether this function code is eligible to be saved during processing.
boolean	`isKeyedLabelMayPrecede()` Indicates if a keyed label may precede this function code.
boolean	`isKeyRange()` Indicates if this function code is a range of values.
boolean	`isManagersKeyRequired()` Indicates if a manager's key is required.
boolean	`isMotorKey()` Indicates if the function key is a motor key.
void	`setClearKey(boolean value)` Set Is this function code a clear key.
void	`setDataAllowed(boolean value)` Set Is data allowed for this function code
void	`setDataDisplayed(boolean value)` Set if the data for this function code should be displayed on the 2x20 display.
void	`setDataLengthMaximum(int value)` Sets the maximum length for the data using information in the input state table.
void	`setDataLengthMinimum(int value)` Sets the minimum length for the data using information in the input state table.
void	`setDataLengthSet(boolean value)` Set if data length is available in the table.
void	`setDataOptional(boolean value)` Set Is data optional for this function code
void	`setDataPrecedesFunctionCode(boolean value)` Set true if data should precede this function code.
void	`setDataRequired(boolean value)` Set Is data required for this function code.
void	`setDataValueMaximum(int value)` Sets the maximum value for the data using information in the input state table.

void	`setDataValueMinimum(int value)` Sets the minimum value for the data using information in the input state table.
void	`setDataValueSet(boolean value)` Set if data value range is set in the table.
void	`setEligibleForSave(boolean value)` Set whether this function code is eligible to be saved during processing.
void	`setHighRange(int value)` If this function code defines a range of values, sets the high end value.
void	`setId(int value)` Sets this function codes identifier.
void	`setKeyedLabelMayPrecede(boolean value)` Set true if a keyed label may precede this function code.
void	`setKeyRange(boolean value)` Set Is this function code a range of values.
void	`setLowRange(int value)` If this function code defines a range of values, sets the low end value.
void	`setManagersKeyRequired(boolean value)` Set if a manager's key is required.
void	`setMotorKey(boolean value)` Set Is the function key a motor key, pressing a motor key sends the input sequence to the application.
void	`setStateID(int value)` Sets the id of state this function code belongs to, or zero if none.

Methods

getId

`public int **getId**()`

Returns the function code identifier.

Returns:

identifier

See Also:

FunctionCodeDefinition#isKeyRange

isDataRequired

`public boolean **isDataRequired**()`

(continued from last page)

Indicates if data is required for this function code.

Returns:

true if data is required

isDataAllowed

`public boolean isDataAllowed()`

Indicates if data is allowed for this function code.

Returns:

true if data is allowed

isDataOptional

`public boolean isDataOptional()`

Indicates if data is optional for this function code.

Returns:

true if data is optional

getStateID

`public int getStateID()`

Returns id of POS State this function code belongs to, or zero if none.

Returns:

int

See Also:

com.ibm.retail.AEF.data.StateDefinition StateDefinition

isMotorKey

`public boolean isMotorKey()`

Indicates if the function key is a motor key. Pressing a motor key sends the input sequence to the application.

Returns:

true if this function code is a motor key

isClearKey

`public boolean isClearKey()`

Indicates if this function code is a clear key.

Returns:

true if this function code is the clear key

(continued from last page)

isKeyedLabelMayPrecede

`public boolean` **`isKeyedLabelMayPrecede()`**

Indicates if a keyed label may precede this function code.

Returns:

true if a keyed label may precede, false otherwise

isDataPrecedesFunctionCode

`public boolean` **`isDataPrecedesFunctionCode()`**

Indicates if data should precede this function code.

Returns:

true if data should precede this function code

isKeyRange

`public boolean` **`isKeyRange()`**

Indicates if this function code is a range of values.

Returns:

returns true if this is a range of keys

getLowRange

`public int` **`getLowRange()`**

If this function code defines a range of values, gets the low end value.

Returns:

range value

See Also:

FunctionCodeDefinition#isKeyRange

getHighRange

`public int` **`getHighRange()`**

If this function code defines a range of values, gets the high end value.

Returns:

range value

See Also:

FunctionCodeDefinition#isKeyRange

isManagersKeyRequired

`public boolean` **`isManagersKeyRequired()`**

Indicates if a manager's key is required.

(continued from last page)

Returns:

true if a manager's key is required for this function code

getDataLengthMaximum

`public int` **`getDataLengthMaximum()`**

Returns the maximum length for the data using information in the input state table

Returns:

maximum data length

getDataLengthMinimum

`public int` **`getDataLengthMinimum()`**

Returns the minimum length for the data using information in the input state table

Returns:

minimum data length

getDataValueMaximum

`public int` **`getDataValueMaximum()`**

Returns the maximum value for the data using information in the input state table

Returns:

maximum data value

getDataValueMinimum

`public int` **`getDataValueMinimum()`**

Returns the minimum value for the data using information in the input state table

Returns:

minimum data value

hasDataValueSet

`public boolean` **`hasDataValueSet()`**

Indicates if data value is set in the table.

Returns:

true if a data value range is set in the table

hasDataLengthSet

`public boolean` **`hasDataLengthSet()`**

Indicates if data length is available in the table.

Returns:

(continued from last page)

true if a data length range is set in the table

isDataDisplayed

`public boolean **isDataDisplayed**()`

Indicates if the data for this function code should be displayed on the 2x20 display.

Returns:

true if data is displayed

isEligibleForSave

`public boolean **isEligibleForSave**()`

Indicates whether this function code is eligible to be saved during processing.

Returns:

true if eligible for save

setId

`public void **setId**(int value)`

Sets this function codes identifier.

Parameters:

`value` -
function code identifier

setDataRequired

`public void **setDataRequired**(boolean value)`

Set Is data required for this function code.

Parameters:

`value` -
the boolean to be set

setDataAllowed

`public void **setDataAllowed**(boolean value)`

Set Is data allowed for this function code

Parameters:

`value` -
the boolean to be set

setDataOptional

`public void **setDataOptional**(boolean value)`

Set Is data optional for this function code

Parameters:

`true` -
if data is optional

(continued from last page)

setStateID

`public void **setStateID**(int value)`

Sets the id of state this function code belongs to, or zero if none.

Parameters:
`value -`
state ID

setMotorKey

`public void **setMotorKey**(boolean value)`

Set Is the function key a motor key, pressing a motor key sends the input sequence to the application.

Parameters:
`value -`
the boolean to be set

setClearKey

`public void **setClearKey**(boolean value)`

Set Is this function code a clear key.

Parameters:
`value -`
the boolean to be set

setKeyedLabelMayPrecede

`public void **setKeyedLabelMayPrecede**(boolean value)`

Set true if a keyed label may precede this function code.

Parameters:
`value -`
the boolean to be set

setDataPrecedesFunctionCode

`public void **setDataPrecedesFunctionCode**(boolean value)`

Set true if data should precede this function code.

Parameters:
`value -`
the boolean to be set

setKeyRange

`public void **setKeyRange**(boolean value)`

Set Is this function code a range of values.

Parameters:
`value -`
the boolean to be set

setLowRange

`public void **setLowRange**(int value)`

If this function code defines a range of values, sets the low end value.

(continued from last page)

Parameters:
`value -`
low range value

setHighRange

`public void **setHighRange**(int value)`

If this function code defines a range of values, sets the high end value.

Parameters:
`value -`
high range value

setManagersKeyRequired

`public void **setManagersKeyRequired**(boolean value)`

Set if a manager's key is required.

Parameters:
`value -`
the boolean to be set

setDataLengthMaximum

`public void **setDataLengthMaximum**(int value)`

Sets the maximum length for the data using information in the input state table.

Parameters:
`value -`
maximum data length

setDataLengthMinimum

`public void **setDataLengthMinimum**(int value)`

Sets the minimum length for the data using information in the input state table.

Parameters:
`value -`
minimum data length

setDataValueMaximum

`public void **setDataValueMaximum**(int value)`

Sets the maximum value for the data using information in the input state table.

Parameters:
`value -`
maximum data value

setDataValueMinimum

`public void **setDataValueMinimum**(int value)`

Sets the minimum value for the data using information in the input state table.

Parameters:
`value -`
minimum data value

setDataValueSet

`public void **setDataValueSet**(boolean value)`

Set if data value range is set in the table.

Parameters:

`value -`
the boolean to be set

setDataLengthSet

`public void **setDataLengthSet**(boolean value)`

Set if data length is available in the table.

Parameters:

`value -`
the boolean to be set

setDataDisplayed

`public void **setDataDisplayed**(boolean value)`

Set if the data for this function code should be displayed on the 2x20 display.

Parameters:

`value -`
the boolean to be set

setEligibleForSave

`public void **setEligibleForSave**(boolean value)`

Set whether this function code is eligible to be saved during processing.

Parameters:

`value -`
the boolean to be set

com.ibm.retail.AEF.data

Interface ItemSalesProperties

All Superinterfaces:
LineItemProperties, POSDataProperties

public interface **ItemSalesProperties**
extends LineItemProperties

`ItemSalesProperties`
contains the data provider property names for line item sales properties. These properties are communicated to client listeners through the `ItemSalesListener`interface. Individual property values for the current item sold are also accessible through the `POSDataProvider`interface.

See Also:
com.ibm.retail.AEF.event.ItemSalesListener ItemSalesListener,

com.ibm.retail.AEF.data.POSDataProvider#getPropertyValue getPropertyValue

Field Summary

static java.lang.String	`CATEGORY` Category identifier
static java.lang.String	`DEAL_PRICE` Item deal price
static java.lang.String	`DEAL_QUANTITY` Item deal quantity
static java.lang.String	`ENTERED_PRICE_USED` Entered Price Used (price required or price override)
static java.lang.String	`EXTENDED_PRICE` Extended price
static java.lang.String	`FOOD_STAMP_ELIGIBLE` Is this item eligible for food stamps?
static java.lang.String	`IS_DEPOSIT` Item is a Deposit
static java.lang.String	`IS_REFUNDED` Item was Refunded
static java.lang.String	`IS_RETURN` Is this item sale a return?
static java.lang.String	`IS_VOIDED` Item Voided

static java.lang.String	ITEM_MODIFIER Item modifier (provides addtional item type information)
static java.lang.String	ITEM_REPEAT_ALLOWED Is this item sale repeatable using a repeat key?
static java.lang.String	ITEM_TAXABLE Item is Taxable
static java.lang.String	LINKED_ITEM_ID Linked item identifier
static java.lang.String	LINKED_ITEM_ID_QUALIFIER Linked item identifier type of ID
static java.lang.String	MULTI_PRICING_GROUP Item multi pricing group id
static java.lang.String	ORIGINAL_SALESPERSON If the item was returned who was the original salesperson.
static java.lang.String	PRICING_METHOD Item pricing method
static java.lang.String	REDUCED_PRICE Item reduced price
static java.lang.String	REGULAR_PRICE Regular price per unit
static java.lang.String	REQUIRED_AGE Required Age for age restricted items
static java.lang.String	RESTRICTED_PERIODS Restricted periods
static java.lang.String	RETURN_REASON Reason for the return
static java.lang.String	UNIT_PRICE Sales price per unit
static java.lang.String	WEIGHT Weight sold if sold by weight
static java.lang.String	WIC_ELIGIBLE Is this item eligible for WIC?

Fields

(continued from last page)

CATEGORY

`public static final java.lang.String` **`CATEGORY`**

Category identifier

ITEM_MODIFIER

`public static final java.lang.String` **`ITEM_MODIFIER`**

Item modifier (provides addtional item type information)

UNIT_PRICE

`public static final java.lang.String` **`UNIT_PRICE`**

Sales price per unit

REGULAR_PRICE

`public static final java.lang.String` **`REGULAR_PRICE`**

Regular price per unit

WEIGHT

`public static final java.lang.String` **`WEIGHT`**

Weight sold if sold by weight

EXTENDED_PRICE

`public static final java.lang.String` **`EXTENDED_PRICE`**

Extended price

ITEM_REPEAT_ALLOWED

`public static final java.lang.String` **`ITEM_REPEAT_ALLOWED`**

Is this item sale repeatable using a repeat key?

IS_RETURN

`public static final java.lang.String` **`IS_RETURN`**

Is this item sale a return?

RETURN_REASON

`public static final java.lang.String` **`RETURN_REASON`**

Reason for the return

REQUIRED_AGE

`public static final java.lang.String` **`REQUIRED_AGE`**

Required Age for age restricted items

RESTRICTED_PERIODS

`public static final java.lang.String` **`RESTRICTED_PERIODS`**

(continued from last page)

Restricted periods

FOOD_STAMP_ELIGIBLE

`public static final java.lang.String` **`FOOD_STAMP_ELIGIBLE`**

Is this item eligible for food stamps?

WIC_ELIGIBLE

`public static final java.lang.String` **`WIC_ELIGIBLE`**

Is this item eligible for WIC?

LINKED_ITEM_ID

`public static final java.lang.String` **`LINKED_ITEM_ID`**

Linked item identifier

LINKED_ITEM_ID_QUALIFIER

`public static final java.lang.String` **`LINKED_ITEM_ID_QUALIFIER`**

Linked item identifier type of ID

DEAL_QUANTITY

`public static final java.lang.String` **`DEAL_QUANTITY`**

Item deal quantity

DEAL_PRICE

`public static final java.lang.String` **`DEAL_PRICE`**

Item deal price

REDUCED_PRICE

`public static final java.lang.String` **`REDUCED_PRICE`**

Item reduced price

MULTI_PRICING_GROUP

`public static final java.lang.String` **`MULTI_PRICING_GROUP`**

Item multi pricing group id

PRICING_METHOD

`public static final java.lang.String` **`PRICING_METHOD`**

Item pricing method

IS_VOIDED

`public static final java.lang.String` **`IS_VOIDED`**

Item Voided

(continued from last page)

ITEM_TAXABLE

`public static final java.lang.String` **`ITEM_TAXABLE`**

Item is Taxable

IS_DEPOSIT

`public static final java.lang.String` **`IS_DEPOSIT`**

Item is a Deposit

IS_REFUNDED

`public static final java.lang.String` **`IS_REFUNDED`**

Item was Refunded

ENTERED_PRICE_USED

`public static final java.lang.String` **`ENTERED_PRICE_USED`**

Entered Price Used (price required or price override)

ORIGINAL_SALESPERSON

`public static final java.lang.String` **`ORIGINAL_SALESPERSON`**

If the item was returned who was the original salesperson.

com.ibm.retail.AEF.data

Interface LineItemProperties

All Superinterfaces:
POSDataProperties

All Subinterfaces:
TenderProperties, PointsProperties, ItemSalesProperties, DiscountProperties, CouponProperties

---

public interface **LineItemProperties**
extends POSDataProperties

`LineItemProperties`
contains the data provider property names for line item attributes. It is the base class for common line item properties.

Field Summary

static java.lang.String	`DESCRIPTION` Description of item display by POS.
static java.lang.String	`IS_DEPOSIT` Indicates if ttem is a deposit.
static java.lang.String	`IS_REFUNDED` Indicates if ttem was refunded.
static java.lang.String	`IS_VOIDED` Indicates if item is voided.
static java.lang.String	`ITEM_ID` Item identifier (e.
static java.lang.String	`ITEM_ID_QUALIFIER` ItemID qualifier (type of number)
static java.lang.String	`ITEM_MODIFIER` Item modifier (provides addtional item type information)
static java.lang.String	`PRINT_LINES` Collection of print lines (without printer attributes).
static java.lang.String	`QUANTITY` Quantity sold if sold by eaches.
static java.lang.String	`RAW_PRINT_LINES` Collection of print lines (with printer attributes).

Fields

(continued from last page)

ITEM_ID

`public static final java.lang.String` **`ITEM_ID`**

Item identifier (e.g., UPC)

ITEM_ID_QUALIFIER

`public static final java.lang.String` **`ITEM_ID_QUALIFIER`**

ItemID qualifier (type of number)

ITEM_MODIFIER

`public static final java.lang.String` **`ITEM_MODIFIER`**

Item modifier (provides addtional item type information)

QUANTITY

`public static final java.lang.String` **`QUANTITY`**

Quantity sold if sold by eaches.

DESCRIPTION

`public static final java.lang.String` **`DESCRIPTION`**

Description of item display by POS.

IS_VOIDED

`public static final java.lang.String` **`IS_VOIDED`**

Indicates if item is voided.

IS_DEPOSIT

`public static final java.lang.String` **`IS_DEPOSIT`**

Indicates if ttem is a deposit.

IS_REFUNDED

`public static final java.lang.String` **`IS_REFUNDED`**

Indicates if ttem was refunded.

PRINT_LINES

`public static final java.lang.String` **`PRINT_LINES`**

Collection of print lines (without printer attributes).

RAW_PRINT_LINES

`public static final java.lang.String` **`RAW_PRINT_LINES`**

Collection of print lines (with printer attributes).

com.ibm.retail.AEF.data

Interface OperatorAuthorizationProperties

All Superinterfaces:
POSDataProperties

public interface **OperatorAuthorizationProperties**
extends POSDataProperties

OperatorAuthorizationProperties contains the data provider property names for operator properties and authorization flags. These properties are communicated to client listeners through the `OperatorListener`interface. Individual property values for the current operator are also accessible through the `POSDataProvider`interface.

See Also:
com.ibm.retail.AEF.event.OperatorListener OperatorListener, com.ibm.retail.AEF.data.POSDataProvider#getPropertyValue getPropertyValue

Field Summary

static java.lang.String	`ALLOWANCE_ALLOWED` Allowance Allowed identifier
static java.lang.String	`CASH_COUNT_ALLOWED` Cash Count Allowed identifier
static java.lang.String	`CASH_DOC_ALLOWED` Cash Document Transaction Allowed identifier
static java.lang.String	`CASH_SPEC_ALLOWED` Cash Special Transaction Allowed identifier
static java.lang.String	`CASH_TRANSACTION_ALLOWED` Cash Transaction Allowed identifier
static java.lang.String	`CATEGORY` Category identifier
static java.lang.String	`CHARGE_TYPE_6_ALLOWED` Charge Type 6 Transaction Allowed identifier
static java.lang.String	`CHARGE_TYPE_7_ALLOWED` Charge Type 7 Transaction Allowed identifier
static java.lang.String	`CHARGE_TYPE_8_ALLOWED` Charge Type 8 Transaction Allowed identifier
static java.lang.String	`CHARGE_TYPE_9_ALLOWED` Charge Type 9 Transaction Allowed identifier
static java.lang.String	`CHECKOUT_TRANSACTION_ALLOWED` Checkout Transaction Allowed identifier

static java.lang.String	`COD_ALLOWED` COD Transaction Allowed identifier
static java.lang.String	`DELAYED_IR_CHANGES_ALLOWED` Delayed IR Changes Allowed identifier
static java.lang.String	`DELAYED_PRICE_CHANGE_ALLOWED` Delayed Item Price Change Allowed identifier
static java.lang.String	`DEPT_TOTALS_REPORT_ALLOWED` Department Totals Report Allowed identifier
static java.lang.String	`DISCOUNTS_ALLOWED` Discounts Allowed identifier
static java.lang.String	`FRONT_END_CASHIER_ALLOWED` Front End Cashier Allowed identifier
static java.lang.String	`IMMEDIATE_IR_CHANGES_ALLOWED` Immediate IR Changes Allowed identifier
static java.lang.String	`ITEM_PRICE_CHANGE_ALLOWED` Item Price Change Allowed identifier
static java.lang.String	`LAYAWAY_ALLOWED` Layaway Transaction Allowed identifier
static java.lang.String	`LOAN_ALLOWED` Loan Transaction Allowed identifier
static java.lang.String	`MANAGERS_PROCEDURES_ALLOWED` Manager's Procedures Allowed identifier
static java.lang.String	`MISC_ITEM_PAYOUTS_ALLOWED` Misc.
static java.lang.String	`MORE_THAN_PRICE_CHANGES_ALLOWED` More than price changes Allowed identifier
static java.lang.String	`NO_SALE_ALLOWED` User No-sale Open Cash Drawer Allowed identifier
static java.lang.String	`NO_SALE_OPEN_CASH_DRAWER_ALLOWED` No-sale Open Cash Drawer Allowed identifier
static java.lang.String	`NO_SALE_PRICE_VERIFY_ALLOWED` No-sale Price Verify Allowed identifier
static java.lang.String	`NO_SALE_TENDER_REMOVAL_ALLOWED` No-sale Tender Removal Allowed identifier
static java.lang.String	`NO_SALE_TENDER_VERIFY_ALLOWED` No-sale Tender Verify Allowed identifier

static java.lang.String	NO_SALE_TILL_EXCHANGE_ALLOWED No-sale Till Exchange Allowed identifier
static java.lang.String	NO_SALE_TILL_REPORT_ALLOWED No-sale Till Report Allowed identifier
static java.lang.String	OBTAIN_DEPT_TOTALS_ALLOWED Department Totals Report Allowed identifier
static java.lang.String	OBTAIN_ITEM_MOVEMENT_ALLOWED Obtain Item Movement Allowed identifier
static java.lang.String	OPERATOR_AUTHORIZATION Operator Authorization object identifier
static java.lang.String	OPERATOR_ID Operator ID identifier
static java.lang.String	OPERATOR_NAME Operator Name identifier
static java.lang.String	OPERATOR_TRAINING_ALLOWED Operator Training Allowed identifier
static java.lang.String	PASSWORD_REQUIRED Password Required identifier
static java.lang.String	PICKUP_TRANSACTION_ALLOWED Pickup Transaction Allowed identifier
static java.lang.String	PRICE_VERIFICATION_ALLOWED Price Verification Allowed identifier
static java.lang.String	QUERY_EXCHANGE_RATE_ALLOWED Query Exchange Rate Allowed identifier
static java.lang.String	REENTRY_OFFLINE_SALES_ALLOWED Reentry of Offline Sales Allowed identifier
static java.lang.String	REFUNDS_ALLOWED Refunds Allowed identifier
static java.lang.String	REGISTER_READOUT_ALLOWED Register Readout Transaction Allowed identifier
static java.lang.String	REGISTER_RESET_ALLOWED Register Reset Allowed identifier
static java.lang.String	REPRINT_TENDER_RECEIPT_ALLOWED Reprint Tender Receipt Allowed identifier
static java.lang.String	RETURN_ITEM_ALLOWED Return Item Allowed identifier

static java.lang.String	RETURNS_ALLOWED Returns Allowed identifier
static java.lang.String	SET_TRANSACTION_NUMBER_ALLOWED Set Transaction Number Allowed identifier
static java.lang.String	SUSPEND_TRANSACTION_ALLOWED Suspend Transaction Allowed identifier
static java.lang.String	TENDER_CASHING_ALLOWED Tender Cashing Allowed identifier
static java.lang.String	TENDER_COUNT_ALLOWED Tender Count Allowed identifier
static java.lang.String	TENDER_EXCHANGE_ALLOWED Tender Exchange Allowed identifier
static java.lang.String	TENDER_LISTING_ALLOWED Tender Listing Allowed identifier
static java.lang.String	TENDER_REMOVAL_ALLOWED Tender Removal Allowed identifier
static java.lang.String	TERMINAL_MONITOR_ALLOWED Terminal Monitor Allowed identifier
static java.lang.String	TERMINAL_TRANSFER_ALLOWED Terminal Transfer Allowed identifier
static java.lang.String	TPL_TRANSACTION_ALLOWED TPL Transaction Allowed identifier
static java.lang.String	USER_FUNCTION_1_ALLOWED User Function 1 Allowed identifier
static java.lang.String	USER_NON_SALES_1_ALLOWED User Non-sales 1 Allowed identifier
static java.lang.String	USER_NON_SALES_2_ALLOWED User Non-sales 2 Allowed identifier
static java.lang.String	VOID_PREVIOUS_BY_LINE_ALLOWED Void Previous By Line Allowed identifier
static java.lang.String	VOID_PREVIOUS_TRANSACTION_ALLOWED Void Transaction Allowed identifier
static java.lang.String	VOID_TRANSACTION_ALLOWED Void Transaction Allowed identifier
static java.lang.String	WIC_TRANSACTION_ALLOWED WIC Transaction Allowed identifier

static java.lang.String	WITHDRAWALS_TRANSACTION_ALLOWED Withdrawals Transaction Allowed identifier

Fields

CATEGORY

`public static final java.lang.String` **`CATEGORY`**

Category identifier

OPERATOR_NAME

`public static final java.lang.String` **`OPERATOR_NAME`**

Operator Name identifier

OPERATOR_ID

`public static final java.lang.String` **`OPERATOR_ID`**

Operator ID identifier

OPERATOR_AUTHORIZATION

`public static final java.lang.String` **`OPERATOR_AUTHORIZATION`**

Operator Authorization object identifier

ITEM_PRICE_CHANGE_ALLOWED

`public static final java.lang.String` **`ITEM_PRICE_CHANGE_ALLOWED`**

Item Price Change Allowed identifier

LAYAWAY_ALLOWED

`public static final java.lang.String` **`LAYAWAY_ALLOWED`**

Layaway Transaction Allowed identifier

LOAN_ALLOWED

`public static final java.lang.String` **`LOAN_ALLOWED`**

Loan Transaction Allowed identifier

OBTAIN_ITEM_MOVEMENT_ALLOWED

`public static final java.lang.String` **`OBTAIN_ITEM_MOVEMENT_ALLOWED`**

Obtain Item Movement Allowed identifier

OBTAIN_DEPT_TOTALS_ALLOWED

`public static final java.lang.String` **`OBTAIN_DEPT_TOTALS_ALLOWED`**

Department Totals Report Allowed identifier

(continued from last page)

OPERATOR_TRAINING_ALLOWED

`public static final java.lang.String` **`OPERATOR_TRAINING_ALLOWED`**

Operator Training Allowed identifier

REENTRY_OFFLINE_SALES_ALLOWED

`public static final java.lang.String` **`REENTRY_OFFLINE_SALES_ALLOWED`**

Reentry of Offline Sales Allowed identifier

REGISTER_READOUT_ALLOWED

`public static final java.lang.String` **`REGISTER_READOUT_ALLOWED`**

Register Readout Transaction Allowed identifier

RETURNS_ALLOWED

`public static final java.lang.String` **`RETURNS_ALLOWED`**

Returns Allowed identifier

DISCOUNTS_ALLOWED

`public static final java.lang.String` **`DISCOUNTS_ALLOWED`**

Discounts Allowed identifier

VOID_TRANSACTION_ALLOWED

`public static final java.lang.String` **`VOID_TRANSACTION_ALLOWED`**

Void Transaction Allowed identifier

PRICE_VERIFICATION_ALLOWED

`public static final java.lang.String` **`PRICE_VERIFICATION_ALLOWED`**

Price Verification Allowed identifier

TERMINAL_MONITOR_ALLOWED

`public static final java.lang.String` **`TERMINAL_MONITOR_ALLOWED`**

Terminal Monitor Allowed identifier

TENDER_LISTING_ALLOWED

`public static final java.lang.String` **`TENDER_LISTING_ALLOWED`**

Tender Listing Allowed identifier

PASSWORD_REQUIRED

`public static final java.lang.String` **`PASSWORD_REQUIRED`**

Password Required identifier

ALLOWANCE_ALLOWED

`public static final java.lang.String` **`ALLOWANCE_ALLOWED`**

Allowance Allowed identifier

CASH_TRANSACTION_ALLOWED

`public static final java.lang.String` **`CASH_TRANSACTION_ALLOWED`**

Cash Transaction Allowed identifier

CASH_DOC_ALLOWED

`public static final java.lang.String` **`CASH_DOC_ALLOWED`**

Cash Document Transaction Allowed identifier

CASH_SPEC_ALLOWED

`public static final java.lang.String` **`CASH_SPEC_ALLOWED`**

Cash Special Transaction Allowed identifier

CHARGE_TYPE_6_ALLOWED

`public static final java.lang.String` **`CHARGE_TYPE_6_ALLOWED`**

Charge Type 6 Transaction Allowed identifier

CHARGE_TYPE_7_ALLOWED

`public static final java.lang.String` **`CHARGE_TYPE_7_ALLOWED`**

Charge Type 7 Transaction Allowed identifier

CHARGE_TYPE_8_ALLOWED

`public static final java.lang.String` **`CHARGE_TYPE_8_ALLOWED`**

Charge Type 8 Transaction Allowed identifier

CHARGE_TYPE_9_ALLOWED

`public static final java.lang.String` **`CHARGE_TYPE_9_ALLOWED`**

Charge Type 9 Transaction Allowed identifier

TENDER_REMOVAL_ALLOWED

`public static final java.lang.String` **`TENDER_REMOVAL_ALLOWED`**

Tender Removal Allowed identifier

COD_ALLOWED

`public static final java.lang.String` **`COD_ALLOWED`**

COD Transaction Allowed identifier

DELAYED_PRICE_CHANGE_ALLOWED

`public static final java.lang.String` **`DELAYED_PRICE_CHANGE_ALLOWED`**

Delayed Item Price Change Allowed identifier

(continued from last page)

REGISTER_RESET_ALLOWED

`public static final java.lang.String` **`REGISTER_RESET_ALLOWED`**

Register Reset Allowed identifier

SET_TRANSACTION_NUMBER_ALLOWED

`public static final java.lang.String` **`SET_TRANSACTION_NUMBER_ALLOWED`**

Set Transaction Number Allowed identifier

SUSPEND_TRANSACTION_ALLOWED

`public static final java.lang.String` **`SUSPEND_TRANSACTION_ALLOWED`**

Suspend Transaction Allowed identifier

TPL_TRANSACTION_ALLOWED

`public static final java.lang.String` **`TPL_TRANSACTION_ALLOWED`**

TPL Transaction Allowed identifier

VOID_PREVIOUS_BY_LINE_ALLOWED

`public static final java.lang.String` **`VOID_PREVIOUS_BY_LINE_ALLOWED`**

Void Previous By Line Allowed identifier

VOID_PREVIOUS_TRANSACTION_ALLOWED

`public static final java.lang.String` **`VOID_PREVIOUS_TRANSACTION_ALLOWED`**

Void Transaction Allowed identifier

QUERY_EXCHANGE_RATE_ALLOWED

`public static final java.lang.String` **`QUERY_EXCHANGE_RATE_ALLOWED`**

Query Exchange Rate Allowed identifier

NO_SALE_ALLOWED

`public static final java.lang.String` **`NO_SALE_ALLOWED`**

User No-sale Open Cash Drawer Allowed identifier

WITHDRAWALS_TRANSACTION_ALLOWED

`public static final java.lang.String` **`WITHDRAWALS_TRANSACTION_ALLOWED`**

Withdrawals Transaction Allowed identifier

CASH_COUNT_ALLOWED

`public static final java.lang.String` **`CASH_COUNT_ALLOWED`**

Cash Count Allowed identifier

CHECKOUT_TRANSACTION_ALLOWED

`public static final java.lang.String` **`CHECKOUT_TRANSACTION_ALLOWED`**

Checkout Transaction Allowed identifier

TENDER_CASHING_ALLOWED

`public static final java.lang.String` **`TENDER_CASHING_ALLOWED`**

Tender Cashing Allowed identifier

TENDER_EXCHANGE_ALLOWED

`public static final java.lang.String` **`TENDER_EXCHANGE_ALLOWED`**

Tender Exchange Allowed identifier

PICKUP_TRANSACTION_ALLOWED

`public static final java.lang.String` **`PICKUP_TRANSACTION_ALLOWED`**

Pickup Transaction Allowed identifier

TERMINAL_TRANSFER_ALLOWED

`public static final java.lang.String` **`TERMINAL_TRANSFER_ALLOWED`**

Terminal Transfer Allowed identifier

TENDER_COUNT_ALLOWED

`public static final java.lang.String` **`TENDER_COUNT_ALLOWED`**

Tender Count Allowed identifier

RETURN_ITEM_ALLOWED

`public static final java.lang.String` **`RETURN_ITEM_ALLOWED`**

Return Item Allowed identifier

WIC_TRANSACTION_ALLOWED

`public static final java.lang.String` **`WIC_TRANSACTION_ALLOWED`**

WIC Transaction Allowed identifier

USER_NON_SALES_1_ALLOWED

`public static final java.lang.String` **`USER_NON_SALES_1_ALLOWED`**

User Non-sales 1 Allowed identifier

USER_NON_SALES_2_ALLOWED

`public static final java.lang.String` **`USER_NON_SALES_2_ALLOWED`**

User Non-sales 2 Allowed identifier

NO_SALE_OPEN_CASH_DRAWER_ALLOWED

`public static final java.lang.String` **`NO_SALE_OPEN_CASH_DRAWER_ALLOWED`**

No-sale Open Cash Drawer Allowed identifier

(continued from last page)

NO_SALE_TENDER_REMOVAL_ALLOWED

`public static final java.lang.String` **`NO_SALE_TENDER_REMOVAL_ALLOWED`**

No-sale Tender Removal Allowed identifier

NO_SALE_TILL_EXCHANGE_ALLOWED

`public static final java.lang.String` **`NO_SALE_TILL_EXCHANGE_ALLOWED`**

No-sale Till Exchange Allowed identifier

NO_SALE_TENDER_VERIFY_ALLOWED

`public static final java.lang.String` **`NO_SALE_TENDER_VERIFY_ALLOWED`**

No-sale Tender Verify Allowed identifier

NO_SALE_TILL_REPORT_ALLOWED

`public static final java.lang.String` **`NO_SALE_TILL_REPORT_ALLOWED`**

No-sale Till Report Allowed identifier

NO_SALE_PRICE_VERIFY_ALLOWED

`public static final java.lang.String` **`NO_SALE_PRICE_VERIFY_ALLOWED`**

No-sale Price Verify Allowed identifier

REFUNDS_ALLOWED

`public static final java.lang.String` **`REFUNDS_ALLOWED`**

Refunds Allowed identifier

MISC_ITEM_PAYOUTS_ALLOWED

`public static final java.lang.String` **`MISC_ITEM_PAYOUTS_ALLOWED`**

Misc. Item Payouts Allowed identifier

IMMEDIATE_IR_CHANGES_ALLOWED

`public static final java.lang.String` **`IMMEDIATE_IR_CHANGES_ALLOWED`**

Immediate IR Changes Allowed identifier

DELAYED_IR_CHANGES_ALLOWED

`public static final java.lang.String` **`DELAYED_IR_CHANGES_ALLOWED`**

Delayed IR Changes Allowed identifier

MORE_THAN_PRICE_CHANGES_ALLOWED

`public static final java.lang.String` **`MORE_THAN_PRICE_CHANGES_ALLOWED`**

More than price changes Allowed identifier

MANAGERS_PROCEDURES_ALLOWED

`public static final java.lang.String` **`MANAGERS_PROCEDURES_ALLOWED`**

Manager's Procedures Allowed identifier

REPRINT_TENDER_RECEIPT_ALLOWED

`public static final java.lang.String` **`REPRINT_TENDER_RECEIPT_ALLOWED`**

Reprint Tender Receipt Allowed identifier

USER_FUNCTION_1_ALLOWED

`public static final java.lang.String` **`USER_FUNCTION_1_ALLOWED`**

User Function 1 Allowed identifier

FRONT_END_CASHIER_ALLOWED

`public static final java.lang.String` **`FRONT_END_CASHIER_ALLOWED`**

Front End Cashier Allowed identifier

DEPT_TOTALS_REPORT_ALLOWED

`public static final java.lang.String` **`DEPT_TOTALS_REPORT_ALLOWED`**

Department Totals Report Allowed identifier

com.ibm.retail.AEF.data

Interface OptionsProperties

All Superinterfaces:
POSDataProperties

public interface **OptionsProperties**
extends POSDataProperties

`OptionsProperties`
contains the data provider property names for terminal and store level options. These properties are communicated to client listeners through the `OptionsListener`interface. Individual property values for the current options are also accessible through the `POSDataProvider`interface.

See Also:
com.ibm.retail.AEF.event.OptionsListener OptionsListener, com.ibm.retail.AEF.data.POSDataProvider#getPropertyValue getPropertyValue

Field Summary

static java.lang.String	`ALTERNATE_TAX_CODE` Alternate Tax codes identifier references a collection of `TaxCode`objects.
static java.lang.String	`CATEGORY` Category identifier
static java.lang.String	`DEPARTMENT_DEFINITION` Department identifier references a collection of `DepartmentDefinition`objects
static java.lang.String	`ITEM_DISCOUNT_REASON_CODE` Item discount reason codes identifier references a collection of `DiscountReasonCode`objects.
static java.lang.String	`MANUAL_TAX_CODE` Manual Tax codes identifier references a collection of `TaxCode`objects.
static java.lang.String	`NO_TAX_CODE` NoTax tax codes identifier references a collection of `TaxCode`objects.
static java.lang.String	`PRICE_OVERRIDE_REASON` Price override codes identifier references a collection of `ReasonCode`objects.
static java.lang.String	`REFUND_REASON` Refund reason codes identifier references a collection of `ReasonCode`objects.
static java.lang.String	`STORE_DEFINITION` Store definition object identifier references a `StoreDefinition`object.
static java.lang.String	`STORE_OPTIONS` Store Options identifier references a `StoreOptions`object which defines the store level settings.

static java.lang.String	TARE_CODE Tare codes identifier references a collection of TareCodeobjects.
static java.lang.String	TENDER_DEFINITION Tender definition identifier references a collection of TenderDefinitionobjects.
static java.lang.String	TERMINAL_OPTIONS Terminal Options identifier references a TerminalOptionsobject which defines the terminal settings.
static java.lang.String	TRANSACTION_DISCOUNT_REASON_CODE Transaction discount reason codes identifier references a collection of TransactionDiscountReasonCodeobjects.
static java.lang.String	VAT_TAX_CODE VAT Tax codes identifier references a collection of TaxCodeobjects.
static java.lang.String	VOID_REASON Void reason codes identifier references a collection of ReasonCodeobjects.

Fields

CATEGORY

`public static final java.lang.String` **`CATEGORY`**

Category identifier

STORE_DEFINITION

`public static final java.lang.String` **`STORE_DEFINITION`**

Store definition object identifier references a StoreDefinitionobject.

See Also:

com.ibm.retail.AEF.event.StoreDefinition

TENDER_DEFINITION

`public static final java.lang.String` **`TENDER_DEFINITION`**

Tender definition identifier references a collection of TenderDefinitionobjects.

See Also:

com.ibm.retail.AEF.event.TenderDefinition

PRICE_OVERRIDE_REASON

`public static final java.lang.String` **`PRICE_OVERRIDE_REASON`**

Price override codes identifier references a collection of ReasonCodeobjects.

See Also:

com.ibm.retail.AEF.event.ReasonCode

REFUND_REASON

`public static final java.lang.String` **`REFUND_REASON`**

Refund reason codes identifier references a collection of `ReasonCode`objects.

See Also:

com.ibm.retail.AEF.event.ReasonCode

VOID_REASON

`public static final java.lang.String` **`VOID_REASON`**

Void reason codes identifier references a collection of `ReasonCode`objects.

See Also:

com.ibm.retail.AEF.event.ReasonCode

TARE_CODE

`public static final java.lang.String` **`TARE_CODE`**

Tare codes identifier references a collection of `TareCode`objects.

See Also:

com.ibm.retail.AEF.event.TareCode

VAT_TAX_CODE

`public static final java.lang.String` **`VAT_TAX_CODE`**

VAT Tax codes identifier references a collection of `TaxCode`objects.

See Also:

com.ibm.retail.AEF.event.TaxCode

ALTERNATE_TAX_CODE

`public static final java.lang.String` **`ALTERNATE_TAX_CODE`**

Alternate Tax codes identifier references a collection of `TaxCode`objects.

See Also:

com.ibm.retail.AEF.event.TaxCode

MANUAL_TAX_CODE

`public static final java.lang.String` **`MANUAL_TAX_CODE`**

Manual Tax codes identifier references a collection of `TaxCode`objects.

See Also:

com.ibm.retail.AEF.event.TaxCode

(continued from last page)

NO_TAX_CODE

`public static final java.lang.String` **`NO_TAX_CODE`**

NoTax tax codes identifier references a collection of `TaxCode`objects.

See Also:

com.ibm.retail.AEF.event.TaxCode

TRANSACTION_DISCOUNT_REASON_CODE

`public static final java.lang.String` **`TRANSACTION_DISCOUNT_REASON_CODE`**

Transaction discount reason codes identifier references a collection of `TransactionDiscountReasonCode`objects.

See Also:

com.ibm.retail.AEF.event.TransactionDiscountReasonCode

ITEM_DISCOUNT_REASON_CODE

`public static final java.lang.String` **`ITEM_DISCOUNT_REASON_CODE`**

Item discount reason codes identifier references a collection of `DiscountReasonCode`objects.

See Also:

com.ibm.retail.AEF.event.DiscountReasonCode

DEPARTMENT_DEFINITION

`public static final java.lang.String` **`DEPARTMENT_DEFINITION`**

Department identifier references a collection of `DepartmentDefinition`objects

See Also:

com.ibm.retail.AEF.event.DepartmentDefinition

TERMINAL_OPTIONS

`public static final java.lang.String` **`TERMINAL_OPTIONS`**

Terminal Options identifier references a `TerminalOptions`object which defines the terminal settings.

See Also:

com.ibm.retail.AEF.event.TerminalOptions

STORE_OPTIONS

`public static final java.lang.String` **`STORE_OPTIONS`**

Store Options identifier references a `StoreOptions`object which defines the store level settings.

See Also:

com.ibm.retail.AEF.event.StoreOptions

com.ibm.retail.AEF.data

Interface PointsProperties

All Superinterfaces:
LineItemProperties, POSDataProperties

public interface **PointsProperties**
extends LineItemProperties

PointsProperties contains the data provider property names for points attributes. These properties are communicated to client listeners through the `PointsListener`interface. Individual property values for the current points are also accessible through the `POSDataProvider`interface.

See Also:
com.ibm.retail.AEF.event.PointsListener PointsListener, com.ibm.retail.AEF.data.POSDataProvider#getPropertyValue getPropertyValue

Field Summary

static java.lang.String	`ADDITIONAL_POINTS_TOTALS` Additional Points totals
static java.lang.String	`CATEGORY` Category identifier
static java.lang.String	`ID` Points identifier
static java.lang.String	`ID_QUALIFIER` Points ID qualifier (type of number)
static java.lang.String	`POINTS_REDEEMED` Are points redeemed?
static java.lang.String	`POINTS_VOIDED` Are points voided?
static java.lang.String	`TOTAL` Total points
static java.lang.String	`TYPE` Type of points

Fields

CATEGORY

`public static final java.lang.String` **`CATEGORY`**

Category identifier

ID

`public static final java.lang.String` **`ID`**

Points identifier

ID_QUALIFIER

`public static final java.lang.String` **`ID_QUALIFIER`**

Points ID qualifier (type of number)

TYPE

`public static final java.lang.String` **`TYPE`**

Type of points

TOTAL

`public static final java.lang.String` **`TOTAL`**

Total points

POINTS_REDEEMED

`public static final java.lang.String` **`POINTS_REDEEMED`**

Are points redeemed?

POINTS_VOIDED

`public static final java.lang.String` **`POINTS_VOIDED`**

Are points voided?

ADDITIONAL_POINTS_TOTALS

`public static final java.lang.String` **`ADDITIONAL_POINTS_TOTALS`**

Additional Points totals

com.ibm.retail.AEF.data

Interface POSDataProperties

All Subinterfaces:

TransactionTotalsProperties, OptionsProperties, OperatorAuthorizationProperties, LineItemProperties, TenderProperties, PointsProperties, ItemSalesProperties, DiscountProperties, CouponProperties, CustomerProperties

public interface **POSDataProperties**

POSDataProperties contains the base data provider property names.

Field Summary

static java.lang.String	ELEMENT_NAME element-type identifier

Fields

ELEMENT_NAME

`public static final java.lang.String` **`ELEMENT_NAME`**

element-type identifier

com.ibm.retail.AEF.data

Interface POSDataProvider

public interface **POSDataProvider**
extends java.rmi.Remote

The `POSDataProvider`provides listener registration to monitor the properties of its associated terminal session. It also maintains and provides access to the state information for the session.

A `POSDataProvider`object is created by an `AEFSession`and is accessed through the `getPOSDataProvider`accessor in the `AEFSession`interface.

The `POSDataProvider`supports a variety of event listener interfaces which can be divided into 2 main categories:

1. `AEFPropertyChangeListeners`are fine-grained property change events which allow a specific, individual property to be observed. Example: transaction total for tax.
2. `POSAppEventListeners`are coarse-grained events which aggregate related data. This allows fewer events but may provide the listener with more data than required. Example: all transaction totals.

State information for the associated session is maintained and is accessible the `getPropertyValue`method. Property names are contained in the associated `POSDataProperties`interface class (e.g., property names for options are contained in the `OptionsProperties`interface).

See Also:
com.ibm.retail.AEF.session.AEFSession#getPOSDataProvider getPOSDataProvider,

com.ibm.retail.AEF.event.POSAppEventListener POSAppEventListener,

com.ibm.retail.AEF.event.AEFPropertyChangeListener AEFPropertyChangeListener,

com.ibm.retail.AEF.data.POSDataProvider#getPropertyValue getPropertyValue

Method Summary

void	`addAEFPropertyChangeListener(AEFPropertyChangeListener listener)` Add a property change listener to listen to all data provider property change events.
void	`addAEFPropertyChangeListener(AEFPropertyChangeListener listener,java.lang.String listener)` Add a filtered property change listener to listen to data provider events for a specific category.
void	`addAEFPropertyChangeListener(AEFPropertyChangeListener listener,java.lang.String listener,java.lang.String listener)` Add a filtered property change listener to listen to data provider events for a specific property.
void	`addCashReceiptListener(CashReceiptListener listener)` Add a listener for cash receipt events
void	`addCouponListener(CouponListener listener)` Add a listener for coupon events
void	`addCustomerListener(CustomerListener listener)` Add a listener for customer events

void	`addDiscountListener(DiscountListener listener)` Add a listener for Discount events
void	`addItemSalesListener(ItemSalesListener listener)` Add a listener for item sales events
void	`addOperatorListener(OperatorListener listener)` Add a listener for operator events
void	`addOptionsListener(OptionsListener listener)` Add a listener for initial load or modification to POS options (POS terminal and store options)
void	`addPointsListener(PointsListener listener)` Add a listener for points events
void	`addPOSAppEventListener(java.lang.String listenerType,POSAppEventListener listenerType)` Generic interface for adding listeners for a specified POSAppEvent listeners.
void	`addPOSAppEventListenerSupport(POSAppEventListenerSupport listenerSupport)` Generic "plug-in" interface for adding listener support for POSAppEvent listeners.
void	`addReportListener(ReportListener listener)` Add a listener for Report events generated when report data is available
void	`addScaleListener(ScaleListener listener)` Add a listener for Scale events generated when an item is weighed
void	`addStateChangeListener(StateChangeListener listener)` Add a listener for POS State Change events
void	`addTenderListener(TenderListener listener)` Add a listener for tender entry events
void	`addTransactionStatusListener(TransactionStatusListener listener)` Add a listener for TransactionStatus events
void	`addTransactionTotalsListener(TransactionTotalsListener listener)` Add a listener for TransactionTotals events
void	`addWorkstationStatusListener(WorkstationStatusListener listener)` Add a listener for Workstation status events
FunctionCodeDefinition[]	`getGlobalFunctionCodes()` Gets the global functioncodes from the POS
java.lang.String	`getPropertiesAsString()` Returns a string containing all the property values.
java.lang.Object	`getPropertyValue(java.lang.String property)` Get a property value for a specified property name.

java.lang.Object	`getPropertyValue(java.lang.String category,java.lang.String category)` Get a property value for a specified category and property name.
java.lang.Object	`getPropertyValue(java.lang.String category,java.lang.String category,boolean category)` Get a property value for a specified category and property name.
AEFSession	`getSession()` Gets the AEFSession for this instance.
StateDefinition	`getStateDefinition(int stateID)` Gets the POS StateDefinition for a state ID.
void	`removeAEFPropertyChangeListener(AEFPropertyChangeListener listener)` Remove a property change listener.
void	`removeAEFPropertyChangeListener(AEFPropertyChangeListener listener,java.lang.String listener)` Remove a filtered property change listener
void	`removeAEFPropertyChangeListener(AEFPropertyChangeListener listener,java.lang.String listener,java.lang.String listener)` Remove a filtered property change listener
void	`removeAllListeners(java.lang.Object listener)` Remove a listener from all listener vectors
void	`removeCashReceiptListener(CashReceiptListener listener)` Remove a listener for cash receipt events
void	`removeCouponListener(CouponListener listener)` Remove a listener for coupon events
void	`removeCustomerListener(CustomerListener listener)` Remove a listener for customer events
void	`removeDiscountListener(DiscountListener listener)` Remove a listener for Discount events
void	`removeItemSalesListener(ItemSalesListener listener)` Remove a listener for item sales events
void	`removeOperatorListener(OperatorListener listener)` Remove a listener for operator events
void	`removeOptionsListener(OptionsListener listener)` Remove a listener for initial load or modification to POS options (POS terminal and store options)
void	`removePointsListener(PointsListener listener)` Remove a listener for points events
void	`removePOSAppEventListener(java.lang.String listenerType,POSAppEventListener listenerType)` Generic interface for removing listeners for a specified POSAppEvent listeners.

void	`removePOSAppEventListenerSupport(POSAppEventListenerSupport listenerSupport)` Generic "plug-in" interface for removing listener support for POSAppEvent listeners
void	`removeReportListener(ReportListener listener)` Remove a report listener
void	`removeScaleListener(ScaleListener listener)` Remove a scale listener
void	`removeStateChangeListener(StateChangeListener listener)` Remove a listener for StateChange events
void	`removeTenderListener(TenderListener listener)` Remove a listener for tender entry events
void	`removeTransactionStatusListener(TransactionStatusListener listener)` Remove a listener for TransactionStatus events
void	`removeTransactionTotalsListener(TransactionTotalsListener listener)` Remove a listener for TransactionTotals events
void	`removeWorkstationStatusListener(WorkstationStatusListener listener)` Remove a listener for WorkstationStatus events
void	`setPropertyValue(java.lang.String category,java.lang.String category,java.lang.Object category)` Set a property value for a specified category and property name.
void	`setSession(AEFSession session)` Sets the AEFSession for this instance.

Methods

addItemSalesListener

```
public void addItemSalesListener(ItemSalesListener listener)
                          throws java.rmi.RemoteException,
                                 AEFException
```

Add a listener for item sales events

Parameters:

`listener -`
the ItemSalesListener to be added

Exceptions:

`java.rmi.RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

(continued from last page)

removeItemSalesListener

```
public void removeItemSalesListener(ItemSalesListener listener)
                              throws java.rmi.RemoteException,
                                     AEFException
```

Remove a listener for item sales events

Parameters:

`listener -`
the ItemSalesListener to be removed

Exceptions:

`RemoteException -`
if remote access fails
`AEFException -`
if listener can not be added

---

addCashReceiptListener

```
public void addCashReceiptListener(CashReceiptListener listener)
                             throws java.rmi.RemoteException,
                                    AEFException
```

Add a listener for cash receipt events

Parameters:

`listener -`
the CashReceiptListener to be added

Exceptions:

`java.rmi.RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

---

removeCashReceiptListener

```
public void removeCashReceiptListener(CashReceiptListener listener)
                                throws java.rmi.RemoteException,
                                       AEFException
```

Remove a listener for cash receipt events

Parameters:

`listener -`
the CashReceiptListener to be removed

Exceptions:

`java.rmi.RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

---

addTenderListener

```
public void addTenderListener(TenderListener listener)
                        throws java.rmi.RemoteException,
                               AEFException
```

Add a listener for tender entry events

Parameters:

`listener -`
the TenderListener to be added

(continued from last page)

Exceptions:

`java.rmi.RemoteException`

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

removeTenderListener

```
public void removeTenderListener(TenderListener listener)
                          throws java.rmi.RemoteException,
                                 AEFException
```

Remove a listener for tender entry events

Parameters:

`listener -`
the TenderListener to be removed

Exceptions:

`java.rmi.RemoteException`

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

addCouponListener

```
public void addCouponListener(CouponListener listener)
                       throws java.rmi.RemoteException,
                              AEFException
```

Add a listener for coupon events

Parameters:

`listener -`
the CoupnListener to be added

Exceptions:

`java.rmi.RemoteException`

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

removeCouponListener

```
public void removeCouponListener(CouponListener listener)
                          throws java.rmi.RemoteException,
                                 AEFException
```

Remove a listener for coupon events

Parameters:

`listener -`
the CouponListener to be removed

Exceptions:

`java.rmi.RemoteException`

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

(continued from last page)

addCustomerListener

```
public void addCustomerListener(CustomerListener listener)
                         throws java.rmi.RemoteException,
                                AEFException
```

Add a listener for customer events

Parameters:

`listener -`
the CustomerListener to be added

Exceptions:

`java.rmi.RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

removeCustomerListener

```
public void removeCustomerListener(CustomerListener listener)
                           throws java.rmi.RemoteException,
                                  AEFException
```

Remove a listener for customer events

Parameters:

`listener -`
the CustomerListener to be removed

Exceptions:

`java.rmi.RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

addPointsListener

```
public void addPointsListener(PointsListener listener)
                        throws java.rmi.RemoteException,
                               AEFException
```

Add a listener for points events

Parameters:

`listener -`
the PointsListener to be added

Exceptions:

`java.rmi.RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

removePointsListener

```
public void removePointsListener(PointsListener listener)
                         throws java.rmi.RemoteException,
                                AEFException
```

Remove a listener for points events

Parameters:

`listener -`
the PointsListener to be removed

(continued from last page)

Exceptions:

`java.rmi.RemoteException`

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

addOperatorListener

```
public void addOperatorListener(OperatorListener listener)
                         throws java.rmi.RemoteException,
                                AEFException
```

Add a listener for operator events

Parameters:

`listener -`
the OperatorListener to be added

Exceptions:

`java.rmi.RemoteException`

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

removeOperatorListener

```
public void removeOperatorListener(OperatorListener listener)
                            throws java.rmi.RemoteException,
                                   AEFException
```

Remove a listener for operator events

Parameters:

`listener -`
the OperatorListener to be removed

Exceptions:

`java.rmi.RemoteException`

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

addOptionsListener

```
public void addOptionsListener(OptionsListener listener)
                        throws java.rmi.RemoteException,
                               AEFException
```

Add a listener for initial load or modification to POS options (POS terminal and store options)

Parameters:

`listener -`
the OptionsListener to be added

Exceptions:

`java.rmi.RemoteException`

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

(continued from last page)

removeOptionsListener

```
public void removeOptionsListener(OptionsListener listener)
                            throws java.rmi.RemoteException,
                                   AEFException
```

Remove a listener for initial load or modification to POS options (POS terminal and store options)

Parameters:

`listener -`
the OperatorListener to be removed

Exceptions:

`java.rmi.RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

addDiscountListener

```
public void addDiscountListener(DiscountListener listener)
                          throws java.rmi.RemoteException,
                                 AEFException
```

Add a listener for Discount events

Parameters:

`listener -`
the DiscountListener to be added

Exceptions:

`java.rmi.RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

removeDiscountListener

```
public void removeDiscountListener(DiscountListener listener)
                             throws java.rmi.RemoteException,
                                    AEFException
```

Remove a listener for Discount events

Parameters:

`listener -`
the DiscountListener to be removed

Exceptions:

`java.rmi.RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

addTransactionTotalsListener

```
public void addTransactionTotalsListener(TransactionTotalsListener listener)
                                   throws java.rmi.RemoteException,
                                          AEFException
```

Add a listener for TransactionTotals events

Parameters:

`listener -`
the TransactionTotalsListener to be added

(continued from last page)

Exceptions:

`java.rmi.RemoteException`

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

removeTransactionTotalsListener

```
public void removeTransactionTotalsListener(TransactionTotalsListener listener)
                                      throws java.rmi.RemoteException,
                                             AEFException
```

Remove a listener for TransactionTotals events

Parameters:

`listener -`
the TransactionTotalsListener to be removed

Exceptions:

`java.rmi.RemoteException`

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

addTransactionStatusListener

```
public void addTransactionStatusListener(TransactionStatusListener listener)
                                    throws java.rmi.RemoteException,
                                           AEFException
```

Add a listener for TransactionStatus events

Parameters:

`listener -`
the TransactionStatusListener to be added

Exceptions:

`java.rmi.RemoteException`

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

removeTransactionStatusListener

```
public void removeTransactionStatusListener(TransactionStatusListener listener)
                                      throws java.rmi.RemoteException,
                                             AEFException
```

Remove a listener for TransactionStatus events

Parameters:

`listener -`
the TransactionStatusListener to be removed

Exceptions:

`java.rmi.RemoteException`

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

(continued from last page)

addReportListener

```
public void addReportListener(ReportListener listener)
                        throws java.rmi.RemoteException,
                               AEFException
```

Add a listener for Report events generated when report data is available

Parameters:

`listener -`
the ReportListener to be added

Exceptions:

`java.rmi.RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

removeReportListener

```
public void removeReportListener(ReportListener listener)
                          throws java.rmi.RemoteException,
                                 AEFException
```

Remove a report listener

Parameters:

`listener -`
the ReportListener to be removed

Exceptions:

`java.rmi.RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

addScaleListener

```
public void addScaleListener(ScaleListener listener)
                       throws java.rmi.RemoteException,
                              AEFException
```

Add a listener for Scale events generated when an item is weighed

Parameters:

`listener -`
the ScaleListener to be added

Exceptions:

`java.rmi.RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

removeScaleListener

```
public void removeScaleListener(ScaleListener listener)
                         throws java.rmi.RemoteException,
                                AEFException
```

Remove a scale listener

Parameters:

`listener -`
the ScaleListener to be removed

(continued from last page)

Exceptions:

`java.rmi.RemoteException`

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

addStateChangeListener

```
public void addStateChangeListener(StateChangeListener listener)
                              throws java.rmi.RemoteException,
                                     AEFException
```

Add a listener for POS State Change events

Parameters:

`listener -`
the StateChangeListener to be added

Exceptions:

`java.rmi.RemoteException`

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

removeStateChangeListener

```
public void removeStateChangeListener(StateChangeListener listener)
                                 throws java.rmi.RemoteException,
                                        AEFException
```

Remove a listener for StateChange events

Parameters:

`listener -`
the StateChangeListener to be removed

Exceptions:

`java.rmi.RemoteException`

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

addWorkstationStatusListener

```
public void addWorkstationStatusListener(WorkstationStatusListener listener)
                                    throws java.rmi.RemoteException,
                                           AEFException
```

Add a listener for Workstation status events

Parameters:

`listener -`
the WorkstationStatusListener to be added

Exceptions:

`java.rmi.RemoteException`

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

(continued from last page)

removeWorkstationStatusListener

```
public void removeWorkstationStatusListener(WorkstationStatusListener listener)
                                      throws java.rmi.RemoteException,
                                             AEFException
```

Remove a listener for WorkstationStatus events

Parameters:

`listener -`
the WorkstationStatusListener to be removed

Exceptions:

`java.rmi.RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

---

addPOSAppEventListenerSupport

```
public void addPOSAppEventListenerSupport(POSAppEventListenerSupport listenerSupport)
                                    throws java.rmi.RemoteException
```

Generic "plug-in" interface for adding listener support for POSAppEvent listeners. This method allows additional listener interfaces to be added to the POSDataProvider.

Parameters:

`listenerSupport -`
the listenerSupport object to be added

Exceptions:

`RemoteException -`
if remote access fails

---

removePOSAppEventListenerSupport

```
public void removePOSAppEventListenerSupport(POSAppEventListenerSupport
listenerSupport)
                                       throws java.rmi.RemoteException
```

Generic "plug-in" interface for removing listener support for POSAppEvent listeners

Parameters:

`listenerSupport -`
the listenerSupport object to be removed

Exceptions:

`java.rmi.RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

---

addPOSAppEventListener

```
public void addPOSAppEventListener(java.lang.String listenerType,
                                   POSAppEventListener listener)
                             throws java.rmi.RemoteException,
                                   AEFException
```

Generic interface for adding listeners for a specified POSAppEvent listeners. This method provides a generic interface for adding a listener. It is provided as an extension mechanism to allow additional listener interfaces to be added. *Clients should use the provided specific* `addListener` *methods for adding AEF provided listeners such as* `CashReceipt, TransactionTotalsListener, ....`

Parameters:

(continued from last page)

`listenerType -`
fully qualified classname of listener interface
`listener -`
the listener object to be added

Exceptions:
`java.rmi.RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

removePOSAppEventListener

```
public void removePOSAppEventListener(java.lang.String listenerType,
                                      POSAppEventListener listener)
                                 throws java.rmi.RemoteException,
                                      AEFException
```

Generic interface for removing listeners for a specified POSAppEvent listeners. This method provides a generic interface for removing a listener. It is provided as an extension mechanism to allow additional listener interfaces to be added. *Clients should use the provided specific* `removeListener` *methods for removing AEF provided listeners such as* `CashReceipt, TransactionTotalsListener,....`

Parameters:
`listenerType -`
fully qualified classname of listener interface
`listener -`
the listener object to be added

Exceptions:
`java.rmi.RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_LISTENER_SUPPORT

addAEFPropertyChangeListener

```
public void addAEFPropertyChangeListener(AEFPropertyChangeListener listener)
                                    throws java.rmi.RemoteException
```

Add a property change listener to listen to all data provider property change events. *Note: Due to the large number of property change events, it is best to use a specific event listener (e.g., TransactionTotalsListener) or a filtered property change listener (e.g., a listener for a change to the POS_STATE property). Filtered listeners restrict the events to a specific category or property to be monitored.*

Parameters:
`listener -`
the AEFPropertyChangeListener to be added

Exceptions:
`java.rmi.RemoteException`

removeAEFPropertyChangeListener

```
public void removeAEFPropertyChangeListener(AEFPropertyChangeListener listener)
                                   throws java.rmi.RemoteException
```

Remove a property change listener.

Parameters:
`listener -`
the AEFPropertyChangeListener to be removed

Exceptions:
`RemoteException`

addAEFPropertyChangeListener

```
public void addAEFPropertyChangeListener(AEFPropertyChangeListener listener,
                                         java.lang.String category)
                                  throws java.rmi.RemoteException
```

Add a filtered property change listener to listen to data provider events for a specific category. *Note: Due to the large number of property change events, it is best to use a specific event listener (e.g., TransactionTotalsListener) or a filtered property change listener (e.g., a listener for a change to the POS_STATE property).*

Parameters:

`listener -`
the AEFPropertyChangeListener to be added
`category -`
identifies the category of data (provides filter)

Exceptions:

`java.rmi.RemoteException`

removeAEFPropertyChangeListener

```
public void removeAEFPropertyChangeListener(AEFPropertyChangeListener listener,
                                            java.lang.String category)
                                     throws java.rmi.RemoteException
```

Remove a filtered property change listener

Parameters:

`listener -`
the AEFPropertyChangeListener to be removed
`category -`
identifies the category of data

Exceptions:

`java.rmi.RemoteException`

addAEFPropertyChangeListener

```
public void addAEFPropertyChangeListener(AEFPropertyChangeListener listener,
                                         java.lang.String category,
                                         java.lang.String property)
                                  throws java.rmi.RemoteException
```

Add a filtered property change listener to listen to data provider events for a specific property.

Parameters:

`listener -`
the PropertyChangeListener to be added
`category -`
identifies the category of data
`property -`
identifies the property data

Exceptions:

`java.rmi.RemoteException`

removeAEFPropertyChangeListener

```
public void removeAEFPropertyChangeListener(AEFPropertyChangeListener listener,
                                            java.lang.String category,
                                            java.lang.String property)
                                     throws java.rmi.RemoteException
```

Remove a filtered property change listener

(continued from last page)

Parameters:

`listener -`
the PropertyChangeListener to be removed
`category -`
identifies the category of data
`property -`
identifies the property data

Exceptions:

`java.rmi.RemoteException`

---

setPropertyValue

```
public void setPropertyValue(java.lang.String category,
                             java.lang.String property,
                             java.lang.Object value)
                      throws java.rmi.RemoteException
```

Set a property value for a specified category and property name.

Parameters:

`category -`
identifies the category name of the data
`property -`
identifies the property name of the data
`value -`
the property value to be added or updated

Exceptions:

`java.rmi.RemoteException`

---

getPropertyValue

```
public java.lang.Object getPropertyValue(java.lang.String category,
                                         java.lang.String property)
                                  throws java.rmi.RemoteException
```

Get a property value for a specified category and property name. This version will wait for any pending property changes on the queue. A null value is returned if the property is not found, and no exception is thrown.

Parameters:

`category -`
identifies the category name of the data
`property -`
identifies the property name of the data

Returns:

the property value requested or null if the property does not exist

Exceptions:

`java.rmi.RemoteException`

---

getPropertyValue

```
public java.lang.Object getPropertyValue(java.lang.String property)
                                  throws java.rmi.RemoteException
```

Get a property value for a specified property name. This version will wait for any pending property changes on the queue. Each category will be examined until the first property with the specified name is located. The value of the first property located with the requested name will be returned. A null value is returned if the property is not found, and no exception is thrown.

Parameters:

(continued from last page)

`property -`
identifies the property name of the data

Returns:

the property value requested or null if the property does not exist

Exceptions:
`java.rmi.RemoteException`

getPropertyValue

```
public java.lang.Object getPropertyValue(java.lang.String category,
                                         java.lang.String property,
                                         boolean waitForPending)
                                  throws java.rmi.RemoteException
```

Get a property value for a specified category and property name. A null value is returned if the property is not found, and no exception is thrown.

Parameters:
`category -`
identifies the category name of the data
`property -`
identifies the property name of the data
`waitForPending -`
True to wait for any pending property changes already on the queue.

Returns:

the property value requested or null if the property does not exist

Exceptions:
`java.rmi.RemoteException`

removeAllListeners

```
public void removeAllListeners(java.lang.Object listener)
                        throws java.rmi.RemoteException
```

Remove a listener from all listener vectors

Parameters:
`listener -`
Object to be removed

Exceptions:
`java.rmi.RemoteException`

getPropertiesAsString

```
public java.lang.String getPropertiesAsString()
                                       throws java.rmi.RemoteException
```

Returns a string containing all the property values. Useful for debugging.

Returns:

String

Exceptions:
`java.rmi.RemoteException`

setSession

```
public void setSession(AEFSession session)
                 throws java.rmi.RemoteException
```

Sets the AEFSession for this instance.

Parameters:

`session -`
AEFSession associated with this object

Exceptions:

`java.rmi.RemoteException`

getSession

```
public AEFSession getSession()
                       throws java.rmi.RemoteException
```

Gets the AEFSession for this instance.

Returns:

AEFSession associated with this object

Exceptions:

`java.rmi.RemoteException`

getStateDefinition

```
public StateDefinition getStateDefinition(int stateID)
                                   throws java.rmi.RemoteException
```

Gets the POS StateDefinition for a state ID.

Parameters:

`stateID -`
the identifier for the state

Returns:

StateDefinition

Exceptions:

`java.rmi.RemoteException`

getGlobalFunctionCodes

```
public FunctionCodeDefinition[] getGlobalFunctionCodes()
                                                 throws java.rmi.RemoteException
```

Gets the global functioncodes from the POS

Returns:

FunctionCodeDefinition[]

Exceptions:

`java.rmi.RemoteException`

com.ibm.retail.AEF.data

Interface POSDeviceProperties

public interface **POSDeviceProperties**

`POSDeviceProperties`
contains the data provider property names for POS device and IO processor attributes. Devices include:
- CashReceiptPrinter
- Scanner
- 2x20 Prompt
- IO Processor (state changes)
- Scales
- Keylock
- Cashdrawer

These properties are communicated to client listeners through the `AEFPropertyChangeListener`interface. Listener registration for `POSDevicePropertiesis` through the `addAEFPropertyChangeListener`method of `POSDataProvider`.

See Also:
com.ibm.retail.AEF.event.AEFPropertyChangeListener AEFPropertyChangeListener,

com.ibm.retail.AEF.data.POSDataProvider#getPropertyValue getPropertyValue

Field Summary

static java.lang.String	`ANPROMPT_LINE` ANPrompt (2x20 operator display) line identifier
static java.lang.String	`ANPROMPT_LINE1` ANPrompt (2x20 operator display) line 1 identifier
static java.lang.String	`ANPROMPT_LINE2` ANPrompt (2x20 operator display) line 2 identifier
static java.lang.String	`APP_WILL_PROVIDE_DATA` POS Application provides print lines
static java.lang.String	`CASH_DRAWER_OPEN` Indicates whether the cash drawer is open.
static java.lang.String	`CATEGORY` POSDevice Category identifier
static java.lang.String	`CHANGE_TO_CURRENT` State change-to-current identifier
static java.lang.String	`CUST_PROMPT_LINE` Customer Prompt line display identifier
static java.lang.String	`CUST_PROMPT_LINE1` Customer Prompt line display line 1 identifier
static java.lang.String	`CUST_PROMPT_LINE2` Customer Prompt line display line 2 identifier

static java.lang.String	`CUSTOMER_DISPLAY` Logical device name of the customer line display
static java.lang.String	`INVALID_KEYED_LABEL` Invalid Label keyed
static java.lang.String	`KEYLOCK_IS_SUPERVISOR` Indicates whether the keylock is in mgr position.
static java.lang.String	`LAST_SCAN_LABEL` Last Label Scanned (data)
static java.lang.String	`LAST_SCAN_LABEL_TYPE` Last Label Scanned (type)
static java.lang.String	`LINE_DISPLAY_ROW` Line display devices generic identifiers Property name is logical name + "_Row_" + row number e.
static java.lang.String	`MSR_TRACK_1` MSR track 1
static java.lang.String	`MSR_TRACK_2` MSR track 2
static java.lang.String	`MSR_TRACK_3` MSR track 3
static java.lang.String	`OPERATOR_DISPLAY` Logical device name of the operator line display
static java.lang.String	`OPERATOR_PROMPT_LINE1` Operator prompt line 1 identifier
static java.lang.String	`OPERATOR_PROMPT_LINE2` Operator prompt line 2 identifier
static java.lang.String	`PAPER_CUT` Paper cut sent to printer
static java.lang.String	`POS_STATE` POS State change identifier
static java.lang.String	`POS_SUB_STATE` POS Substate change identifier
static java.lang.String	`PRINT_LINE` Lines to be printed
static java.lang.String	`PRINT_LINE_ARRAY` Lines to be printed
static java.lang.String	`PRINT_LINE_FEEDS` Line feeds to be printed

static java.lang.String	PRINTER_COVER_OPEN Indicates whether the printer cover is open.
static java.lang.String	PRINTER_DOC_INSERT Indicates whether a document (slip form) is inserted in the printer.
static java.lang.String	PRINTER_OUT_OF_RECEIPT_PAPER Indicates whether the printer is out of paper.
static java.lang.String	RAW_PRINT_LINE_ARRAY Raw lines to be printed
static java.lang.String	SCALE_WEIGHT_LABELS Scale Weight Labels
static java.lang.String	SCALE_WEIGHT_UNIT Scale Weight Unit of Measure
static java.lang.String	SCALE_WEIGHT_VALUE Scale Weight Value
static java.lang.String	SYSTEM_BUSY System busy (input queue locked)

Fields

CATEGORY

`public static final java.lang.String` **`CATEGORY`**

POSDevice Category identifier

ANPROMPT_LINE

`public static final java.lang.String` **`ANPROMPT_LINE`**

ANPrompt (2x20 operator display) line identifier

ANPROMPT_LINE1

`public static final java.lang.String` **`ANPROMPT_LINE1`**

ANPrompt (2x20 operator display) line 1 identifier

ANPROMPT_LINE2

`public static final java.lang.String` **`ANPROMPT_LINE2`**

ANPrompt (2x20 operator display) line 2 identifier

OPERATOR_PROMPT_LINE1

`public static final java.lang.String` **`OPERATOR_PROMPT_LINE1`**

Operator prompt line 1 identifier

OPERATOR_PROMPT_LINE2

`public static final java.lang.String` **`OPERATOR_PROMPT_LINE2`**

Operator prompt line 2 identifier

CUST_PROMPT_LINE

`public static final java.lang.String` **`CUST_PROMPT_LINE`**

Customer Prompt line display identifier

CUST_PROMPT_LINE1

`public static final java.lang.String` **`CUST_PROMPT_LINE1`**

Customer Prompt line display line 1 identifier

CUST_PROMPT_LINE2

`public static final java.lang.String` **`CUST_PROMPT_LINE2`**

Customer Prompt line display line 2 identifier

LINE_DISPLAY_ROW

`public static final java.lang.String` **`LINE_DISPLAY_ROW`**

Line display devices generic identifiers Property name is logical name + "_Row_" + row number e.g., data on device LineDisplay1 row1, property name = "LineDisplay1_Row_1"

OPERATOR_DISPLAY

`public static final java.lang.String` **`OPERATOR_DISPLAY`**

Logical device name of the operator line display

CUSTOMER_DISPLAY

`public static final java.lang.String` **`CUSTOMER_DISPLAY`**

Logical device name of the customer line display

POS_SUB_STATE

`public static final java.lang.String` **`POS_SUB_STATE`**

POS Substate change identifier

POS_STATE

`public static final java.lang.String` **`POS_STATE`**

POS State change identifier

CHANGE_TO_CURRENT

`public static final java.lang.String` **`CHANGE_TO_CURRENT`**

State change-to-current identifier

(continued from last page)

SYSTEM_BUSY

`public static final java.lang.String` **`SYSTEM_BUSY`**

System busy (input queue locked)

PAPER_CUT

`public static final java.lang.String` **`PAPER_CUT`**

Paper cut sent to printer

PRINT_LINE

`public static final java.lang.String` **`PRINT_LINE`**

Lines to be printed

PRINT_LINE_ARRAY

`public static final java.lang.String` **`PRINT_LINE_ARRAY`**

Lines to be printed

RAW_PRINT_LINE_ARRAY

`public static final java.lang.String` **`RAW_PRINT_LINE_ARRAY`**

Raw lines to be printed

PRINT_LINE_FEEDS

`public static final java.lang.String` **`PRINT_LINE_FEEDS`**

Line feeds to be printed

APP_WILL_PROVIDE_DATA

`public static final java.lang.String` **`APP_WILL_PROVIDE_DATA`**

POS Application provides print lines

SCALE_WEIGHT_VALUE

`public static final java.lang.String` **`SCALE_WEIGHT_VALUE`**

Scale Weight Value

SCALE_WEIGHT_UNIT

`public static final java.lang.String` **`SCALE_WEIGHT_UNIT`**

Scale Weight Unit of Measure

SCALE_WEIGHT_LABELS

`public static final java.lang.String` **`SCALE_WEIGHT_LABELS`**

Scale Weight Labels

INVALID_KEYED_LABEL

`public static final java.lang.String` **`INVALID_KEYED_LABEL`**

Invalid Label keyed

LAST_SCAN_LABEL

`public static final java.lang.String` **`LAST_SCAN_LABEL`**

Last Label Scanned (data)

LAST_SCAN_LABEL_TYPE

`public static final java.lang.String` **`LAST_SCAN_LABEL_TYPE`**

Last Label Scanned (type)

KEYLOCK_IS_SUPERVISOR

`public static final java.lang.String` **`KEYLOCK_IS_SUPERVISOR`**

Indicates whether the keylock is in mgr position.

CASH_DRAWER_OPEN

`public static final java.lang.String` **`CASH_DRAWER_OPEN`**

Indicates whether the cash drawer is open.

MSR_TRACK_1

`public static final java.lang.String` **`MSR_TRACK_1`**

MSR track 1

MSR_TRACK_2

`public static final java.lang.String` **`MSR_TRACK_2`**

MSR track 2

MSR_TRACK_3

`public static final java.lang.String` **`MSR_TRACK_3`**

MSR track 3

PRINTER_COVER_OPEN

`public static final java.lang.String` **`PRINTER_COVER_OPEN`**

Indicates whether the printer cover is open.

PRINTER_DOC_INSERT

`public static final java.lang.String` **`PRINTER_DOC_INSERT`**

Indicates whether a document (slip form) is inserted in the printer.

PRINTER_OUT_OF_RECEIPT_PAPER

`public static final java.lang.String` **`PRINTER_OUT_OF_RECEIPT_PAPER`**

Indicates whether the printer is out of paper.

com.ibm.retail.AEF.data

Interface StoreOptionsProperties

public interface **StoreOptionsProperties**

`StoreOptionsProperties`
contains the `POSDataProvider`property names for store options attributes. These properties are communicated to client listeners through the `OptionsListener`interface.

See Also:
com.ibm.retail.AEF.event.OptionsListener OptionsListener, com.ibm.retail.AEF.event.OptionsEvent#getStoreOptions getStoreOptions, com.ibm.retail.AEF.data.POSDataProvider#getPropertyValue getPropertyValue

Field Summary

static java.lang.String	`ALPHA_DRIVERS_LICENSE_INPUT_ALLOWED` Alpha drivers license input allowed identifier
static java.lang.String	`ALPHA_STATE_INPUT_ALLOWED` Alpha state input allowed identifier
static java.lang.String	`COUPON_MULTIPLIER_ENABLED` Coupon multiplier enabled (ACE,SA)
static java.lang.String	`CUSTOMER_FUNCTION_CODE` Customer Function Code identifier
static java.lang.String	`EATIN_TAKEOUT_PROMPT_ENABLED` Eat-in/Take-out prompt is enabled (ACE,SA)
static java.lang.String	`FOOD_STAMPS_ALLOWED` Food Stamps Allowed identifier
static java.lang.String	`FOOD_STAMPS_ONLY_ALLOWED` Allow only Food Stamps after a Food Stamp Total identifier
static java.lang.String	`FOREIGN_TENDER_SUPPORTED` Foreign Tender Supported identifier
static java.lang.String	`GIFT_RECEIPT_PRINTING_ENABLED` Gift receipt printing enabled identifier
static java.lang.String	`MANAGERS_KEY_NEEDED` Manager's Key Needed identifier
static java.lang.String	`MAX_SUSPENDED_TRANSACTIONS` Max number of suspended transactions allowed per terminal identifier
static java.lang.String	`MAXIMUM_TRANSACTION_SIZE` Maximum transaction size identifier

static java.lang.String	MULTIPLE_CASH_DRAWER_SUPPORT Multiple cash drawer support enabled identifier
static java.lang.String	NO_SALE_PRICE_VERIFY_IN_TRANSACTION_ALLOWED No Sale Price Verify Allowed identifier
static java.lang.String	OTR_ENABLED OTR display is active (ACE,SA)
static java.lang.String	OTR_FUNCTIONS_ENABLED OTR function is active (ACE,SA)
static java.lang.String	OTR_PRINT_ENABLED OTR print is active (ACE,SA)
static java.lang.String	OTR_REFRESH_ENABLED OTR refresh is active (ACE,SA)
static java.lang.String	SCAN_MANAGER_ID_REQUIRED Scan manager ID required identifier
static java.lang.String	SUSPEND_TRANSACTION_ALLOWED Suspend Transaction Allowed identifier
static java.lang.String	TRANSACTION_DEFINITION Transaction definition identifier references a collection of TransactionDefinition objects
static java.lang.String	TRANSACTION_WARNING_SIZE Transaction warning size identifier
static java.lang.String	VOLUME_INPUT_DECIMAL_PLACES Number of decimal places for measure of items sold by volume (fuel)
static java.lang.String	VOLUME_UNIT_PRICE_DECIMAL_PLACES Number of decimal places for unit price of items sold by volume (fuel)
static java.lang.String	WEIGHT_DECIMAL_PLACES Format options for weight items
static java.lang.String	WIC_TENDER_ONLY_IN_WIC_TRANS Identifier that denotes if only WIC tenders are available in a WIC Transaction
static java.lang.String	WICEBT_ID WICEBT identifier

Fields

NO_SALE_PRICE_VERIFY_IN_TRANSACTION_ALLOWED

`public static final java.lang.String` **`NO_SALE_PRICE_VERIFY_IN_TRANSACTION_ALLOWED`**

No Sale Price Verify Allowed identifier

MANAGERS_KEY_NEEDED

`public static final java.lang.String` **`MANAGERS_KEY_NEEDED`**

Manager's Key Needed identifier

MAXIMUM_TRANSACTION_SIZE

`public static final java.lang.String` **`MAXIMUM_TRANSACTION_SIZE`**

Maximum transaction size identifier

TRANSACTION_WARNING_SIZE

`public static final java.lang.String` **`TRANSACTION_WARNING_SIZE`**

Transaction warning size identifier

FOOD_STAMPS_ALLOWED

`public static final java.lang.String` **`FOOD_STAMPS_ALLOWED`**

Food Stamps Allowed identifier

FOOD_STAMPS_ONLY_ALLOWED

`public static final java.lang.String` **`FOOD_STAMPS_ONLY_ALLOWED`**

Allow only Food Stamps after a Food Stamp Total identifier

MAX_SUSPENDED_TRANSACTIONS

`public static final java.lang.String` **`MAX_SUSPENDED_TRANSACTIONS`**

Max number of suspended transactions allowed per terminal identifier

CUSTOMER_FUNCTION_CODE

`public static final java.lang.String` **`CUSTOMER_FUNCTION_CODE`**

Customer Function Code identifier

SUSPEND_TRANSACTION_ALLOWED

`public static final java.lang.String` **`SUSPEND_TRANSACTION_ALLOWED`**

Suspend Transaction Allowed identifier

WIC_TENDER_ONLY_IN_WIC_TRANS

`public static final java.lang.String` **`WIC_TENDER_ONLY_IN_WIC_TRANS`**

Identifier that denotes if only WIC tenders are available in a WIC Transaction

WEIGHT_DECIMAL_PLACES

`public static final java.lang.String` **`WEIGHT_DECIMAL_PLACES`**

Format options for weight items

(continued from last page)

FOREIGN_TENDER_SUPPORTED

`public static final java.lang.String` **`FOREIGN_TENDER_SUPPORTED`**

Foreign Tender Supported identifier

ALPHA_STATE_INPUT_ALLOWED

`public static final java.lang.String` **`ALPHA_STATE_INPUT_ALLOWED`**

Alpha state input allowed identifier

ALPHA_DRIVERS_LICENSE_INPUT_ALLOWED

`public static final java.lang.String` **`ALPHA_DRIVERS_LICENSE_INPUT_ALLOWED`**

Alpha drivers license input allowed identifier

MULTIPLE_CASH_DRAWER_SUPPORT

`public static final java.lang.String` **`MULTIPLE_CASH_DRAWER_SUPPORT`**

Multiple cash drawer support enabled identifier

OTR_ENABLED

`public static final java.lang.String` **`OTR_ENABLED`**

OTR display is active (ACE,SA)

OTR_PRINT_ENABLED

`public static final java.lang.String` **`OTR_PRINT_ENABLED`**

OTR print is active (ACE,SA)

OTR_REFRESH_ENABLED

`public static final java.lang.String` **`OTR_REFRESH_ENABLED`**

OTR refresh is active (ACE,SA)

OTR_FUNCTIONS_ENABLED

`public static final java.lang.String` **`OTR_FUNCTIONS_ENABLED`**

OTR function is active (ACE,SA)

COUPON_MULTIPLIER_ENABLED

`public static final java.lang.String` **`COUPON_MULTIPLIER_ENABLED`**

Coupon multiplier enabled (ACE,SA)

EATIN_TAKEOUT_PROMPT_ENABLED

`public static final java.lang.String` **`EATIN_TAKEOUT_PROMPT_ENABLED`**

Eat-in/Take-out prompt is enabled (ACE,SA)

VOLUME_INPUT_DECIMAL_PLACES

`public static final java.lang.String` **`VOLUME_INPUT_DECIMAL_PLACES`**

Number of decimal places for measure of items sold by volume (fuel)

VOLUME_UNIT_PRICE_DECIMAL_PLACES

`public static final java.lang.String` **`VOLUME_UNIT_PRICE_DECIMAL_PLACES`**

Number of decimal places for unit price of items sold by volume (fuel)

GIFT_RECEIPT_PRINTING_ENABLED

`public static final java.lang.String` **`GIFT_RECEIPT_PRINTING_ENABLED`**

Gift receipt printing enabled identifier

SCAN_MANAGER_ID_REQUIRED

`public static final java.lang.String` **`SCAN_MANAGER_ID_REQUIRED`**

Scan manager ID required identifier

WICEBT_ID

`public static final java.lang.String` **`WICEBT_ID`**

WICEBT identifier

TRANSACTION_DEFINITION

`public static final java.lang.String` **`TRANSACTION_DEFINITION`**

Transaction definition identifier references a collection of TransactionDefinition objects

See Also:

com.ibm.retail.AEF.event.TransactionDefinition

com.ibm.retail.AEF.data

Interface StoreProperties

public interface **StoreProperties**

`StoreProperties`
contains the data provider property names for store attributes. These properties are communicated to client listeners through the `OptionsListener`interface.

See Also:
com.ibm.retail.AEF.event.OptionsListener OptionsListener, com.ibm.retail.AEF.event.OptionsEvent#getStoreDefinition getStoreDefinition, com.ibm.retail.AEF.data.POSDataProvider#getPropertyValue getPropertyValue

Field Summary

static java.lang.String	`CATEGORY` Category identifier
static java.lang.String	`STORE_ADDRESS1` Store Address 1 identifier
static java.lang.String	`STORE_ADDRESS2` Store Address 2 identifier
static java.lang.String	`STORE_CITY` Store City identifier
static java.lang.String	`STORE_DIVISION` Store Division identifier
static java.lang.String	`STORE_NAME` Store Name identifier
static java.lang.String	`STORE_NUMBER` Store Number identifier
static java.lang.String	`STORE_PHONE1` Store Phone 1 identifier
static java.lang.String	`STORE_PHONE2` Store Phone 2 identifier
static java.lang.String	`STORE_STATE` Store State identifier
static java.lang.String	`STORE_ZIP` Store Zip identifier

Fields

(continued from last page)

CATEGORY

`public static final java.lang.String` **`CATEGORY`**

Category identifier

STORE_NAME

`public static final java.lang.String` **`STORE_NAME`**

Store Name identifier

STORE_DIVISION

`public static final java.lang.String` **`STORE_DIVISION`**

Store Division identifier

STORE_NUMBER

`public static final java.lang.String` **`STORE_NUMBER`**

Store Number identifier

STORE_ADDRESS1

`public static final java.lang.String` **`STORE_ADDRESS1`**

Store Address 1 identifier

STORE_ADDRESS2

`public static final java.lang.String` **`STORE_ADDRESS2`**

Store Address 2 identifier

STORE_CITY

`public static final java.lang.String` **`STORE_CITY`**

Store City identifier

STORE_STATE

`public static final java.lang.String` **`STORE_STATE`**

Store State identifier

STORE_ZIP

`public static final java.lang.String` **`STORE_ZIP`**

Store Zip identifier

STORE_PHONE1

`public static final java.lang.String` **`STORE_PHONE1`**

Store Phone 1 identifier

STORE_PHONE2

`public static final java.lang.String` **`STORE_PHONE2`**

(continued from last page)

Store Phone 2 identifier

com.ibm.retail.AEF.data

Interface TenderProperties

All Superinterfaces:
LineItemProperties, POSDataProperties

public interface **TenderProperties**
extends LineItemProperties

`TenderProperties`
contains the `POSDataProvider`names for tender attributes. These properties are communicated to client listeners through the `TenderListener`interface. Individual property values for the current tender are also accessible through the *POSDataProvider* interface.

See Also:
com.ibm.retail.AEF.event.TenderListener TenderListener, com.ibm.retail.AEF.data.POSDataProvider#getPropertyValue getPropertyValue

Field Summary

static java.lang.String	`ACCOUNT_NUMBER` account number
static java.lang.String	`AMOUNT` amount tendered
static java.lang.String	`CARD_ID` card id (a credit card security it)
static java.lang.String	`CATEGORY` Category identifier
static java.lang.String	`CURRENCY` tender currency
static java.lang.String	`DESCRIPTION` tender description
static java.lang.String	`EXPIRATION_DATE` expiration date
static java.lang.String	`FEE` fee
static java.lang.String	`REFERENCE_NUMBER` reference number
static java.lang.String	`TENDER_ACTION` tender action performed (added, voided)

static java.lang.String	TENDER_TYPE tender type (cash, check, credit, debit)
static java.lang.String	TENDER_VARIETY tender variety

Fields

CATEGORY

`public static final java.lang.String` **`CATEGORY`**

Category identifier

TENDER_ACTION

`public static final java.lang.String` **`TENDER_ACTION`**

tender action performed (added, voided)

TENDER_TYPE

`public static final java.lang.String` **`TENDER_TYPE`**

tender type (cash, check, credit, debit)

TENDER_VARIETY

`public static final java.lang.String` **`TENDER_VARIETY`**

tender variety

CURRENCY

`public static final java.lang.String` **`CURRENCY`**

tender currency

DESCRIPTION

`public static final java.lang.String` **`DESCRIPTION`**

tender description

AMOUNT

`public static final java.lang.String` **`AMOUNT`**

amount tendered

ACCOUNT_NUMBER

`public static final java.lang.String` **`ACCOUNT_NUMBER`**

account number

EXPIRATION_DATE

`public static final java.lang.String` **`EXPIRATION_DATE`**

(continued from last page)

expiration date

CARD_ID

`public static final java.lang.String` **`CARD_ID`**

card id (a credit card security it)

REFERENCE_NUMBER

`public static final java.lang.String` **`REFERENCE_NUMBER`**

reference number

FEE

`public static final java.lang.String` **`FEE`**

fee

com.ibm.retail.AEF.data

Interface TerminalOptionsProperties

public interface **TerminalOptionsProperties**

`TerminalOptionsProperties`
contains the property name identifiers for `TerminalOptions`objects contained in `OptionsEvent`. These properties are communicated to client listeners through the `OptionsListener`interface. Individual property values for the current options are also accessible through the `POSDataProvider`interface.

See Also:
com.ibm.retail.AEF.event.OptionsListener OptionsListener, com.ibm.retail.AEF.data.POSDataProvider#getPropertyValue getPropertyValue

Field Summary

static java.lang.String	`ALLOWANCE_ALLOWED` Allowance Allowed identifier (GSA)
static java.lang.String	`CASH_DOC_ALLOWED` Cash Document Transaction Allowed identifier (GSA)
static java.lang.String	`CASH_SPEC_ALLOWED` Cash Special Transaction Allowed identifier (GSA)
static java.lang.String	`CASH_TRANSACTION_ALLOWED` Cash Transaction Allowed identifier
static java.lang.String	`CHARGE_PLAN_A_ALLOWED` Charge Plan A Transaction Allowed identifier (GSA)
static java.lang.String	`CHARGE_PLAN_B_ALLOWED` Charge Plan B Transaction Allowed identifier (GSA)
static java.lang.String	`CHARGE_PLAN_C_ALLOWED` Charge Plan C Transaction Allowed identifier (GSA)
static java.lang.String	`CHARGE_PLAN_D_ALLOWED` Charge Plan D Transaction Allowed identifier (GSA)
static java.lang.String	`COD_ALLOWED` COD Transaction Allowed identifier (GSA)
static java.lang.String	`DISCOUNTS_ALLOWED` Discounts Allowed identifier (GSA)
static java.lang.String	`LAYAWAY_ALLOWED` Layaway Transaction Allowed identifier
static java.lang.String	`NO_SALE_ALLOWED` No Sale Transaction Allowed identifier

static java.lang.String	OPERATOR_ID_REQUIRED Operator ID required identifier (GSA)
static java.lang.String	ORIGINAL_SALES_PERSON_REQUIRED Original Sales Person Required identifier (GSA)
static java.lang.String	PAYMENTS_ALLOWED Payments Allowed identifier (GSA)
static java.lang.String	RETURNS_ALLOWED Returns Allowed identifier
static java.lang.String	SEND_ALLOWED Send Transaction Allowed identifier (GSA)
static java.lang.String	SUSPEND_TRANSACTION_ALLOWED Suspend Transaction Allowed identifier (GSA)
static java.lang.String	TAX_CODE_REQUIRED Tax code required identifier
static java.lang.String	TERMS_OF_SALE_REQUIRED Terms of Sale Required identifier (GSA)
static java.lang.String	VOID_TRANSACTION_ALLOWED Void Transaction Allowed identifier

Fields

NO_SALE_ALLOWED

`public static final java.lang.String` **`NO_SALE_ALLOWED`**

No Sale Transaction Allowed identifier

CASH_TRANSACTION_ALLOWED

`public static final java.lang.String` **`CASH_TRANSACTION_ALLOWED`**

Cash Transaction Allowed identifier

LAYAWAY_ALLOWED

`public static final java.lang.String` **`LAYAWAY_ALLOWED`**

Layaway Transaction Allowed identifier

RETURNS_ALLOWED

`public static final java.lang.String` **`RETURNS_ALLOWED`**

Returns Allowed identifier

(continued from last page)

VOID_TRANSACTION_ALLOWED

`public static final java.lang.String` **`VOID_TRANSACTION_ALLOWED`**

Void Transaction Allowed identifier

TAX_CODE_REQUIRED

`public static final java.lang.String` **`TAX_CODE_REQUIRED`**

Tax code required identifier

OPERATOR_ID_REQUIRED

`public static final java.lang.String` **`OPERATOR_ID_REQUIRED`**

Operator ID required identifier (GSA)

CASH_SPEC_ALLOWED

`public static final java.lang.String` **`CASH_SPEC_ALLOWED`**

Cash Special Transaction Allowed identifier (GSA)

CASH_DOC_ALLOWED

`public static final java.lang.String` **`CASH_DOC_ALLOWED`**

Cash Document Transaction Allowed identifier (GSA)

CHARGE_PLAN_A_ALLOWED

`public static final java.lang.String` **`CHARGE_PLAN_A_ALLOWED`**

Charge Plan A Transaction Allowed identifier (GSA)

CHARGE_PLAN_B_ALLOWED

`public static final java.lang.String` **`CHARGE_PLAN_B_ALLOWED`**

Charge Plan B Transaction Allowed identifier (GSA)

CHARGE_PLAN_C_ALLOWED

`public static final java.lang.String` **`CHARGE_PLAN_C_ALLOWED`**

Charge Plan C Transaction Allowed identifier (GSA)

CHARGE_PLAN_D_ALLOWED

`public static final java.lang.String` **`CHARGE_PLAN_D_ALLOWED`**

Charge Plan D Transaction Allowed identifier (GSA)

COD_ALLOWED

`public static final java.lang.String` **`COD_ALLOWED`**

COD Transaction Allowed identifier (GSA)

SEND_ALLOWED

`public static final java.lang.String` **`SEND_ALLOWED`**

Send Transaction Allowed identifier (GSA)

PAYMENTS_ALLOWED

`public static final java.lang.String` **`PAYMENTS_ALLOWED`**

Payments Allowed identifier (GSA)

ALLOWANCE_ALLOWED

`public static final java.lang.String` **`ALLOWANCE_ALLOWED`**

Allowance Allowed identifier (GSA)

DISCOUNTS_ALLOWED

`public static final java.lang.String` **`DISCOUNTS_ALLOWED`**

Discounts Allowed identifier (GSA)

ORIGINAL_SALES_PERSON_REQUIRED

`public static final java.lang.String` **`ORIGINAL_SALES_PERSON_REQUIRED`**

Original Sales Person Required identifier (GSA)

TERMS_OF_SALE_REQUIRED

`public static final java.lang.String` **`TERMS_OF_SALE_REQUIRED`**

Terms of Sale Required identifier (GSA)

SUSPEND_TRANSACTION_ALLOWED

`public static final java.lang.String` **`SUSPEND_TRANSACTION_ALLOWED`**

Suspend Transaction Allowed identifier (GSA)

com.ibm.retail.AEF.data

Interface TransactionStatusProperties

public interface **TransactionStatusProperties**

`TransactionStatusProperties`
provides the `POSDataProvider`for the transaction status-related data and events provided through the `POSDataProvider`API. These properties are communicated to client listeners through the `TransactionStatusListener`interface. Individual property values for the current transaction are also accessible through the `POSDataProvider`interface.

See Also:
com.ibm.retail.AEF.event.TransactionStatusListener#transactionStatusEventOccurred,

com.ibm.retail.AEF.data.POSDataProvider#getPropertyValue

Field Summary

static java.lang.String	`ACCOUNT_NUMBER` Account number for this transaction
static java.lang.String	`CANCEL_ALL_ITEMS` Cancel all items option (GSA Layaway)
static java.lang.String	`CATEGORY` Category identifier
static java.lang.String	`DATE` Transaction creation date
static java.lang.String	`EXPECTING_ITEM` EXPECTING_ITEM substate received.
static java.lang.String	`ID` Transaction identifier
static java.lang.String	`ITEM_ALLOWANCE_ALLOWED` Item allowance allowed
static java.lang.String	`ITEM_DISCOUNT_ALLOWED` Item discount allowed
static java.lang.String	`ITEM_SALE_IN_PROGRESS` Item sale in progress.
static java.lang.String	`LAST_CREDIT_APPROVED` Last credit attempt approved.
static java.lang.String	`LAST_ITEM_ADDED` Last item added.
static java.lang.String	`MODIFIER` Transaction modifier

static java.lang.String	RETURN_PAYMENTS Return payments option (GSA Layaway)
static java.lang.String	TAX_REASON Tax reason for transaction tax changes
static java.lang.String	TAX_TYPE Tax types for transaction tax changes
static java.lang.String	TAX_VOIDED Is Tax change voided for transaction tax changes
static java.lang.String	TIME Transaction creation date
static java.lang.String	TRANSACTION_CATEGORY Transaction category (sales/non-sales)
static java.lang.String	TRANSACTION_DISCOUNT_ALLOWED Transaction discount allowed
static java.lang.String	TRANSACTION_TYPE Transaction type (no sale, cash, loan, etc)
static java.lang.String	VOID_LINE_ITEM_ALLOWED Void line item allowed
static java.lang.String	VOID_TRANSACTION_DISCOUNT_ALLOWED Transaction discount allowed

Fields

CATEGORY

`public static final java.lang.String` **`CATEGORY`**

Category identifier

ID

`public static final java.lang.String` **`ID`**

Transaction identifier

DATE

`public static final java.lang.String` **`DATE`**

Transaction creation date

TIME

`public static final java.lang.String` **`TIME`**

Transaction creation date

TRANSACTION_CATEGORY

`public static final java.lang.String` **`TRANSACTION_CATEGORY`**

Transaction category (sales/non-sales)

TRANSACTION_TYPE

`public static final java.lang.String` **`TRANSACTION_TYPE`**

Transaction type (no sale, cash, loan, etc)

ACCOUNT_NUMBER

`public static final java.lang.String` **`ACCOUNT_NUMBER`**

Account number for this transaction

TAX_TYPE

`public static final java.lang.String` **`TAX_TYPE`**

Tax types for transaction tax changes

TAX_REASON

`public static final java.lang.String` **`TAX_REASON`**

Tax reason for transaction tax changes

TAX_VOIDED

`public static final java.lang.String` **`TAX_VOIDED`**

Is Tax change voided for transaction tax changes

RETURN_PAYMENTS

`public static final java.lang.String` **`RETURN_PAYMENTS`**

Return payments option (GSA Layaway)

CANCEL_ALL_ITEMS

`public static final java.lang.String` **`CANCEL_ALL_ITEMS`**

Cancel all items option (GSA Layaway)

MODIFIER

`public static final java.lang.String` **`MODIFIER`**

Transaction modifier

ITEM_ALLOWANCE_ALLOWED

`public static final java.lang.String` **`ITEM_ALLOWANCE_ALLOWED`**

Item allowance allowed

(continued from last page)

ITEM_DISCOUNT_ALLOWED

`public static final java.lang.String` **`ITEM_DISCOUNT_ALLOWED`**

Item discount allowed

VOID_LINE_ITEM_ALLOWED

`public static final java.lang.String` **`VOID_LINE_ITEM_ALLOWED`**

Void line item allowed

TRANSACTION_DISCOUNT_ALLOWED

`public static final java.lang.String` **`TRANSACTION_DISCOUNT_ALLOWED`**

Transaction discount allowed

VOID_TRANSACTION_DISCOUNT_ALLOWED

`public static final java.lang.String` **`VOID_TRANSACTION_DISCOUNT_ALLOWED`**

Transaction discount allowed

LAST_CREDIT_APPROVED

`public static final java.lang.String` **`LAST_CREDIT_APPROVED`**

Last credit attempt approved. This is only set to false when the appropriate credit tender action is called, and will then be set true if the credit is approved.

ITEM_SALE_IN_PROGRESS

`public static final java.lang.String` **`ITEM_SALE_IN_PROGRESS`**

Item sale in progress. This property is set to true in LineItemDetectorImpl when a line item is added to the transaction and set to false when the unit of work property change is received.

LAST_ITEM_ADDED

`public static final java.lang.String` **`LAST_ITEM_ADDED`**

Last item added. This is only set to false when the appropriate item entry action is called, and will then be set true if the item is added.

EXPECTING_ITEM

`public static final java.lang.String` **`EXPECTING_ITEM`**

EXPECTING_ITEM substate received. This is only set to false when the appropriate item entry action is called, and will then be set true if an EXPECTING_ITEM or EXPECTING_COUPON substate is received. This is a workaround for an ACE bug (defect 16154) where the UOW is received prior to the final substate change.

com.ibm.retail.AEF.data

Interface TransactionTotalsProperties

All Superinterfaces:
POSDataProperties

public interface **TransactionTotalsProperties**
extends POSDataProperties

`TransactionTotalsProperties`
provides `POSDataProvider`property names for the totals associated with a transaction. These properties are communicated to client listeners through the `TransactionTotalsListener`interface. Individual property values for the current transaction are also accessible through the `POSDataProvider`interface.

See Also:
com.ibm.retail.AEF.event.TransactionTotalsListener TransactionTotalsListener,

com.ibm.retail.AEF.data.POSDataProvider#getPropertyValue getPropertyValue

Field Summary

static java.lang.String	`AMOUNT_DUE` Total Amount Due
static java.lang.String	`CATEGORY` Category identifier
static java.lang.String	`CHANGE_DUE` Change due
static java.lang.String	`COUPON_TOTAL` Coupon total
static java.lang.String	`FOOD_STAMP_BALANCE` Food Stamp Balance
static java.lang.String	`FOOD_STAMP_CHANGE` Food Stamp Change
static java.lang.String	`FOOD_STAMP_TOTAL` Food Stamp Total
static java.lang.String	`FOREIGN_BALANCE_DUE` Foreign balance
static java.lang.String	`LAYAWAY_BALANCE_DUE` Layaway balance due
static java.lang.String	`LAYAWAY_DEPOSIT` Layaway deposit

static java.lang.String	LAYAWAY_FEE Layaway fee
static java.lang.String	SUB_TOTAL Transaction Subtotal
static java.lang.String	TAX Tax total
static java.lang.String	TENDER_EXCHANGED Tender Exchanged
static java.lang.String	TOTAL Transaction Total
static java.lang.String	TOTAL_COUPONS Number of coupons in transaction
static java.lang.String	TOTAL_ITEMS Number of items in transaction
static java.lang.String	TOTAL_SAVINGS Total savings from discounts

Fields

CATEGORY

`public static final java.lang.String` **`CATEGORY`**

Category identifier

TOTAL

`public static final java.lang.String` **`TOTAL`**

Transaction Total

SUB_TOTAL

`public static final java.lang.String` **`SUB_TOTAL`**

Transaction Subtotal

TOTAL_SAVINGS

`public static final java.lang.String` **`TOTAL_SAVINGS`**

Total savings from discounts

TAX

`public static final java.lang.String` **`TAX`**

Tax total

AMOUNT_DUE

`public static final java.lang.String` **`AMOUNT_DUE`**

Total Amount Due

CHANGE_DUE

`public static final java.lang.String` **`CHANGE_DUE`**

Change due

FOOD_STAMP_TOTAL

`public static final java.lang.String` **`FOOD_STAMP_TOTAL`**

Food Stamp Total

FOOD_STAMP_BALANCE

`public static final java.lang.String` **`FOOD_STAMP_BALANCE`**

Food Stamp Balance

FOOD_STAMP_CHANGE

`public static final java.lang.String` **`FOOD_STAMP_CHANGE`**

Food Stamp Change

TENDER_EXCHANGED

`public static final java.lang.String` **`TENDER_EXCHANGED`**

Tender Exchanged

COUPON_TOTAL

`public static final java.lang.String` **`COUPON_TOTAL`**

Coupon total

TOTAL_ITEMS

`public static final java.lang.String` **`TOTAL_ITEMS`**

Number of items in transaction

TOTAL_COUPONS

`public static final java.lang.String` **`TOTAL_COUPONS`**

Number of coupons in transaction

LAYAWAY_BALANCE_DUE

`public static final java.lang.String` **`LAYAWAY_BALANCE_DUE`**

Layaway balance due

(continued from last page)

LAYAWAY_FEE

`public static final java.lang.String` **`LAYAWAY_FEE`**

Layaway fee

LAYAWAY_DEPOSIT

`public static final java.lang.String` **`LAYAWAY_DEPOSIT`**

Layaway deposit

FOREIGN_BALANCE_DUE

`public static final java.lang.String` **`FOREIGN_BALANCE_DUE`**

Foreign balance

com.ibm.retail.AEF.data

Interface WorkstationStatusProperties

public interface **WorkstationStatusProperties**

`WorkstationStatusProperties`
contains the `POSDataProvider`property names for workstation attributes. These properties are communicated to client listeners through the `WorkstationStatusListener`interface. Individual property values for the current status are also accessible through the `POSDataProvider`interface.

See Also:
com.ibm.retail.AEF.event.WorkstationStatusListener WorkstationStatusListener,

com.ibm.retail.AEF.data.POSDataProvider#getPropertyValue getPropertyValue

Field Summary

static java.lang.String	`CATEGORY` Category identifier
static java.lang.String	`MESSAGE_PENDING` Message pending identifier
static java.lang.String	`NONSALES_TRANSACTION_IN_PROGRESS` NonSales Transaction in progress identifier
static java.lang.String	`OFFLINE_MODE` Offline mode identifier
static java.lang.String	`OPTIONS_LOADING_IN_PROGRESS` Options Loading in Progress
static java.lang.String	`QUEUE_LOCKED` Queue locked identifier
static java.lang.String	`REENTRY_OFFLINE_TRANSACTION` Reentry of Offline Transaction identifier
static java.lang.String	`SALES_TRANSACTION_IN_PROGRESS` Sales Transaction in progress identifier
static java.lang.String	`SIGNON_STATUS` Signon Status, values=true,false,secureMode
static java.lang.String	`TERMINAL_DISABLED` Terminal is disabled/enabled
static java.lang.String	`TERMINAL_NUMBER` Register Number identifier
static java.lang.String	`TERMINAL_STATUS` Current workstation status event identifier

static java.lang.String	TRAINING_MODE Training mode identifier
static java.lang.String	TRANSACTION_IN_PROGRESS Transaction in progress identifier
static java.lang.String	TRANSACTION_NUMBER Transaction Number identifier
static java.lang.String	UNIT_OF_WORK Unit of Work Marker

Fields

CATEGORY

`public static final java.lang.String` **`CATEGORY`**

Category identifier

TERMINAL_STATUS

`public static final java.lang.String` **`TERMINAL_STATUS`**

Current workstation status event identifier

TERMINAL_NUMBER

`public static final java.lang.String` **`TERMINAL_NUMBER`**

Register Number identifier

TRANSACTION_NUMBER

`public static final java.lang.String` **`TRANSACTION_NUMBER`**

Transaction Number identifier

SALES_TRANSACTION_IN_PROGRESS

`public static final java.lang.String` **`SALES_TRANSACTION_IN_PROGRESS`**

Sales Transaction in progress identifier

NONSALES_TRANSACTION_IN_PROGRESS

`public static final java.lang.String` **`NONSALES_TRANSACTION_IN_PROGRESS`**

NonSales Transaction in progress identifier

TRANSACTION_IN_PROGRESS

`public static final java.lang.String` **`TRANSACTION_IN_PROGRESS`**

Transaction in progress identifier

(continued from last page)

REENTRY_OFFLINE_TRANSACTION

`public static final java.lang.String` **`REENTRY_OFFLINE_TRANSACTION`**

Reentry of Offline Transaction identifier

TRAINING_MODE

`public static final java.lang.String` **`TRAINING_MODE`**

Training mode identifier

OFFLINE_MODE

`public static final java.lang.String` **`OFFLINE_MODE`**

Offline mode identifier

MESSAGE_PENDING

`public static final java.lang.String` **`MESSAGE_PENDING`**

Message pending identifier

QUEUE_LOCKED

`public static final java.lang.String` **`QUEUE_LOCKED`**

Queue locked identifier

UNIT_OF_WORK

`public static final java.lang.String` **`UNIT_OF_WORK`**

Unit of Work Marker

OPTIONS_LOADING_IN_PROGRESS

`public static final java.lang.String` **`OPTIONS_LOADING_IN_PROGRESS`**

Options Loading in Progress

SIGNON_STATUS

`public static final java.lang.String` **`SIGNON_STATUS`**

Signon Status, values=true,false,secureMode

TERMINAL_DISABLED

`public static final java.lang.String` **`TERMINAL_DISABLED`**

Terminal is disabled/enabled

Package
com.ibm.retail.AEF.event

Provides listener and event interfaces used by the AEF `POSDataProvider`.

The event package defines the events and listener interfaces used by the POSDataProvider. The `POSDataProvider`allows applications to monitor the data events associated with a POS application. Data events provide information which reflects the current state of the POS application including transaction totals, line items, and customer loyalty information.

Event information is obtained through a standard event listener registration and notification process. Listeners must implement the appropriate listener interface and register with the `POSDataProvider`through an `addListener`method. When an event occurs (e.g., the transaction totals change), the client listener is notified by a call through the listener interface. Event objects contain the details of the event.

Usage Notes:

Handling Missing Event Properties

AEF event objects provide a means of passing data from a POS terminal sales application to the AEF POSDataProvider listeners. Since POS applications differ in their data requirements, the event objects provide a flexible interface that allows for both adding new attibutes and removing undefined properties from an event. For example, while the `TransactionTotalsEvent`provides an accessor for `getFoodStampTotal`, some POS applications may not provide this data. In this case, a `null`value is returned by the `getFoodStampTotal`method. If a method with a boolean return value is invoked and the property is missing from the event object, an `AEFException`is thrown by the event interface with an error code of `AEFConst.NO_SUCH_PROPERTY`. It is the responsibility of the client listener to determine the appropriate action for undefined event properties.

Accessing Additional Event Properties

An AEF POS application may add additional properties to an event without electing to extend the event interface with a new accessor. All data is contained within the event as *property-value*pairs and may be accessed through the getPropertymethod of the `POSAppEventElement`base class. The getPropertyEntriesmethod provides an `Iterator`of all *property-value*pairs.

Sample Usage:

```
sample code here...
```

com.ibm.retail.AEF.event

Class AEFPropertyChangeEvent

```
java.lang.Object
  +-java.util.EventObject
    +-java.beans.PropertyChangeEvent
      +-com.ibm.retail.AEF.event.AEFPropertyChangeEvent
```

public class **AEFPropertyChangeEvent**
extends java.beans.PropertyChangeEvent

An `AEFPropertyChangeEvent`is created when a property contained in the AEF `POSDataProvider`is modified. AEF properties are identified by a *property name*and are classified into *categories*. Categories allow a listener to monitor either a single property (e.g, `TransactionTotalsProperties.TOTAL_ITEMS`) or an entire category of properties (e.g., `TransactionTotalsProperties.CATEGORY`).

See Also:
com.ibm.retail.AEF.data.POSDataProvider#addAEFPropertyChangeListener addAEFPropertyChangeListener

Field Summary

`java.lang.String`	`categoryName`

Fields inherited from : class `java.util.EventObject`
`source`

Constructor Summary

`AEFPropertyChangeEvent(java.lang.Object source,java.lang.String source,java.lang.String source,java.lang.Object source,java.lang.Object source)`

Construct an AEFPropertyChangeEvent

Method Summary

`java.lang.String`	`getCategoryName()` Get the category name of this AEFPropertyChangeEvent.
`void`	`setCategoryName(java.lang.String categoryName)` Set the category name of this AEFPropertyChangeEvent.

Methods inherited from : class `java.beans.PropertyChangeEvent`
`getNewValue, getOldValue, getPropagationId, getPropertyName, setPropagationId`

Methods inherited from : class `java.util.EventObject`
`getSource, toString`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

categoryName

```
protected java.lang.String categoryName
```

Constructors

AEFPropertyChangeEvent

```
public AEFPropertyChangeEvent(java.lang.Object source,
                              java.lang.String categoryName,
                              java.lang.String propertyName,
                              java.lang.Object oldValue,
                              java.lang.Object newValue)
```

Construct an AEFPropertyChangeEvent

Parameters:

`source` -
the object firing this event
`categoryName` -
the category of the property change
`propertyName` -
the name of the propery changed
`oldValue` -
the previous value
`newValue` -
the new value

Methods

getCategoryName

```
public java.lang.String getCategoryName()
```

Get the category name of this AEFPropertyChangeEvent.

Returns:

the category name for this event. Among the possible values are:
CouponProperties.CATEGORY
CustomerProperties.CATEGORY
DiscountProperties.CATEGORY
ItemSalesProperties.CATEGORY
OptionsProperties.CATEGORY
POSDeviceProperties.CATEGORY
TenderProperties.CATEGORY
TransactionStatusProperties.CATEGORY
TransactionTotalsProperties.CATEGORY

setCategoryName

`public void` **`setCategoryName`**`(java.lang.String categoryName)`

Set the category name of this AEFPropertyChangeEvent.

Parameters:

`categoryName -`
the category name for this event

com.ibm.retail.AEF.event

Interface AEFPropertyChangeListener

All Known Implementing Classes:
AEFPropertyListenerProxyImpl, AbstractPropertyCondition

---

public interface **AEFPropertyChangeListener**
extends java.rmi.Remote

`AEFPropertyChangeListener`
interface is the interface for receiving AEF property change events from the `POSDataProvider`API.

See Also:
com.ibm.retail.AEF.data.POSDataProvider#addAEFPropertyChangeListener addAEFPropertyChangeListener

---

Method Summary

void	`propertyChanged(AEFPropertyChangeEvent evt)` An AEF property has changed.

Methods

propertyChanged

```
public void propertyChanged(AEFPropertyChangeEvent evt)
                    throws java.rmi.RemoteException
```

An AEF property has changed.

Parameters:
`evt` -
contains details of the event

Exceptions:
`RemoteException` -
if remote access fails

com.ibm.retail.AEF.event

Interface CashReceiptEvent

All Superinterfaces:
POSAppEvent, POSAppEventElement

public interface **CashReceiptEvent**
extends POSAppEvent

The `CashReceiptEvent`encapsulates cash receipt printer data obtained from the terminal session.

Method Summary

`int`	`getLineFeeds()` Get number of line feeds.
`java.util.Collection`	`getPrintLines()` Returns the print lines associated with this event.
`java.util.Collection`	`getRawPrintLines()` Return the collection of raw print lines for the event.
`boolean`	`isPaperCut()` Return true if this a paper cut for the receipt printer.

Methods

getPrintLines

```
public java.util.Collection getPrintLines()
```

Returns the print lines associated with this event. All special characters (escape sequences) have been filtered out.

Returns:
`Collection`
of strings containing the print lines. - returns empty collection if no print lines exist

getRawPrintLines

```
public java.util.Collection getRawPrintLines()
```

Return the collection of raw print lines for the event.

Returns:
`Collection`
of strings containing the raw print lines. - returns empty collection if no print lines exist

isPaperCut

```
public boolean isPaperCut()
                    throws AEFException
```

Return true if this a paper cut for the receipt printer.

(continued from last page)

Returns:

true if paper cut

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

getLineFeeds

```
public int getLineFeeds()
                 throws AEFException
```

Get number of line feeds.

Returns:

number of line feeds

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

com.ibm.retail.AEF.event

Interface CashReceiptListener

All Superinterfaces:
POSAppEventListener

All Known Implementing Classes:
CashReceiptListenerProxy

---

public interface **CashReceiptListener**
extends POSAppEventListener

Listener interface for receiving cash receipt printer events from the terminal session.

See Also:
com.ibm.retail.AEF.data.POSDataProvider#addCashReceiptListener addCashReceiptListener

---

Method Summary

`void`	`linePrinted(CashReceiptEvent evt)` A line was printed on the receipt.

Methods

linePrinted

```
public void linePrinted(CashReceiptEvent evt)
                 throws java.rmi.RemoteException
```

A line was printed on the receipt.

Parameters:
`evt` -
contains details of the print line

Exceptions:
`RemoteException` -
if a listener can not be notified

com.ibm.retail.AEF.event

Interface CouponEvent

All Superinterfaces:
LineItemEvent, POSAppEvent, POSAppEventElement

public interface **CouponEvent**
extends LineItemEvent

A `CouponEvent`is generated when a coupon is applied to an item or transaction. A coupon may be applied by scanning a manufacturer or store coupon, or it may be applied automatically through a loyalty program.

Method Summary

int	`getAgeRestriction()` Returns the age restriction in years for an age restricted item.
java.lang.String	`getCouponType()` Returns the coupon type.
java.lang.String	`getDealPrice()` Returns the deal price.
int	`getDealQuantity()` Returns the item deal quantity.
java.lang.String	`getItemID()` Returns the item code used to sell the item (e.
java.lang.String	`getItemIDQualifier()` Returns the item code qualifier.
java.lang.String	`getManufacturerNumber()` Returns the manufacturer's number for the coupon.
java.lang.String	`getMultiPricingGroup()` Returns the multi pricing group used for deal pricing.
java.lang.String	`getPricingMethod()` Gets the pricing method for this item.
int	`getQuantity()` Returns the quantity sold.
java.lang.String	`getReducedPrice()` Returns the reduced price used for deal pricing.
java.util.Collection	`getRestrictedPeriods()` Returns a list of restricted/unrestricted time periods for the item.

java.lang.String	getUnitPrice() Returns the coupon unit price (without currency symbol).
java.lang.String	getValue() Returns the extended value of the coupon.
java.lang.String	getWeight() Returns the item sale weight for an item sold by weight (e.
boolean	isItemRepeatAllowed() Indicates whether item repeat is allowed for this item.
boolean	isTimeRestricted() Indicates whether the item is time restricted.
boolean	isWICEligible() Indicates whether the item is WIC eligible.
boolean	reducesFoodstampBalanceDue() Indicates whether the coupon reduces the foodstamp balance due.
boolean	reducesTaxDue() Indicates whether the coupon reduces the taxable transaction amount.

Methods

getItemID

`public java.lang.String` **`getItemID`**`()`

Returns the item code used to sell the item (e.g., the UPC number)

Returns:

item identifier - returns null if the property is undefined

getItemIDQualifier

`public java.lang.String` **`getItemIDQualifier`**`()`

Returns the item code qualifier. Indicates the format of the item code (may be scanned, keyed). For example, IBM 4690 Supermarket Application, the values may be:
ScannedItemCodeKeyedItemCodeItemLookupKeyedLinkedItemCodeWandedItemCode

Returns:

item ID qualifier - returns null if the property is undefined

getCouponType

`public java.lang.String` **`getCouponType`**`()`

Returns the coupon type. Coupon types differentiate different forms of coupons. Eaxmples are "Store", "Manufacturer", and "Electronic".

Returns:

(continued from last page)

coupont type - returns null if the property is undefined

getUnitPrice

```
public java.lang.String getUnitPrice()
```

Returns the coupon unit price (without currency symbol).

Returns:

formatted string representation of the item price per unit - returns null if the property is undefined

getQuantity

```
public int getQuantity()
                throws AEFException
```

Returns the quantity sold.

Returns:

quantity sold as an int value

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

getDealQuantity

```
public int getDealQuantity()
                    throws AEFException
```

Returns the item deal quantity. A deal quantity is used for items priced in multiples (e.g, 3 for 1.00)

Returns:

deal quantity as int value

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

getWeight

```
public java.lang.String getWeight()
```

Returns the item sale weight for an item sold by weight (e.g., produce).

Returns:

item sale weight as a formatted string. - returns null if the property is undefined

getValue

```
public java.lang.String getValue()
```

Returns the extended value of the coupon.

Returns:

(continued from last page)

string representation of the extended value. - returns null if the property is undefined

getDealPrice

`public java.lang.String getDealPrice()`

Returns the deal price. Deal prices are defined for items sold as multiples (e.g., 3 for 1.00)

Returns:

string representation of the deal price - returns null if the property is undefined

getReducedPrice

`public java.lang.String getReducedPrice()`

Returns the reduced price used for deal pricing.

Returns:

string representation of the reduced price - returns null if the property is undefined

getMultiPricingGroup

`public java.lang.String getMultiPricingGroup()`

Returns the multi pricing group used for deal pricing.

Returns:

multi pricing group - returns null if the property is undefined

getAgeRestriction

```
public int getAgeRestriction()
                      throws AEFException
```

Returns the age restriction in years for an age restricted item.

Returns:

age restriction in years for an age restricted item

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

isTimeRestricted

`public boolean isTimeRestricted()`

Indicates whether the item is time restricted.

Returns:

true if the item is time restricted

(continued from last page)

reducesFoodstampBalanceDue

```
public boolean reducesFoodstampBalanceDue()
                                    throws AEFException
```

Indicates whether the coupon reduces the foodstamp balance due.

Returns:

true if the coupon reduces the foodstamp balance due.

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

---

isWICEligible

```
public boolean isWICEligible()
                      throws AEFException
```

Indicates whether the item is WIC eligible.

Returns:

true if the item is WIC eligible

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

---

isItemRepeatAllowed

```
public boolean isItemRepeatAllowed()
                             throws AEFException
```

Indicates whether item repeat is allowed for this item. Item repeat is used by operators to indicate an item is sold multiple times without rescanning the item.

Returns:

true if item repeat is allowed

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

---

reducesTaxDue

```
public boolean reducesTaxDue()
                      throws AEFException
```

Indicates whether the coupon reduces the taxable transaction amount.

Returns:

true if the coupon reduces the taxable transaction amount.

Exceptions:

(continued from last page)

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

getPricingMethod

`public java.lang.String` **`getPricingMethod`**`()`

Gets the pricing method for this item.

Returns:

pricing method - returns null if the property is undefined

getRestrictedPeriods

`public java.util.Collection` **`getRestrictedPeriods`**`()`

Returns a list of restricted/unrestricted time periods for the item.

Returns:
`Collection`
of `TimeInterval`objects or an empty collection.

See Also:

com.ibm.retail.AEF.event.TimeInterval TimeInterval

getManufacturerNumber

`public java.lang.String` **`getManufacturerNumber`**`()`

Returns the manufacturer's number for the coupon.

Returns:

manufacturer number - returns null if the property is undefined

com.ibm.retail.AEF.event

Interface CouponListener

All Superinterfaces:
POSAppEventListener

All Known Implementing Classes:
CouponListenerProxy

---

public interface **CouponListener**
extends POSAppEventListener

Listener interface for receiving `CouponEvents`from the terminal session. A `CouponEvent`is fired whenever a coupon is redeemed or applied to the POS transaction to reduce the transaction amount.

See Also:
com.ibm.retail.AEF.data.POSDataProvider#addCouponListener addCouponListener

---

Method Summary

`void`	`couponApplied(CouponEvent evt)` A coupon was added or removed from a transaction.

Methods

couponApplied

```
public void couponApplied(CouponEvent evt)
                   throws java.rmi.RemoteException
```

A coupon was added or removed from a transaction.

Parameters:
`evt` -
contains coupon details

Exceptions:
`RemoteException` -
if remote access fails

com.ibm.retail.AEF.event

Interface CustomerEvent

All Superinterfaces:
POSAppEvent, POSAppEventElement

public interface **CustomerEvent**
extends POSAppEvent

A `CustomerEvent`is generated when a customer is identified by scanning or swiping a customer identification or loyalty card. The `CustomerEvent`contains information about the customer defined for the current transaction.

Method Summary

`java.lang.String`	`getAddress1()` Get address line 1 of this Customer.
`java.lang.String`	`getAddress2()` Get address line 2 of this Customer.
`java.lang.String`	`getCity()` Returns the city of this Customer.
`java.lang.String`	`getContact()` Returns the contact number of this Customer.
`java.lang.String`	`getEmail()` Returns the email address of this Customer.
`java.lang.String`	`getFax()` Returns the fax number of this Customer.
`java.lang.String`	`getID()` Returns the ID of this customer (Customer Loyalty Number).
`java.lang.String`	`getIDExpirationDate()` Returns the ID expiration date.
`java.lang.String`	`getIDType()` Returns the ID type ('primary' or 'alternate').
`java.util.Collection`	`getMessages()` Returns all loyalty messages generated for this customer.
`java.lang.String`	`getName()` Returns the full name of this Customer.
`java.lang.String`	`getPhone()` Returns the phone number of this Customer.

java.util.Collection	getPointsBalances() Returns list of the points balances for all accounts and clubs.
java.util.Collection	getPointsTotals() Returns list of the points totals for all accounts and clubs.
java.lang.String	getState() Returns the state of this Customer.
java.util.Collection	getTargetedCoupons() Returns list of targeted coupons.
java.lang.String	getYTDPoints() Returns the year-to-date points of this Customer.
java.lang.String	getYTDSaved() Returns the year-to-date savings of this Customer.
java.lang.String	getZip() Returns the zipcode of this Customer.

Methods

getID

`public java.lang.String **getID**()`

Returns the ID of this customer (Customer Loyalty Number).

Returns:

customer ID - null if the property is undefined in this event

getIDType

`public java.lang.String **getIDType**()`

Returns the ID type ('primary' or 'alternate').

Returns:

ID type - null if the property is undefined in this event

getIDExpirationDate

`public java.lang.String **getIDExpirationDate**()`

Returns the ID expiration date.

Returns:

string representation of expiration date - null if the property is undefined in this event

(continued from last page)

getName

`public java.lang.String getName()`

Returns the full name of this Customer.

Returns:

customer full name - null if the property is undefined in this event

getAddress1

`public java.lang.String getAddress1()`

Get address line 1 of this Customer.

Returns:

address1 - null if the property is undefined in this event

getAddress2

`public java.lang.String getAddress2()`

Get address line 2 of this Customer.

Returns:

address2 - null if the property is undefined in this event

getCity

`public java.lang.String getCity()`

Returns the city of this Customer.

Returns:

city - null if the property is undefined in this event

getState

`public java.lang.String getState()`

Returns the state of this Customer.

Returns:

state - null if the property is undefined in this event

getZip

`public java.lang.String getZip()`

Returns the zipcode of this Customer.

Returns:

zipcode - null if the property is undefined in this event

(continued from last page)

getPhone

`public java.lang.String getPhone()`

Returns the phone number of this Customer.

Returns:

phone number - null if the property is undefined in this event

getContact

`public java.lang.String getContact()`

Returns the contact number of this Customer.

Returns:

contact name or number - null if the property is undefined in this event

getFax

`public java.lang.String getFax()`

Returns the fax number of this Customer.

Returns:

Fax number - null if the property is undefined in this event

getEmail

`public java.lang.String getEmail()`

Returns the email address of this Customer.

Returns:

email address - null if the property is undefined in this event

getYTDPoints

`public java.lang.String getYTDPoints()`

Returns the year-to-date points of this Customer.

Returns:

string representation of year-to-date points - null if the property is undefined in this event

getYTDSaved

`public java.lang.String getYTDSaved()`

Returns the year-to-date savings of this Customer.

Returns:

string representation of year-to-date savings - null if the property is undefined in this event

(continued from last page)

getTargetedCoupons

`public java.util.Collection` **`getTargetedCoupons`**`()`

Returns list of targeted coupons.

Returns:
`Collection`
of strings defining coupon codes of targeted coupons. - null if the property is undefined in this event

getPointsTotals

`public java.util.Collection` **`getPointsTotals`**`()`

Returns list of the points totals for all accounts and clubs.

Returns:
`Collection`
of `PointsTotal`objects - null if the property is undefined in this event

See Also:

com.ibm.retail.AEF.event.PointsTotal PointsTotal

getPointsBalances

`public java.util.Collection` **`getPointsBalances`**`()`

Returns list of the points balances for all accounts and clubs.

Returns:
`Collection`
of `PointsTotal`objects - null if the property is undefined in this event

See Also:

com.ibm.retail.AEF.event.PointsTotal PointsTotal

getMessages

`public java.util.Collection` **`getMessages`**`()`

Returns all loyalty messages generated for this customer. Loyalty messages may be used to inform customer of loyalty account information.

Returns:
`Collection`
of strings containing customer messages. - null if the property is undefined in this event

com.ibm.retail.AEF.event

Interface CustomerListener

All Superinterfaces:
POSAppEventListener

All Known Implementing Classes:
CustomerListenerProxy

---

public interface **CustomerListener**
extends POSAppEventListener

Listener interface for receiving `CustomerEvent`from the terminal session. A `CustomerEvent`is fired whenever a customer is identified by scanning or swiping a customer loyalty card.

See Also:
com.ibm.retail.AEF.data.POSDataProvider#addCustomerListener addCustomerListener

---

Method Summary

`void`	`customerCardEntered(CustomerEvent evt)` A customer card number was entered by scanning or swiping the loyalty card.

Methods

customerCardEntered

```
public void customerCardEntered(CustomerEvent evt)
                        throws java.rmi.RemoteException
```

A customer card number was entered by scanning or swiping the loyalty card.

Parameters:
`evt` -
contains customer details

com.ibm.retail.AEF.event

Interface DepartmentDefinition

All Superinterfaces:
POSAppEventElement

public interface **DepartmentDefinition**
extends POSAppEventElement

A `DepartmentDefinition`is a simple data object to contain information about POS departments.

`DepartmentDefinitions`are accessible from an `OptionsEvent`.

See Also:
com.ibm.retail.AEF.event.OptionsEvent#getDepartmentDefinitions getDepartmentDefinitions

Field Summary

static java.lang.String	`DEPT_DESCRIPTION` Property name identifier for: Description of the department
static java.lang.String	`DEPT_KEY` Property name identifier for: Function code or POS key used to identify this department
static java.lang.String	`DEPT_NUMBER` Property name identifier for: Department number or identifier
static java.lang.String	`MANAGER_OVERRIDE` Property value identifier for manager override
static java.lang.String	`MANUFACTURER_COUPON_ALLOWED` Property identifier for whether manufacturer coupon is allowed.
static java.lang.String	`OPERATOR_OVERRIDE` Property value identifier for operator override
static java.lang.String	`PRICE_ALLOWED` Property identifier for whether price change is allowed.
static java.lang.String	`REFUND_ALLOWED` Property identifier for whether refund is allowed.
static java.lang.String	`STORE_COUPON_ALLOWED` Property identifier for whether store coupon is allowed.
static java.lang.String	`TOGGLE_FOODSTAMP_ALLOWED` Property identifier for whether toggle of foodstampability is allowed.
static java.lang.String	`TOGGLE_TAX_ALLOWED` Property identifier for whether toggle of taxability is allowed.

Method Summary

java.lang.String	getDepartmentNumber() Get the department number.
java.lang.String	getDescription() Get the description of the department.
java.lang.String	getKey() Get the function code or POS key used to identify this department.
java.lang.String	getManufacturerCouponAllowed() Get the value of the manufacturer coupon allowed property.
java.lang.String	getPriceAllowed() Get the value of the price allowed property.
java.lang.String	getRefundAllowed() Get the value of the toggle foodstampability allowed property.
java.lang.String	getStoreCouponAllowed() Get the value of the store coupon allowed property.
java.lang.String	getToggleFoodstampAllowed() Get the value of the toggle taxability allowed property.
java.lang.String	getToggleTaxAllowed() Get the value of the tax/no tax allowed property.

Fields

DEPT_NUMBER

`public static final java.lang.String` **`DEPT_NUMBER`**

Property name identifier for: Department number or identifier

DEPT_DESCRIPTION

`public static final java.lang.String` **`DEPT_DESCRIPTION`**

Property name identifier for: Description of the department

DEPT_KEY

`public static final java.lang.String` **`DEPT_KEY`**

Property name identifier for: Function code or POS key used to identify this department

TOGGLE_TAX_ALLOWED

`public static final java.lang.String` **`TOGGLE_TAX_ALLOWED`**

Property identifier for whether toggle of taxability is allowed. Possible values are "true"|"false"|"managerOverride"|"operatorOverride"

TOGGLE_FOODSTAMP_ALLOWED

`public static final java.lang.String` **`TOGGLE_FOODSTAMP_ALLOWED`**

Property identifier for whether toggle of foodstampability is allowed. Possible values are "true"|"false"|"managerOverride"|"operatorOverride"

REFUND_ALLOWED

`public static final java.lang.String` **`REFUND_ALLOWED`**

Property identifier for whether refund is allowed. Possible values are "true"|"false"|"managerOverride"|"operatorOverride"

STORE_COUPON_ALLOWED

`public static final java.lang.String` **`STORE_COUPON_ALLOWED`**

Property identifier for whether store coupon is allowed. Possible values are "true"|"false"|"managerOverride"|"operatorOverride"

MANUFACTURER_COUPON_ALLOWED

`public static final java.lang.String` **`MANUFACTURER_COUPON_ALLOWED`**

Property identifier for whether manufacturer coupon is allowed. Possible values are "true"|"false"|"managerOverride"|"operatorOverride"

PRICE_ALLOWED

`public static final java.lang.String` **`PRICE_ALLOWED`**

Property identifier for whether price change is allowed. Possible values are "true"|"false"|"managerOverride"|"operatorOverride"

MANAGER_OVERRIDE

`public static final java.lang.String` **`MANAGER_OVERRIDE`**

Property value identifier for manager override

OPERATOR_OVERRIDE

`public static final java.lang.String` **`OPERATOR_OVERRIDE`**

Property value identifier for operator override

Methods

getDepartmentNumber

`public java.lang.String` **`getDepartmentNumber`**`()`

Get the department number.

Returns:

department number

getDescription

`public java.lang.String` **`getDescription`**`()`

Get the description of the department.

(continued from last page)

Returns:

department description

getKey

`public java.lang.String getKey()`

Get the function code or POS key used to identify this department.

Returns:

string repersentation of function code or POS key used for this department

getToggleTaxAllowed

`public java.lang.String getToggleTaxAllowed()`

Get the value of the tax/no tax allowed property.

Returns:

tax/no tax allowed property value.

Possible values are:
true
false
DepartmentDefinition.MANAGER_OVERRIDE
DepartmentDefinition.OPERATOR_OVERRIDE
null (undefined)

getToggleFoodstampAllowed

`public java.lang.String getToggleFoodstampAllowed()`

Get the value of the toggle taxability allowed property.

Returns:

toggle taxability allowed property

Possible values are:
true
false
DepartmentDefinition.MANAGER_OVERRIDE
DepartmentDefinition.OPERATOR_OVERRIDE
null (undefined)

getRefundAllowed

`public java.lang.String getRefundAllowed()`

Get the value of the toggle foodstampability allowed property.

Returns:

value of the toggle foodstampability allowed property.

Possible values are:
true
false
DepartmentDefinition.MANAGER_OVERRIDE
DepartmentDefinition.OPERATOR_OVERRIDE
null (undefined)

getStoreCouponAllowed

`public java.lang.String` **`getStoreCouponAllowed`**`()`

Get the value of the store coupon allowed property.

Returns:

value of the store coupon allowed property

Possible values are:
true
false
DepartmentDefinition.MANAGER_OVERRIDE
DepartmentDefinition.OPERATOR_OVERRIDE
null (undefined)

getManufacturerCouponAllowed

`public java.lang.String` **`getManufacturerCouponAllowed`**`()`

Get the value of the manufacturer coupon allowed property.

Returns:

value of the manufacturer coupon allowed property

Possible values are:
true
false
DepartmentDefinition.MANAGER_OVERRIDE
DepartmentDefinition.OPERATOR_OVERRIDE
null (undefined)

getPriceAllowed

`public java.lang.String` **`getPriceAllowed`**`()`

Get the value of the price allowed property.

Returns:

value of the price allowed property

Possible values are:
true
false
DepartmentDefinition.MANAGER_OVERRIDE
DepartmentDefinition.OPERATOR_OVERRIDE
null (undefined)

com.ibm.retail.AEF.event

Interface DiscountEvent

All Superinterfaces:
LineItemEvent, POSAppEvent, POSAppEventElement

public interface **DiscountEvent**
extends LineItemEvent

A `DiscountEvent`is generated by a POS transaction when a discount is applied to the transaction or an item. The `DiscountEvent`object contains details about the discount including the amount and type of the discount.

Method Summary

`java.lang.String`	`getAmount()` Returns the discount amount.
`java.lang.String`	`getAppliesTo()` Returns indication of what line item this discount is applied to.
`java.lang.String`	`getDiscountMethod()` Returns the discount method.
`java.lang.String`	`getDiscountRate()` Returns the discount rate (percentage off).
`java.lang.String`	`getDiscountReason()` Returns the reason for the discount (may be application specific).
`java.lang.String`	`getDiscountType()` Returns the type of discount (may be application specific).
`boolean`	`isTransactionDiscount()` Indicates if this is a transaction or line item discount.
`boolean`	`isVoided()` Indicates if this is discount was voided.
`boolean`	`reducesTaxBalanceDue()` Indicates whether the discount reduces the tax balance due.

Methods

getAppliesTo

`public java.lang.String` **`getAppliesTo()`**

Returns indication of what line item this discount is applied to. (Note: Used for line item discounts only) Examples are: "previous", "next", "1", "10".

(continued from last page)

Returns:

indication of what line item this discount is applied to - returns null if the property is undefined

getDiscountMethod

`public java.lang.String` **`getDiscountMethod`**`()`

Returns the discount method. The discount method is either "percent" or "allowance".

Returns:

discount method - returns null if the property is undefined

getDiscountRate

`public java.lang.String` **`getDiscountRate`**`()`

Returns the discount rate (percentage off). Applies only to discounts with discount method of "percent". Example: "25.0" for a 25% discount.

Returns:

string representation of the discount rate - returns null if the property is undefined

getAmount

`public java.lang.String` **`getAmount`**`()`

Returns the discount amount. Applies only to discounts with discount method of "allowance". Example: "0.50" for a fifty cent reduction.

Returns:

string representation of the discount amount - returns null if the property is undefined

getDiscountType

`public java.lang.String` **`getDiscountType`**`()`

Returns the type of discount (may be application specific).

Returns:

string representation of the discount type - returns null if the property is undefined

getDiscountReason

`public java.lang.String` **`getDiscountReason`**`()`

Returns the reason for the discount (may be application specific).

Returns:

reason for discount - returns null if the property is undefined

reducesTaxBalanceDue

```
public boolean reducesTaxBalanceDue()
                              throws AEFException
```

Indicates whether the discount reduces the tax balance due.

(continued from last page)

Returns:

true if the discount reduces the tax balance due.

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

isTransactionDiscount

```
public boolean isTransactionDiscount()
                              throws AEFException
```

Indicates if this is a transaction or line item discount.

Returns:

true if this is a transaction level discount, false if it is a line item discount.

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

isVoided

```
public boolean isVoided()
                 throws AEFException
```

Indicates if this is discount was voided.

Returns:

true if this is discount was voided.

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

com.ibm.retail.AEF.event

Interface DiscountListener

All Superinterfaces:
POSAppEventListener

All Known Implementing Classes:
DiscountListenerProxy

---

public interface **DiscountListener**
extends POSAppEventListener

Listener interface for receiving discount events from the terminal session. Discount events are generated as items are sold in a POS transaction and discounts are applied to reduce the total of the item or transaction.

See Also:
com.ibm.retail.AEF.data.POSDataProvider#addDiscountListener addDiscountListener

---

Method Summary

`void`	`discountApplied(DiscountEvent evt)` A discount was applied to an item or transaction.

Methods

discountApplied

```
public void discountApplied(DiscountEvent evt)
                    throws java.rmi.RemoteException
```

A discount was applied to an item or transaction.

Parameters:
`evt` -
contains details of the discount event

Exceptions:
`RemoteException` -
if a listener can not be notified

com.ibm.retail.AEF.event

Interface DiscountReasonCode

All Superinterfaces:
ReasonCode, POSAppEventElement

All Subinterfaces:
TransactionDiscountReasonCode

---

public interface **DiscountReasonCode**
extends ReasonCode

A DiscountReasonCode is a simple data object to contain information about a POS discount reason code.

A discount reason code contains a numeric code, a text description, and the rate (percentage) of the discount.

Field Summary

static java.lang.String	RATE Identifier for the reason code rate

Method Summary

java.lang.String	getRate() Get the rate value (percent-off).

Fields

RATE

`public static final java.lang.String RATE`

Identifier for the reason code rate

Methods

getRate

`public java.lang.String getRate()`

Get the rate value (percent-off).

Returns:

string representation of the rate (e.g. "25.0")

com.ibm.retail.AEF.event

Interface GenericEventListener

All Superinterfaces:
POSAppEventListener

All Known Implementing Classes:
GenericListenerProxy

---

public interface **GenericEventListener**
extends POSAppEventListener

`GenericEventListener`
is a general interface for listeners of `POSAppEvents`are generated by the `POSDataProvider`API. It is used primarily to support extensions to the event listener API where a specific listener interface is not needed.

See Also:
com.ibm.retail.AEF.data.POSDataProvider#addPOSAppEventListener addPOSAppEventListener

---

Method Summary

`void`	`eventOccurred(POSAppEvent evt)` A POS data event occurred.

Methods

eventOccurred

```
public void eventOccurred(POSAppEvent evt)
                    throws java.rmi.RemoteException
```

A POS data event occurred.

Parameters:
`evt -`
contains details of the event

Exceptions:
`RemoteException -`
if a listener can not be notified

com.ibm.retail.AEF.event

Interface ItemSalesEvent

All Superinterfaces:
LineItemEvent, POSAppEvent, POSAppEventElement

public interface **ItemSalesEvent**
extends LineItemEvent

An `ItemSalesEvent`is generated when an item is added or removed from a transaction. The `ItemSalesEvent`object contains details about the item sale.

Field Summary

static java.lang.String	`ITEM_SOLD` item sold action identifier

Method Summary

boolean	`enteredPriceUsed()` Indicates if the item price was overridden, or the item was price required.
int	`getAgeRestriction()` Returns the age restriction in years for an age restricted item.
java.lang.String	`getDealPrice()` Returns the deal price.
int	`getDealQuantity()` Returns the item deal quantity.
java.lang.String	`getExtendedPrice()` Returns the extended price.
java.lang.String	`getItemID()` Returns the item code used to sell the item (e.
java.lang.String	`getItemIDQualifier()` Returns the item code qualifier.
java.lang.String	`getLinkedItemID()` Get the linked item code (if any).
java.lang.String	`getLinkedItemIDQualifier()` Get the linked item code qualifier.
java.lang.String	`getMultiPricingGroup()` Returns the multi pricing group used for deal pricing.

java.lang.String	getPricingMethod() Gets the pricing method for this item.
int	getQuantity() Returns the quantity sold.
java.lang.String	getReducedPrice() Returns the reduced price used for deal pricing.
java.lang.String	getRegularPrice() Returns the regular (non-discounted) price (without currency symbol).
java.util.Collection	getRestrictedPeriods() Returns a list of restricted/unrestricted time periods for the item.
java.lang.String	getReturnReason() Returns the return reason (for a returned item only.
java.lang.String	getUnitPrice() Returns the item unit price (without currency symbol).
java.lang.String	getWeight() Returns the item sale weight for an item sold by weight (e.
boolean	isFoodstampEligible() Indicates whether the item is foodstamp eligible.
boolean	isItemRepeatAllowed() Indicates whether item repeat is allowed for this item.
boolean	isReturn() Indicates whether the item is a return
boolean	isTaxable() Indicates whether the item is taxable.
boolean	isTimeRestricted() Indicates whether the item is time restricted.
boolean	isWICEligible() Indicates whether the item is WIC eligible.

Fields

ITEM_SOLD

`public static final java.lang.String` **`ITEM_SOLD`**

item sold action identifier

Methods

(continued from last page)

getItemID

`public java.lang.String getItemID()`

Returns the item code used to sell the item (e.g., the UPC number)

Returns:

item identifier - returns null if the property is undefined

getItemIDQualifier

`public java.lang.String getItemIDQualifier()`

Returns the item code qualifier. Indicates the format of the item code (may be scanned, keyed). For example, IBM 4690 Supermarket Application, the values may be:
ScannedItemCodeKeyedItemCodeItemLookupKeyedLinkedItemCodeWandedItemCode

Returns:

item ID qualifier - returns null if the property is undefined

getUnitPrice

`public java.lang.String getUnitPrice()`

Returns the item unit price (without currency symbol).

Returns:

formatted string representation of the item price per unit - returns null if the property is undefined

getRegularPrice

`public java.lang.String getRegularPrice()`

Returns the regular (non-discounted) price (without currency symbol).

Returns:

formatted string representation of the item regular selling price - returns null if the property is undefined

getReturnReason

`public java.lang.String getReturnReason()`

Returns the return reason (for a returned item only.)

Returns:

reason code identifier - returns null if the property is undefined

getQuantity

```
public int getQuantity()
                 throws AEFException
```

Returns the quantity sold.

Returns:

quantity sold as an int value

(continued from last page)

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

getDealQuantity

```
public int getDealQuantity()
                    throws AEFException
```

Returns the item deal quantity. A deal quantity is used for items priced in multiples (e.g, 3 for 1.00)

Returns:

deal quantity as int value

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

getWeight

```
public java.lang.String getWeight()
```

Returns the item sale weight for an item sold by weight (e.g., produce).

Returns:

item sale weight as a formatted string. - returns null if the property is undefined

getExtendedPrice

```
public java.lang.String getExtendedPrice()
```

Returns the extended price. The extended price is the amount charged for this item sale based on the quantity or weight.

Returns:

extended price as a formatted string. - returns null if the property is undefined

enteredPriceUsed

```
public boolean enteredPriceUsed()
                         throws AEFException
```

Indicates if the item price was overridden, or the item was price required.

Returns:

true if the price was entered by the operator.

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

getDealPrice

```
public java.lang.String getDealPrice()
```

(continued from last page)

Returns the deal price. Deal prices are defined for items sold as multiples (e.g., 3 for 1.00)

Returns:

string representation of the deal price - returns null if the property is undefined

getReducedPrice

```
public java.lang.String getReducedPrice()
```

Returns the reduced price used for deal pricing.

Returns:

string representation of the reduced price - returns null if the property is undefined

getMultiPricingGroup

```
public java.lang.String getMultiPricingGroup()
```

Returns the multi pricing group used for deal pricing.

Returns:

multi pricing group - returns null if the property is undefined

getAgeRestriction

```
public int getAgeRestriction()
                        throws AEFException
```

Returns the age restriction in years for an age restricted item.

Returns:

age restriction in years for an age restricted item

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

isTimeRestricted

```
public boolean isTimeRestricted()
```

Indicates whether the item is time restricted.

Returns:

true if the item is time restricted

isFoodstampEligible

```
public boolean isFoodstampEligible()
                              throws AEFException
```

Indicates whether the item is foodstamp eligible.

Returns:

true if the item is foodstamp eligible

(continued from last page)

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

isWICEligible

```
public boolean isWICEligible()
                        throws AEFException
```

Indicates whether the item is WIC eligible.

Returns:

true if the item is WIC eligible

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

isItemRepeatAllowed

```
public boolean isItemRepeatAllowed()
                              throws AEFException
```

Indicates whether item repeat is allowed for this item. Item repeat is used by operators to indicate an item is sold multiple times without rescanning the item.

Returns:

true if item repeat is allowed

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

isTaxable

```
public boolean isTaxable()
                  throws AEFException
```

Indicates whether the item is taxable.

Returns:

true if the item is taxable

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

isReturn

```
public boolean isReturn()
                 throws AEFException
```

Indicates whether the item is a return

(continued from last page)

Returns:

boolean

Exceptions:

`AEFException -`
if the property is undefined by the event
AEFException error codes are: AEFConst.NO_SUCH_PROPERTYAEFConst.INVALID_PROPERTY_VALUE

getPricingMethod

`public java.lang.String` **`getPricingMethod`**`()`

Gets the pricing method for this item.

Returns:

pricing method - returns null if the property is undefined

getLinkedItemID

`public java.lang.String` **`getLinkedItemID`**`()`

Get the linked item code (if any). Linked items are often used to associate coupons to an item or to define deals.

Returns:

linked item identifier. - returns null if the property is undefined

getLinkedItemIDQualifier

`public java.lang.String` **`getLinkedItemIDQualifier`**`()`

Get the linked item code qualifier. Indicates the data format of the item code (scanned, keyed,..).

Returns:

linked item id qualifier - returns null if the property is undefined

getRestrictedPeriods

`public java.util.Collection` **`getRestrictedPeriods`**`()`

Returns a list of restricted/unrestricted time periods for the item.

Returns:

`Collection`
of `TimeInterval`objects or an empty collection.

See Also:

com.ibm.retail.AEF.event.TimeInterval TimeInterval

com.ibm.retail.AEF.event

Interface ItemSalesListener

All Superinterfaces:
POSAppEventListener

All Known Implementing Classes:
ItemSalesListenerProxy

---

public interface **ItemSalesListener**
extends POSAppEventListener

Listener interface for receiving item events from the terminal session. Item events are generated as items are added (sold) or removed (voided) in a POS transaction.

See Also:
com.ibm.retail.AEF.data.POSDataProvider#addItemSalesListener addItemSalesListener

---

Method Summary

void	`itemAdded(ItemSalesEvent evt)` An item was added to the transaction
void	`itemDepositAdded(ItemSalesEvent evt)` A deposit was added to the transaction
void	`itemDepositRemoved(ItemSalesEvent evt)` An item deposit was removed from the transaction
void	`itemRefund(ItemSalesEvent evt)` An item refund was performed in this transaction
void	`itemRemoved(ItemSalesEvent evt)` An item was removed from the transaction

Methods

itemAdded

```
public void itemAdded(ItemSalesEvent evt)
                throws java.rmi.RemoteException
```

An item was added to the transaction

Parameters:
`evt` -
contains details of the item event

Exceptions:
`RemoteException` -
if a listener can not be notified

itemRemoved

```
public void itemRemoved(ItemSalesEvent evt)
                  throws java.rmi.RemoteException
```

An item was removed from the transaction

Parameters:

`evt -`
contains details of the item event

Exceptions:

`RemoteException -`
if a listener can not be notified

itemDepositAdded

```
public void itemDepositAdded(ItemSalesEvent evt)
                      throws java.rmi.RemoteException
```

A deposit was added to the transaction

Parameters:

`evt -`
contains details of the item event

Exceptions:

`RemoteException -`
if a listener can not be notified

itemDepositRemoved

```
public void itemDepositRemoved(ItemSalesEvent evt)
                        throws java.rmi.RemoteException
```

An item deposit was removed from the transaction

Parameters:

`evt -`
contains details of the item event

Exceptions:

`RemoteException -`
if a listener can not be notified

itemRefund

```
public void itemRefund(ItemSalesEvent evt)
                 throws java.rmi.RemoteException
```

An item refund was performed in this transaction

Parameters:

`evt -`
contains details of the item event

Exceptions:

`RemoteException -`
if a listener can not be notified

com.ibm.retail.AEF.event

Interface LineItemEvent

All Superinterfaces:
POSAppEvent, POSAppEventElement

All Subinterfaces:
TenderEvent, PointsEvent, ItemSalesEvent, DiscountEvent, CouponEvent

---

public interface **LineItemEvent**
extends POSAppEvent

`LineItemEvent`
is the parent interface for the hierarchy of events related to items to a POS transaction.

A `LineItemEvent`is generated when an item is added or removed from a transaction. The `LineItemEvent`object contains details about the item.

Method Summary

`java.lang.String`	`getDescription()` Returns the short item description used by POS.
`java.lang.String`	`getItemModifier()` Returns the item modifier.
`java.util.Collection`	`getPrintLines()` Returns a collection of print lines (Strings) for the line item.
`java.util.Collection`	`getRawPrintLines()` Returns a collection of raw print lines (Strings) for the line item.
`boolean`	`isDeposit()` Indicates whether the line item is a deposit line item.
`boolean`	`isRefunded()` Indicates whether the line item is a refund line item.
`boolean`	`isVoided()` Indicates whether the line item is a voided line item.

Methods

getItemModifier

`public java.lang.String` **`getItemModifier()`**

Returns the item modifier. The item modifier is an application dependent value which indicates some additional information about the item.

Returns:

(continued from last page)

String - returns null if the property is undefined in this event

isVoided

```
public boolean isVoided()
                 throws AEFException
```

Indicates whether the line item is a voided line item.

Returns:

true if the line item is voided.

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

isRefunded

```
public boolean isRefunded()
                   throws AEFException
```

Indicates whether the line item is a refund line item.

Returns:

true if the line item is refunded.

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

isDeposit

```
public boolean isDeposit()
                  throws AEFException
```

Indicates whether the line item is a deposit line item.

Returns:

true if the line item is a deposit.

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

getDescription

```
public java.lang.String getDescription()
```

Returns the short item description used by POS.

Returns:

short description of the item as displayed by the POS terminal. - returns null if the property is undefined

getPrintLines

`public java.util.Collection` **`getPrintLines`**`()`

Returns a collection of print lines (Strings) for the line item. All special characters (escape sequences) have been filtered out.

Returns:
`Collection`
of `Strings`- returns empty collection if no print lines

getRawPrintLines

`public java.util.Collection` **`getRawPrintLines`**`()`

Returns a collection of raw print lines (Strings) for the line item.

Returns:
`Collection`
of `Strings`- returns empty collection if no print lines

com.ibm.retail.AEF.event

Interface OperatorEvent

All Superinterfaces:
POSAppEvent, POSAppEventElement

public interface **OperatorEvent**
extends POSAppEvent

An `OperatorEvent`is generated when an operator signs on or off a POS terminal. The `OperatorEvent`object contains details about the operator and the authorization levels for the operator.

Field Summary

static java.lang.String	`LOCK_ACTION` Lock action identifier
static java.lang.String	`SIGNOFF_ACTION` Signoff action identifier
static java.lang.String	`SIGNON_ACTION` Signon action identifier
static java.lang.String	`UNLOCK_ACTION` Unlock action identifier

Method Summary

java.lang.String	`getID()` Get the operator logon ID.
java.lang.String	`getName()` Get the operator name.
OperatorAuthorization	`getOperatorAuthorization()` Returns information describing the functions authorized for this operator.
boolean	`isAuthorized(java.lang.String authProperty)` Indicates whether the operator is authorized for a specified operation (e.

Fields

SIGNON_ACTION

`public static final java.lang.String` **`SIGNON_ACTION`**

Signon action identifier

(continued from last page)

SIGNOFF_ACTION

`public static final java.lang.String` **`SIGNOFF_ACTION`**

Signoff action identifier

LOCK_ACTION

`public static final java.lang.String` **`LOCK_ACTION`**

Lock action identifier

UNLOCK_ACTION

`public static final java.lang.String` **`UNLOCK_ACTION`**

Unlock action identifier

Methods

getName

`public java.lang.String` **`getName`**`()`

Get the operator name.

Returns:

the name of the operator

getID

`public java.lang.String` **`getID`**`()`

Get the operator logon ID. The ID is used by the operator to sign on to a POS terminal.

Returns:

String

getOperatorAuthorization

`public OperatorAuthorization` **`getOperatorAuthorization`**`()`

Returns information describing the functions authorized for this operator.

Returns:

An `OperatorAuthorization`object containing authorization data.

isAuthorized

```
public boolean isAuthorized(java.lang.String authProperty)
                      throws AEFException
```

Indicates whether the operator is authorized for a specified operation (e.g., "NO_SALE_ALLOWED"). Authorizations are POS application specific, and therefore this method provides for a generic means of checking an authorization based on a property name supplied by the application. A list of property name identifiers is provided in `OperatorAuthorizationProperties`. An exception is thrown if `isAuthorized`is called and the property is undefined by the operator event.

Parameters:

`authProperty` -
property name

(continued from last page)

Returns:

true if the operator is authorized for the function defined by the property name.

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

See Also:

com.ibm.retail.AEF.data.OperatorAuthorizationProperties OperatorAuthorizationProperties

com.ibm.retail.AEF.event

Interface OperatorListener

All Superinterfaces:
POSAppEventListener

All Known Implementing Classes:
OperatorListenerProxy

---

public interface **OperatorListener**
extends POSAppEventListener

Listener interface for receiving `OperatorEvents`from the terminal session. An `OperatorEvent`is fired whenever an operator signs on or off of a POS terminal session.

See Also:
com.ibm.retail.AEF.data.POSDataProvider#addOperatorListener addOperatorListener

---

Method Summary

`void`	`operatorEventOccurred(OperatorEvent evt)` An operator action has been performed on the POS terminal session.

Methods

operatorEventOccurred

```
public void operatorEventOccurred(OperatorEvent evt)
                                  throws java.rmi.RemoteException
```

An operator action has been performed on the POS terminal session.

Parameters:
`evt -`
contains operator details

com.ibm.retail.AEF.event

Interface OptionsEvent

All Superinterfaces:
POSAppEvent, POSAppEventElement

public interface **OptionsEvent**
extends POSAppEvent

An `OptionsEvent`is generated whenever terminal or store level options are loaded for the POS Terminal Sales Application. This options determine the functions available at the terminal and modify the configurable behavior of the terminal.

Method Summary

`java.util.Collection`	`getAlternateTaxCodes()` Returns list of the tax codes used for identifying alternate tax codes
`java.util.Collection`	`getDepartmentDefinitions()` Returns list of the departments defined
`java.util.Collection`	`getItemDiscountReasons()` Returns list of the discount reason codes used for processing item discounts
`java.util.Collection`	`getManualTaxCodes()` Returns list of the tax codes used for identifying manual tax codes
`java.util.Collection`	`getNoTaxCodes()` Returns list of the tax codes used for identifying no tax codes
`java.util.Collection`	`getPriceOverrideReasons()` Returns list of the reason codes used for price overrides
`java.util.Collection`	`getRefundReasons()` Returns list of the reason codes used for refunds
`StoreDefinition`	`getStoreDefinition()` Returns the information about the store (name, address,.
`StoreOptions`	`getStoreOptions()` Returns information about store wide options
`java.util.Collection`	`getTareCodes()` Returns list of the codes used for identifying tares
`java.util.Collection`	`getTenderDefinitions()` Returns a list of tenders defined
`TerminalOptions`	`getTerminalOptions()` Returns information about the terminal specific options

java.util.Collection	getTransactionDiscountReasons() Returns list of the discount reason codes used for processing transaction discounts
java.util.Collection	getVATTaxCodes() Returns list of the tax codes used for identifying VAT
java.util.Collection	getVoidReasons() Returns list of the reason codes used for voids

Methods

getStoreDefinition

`public StoreDefinition` **`getStoreDefinition`**`()`

Returns the information about the store (name, address,..)

Returns:
`StoreDefinition`
object

getTenderDefinitions

`public java.util.Collection` **`getTenderDefinitions`**`()`

Returns a list of tenders defined

Returns:
`Collection`
of `TenderDefinition`objects - empty collection if no tender definitions were contained in the event

See Also:

com.ibm.retail.AEF.event.TenderDefinition TenderDefinition

getPriceOverrideReasons

`public java.util.Collection` **`getPriceOverrideReasons`**`()`

Returns list of the reason codes used for price overrides

Returns:
`Collection`
of `ReasonCode`objects - empty collection if no reason codes for price overrides were contained in the event

See Also:

com.ibm.retail.AEF.event.ReasonCode ReasonCode

getRefundReasons

`public java.util.Collection` **`getRefundReasons`**`()`

Returns list of the reason codes used for refunds

Returns:
`Collection`
of `ReasonCode`objects - empty collection if no reason codes for refunds were contained in the event

(continued from last page)

See Also:

com.ibm.retail.AEF.event.ReasonCode ReasonCode

getVoidReasons

`public java.util.Collection` **`getVoidReasons`**`()`

Returns list of the reason codes used for voids

Returns:
`Collection`
of `ReasonCode`objects - empty collection if no reason codes for voids were contained in the event

See Also:

com.ibm.retail.AEF.event.ReasonCode ReasonCode

getTareCodes

`public java.util.Collection` **`getTareCodes`**`()`

Returns list of the codes used for identifying tares

Returns:
`Collection`
of `TareCode`objects - empty collection if no tare codes were contained in the event

See Also:

com.ibm.retail.AEF.event.TareCode TareCode

getVATTaxCodes

`public java.util.Collection` **`getVATTaxCodes`**`()`

Returns list of the tax codes used for identifying VAT

Returns:
`Collection`
of `TaxCode`objects - empty collection if no VAT tax codes were contained in the event

See Also:

com.ibm.retail.AEF.event.TaxCode TaxCode

getAlternateTaxCodes

`public java.util.Collection` **`getAlternateTaxCodes`**`()`

Returns list of the tax codes used for identifying alternate tax codes

Returns:
`Collection`
of `TaxCode`objects - empty collection if no alternate tax codes were contained in the event

See Also:

com.ibm.retail.AEF.event.TaxCode TaxCode

getManualTaxCodes

`public java.util.Collection` **`getManualTaxCodes`**`()`

(continued from last page)

Returns list of the tax codes used for identifying manual tax codes

Returns:
`Collection` of `TaxCode`objects - empty collection if no manual tax codes were contained in the event

See Also:

com.ibm.retail.AEF.event.TaxCode TaxCode

getNoTaxCodes

`public java.util.Collection` **`getNoTaxCodes`**`()`

Returns list of the tax codes used for identifying no tax codes

Returns:
`Collection` of `TaxCode`objects - empty collection if no manual tax codes were contained in the event

See Also:

com.ibm.retail.AEF.event.TaxCode TaxCode

getItemDiscountReasons

`public java.util.Collection` **`getItemDiscountReasons`**`()`

Returns list of the discount reason codes used for processing item discounts

Returns:
`Collection` of `DiscountReasonCode`objects - empty collection if no item discount reason codes were contained in the event

See Also:

com.ibm.retail.AEF.event.DiscountReasonCode DiscountReasonCode

getTransactionDiscountReasons

`public java.util.Collection` **`getTransactionDiscountReasons`**`()`

Returns list of the discount reason codes used for processing transaction discounts

Returns:
`Collection` of `TransactionDiscountReasonCode`objects - empty collection if no transaction discount reason codes were contained in the event

See Also:

com.ibm.retail.AEF.event.TransactionDiscountReasonCode TransactionDiscountReasonCode

getDepartmentDefinitions

`public java.util.Collection` **`getDepartmentDefinitions`**`()`

Returns list of the departments defined

Returns:
`Collection` of `DepartmentDefinition`objects - empty collection if no departments were defined in the event

See Also:

(continued from last page)

com.ibm.retail.AEF.event.DepartmentDefinition DepartmentDefinition

getTerminalOptions

`public TerminalOptions` **`getTerminalOptions`**`()`

Returns information about the terminal specific options

Returns:

`TerminalOptions`
object describing the options for this terminal - null if no terminal options defined

getStoreOptions

`public StoreOptions` **`getStoreOptions`**`()`

Returns information about store wide options

Returns:

`StoreOptions`
object describing store-wide options - null if no terminal options defined

com.ibm.retail.AEF.event

Interface OptionsListener

All Superinterfaces:
POSAppEventListener

All Known Implementing Classes:
OptionsListenerProxy

---

public interface **OptionsListener**
extends POSAppEventListener

Listener interface for receiving `OptionsEvents`from the terminal session. An `OptionsEvent`is fired whenever store or terminal options (configuration information) is loaded into the terminal session.

This event should be generated whenever a change occurs to the options and during the initial load of options data for the terminal session.

See Also:
com.ibm.retail.AEF.data.POSDataProvider#addOptionsListener addOptionsListener

---

Method Summary

`void`	`optionsLoaded(OptionsEvent evt)` Options information has been loaded for the terminal session.

Methods

optionsLoaded

```
public void optionsLoaded(OptionsEvent evt)
                    throws java.rmi.RemoteException
```

Options information has been loaded for the terminal session.

Parameters:
`evt` -
contains options details

com.ibm.retail.AEF.event

Interface PointsEvent

All Superinterfaces:
LineItemEvent, POSAppEvent, POSAppEventElement

public interface **PointsEvent**
extends LineItemEvent

A `PointsEvent`is generated when loyalty points are awarded or redeemed in a transaction.

A `PointsEvent`may affect multiple points totals for a loyalty account. Additional totals (e.g., for clubs) are contained in the Collection accessible through the method `getPointsTotals`.

Method Summary

`java.util.Collection`	`getAdditionalPointsTotals()` Get any additional points totals for clubs or secondary accounts.
`java.lang.String`	`getID()` Returns a unique identifier of the points reward.
`java.lang.String`	`getIDQualifier()` Returns the points ID qualifier.
`java.lang.String`	`getPoints()` Returns the total points for the primary account.
`int`	`getQuantity()` Returns the points quantity as an integer value.
`java.lang.String`	`getType()` Returns the type of points (e.
`boolean`	`isAwarded()` Indicates if the points are awarded.
`boolean`	`isRedeemed()` Indicates if the points are redeemed.

Methods

getType

`public java.lang.String` **`getType()`**

Returns the type of points (e.g., "primary").

Returns:

(continued from last page)

String - returns null if the property is undefined

getPoints

`public java.lang.String getPoints()`

Returns the total points for the primary account.

Returns:

string representation of the total number of points received - returns null if the property is undefined

getID

`public java.lang.String getID()`

Returns a unique identifier of the points reward.

Returns:

points ID - returns null if the property is undefined

getIDQualifier

`public java.lang.String getIDQualifier()`

Returns the points ID qualifier. The ID qualifier indicates the data format of the ID.

Returns:

points ID qualifer - returns null if the property is undefined

getQuantity

```
public int getQuantity()
                throws AEFException
```

Returns the points quantity as an integer value.

Returns:

the points quantity as an integer value.

Exceptions:

`AEFException` -
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

getAdditionalPointsTotals

`public java.util.Collection getAdditionalPointsTotals()`

Get any additional points totals for clubs or secondary accounts.

Returns:

`Collection`
of `PointsTotal`objects (returns null if no totals)

See Also:

com.ibm.retail.AEF.event.PointsTotal PointsTotal

isRedeemed

```
public boolean isRedeemed()
                    throws AEFException
```

Indicates if the points are redeemed. If not redeemed, then they are awarded.

Returns:

true if the points are redeemed

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

isAwarded

```
public boolean isAwarded()
                   throws AEFException
```

Indicates if the points are awarded. If not awarded, then they are redeemed.

Returns:

true if the points are awarded.

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

com.ibm.retail.AEF.event

Interface PointsListener

All Superinterfaces:
POSAppEventListener

All Known Implementing Classes:
PointsListenerProxy

---

public interface **PointsListener**
extends POSAppEventListener

Listener interface for receiving `PointsEvents`from the terminal session. A `PointsEvent`is fired whenever loyalty points are awarded or redeemed in a transaction.

See Also:
com.ibm.retail.AEF.data.POSDataProvider#addPointsListener addPointsListener

---

Method Summary

`void`	`pointsOccurred(PointsEvent evt)` Points were awarded or redeemed.

Methods

pointsOccurred

```
public void pointsOccurred(PointsEvent evt)
                    throws java.rmi.RemoteException
```

Points were awarded or redeemed.

Parameters:
`evt` -
contains points details

Exceptions:
`RemoteException` -
if remote access fails

com.ibm.retail.AEF.event

Class PointsTotal

```
java.lang.Object
  |
  +-com.ibm.retail.AEF.event.PointsTotal
```

All Implemented interfaces:
java.io.Serializable

---

public class **PointsTotal**
extends java.lang.Object
implements java.io.Serializable

A PointsTotal is a simple data object to contain information about customer loyalty points.

Field Summary

java.lang.String	description
java.lang.String	type
java.lang.String	value

Constructor Summary

PointsTotal() Default Constructor
PointsTotal(java.lang.String type,java.lang.String type,java.lang.String type) Constructor

Method Summary

java.lang.String	getDescription() Get the description
java.lang.String	getType() Get the type of points total
java.lang.String	getValue() Get the value
void	setDescription(java.lang.String desc) Set the description
void	setType(java.lang.String type) Set the type of points total

void	`setValue(java.lang.String value)` Set the value
java.lang.String	`toString()` Returns the event as a string.

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

type

`protected java.lang.String` **`type`**

description

`protected java.lang.String` **`description`**

value

`protected java.lang.String` **`value`**

Constructors

PointsTotal

`public` **`PointsTotal`**`()`

Default Constructor

PointsTotal

```
public PointsTotal(java.lang.String type,
                   java.lang.String desc,
                   java.lang.String value)
```

Constructor

Methods

getType

`public java.lang.String` **`getType`**`()`

Get the type of points total

Returns:

(continued from last page)

String

getDescription

`public java.lang.String **getDescription**()`

Get the description

Returns:

String

getValue

`public java.lang.String **getValue**()`

Get the value

Returns:

String

setType

`public void **setType**(java.lang.String type)`

Set the type of points total

Parameters:

`type` -
the points type

setDescription

`public void **setDescription**(java.lang.String desc)`

Set the description

Parameters:

`desc` -
description

setValue

`public void **setValue**(java.lang.String value)`

Set the value

Parameters:

`value` -
the points value

toString

`public java.lang.String **toString**()`

Returns the event as a string.

Returns:

String

com.ibm.retail.AEF.event

Interface POSAppEvent

All Superinterfaces:
POSAppEventElement

All Subinterfaces:
WorkstationStatusEvent, TransactionTotalsEvent, TransactionStatusEvent, StateChangeEvent, ScaleEvent, ReportEvent, OptionsEvent, OperatorEvent, LineItemEvent, TenderEvent, PointsEvent, ItemSalesEvent, DiscountEvent, CouponEvent, CustomerEvent, CashReceiptEvent

---

public interface **POSAppEvent**
extends POSAppEventElement

`POSAppEvent`
is the base interface for events generated for POS application data.

Method Summary

`java.lang.String`	`getEventType()` Gets the XML element tag (event type) for this event.
`java.lang.String`	`getPropertyChangeCategory()` Gets the Satin property change category defined for this event.
`java.lang.String`	`getTerminalNumber()` Gets the terminal number of the event source.
`void`	`setEventType(java.lang.String value)` Sets the XML element tag (event type) for this event.
`void`	`setTerminalNumber(java.lang.String term)` Sets the terminal number of the event source.

Methods

getTerminalNumber

`public java.lang.String` **`getTerminalNumber`**`()`

Gets the terminal number of the event source.

Returns:

terminal number

---

setTerminalNumber

`public void` **`setTerminalNumber`**`(java.lang.String term)`

Sets the terminal number of the event source.

(continued from last page)

Parameters:

`term -`
terminal number

getEventType

`public java.lang.String` **`getEventType`**`()`

Gets the XML element tag (event type) for this event.

Returns:

String The event type.

setEventType

`public void` **`setEventType`**`(java.lang.String value)`

Sets the XML element tag (event type) for this event.

Parameters:

`String -`
The event type.

getPropertyChangeCategory

`public java.lang.String` **`getPropertyChangeCategory`**`()`

Gets the Satin property change category defined for this event.

Returns:

String category id

com.ibm.retail.AEF.event

Interface POSAppEventElement

All Subinterfaces:

TransactionDefinition, TimeInterval, TerminalOptions, TenderDefinition, TaxCode, TareCode, StoreOptions, StoreDefinition, ReportSection, ReasonCode, DiscountReasonCode, TransactionDiscountReasonCode, POSAppEvent, WorkstationStatusEvent, TransactionTotalsEvent, TransactionStatusEvent, StateChangeEvent, ScaleEvent, ReportEvent, OptionsEvent, OperatorEvent, LineItemEvent, TenderEvent, PointsEvent, ItemSalesEvent, DiscountEvent, CouponEvent, CustomerEvent, CashReceiptEvent, DepartmentDefinition

---

public interface **POSAppEventElement**
extends java.io.Serializable

POSAppEventElement is an interface which encapsulates events generated via POS application data.

Method Summary

boolean	`getBooleanPropertyValue(java.lang.String propertyName)` get a boolean property value
java.lang.String	`getData()` Gets the XML element (non-attribute) character data associated with this element For example.
int	`getIntegerPropertyValue(java.lang.String propertyName)` Get an int property value
java.lang.Object	`getProperty(java.lang.String name)` Gets an event property.
java.util.Iterator	`getPropertyEntries()` Get an iterator of property name/value entries for this event.
void	`setData(java.lang.String data)` Sets the XML element character data associated with this element.
void	`setProperty(java.lang.String name,java.lang.Object name)` Set an event property.
java.lang.String	`toString()` Returns the element as a string.

Methods

getData

`public java.lang.String` **`getData`**`()`

Gets the XML element (non-attribute) character data associated with this element For example. If the event data is passed via XML as: This is data, Then "This is data" would be returned by this method.

(continued from last page)

Returns:

String character data

setData

```
public void setData(java.lang.String data)
```

Sets the XML element character data associated with this element.

Parameters:

`String -`
character data

setProperty

```
public void setProperty(java.lang.String name,
                        java.lang.Object value)
```

Set an event property. (attribute or child element).

Parameters:

`propName -`
The property name.
`object -`
The property value.

getProperty

```
public java.lang.Object getProperty(java.lang.String name)
```

Gets an event property. (attribute or child element).

Parameters:

`propName -`
The property name.

Returns:

Object The property value.

getPropertyEntries

```
public java.util.Iterator getPropertyEntries()
```

Get an iterator of property name/value entries for this event.

Returns:

iterator of property name/value pairs as Map.Entry objects

getBooleanPropertyValue

```
public boolean getBooleanPropertyValue(java.lang.String propertyName)
                                 throws AEFException
```

get a boolean property value

Returns:

boolean

Exceptions:

(continued from last page)

`AEFException -`
if a property is undefined

getIntegerPropertyValue

```
public int getIntegerPropertyValue(java.lang.String propertyName)
                                   throws AEFException
```

Get an int property value

Returns:

int

Exceptions:

`AEFException -`
if a property is undefined

toString

```
public java.lang.String toString()
```

Returns the element as a string.

Returns:

String

com.ibm.retail.AEF.event

Interface POSAppEventListener

All Subinterfaces:

WorkstationStatusListener, TransactionTotalsListener, TransactionStatusListener, TenderListener, StateChangeListener, ScaleListener, ReportListener, PointsListener, OptionsListener, OperatorListener, ItemSalesListener, GenericEventListener, DiscountListener, CustomerListener, CouponListener, CashReceiptListener

---

public interface **POSAppEventListener**
extends java.rmi.Remote

`POSAppEventListener`
is the base interface for all listeners of `POSAppEvents`generated by the `POSDataProvider`API.

---

com.ibm.retail.AEF.event

Interface ReasonCode

All Superinterfaces:
POSAppEventElement

All Subinterfaces:
DiscountReasonCode, TransactionDiscountReasonCode

---

public interface **ReasonCode**
extends POSAppEventElement

A ReasonCode is a simple data object to contain information about POS reason codes for refunds, discounts, etc.

A reason code contains a numeric code and a text description.

Field Summary

static java.lang.String	`REASON_CODE` Reason code identifier (numeric code)
static java.lang.String	`REASON_DESCRIPTION` Tender description

Method Summary

java.lang.String	`getCode()` Get the code value.
java.lang.String	`getDescription()` Get the description of the reason.

Fields

REASON_CODE

`public static final java.lang.String` **`REASON_CODE`**

Reason code identifier (numeric code)

REASON_DESCRIPTION

`public static final java.lang.String` **`REASON_DESCRIPTION`**

Tender description

Methods

(continued from last page)

getCode

`public java.lang.String` **`getCode`**`()`

Get the code value.

Returns:

value of the reason code identifier

getDescription

`public java.lang.String` **`getDescription`**`()`

Get the description of the reason.

Returns:

description of the reason.

com.ibm.retail.AEF.event

Interface ReportEvent

All Superinterfaces:
POSAppEvent, POSAppEventElement

public interface **ReportEvent**
extends POSAppEvent

A `ReportEvent`contains report information from the POS application.

Method Summary

`java.lang.String`	`getID()` Get the report identifier
`java.util.Collection`	`getReportSections()` Get the collection of report sections.

Methods

getID

`public java.lang.String` **`getID`**`()`

Get the report identifier

Returns:

String

getReportSections

`public java.util.Collection` **`getReportSections`**`()`

Get the collection of report sections.

Returns:

Collection of `ReportSection`objects

See Also:

com.ibm.retail.AEF.event.ReportSection ReportSection

com.ibm.retail.AEF.event

Interface ReportListener

All Superinterfaces:
POSAppEventListener

All Known Implementing Classes:
ReportListenerProxy

---

public interface **ReportListener**
extends POSAppEventListener

Listener interface for receiving `ReportEvents`from the terminal session. A `ReportEvent`is fired to communicate reporting information from the POS application.

See Also:
com.ibm.retail.AEF.data.POSDataProvider#addReportListener addReportListener

---

Method Summary

`void`	`reportDataAvailable(ReportEvent evt)` Report event occurred

Methods

reportDataAvailable

```
public void reportDataAvailable(ReportEvent evt)
                        throws java.rmi.RemoteException
```

Report event occurred

Parameters:
`evt` -
contains report information

Exceptions:
`RemoteException` -
if remote access fails

com.ibm.retail.AEF.event

Interface ReportSection

All Superinterfaces:
POSAppEventElement

public interface **ReportSection**
extends POSAppEventElement

A `ReportSection`is a data object containing information about a section of a report. A section contains `ReportLine`objects which describe the detail of a report.

Method Summary

`java.lang.String`	`getID()` Get the identifier of this section
`java.util.Collection`	`getReportLines()` Get the ReportLines for this section.

Methods

getID

`public java.lang.String` **`getID`**`()`

Get the identifier of this section

Returns:

String

getReportLines

`public java.util.Collection` **`getReportLines`**`()`

Get the ReportLines for this section.

Returns:

Collection of `ReportLine`objects.

com.ibm.retail.AEF.event

Interface ScaleEvent

All Superinterfaces:
POSAppEvent, POSAppEventElement

public interface **ScaleEvent**
extends POSAppEvent

A `ScaleEvent`is generated when an item is weighed.

Method Summary

`java.lang.String`	`getWeightUnit()` Get the scale weight unit of measure (e.
`java.lang.String`	`getWeightValue()` Get the scale weight value as a formatted string (e.

Methods

getWeightValue

`public java.lang.String` **`getWeightValue()`**

Get the scale weight value as a formatted string (e.g., 1.79).

Returns:

string representation of the weight value

getWeightUnit

`public java.lang.String` **`getWeightUnit()`**

Get the scale weight unit of measure (e.g., LB).

Returns:

string representation of the weight unit of measure

com.ibm.retail.AEF.event

Interface ScaleListener

All Superinterfaces:
POSAppEventListener

All Known Implementing Classes:
ScaleListenerProxy

---

public interface **ScaleListener**
extends POSAppEventListener

Listener interface for receiving `ScaleEvents`from the terminal session. A `ScaleEvent`is fired whenever an item is weighed.

See Also:
com.ibm.retail.AEF.data.POSDataProvider#addScaleListener addScaleListener

---

Method Summary

`void`	`itemWeighed(ScaleEvent evt)` Scale event occurred (an item was weighed).

Methods

itemWeighed

```
public void itemWeighed(ScaleEvent evt)
                 throws java.rmi.RemoteException
```

Scale event occurred (an item was weighed).

Parameters:
`evt` -
contains weight information

Exceptions:
`RemoteException` -
if remote access fails

com.ibm.retail.AEF.event

Interface StateChangeEvent

All Superinterfaces:
POSAppEvent, POSAppEventElement

public interface **StateChangeEvent**
extends POSAppEvent

`StateChangeEvent`
is an event object generated when the State of the POS application changes. State changes are associated with POS applications that use state tables to drive POS processing.

Method Summary

`int`	`getStateId()` Returns the state identifier.
`boolean`	`isChangeToCurrent()` Indicates if this state change is merely a change to the current state.

Methods

getStateId

`public int` **`getStateId`**`()`

Returns the state identifier.

Returns:

state identifier

isChangeToCurrent

`public boolean` **`isChangeToCurrent`**`()`

Indicates if this state change is merely a change to the current state.

Returns:

true if this state change is a change to CURRENT

com.ibm.retail.AEF.event

Interface StateChangeListener

All Superinterfaces:
POSAppEventListener

All Known Implementing Classes:
StateChangeListenerProxy

---

public interface **StateChangeListener**
extends POSAppEventListener

Listener interface for receiving `StateChangeEvents`from the terminal POS session. A `StateChangeEvent`is fired whenever the POS application enters a new state.

States are identified by a number "stateID". To access details about a state, use the methods provided in `POSDataProvider`to get the corresponding `State`object.

See Also:
com.ibm.retail.AEF.data.POSDataProvider#addStateChangeListener addStateChangeListener

---

Method Summary

`void`	`stateChanged(StateChangeEvent evt)` A state change occurred in the POS application

Methods

stateChanged

```
public void stateChanged(StateChangeEvent evt)
                  throws java.rmi.RemoteException
```

A state change occurred in the POS application

Parameters:
`evt` -
contains state change details

com.ibm.retail.AEF.event

Interface StoreDefinition

All Superinterfaces:
POSAppEventElement

public interface **StoreDefinition**
extends POSAppEventElement

`StoreDefinition`
is an interface for objects containing the store attrbutes. The `StoreDefintion`is accessible from `OptionsEvent`.

See Also:
com.ibm.retail.AEF.event.OptionsEvent#getStoreDefinition getStoreDefinition

Method Summary

`java.lang.String`	`getAddress1()` Get the store address1.
`java.lang.String`	`getAddress2()` Get the store address2.
`java.lang.String`	`getCity()` Get the store city.
`java.lang.String`	`getDivision()` Get the store division.
`java.lang.String`	`getName()` Get the store name or ID.
`java.lang.String`	`getPhone1()` Get the store phone number1.
`java.lang.String`	`getPhone2()` Get the store phone number2.
`java.lang.String`	`getState()` Get the store state.
`java.lang.String`	`getStoreNumber()` Get the store number.
`java.lang.String`	`getZip()` Get the store zipcode.

Methods

(continued from last page)

getName

`public java.lang.String getName()`

Get the store name or ID.

Returns:

store name

getDivision

`public java.lang.String getDivision()`

Get the store division.

Returns:

store division

getStoreNumber

`public java.lang.String getStoreNumber()`

Get the store number.

Returns:

store number

getAddress1

`public java.lang.String getAddress1()`

Get the store address1.

Returns:

store address1

getAddress2

`public java.lang.String getAddress2()`

Get the store address2.

Returns:

store address2

getCity

`public java.lang.String getCity()`

Get the store city.

Returns:

store city

(continued from last page)

getState

`public java.lang.String getState()`

Get the store state.

Returns:

store state

getZip

`public java.lang.String getZip()`

Get the store zipcode.

Returns:

store zipcode

getPhone1

`public java.lang.String getPhone1()`

Get the store phone number1.

Returns:

store phone

getPhone2

`public java.lang.String getPhone2()`

Get the store phone number2.

Returns:

store phone

com.ibm.retail.AEF.event

Interface StoreOptions

All Superinterfaces:
POSAppEventElement

public interface **StoreOptions**
extends POSAppEventElement

`StoreOptions`
is a simple data object to contain information about POS store level options. Store options determine what POS functions are enabled store-wide. `StoreOptions`are accessible from the `OptionsEvent`.

See Also:
com.ibm.retail.AEF.event.OptionsEvent#getStoreOptions getStoreOptions

Method Summary

`java.lang.String`	`getCustomerFunctionCode()` Get the customer function code.
`java.lang.String`	`getMaximumSuspendedTransactions()` Get the maximum number of suspended transactions allowed.
`java.lang.String`	`getMaximumTransactionSize()` Get the maximum size allowed for a transaction.
`java.util.Collection`	`getTransactionDefinitions()` Get the collection of transaction definitions.
`java.lang.String`	`getTransactionWarningSize()` Get the warning size for transactions.
`java.lang.String`	`getVolumeInputDecimalPlaces()` Get the number of decimal places for inputting volumes.
`java.lang.String`	`getVolumeUnitPriceDecimalPlaces()` Get the number of decimal places for volume unit prices.
`java.lang.String`	`getWeightInputDecimalPlaces()` Get the number of decimal places for inputting weights.
`java.lang.String`	`getWICEBT()` Get the identifier for WIC EBT
`boolean`	`isEnabled(java.lang.String propertyName)` Indicates if a function enabled or allowed.

Methods

(continued from last page)

isEnabled

```
public boolean isEnabled(java.lang.String propertyName)
                    throws AEFException
```

Indicates if a function enabled or allowed. Since store options are application specific, `isEnabled`provides a method to determine the value of an option based on the property name of the function. Property names are contained in the `StoreOptionsProperties`interface.

Parameters:

`propertyName -`
name of the property or function

Returns:

true if the function is enabled

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

See Also:

com.ibm.retail.AEF.data.StoreOptionsProperties StoreOptionsProperties

---

getMaximumTransactionSize

```
public java.lang.String getMaximumTransactionSize()
```

Get the maximum size allowed for a transaction.

Returns:

string representation of the maximum transaction size

---

getMaximumSuspendedTransactions

```
public java.lang.String getMaximumSuspendedTransactions()
```

Get the maximum number of suspended transactions allowed.

Returns:

string representation of the maximum number of suspended transactions

---

getTransactionWarningSize

```
public java.lang.String getTransactionWarningSize()
```

Get the warning size for transactions.

Returns:

string representation of the warning size

---

getCustomerFunctionCode

```
public java.lang.String getCustomerFunctionCode()
```

Get the customer function code.

(continued from last page)

Returns:

customer function code or POS key used

getWeightInputDecimalPlaces

`public java.lang.String` **`getWeightInputDecimalPlaces`**`()`

Get the number of decimal places for inputting weights.

Returns:

string representation of the number of decimal places

getVolumeInputDecimalPlaces

`public java.lang.String` **`getVolumeInputDecimalPlaces`**`()`

Get the number of decimal places for inputting volumes.

Returns:

string representation of the number of decimal places

getVolumeUnitPriceDecimalPlaces

`public java.lang.String` **`getVolumeUnitPriceDecimalPlaces`**`()`

Get the number of decimal places for volume unit prices.

Returns:

string representation of the number of decimal places

getWICEBT

`public java.lang.String` **`getWICEBT`**`()`

Get the identifier for WIC EBT

Returns:

string identifier of WIC EBT

getTransactionDefinitions

`public java.util.Collection` **`getTransactionDefinitions`**`()`

Get the collection of transaction definitions.

Returns:

Collection of `TransactionDefinition`objects - empty of no definitions included in this event

See Also:

com.ibm.retail.AEF.event.TransactionDefinition TransactionDefinition

com.ibm.retail.AEF.event

Interface TareCode

All Superinterfaces:
POSAppEventElement

public interface **TareCode**
extends POSAppEventElement

`TareCode`
is a simple data object to contain information about POS tares.

A tare code contains a numeric code and a text description.

Field Summary

static java.lang.String	`TARE_CODE` Property name identifier for: Tare code
static java.lang.String	`TARE_DESCRIPTION` Property name identifier for: Tare description

Method Summary

java.lang.String	`getCode()` Get the tare code value.
java.lang.String	`getDescription()` Get the description of the tare.

Fields

TARE_CODE

`public static final java.lang.String` **`TARE_CODE`**

Property name identifier for: Tare code

TARE_DESCRIPTION

`public static final java.lang.String` **`TARE_DESCRIPTION`**

Property name identifier for: Tare description

Methods

getCode

`public java.lang.String` **`getCode`**`()`

(continued from last page)

Get the tare code value.

Returns:

tare code value

getDescription

`public java.lang.String` **`getDescription`**`()`

Get the description of the tare.

Returns:

tare description

com.ibm.retail.AEF.event

Interface TaxCode

All Superinterfaces:
POSAppEventElement

public interface **TaxCode**
extends POSAppEventElement

A `TaxCode`is a simple data object to contain information about POS tax code definitions.

Field Summary

static java.lang.String	`TAX_CODE` Identifier for tax code property
static java.lang.String	`TAX_DESCRIPTION` Identifier for tax description property
static java.lang.String	`TAX_RATE` Identifier for tax percentage property

Method Summary

java.lang.String	`getCode()` Get the tax code identifier.
java.lang.String	`getDescription()` Get the description of the tax.
java.lang.String	`getRate()` Get the tax rate as a percentage.

Fields

TAX_CODE

`public static final java.lang.String` **`TAX_CODE`**

Identifier for tax code property

TAX_DESCRIPTION

`public static final java.lang.String` **`TAX_DESCRIPTION`**

Identifier for tax description property

(continued from last page)

TAX_RATE

`public static final java.lang.String` **`TAX_RATE`**

Identifier for tax percentage property

Methods

getCode

`public java.lang.String` **`getCode`**`()`

Get the tax code identifier.

Returns:

tax code identifier

getDescription

`public java.lang.String` **`getDescription`**`()`

Get the description of the tax.

Returns:

tax description

getRate

`public java.lang.String` **`getRate`**`()`

Get the tax rate as a percentage.

Returns:

string representation of the tax percentage (e.g. "7.0") - null if no rate is defined

com.ibm.retail.AEF.event

Interface TenderDefinition

All Superinterfaces:
POSAppEventElement

public interface **TenderDefinition**
extends POSAppEventElement

A TenderDefinition is a simple data object to contain information about POS tenders (Cash, Check, Credit,..). Includes the description and the POS keycode used to process the tender type.

Field Summary

static java.lang.String	`CAN_COUNT` Property name identifier for: Indicates if this tender can be counted at this POS terminal
static java.lang.String	`CAN_LOAN` Property name identifier for: Indicates if this tender can be loaned at this POS terminal
static java.lang.String	`CAN_PICKUP` Property name identifier for: Indicates if this tender can be picked up at this POS terminal
static java.lang.String	`CAN_REFUND` Property name identifier for: Indicates if this tender can be refunded at this POS terminal
static java.lang.String	`CAN_VERIFY` Property name identifier for: Indicates if this tender can be verified at this POS terminal
static java.lang.String	`FOREIGN_TENDER` Property name identifier for: Indicates if this tender is a foreign tender
static java.lang.String	`TENDER_ALLOWED` Property name identifier for: Indicates if this tender is allowed at this POS terminal
static java.lang.String	`TENDER_DESCRIPTION` Property name identifier for: Tender description is a textual description of the tender (e.
static java.lang.String	`TENDER_EXCHANGE_RANK` Property name identifier for: Tender exchange rank is an application specified rank of the tender
static java.lang.String	`TENDER_KEY` Property name identifier for: Tender key is the function code used for the tender on a POS keyboard
static java.lang.String	`TENDER_SHORT_DESCRIPTION` Property name identifier for: Tender short description is an abbreviated description of the tender

static java.lang.String	TENDER_TYPE Property name identifier for: Tender type as defined by the POS application, typically a 2-digit number
static java.lang.String	TENDER_VARIETY Property name identifier for: Tender variety as defined by the POS application

Method Summary

boolean	canCount() Indicates if this tender be counted at this POS terminal.
boolean	canLoan() Indicates if this tender be loaned at this POS terminal
boolean	canPickUp() Indicates if this tender be picked up at this POS terminal
boolean	canRefund() Indicates if this tender be refunded at this POS terminal.
boolean	canVerify() Indicates if this tender be verified at this POS terminal.
java.lang.String	getDescription() Get the description.
java.lang.String	getKey() Get the tender key.
java.lang.String	getShortDescription() Get the tender short description.
java.lang.String	getTenderExchangeRank() Get the tender exchange rank.
java.lang.String	getType() Get the type value as defined by the POS application.
java.lang.String	getVariety() Get the tender variety.
boolean	isAllowed() Indicates if this is tender allowed at this POS terminal.
boolean	isForeignTender() Indicates if this is a foreign tender.

Fields

(continued from last page)

TENDER_TYPE

`public static final java.lang.String` **`TENDER_TYPE`**

Property name identifier for: Tender type as defined by the POS application, typically a 2-digit number

TENDER_DESCRIPTION

`public static final java.lang.String` **`TENDER_DESCRIPTION`**

Property name identifier for: Tender description is a textual description of the tender (e.g. "CASH")

TENDER_SHORT_DESCRIPTION

`public static final java.lang.String` **`TENDER_SHORT_DESCRIPTION`**

Property name identifier for: Tender short description is an abbreviated description of the tender

TENDER_EXCHANGE_RANK

`public static final java.lang.String` **`TENDER_EXCHANGE_RANK`**

Property name identifier for: Tender exchange rank is an application specified rank of the tender

TENDER_VARIETY

`public static final java.lang.String` **`TENDER_VARIETY`**

Property name identifier for: Tender variety as defined by the POS application

TENDER_KEY

`public static final java.lang.String` **`TENDER_KEY`**

Property name identifier for: Tender key is the function code used for the tender on a POS keyboard

FOREIGN_TENDER

`public static final java.lang.String` **`FOREIGN_TENDER`**

Property name identifier for: Indicates if this tender is a foreign tender

TENDER_ALLOWED

`public static final java.lang.String` **`TENDER_ALLOWED`**

Property name identifier for: Indicates if this tender is allowed at this POS terminal

CAN_LOAN

`public static final java.lang.String` **`CAN_LOAN`**

Property name identifier for: Indicates if this tender can be loaned at this POS terminal

CAN_PICKUP

`public static final java.lang.String` **`CAN_PICKUP`**

Property name identifier for: Indicates if this tender can be picked up at this POS terminal

CAN_COUNT

`public static final java.lang.String` **`CAN_COUNT`**

(continued from last page)

Property name identifier for: Indicates if this tender can be counted at this POS terminal

CAN_REFUND

`public static final java.lang.String` **`CAN_REFUND`**

Property name identifier for: Indicates if this tender can be refunded at this POS terminal

CAN_VERIFY

`public static final java.lang.String` **`CAN_VERIFY`**

Property name identifier for: Indicates if this tender can be verified at this POS terminal

Methods

getType

`public java.lang.String` **`getType`**`()`

Get the type value as defined by the POS application. Type values are often numeric, (e.g., "1" for "CASH")

Returns:

tender type identifier

getDescription

`public java.lang.String` **`getDescription`**`()`

Get the description. The tender description is a textual desciption such as "CASH".

Returns:

tender description

getVariety

`public java.lang.String` **`getVariety`**`()`

Get the tender variety. Tender variety is application specific, and may be a numeric code to further identify the tender.

Returns:

tender variety

getKey

`public java.lang.String` **`getKey`**`()`

Get the tender key. The tender key identifies a key or function code on the POS keyboard used to identify the tender.

Returns:

tender key

getShortDescription

`public java.lang.String` **`getShortDescription`**`()`

Get the tender short description.

Returns:

(continued from last page)

short description

getTenderExchangeRank

```
public java.lang.String getTenderExchangeRank()
```

Get the tender exchange rank.

Returns:

tender exchange rank.

isForeignTender

```
public boolean isForeignTender()
                        throws AEFException
```

Indicates if this is a foreign tender.

Returns:

true if this is a foreign tender.

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

isAllowed

```
public boolean isAllowed()
                  throws AEFException
```

Indicates if this is tender allowed at this POS terminal.

Returns:

true if this is tender allowed at this POS terminal.

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

canLoan

```
public boolean canLoan()
                throws AEFException
```

Indicates if this tender be loaned at this POS terminal

Returns:

true if this tender be loaned at this POS terminal

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

canPickUp

```
public boolean canPickUp()
                  throws AEFException
```

Indicates if this tender be picked up at this POS terminal

Returns:

true if this tender be picked up at this POS terminal

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

canCount

```
public boolean canCount()
                 throws AEFException
```

Indicates if this tender be counted at this POS terminal.

Returns:

true if this tender be counted at this POS terminal

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

canRefund

```
public boolean canRefund()
                  throws AEFException
```

Indicates if this tender be refunded at this POS terminal.

Returns:

true if this tender be refunded at this POS terminal

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

canVerify

```
public boolean canVerify()
                  throws AEFException
```

Indicates if this tender be verified at this POS terminal.

Returns:

true if this tender be verified at this POS terminal

Exceptions:

(continued from last page)

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

com.ibm.retail.AEF.event

Interface TenderEvent

All Superinterfaces:
LineItemEvent, POSAppEvent, POSAppEventElement

public interface **TenderEvent**
extends LineItemEvent

A `TenderEvent`is generated when a tender is accepted or cancelled within a sales transaction.

Method Summary

`java.lang.String`	`getAccountNumber()` Get the account number of this tender.
`java.lang.String`	`getAmount()` Get the amount of this tender.
`java.lang.String`	`getCardID()` Get the card ID of this tender.
`java.lang.String`	`getCurrency()` Get the currency of the tender as defined by the three-character country code in ISO 4217.
`java.lang.String`	`getDescription()` Get the textual description of this tender.
`java.lang.String`	`getExpirationDate()` Get the expiration date of this tender.
`java.lang.String`	`getFee()` Get any fee charged for accepting this tender.
`java.lang.String`	`getReferenceNumber()` Get the approval reference number of this tender.
`java.lang.String`	`getTenderType()` Get the type of this tender (cash/credit/check/debit).
`java.lang.String`	`getTenderVariety()` Get the variety of this Tender.

Methods

getTenderType

`public java.lang.String` **`getTenderType`**`()`

(continued from last page)

Get the type of this tender (cash/credit/check/debit). Identifiers for tender types are defined by the POS application.

Returns:

string identification of the tender type - returns null if the property is undefined

getTenderVariety

`public java.lang.String` **`getTenderVariety`**`()`

Get the variety of this Tender. Identifiers for tender varieties are defined by the POS application.

Returns:

string identification of the tender variety - returns null if the property is undefined

getDescription

`public java.lang.String` **`getDescription`**`()`

Get the textual description of this tender.

Returns:

tender description - returns null if the property is undefined

getAmount

`public java.lang.String` **`getAmount`**`()`

Get the amount of this tender.

Returns:

string representation of the tender amount - returns null if the property is undefined

getFee

`public java.lang.String` **`getFee`**`()`

Get any fee charged for accepting this tender.

Returns:

string representation of the fee amount - returns null if the property is undefined

getAccountNumber

`public java.lang.String` **`getAccountNumber`**`()`

Get the account number of this tender.

Returns:

account number - returns null if the property is undefined

getExpirationDate

`public java.lang.String` **`getExpirationDate`**`()`

Get the expiration date of this tender.

Returns:

(continued from last page)

string representation of the expiration date (format is MMYY) - returns null if the property is undefined

getCardID

`public java.lang.String` **`getCardID`**`()`

Get the card ID of this tender. The card id is the security number used as an extension of a credit card account number.

Returns:

card ID - returns null if the property is undefined

getCurrency

`public java.lang.String` **`getCurrency`**`()`

Get the currency of the tender as defined by the three-character country code in ISO 4217.

Returns:

currency - returns null if the property is undefined

getReferenceNumber

`public java.lang.String` **`getReferenceNumber`**`()`

Get the approval reference number of this tender.

Returns:

reference number - returns null if the property is undefined

com.ibm.retail.AEF.event

Interface TenderListener

All Superinterfaces:
POSAppEventListener

All Known Implementing Classes:
TenderListenerProxy

---

public interface **TenderListener**
extends POSAppEventListener

Listener interface for receiving `TenderEvents`from the terminal session. A `TenderEvent`is fired whenever tender is accepted or voided in the POS transaction.

See Also:
com.ibm.retail.AEF.data.POSDataProvider#addTenderListener addTenderListener

---

Method Summary

`void`	`tenderAccepted(TenderEvent evt)` A tender was aaccepted in the transaction

Methods

tenderAccepted

```
public void tenderAccepted(TenderEvent evt)
                    throws java.rmi.RemoteException
```

A tender was aaccepted in the transaction

Parameters:
`evt -`
contains tender details

com.ibm.retail.AEF.event

Interface TerminalOptions

All Superinterfaces:
POSAppEventElement

public interface **TerminalOptions**
extends POSAppEventElement

`TerminalOptions`
is a simple data object to contain information about POS terminal options. Terminal options determine what POS functions are allowed at a specific POS terminal. `TerminalOptions`are accesible from the `OptionsEvent`.

See Also:
com.ibm.retail.AEF.event.OptionsEvent#getTerminalOptions getTerminalOptions

Method Summary

boolean	`isEnabled(java.lang.String propertyName)` Is this terminal enabled for a specific function? Since options are application specific, `isEnabled`provides a method to determine the value of an option based on the property name of the function.

Methods

isEnabled

```
public boolean isEnabled(java.lang.String propertyName)
                   throws AEFException
```

Is this terminal enabled for a specific function? Since options are application specific, `isEnabled`provides a method to determine the value of an option based on the property name of the function. Property names are contained in the `TerminalOptionsProperties`interface.

Parameters:
`name -`
of the property or function

Returns:

true if the function is enabled

Exceptions:
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

See Also:

com.ibm.retail.AEF.data.TerminalOptionsProperties TerminalOptionsProperties

com.ibm.retail.AEF.event

Interface TimeInterval

All Superinterfaces:
POSAppEventElement

public interface **TimeInterval**
extends POSAppEventElement

TimeInterval - Encapsulates a time interval when an item may made available for sale, or restricted from sale.

Field Summary

static int	RESTRICTED
static int	UNRESTRICTED

Method Summary

java.lang.String	getEndDateOfMonth() Returns the end date of the month.
java.lang.String	getEndDayOfWeek() Returns the end day of the week of the interval.
java.lang.String	getEndMonth() Get the end month
java.lang.String	getEndTime() Returns the end of the time interval.
java.lang.String	getStartDateOfMonth() Returns the start date of the month.
java.lang.String	getStartDayOfWeek() Returns the start day of the week of the interval.
java.lang.String	getStartMonth() Returns the start month
java.lang.String	getStartTime() Returns the start of the time interval.
int	getType() Indicates whether the period represents a restricted period, or an unrestricted period.

Fields

(continued from last page)

RESTRICTED

`public static final int` **`RESTRICTED`**

UNRESTRICTED

`public static final int` **`UNRESTRICTED`**

Methods

getStartMonth

`public java.lang.String` **`getStartMonth`**`()`

Returns the start month

Returns:

String

getEndMonth

`public java.lang.String` **`getEndMonth`**`()`

Get the end month

Returns:

String

getStartDateOfMonth

`public java.lang.String` **`getStartDateOfMonth`**`()`

Returns the start date of the month.

Returns:

String

getEndDateOfMonth

`public java.lang.String` **`getEndDateOfMonth`**`()`

Returns the end date of the month.

Returns:

String

getStartDayOfWeek

`public java.lang.String` **`getStartDayOfWeek`**`()`

Returns the start day of the week of the interval.

Returns:

(continued from last page)

String

getEndDayOfWeek

`public java.lang.String` **`getEndDayOfWeek`**`()`

Returns the end day of the week of the interval.

Returns:

String

getStartTime

`public java.lang.String` **`getStartTime`**`()`

Returns the start of the time interval.

Returns:

String

getEndTime

`public java.lang.String` **`getEndTime`**`()`

Returns the end of the time interval.

Returns:

String

getType

`public int` **`getType`**`()`

Indicates whether the period represents a restricted period, or an unrestricted period.

Returns:

int TimeInterval.RESTRICTED, TimeInterval.UNRESTRICTED

com.ibm.retail.AEF.event

Interface TransactionDefinition

All Superinterfaces:
POSAppEventElement

public interface **TransactionDefinition**
extends POSAppEventElement

A `TransactionDefinition`is a simple data object to contain information about POS transactions (Regular Sale, No Sale, COD,..).

The object contains indicators to determine what functions are valid for various transaction types.

Field Summary

static java.lang.String	`ALLOWANCES_ALLOWED` Identifier for transaction allows allowances
static java.lang.String	`DISCOUNTS_ALLOWED` Identifier for transaction allows discounts
static java.lang.String	`DOCUMENT_INSERT_USED` Identifier for document insert used
static java.lang.String	`PAYMENTS_ALLOWED` Identifier for transaction allows payments
static java.lang.String	`PROMPT_FOR_ACCOUNT_NUMBER` Identifier for prompt for account number
static java.lang.String	`PROMPT_FOR_ORIGINAL_SALESPERSON` Identifier for original sales person prompt
static java.lang.String	`PROMPT_FOR_TERMS_OF_SALE` Identifier for terms of sale prompt
static java.lang.String	`RETURNS_ALLOWED` Identifier for transaction allows returns
static java.lang.String	`SUSPEND_TRANSACTION_ALLOWED` Identifier for suspend transaction allowed
static java.lang.String	`TRANSACTION_TYPE` Identifier for transaction type (application specific, typically a short descriptor e.
static java.lang.String	`VOID_TRANSACTION_ALLOWED` Identifier for void transaction allowed

Method Summary

boolean	`allowancesAllowed()` Indicates if this transaction allows allowances to be applied.
boolean	`discountsAllowed()` Indicates if this transaction allows discounts to be applied.
boolean	`documentInsertUsed()` Indicates if this transaction requires a document insert.
java.lang.String	`getType()` Get the transaction type value.
boolean	`paymentsAllowed()` Indicates if this transaction allows payments.
boolean	`promptForAccountNumber()` Indicates if this transaction type requires an account number.
boolean	`promptForOriginalSalesperson()` Indicates if a prompt for original sales person is required.
boolean	`promptForTermsOfSale()` Indicates if a prompt for terms of sale is required.
boolean	`returnsAllowed()` Indicates if this transaction allows returns.
boolean	`suspendTransactionAllowed()` Indicates if suspend transaction is allowed.
boolean	`voidTransactionAllowed()` Indicates if void transaction is allowed.

Fields

TRANSACTION_TYPE

`public static final java.lang.String` **`TRANSACTION_TYPE`**

Identifier for transaction type (application specific, typically a short descriptor e.g., "noSale").

PROMPT_FOR_ACCOUNT_NUMBER

`public static final java.lang.String` **`PROMPT_FOR_ACCOUNT_NUMBER`**

Identifier for prompt for account number

DOCUMENT_INSERT_USED

`public static final java.lang.String` **`DOCUMENT_INSERT_USED`**

Identifier for document insert used

(continued from last page)

PAYMENTS_ALLOWED

`public static final java.lang.String` **`PAYMENTS_ALLOWED`**

Identifier for transaction allows payments

ALLOWANCES_ALLOWED

`public static final java.lang.String` **`ALLOWANCES_ALLOWED`**

Identifier for transaction allows allowances

DISCOUNTS_ALLOWED

`public static final java.lang.String` **`DISCOUNTS_ALLOWED`**

Identifier for transaction allows discounts

RETURNS_ALLOWED

`public static final java.lang.String` **`RETURNS_ALLOWED`**

Identifier for transaction allows returns

PROMPT_FOR_ORIGINAL_SALESPERSON

`public static final java.lang.String` **`PROMPT_FOR_ORIGINAL_SALESPERSON`**

Identifier for original sales person prompt

PROMPT_FOR_TERMS_OF_SALE

`public static final java.lang.String` **`PROMPT_FOR_TERMS_OF_SALE`**

Identifier for terms of sale prompt

VOID_TRANSACTION_ALLOWED

`public static final java.lang.String` **`VOID_TRANSACTION_ALLOWED`**

Identifier for void transaction allowed

SUSPEND_TRANSACTION_ALLOWED

`public static final java.lang.String` **`SUSPEND_TRANSACTION_ALLOWED`**

Identifier for suspend transaction allowed

Methods

getType

`public java.lang.String` **`getType`**`()`

Get the transaction type value. Transaction types are application specific but are typically a short descriptor.

Returns:

the type of transaction

(continued from last page)

promptForAccountNumber

```
public boolean promptForAccountNumber()
                                 throws AEFException
```

Indicates if this transaction type requires an account number.

Returns:

true if this transaction type requires an account number

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

documentInsertUsed

```
public boolean documentInsertUsed()
                           throws AEFException
```

Indicates if this transaction requires a document insert.

Returns:

true if this transaction requires a document insert.

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

paymentsAllowed

```
public boolean paymentsAllowed()
                         throws AEFException
```

Indicates if this transaction allows payments.

Returns:

true if this transaction allows payments.

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

allowancesAllowed

```
public boolean allowancesAllowed()
                           throws AEFException
```

Indicates if this transaction allows allowances to be applied.

Returns:

true if this transaction allows allowances to be applied.

Exceptions:

(continued from last page)

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

discountsAllowed

```
public boolean discountsAllowed()
                          throws AEFException
```

Indicates if this transaction allows discounts to be applied.

Returns:

true if this transaction allows discounts to be applied.

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

returnsAllowed

```
public boolean returnsAllowed()
                        throws AEFException
```

Indicates if this transaction allows returns.

Returns:

true if this transaction allows returns.

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

promptForOriginalSalesperson

```
public boolean promptForOriginalSalesperson()
                                      throws AEFException
```

Indicates if a prompt for original sales person is required.

Returns:

true if a prompt for original sales person is required.

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

promptForTermsOfSale

```
public boolean promptForTermsOfSale()
                              throws AEFException
```

Indicates if a prompt for terms of sale is required.

Returns:

(continued from last page)

true if a prompt for terms of sale is required.

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

voidTransactionAllowed

```
public boolean voidTransactionAllowed()
                                throws AEFException
```

Indicates if void transaction is allowed.

Returns:

true if void transaction is allowed.

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

suspendTransactionAllowed

```
public boolean suspendTransactionAllowed()
                                   throws AEFException
```

Indicates if suspend transaction is allowed.

Returns:

true if suspend transaction is allowed.

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

com.ibm.retail.AEF.event

Interface TransactionDiscountReasonCode

All Superinterfaces:
DiscountReasonCode, ReasonCode, POSAppEventElement

public interface **TransactionDiscountReasonCode**
extends DiscountReasonCode

`TransactionDiscountReasonCode`
is a simple data object to contain information about POS discount reason code for a transaction.

A transaction discount reason code contains a numeric code, a text description, the rate (percentage) of the discount. It also includes tax plan exemption flags.

Field Summary

static java.lang.String	`TAX_EXEMPTION` Identifier for Tax Exemption property name
static java.lang.String	`TAX_PLAN1_EXEMPT` Identifier for Tax Plan1 exempt property name
static java.lang.String	`TAX_PLAN2_EXEMPT` Identifier for Tax Plan2 exempt property name
static java.lang.String	`TAX_PLAN3_EXEMPT` Identifier for Tax Plan3 exempt property name
static java.lang.String	`TAX_PLAN4_EXEMPT` Identifier for Tax Plan4 exempt property name
static java.lang.String	`TAX_PLAN5_EXEMPT` Identifier for Tax Plan5 exempt property name
static java.lang.String	`TAX_PLAN6_EXEMPT` Identifier for Tax Plan6 exempt property name
static java.lang.String	`TAX_PLAN7_EXEMPT` Identifier for Tax Plan7 exempt property name
static java.lang.String	`TAX_PLAN8_EXEMPT` Identifier for Tax Plan8 exempt property name

Method Summary

boolean	`isTaxExemption()` Indicates if this discount is a tax exemption.

boolean	`isTaxPlan1Exemption()` Is this discount a tax exemption to a tax plan1?
boolean	`isTaxPlan2Exemption()` Is this discount a tax exemption to a tax plan2?
boolean	`isTaxPlan3Exemption()` Is this discount a tax exemption to a tax plan3?
boolean	`isTaxPlan4Exemption()` Is this discount a tax exemption to a tax plan4?
boolean	`isTaxPlan5Exemption()` Is this discount a tax exemption to a tax plan5?
boolean	`isTaxPlan6Exemption()` Is this discount a tax exemption to a tax plan6?
boolean	`isTaxPlan7Exemption()` Is this discount a tax exemption to a tax plan7?
boolean	`isTaxPlan8Exemption()` Is this discount a tax exemption to a tax plan8?

Fields

TAX_EXEMPTION

`public static final java.lang.String` **`TAX_EXEMPTION`**

Identifier for Tax Exemption property name

TAX_PLAN1_EXEMPT

`public static final java.lang.String` **`TAX_PLAN1_EXEMPT`**

Identifier for Tax Plan1 exempt property name

TAX_PLAN2_EXEMPT

`public static final java.lang.String` **`TAX_PLAN2_EXEMPT`**

Identifier for Tax Plan2 exempt property name

TAX_PLAN3_EXEMPT

`public static final java.lang.String` **`TAX_PLAN3_EXEMPT`**

Identifier for Tax Plan3 exempt property name

TAX_PLAN4_EXEMPT

`public static final java.lang.String` **`TAX_PLAN4_EXEMPT`**

Identifier for Tax Plan4 exempt property name

TAX_PLAN5_EXEMPT

`public static final java.lang.String` **`TAX_PLAN5_EXEMPT`**

Identifier for Tax Plan5 exempt property name

TAX_PLAN6_EXEMPT

`public static final java.lang.String` **`TAX_PLAN6_EXEMPT`**

Identifier for Tax Plan6 exempt property name

TAX_PLAN7_EXEMPT

`public static final java.lang.String` **`TAX_PLAN7_EXEMPT`**

Identifier for Tax Plan7 exempt property name

TAX_PLAN8_EXEMPT

`public static final java.lang.String` **`TAX_PLAN8_EXEMPT`**

Identifier for Tax Plan8 exempt property name

Methods

isTaxExemption

```
public boolean isTaxExemption()
                        throws AEFException
```

Indicates if this discount is a tax exemption.

Returns:

true if this discount is a tax exemption.

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

isTaxPlan1Exemption

```
public boolean isTaxPlan1Exemption()
                              throws AEFException
```

Is this discount a tax exemption to a tax plan1?

Returns:

boolean

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

(continued from last page)

isTaxPlan2Exemption

```
public boolean isTaxPlan2Exemption()
                              throws AEFException
```

Is this discount a tax exemption to a tax plan2?

Returns:

boolean

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

isTaxPlan3Exemption

```
public boolean isTaxPlan3Exemption()
                              throws AEFException
```

Is this discount a tax exemption to a tax plan3?

Returns:

boolean

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

isTaxPlan4Exemption

```
public boolean isTaxPlan4Exemption()
                              throws AEFException
```

Is this discount a tax exemption to a tax plan4?

Returns:

boolean

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

isTaxPlan5Exemption

```
public boolean isTaxPlan5Exemption()
                              throws AEFException
```

Is this discount a tax exemption to a tax plan5?

Returns:

boolean

Exceptions:

(continued from last page)

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

isTaxPlan6Exemption

```
public boolean isTaxPlan6Exemption()
                              throws AEFException
```

Is this discount a tax exemption to a tax plan6?

Returns:

boolean

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

isTaxPlan7Exemption

```
public boolean isTaxPlan7Exemption()
                              throws AEFException
```

Is this discount a tax exemption to a tax plan7?

Returns:

boolean

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

isTaxPlan8Exemption

```
public boolean isTaxPlan8Exemption()
                              throws AEFException
```

Is this discount a tax exemption to a tax plan8?

Returns:

boolean

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

com.ibm.retail.AEF.event

Interface TransactionStatusEvent

All Superinterfaces:
POSAppEvent, POSAppEventElement

public interface **TransactionStatusEvent**
extends POSAppEvent

A `TransactionStatusEvent` is created when the status of a sales transaction is changed.

Status changes occur when a transaction starts and ends, or when the allowed actions for a transaction are updated (e.g., discount allowed).

Field Summary

static java.lang.String	`ALTERNATE_TAX_TYPE` Transaction tax type: Tax type for "alternate tax" transaction tax changes
static java.lang.String	`CASH` Transaction type identifier: Cash
static java.lang.String	`CASH_DOCUMENT` Transaction type identifier: Cash Document transaction type.
static java.lang.String	`CASH_SPECIAL` Transaction type identifier: Cash Special transaction type.
static java.lang.String	`CHARGE_PLAN_A` Transaction type identifier: Charge Plan A transaction type.
static java.lang.String	`CHARGE_PLAN_B` Transaction type identifier: Charge Plan B transaction type.
static java.lang.String	`CHARGE_PLAN_C` Transaction type identifier: Charge Plan C transaction type.
static java.lang.String	`CHARGE_PLAN_D` Transaction type identifier: Charge Plan D transaction type.
static java.lang.String	`COD` Transaction type identifier: COD transaction type.
static java.lang.String	`DEPT_TOTALS_REPORT` Transaction type identifier: Department Totals Report transaction type
static java.lang.String	`EBT_BALANCE_INQUIRY` Transaction type identifier: EBT Balance Inquiry transaction type
static java.lang.String	`EFT_BALANACE_INQUIRY` Transaction type identifier: EFT Balance Inquiry transaction type

static java.lang.String	EXEMPT_TAX_TYPE Transaction tax type: "tax exempt" transaction tax changes
static java.lang.String	ITEM_MOVEMENT_REPORT Transaction type identifier: Item Movement Report transaction type.
static java.lang.String	ITEM_PRICE_CHANGE Transaction type identifier: Item Price Change transaction type.
static java.lang.String	ITEM_RETURN Transaction type identifier: Item Return transaction type
static java.lang.String	LAYAWAY Transaction type identifier: Layaway transaction type.
static java.lang.String	LAYAWAY_CANCEL Transaction type identifier: Layaway Cancel transaction type
static java.lang.String	LAYAWAY_PAYMENT Transaction type identifier: Layaway Payment transaction type
static java.lang.String	LOAN Transaction type identifier: Loan transaction type.
static java.lang.String	MANUAL_TAX_TYPE Transaction tax type: Tax type for "manual tax" transaction tax changes
static java.lang.String	MODIFY_DEPT_PRESETS Transaction type identifier: Modify Department Presets transaction type
static java.lang.String	NO_SALE Transaction type identifier: No Sale
static java.lang.String	NONSALES_CATEGORY Transaction category: "nonsales" transactions
static java.lang.String	NOSALE_PRICE_VERIFY Transaction type identifier: Nosale Price Verify transaction type
static java.lang.String	OFFLINE_REENTRY Transaction type identifier: Offline Reentry transaction type.
static java.lang.String	OPEN_TRANS_REPORT Transaction type identifier: Open Transaction Report transaction type
static java.lang.String	PICKUP Transaction type identifier: Pickup transaction type.
static java.lang.String	PRICE_CHANGE Transaction type identifier: Price Change transaction type
static java.lang.String	PRICE_VERIFY Transaction type identifier: Price Verify transaction type

static java.lang.String	QUERY_EXCHANGE_RATE Transaction type identifier: Query Exchange Rate transaction type.
static java.lang.String	REGULAR_SALE Transaction type identifier: Regular Sale
static java.lang.String	REGULAR_SALE_TRAINING Transaction type modifier: Regular Sale Training
static java.lang.String	REPRINT_PARTIAL Transaction type identifier: Reprint Partial Receipt transaction type
static java.lang.String	REPRINT_TENDER_RECEIPT Transaction type identifier: Reprint Tender Receipt transaction type
static java.lang.String	SALES_CATEGORY Transaction category: "sales" transactions
static java.lang.String	SET_TRANS_NUMBER Transaction type identifier: Set Transaction Number transaction type.
static java.lang.String	SUSPENDED_TRANS_REPORT Transaction type identifier: Suspended Transaction Report transaction type
static java.lang.String	TENDER_CASHING Transaction type identifier: Tender Cashing transaction type.
static java.lang.String	TENDER_COUNT Transaction type identifier: Tender Count transaction type.
static java.lang.String	TENDER_EXCHANGE Transaction type identifier: Tender Exchange transaction type
static java.lang.String	TENDER_FEE_REFUND Transaction type identifier: Tender Fee Refund transaction type
static java.lang.String	TENDER_LIST_REPORT Transaction type identifier: Tender Listing Report transaction type.
static java.lang.String	TENDER_REMOVAL Transaction type identifier: Tender Removal transaction type.
static java.lang.String	TERMINAL_MONITOR Transaction type identifier: Terminal Monitor transaction type.
static java.lang.String	TERMINAL_PROGRAM_LOAD Transaction type identifier: Terminal Program Load transaction type
static java.lang.String	TERMINAL_TRANSFER Transaction type identifier: Terminal Transfer transaction type
static java.lang.String	TILL_EXCHANGE Transaction type identifier: Till Exchange transaction type

static java.lang.String	TILL_REPORT Transaction type identifier: Till Report transaction type
static java.lang.String	TOTALS_READOUT_RESET Transaction type identifier: Totals Readout Reset transaction type.
static java.lang.String	TRAINING Transaction type identifier: Training transaction type.
static java.lang.String	TRANSACTION_END Event type identifier: transaction ended.
static java.lang.String	TRANSACTION_START Event type identifier: transaction started.
static java.lang.String	TRANSACTION_SUSPEND Event type identifier: transaction suspended.
static java.lang.String	TRANSACTION_TAX_CHANGE Event type identifier: transaction tax change.
static java.lang.String	TRANSACTION_UPDATE Event type identifier: transaction status updated.
static java.lang.String	TRANSACTION_VOID Event type identifier: transaction voided.
static java.lang.String	VALUE_CARD_BAL_INQUIRY Transaction type identifier: Value Card Balance Inquiry transaction type
static java.lang.String	VERIFY_TENDER Transaction type identifier: Verify Tender transaction type
static java.lang.String	VOID_PREVIOUS_BY_LINE Transaction type identifier: Void Previous By Line Item transaction type.
static java.lang.String	VOID_PREVIOUS_TRANS Transaction type identifier: Void Previous Transaction transaction type.
static java.lang.String	WIC Transaction type identifier: WIC transaction type

Method Summary

boolean	cancelAllItems() For "layaway cancel" transactions, indicates if the cancel applies to all items.
java.lang.String	getAccountNumber() Get the account number for this transaction.
java.lang.String	getCategory() Get the transaction category identifier (sales/nonsales).

java.lang.String	getDate() Get the transaction date.
java.lang.String	getID() Get the transaction identifier.
java.lang.String	getModifier() Get the transaction modifier (used for additional transaction information).
java.lang.String	getTaxReason() Get the Tax Reason (tax code or discount group).
java.lang.String	getTaxType() Get the Tax Type (exempt, manual, alternate).
java.lang.String	getTime() Get the transaction time.
java.lang.String	getType() Get the transaction type.
boolean	isItemAllowanceAllowed() Indicates if an item allowance is allowed for this transaction.
boolean	isItemDiscountAllowed() Indicates if an item discount is allowed for this transaction.
boolean	isTaxChangeVoided() Indicates if this tax change is voided for this transaction.
boolean	isTransactionDiscountAllowed() Indicates if a transaction discount is allowed for this transaction.
boolean	isVoidLineItemAllowed() Indicates if a void line item is allowed for this transaction.
boolean	returnPayments() For "layaway cancel" transactions, indicates if payments are returned to the customer.

Fields

TRANSACTION_START

`public static final java.lang.String` **`TRANSACTION_START`**

Event type identifier: transaction started.

See Also:

#getEventType()

(continued from last page)

TRANSACTION_END

`public static final java.lang.String` **`TRANSACTION_END`**

Event type identifier: transaction ended.

See Also:

#getEventType

TRANSACTION_VOID

`public static final java.lang.String` **`TRANSACTION_VOID`**

Event type identifier: transaction voided.

See Also:

#getEventType()

TRANSACTION_SUSPEND

`public static final java.lang.String` **`TRANSACTION_SUSPEND`**

Event type identifier: transaction suspended.

See Also:

#getEventType()

TRANSACTION_UPDATE

`public static final java.lang.String` **`TRANSACTION_UPDATE`**

Event type identifier: transaction status updated.

See Also:

#getEventType()

TRANSACTION_TAX_CHANGE

`public static final java.lang.String` **`TRANSACTION_TAX_CHANGE`**

Event type identifier: transaction tax change.

See Also:

#getEventType()

REGULAR_SALE

`public static final java.lang.String` **`REGULAR_SALE`**

Transaction type identifier: Regular Sale

See Also:

#getType()

(continued from last page)

NO_SALE

`public static final java.lang.String` **`NO_SALE`**

Transaction type identifier: No Sale

See Also:

#getType()

CASH

`public static final java.lang.String` **`CASH`**

Transaction type identifier: Cash

See Also:

#getType()

CASH_SPECIAL

`public static final java.lang.String` **`CASH_SPECIAL`**

Transaction type identifier: Cash Special transaction type.

See Also:

#getType()

CASH_DOCUMENT

`public static final java.lang.String` **`CASH_DOCUMENT`**

Transaction type identifier: Cash Document transaction type.

See Also:

#getType()

COD

`public static final java.lang.String` **`COD`**

Transaction type identifier: COD transaction type.

See Also:

#getType()

LAYAWAY

`public static final java.lang.String` **`LAYAWAY`**

Transaction type identifier: Layaway transaction type.

See Also:

#getType()

(continued from last page)

CHARGE_PLAN_A

`public static final java.lang.String` **`CHARGE_PLAN_A`**

Transaction type identifier: Charge Plan A transaction type.

See Also:

#getType()

CHARGE_PLAN_B

`public static final java.lang.String` **`CHARGE_PLAN_B`**

Transaction type identifier: Charge Plan B transaction type.

See Also:

#getType()

CHARGE_PLAN_C

`public static final java.lang.String` **`CHARGE_PLAN_C`**

Transaction type identifier: Charge Plan C transaction type.

See Also:

#getType()

CHARGE_PLAN_D

`public static final java.lang.String` **`CHARGE_PLAN_D`**

Transaction type identifier: Charge Plan D transaction type.

See Also:

#getType()

LOAN

`public static final java.lang.String` **`LOAN`**

Transaction type identifier: Loan transaction type.

See Also:

#getType()

PICKUP

`public static final java.lang.String` **`PICKUP`**

Transaction type identifier: Pickup transaction type.

See Also:

#getType()

(continued from last page)

TENDER_COUNT

`public static final java.lang.String` **`TENDER_COUNT`**

Transaction type identifier: Tender Count transaction type.

See Also:

#getType()

TOTALS_READOUT_RESET

`public static final java.lang.String` **`TOTALS_READOUT_RESET`**

Transaction type identifier: Totals Readout Reset transaction type.

See Also:

#getType()

ITEM_MOVEMENT_REPORT

`public static final java.lang.String` **`ITEM_MOVEMENT_REPORT`**

Transaction type identifier: Item Movement Report transaction type.

See Also:

#getType()

ITEM_PRICE_CHANGE

`public static final java.lang.String` **`ITEM_PRICE_CHANGE`**

Transaction type identifier: Item Price Change transaction type.

See Also:

#getType()

SET_TRANS_NUMBER

`public static final java.lang.String` **`SET_TRANS_NUMBER`**

Transaction type identifier: Set Transaction Number transaction type.

See Also:

#getType()

OFFLINE_REENTRY

`public static final java.lang.String` **`OFFLINE_REENTRY`**

Transaction type identifier: Offline Reentry transaction type.

See Also:

#getType()

(continued from last page)

TRAINING

`public static final java.lang.String` **`TRAINING`**

Transaction type identifier: Training transaction type.

See Also:

#getType()

VOID_PREVIOUS_TRANS

`public static final java.lang.String` **`VOID_PREVIOUS_TRANS`**

Transaction type identifier: Void Previous Transaction transaction type.

See Also:

#getType()

VOID_PREVIOUS_BY_LINE

`public static final java.lang.String` **`VOID_PREVIOUS_BY_LINE`**

Transaction type identifier: Void Previous By Line Item transaction type.

See Also:

#getType()

TENDER_LIST_REPORT

`public static final java.lang.String` **`TENDER_LIST_REPORT`**

Transaction type identifier: Tender Listing Report transaction type.

See Also:

#getType()

TENDER_REMOVAL

`public static final java.lang.String` **`TENDER_REMOVAL`**

Transaction type identifier: Tender Removal transaction type.

See Also:

#getType()

TERMINAL_MONITOR

`public static final java.lang.String` **`TERMINAL_MONITOR`**

Transaction type identifier: Terminal Monitor transaction type.

See Also:

#getType()

(continued from last page)

QUERY_EXCHANGE_RATE

`public static final java.lang.String` **`QUERY_EXCHANGE_RATE`**

Transaction type identifier: Query Exchange Rate transaction type.

See Also:

#getType()

TENDER_CASHING

`public static final java.lang.String` **`TENDER_CASHING`**

Transaction type identifier: Tender Cashing transaction type.

See Also:

#getType()

TENDER_EXCHANGE

`public static final java.lang.String` **`TENDER_EXCHANGE`**

Transaction type identifier: Tender Exchange transaction type

See Also:

#getType()

PRICE_VERIFY

`public static final java.lang.String` **`PRICE_VERIFY`**

Transaction type identifier: Price Verify transaction type

See Also:

#getType()

TERMINAL_TRANSFER

`public static final java.lang.String` **`TERMINAL_TRANSFER`**

Transaction type identifier: Terminal Transfer transaction type

See Also:

#getType()

TERMINAL_PROGRAM_LOAD

`public static final java.lang.String` **`TERMINAL_PROGRAM_LOAD`**

Transaction type identifier: Terminal Program Load transaction type

See Also:

#getType()

(continued from last page)

ITEM_RETURN

`public static final java.lang.String` **`ITEM_RETURN`**

Transaction type identifier: Item Return transaction type

See Also:

#getType()

WIC

`public static final java.lang.String` **`WIC`**

Transaction type identifier: WIC transaction type

See Also:

#getType()

REPRINT_PARTIAL

`public static final java.lang.String` **`REPRINT_PARTIAL`**

Transaction type identifier: Reprint Partial Receipt transaction type

See Also:

#getType()

REPRINT_TENDER_RECEIPT

`public static final java.lang.String` **`REPRINT_TENDER_RECEIPT`**

Transaction type identifier: Reprint Tender Receipt transaction type

See Also:

#getType()

EBT_BALANCE_INQUIRY

`public static final java.lang.String` **`EBT_BALANCE_INQUIRY`**

Transaction type identifier: EBT Balance Inquiry transaction type

See Also:

#getType()

VALUE_CARD_BAL_INQUIRY

`public static final java.lang.String` **`VALUE_CARD_BAL_INQUIRY`**

Transaction type identifier: Value Card Balance Inquiry transaction type

See Also:

#getType()

(continued from last page)

DEPT_TOTALS_REPORT

`public static final java.lang.String` **`DEPT_TOTALS_REPORT`**

Transaction type identifier: Department Totals Report transaction type

See Also:

#getType()

LAYAWAY_PAYMENT

`public static final java.lang.String` **`LAYAWAY_PAYMENT`**

Transaction type identifier: Layaway Payment transaction type

See Also:

#getType()

LAYAWAY_CANCEL

`public static final java.lang.String` **`LAYAWAY_CANCEL`**

Transaction type identifier: Layaway Cancel transaction type

See Also:

#getType()

PRICE_CHANGE

`public static final java.lang.String` **`PRICE_CHANGE`**

Transaction type identifier: Price Change transaction type

See Also:

#getType()

TENDER_FEE_REFUND

`public static final java.lang.String` **`TENDER_FEE_REFUND`**

Transaction type identifier: Tender Fee Refund transaction type

See Also:

#getType()

SUSPENDED_TRANS_REPORT

`public static final java.lang.String` **`SUSPENDED_TRANS_REPORT`**

Transaction type identifier: Suspended Transaction Report transaction type

See Also:

#getType()

(continued from last page)

MODIFY_DEPT_PRESETS

`public static final java.lang.String` **`MODIFY_DEPT_PRESETS`**

Transaction type identifier: Modify Department Presets transaction type

See Also:

#getType()

OPEN_TRANS_REPORT

`public static final java.lang.String` **`OPEN_TRANS_REPORT`**

Transaction type identifier: Open Transaction Report transaction type

See Also:

#getType()

EFT_BALANACE_INQUIRY

`public static final java.lang.String` **`EFT_BALANACE_INQUIRY`**

Transaction type identifier: EFT Balance Inquiry transaction type

See Also:

#getType()

TILL_EXCHANGE

`public static final java.lang.String` **`TILL_EXCHANGE`**

Transaction type identifier: Till Exchange transaction type

See Also:

#getType()

VERIFY_TENDER

`public static final java.lang.String` **`VERIFY_TENDER`**

Transaction type identifier: Verify Tender transaction type

See Also:

#getType()

TILL_REPORT

`public static final java.lang.String` **`TILL_REPORT`**

Transaction type identifier: Till Report transaction type

See Also:

#getType()

(continued from last page)

NOSALE_PRICE_VERIFY

`public static final java.lang.String` **`NOSALE_PRICE_VERIFY`**

Transaction type identifier: Nosale Price Verify transaction type

See Also:

#getType()

REGULAR_SALE_TRAINING

`public static final java.lang.String` **`REGULAR_SALE_TRAINING`**

Transaction type modifier: Regular Sale Training

See Also:

#getModifier()

SALES_CATEGORY

`public static final java.lang.String` **`SALES_CATEGORY`**

Transaction category: "sales" transactions

See Also:

#getCategory()

NONSALES_CATEGORY

`public static final java.lang.String` **`NONSALES_CATEGORY`**

Transaction category: "nonsales" transactions

See Also:

#getCategory()

EXEMPT_TAX_TYPE

`public static final java.lang.String` **`EXEMPT_TAX_TYPE`**

Transaction tax type: "tax exempt" transaction tax changes

See Also:

#getTaxType()

ALTERNATE_TAX_TYPE

`public static final java.lang.String` **`ALTERNATE_TAX_TYPE`**

Transaction tax type: Tax type for "alternate tax" transaction tax changes

See Also:

#getTaxType()

(continued from last page)

MANUAL_TAX_TYPE

`public static final java.lang.String` **`MANUAL_TAX_TYPE`**

Transaction tax type: Tax type for "manual tax" transaction tax changes

See Also:

#getTaxType()

Methods

getID

`public java.lang.String` **`getID`**`()`

Get the transaction identifier.

Returns:

a string representation of the transaction number

---

getDate

`public java.lang.String` **`getDate`**`()`

Get the transaction date.

Returns:

a string representation of the transaction date (format: YYMMDD)

---

getModifier

`public java.lang.String` **`getModifier`**`()`

Get the transaction modifier (used for additional transaction information).

Returns:

String

---

getCategory

`public java.lang.String` **`getCategory`**`()`

Get the transaction category identifier (sales/nonsales).

Returns:

transaction category (sales or nonsales). Among the possible values are:
TransactionStatusEvent.SALES_CATEGORY
TransactionStatusEvent.NOSALES_CATEGORY

---

getType

`public java.lang.String` **`getType`**`()`

Get the transaction type. Transaction types include cash, loan, pickup. The list of identifiers for transaction types are defined in this interface *(See field identifiers labeled "Transaction type identifier").*

Returns:

(continued from last page)

type of transaction. Among the possible values are:
TransactionStatusEvent.CASH
TransactionStatusEvent.NO_SALE

getTime

`public java.lang.String getTime()`

Get the transaction time.

Returns:

a string representation of the transaction time (format: HH:MM)

getAccountNumber

`public java.lang.String getAccountNumber()`

Get the account number for this transaction. Typically only used for layaway or "charge plan" transactions. Account numbers for credit/debit are entered during tendering and are not passed through this event.

Returns:

account number

returnPayments

```
public boolean returnPayments()
                        throws AEFException
```

For "layaway cancel" transactions, indicates if payments are returned to the customer.

Returns:

true if payments are returned to the customer

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

cancelAllItems

```
public boolean cancelAllItems()
                        throws AEFException
```

For "layaway cancel" transactions, indicates if the cancel applies to all items.

Returns:

true if cancel applies to all items.

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

(continued from last page)

isItemAllowanceAllowed

```
public boolean isItemAllowanceAllowed()
                                 throws AEFException
```

Indicates if an item allowance is allowed for this transaction.

Returns:

true if an item allowance is allowed for this transaction.

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

isItemDiscountAllowed

```
public boolean isItemDiscountAllowed()
                              throws AEFException
```

Indicates if an item discount is allowed for this transaction.

Returns:

true if an item discount is allowed for this transaction.

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

isTransactionDiscountAllowed

```
public boolean isTransactionDiscountAllowed()
                                     throws AEFException
```

Indicates if a transaction discount is allowed for this transaction.

Returns:

true if a transaction discount is allowed for this transaction.

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

isVoidLineItemAllowed

```
public boolean isVoidLineItemAllowed()
                              throws AEFException
```

Indicates if a void line item is allowed for this transaction.

Returns:

true if a void line item allowed for this transaction.

Exceptions:

(continued from last page)

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

getTaxType

`public java.lang.String` **`getTaxType`**`()`

Get the Tax Type (exempt, manual, alternate). (Valid for transaction tax change events only)

Returns:

identifier of tax type. Among the possible values are:
TransactionStatusEvent.EXEMPT_TAX_TYPE
TransactionStatusEvent.ALTERNATE_TAX_TYPE
TransactionStatusEvent.MANUAL_TAX_TYPE

getTaxReason

`public java.lang.String` **`getTaxReason`**`()`

Get the Tax Reason (tax code or discount group). (Valid for transaction tax change events only)

Returns:

string representation or identifier of the tax reason.

isTaxChangeVoided

```
public boolean isTaxChangeVoided()
                          throws AEFException
```

Indicates if this tax change is voided for this transaction. (Valid for transaction tax change events only)

Returns:

boolean

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.NO_SUCH_PROPERTY
AEFConst.INVALID_PROPERTY_VALUE

com.ibm.retail.AEF.event

Interface TransactionStatusListener

All Superinterfaces:
POSAppEventListener

All Known Implementing Classes:
TransactionStatusListenerProxy

---

public interface **TransactionStatusListener**
extends POSAppEventListener

Listener interface for receiving transaction status events from the terminal session. Transaction status events occur when a transaction starts, ends, or transaction properties change.

See Also:
com.ibm.retail.AEF.data.POSDataProvider#addTransactionStatusListener addTransactionStatusListener

---

Method Summary

`void`	`transactionStatusEventOccurred(TransactionStatusEvent evt)` A transaction status update has occurred.

Methods

transactionStatusEventOccurred

```
public void transactionStatusEventOccurred(TransactionStatusEvent evt)
                                    throws java.rmi.RemoteException
```

A transaction status update has occurred.

Parameters:
`evt` -
contains details of the event

com.ibm.retail.AEF.event

Interface TransactionTotalsEvent

All Superinterfaces:
POSAppEvent, POSAppEventElement

public interface **TransactionTotalsEvent**
extends POSAppEvent

A `TransactionTotalsEvent`is created when the totals for a sales transaction are updated. The event object includes any properties describing the transaction amounts such as the balance due, tax amount, change due, etc. Monetary amounts are returned as strings in the format provided by the base POS application.

Method Summary

`java.lang.String`	`getAmountDue()` Get total amount due.
`java.lang.String`	`getChangeDue()` Get change due.
`java.lang.String`	`getCouponTotal()` Get coupon total.
`java.lang.String`	`getFoodStampBalance()` Get food stamp balance.
`java.lang.String`	`getFoodStampChange()` Get food stamp change.
`java.lang.String`	`getFoodStampTotal()` Get food stamp total.
`java.lang.String`	`getForeignBalanceDue()` Get the foreign balance due.
`java.lang.String`	`getLayawayBalanceDue()` Get layaway balance due.
`java.lang.String`	`getLayawayDeposit()` Get layaway deposit.
`java.lang.String`	`getLayawayFee()` Get layaway fee.
`java.lang.String`	`getNumberOfCoupons()` Get the total number of coupons.
`java.lang.String`	`getSubTotal()` Get transaction subTotal.

java.lang.String	getTax() Get tax total.
java.lang.String	getTenderExchanged() Get tender exchanged.
java.lang.String	getTotal() Get transaction total.
java.lang.String	getTotalItems() Get total items.
java.lang.String	getTotalSavings() Get total savings.

Methods

getTotal

`public java.lang.String **getTotal**()`

Get transaction total.

Returns:

a string representation of the transaction total - null if property is unsupported or not contained in this event

getSubTotal

`public java.lang.String **getSubTotal**()`

Get transaction subTotal.

Returns:

a string representation of the transaction subtotal - null if property is unsupported or not contained in this event

getTotalSavings

`public java.lang.String **getTotalSavings**()`

Get total savings.

Returns:

a string representation of the total savings in this transaction - null if property is unsupported or not contained in this event

getTax

`public java.lang.String **getTax**()`

Get tax total.

Returns:

a string representation of the total tax amount - null if property is unsupported or not contained in this event

getAmountDue

`public java.lang.String` **`getAmountDue`**`()`

Get total amount due.

Returns:

a string representation of the total amount due - null if property is unsupported or not contained in this event

getChangeDue

`public java.lang.String` **`getChangeDue`**`()`

Get change due.

Returns:

a string representation of the change due the customer - null if property is unsupported or not contained in this event

getFoodStampTotal

`public java.lang.String` **`getFoodStampTotal`**`()`

Get food stamp total.

Returns:

a string representation of the food stamp total - null if property is unsupported or not contained in this event

getFoodStampBalance

`public java.lang.String` **`getFoodStampBalance`**`()`

Get food stamp balance.

Returns:

a string representation of the food stamp balance - null if property is unsupported

getFoodStampChange

`public java.lang.String` **`getFoodStampChange`**`()`

Get food stamp change.

Returns:

a string representation of the food stamp change - null if property is unsupported

getTenderExchanged

`public java.lang.String` **`getTenderExchanged`**`()`

Get tender exchanged.

Returns:

a string representation of the tender amount - null if property is unsupported or not contained in this event

(continued from last page)

getLayawayBalanceDue

`public java.lang.String` **`getLayawayBalanceDue()`**

Get layaway balance due.

Returns:

a string representation of the layaway balance - null if property is unsupported or not contained in this event

---

getLayawayDeposit

`public java.lang.String` **`getLayawayDeposit()`**

Get layaway deposit.

Returns:

a string representation of the layaway deposit - null if property is unsupported or not contained in this event

---

getLayawayFee

`public java.lang.String` **`getLayawayFee()`**

Get layaway fee.

Returns:

a string representation of the layaway fee - null if property is unsupported or not contained in this event

---

getForeignBalanceDue

`public java.lang.String` **`getForeignBalanceDue()`**

Get the foreign balance due.

Returns:

a string representation of the foreign balance due - null if property is unsupported or not contained in this event

---

getCouponTotal

`public java.lang.String` **`getCouponTotal()`**

Get coupon total.

Returns:

a string representation of the total value of coupons - null if property is unsupported or not contained in this event

---

getTotalItems

`public java.lang.String` **`getTotalItems()`**

Get total items.

Returns:

a string representation of the total number of items - null if property is unsupported or not contained in this event

---

(continued from last page)

getNumberOfCoupons

`public java.lang.String` **`getNumberOfCoupons`**`()`

Get the total number of coupons.

Returns:

a string representation of the total number of coupons - null if property is unsupported or not contained in this event

com.ibm.retail.AEF.event

Interface TransactionTotalsListener

All Superinterfaces:
POSAppEventListener

All Known Implementing Classes:
TransactionTotalsListenerProxy

---

public interface **TransactionTotalsListener**
extends POSAppEventListener

Listener interface for receiving transaction running totals from the terminal session.

See Also:
com.ibm.retail.AEF.data.POSDataProvider#addTransactionTotalsListener addTransactionTotalsListener

---

Method Summary

`void`	`totalsChanged(TransactionTotalsEvent evt)` A transaction total was updated

Methods

totalsChanged

```
public void totalsChanged(TransactionTotalsEvent evt)
                    throws java.rmi.RemoteException
```

A transaction total was updated

Parameters:
`evt` -
contains details of the totals event

Exceptions:
`RemoteException` -
if a listener can not be notified

com.ibm.retail.AEF.event

Interface WorkstationStatusEvent

All Superinterfaces:
POSAppEvent, POSAppEventElement

public interface **WorkstationStatusEvent**
extends POSAppEvent

`WorkstationStatusEvent`
is an event object generated when the status of the POS workstation changes. String identifiers for the event types are passed through the `getTerminalStatus()`method.

Example usage:

```
if (evt.getTerminalStatus() != null)
{
  if (evt.getTerminalStatus().equals(WorkstationStatusEvent.INPUT_SEQUENCE_CLEARED)
  {
      System.out.println("Input sequence cleared");
  }
}
```

Field Summary

static java.lang.String	`DISABLE_TERMINAL_ACKNOWLEDGED` Terminal status identifier for "terminal has acknowledged disable terminal command"
static java.lang.String	`INPUT_SEQUENCE_CLEARED` Terminal status identifier for "input sequence was cleared" events.
static java.lang.String	`IS_DOUBLE_CLEAR` Double clear was pressed.

Method Summary

java.lang.String	`getTerminalStatus()` Get the current terminal status
void	`setTerminalStatus(java.lang.String status)` Set the terminal device status

Fields

INPUT_SEQUENCE_CLEARED

`public static final java.lang.String` **`INPUT_SEQUENCE_CLEARED`**

Terminal status identifier for "input sequence was cleared" events. To determine if the double clear key was pressed, use `getBooleanProperty(WorkstationStatusEvent.IS_DOUBLE_CLEAR)`

IS_DOUBLE_CLEAR

`public static final java.lang.String` **`IS_DOUBLE_CLEAR`**

Double clear was pressed.

DISABLE_TERMINAL_ACKNOWLEDGED

`public static final java.lang.String` **`DISABLE_TERMINAL_ACKNOWLEDGED`**

Terminal status identifier for "terminal has acknowledged disable terminal command"

Methods

getTerminalStatus

`public java.lang.String` **`getTerminalStatus`**`()`

Get the current terminal status

Returns:

identifer for terminal status

See Also:

com.ibm.retail.AEF.event.WorkstationStatusEvent#INPUT_SEQUENCE_CLEARED
INPUT_SEQUENCE_CLEARED

setTerminalStatus

`public void` **`setTerminalStatus`**`(java.lang.String status)`

Set the terminal device status

Parameters:

`status -`
the identifier of the current status

See Also:

com.ibm.retail.AEF.event.WorkstationStatusEvent#INPUT_SEQUENCE_CLEARED
INPUT_SEQUENCE_CLEARED

com.ibm.retail.AEF.event

Interface WorkstationStatusListener

All Superinterfaces:
POSAppEventListener

All Known Implementing Classes:
WorkstationStatusListenerProxy

---

public interface **WorkstationStatusListener**
extends POSAppEventListener

Listener interface for receiving `WorkstationStatusEvents`from the terminal POS session. A `WorkstationStatusEvent`is fired whenever the POS workstation enters a new "mode". For example, when a terminal is offline a workstation status event is fired to signal this change.

See Also:
com.ibm.retail.AEF.data.POSDataProvider#addWorkstationStatusListener addWorkstationStatusListener

---

Method Summary

`void`	`workstationStatusChanged(WorkstationStatusEvent evt)` A change occurred to the workstation status.

Methods

workstationStatusChanged

```
public void workstationStatusChanged(WorkstationStatusEvent evt)
                                     throws java.rmi.RemoteException
```

A change occurred to the workstation status.

Parameters:
`evt -`
contains workstation status details

Exceptions:
`RemoteException -`
if listener can not be notified

Package
com.ibm.retail.AEF.factory

Provides `factory`classes to create AEF objects.

com.ibm.retail.AEF.factory

Interface AEFSessionFactory

public interface **AEFSessionFactory**
extends java.rmi.Remote

An AEFSessionFactory provides access to AEFSession objects through an implementation of the Factory design pattern. The factory represents a collection of virtual terminals which can be accessed through the factory's getSession() methods.

The AEFSessionFactory implementation provides resource pooling to facilitate efficient use of the AEFSession instances. The factory is configured in *config.properties*, where values for the range of valid terminal numbers and the capacity of the factory are specified.

The AEFSessionFactory registers as a provider to one or more SessionServers which in turn provide client access to sessions while distributing the load among the factory instances.

Field Summary

static java.lang.String	`NAME` Name used for RMI naming

Method Summary

void	`destroySession(java.lang.String terminalNumber)` Destroys an AEFSession.
java.util.Collection	`getActiveTerminalNumbers()` Get an enumeration of the terminal numbers of active AEFSessions in this factory.
AEFSession	`getAvailableSession(SessionParameters parms)` Get an AEFSession object with a status of "available".
AEFSessionFactoryInfo	`getFactoryInfo()` Get information about this factory.
AEFSessionFactory	`getSelfReference()` Returns a self reference.
AEFSession	`getSession(java.lang.String terminalNumber,SessionParameters terminalNumber)` Get an AEFSession object by terminal number.
boolean	`sessionExists(java.lang.String terminalNumber)` Returns true if a session for this terminal number has been created.

Fields

(continued from last page)

NAME

```
public static final java.lang.String NAME
```

Name used for RMI naming

Methods

getAvailableSession

```
public AEFSession getAvailableSession(SessionParameters parms)
                                 throws java.rmi.RemoteException,
                                        AEFException
```

Get an AEFSession object with a status of "available". An AEFSession is in the available state if no other clients have references to it. This method establishes a client reference to the session and removes it from the pool of available sessions.

Parameters:

`parms` -
SessionParameters object contains all parameters for the session as name-value pairs

Returns:

an instance of an AEFSession object

Exceptions:

`RemoteException` -
if remote access fails
`AEFException` -
if a session can not be created

---

getSession

```
public AEFSession getSession(java.lang.String terminalNumber,
                             SessionParameters parms)
                       throws java.rmi.RemoteException,
                             AEFException
```

Get an AEFSession object by terminal number. If this is a valid terminal number for the factory, the session is returned regardless of the state. This allows a session to be shared among multiple clients. This method does not remove the session from the pool of available sessions. Clients can remove the session from the available session pool by invoking the *reference()* method of the particular AEFSession.

Parameters:

`terminalNumber` -
number of the terminal session requested
`parms` -
SessionParameters object contains all parameters for the session as name-value pairs

Returns:

an instance of an AEFSession object

Exceptions:

`RemoteException` -
if remote access fails
`AEFException` -
if a session can not be created

(continued from last page)

destroySession

```
public void destroySession(java.lang.String terminalNumber)
                    throws java.rmi.RemoteException,
                           AEFException
```

Destroys an AEFSession. This method applies only to virtual (CSS) sessions. Invoking this method terminates the POS application and removes the AEFSession from the SessionPool.

Parameters:

`terminalNumber -`
identifies the session to be destroyed

Exceptions:

`RemoteException -`
if server is not available
`AEFException -`
if an error occurs destroying the session AEFException error codes are:
AEFConst.API_VALID_FOR_TSS_ONLY
AEFConst.SESSION_NOT_ACTIVE

---

getFactoryInfo

```
public AEFSessionFactoryInfo getFactoryInfo()
                                     throws java.rmi.RemoteException
```

Get information about this factory. Information can be used to determine factory load (utilization) for load balancing across multiple factories.

Returns:

an instance of an AEFFactorySessionInfo object

Exceptions:

`RemoteException -`
if an instance of an object can not be created

---

sessionExists

```
public boolean sessionExists(java.lang.String terminalNumber)
                       throws java.rmi.RemoteException
```

Returns true if a session for this terminal number has been created.

Parameters:

`terminalNumber -`
identifies the session

Returns:

true if the session for this terminal number exists

Exceptions:

`RemoteException -`
if error occurs in remote access

---

getActiveTerminalNumbers

```
public java.util.Collection getActiveTerminalNumbers()
                                                 throws java.rmi.RemoteException
```

Get an enumeration of the terminal numbers of active AEFSessions in this factory.

Returns:

(continued from last page)

Collection of active terminal numbers (Strings) sorted in ascending order

Exceptions:

`RemoteException -`
if an instance of an object can not be created

getSelfReference

```
public AEFSessionFactory getSelfReference()
                                 throws java.rmi.RemoteException
```

Returns a self reference. This method is used to retrieve the actual factory reference if given a proxy reference of the type AEFSessionFactoryRef.

Returns:

AEFSessionFactory

Exceptions:

`RemoteException`

com.ibm.retail.AEF.factory

Class AEFSessionFactoryInfo

```
java.lang.Object
   |
   +-com.ibm.retail.AEF.factory.AEFSessionFactoryInfo
```

All Implemented interfaces:
java.io.Serializable

public class **AEFSessionFactoryInfo**
extends java.lang.Object
implements java.io.Serializable

AEFSessionFactoryInfo encapsulates information about the current status of the factory. It is primarily used to determine factory capability and utilization for the purposes of load balancing among multiple factories through a single SessionServer.

Field Summary

java.lang.String	beaconStr
int	currentPoolSize
java.lang.String	ID
java.lang.String	localTerminalNumber
java.util.TreeSet	reservedTerminalNumberList
java.lang.String	reservedTerminalNumbers
java.util.TreeSet	terminalNumberList
java.lang.String	terminalNumbers
int	totalAvailableSessions
boolean	tss
java.lang.String	URI

Constructor Summary

AEFSessionFactoryInfo()
AEFSessionFactoryInfo constructor

Method Summary

java.lang.String	`getBeaconString()` Returns a String suitable for a beacon buffer containing the state information of this info object.
int	`getCurrentPoolSize()` Get the total number of sessions currently in the session pool
java.lang.String	`getID()` Get the factory ID
java.lang.String	`getLocalTerminalNumber()` Get the terminal number for AEF running in real terminal.
java.util.Collection	`getReservedTerminalNumbers()` Get the terminal numbers that are considered "reserved" for this factory.
java.util.Collection	`getTerminalNumbers()` Get the terminal numbers that this factory assigns.
int	`getTotalAvailableSessions()` Get the total number of sessions that are available
int	`getTotalCapacity()` Get the total factory capacity
java.lang.String	`getURI()` Get the factory URI if factory is exposed via RMI.
void	`initWithBeaconData(java.lang.String beaconStr)` Initialize the member data from a byte array.
boolean	`isTSS()` Get the TSS (virtual terminals) flag
boolean	`isValidTerminalNumber(java.lang.String terminalNumber)` Determine if this factory can create a specific terminal
java.util.TreeSet	`parseTerminalNumbers(java.lang.String termNums)` Set the terminal numbers that this factory assigns Format is a comma separated list or range (e.
void	`setCurrentPoolSize(int t)` Set the total number of sessions currently in the session pool
void	`setID(java.lang.String s)` Set the factory ID
void	`setLocalTerminalNumber(java.lang.String t)` Set the local terminal number for this factory (non-tss)
java.lang.String	`setReservedTerminalNumbers(java.lang.String t)` Set the reserved terminal numbers for this factory.

void	`setTerminalNumbers(java.lang.String t)` Set the terminal numbers that this factory assigns.
void	`setTotalAvailableSessions(int t)` Set the total number of sessions that are available
void	`setTSS(boolean flag)` Set the TSS flag
void	`setURI(java.lang.String s)` Set the factory URI
java.lang.String	`toString()` Returns the internal representation of this object as a string.
void	`validateReservedTerminalNumbers()` Validate validate that the reserved terminal numbers are contained in the set of specified terminal numbers - if not, a warning is logged and the offending terminals are removed from the set of reserved terminal numbers.

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

currentPoolSize

`protected int` **`currentPoolSize`**

totalAvailableSessions

`protected int` **`totalAvailableSessions`**

tss

`protected boolean` **`tss`**

localTerminalNumber

`protected java.lang.String` **`localTerminalNumber`**

terminalNumbers

`protected java.lang.String` **`terminalNumbers`**

terminalNumberList

`protected java.util.TreeSet` **`terminalNumberList`**

reservedTerminalNumbers

`protected java.lang.String` **`reservedTerminalNumbers`**

reservedTerminalNumberList

`protected java.util.TreeSet` **`reservedTerminalNumberList`**

ID

`protected java.lang.String` **`ID`**

URI

`protected java.lang.String` **`URI`**

beaconStr

`protected java.lang.String` **`beaconStr`**

Constructors

AEFSessionFactoryInfo

`public` **`AEFSessionFactoryInfo`**`()`

AEFSessionFactoryInfo constructor

Methods

initWithBeaconData

`public void` **`initWithBeaconData`**`(java.lang.String beaconStr)`

Initialize the member data from a byte array. This method is intended to populate the info object with the data from the factory beacon. The beacon data is of the form f|factory_id|RMI_URI|current_pools_size|total_available_sessions|is_TSS(T or F)|terminal_numbers

Parameters:

`String`

getID

`public java.lang.String` **`getID`**`()`

Get the factory ID

(continued from last page)

Returns:

ID

getURI

`public java.lang.String getURI()`

Get the factory URI if factory is exposed via RMI. If not, this method returns "local".

Returns:

URI

getTotalCapacity

`public int getTotalCapacity()`

Get the total factory capacity

Returns:

total number of sessions this factory can create

isTSS

`public boolean isTSS()`

Get the TSS (virtual terminals) flag

Returns:

true if TSS factory

getCurrentPoolSize

`public int getCurrentPoolSize()`

Get the total number of sessions currently in the session pool

Returns:

total number of sessions in pool

getTerminalNumbers

`public java.util.Collection getTerminalNumbers()`

Get the terminal numbers that this factory assigns.

Returns:

Collection of terminal numbers as string values

getReservedTerminalNumbers

`public java.util.Collection getReservedTerminalNumbers()`

Get the terminal numbers that are considered "reserved" for this factory.

Returns:

(continued from last page)

Collection of reserved terminal numbers as string values

getLocalTerminalNumber

`public java.lang.String` **`getLocalTerminalNumber`**`()`

Get the terminal number for AEF running in real terminal.

Returns:

terminal number

getTotalAvailableSessions

`public int` **`getTotalAvailableSessions`**`()`

Get the total number of sessions that are available

Returns:

total available sessions

setID

`public void` **`setID`**`(java.lang.String s)`

Set the factory ID

Parameters:

`s` -
factory ID

setURI

`public void` **`setURI`**`(java.lang.String s)`

Set the factory URI

Parameters:

`s` -
factory URI

setTerminalNumbers

`public void` **`setTerminalNumbers`**`(java.lang.String t)`

Set the terminal numbers that this factory assigns. Format is a comma separated list or range (e.g., 10-20,22,25)

Parameters:

`t` -
terminal numbers as comma separated list

setReservedTerminalNumbers

`public java.lang.String` **`setReservedTerminalNumbers`**`(java.lang.String t)`

Set the reserved terminal numbers for this factory. Format is a comma separated list or range (e.g., 10-20,22,25). This list of terminal numbers is validated against the overall set of terminal numbers (set by #setTerminalNumbers(String). Any terminal numbers in the reserved list that are not in the overall list are removed. A String representation of the validated list (which may have been modified) is returned.

Parameters:

(continued from last page)

`t -`
terminal numbers as comma separated list

Returns:

the validated list of reserved terminal numbers.

parseTerminalNumbers

`protected java.util.TreeSet` **`parseTerminalNumbers`**`(java.lang.String termNums)`

Set the terminal numbers that this factory assigns Format is a comma separated list or range (e.g., 10-20,22,25)

Parameters:

`t -`
terminal numbers as comma separated list

validateReservedTerminalNumbers

`protected void` **`validateReservedTerminalNumbers`**`()`

Validate validate that the reserved terminal numbers are contained in the set of specified terminal numbers - if not, a warning is logged and the offending terminals are removed from the set of reserved terminal numbers. Reconstruct the String representation of the set of reserved terminal numbers.

setCurrentPoolSize

`public void` **`setCurrentPoolSize`**`(int t)`

Set the total number of sessions currently in the session pool

Parameters:

`t -`
total number of sessions in pool

setLocalTerminalNumber

`public void` **`setLocalTerminalNumber`**`(java.lang.String t)`

Set the local terminal number for this factory (non-tss)

Parameters:

`t -`
local terminal number

setTotalAvailableSessions

`public void` **`setTotalAvailableSessions`**`(int t)`

Set the total number of sessions that are available

Parameters:

`t -`
total available sessions

setTSS

`public void` **`setTSS`**`(boolean flag)`

Set the TSS flag

Parameters:

`flag -`
true if TSS (virtual terminals) factory

isValidTerminalNumber

`public boolean **isValidTerminalNumber**(java.lang.String terminalNumber)`

Determine if this factory can create a specific terminal

Parameters:

`terminalNumber -`
identifies a terminal

Returns:

true if this is the factory handling a specified terminal number

getBeaconString

`public java.lang.String **getBeaconString**()`

Returns a String suitable for a beacon buffer containing the state information of this info object.

Returns:

String

toString

`public java.lang.String **toString**()`

Returns the internal representation of this object as a string.

Returns:

String

com.ibm.retail.AEF.factory

Class AEFSessionPool

```
java.lang.Object
  +-java.util.Dictionary
      +-java.util.Hashtable
          +-com.ibm.retail.AEF.factory.AEFSessionPool
```

public class **AEFSessionPool**
extends java.util.Hashtable

A AEFSessionPool serves as a resource pool for AEFSession objects.

Field Summary

`java.util.Hashtable`	`pool`

Constructor Summary

`AEFSessionPool()` Construct AEFSessionPool

Method Summary

`void`	`addSession(AEFSession session)` Add a session to the pool
`boolean`	`containsSession(java.lang.String terminalNumber)` Does this pool contain a session.
`void`	`destroySession(java.lang.String terminalNumber)` Destroy an AEFSession and remove from the session pool.
`java.util.Collection`	`getActiveTerminalNumbers()` Get an enumeration of terminal numbers of active AEFSessions in this factory.
`AEFSession`	`getAvailableSession()` Get any available session from the pool
`int`	`getCurrentPoolSize()` Get the total number of sessions that are pooled.
`AEFSession`	`getSession(java.lang.String terminalNumber)` Get a session from the pool
`int`	`getTotalAvailableSessions()` Get the total number of sessions that are available

void	`releaseSession(AEFSession session)` Release a session.

Methods inherited from : class `java.util.Hashtable`
`clear, clone, contains, containsKey, containsValue, elements, entrySet, equals, get, hashCode, isEmpty, keys, keySet, put, putAll, rehash, remove, size, toString, values`

Methods inherited from : class `java.util.Dictionary`
`elements, get, isEmpty, keys, put, remove, size`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

pool

`protected java.util.Hashtable` **`pool`**

Constructors

AEFSessionPool

`public` **`AEFSessionPool`**`()`

Construct AEFSessionPool

Methods

addSession

`public void` **`addSession`**`(AEFSession session)`

Add a session to the pool

Parameters:

`session -`
AEFSession to be added

getSession

`public AEFSession` **`getSession`**`(java.lang.String terminalNumber)`

Get a session from the pool

Parameters:

`terminalNumber -`
terminal number of session

Returns:

(continued from last page)

AEFSession or null if no session in pool

getAvailableSession

```
public AEFSession getAvailableSession()
```

Get any available session from the pool

Returns:

AEFSession or null if no session in pool

releaseSession

```
public void releaseSession(AEFSession session)
```

Release a session.

Parameters:

`session -`
the session to be released

destroySession

```
public void destroySession(java.lang.String terminalNumber)
                    throws java.rmi.RemoteException,
                           AEFException
```

Destroy an AEFSession and remove from the session pool. This method applies only to virtual (CSS) sessions. It terminates the POS sales application and removes the AEFSession from the SessionPool.

Parameters:

`terminalNumber -`
terminal number to destroy

Exceptions:

`java.rmi.RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.SESSION_NOT_ACTIVE
AEFConst.API_VALID_FOR_TSS_ONLY

getTotalAvailableSessions

```
public int getTotalAvailableSessions()
```

Get the total number of sessions that are available

Returns:

total available sessions

getCurrentPoolSize

```
public int getCurrentPoolSize()
```

Get the total number of sessions that are pooled. This number includes all sessions.

Returns:

total sessions in pool

(continued from last page)

getActiveTerminalNumbers

```
public java.util.Collection getActiveTerminalNumbers()
                                                    throws java.rmi.RemoteException
```

Get an enumeration of terminal numbers of active AEFSessions in this factory.

Returns:

Collection of active terminal numbers sorted in ascending order (strings)

Exceptions:

`RemoteException` -
if an instance of an object cannot be created

containsSession

```
public boolean containsSession(java.lang.String terminalNumber)
```

Does this pool contain a session.

Parameters:

`terminalNumber` -
identifies the session

Returns:

Return true if the session pool contains a session with specified terminal number

Package
com.ibm.retail.AEF.mgmt

Provides JMX Mbean classes to manage various AEF objects.

com.ibm.retail.AEF.mgmt

Interface DebugMemoryMBean

public interface **DebugMemoryMBean**

Supports the capability to dump the SIMemoryFileHandler4690 memory log using the AEF DebugMemory object.

The purpose of this MBean is to give the user the ability to dump the in-memory log records that are generated by AEF. In this case, the user could be store personnel or a regional help desk using a management console or an intelligent management application that can react to events occuring within the system.

The `ObjectName`of this MBean includes the following attributes, in addition to the SIF attribute of `DeviceID`:
- SIFComponent=AEF
- Id=AEFSupport
- DMajorVer=x
- DMinorVer=x
- AEFType=DebugMemory

This management interface includes the following attributes. These attributes are described in more detail in the accessor methods.
- LogEnabled

The following operations are included in this management interface. These are described in more detail in the corresponding method documentation.
- dumpMemoryLog

No notifications are emitted by classes implementing this interface.

Method Summary

void	`dumpMemoryLog()` Sends a severe message to the log to dump the SIMemoryFileHandler4690 memory log.
boolean	`isLogEnabled()`
void	`setLogEnabled(boolean enable)` Enables logging of the methods.

Methods

dumpMemoryLog

`public void` **`dumpMemoryLog`**`()`

Sends a severe message to the log to dump the SIMemoryFileHandler4690 memory log.

setLogEnabled

`public void` **`setLogEnabled`**`(boolean enable)`

Enables logging of the methods. Logging is off by default.

isLogEnabled

`public boolean` **`isLogEnabled`**`()`

(continued from last page)

Returns:

true if logging is enabled; false otherwise. Logging is off by default.

com.ibm.retail.AEF.mgmt

Interface EventQueueMBean

public interface **EventQueueMBean**

Specifies the management interface for the AEFEventQueue object.

The `AEFEventQueue`is a dispatch queue for AEF `POSDataProvider`events. When property changes occur, an event is put on the queue representing that property change. This event is processed by notifying all registered listeners for that particular property.

Certain statistical information is kept by the `AEFEventQueue`such as maximum queue length, average/max wait time for elements on the queue, etc. The purpose of this MBean is to expose this information.

The `ObjectName`of this MBean includes the following attributes, in addition to the SIF attributes of `StoreID`and `DeviceID`:
- SIFComponent=AEF
- Id=EventQueue
- DMajorVer=x
- DMinorVer=x
- AEFType=EventQueue
- Environment=[virtual | real]

This management interface includes the following attributes. These attributes are described in more detail in the accessor methods. A marker is used internally by the `AEFEventQueue`to synchronize requests for data with pending requests that may cause changes in the data being requested.
- AllPerformanceStatistics
- TotalProcessed
- CurrentQueueLength
- MaximumQueueLength
- MaximumQueueWait
- MaximumProcessTime
- AverageQueueWait
- AverageProcessTime
- TotalMarkersProcessed
- AverageMarkerWait
- MaximumMarkerWait

The following operations are included in this management interface. These are described in more detail in the corresponding method documentation.
- resetStatistics
- logStatistics

No notifications are generated by any classes that implement this management interface.

Method Summary

`java.lang.Object[]`	`getAllPerformanceStatistics()` Retrieve the current performance statistics associated with this `EventQueue`.
`double`	`getAverageMarkerWait()` Retrieve the average amount of time that a marker has had to wait before being processed by this `EventQueue`, since the last reset.
`double`	`getAverageProcessTime()` Retrieve the average amount of time needed to process a request, since the last reset.
`double`	`getAverageQueueWait()` Retrieve the average amount of time that a request has had to wait before being processed by this `EventQueue`, since the last reset.

int	`getCurrentQueueLength()` Retrieve the current number of requests waiting to be processed by this `EventQueue`.
long	`getMaximumMarkerWait()` Retrieve the maximum amount of time that a marker has had to wait before being processed by this `EventQueue`, since the last reset.
long	`getMaximumProcessTime()` Retrieve the maximum amount of time needed to process a request, since the last reset.
int	`getMaximumQueueLength()` Retrieve the maximum number of requests that have ever waiting to be processed by this `EventQueue`, since the last reset.
long	`getMaximumQueueWait()` Retrieve the maximum amount of time that a request has had to wait before being processed by this `EventQueue`, since the last reset.
long	`getTotalMarkersProcessed()` Retrieve the total number of markers processed by this `EventQueue`, since the last reset.
long	`getTotalProcessed()` Retrieve the total number of requests processed by this `EventQueue`, since the last reset.
void	`logStatistics()` Record all of the performance statistics for this `EventQueue`to the AEF log.
void	`resetStatistics()` Reset all of the performance statistics for this `EventQueue`to zero.

Methods

getAllPerformanceStatistics

`public java.lang.Object[] getAllPerformanceStatistics()`

Retrieve the current performance statistics associated with this `EventQueue`. This data is returned as an array of objects as follows. The meaning of the various statistics are described in more detail in the getter method for the specific attribute. Object[0] = (Long) Total processedObject[1] = (Integer) Current queue lengthObject[2] = (Integer) Maximum queue lengthObject[3] = (Long) Maximum queue waitObject[4] = (Long) Maximum process timeObject[5] = (Double) Average queue waitObject[6] = (Double) Average process timeObject[7] = (Long) Total markersObject[8] = (Double) Average marker waitObject[9] = (Long) Maximum marker wait

Returns:

performance statistics as described above.

getTotalProcessed

`public long getTotalProcessed()`

Retrieve the total number of requests processed by this `EventQueue`, since the last reset. This number includes markers.

Returns:

total number of requests processed since the last reset.

getCurrentQueueLength

`public int getCurrentQueueLength()`

Retrieve the current number of requests waiting to be processed by this `EventQueue`. This number includes markers.

Returns:

number of requests waiting to be processed.

getMaximumQueueLength

`public int getMaximumQueueLength()`

Retrieve the maximum number of requests that have ever waiting to be processed by this `EventQueue`, since the last reset. This number includes markers.

Returns:

maximum number of requests that have ever been waiting to be processed.

getMaximumQueueWait

`public long getMaximumQueueWait()`

Retrieve the maximum amount of time that a request has had to wait before being processed by this `EventQueue`, since the last reset.

Returns:

maximum request wait time in milliseconds.

getMaximumProcessTime

`public long getMaximumProcessTime()`

Retrieve the maximum amount of time needed to process a request, since the last reset.

Returns:

maximum amount of time needed to process a request in milliseconds.

getAverageQueueWait

`public double getAverageQueueWait()`

Retrieve the average amount of time that a request has had to wait before being processed by this `EventQueue`, since the last reset.

Returns:

average request wait time in milliseconds.

getAverageProcessTime

`public double getAverageProcessTime()`

Retrieve the average amount of time needed to process a request, since the last reset.

Returns:

average amount of time needed to process a request in milliseconds.

getTotalMarkersProcessed

`public long` **`getTotalMarkersProcessed`**`()`

Retrieve the total number of markers processed by this `EventQueue`, since the last reset.

Returns:

total number of markers processed since the last reset.

getAverageMarkerWait

`public double` **`getAverageMarkerWait`**`()`

Retrieve the average amount of time that a marker has had to wait before being processed by this `EventQueue`, since the last reset.

Returns:

average marker wait time in milliseconds.

getMaximumMarkerWait

`public long` **`getMaximumMarkerWait`**`()`

Retrieve the maximum amount of time that a marker has had to wait before being processed by this `EventQueue`, since the last reset.

Returns:

maximum marker wait time in milliseconds.

resetStatistics

`public void` **`resetStatistics`**`()`

Reset all of the performance statistics for this `EventQueue`to zero.

logStatistics

`public void` **`logStatistics`**`()`

Record all of the performance statistics for this `EventQueue`to the AEF log.

com.ibm.retail.AEF.mgmt

Interface SessionControlMBean

public interface **SessionControlMBean**

Specifies a "control" management interface for AEFSession objects.

The purpose of this MBean is to allow more detailed control of an `AEFSession`than what is available through the base SessionMBean management interface. MBeans of this type are created on demand through SessionMBean#enableDetailedControl(). When not needed anymore, these MBeans should be destroyed via SessionMBean#destroyDetailedControl().

The `ObjectName`of this MBean includes the following attributes, in addition to the SIF attributes of `StoreID`and `DeviceID`:
- SIFComponent=AEF
- Id=[session#]-Control
- DMajorVer=x
- DMinorVer=x
- AEFType=SessionControl
- SessionID=x
- Environment=[virtual | real]

This management interface does not include any attributes.

The following operations are included in this management interface. These are described in more detail in the corresponding method documentation.
- initializeSession
- sendKeySequence
- setKeyLockPosition

No notifications are generated by classes implementing this interface.

Method Summary

void	`initializeSession(int initialState)` Initialize the `AEFSession`.
void	`sendKeySequence(java.lang.String keySequence)` Send a key sequence to the POS application associated with this session.
void	`setKeyLockPosition(int lock)` Set the keylock position of the terminal associated with this session.

Methods

initializeSession

```
public void initializeSession(int initialState)
                        throws AEFException
```

Initialize the `AEFSession`. Refer to POSAutomationProvider.initialize() for information on valid initial states, results of this operation, and pertinent error codes that can be thrown with the `AEFException`. Note: this method applies only to virtual (CSS) sessions.

Exceptions:

`AEFException`

See Also:

(continued from last page)

com.ibm.retail.AEF.automation.POSAutomationProvider#initialize(int)

sendKeySequence

```
public void sendKeySequence(java.lang.String keySequence)
                      throws AEFException
```

Send a key sequence to the POS application associated with this session. Refer to Workstation.sendKeySequence() for information on the format of a valid key sequence as well as pertinent error codes that can be thrown with the `AEFException`.

Exceptions:

`AEFException`

See Also:

com.ibm.retail.AEF.workstation.Workstation#sendKeySequence(String)

setKeyLockPosition

```
public void setKeyLockPosition(int lock)
                         throws AEFException
```

Set the keylock position of the terminal associated with this session. Refer to Workstation.setKeyLockPosition() for information on valid keylock positions as well as pertinent error codes that can be thrown with the `AEFException`.

Exceptions:

`AEFException`

See Also:

com.ibm.retail.AEF.workstation.Workstation#setKeyLockPosition(int)

com.ibm.retail.AEF.mgmt

Interface SessionDebugMBean

public interface **SessionDebugMBean**

Specifies a "debug" management interface for AEFSession objects.

The purpose of this MBean is to allow more detailed monitoring of an `AEFSession`than what is available through the base SessionMBean management interface. Due to the potential for excessive notifications because of frequent state changes, MBeans of this type are created on demand through SessionMBean#enableDebug(). When the detailed monitoring is no longer needed, these MBeans should be destroyed via SessionMBean#destroyDebug().

The `ObjectName`of this MBean includes the following attributes, in addition to the SIF attribute of `DeviceID`:
- SIFComponent=AEF
- Id=[session#]-Debug
- DMajorVer=x
- DMinorVer=x
- AEFType=SessionDebug
- SessionID=x
- Environment=[virtual | real]

This management interface includes the following attributes. These attributes are described in more detail in the accessor methods.
- TransactionStatus
- OperatorDisplay
- CustomerDisplay
- PrintLines
- POSApplicationState
- POSApplicationSubstate
- TraceLevel

The following operations are included in this management interface. These are described in more detail in the corresponding method documentation.
- dumpSessionTrace

An `RtlDebugNotification`is emitted by classes implementing this interface when the following conditions occur.

If present, the constants referenced in the following notification descriptions indicate the possible values of the notification `userData`; else, the `userData`contains the new state data (since all of the notifications represent state changes). Unless otherwise noted, the userData will be of type `String`. The notification names in the list below indicate the constants that are used for the notification message text.
- TRANSACTION_STATUS_UPDATE - indicates that a transaction status change has occured (i.e., a `SALES_TRANSACTION_IN_PROGRESS`, a `NONSALES_TRANSACTION_IN_PROGRESS`or a `TRANSACTION_NOT_IN_PROGRESS`).
- OPERATOR_DISPLAY_UPDATE - indicates that the contents of OperatorDisplay for this session has changed. The userData in this notification contains the new contents.
- CUSTOMER_DISPLAY_UPDATE - indicates that the contents of CustomerDisplay for this session has changed. The userData in this notification contains the new contents.
- PRINT_LINES_UPDATE - indicates that data was sent to the printer associated with this session. The userData in this notification contains the new contents, which is a `Collection`of `String`s.
- POS_APPLICATION_STATE_UPDATE - indicates that the state of the POS application for this session has changed. Valid values for the state are defined in the AEFBundle that has "state.properties" as its root. The userData in this notification contains the new state.
- POS_APPLICATION_SUBSTATE_UPDATE - indicates that the substate of the POS application for this session has changed. Valid values for the substate are defined in the AEFBundle that has "substate.properties" as its root. The userData in this notification contains the new substate.

Field Summary

static java.lang.String	`CUSTOMER_DISPLAY_UPDATE`

static java.lang.String	NONSALES_TRANSACTION_IN_PROGRESS
static java.lang.String	NOTIFICATION_TYPE
static java.lang.String	OPERATOR_DISPLAY_UPDATE
static java.lang.String	POS_APPLICATION_STATE_UPDATE
static java.lang.String	POS_APPLICATION_SUBSTATE_UPDATE
static java.lang.String	PRINT_LINES_UPDATE
static java.lang.String	SALES_TRANSACTION_IN_PROGRESS
static java.lang.String	TRANSACTION_NOT_IN_PROGRESS
static java.lang.String	TRANSACTION_STATUS_UPDATE

Method Summary

void	dumpSessionTrace() Dumps the AEF session trace to a file.
java.lang.String	getCustomerDisplay() Returns the current value of the CustomerDisplay attribute.
java.lang.String	getOperatorDisplay() Returns the current value of the OperatorDisplay attribute.
java.lang.String	getPOSApplicationState() Returns the current value of the POSApplicationState attribute.
java.lang.String	getPOSApplicationSubstate() Returns the current value of the POSApplicationSubstate attribute.
java.util.Collection	getPrintLines() Returns the current value of the PrintLines attribute.
int	getTraceLevel() Returns the current AEF session trace level.
java.lang.String	getTransactionStatus() Returns the current TransactionStatus attribute value.
void	setTraceLevel(int level) Specifies the AEF session trace level.

(continued from last page)

Fields

NOTIFICATION_TYPE

public static final java.lang.String **NOTIFICATION_TYPE**

TRANSACTION_STATUS_UPDATE

public static final java.lang.String **TRANSACTION_STATUS_UPDATE**

CUSTOMER_DISPLAY_UPDATE

public static final java.lang.String **CUSTOMER_DISPLAY_UPDATE**

OPERATOR_DISPLAY_UPDATE

public static final java.lang.String **OPERATOR_DISPLAY_UPDATE**

PRINT_LINES_UPDATE

public static final java.lang.String **PRINT_LINES_UPDATE**

POS_APPLICATION_STATE_UPDATE

public static final java.lang.String **POS_APPLICATION_STATE_UPDATE**

POS_APPLICATION_SUBSTATE_UPDATE

public static final java.lang.String **POS_APPLICATION_SUBSTATE_UPDATE**

SALES_TRANSACTION_IN_PROGRESS

public static final java.lang.String **SALES_TRANSACTION_IN_PROGRESS**

NONSALES_TRANSACTION_IN_PROGRESS

public static final java.lang.String **NONSALES_TRANSACTION_IN_PROGRESS**

TRANSACTION_NOT_IN_PROGRESS

public static final java.lang.String **TRANSACTION_NOT_IN_PROGRESS**

(continued from last page)

Methods

getTransactionStatus

`public java.lang.String` **`getTransactionStatus`**`()`

Returns the current TransactionStatus attribute value. Valid return values are `SALES_TRANSACTION_IN_PROGRESS`, `NONSALES_TRANSACTION_IN_PROGRESS`, or `TRANSACTION_NOT_IN_PROGRESS`.

Returns:

the current contents of the LineDisplay1 device

getOperatorDisplay

`public java.lang.String` **`getOperatorDisplay`**`()`

Returns the current value of the OperatorDisplay attribute. This may take the form of two or more "lines" of characters, each separated by a newline character.

Returns:

the current contents of the Operator Display device

getCustomerDisplay

`public java.lang.String` **`getCustomerDisplay`**`()`

Returns the current value of the CustomerDisplay attribute. This may take the form of two or more "lines" of characters, each separated by a newline character.

Returns:

the current contents of the Operator Display device

getPrintLines

`public java.util.Collection` **`getPrintLines`**`()`

Returns the current value of the PrintLines attribute. This `Collection`is a composed of `String`s that represent output to the printer.

Returns:

the collection of print lines that were last sent to the printer

getPOSApplicationState

`public java.lang.String` **`getPOSApplicationState`**`()`

Returns the current value of the POSApplicationState attribute. Valid values for the state are defined in the AEFBundle that has "state.properties" as its root.

Returns:

the current state of the POS application

getPOSApplicationSubstate

`public java.lang.String` **`getPOSApplicationSubstate`**`()`

Returns the current value of the POSApplicationSubstate attribute. Valid values for the substate are defined in the AEFBundle that has "substate.properties" as its root.

(continued from last page)

Returns:

the current substate of the POS application

setTraceLevel

`public void **setTraceLevel**(int level)`

Specifies the AEF session trace level. Refer to `AEFSession.setSessionTraceLevel()` for information on valid parameter values.

See Also:

com.ibm.retail.AEF.session.AEFSession#setSessionTraceLevel(int)

getTraceLevel

`public int **getTraceLevel**()`

Returns the current AEF session trace level. Refer to `AEFSession.getSessionTraceLevel()` for information on valid return values.

Returns:

the current TraceLevel attribute value

See Also:

com.ibm.retail.AEF.session.AEFSession#getSessionTraceLevel()

dumpSessionTrace

`public void **dumpSessionTrace**()`

Dumps the AEF session trace to a file. Refer to `AEFSession.dumpSessionTrace()` for more information.

See Also:

com.ibm.retail.AEF.session.AEFSession#dumpSessionTrace()

com.ibm.retail.AEF.mgmt

Interface SessionFactoryMBean

public interface **SessionFactoryMBean**

Specifies the management interface for the AEFSessionFactory object.

There is at most one `AEFSessionFactory`object per AEF JVM. `AEFSessionFactory`s are responsible for supporting AEF client requests for `AEFSession`s. `AEFSessionFactory`s maintain counts of the number of sessions active, the maximum number of sessions the factory can support, the set of session IDs for this factory, etc.

The purpose of this MBean is to expose information related to `AEFSessionFactory`objects. In addition, notifications are emitted when sessions are created or destroyed.

The `ObjectName`of this MBean includes the following attributes, in addition to the SIF attributes of `StoreID`and `DeviceID`:
- SIFComponent=AEF
- Id=SessionFactory
- DMajorVer=x
- DMinorVer=x
- AEFType=SessionFactory
- SessionFactoryID=x
- Environment=[virtual | real]

This management interface includes the following attributes. These attributes are described in more detail in the accessor methods.
- FactoryID
- TotalCapacity
- CurrentPoolSize
- TotalAvailable
- TotalReserved
- SessionIDs
- ActiveSessionIDs
- ReservedSessionIDs

No operations are included in this management interface.

An `RtlDebugNotification`is emitted by classes implementing this interface when the following conditions occur. The notification type is `SESSION_FACTORY_TYPE`. The notification names in the list below indicate the constants that are used for the notification message text.
- SESSION_CREATED - indicates that an `AEFSession`has been created. This occurs 1) during AEF initialization if sessions are to be pre-created and 2) when a client requests an available session and there are no sessions available and the factory capacity has not been reached. The notification userData contains the ID (of type `String`) of the session that was created.
- SESSION_DESTROYED - indicates that an `AEFSession`has been destroyed. This occurs when the `SessionServer.destroySession()`method is invoked. The notification userData contains the ID (of type `String`) of the session that was destroyed.

Field Summary

static java.lang.String	`SESSION_CREATED` Used as the notification message text
static java.lang.String	`SESSION_DESTROYED` Used as the notification message text
static java.lang.String	`SESSION_FACTORY_TYPE` Used as notification type for notifications generated by classes implementing this interface

Method Summary

java.util.Collection	getActiveSessionIDs()
int	getCurrentPoolSize()
java.lang.String	getFactoryID()
java.util.Collection	getReservedSessionIDs() Return the collection of session IDs that are currently reserved by this particular factory.
java.util.Collection	getSessionIDs()
int	getTotalAvailable()
int	getTotalCapacity()
int	getTotalReserved()

Fields

SESSION_FACTORY_TYPE

`public static final java.lang.String` **`SESSION_FACTORY_TYPE`**

Used as notification type for notifications generated by classes implementing this interface

SESSION_CREATED

`public static final java.lang.String` **`SESSION_CREATED`**

Used as the notification message text

SESSION_DESTROYED

`public static final java.lang.String` **`SESSION_DESTROYED`**

Used as the notification message text

Methods

getFactoryID

`public java.lang.String` **`getFactoryID`**`()`

Returns:

the ID of this `SessionFactory`, as specified in the AEF configuration file. This ID is used to derive the RMI name of the `SessionFactory`if it is able to be accessed remotely.

getTotalCapacity

`public int` **`getTotalCapacity`**`()`

(continued from last page)

Returns:

the total capacity of this `SessionFactory`. If this factory is for a real session, the total capacity will always be 1. For virtual sessions, the capacity of the corresponding factory is based on the CSS configuration settings.

getCurrentPoolSize

`public int` **`getCurrentPoolSize`**`()`

Returns:

the number of `AEFSession`s that currently exist for this factory, where 0 <= CurrentPoolSize <= TotalCapacity.

getTotalAvailable

`public int` **`getTotalAvailable`**`()`

Returns:

the total number of existing `AEFSession`s that are available to be allocated by this factory. This does not include sessions that would be created on demand, and therefore CurrentPoolSize + TotalAvailable <= TotalCapacity.

getTotalReserved

`public int` **`getTotalReserved`**`()`

Returns:

the total number of reserved `AEFSession`s. Refer to #getReservedSessionIDs() for a description of a *reserved* session.

getSessionIDs

`public java.util.Collection` **`getSessionIDs`**`()`

Returns:

the collection of session IDs that this particular factory is responsible for. This includes sessions that exist as well as those that are yet to be created. This set of session IDs should be disjoint for every `SessionFactory` in the system; i.e., there should never be more than one factory that is able to create a given `AEFSession`.

getActiveSessionIDs

`public java.util.Collection` **`getActiveSessionIDs`**`()`

Returns:

the collection of session IDs that are currently active for this particular factory.

getReservedSessionIDs

`public java.util.Collection` **`getReservedSessionIDs`**`()`

(continued from last page)

Return the collection of session IDs that are currently reserved by this particular factory. Reserved sessions are only available to AEF clients that obtain specific sessions via SessionServer.getSession(). Reserved sessions are not considered "available" and therefore are not included in the count of available sessions. Reserved sessions may or may not be active and therefore may or may not be included in the list of active sessions returned by {@link #getActiveSessionIDs()}. Reserved session IDs are always included in the list of session IDs returned by {@link #getSessionIDs()}.

Returns:

the collection of reserved session IDs

com.ibm.retail.AEF.mgmt

Interface SessionMBean

All Superinterfaces:
MgmtExtendedControlMBean

---

public interface **SessionMBean**
extends MgmtExtendedControlMBean

Specifies the management interface for the AEFSession object.

An `AEFSession`represents a POS terminal (virtual or real) that can be used to access POS terminal application functionality without regard to the particular type of application that is providing that functionality. The purpose of this MBean is to expose certain information about the `AEFSession`and to emit notifications when various status changes occur.

This interface extends the `com.ibm.retail.si.mgmt.MgmtExtendedControlMBean`interface which allows "helper" MBeans to be dynamically created to provide extended control functions (e.g., debug and logging). See SessionDebugMBean and SessionControlMBean for more information.

The `ObjectName`of this MBean includes the following attributes, in addition to the SIF attributes of `StoreID`and `DeviceID`:
- SIFComponent=AEF
- Id=session#
- DMajorVer=x
- DMinorVer=y
- AEFType=Session
- SessionID=session#
- Environment=[virtual | real]

This management interface includes the following attributes. These attributes are described in more detail in the accessor methods.
- SessionID
- OnlineStatus
- OnlineStatus
- OnlineStatus
- POSApplicationStatus
- OperatorStatus

No operations are included in this management interface other than those specified by the MgmtExtendedControlMbean interface.

An `RtlDebugNotification`is emitted by classes implementing this interface when the following conditions occur. The type of the notification is `SESSION_TYPE`. The constants referenced in the notification description indicate the possible values of the notification `userData`. Unless otherwise noted, the userData will be of type `String`. The notification names in the list below indicate the constants that are used for the notification message text.
- ONLINE_STATUS_UPDATE - indicates that the online status of the POS terminal associated with this AEFSession has changed to `ONLINE`or `OFFLINE`.
- OPERATOR_STATUS_UPDATE - indicates that the state of the operator has changed to either `OPERATOR_LOGGED_ON`, `OPERATOR_LOGGED_OFF`, or `TERMINAL_SECURED`.

Field Summary

static java.lang.String	`ONLINE_STATUS_UPDATE` Used as notification message text
static java.lang.String	`OPERATOR_LOGGED_OFF` Used as a return value for `getOperatorStatus`to indicate that the operator is logged off

static java.lang.String	OPERATOR_LOGGED_ON Used as a return value for getOperatorStatusto indicate that the operator is logged on
static java.lang.String	OPERATOR_STATUS_UPDATE Used as notification message text
static java.lang.String	POS_APPLICATION_ACTIVE Used as a return value for getPOSApplicationStatusto indicate that the application is active
static java.lang.String	POS_APPLICATION_INACTIVE Used as a return value for getPOSApplicationStatusto indicate that the application is inactive
static java.lang.String	SESSION_TYPE Used as notification type for notifications generated by classes implementing this interface
static java.lang.String	TERMINAL_OFFLINE Used as a return value for getOnlineStatusto indicate that the terminal is offline
static java.lang.String	TERMINAL_ONLINE Used as a return value for getOnlineStatusto indicate that the terminal is online
static java.lang.String	TERMINAL_SECURED Used as a return value for getOperatorStatusto indicate that the terminal is in "secured" mode (i.

Method Summary

int	destroyDebug()
int	destroyDetailedControl()
int	destroyDetailedLogControl() Not supported.
javax.management.ObjectName	enableDebug()
javax.management.ObjectName	enableDetailedControl()
javax.management.ObjectName	enableDetailedLogControl() Not supported.
int	getCurrentlyActiveCapabilities()
int	getExtendedCapabilities()
java.lang.String	getOnlineStatus()
java.lang.String	getOperatorStatus()

java.lang.String	getPOSApplicationStatus()
java.lang.String	getSessionID()
boolean	isSessionAvailable()
boolean	isSessionReady()
boolean	isSessionReserved()

Fields

SESSION_TYPE

`public static final java.lang.String` **`SESSION_TYPE`**

Used as notification type for notifications generated by classes implementing this interface

ONLINE_STATUS_UPDATE

`public static final java.lang.String` **`ONLINE_STATUS_UPDATE`**

Used as notification message text

OPERATOR_STATUS_UPDATE

`public static final java.lang.String` **`OPERATOR_STATUS_UPDATE`**

Used as notification message text

TERMINAL_ONLINE

`public static final java.lang.String` **`TERMINAL_ONLINE`**

Used as a return value for `getOnlineStatus`to indicate that the terminal is online

TERMINAL_OFFLINE

`public static final java.lang.String` **`TERMINAL_OFFLINE`**

Used as a return value for `getOnlineStatus`to indicate that the terminal is offline

POS_APPLICATION_ACTIVE

`public static final java.lang.String` **`POS_APPLICATION_ACTIVE`**

Used as a return value for `getPOSApplicationStatus`to indicate that the application is active

POS_APPLICATION_INACTIVE

`public static final java.lang.String` **`POS_APPLICATION_INACTIVE`**

Used as a return value for `getPOSApplicationStatus`to indicate that the application is inactive

(continued from last page)

OPERATOR_LOGGED_ON

`public static final java.lang.String` **`OPERATOR_LOGGED_ON`**

Used as a return value for `getOperatorStatus`to indicate that the operator is logged on

OPERATOR_LOGGED_OFF

`public static final java.lang.String` **`OPERATOR_LOGGED_OFF`**

Used as a return value for `getOperatorStatus`to indicate that the operator is logged off

TERMINAL_SECURED

`public static final java.lang.String` **`TERMINAL_SECURED`**

Used as a return value for `getOperatorStatus`to indicate that the terminal is in "secured" mode (i.e., password needed to unlock)

Methods

destroyDebug

`public int` **`destroyDebug`**`()`

destroyDetailedControl

`public int` **`destroyDetailedControl`**`()`

destroyDetailedLogControl

`public int` **`destroyDetailedLogControl`**`()`

Not supported. Throws `UnsupportedOperationException`.

See Also:

com.ibm.retail.si.mgmt.MgmtExtendedControlMBean#destroyDetailedLogControl()

enableDebug

`public javax.management.ObjectName` **`enableDebug`**`()`

enableDetailedControl

`public javax.management.ObjectName` **`enableDetailedControl`**`()`

enableDetailedLogControl

`public javax.management.ObjectName` **`enableDetailedLogControl`**`()`

Not supported. Throws `UnsupportedOperationException`.

See Also:

(continued from last page)

com.ibm.retail.si.mgmt.MgmtExtendedControlMBean#enableDetailedLogControl()

getCurrentlyActiveCapabilities

`public int **getCurrentlyActiveCapabilities**()`

getExtendedCapabilities

`public int **getExtendedCapabilities**()`

Returns:

com.ibm.retail.si.mgmt.MgmtExtendedControlMBean.EXTENDED_CTRL || com.ibm.retail.si.mgmt.MgmtExtendedControlMBean.EXTENDED_DEBUG

See Also:

com.ibm.retail.si.mgmt.MgmtExtendedControlMBean#getExtendedCapabilities()

getSessionID

`public java.lang.String **getSessionID**()`

Returns:

the identifier of this `AEFSession`. This ID is typically a String of three digits.

isSessionReady

`public boolean **isSessionReady**()`

Returns:

an indicator of whether the session is ready to be used. The session is not considered ready until it has completed initialization and is ready to accept input.

isSessionAvailable

`public boolean **isSessionAvailable**()`

Returns:

an indicator of whether the session is available (not in use).

isSessionReserved

`public boolean **isSessionReserved**()`

Returns:

(continued from last page)

an indicator of whether the session is reserved. A session that is not reserved can be obtained via SessionServer.getAvailableSession(); a reserved session can only be obtained via SessionServer.getSession().

getOnlineStatus

`public java.lang.String` **`getOnlineStatus`**`()`

Returns:

the online status of the terminal associated with this `AEFSession`. The terminal is considered "online" if it is able to communicate with its local controller. Valid values include `TERMINAL_ONLINE`and `TERMINAL_OFFLINE`.

Note: Virtual sessions are always considered to be online. Only real (non-virtual) sessions can ever be offline.

getPOSApplicationStatus

`public java.lang.String` **`getPOSApplicationStatus`**`()`

Returns:

the status of the POS application associated with this `AEFSession`. The application is considered "active" if it is able to accept and process requests. Valid values include `POS_APPLICATION_ACTIVE`and `POS_APPLICATION_INACTIVE`.

Note: For real (non-virtual) sessions the application is always considered to be active.

getOperatorStatus

`public java.lang.String` **`getOperatorStatus`**`()`

Returns:

the status of the operator associated with this `AEFSession`. Valid values include `OPERATOR_LOGGED_ON`, `OPERATOR_LOGGED_OFF`, and `SECURED_MODE`.

com.ibm.retail.AEF.mgmt

Interface SessionServerControlMBean

public interface **SessionServerControlMBean**

Specifies a "control" management interface for the AEF SessionServer object.

The purpose of this MBean is to allow more detailed control of the AEF `SessionServer`than what is available through the base SessionServerMBean management interface. MBeans of this type are created on demand through SessionServerMBean#enableDetailedControl(). When not needed anymore, these MBeans should be destroyed via SessionServerMBean#destroyDetailedControl().

The `ObjectName`of this MBean includes the following attributes, in addition to the SIF attributes of `StoreID`and `DeviceID`:
- SIFComponent=AEF
- Id=SessionServerControl
- DMajorVer=x
- DMinorVer=x
- AEFType=SessionServerControl
- SessionServerID=x
- Environment=[virtual | real]
  This management interface does not include any attributes.

The following operations are included in this management interface:
- destroySession
  No notifications are generated by classes implementing this interface.

Method Summary

`void`	`destroySession(java.lang.String sessionID)` Destroy the session corresponding to the specified session ID.

Methods

destroySession

```
public void destroySession(java.lang.String sessionID)
                    throws AEFException
```

Destroy the session corresponding to the specified session ID. This action stops the POS application associated with the session and releases any resources associated with it. This action is valid only for virtual sessions. Refer to SessionServer.destroySession() for information on specific error codes that can be returned with the `AEFException`. Note: this method can only be invoked for virtual (CSS) sessions.

Exceptions:

`AEFException`

See Also:

com.ibm.retail.AEF.server.SessionServer#destroySession(String)

com.ibm.retail.AEF.mgmt

Interface SessionServerMBean

All Superinterfaces:
MgmtExtendedControlMBean

---

public interface **SessionServerMBean**
extends MgmtExtendedControlMBean

Specifies the management interface for the AEF SessionServer object.

The AEF `SessionServer`controls access to AEFSessions, which represent POS terminals (either real or virtual). A `SessionServer`can allocate sessions from multiple AEFSessionFactorys, each of which are active on a different store controller. The purpose of this MBean is to expose information about the various `AEFSessionFactory`s that the `SessionServer`is aware of. In addition, classes implementing this interface may emit notifications when factories are discovered or removed, or if an `AEFSession`is unable to be allocated due to insufficient capacity.

This interface extends the `com.ibm.retail.si.mgmt.MgmtExtendedControlMBean`interface which allows "helper" MBeans to be dynamically created to provide extended control functions (e.g., debug and logging).

The `ObjectName`of this MBean includes the following attributes, in addition to the SIF attributes of `StoreID`and `DeviceID`:
- SIFComponent=AEF
- Id=SessionServer
- DMajorVer=x
- DMinorVer=x
- AEFType=SessionServer
- SessionServerID=x
- Environment=[virtual | real]

This management interface includes the following attributes. These attributes are described in more detail in the accessor methods.
- ServerID
- ServerState
- SessionFactoryIDs

No operations are included in this management interface other than those specified by the MgmtExtendedControlMbean interface.

Notifications are generated by classes implementing this interface when the following conditions occur. The type of the notification is `SESSION_SERVER_TYPE`. The constants referenced in the notification description indicate the possible values of the notification `userData`. Unless otherwise noted, the userData will be of type `String`. There are additional constants defined in this interface that are used for the notification message. The `Notification`class is a subclass of `com.ibm.retail.si.mgmt.notifications.RtlNotification`and is identified in the description of each notification. The notification names in the list below indicate the constants that are used for the notification message text.
- SERVER_STATE_CHANGE - indicates that the state of the server has changed to either `SERVER_INITIALIZING`or `SERVER_ACTIVE`. The notification class is `RtlInformationNotification`.
- SESSION_FACTORY_ADDED - indicates that a new SessionFactory has been detected. The ID of the SessionFactory is the userData of the notification. The notification class is `RtlInformationNotification`.
- SESSION_FACTORY_REMOVED - indicates that a SessionFactory has been removed from the set of SessionFactory objects that this SessionServer will use to satisfy `getAvailableSession()`requests. This can occur when a known SessionFactory does not respond to requests or a RemoteException occurs attempting to access it. The ID of the SessionFactory is the userData of the notification. The notification class is `RtlInformationNotification`.
- SESSION_FACTORY_UPDATED - indicates that information about a SessionFactory has been updated. This information could include the set of sessions that are currently available for the factory as well as the current pool size.
- NO_SESSION_AVAILABLE - indicates that a request for an available session was not able to be satisfied. This can occur when all known `SessionFactory`objects are at their maximum capacity. This notification does not include a userData object. The notification class is `RtlWarningNotification`.

Field Summary

static java.lang.String	NO_SESSION_AVAILABLE Used as notification message text
static java.lang.String	SERVER_ACTIVE Used as a return value for getServerStateto indicate that the server is active
static java.lang.String	SERVER_INITIALIZING Used as a return value for getServerStateto indicate that the server is initializing
static java.lang.String	SERVER_STATE_CHANGE Used as notification message text
static java.lang.String	SESSION_FACTORY_ADDED Used as notification message text
static java.lang.String	SESSION_FACTORY_REMOVED Used as notification message text
static java.lang.String	SESSION_FACTORY_UPDATED Used as notification message text
static java.lang.String	SESSION_SERVER_TYPE Used as notification type for notifications generated by classes implementing this interface

Method Summary

int	destroyDebug() Not supported.
int	destroyDetailedControl()
int	destroyDetailedLogControl() Not supported.
javax.management.ObjectName	enableDebug() Not supported.
javax.management.ObjectName	enableDetailedControl()
javax.management.ObjectName	enableDetailedLogControl() Not supported.
int	getCurrentlyActiveCapabilities()
int	getExtendedCapabilities()
java.lang.String	getServerID()
java.lang.String	getServerState()

java.util.List	getSessionFactoryIDs()

Fields

SESSION_SERVER_TYPE

`public static final java.lang.String` **`SESSION_SERVER_TYPE`**

Used as notification type for notifications generated by classes implementing this interface

SERVER_STATE_CHANGE

`public static final java.lang.String` **`SERVER_STATE_CHANGE`**

Used as notification message text

SESSION_FACTORY_ADDED

`public static final java.lang.String` **`SESSION_FACTORY_ADDED`**

Used as notification message text

SESSION_FACTORY_REMOVED

`public static final java.lang.String` **`SESSION_FACTORY_REMOVED`**

Used as notification message text

SESSION_FACTORY_UPDATED

`public static final java.lang.String` **`SESSION_FACTORY_UPDATED`**

Used as notification message text

NO_SESSION_AVAILABLE

`public static final java.lang.String` **`NO_SESSION_AVAILABLE`**

Used as notification message text

SERVER_ACTIVE

`public static final java.lang.String` **`SERVER_ACTIVE`**

Used as a return value for `getServerState`to indicate that the server is active

SERVER_INITIALIZING

`public static final java.lang.String` **`SERVER_INITIALIZING`**

Used as a return value for `getServerState`to indicate that the server is initializing

Methods

destroyDebug

`public int` **`destroyDebug`**`()`

Not supported. Throws `UnsupportedOperationException`.

(continued from last page)

See Also:

com.ibm.retail.si.mgmt.MgmtExtendedControlMBean#destroyDebug()

destroyDetailedControl

`public int destroyDetailedControl()`

destroyDetailedLogControl

`public int destroyDetailedLogControl()`

Not supported. Throws `UnsupportedOperationException`.

See Also:

com.ibm.retail.si.mgmt.MgmtExtendedControlMBean#destroyDetailedLogControl()

enableDebug

`public javax.management.ObjectName enableDebug()`

Not supported. Throws `UnsupportedOperationException`.

See Also:

com.ibm.retail.si.mgmt.MgmtExtendedControlMBean#enableDebug()

enableDetailedControl

`public javax.management.ObjectName enableDetailedControl()`

enableDetailedLogControl

`public javax.management.ObjectName enableDetailedLogControl()`

Not supported. Throws `UnsupportedOperationException`.

See Also:

com.ibm.retail.si.mgmt.MgmtExtendedControlMBean#enableDetailedLogControl()

getCurrentlyActiveCapabilities

`public int getCurrentlyActiveCapabilities()`

getExtendedCapabilities

`public int getExtendedCapabilities()`

Returns:

com.ibm.retail.si.mgmt.MgmtExtendedControlMBean.EXTENDED_CTRL

(continued from last page)

See Also:

com.ibm.retail.si.mgmt.MgmtExtendedControlMBean#getExtendedCapabilities()

getServerID

`public java.lang.String` **`getServerID`**`()`

Returns:

the ID of this `SessionServer`, as specified in the AEF configuration file. This ID is used to derive the RMI name of the `SessionServer`if it is able to be accessed remotely.

getServerState

`public java.lang.String` **`getServerState`**`()`

Returns:

an indication of the state of the SessionServer. Valid values include `SERVER_INITIALIZING`and `SERVER_ACTIVE`.

getSessionFactoryIDs

`public java.util.List` **`getSessionFactoryIDs`**`()`

Returns:

a list of active `SessionFactory`IDs as `String`objects. These IDs can be used to query the MBeanServer since each SessionFactory MBean includes its ID in its ObjectName

See Also:

com.ibm.retail.AEF.mgmt.SessionFactoryMBean

com.ibm.retail.AEF.mgmt

Interface SoftwareInventoryMBean

All Superinterfaces:
MgmtSoftwareInventoryMBean, MgmtSimpleInventoryMBean

---

public interface **SoftwareInventoryMBean**
extends MgmtSoftwareInventoryMBean

Provides information about the AEF software level.

The purpose of this MBean is to provide information about the version and release of AEF that is currently installed and running. It extends the `com.ibm.retail.si.mgmt.MgmtSoftwareInventoryMBean`interface. It does not add any new methods; its reason for existence is as a workaround for a current JMX implementation restriction that requires MBean interfaces to be in the same package as the class implementing the interface.

The `ObjectName`of this MBean includes the following attributes, in addition to the SIF attributes of `StoreID`and `DeviceID`:
- SIFComponent=AEF
- Id=AEFSoftwareInventory
- DMajorVer=x
- DMinorVer=x
- AEFType=Version

Package
com.ibm.retail.AEF.server

Provides access to real and virtual sessions through `AEFBase`and the `SessionServer`interface.

The server package includes the classes and interfaces needed for an application to obtain an instance of an `AEFSession`object. An `AEFSession`object encapsulates the data and function of an active real or virtual POS terminal.

`AEFBase`is used to initialize an AEF environment. It reads the settings contained in the `config.properties`chain of files to determine the environment and to instantiate the necessary objects required to support the environment. Refer to Configuring the AEFfor more information.

`SessionServer`provides a simple interface for obtaining an `AEFSession`object from the AEF. It provides two methods for obtaining a session:

- **`getAvailableSession`**
  reserves and returns a session which has not already been reserved by another application. No specific terminal number is requested, so any available session may be selected by the `SessionServer`.
- **`getSession`**
  returns a session for a specific terminal, as identified by terminal number. Since this method can be used to obtain a session already reserved by another application, applications must be careful when using this method in combination with the `POSAutomationProvider`to perform actions. *Multiple applications sending input to the same session may produce unpredictable results.*

Sample Usage:

```
try
{
   SessionServer server = AEFBase.getInstance().getSessionServer();
   if (server != null)
   {
      // connected to server, now get any available AEF terminal session
      AEFSession session = server.getAvailableSession();
      if (session != null)
      {
         // got a session... perform actions or listen to events


      }
      else
      {
         // couldn't get a session -- all sessions in use
         System.err.println("Unable to obtain AEFSession.  ");
      }
   }
   else
   {
      // couldn't connect -- URL is incorrect or server is down
      System.err.println("Unable to connect to AEF SessionServer.  ");
   }
}
catch (AEFException ae)
{
   System.err.println("An AEF exception occurred: " + ae.getMessage());
}
catch (RemoteException re)
{
   System.err.println("A remote exception occurred: " + re.getMessage());
}
```

com.ibm.retail.AEF.server

Class AEFBase

```
java.lang.Object
   |
   +-com.ibm.retail.AEF.server.AEFBase
```

public class **AEFBase**
extends java.lang.Object

`AEFBase`
is the main object used to create an AEF enabled real terminal or virtual terminal environment.

It is a singleton object that creates the appropriate `SessionServer`and `AEFSessionFactory`objects based on the configuration settings. Applications typically use `AEFBase`to access the AEF `SessionServer`object through the `getSessionServer`method. The `SessionServer`provides a number of accessor to obtain instances of `AEFSession`objects. The session instances encapsulate the data and function of an active real or virtual POS terminal.

Sample usage:

```
try
{
   SessionServer server = AEFBase.getInstance().getSessionServer();
   if (server != null)
   {
      // connected to server, now get any available AEF terminal session
      AEFSession session = server.getAvailableSession();
      if (session != null)
      {
         // got a session... perform actions or listen to events


      }
      else
      {
         // couldn't get a session -- all sessions in use
         System.err.println("Unable to obtain AEFSession.  ");
      }
   }
   else
   {
      // couldn't connect -- URL is incorrect or server is down
      System.err.println("Unable to connect to AEF SessionServer.  ");
   }
}
catch (AEFException ae)
{
   System.err.println("An AEF exception occurred: " + ae.getMessage());
}
catch (RemoteException re)
{
   System.err.println("A remote exception occurred: " + re.getMessage());
}
```

Field Summary

com.ibm.retail.si.util.AEFBundle	configBundle
boolean	createAgent

boolean	createDeviceServer
boolean	createFactory
boolean	createMemoryDebug
boolean	createRMI
boolean	createServer
com.ibm.retail.AEF.util.DebugMemory	debugMemory
java.lang.String	defaultHost
com.ibm.retail.AEF.io.DeviceServer	deviceServer
java.lang.Object	deviceServerLock
int	deviceServerPort
com.ibm.retail.AEF.factory.AEFSessionFactory	factory
boolean	factoryBeacon
int	factoryBeaconInterval
java.lang.String	factoryName
boolean	factoryRemote
java.lang.String	generalrmiPort
java.net.InetAddress	group
com.ibm.retail.AEF.server.HealthServer	healthServer
static java.lang.Object	initlock
static java.lang.String	initThread
static com.ibm.retail.AEF.server.AEFBase	instance Singleton instance of AEFBase
boolean	listenForFactories
java.lang.String	localhost

com.ibm.retail.AEF.util.LoggerControl	loggerControl
boolean	loggerRemote
static boolean	mgmtEnabled
java.lang.String	multicastAddress
int	multicastPort
boolean	multicastSocketOK
java.lang.String	os4690rmiPort
java.lang.String	realTerminalNumber
java.lang.String	rmiCheckInterval
java.lang.String	rmiLeaseValue
int	rmiPort
java.lang.String	rmiPortStr
java.lang.String	rmiTimeout
java.util.HashSet	serverEntryList
java.lang.String	serverID
java.util.HashMap	serverIDToEntryMap
java.lang.String	serverName
boolean	serverRemote
com.ibm.retail.si.util.AEFBundle	sessionBundle
com.ibm.retail.AEF.server.SessionServer	sessionServer
java.net.MulticastSocket	socket
boolean	tssFlag
java.lang.Object	userObj
java.lang.String	waitForServerURI

java.lang.Object	waitForServerURILock

Constructor Summary

AEFBase() Construct AEFBase object

Method Summary

void	createDebugMemoryMBean(MgmtAgent agent) Create the DebugMemoryMBean object and register it with the MBeanServer
com.ibm.retail.AEF.io.DeviceServer	createDeviceServer(int port) Create the DeviceServer for remote device support.
void	createFactory() Create the AEFSessionFactoryobject and register with RMI.
void	createInventoryMBean(MgmtAgent agent) Create the SoftwareInventoryMBean object and register it with the MBeanServer
void	createJMXGeneralAgent() Create the JMX GeneralAgent but only if it's not already running
void	createLoggerControl() Create the LoggerControl object and register with RMI.
void	createMBeans(MgmtAgent agent) Create the MBeans
void	createMemoryDebug() Create the MemoryDebug object and register with RMI.
void	createRegistry() Create the RMI registry.
void	createSessionServer() Create the SessionServer object and register with RMI.
AEFSessionFactory	getAEFSessionFactory() Get local AEFSessionFactory.
void	getConfig(java.util.HashMap overrides) Get configuration from properties file and write to log as INFO.
java.lang.String	getDefaultHostname() Determine the host name used for this instance of AEFBase.
java.net.InetSocketAddress	getHealthServerAddress() Get the socket address of the health server.

static AEFBase	`getInstance()` Get the AEFBase singleton instance.
static AEFBase	`getInstance(java.util.HashMap overrides)` Get the AEFBase singleton instance with the overridden properties.
LoggerControl	`getLoggerControl()` Get LoggerControl object for this JVM
boolean	`getPlatformSpecificBoolean(java.lang.String prop,boolean prop)` Utility method to retrieve a boolean value from the configBundle and handle any platform-specifc values.
java.lang.String	`getRealTerminalNumber()` Get the real terminal number (for non-virtual terminals).
SessionServer	`getRemoteServer()` Get a remote SessionServer using RMI lookup.
SessionServer	`getRemoteServer(java.lang.String serverID)` Get a specific remote SessionServer using RMI lookup.
SessionServer	`getRemoteServerFromURI(java.lang.String remoteServerURL)` Get a remote SessionServer given its URI.
SessionServer	`getSessionServer()` Get a SessionServer.
SessionServer	`getSessionServer(java.lang.String serverID)` Get the SessionServer with the specified id.
static AEFBase	`initialize()` Initialize the AEFBase.
static AEFBase	`initialize(java.util.HashMap overrides)` Initialize the AEFBase with the overriden properties
void	`initImpl(java.util.HashMap overrides)` Perform initialization of the AEFBase singleton.
static boolean	`isManagementEnabled()` Return indicator of whether System Management is enabled (which implies that a JMX General Agent has been created).
boolean	`isVirtualEnvironment()` Indicates if this a virtual (CSS) environment.
void	`queryEnvironment()` Use POS services to determine if this is a POS real terminal (4690).
void	`queryHostname()` Log the IP settings.

void	`sendServerRequest(java.lang.String serverID)` Send the request for session servers
void	`setPropertyOverrides(java.lang.String filename,java.util.HashMap filename)` Set configuration property overrides.

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

instance

`protected static com.ibm.retail.AEF.server.AEFBase` **`instance`**

Singleton instance of AEFBase

configBundle

`protected com.ibm.retail.si.util.AEFBundle` **`configBundle`**

sessionBundle

`protected com.ibm.retail.si.util.AEFBundle` **`sessionBundle`**

createServer

`protected boolean` **`createServer`**

createDeviceServer

`protected boolean` **`createDeviceServer`**

createFactory

`protected boolean` **`createFactory`**

createMemoryDebug

`protected boolean` **`createMemoryDebug`**

(continued from last page)

createRMI

protected boolean **createRMI**

createAgent

protected boolean **createAgent**

mgmtEnabled

protected static boolean **mgmtEnabled**

rmiTimeout

protected java.lang.String **rmiTimeout**

rmiLeaseValue

protected java.lang.String **rmiLeaseValue**

rmiCheckInterval

protected java.lang.String **rmiCheckInterval**

serverID

protected java.lang.String **serverID**

realTerminalNumber

protected java.lang.String **realTerminalNumber**

rmiPort

protected int **rmiPort**

sessionServer

protected com.ibm.retail.AEF.server.SessionServer **sessionServer**

factory

protected com.ibm.retail.AEF.factory.AEFSessionFactory **factory**

loggerControl

protected com.ibm.retail.AEF.util.LoggerControl **loggerControl**

debugMemory

protected com.ibm.retail.AEF.util.DebugMemory **debugMemory**

userObj

protected java.lang.Object **userObj**

tssFlag

protected boolean **tssFlag**

serverRemote

protected boolean **serverRemote**

factoryRemote

protected boolean **factoryRemote**

loggerRemote

protected boolean **loggerRemote**

localhost

protected java.lang.String **localhost**

deviceServer

protected com.ibm.retail.AEF.io.DeviceServer **deviceServer**

healthServer

protected com.ibm.retail.AEF.server.HealthServer **healthServer**

(continued from last page)

deviceServerPort

protected int **deviceServerPort**

deviceServerLock

protected java.lang.Object **deviceServerLock**

factoryBeacon

protected boolean **factoryBeacon**

listenForFactories

protected boolean **listenForFactories**

factoryName

protected java.lang.String **factoryName**

serverName

protected java.lang.String **serverName**

initThread

protected static java.lang.String **initThread**

initlock

protected static java.lang.Object **initlock**

defaultHost

protected java.lang.String **defaultHost**

rmiPortStr

protected java.lang.String **rmiPortStr**

os4690rmiPort

protected java.lang.String **os4690rmiPort**

generalrmiPort

protected java.lang.String **generalrmiPort**

factoryBeaconInterval

protected int **factoryBeaconInterval**

multicastPort

protected int **multicastPort**

multicastAddress

protected java.lang.String **multicastAddress**

serverIDToEntryMap

protected java.util.HashMap **serverIDToEntryMap**

serverEntryList

protected java.util.HashSet **serverEntryList**

waitForServerURI

protected java.lang.String **waitForServerURI**

waitForServerURILock

protected java.lang.Object **waitForServerURILock**

socket

protected java.net.MulticastSocket **socket**

group

protected java.net.InetAddress **group**

(continued from last page)

multicastSocketOK

```
protected boolean multicastSocketOK
```

Constructors

AEFBase

```
protected AEFBase()
```

Construct AEFBase object

Methods

getInstance

```
public static AEFBase getInstance()
                              throws AEFException
```

Get the AEFBase singleton instance.

Returns:

instance of the AEFBase object

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.RMI_NAMING_FAILURE

---

getInstance

```
public static AEFBase getInstance(java.util.HashMap overrides)
                              throws AEFException
```

Get the AEFBase singleton instance with the overridden properties.

Returns:

instance of the AEFBase object

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.RMI_NAMING_FAILURE

---

initialize

```
public static AEFBase initialize()
                             throws AEFException
```

Initialize the AEFBase.

Returns:

instance of the AEFBase object

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.RMI_NAMING_FAILURE

initialize

```
public static AEFBase initialize(java.util.HashMap overrides)
                            throws AEFException
```

Initialize the AEFBase with the overriden properties

Returns:

instance of the AEFBase object

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.RMI_NAMING_FAILURE

setPropertyOverrides

```
public void setPropertyOverrides(java.lang.String filename,
                                 java.util.HashMap overrides)
```

Set configuration property overrides. Applications can use this method to pass property values to override configuration property values contained in the AEF property files. This provides a runtime method of changing config.properties and session.properties.

Parameters:

`property -`
file name
`hashmap -`
containing override property name/value pairs

getAEFSessionFactory

```
public AEFSessionFactory getAEFSessionFactory()
```

Get local AEFSessionFactory. Applications should access session through the `SessionServer`interface.

Returns:

AEFSessionFactory - null if no local factory exists

createDeviceServer

```
public com.ibm.retail.AEF.io.DeviceServer createDeviceServer(int port)
                                                   throws AEFException
```

Create the DeviceServer for remote device support. If device server already exists, no action is performed.

Parameters:

`port -`
device server port (use -1 to take DeviceServer default)

Returns:

DeviceServer

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.ERROR_CREATING_DEVICE_SERVER

(continued from last page)

getSessionServer

```
public SessionServer getSessionServer()
                                throws AEFException
```

Get a SessionServer.

Returns:

SessionServer

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.REMOTE_SESSION_SERVER_ERROR, AEFConst.NONE
AEFConst.NETWORK_FAILURE, AEFConst.NONE

getSessionServer

```
public SessionServer getSessionServer(java.lang.String serverID)
                                throws AEFException
```

Get the SessionServer with the specified id.

Parameters:

`serverID -`
The id of the server to return.

Returns:

SessionServer

Exceptions:

`AEFException -`
AEFException error codes are:
AEFConst.REMOTE_SESSION_SERVER_ERROR, AEFConst.NONE
AEFConst.NETWORK_FAILURE, AEFConst.NONE

getRemoteServer

```
protected SessionServer getRemoteServer(java.lang.String serverID)
                                  throws AEFException
```

Get a specific remote SessionServer using RMI lookup.

Parameters:

`serverID -`
The id of the requested server.

Returns:

SessionServer

Exceptions:

`AEFException -`
AEFException error codes are:
AEFConst.REMOTE_SESSION_SERVER_ERROR, AEFConst.NONE

getRemoteServer

```
protected SessionServer getRemoteServer()
                                  throws AEFException
```

Get a remote SessionServer using RMI lookup.

(continued from last page)

Returns:

SessionServer

Exceptions:

`AEFException -`
AEFException error codes are:
AEFConst.REMOTE_SESSION_SERVER_ERROR, AEFConst.NONE

getRemoteServerFromURI

`public SessionServer` **`getRemoteServerFromURI`**`(java.lang.String remoteServerURL)`

Get a remote SessionServer given its URI.

Parameters:

`remoteServerURL`

Returns:

SessionServer

isVirtualEnvironment

`public boolean` **`isVirtualEnvironment`**`()`

Indicates if this a virtual (CSS) environment.

Returns:

true if virtual environment (CSS)

getRealTerminalNumber

`public java.lang.String` **`getRealTerminalNumber`**`()`

Get the real terminal number (for non-virtual terminals).

Returns:

string representation of the terminal number (e.g. "021")

getLoggerControl

`public LoggerControl` **`getLoggerControl`**`()`

Get LoggerControl object for this JVM

Returns:

LoggerControl

initImpl

```
protected void initImpl(java.util.HashMap overrides)
                  throws AEFException
```

Perform initialization of the AEFBase singleton.

Exceptions:

`com.ibm.retail.AEF.util.AEFException`

(continued from last page)

getConfig

```
protected void getConfig(java.util.HashMap overrides)
```

Get configuration from properties file and write to log as INFO.

Parameters:

`overrides -`
contains in proerty value overrides from application

getPlatformSpecificBoolean

```
protected boolean getPlatformSpecificBoolean(java.lang.String prop,
                                             boolean initValue)
```

Utility method to retrieve a boolean value from the configBundle and handle any platform-specifc values. This method supports the "4690terminal", "4690css", and "general" platforms.

createRegistry

```
protected void createRegistry()
                       throws AEFException
```

Create the RMI registry.

createJMXGeneralAgent

```
protected void createJMXGeneralAgent()
                               throws AEFException
```

Create the JMX GeneralAgent but only if it's not already running

createMBeans

```
protected void createMBeans(MgmtAgent agent)
```

Create the MBeans

createInventoryMBean

```
protected void createInventoryMBean(MgmtAgent agent)
```

Create the SoftwareInventoryMBean object and register it with the MBeanServer

createDebugMemoryMBean

```
protected void createDebugMemoryMBean(MgmtAgent agent)
```

Create the DebugMemoryMBean object and register it with the MBeanServer

createSessionServer

```
protected void createSessionServer()
                             throws AEFException
```

Create the SessionServer object and register with RMI.

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.RMI_NAMING_FAILURE

(continued from last page)

createLoggerControl

```
protected void createLoggerControl()
                              throws AEFException
```

Create the LoggerControl object and register with RMI.

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.RMI_NAMING_FAILURE

createFactory

```
protected void createFactory()
                       throws AEFException
```

Create the `AEFSessionFactory`object and register with RMI.

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.RMI_NAMING_FAILURE

createMemoryDebug

```
protected void createMemoryDebug()
                          throws AEFException
```

Create the MemoryDebug object and register with RMI.

Exceptions:

`com.ibm.retail.AEF.util.AEFException`

queryEnvironment

```
protected void queryEnvironment()
```

Use POS services to determine if this is a POS real terminal (4690).

queryHostname

```
protected void queryHostname()
```

Log the IP settings.

getDefaultHostname

```
public java.lang.String getDefaultHostname()
```

Determine the host name used for this instance of AEFBase.

Returns:

default host name

getHealthServerAddress

```
public java.net.InetSocketAddress getHealthServerAddress()
```

Get the socket address of the health server.

Returns:

(continued from last page)

socket address of health server

sendServerRequest

`protected void **sendServerRequest**(java.lang.String serverID)`

Send the request for session servers

Parameters:

`serverID -`
The id of the requested server, or null.

isManagementEnabled

`public static boolean **isManagementEnabled**()`

Return indicator of whether System Management is enabled (which implies that a JMX General Agent has been created). Various AEF components check this flag to determine whether to create MBeans.

Returns:

flag indicating whether system management is enabled.

com.ibm.retail.AEF.server

Class AEFBase.AEFMulticastThread

```
java.lang.Object
  |
  +-com.ibm.retail.AEF.server.AEFBase.AEFMulticastThread
```

All Implemented interfaces:
java.lang.Runnable

---

public class **AEFBase.AEFMulticastThread**
extends java.lang.Object
implements java.lang.Runnable

`AEFMulticastThread`
is a thread which can handle listening to the multicast socket for beaconing factories, as well as beacon on behalf of a factory.

Field Summary

byte[]	blankBuf
byte[]	broadcastBuf
long	lastBeaconTime
byte[]	receiveBuf
byte[]	serverURIBuf
int	timeout

Constructor Summary

AEFBase.AEFMulticastThread() Construct AEFMulticastThread object

Method Summary

java.net.DatagramPacket	listenForIncoming() Listens on the multicast socket for an incoming message.
void	processDuplicateTerminalNumber(java.lang.String incomingBuffer) Process the incoming duplicate terminal message.
void	processFactoryBeacon(java.lang.String factoryBeacon,java.net.DatagramPacket factoryBeacon) Process the incoming factory beacon.

void	`processServerBeacon(java.lang.String serverBeacon)` Process the incoming server beacon.
void	`processServerRequest(java.lang.String serverRequest,java.net.DatagramPacket serverRequest)` Process the incoming server request.
void	`run()` Main processing loop
void	`sendDuplicateTermNumMessage(java.net.DatagramPacket incomingPacket,java.lang.String incomingPacket)` Send a directed packet back to the originator of the factory beacon to indicate that the factory is trying to come on-line with a terminal number already duplicated in another factory.
void	`sendFactoryBeacon(boolean honorBeaconInterval)` Send the factory beacon
void	`sendServerBeacon()` Send an outgoing message containing the server's URI.

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

timeout

`protected int` **`timeout`**

lastBeaconTime

`protected long` **`lastBeaconTime`**

broadcastBuf

`protected byte` **`broadcastBuf`**

receiveBuf

`protected byte` **`receiveBuf`**

blankBuf

`protected byte` **`blankBuf`**

(continued from last page)

serverURIBuf

`protected byte` **`serverURIBuf`**

Constructors

AEFBase.AEFMulticastThread

`public` **`AEFBase.AEFMulticastThread`**`()`

Construct AEFMulticastThread object

Methods

run

`public void` **`run`**`()`

Main processing loop

sendFactoryBeacon

`protected void` **`sendFactoryBeacon`**`(boolean honorBeaconInterval)`

Send the factory beacon

Parameters:

`honorBeaconInterval -`
If true, this method will not send the beacon unless the time since the last beacon is greater than the beacon interval.

processFactoryBeacon

```
protected void processFactoryBeacon(java.lang.String factoryBeacon,
                                    java.net.DatagramPacket incomingPacket)
```

Process the incoming factory beacon.

processServerRequest

```
protected void processServerRequest(java.lang.String serverRequest,
                                    java.net.DatagramPacket incomingPacket)
```

Process the incoming server request. A server request is issued to discover SessionServers on the network. The request may be issued for all servers, or a specific server by name.

processServerBeacon

`protected void` **`processServerBeacon`**`(java.lang.String serverBeacon)`

Process the incoming server beacon.

processDuplicateTerminalNumber

`protected void` **`processDuplicateTerminalNumber`**`(java.lang.String incomingBuffer)`

Process the incoming duplicate terminal message.

(continued from last page)

sendServerBeacon

```
protected void sendServerBeacon()
```

Send an outgoing message containing the server's URI.

sendDuplicateTermNumMessage

```
protected void sendDuplicateTermNumMessage(java.net.DatagramPacket incomingPacket,
                                           java.lang.String factoryID)
```

Send a directed packet back to the originator of the factory beacon to indicate that the factory is trying to come on-line with a terminal number already duplicated in another factory.

listenForIncoming

```
protected java.net.DatagramPacket listenForIncoming()
```

Listens on the multicast socket for an incoming message. We'll block for the timeout period. If we didn't receive anything, a null string is returned.

Returns:

DatagramPacket An incoming packet, or null if we timed out.

com.ibm.retail.AEF.server

Class AEFBase.ServerEntry

```
java.lang.Object
   |
   +-com.ibm.retail.AEF.server.AEFBase.ServerEntry
```

public class **AEFBase.ServerEntry**
extends java.lang.Object

This class is used to hold information about a SessionServer.

Field Summary

com.ibm.retail.AEF.server.SessionServer	server
java.lang.String	serverID
java.lang.String	URI

Constructor Summary

AEFBase.ServerEntry()

Methods inherited from : class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

serverID

public java.lang.String **serverID**

URI

public java.lang.String **URI**

server

public com.ibm.retail.AEF.server.SessionServer **server**

Constructors

(continued from last page)

AEFBase.ServerEntry

```
public AEFBase.ServerEntry()
```

com.ibm.retail.AEF.server

Interface LoadBalancer

public interface **LoadBalancer**

`LoadBalancer`
provides an API to enable load balancing logic to be used by the AEF `SessionServer`.

Classes implementing the `LoadBalancer`interface may be used as the load balancing implementation by overriding the default `LoadBalancer`class name specified in the `classes.properties`file.

Method Summary

`void`	`factoryAdded(AEFSessionFactoryInfo factoryInfo)` Called by the session server when a factory has been added.
`void`	`factoryRemoved(AEFSessionFactoryInfo factoryInfo)` Called by the session server when a factory has been removed.
`void`	`factoryUpdated(AEFSessionFactoryInfo factoryInfo)` Called by the session server when factory info has been updated.
`AEFSession`	`getAvailableSession(SessionParameters parms)` Get an available session using a factory selected by the LoadBalancer.
`SessionServer`	`getSessionServer()` Gets the SessionServer which owns this load balancer.
`void`	`setSessionServer(SessionServer server)` Sets the SessionServer which owns this load balancer.

Methods

factoryAdded

`public void` **`factoryAdded`**`(AEFSessionFactoryInfo factoryInfo)`

Called by the session server when a factory has been added. This method should update any internal data structure required to keep track of the factory. The AEFSessionFactoryInfo.getKey() will return a key that can be used to tell the session server which factory to utilize for a session request.

Parameters:
`factoryInfo` - AEFSessionFactoryInfo

factoryRemoved

`public void` **`factoryRemoved`**`(AEFSessionFactoryInfo factoryInfo)`

Called by the session server when a factory has been removed. This method should update any internal data structure required to keep track of the factory. The AEFSessionFactoryInfo.getKey() will return a key that can be used to tell the session server which factory to utilize for a session request.

(continued from last page)

Parameters:

`factoryInfo -`
AEFSessionFactoryInfo

factoryUpdated

```
public void factoryUpdated(AEFSessionFactoryInfo factoryInfo)
```

Called by the session server when factory info has been updated. This method should update any internal data structure required to keep track of the factory. The AEFSessionFactoryInfo.getKey() will return a key that can be used to tell the session server which factory to utilize for a session request.

Parameters:

`factoryInfo -`
AEFSessionFactoryInfo

getSessionServer

```
public SessionServer getSessionServer()
                                 throws java.rmi.RemoteException
```

Gets the SessionServer which owns this load balancer.

Returns:

SessionServer

Exceptions:

`RemoteException -`
if remote access fails

setSessionServer

```
public void setSessionServer(SessionServer server)
                       throws java.rmi.RemoteException
```

Sets the SessionServer which owns this load balancer.

Parameters:

`SessionServer`

Exceptions:

`RemoteException -`
if remote access fails

getAvailableSession

```
public AEFSession getAvailableSession(SessionParameters parms)
                                 throws AEFException
```

Get an available session using a factory selected by the LoadBalancer. Pluggable LoadBalancer implementations may implement a variety of load balancing algorithms to distribute requests. The `SessionServer`invokes this method whenever a client request for a an available AEFSession is received. The `SessionServer`defers all factory selection to the LoadBalancer. If no factory can satisfy the request, an exception is thrown.

Parameters:

`parms -`
SessionParameters object contains all parms for the session as name-value pairs.

Returns:

an avaiable AEFSession or null if none exists

Exceptions:

(continued from last page)

`AEFException -`
if a session can not be accessed Among the possible AEFException error codes are:
AEFConst.NO_SESSION_FACTORIES_FOUND, AEFConst.NONE
AEFConst.SERVER_NO_SESSION_AVAILABLE, AEFConst.NONE

com.ibm.retail.AEF.server

Interface SessionServer

public interface **SessionServer**
extends java.rmi.Remote

`SessionServer`
provides the client interface for requesting `AEFSession`objects.

Clients may request either an "available" session (i.e., a session that is not in use by another session) or a specific session by terminal number. If no session is available, an `AEFException`is thrown. The session returned via a call to `getAvailableSession`will not be available to any other callers of `getAvailableSession`until the `session.release` method is called. Note that getting an available session does not prevent other clients from accessing the session by specifying the specific terminal number.

Examples: (`server`is an object of type `SessionServer`)

Obtaining any available session:

```
try
{
  AEFSession mySession = server.getAvailableSession();
  if (mySession != null)
  {
     System.out.println("Using terminal number: " + mySession.getTerminalNumber();
  }
}
catch (RemoteException re)
{
   System.err.println("Couldn't access remote session");
}
catch (AEFException ae)
{
   System.err.println("Couldn't obtain an available session");
}
```

Obtaining a specific session without regard to availability:

To obtain a specific session, either specify the terminal number or, if in a non-TSS environment where only a single session exists, use the constant `LOCAL_SESSION`. In this case, a session is returned if it is a valid terminal number, even if it is already assigned to a client.

```
AEFSession mySession = server.getSession("100");
AEFSession aSession = server.getSession(SessionServer.LOCAL_SESSION);
```

Releasing a session:

Clients are responsible for releasing sessions after they are finished using the session. This mechanism allows other clients to access the session through the `getAvailableSession`method. Since sessions are a limited resource, this is a critical responsibility of each client and failure to perform this step may cause performance and memory degradation, and may eventually exhaust the capacity of the server.

Sessions are released through the `releaseSession`method:

```
server.releaseSession(mySession);
mySession = null;
```

(Note: it is important to set the session pointer to null since the client still has a valid pointer to a session, but no longer "owns" the session).

Field Summary

static java.lang.String	`ACTIVE`
static java.lang.String	`INITIALIZING` Valid values for the state of this object, as returned by `getState()`
static java.lang.String	`LOCAL_SESSION` Identifier to be used to obtain the local terminal session in non-TSS.
static java.lang.String	`NAME` Name used to register this object with RMI naming

Method Summary

boolean	`addFactory(AEFSessionFactory factory)` Register an AEFSessionFactory with the session server.
java.lang.String	`addFactoryBeaconInfo(java.lang.String factoryBeaconInfo)` Register a AEFSessionFactory with the session server given it's beacon info.
void	`destroySession(java.lang.String terminalNumber)` Destroy an AEFSession.
AEFSession	`getAvailableSession()` Get an AEFSession object with a status of "available".
AEFSession	`getAvailableSession(int timeout)` Get an AEFSession object with a status of "available".
AEFSession	`getAvailableSession(SessionParameters parms)` Get an AEFSession object with a status of "available".
AEFSessionFactory	`getFactory(int index)` Get a factory registered with this session server.
AEFSessionFactory	`getFactoryFromInfo(AEFSessionFactoryInfo info,java.lang.StringBuffer info)` Get a session factory reference given it's info object.
java.util.List	`getFactoryIDs()` Get the number of factories registered with this session server.
java.lang.String	`getID()` Get the identifier for this session server.
LoggerControl	`getLoggerControl()` Get the LoggerControl object for the Server JVM.

int	`getNumberOfFactories()` Get the IDs of the factories registered with this session server.
SessionServer	`getSelfReference()` Returns a self reference.
AEFSession	`getSession(java.lang.String terminalNumber)` Get a AEFSession object by terminal number.
AEFSession	`getSession(java.lang.String terminalNumber,int terminalNumber)` Get a AEFSession object by terminal number.
AEFSession	`getSession(java.lang.String terminalNumber,SessionParameters terminalNumber)` Get a AEFSession object by terminal number.
AEFSession	`getSessionFromFactory(AEFSessionFactory factory,java.lang.String factory,SessionParameters factory)` Get a session from a factory.
java.lang.String	`getState()` Get the state of this session server.
void	`releaseSession(AEFSession session)` Release an AEFSession to free it for other clients to use
void	`removeFactory(AEFSessionFactory factory)` Remove a registered AEFSessionFactory from the session server.
boolean	`sessionExists(java.lang.String terminalNumber)` Indicates if a session for this terminal number has been created.
void	`updateFactory(AEFSessionFactory factory)` Update factory info from a AEFSessionFactory with the session server.

Fields

NAME

`public static final java.lang.String` **`NAME`**

Name used to register this object with RMI naming

LOCAL_SESSION

`public static final java.lang.String` **`LOCAL_SESSION`**

Identifier to be used to obtain the local terminal session in non-TSS. (Resolves to the actual terminal number). Example: `mySession = server.getSession(SessionServer.LOCAL_SESSION)`

INITIALIZING

`public static final java.lang.String` **`INITIALIZING`**

Valid values for the state of this object, as returned by `getState()`

ACTIVE

```
public static final java.lang.String ACTIVE
```

Methods

getAvailableSession

```
public AEFSession getAvailableSession()
                                 throws java.rmi.RemoteException,
                                        AEFException
```

Get an AEFSession object with a status of "available". An AEFSession is in the available state if no other clients have references to it This method establishes a client reference to the session and removes it from the pool of available sessions. **Clients must use the *release()* method to return the session to the pool of available sessions.** A session must be in a "ready state", ie ready for input. An exception is thrown otherwise. This method uses the default timeout value specified by the property `ready.wait.timeout` contained in the `config.properties` file. If the client wishes the AEF to log some sort of client id with the session number, then the client should call `getAvailableSession(parms)` instead, where parms includes the client ID.

Returns:

an instance of an AEFSession object

Exceptions:

`RemoteException`
`AEFException -`
if a session can not be accessed AEFException error codes are:
AEFConst.NO_SESSION_FACTORIES_FOUND, AEFConst.NONE
AEFConst.SERVER_NO_SESSION_AVAILABLE, AEFConst.NONE
AEFConst.SESSION_READY_WAIT_TIMEOUT, AEFConst.NONE - the timeout associated with ready.wait.timeout expired before the session was ready.

See Also:

com.ibm.retail.AEF.session.AEFSession#release() Releasing a session

getAvailableSession

```
public AEFSession getAvailableSession(int timeout)
                                 throws java.rmi.RemoteException,
                                        AEFException
```

Get an AEFSession object with a status of "available". An AEFSession is in the available state if no other clients have references to it. This method establishes a client reference to the session and removes it from the pool of available sessions. **Clients must use the *release()* method to return the session to the pool of available sessions.** This method provides a parameter for specifying a timeout value for waiting on a "ready" session. A timeout value of -1 can be used to specify that no wait is to be performed and "non-ready" sessions should be returned. This allows clients access to sessions which have not reached the ready state. If the client wishes the AEF to log some sort of client id with the session number, then the client should call `getAvailableSession(parms)` instead, where parms includes the client ID.

Parameters:

`timeout -`
time to wait for sessions which are not ready for input (milliseconds)

Use a timeout value of -1 to indicate that the session should be returned even if it is not in a ready state.

Returns:

an instance of an AEFSession object

Exceptions:

(continued from last page)

`RemoteException`
`AEFException -`
if a session can not be accessed AEFException error codes are:
AEFConst.NO_SESSION_FACTORIES_FOUND, AEFConst.NONE
AEFConst.SERVER_NO_SESSION_AVAILABLE, AEFConst.NONE
AEFConst.SESSION_READY_WAIT_TIMEOUT, AEFConst.NONE - the timeout associated with ready.wait.timeout expired before the session was ready.

See Also:

com.ibm.retail.AEF.session.AEFSession#release() Releasing a session

getAvailableSession

```
public AEFSession getAvailableSession(SessionParameters parms)
                                 throws java.rmi.RemoteException,
                                        AEFException
```

Get an AEFSession object with a status of "available". An AEFSession is in the available state if no other clients have references to it. This method establishes a client reference to the session and removes it from the pool of available sessions. **Clients must use the *release()* method to return the session to the pool of available sessions.** If the client wishes the AEF to log some sort of client id with the session number, then the client should call `setClientID`on the `SessionParamemters`argument. When a session is returned, an AEF log entry will be made to record the association between the client ID and the session ID. This information could be used to determine which session was associated with a particular client. This method provides a SessionParameter object for specifying parameters.

Parameters:

`parms -`
SessionParameters object specifies all parameters for this session

Returns:

an instance of an AEFSession object

Exceptions:

`RemoteException`
`AEFException -`
if a session can not be accessed AEFException error codes are:
AEFConst.NO_SESSION_FACTORIES_FOUND, AEFConst.NONE
AEFConst.SERVER_NO_SESSION_AVAILABLE, AEFConst.NONE
AEFConst.SESSION_READY_WAIT_TIMEOUT, AEFConst.NONE - the timeout associated with ready.wait.timeout expired before the session was ready.

See Also:

com.ibm.retail.AEF.session.AEFSession#release() Releasing a session

getSession

```
public AEFSession getSession(java.lang.String terminalNumber)
                      throws java.rmi.RemoteException,
                             AEFException
```

Get a AEFSession object by terminal number. If this is a valid terminal number for the factory, the session is returned, regardless of the number of client accesses. This allows sharing of a session among multiple clients. This method does not remove the session from pool of available sessions. Clients can remove the session from the available session pool by invoking the *reference()*method of the AEFSession. A session must be in a "ready state", ie ready for input. An exception is thrown otherwise. This method uses the default timeout value specified in the *config.properties*property=ready.wait.timeout. If the client wishes the AEF to log some sort of client id with the session number, then the client should call `getSession(String terminalNumber, SessionParameters parms)`instead, where parms includes the client ID.

Parameters:

`terminalNumber -`
identifies the session. Use `SessionServer.LOCAL_SESSION`to request the real terminal session running in a real terminal.

(continued from last page)

Returns:

an instance of an AEFSession object

Exceptions:

`RemoteException`
`AEFException -`
if a session can not be accessed AEFException error codes are:
AEFConst.ERROR_ACCESSING_FACTORY, AEFConst.NONE
AEFConst.FACTORY_NO_SESSIONS_AVAILABLE, AEFConst.NONE
AEFConst.INVALID_TERMINAL_NUMBER, AEFConst.NONE
AEFConst.SESSION_READY_WAIT_TIMEOUT, AEFConst.NONE

See Also:

com.ibm.retail.AEF.session.AEFSession#reference Referencing a session

getSession

```
public AEFSession getSession(java.lang.String terminalNumber,
                             int timeout)
                      throws java.rmi.RemoteException,
                             AEFException
```

Get a AEFSession object by terminal number. If this is a valid terminal number for the factory, the session is returned, regardless of the number of client accesses. This allows sharing of a session among multiple clients. This method does not remove the session from pool of available sessions. Clients can remove the session from the available session pool by invoking the *reference()*method of the AEFSession. This method provides a parameter for specifying a timeout value for waiting on a "ready" session. A timeout value of -1 can be used to specify that no wait is to be performed and "non-ready" sessions should be returned. This allows clients access to sessions which have not reached the ready state. If the client wishes the AEF to log some sort of client id with the session number, then the client should call `getSession(String terminalNumber, SessionParameters parms)`instead, where parms includes the client ID.

Parameters:

`terminalNumber -`
identifies the session. Use `SessionServer.LOCAL_SESSION`to request the real terminal session running in a real terminal.
`time -`
to wait for sessions which are not ready for input (milliseconds) (Use a timeout value of -1 to indicate that the session should be returned even if it is not in a ready state)

Returns:

an instance of a AEFSession object

Exceptions:

`RemoteException`
`AEFException -`
if a session can not be accessed AEFException error codes are:
AEFConst.INVALID_TERMINAL_NUMBER, AEFConst.NONE
AEFConst.FACTORY_NO_SESSIONS_AVAILABLE, AEFConst.NONE
AEFConst.SESSION_READY_WAIT_TIMEOUT, AEFConst.NONE

See Also:

com.ibm.retail.AEF.session.AEFSession#reference Referencing a session

getSession

```
public AEFSession getSession(java.lang.String terminalNumber,
                             SessionParameters parms)
                      throws java.rmi.RemoteException,
                             AEFException
```

(continued from last page)

Get a AEFSession object by terminal number. If this is a valid terminal number for the factory, the session is returned, regardless of the number of client accesses. This allows sharing of a session among multiple clients. This method does not remove the session from pool of available sessions. Clients can remove the session from the available session pool by invoking the *reference()* method of the AEFSession. This method provides a SessionParameter object for specifying the session parms. If the client wishes the AEF to log some sort of client id with the session number, then the client should call `setClientID` on the `SessionParamemters` argument. When a session is returned, an AEF log entry will be made to record the association between the client ID and the session ID. This information could be used to determine which session was associated with a particular client.

Parameters:

`terminalNumber -`
identifies the session. Use `SessionServer.LOCAL_SESSION` to request the real terminal session running in a real terminal.
`parms -`
SessionParameters object contains the session parameter vales

Returns:

an instance of a AEFSession object

Exceptions:

`RemoteException`
`AEFException -`
if a session can not be accessed AEFException error codes are:
AEFConst.INVALID_TERMINAL_NUMBER, AEFConst.NONE
AEFConst.FACTORY_NO_SESSIONS_AVAILABLE, AEFConst.NONE
AEFConst.SESSION_READY_WAIT_TIMEOUT, AEFConst.NONE

See Also:

com.ibm.retail.AEF.session.AEFSession#reference Referencing a session

releaseSession

```
public void releaseSession(AEFSession session)
                     throws java.rmi.RemoteException,
                            AEFException
```

Release an AEFSession to free it for other clients to use

Parameters:

`session -`
AEFSession to be released

Exceptions:

`RemoteException -`
if server is not available
`AEFException -`
if an error occurs accessing a session

destroySession

```
public void destroySession(java.lang.String terminalNumber)
                     throws java.rmi.RemoteException,
                            AEFException
```

Destroy an AEFSession. This method applies only to virtual (TSS) sessions. It terminates the POS sales application and removes the AEFSession from the SessionPool.

Parameters:

`terminalNumber -`
identifies the session

Exceptions:

`RemoteException`

(continued from last page)

`AEFException -`
if a session can not be destroyed AEFException error codes are:
AEFConst.SESSION_NOT_ACTIVE, AEFConst.NONE
AEFConst.API_VALID_FOR_TSS_ONLY, AEFConst.NONE
AEFConst.INVALID_TERMINAL_NUMBER, AEFConst.NONE

sessionExists

```
public boolean sessionExists(java.lang.String terminalNumber)
                        throws java.rmi.RemoteException
```

Indicates if a session for this terminal number has been created.

Parameters:

`terminal -`
number

Returns:

true if the session for this terminal exists

Exceptions:

`RemoteException`

addFactory

```
public boolean addFactory(AEFSessionFactory factory)
                     throws java.rmi.RemoteException
```

Register an AEFSessionFactory with the session server. This method is used by factories which create and provide session instances to the server. Several factories may exist on local and remote servers, load balanced by the session server.

Parameters:

`factory -`
AEFSessionFactory to be added

Returns:

boolean true if the factory was added successfully.

Exceptions:

`RemoteException`

addFactoryBeaconInfo

```
public java.lang.String addFactoryBeaconInfo(java.lang.String factoryBeaconInfo)
                                       throws java.rmi.RemoteException
```

Register a AEFSessionFactory with the session server given it's beacon info.

Parameters:

`beaconInfo -`
Info from the factory beacon.

Returns:

null if the factory was added successfully; if there was a terminal number collision with an already existing factory, then this method returns the name of the existing factory.

Exceptions:

`RemoteException -`
if server is not available

(continued from last page)

removeFactory

```
public void removeFactory(AEFSessionFactory factory)
                    throws java.rmi.RemoteException
```

Remove a registered AEFSessionFactory from the session server.

Parameters:

`factory -`
AEFSessionFactory to be removed

Exceptions:

`RemoteException -`
if server is not available

updateFactory

```
public void updateFactory(AEFSessionFactory factory)
                    throws java.rmi.RemoteException,
                           AEFException
```

Update factory info from a AEFSessionFactory with the session server. This method is used by factories as a callback to keep the SessionServer up-to-date with factory information. Several factories may exist on local and remote servers, load balanced by the session server.

Parameters:

`factory -`
AEFSessionFactory to be updated

Exceptions:

`RemoteException`
`AEFException`

getFactoryIDs

```
public java.util.List getFactoryIDs()
                              throws java.rmi.RemoteException
```

Get the number of factories registered with this session server. Several factories may exist on local and remote servers, load balanced by the session server.

Returns:

Number of factories

Exceptions:

`RemoteException -`
if server is not available

getNumberOfFactories

```
public int getNumberOfFactories()
                           throws java.rmi.RemoteException
```

Get the IDs of the factories registered with this session server. Several factories may exist on local and remote servers, load balanced by the session server.

Returns:

Number of factories

Exceptions:

`RemoteException -`
if server is not available

getFactoryFromInfo

```
public AEFSessionFactory getFactoryFromInfo(AEFSessionFactoryInfo info,
                                            java.lang.StringBuffer narrative)
                                     throws java.rmi.RemoteException
```

Get a session factory reference given it's info object. This method will attempt to establish a remote reference if we have a URI to go on. If we have a URI, but cannot establish the reference, the factory info will be removed from the session server.

Parameters:

`info -`
The AEFSessionFactoryInfo object which contains the factory URI.
`narrative -`
A StringBuffer to which will be appended a description of the attempt to get the factory. This may be set to null if no narrative is desired.

Returns:

AEFSessionFactory

Exceptions:

`RemoteException`

getSessionFromFactory

```
public AEFSession getSessionFromFactory(AEFSessionFactory factory,
                                        java.lang.String terminalNumber,
                                        SessionParameters parms)
                                  throws java.rmi.RemoteException,
                                        AEFException
```

Get a session from a factory.

Parameters:

`factory -`
AEFSessionFactory
`terminalNumber -`
identifies terminal number or null for any available terminal
`parms -`
SessionParameters object contains all parms for the session as name-value pairs.

Returns:

AEFSession

Exceptions:

`RemoteException`
`AEFException -`
AEFException return codes are:
AEFConst.ERROR_ACCESSING_FACTORY, AEFConst.NONE
AEFConst.INVALID_TERMINAL_NUMBER, AEFConst.NONE
AEFConst.FACTORY_NO_SESSIONS_AVAILABLE, AEFConst.NONE
AEFConst.FACTORY_SESSION_NOT_READY, AEFConst.NONE

getFactory

```
public AEFSessionFactory getFactory(int index)
                             throws java.rmi.RemoteException,
                                    AEFException
```

Get a factory registered with this session server. Several factories may exist on local and remote servers, load balanced by the session server.

(continued from last page)

Parameters:

`index -`
index of factory

Returns:

Factory

Exceptions:

`AEFException`
`RemoteException`

getLoggerControl

```
public LoggerControl getLoggerControl()
                                throws java.rmi.RemoteException,
                                       AEFException
```

Get the LoggerControl object for the Server JVM.

Returns:

instance of LoggerControl

Exceptions:

`RemoteException`
`AEFException`

getID

```
public java.lang.String getID()
                        throws java.rmi.RemoteException
```

Get the identifier for this session server.

Returns:

Server ID

Exceptions:

`RemoteException`

getState

```
public java.lang.String getState()
                           throws java.rmi.RemoteException
```

Get the state of this session server. Valid values are `SessionServerMBean.SERVER_INITIALIZING`and `SessionServerMBean.SERVER_ACTIVE`.

Returns:

state

Exceptions:

`RemoteException`

getSelfReference

```
public SessionServer getSelfReference()
                                throws java.rmi.RemoteException
```

(continued from last page)

Returns a self reference. This method is used to retrieve the actual factory reference if given a proxy reference of the type SessionServerRef.

Returns:

SessionServer

Exceptions:

`RemoteException`

com.ibm.retail.AEF.server

Class StandAloneSessionServer

```
java.lang.Object
   |
   +-com.ibm.retail.AEF.server.StandAloneSessionServer
```

public class **StandAloneSessionServer**
extends java.lang.Object

StandAloneSessionServer includes a main that will produce a SessionServer. This class is intended for use in situations where a standalone session server is desired. For example, this class may be run on a Windows or Linux server in the store to produce an independent session server.

Here is an example of a Windows command file that can be used to execute this class:

```
 rem set %java_home% to root java install directory.
 rem set %aef_jars% to directory where AEF jar files are installed.
 rem set %aef_nic_ip_address% to the IP address of the AEF netword card if multi-homed.
 %java_home%\bin\java -cp %aef_jars%\siutil.jar;%aef_jars%\aef.jar;%aef_jars%\commons-
logging.jar;. -Djava.rmi.server.hostname=%aef_nic_ip_address%
com.ibm.retail.AEF.server.StandAloneSessionServer
```

Constructor Summary

`StandAloneSessionServer()`

Method Summary

`static void`	`main(java.lang.String[] args)` The main routine.

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

StandAloneSessionServer

`public` **`StandAloneSessionServer`**`()`

Methods

main

`public static void` **`main`**`(java.lang.String[] args)`

(continued from last page)

The main routine.

Parameters:

`args` -
an array of strings for command line parameters

Package
com.ibm.retail.AEF.session

Provides the `AEFSession`interface which encapsulates the data and function of a real or virtual terminal session.

The session package defines the `AEFSession`interface which provides the encapsulation of an active POS terminal session. Instances of `AEFSession`are obtained through the `SessionServer`interface using a `getSession`or `getAvailableSession` method.

`AEFSession`provides the interface to obtain and utilize the services provided by the AEF `POSAutomationProvider`and `POSDataProvider`interfaces. These interfaces provide access to the function and data provided by the POS terminal session.

com.ibm.retail.AEF.session

Interface AEFSession

public interface **AEFSession**
extends java.rmi.Remote

An `AEFSession`provides an object encapsulation of a POS terminal session. The `AEFSession`provides access to the data and function provided by the POS application. An instance of an AEFSession is distinguished by the *terminal number*associated with the session. Each session is identified by a unique terminal number. A session runs in either in "real" terminal mode (i.e., in a POS Terminal with real devices) or in a "virtual" terminal mode (i.e., in a server environment with emulated devices.
To perform an action on an `AEFSession`(e.g., logon(), addItem()), use the `POSAutomationProvider`interface.
To access data and monitor events associated with an AEFSession, use the use the `POSDataProvider`interface.

See Also:
#getPOSDataProvider() Accessing the POSDataProvider, #getPOSAutomationProvider() Accessing the POSAutomationProvider

Method Summary

void	`addSessionStatusListener(SessionStatusListener listener)` Add a listener for SessionStatus events.
void	`addSessionTraceSocketHandler(java.lang.String host,int host)` Add a socket handler to the session trace.
void	`addToEventQueue(AEFEventQueueRunnable obj)` Add a runnable object to the sessions event queue.
void	`dumpSessionTrace()` Dump the session trace buffer to the trace file.
ApplicationDataConnector	`getApplicationDataConnector()` Get the `ApplicationDataConnector`for this terminal session.
java.net.InetSocketAddress	`getHealthServerAddress()` Get the address of the AEF Health Server for this session.
POSAutomationProvider	`getPOSAutomationProvider()` Get the `POSAutomationProvider`(automation API) for this terminal session.
POSDataProvider	`getPOSDataProvider()` Get the `POSDataProvider`associated with this terminal session.
int	`getSessionTraceLevel()` Returns the current trace level for the session.
java.lang.String	`getTerminalNumber()` Get the terminal number for this terminal session.
java.lang.String[]	`getTraceBuffer()` Returns a String array of the trace buffer.

Workstation	`getWorkstation()` Returns the Workstation object for this session.
boolean	`isApplicationActive()` Is the terminal sales application active? **Note:**this method applies only to virtual (CSS) sessions.
boolean	`isAutoDumpSessionTrace()` Indicates if the session trace buffer auto dumps for AEF exceptions.
boolean	`isAvailable()` Indicate if this session is available.
boolean	`isReady()` Indicate if this session ready for client actions.
boolean	`isReserved()` Indicate if this session is reserved.
boolean	`isSessionTraceEnabled(int level)` Indicates if session trace is enabled for this session for a specified log level.
boolean	`isTSSSession()` Indicates if this a virtual session object.
boolean	`isVirtualSession()` Indicates if this a virtual session object.
void	`reference()` Increment the client reference count for this session.
void	`release()` Decrement the client reference count for this session.
void	`removeSessionStatusListener(SessionStatusListener listener)` Remove a listener for SessionStatus events.
void	`removeSessionTraceSocketHandler()` Remove the session trace socket handler.
void	`sessionTrace(int level,java.lang.String level)` Write to the session trace buffer with the specified level.
void	`setSessionTraceLevel(int level)` Set the trace level for the session.
void	`startApplication()` Start the terminal sales application.
void	`stopApplication()` Stop the terminal sales application.
void	`terminate()` Cause the session to perform any cleanup required before it is terminated.

void	`waitForPendingEvents()` Wait for events currently pending on the session event queue to complete.
boolean	`waitUntilReady(int timeout)` Wait until this session is ready for processing.

Methods

getPOSDataProvider

```
public POSDataProvider getPOSDataProvider()
                                    throws java.rmi.RemoteException
```

Get the `POSDataProvider`associated with this terminal session. The POSDataProvider provides access to listener interfaces and access to the data associated with the AEFSession.

Returns:

POSDataProvider object for this terminal session

Exceptions:
`RemoteException`

getPOSAutomationProvider

```
public POSAutomationProvider getPOSAutomationProvider()
                                                 throws java.rmi.RemoteException
```

Get the `POSAutomationProvider`(automation API) for this terminal session. The POSAutomationProvider allows actions (e.g., logon, add an item) to be performed on the AEFSession.

Returns:

POSAutomationProvider object for this terminal session

Exceptions:
`RemoteException`

getApplicationDataConnector

```
public ApplicationDataConnector getApplicationDataConnector()
                                                       throws java.rmi.RemoteException
```

Get the `ApplicationDataConnector`for this terminal session. The ApplicationDataConnector provides the interface for sending XML event data from the POS application to the AEFSession.

Returns:

ApplicationDataConnector object for this terminal session

Exceptions:
`RemoteException`

getTerminalNumber

```
public java.lang.String getTerminalNumber()
                                    throws java.rmi.RemoteException
```

Get the terminal number for this terminal session. The terminal number uniquely identifies the session.

(continued from last page)

Returns:

terminal number for this terminal session

Exceptions:
`RemoteException`

isAvailable

```
public boolean isAvailable()
                    throws java.rmi.RemoteException
```

Indicate if this session is available. An AEFSession is *available* if no other client has referenced the session.

Returns:

true if this session is available

Exceptions:
`RemoteException`

See Also:

#reference Referencing a session

com.ibm.retail.AEF.server.SessionServer#getAvailableSession Obtaining an available session

isReserved

```
public boolean isReserved()
                   throws java.rmi.RemoteException
```

Indicate if this session is reserved. An AEFSession that is *reserved* is not considered *available* to clients that attempt to obtain a session through `SessionServer.getAvailableSession()`. Sessions that are reserved can only be obtained through `SessionServer.getSession()`.

Returns:

true if this session is reserved

Exceptions:
`RemoteException`

See Also:

com.ibm.retail.AEF.server.SessionServer#getAvailableSession Obtaining an available session

com.ibm.retail.AEF.server.SessionServer#getSession(String) Obtaining a specific session

isReady

```
public boolean isReady()
                 throws java.rmi.RemoteException
```

Indicate if this session ready for client actions. A sessions must be "ready" before it can process action requests from the automation API.

Returns:

true if this session is ready

Exceptions:

(continued from last page)

`RemoteException`

See Also:

#waitUntilReady waitUntilReady

waitUntilReady

```
public boolean waitUntilReady(int timeout)
                        throws java.rmi.RemoteException
```

Wait until this session is ready for processing. Return true if the session is ready. If not ready , the method will timeout and return false.

Parameters:

`timeout -`
timeout value in milliseconds

Returns:

true if session is ready

Exceptions:

`RemoteException`

See Also:

#isReady isReady

reference

```
public void reference()
               throws java.rmi.RemoteException
```

Increment the client reference count for this session. Referencing a session removes it from the pool of available sessions. Use the `release()`method to return it to the pool of available sessions.

Exceptions:

`RemoteException`

See Also:

#release release

release

```
public void release()
             throws java.rmi.RemoteException,
                    AEFException
```

Decrement the client reference count for this session. When all clients referencing a session have released the session, it is returned to the pool of available sessions.

Exceptions:

`RemoteException`

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.CLIENT_SESSION_MISMATCH

See Also:

#reference() Reference a session

isVirtualSession

```
public boolean isVirtualSession()
                          throws java.rmi.RemoteException
```

Indicates if this a virtual session object. A *virtual session* runs with emulated devices, not in a real POS terminal.

Returns:

true if this session is a virtual session.

Exceptions:
`RemoteException`

isTSSSession

```
public boolean isTSSSession()
                      throws java.rmi.RemoteException
```

Indicates if this a virtual session object. A *virtual session* runs with emulated devices, not in a real POS terminal.

Returns:

true if this session is a virtual session.

Exceptions:
`RemoteException`

addToEventQueue

```
public void addToEventQueue(AEFEventQueueRunnable obj)
                      throws java.rmi.RemoteException
```

Add a runnable object to the sessions event queue. Objects added to the event queue are enqueued on the session's event queue thread and are executed when it reaches the front of the queue.

Parameters:
`obj -`
object to be enqueued

Exceptions:
`RemoteException`

waitForPendingEvents

```
public void waitForPendingEvents()
                           throws java.rmi.RemoteException
```

Wait for events currently pending on the session event queue to complete. Objects added to the event queue are enqueued on the session's event queue thread and are executed when it reaches the front of the queue. This method will block until the current set of pending events have completed.

Exceptions:
`RemoteException`

getWorkstation

```
public Workstation getWorkstation()
                            throws java.rmi.RemoteException
```

Returns the Workstation object for this session.

(continued from last page)

Returns:

Workstation object for this session

Exceptions:
`RemoteException`

isSessionTraceEnabled

```
public boolean isSessionTraceEnabled(int level)
                                     throws java.rmi.RemoteException
```

Indicates if session trace is enabled for this session for a specified log level. Session trace provides an diagnostic in-memory trace of session behavior.

Parameters:
`level -`
level of messages to be logged

Exceptions:
`RemoteException`

sessionTrace

```
public void sessionTrace(int level,
                         java.lang.String msg)
                   throws java.rmi.RemoteException
```

Write to the session trace buffer with the specified level. The message will only be written if the current session trace level is set at or below the given level. `SessionTrace.DUMP = Write an error message`
`SessionTrace.COARSE = Low tracing message`
`SessionTrace.MEDIUM = Medium tracing message`
`SessionTrace.FINE = High tracing message`

Parameters:
`level -`
level of message
`msg -`
message to be written to the session trace buffer

Exceptions:
`RemoteException`

setSessionTraceLevel

```
public void setSessionTraceLevel(int level)
                                 throws java.rmi.RemoteException
```

Set the trace level for the session. This allows you to control the level of trace information written to the session trace buffer. `SessionTrace.OFF = Turn the session trace off`
`SessionTrace.DUMP = Write to trace on error only`
`SessionTrace.COARSE = Low tracing`
`SessionTrace.MEDIUM = Medium tracing`
`SessionTrace.FINE = High tracing`

Parameters:
`level -`
level of message

Exceptions:
`RemoteException`

(continued from last page)

getSessionTraceLevel

```
public int getSessionTraceLevel()
                          throws java.rmi.RemoteException
```

Returns the current trace level for the session.

Returns:

Current session trace level

Exceptions:

`RemoteException`

dumpSessionTrace

```
public void dumpSessionTrace()
                       throws java.rmi.RemoteException
```

Dump the session trace buffer to the trace file. The trace filename pattern is defined in `config.properties`using the property `trace.file.pattern`.

Exceptions:

`RemoteException`

isAutoDumpSessionTrace

```
public boolean isAutoDumpSessionTrace()
                               throws java.rmi.RemoteException
```

Indicates if the session trace buffer auto dumps for AEF exceptions. This property is set in session.properties "trace.autodump".

Returns:

true if trace.autodump is ON, otherwise false.

Exceptions:

`RemoteException`

getTraceBuffer

```
public java.lang.String[] getTraceBuffer()
                                  throws java.rmi.RemoteException
```

Returns a String array of the trace buffer.

Returns:

array of strings containing contents of trace buffer

Exceptions:

`RemoteException`

addSessionTraceSocketHandler

```
public void addSessionTraceSocketHandler(java.lang.String host,
                                         int port)
                                  throws java.rmi.RemoteException,
                                         java.io.IOException
```

Add a socket handler to the session trace. A socket handler allows the session trace buffer to be monitored from a remote client via a socket connection.

(continued from last page)

Parameters:

`host -`
hostname
`port -`
port number

Exceptions:

`RemoteException`

removeSessionTraceSocketHandler

```
public void removeSessionTraceSocketHandler()
                                    throws java.rmi.RemoteException
```

Remove the session trace socket handler.

Exceptions:

`RemoteException`

addSessionStatusListener

```
public void addSessionStatusListener(SessionStatusListener listener)
                                 throws java.rmi.RemoteException
```

Add a listener for SessionStatus events.

Parameters:

`listener -`
SessionStatusListener to be added

Exceptions:

`RemoteException`

removeSessionStatusListener

```
public void removeSessionStatusListener(SessionStatusListener listener)
                                    throws java.rmi.RemoteException
```

Remove a listener for SessionStatus events.

Parameters:

`listener -`
SessionStatusListener to be removed

Exceptions:

`RemoteException`

startApplication

```
public void startApplication()
                      throws java.rmi.RemoteException,
                             AEFException
```

Start the terminal sales application. Note: This method is needed only for sessions where the terminal sales application is not started automatically. This is controlled by the `session.properties`property: `start.pos.sales.application=false`The terminal sales application program name is configured using session.properties: `pos.sales.application=R::h0:/adx_ipgm/EAMTS10L.286`

Exceptions:

`RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.ERROR_STARTING_POS_APPLICATION

stopApplication

```
public void stopApplication()
                      throws java.rmi.RemoteException,
                             AEFException
```

Stop the terminal sales application. **Note:**this method applies only to virtual (CSS) sessions.

Exceptions:

`RemoteException`

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.API_VALID_FOR_TSS_ONLY
AEFConst.ERROR_STOPPING_POS_APPLICATION

terminate

```
public void terminate()
                throws java.rmi.RemoteException,
                       AEFException
```

Cause the session to perform any cleanup required before it is terminated. **Note:**this method applies only to virtual (CSS) sessions.

Exceptions:

`RemoteException`

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.API_VALID_FOR_TSS_ONLY
AEFConst.ERROR_STOPPING_POS_APPLICATION

isApplicationActive

```
public boolean isApplicationActive()
                              throws java.rmi.RemoteException
```

Is the terminal sales application active? **Note:**this method applies only to virtual (CSS) sessions.

Exceptions:

`RemoteException`

getHealthServerAddress

```
public java.net.InetSocketAddress getHealthServerAddress()
                                                    throws java.rmi.RemoteException,
                                                           AEFException
```

Get the address of the AEF Health Server for this session. The HealthServer provides a socket interface for pinging the AEF instance.

Returns:

address of the AEF HealthServer socket

Exceptions:

`RemoteException`

`AEFException`

com.ibm.retail.AEF.session

Interface SessionExtension

public interface **SessionExtension**

A `SessionExtension`provides an interface for extension classes which are instantiated for each AEFSession. The intent is to provide a configurable means by which user server-side code can be configured to run as part of the AEF. Instances of classes which implement this interface will run locally on the machine running the POS application.

To create a server-side session extension, create a class which implements this interface. The class must contain a default constructor, and the `setSession`method. Configure the class to be created in the usersession.properties file by setting session.extensions=packagename.classname where packagename.classname is the package qualified classname of the user extension class. An instance of the user class will be created for each session as the session becomes ready. The default constructor will be called, followed by a call to `setSession`.

Method Summary

`void`	`setSession(AEFSession session)` This method will be called after the session becomes ready.

Methods

setSession

`public void` **`setSession`**`(AEFSession session)`

This method will be called after the session becomes ready.

Parameters:

`session -`
The AEFSession associated with this extension object.

com.ibm.retail.AEF.session

Class SessionParameters

```
java.lang.Object
  +-java.util.AbstractMap
    +-java.util.HashMap
      +-com.ibm.retail.AEF.session.SessionParameters
```

public class **SessionParameters**
extends java.util.HashMap

A `SessionParameters`object is a `HashMap`that contains the input parameters used to construct an `AEFSession`.

Any session parameter contained in `session.properties`may be overridden by using the `put`method to add the property and value to the `SessionParameters`object. Use the `getSession`or `getAvailableSession`method of `SessionServer`to pass the SessionParameters in for session creation.

See Also:
com.ibm.retail.AEF.server.SessionServer#getAvailableSession getAvailableSession,

com.ibm.retail.AEF.server.SessionServer#getSession getSession

Constructor Summary

`SessionParameters()` Constructor

Method Summary

`java.lang.String`	`getClientID()` Get the client ID associated with the session.
`java.lang.String`	`getDeviceServerHost()` Get the hostname used for remote device access.
`java.lang.String`	`getTimeout()` Get the timeout value used for session creation.
`void`	`setClientID(java.lang.String clientID)` Set the client ID which will be associated with the session.
`void`	`setDeviceServerHost(java.lang.String host)` Set the device server host value used for remote device access.
`void`	`setTimeout(int timeout)` Set the timeout value used for session creation.
`java.lang.String`	`toString()` Convert to a string representation

Methods inherited from : class `java.util.HashMap`
`clear, clone, containsKey, containsValue, entrySet, get, isEmpty, keySet, put, putAll, remove, size, values`

Methods inherited from : class `java.util.AbstractMap`
`clear, clone, containsKey, containsValue, entrySet, equals, get, hashCode, isEmpty, keySet, put, putAll, remove, size, toString, values`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

SessionParameters

`public` **`SessionParameters`**`()`

Constructor

Methods

setTimeout

`public void` **`setTimeout`**`(int timeout)`

Set the timeout value used for session creation. This value controls the time to wait for session to be ready for input. A value of -1 will return the session without waiting for a ready event. This method is equivalent to:
`parms.put("ready.wait.timeout",String.valueOf(timeout));`

Parameters:

`timeout` -
timeout value.

getTimeout

`public java.lang.String` **`getTimeout`**`()`

Get the timeout value used for session creation. This method is equivalent to:
`parms.get("ready.wait.timeout");`

Returns:

String

setClientID

`public void` **`setClientID`**`(java.lang.String clientID)`

Set the client ID which will be associated with the session. This value will be logged by the AEF when the session is returned to the client. This value is used for informational purposes only. This method is equivalent to:
`parms.put("client.ID",clientID);`

(continued from last page)

Parameters:

`clientID -`
The client ID.

getClientID

`public java.lang.String` **`getClientID`**`()`

Get the client ID associated with the session. This method is equivalent to: `parms.get("client.ID");`

Returns:

String

setDeviceServerHost

`public void` **`setDeviceServerHost`**`(java.lang.String host)`

Set the device server host value used for remote device access. This method is equivalent to:
`parms.put("device.server.host",host);`

Parameters:

`host -`
host name or ip address of DeviceServer

getDeviceServerHost

`public java.lang.String` **`getDeviceServerHost`**`()`

Get the hostname used for remote device access. This method is equivalent to:
`parms.get("device.server.host");`

Returns:

host name or ip address

toString

`public java.lang.String` **`toString`**`()`

Convert to a string representation

Returns:

parameters in string form

com.ibm.retail.AEF.session

Class SessionStatusEvent

```
java.lang.Object
   |
   +-com.ibm.retail.AEF.session.SessionStatusEvent
```

All Implemented interfaces:
java.io.Serializable

public class **SessionStatusEvent**
extends java.lang.Object
implements java.io.Serializable

A `SessionStatusEvent`is generated when the status of an `AEFSession`changes. Status changes occur during session initialization and when the session ends.

Field Summary

static java.lang.String	CREATING_AUTOMATION
static java.lang.String	CREATING_DATA_PROVIDER
static java.lang.String	CREATING_DEVICE_MANAGER
static java.lang.String	CREATING_DEVICE_REGISTRY
static java.lang.String	CREATING_HOOKS
static java.lang.String	CREATING_JIOP
static java.lang.String	CREATING_SESSION
java.lang.String	description
static java.lang.String	LOADING_DEVICE_HANDLERS
int	percentComplete
static java.lang.String	SESSION_JIOP_READY_DESC
static java.lang.String	SESSION_READY_DESC
static java.lang.String	SESSION_SHUTDOWN_DESC
java.lang.String	terminalNumber

Constructor Summary

`SessionStatusEvent()` Constructor
`SessionStatusEvent(java.lang.String term,java.lang.String term,int term)` Constructor.

Method Summary

`java.lang.String`	`getDescription()` Returns the description of the status event.
`int`	`getPercentComplete()` Returns the percentage of session initialazation that is complete.
`java.lang.String`	`getTerminalNumber()` Returns the session terminal number.
`void`	`setDescription(java.lang.String desc)` Set the description of the status event.
`void`	`setPercentComplete(int percent)` Set the percent complete.
`void`	`setTerminalNumber(java.lang.String term)` Sets the session terminal number.

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

SESSION_READY_DESC

`public static java.lang.String` **`SESSION_READY_DESC`**

SESSION_JIOP_READY_DESC

`public static java.lang.String` **`SESSION_JIOP_READY_DESC`**

SESSION_SHUTDOWN_DESC

`public static java.lang.String` **`SESSION_SHUTDOWN_DESC`**

(continued from last page)

CREATING_SESSION

public static java.lang.String **CREATING_SESSION**

CREATING_DATA_PROVIDER

public static java.lang.String **CREATING_DATA_PROVIDER**

CREATING_AUTOMATION

public static java.lang.String **CREATING_AUTOMATION**

CREATING_DEVICE_MANAGER

public static java.lang.String **CREATING_DEVICE_MANAGER**

CREATING_HOOKS

public static java.lang.String **CREATING_HOOKS**

LOADING_DEVICE_HANDLERS

public static java.lang.String **LOADING_DEVICE_HANDLERS**

CREATING_JIOP

public static java.lang.String **CREATING_JIOP**

CREATING_DEVICE_REGISTRY

public static java.lang.String **CREATING_DEVICE_REGISTRY**

terminalNumber

protected java.lang.String **terminalNumber**

percentComplete

protected int **percentComplete**

description

protected java.lang.String **description**

Constructors

SessionStatusEvent

```
public SessionStatusEvent()
```

Constructor

SessionStatusEvent

```
public SessionStatusEvent(java.lang.String term,
                          java.lang.String desc,
                          int percent)
```

Constructor.

Parameters:

`term` -
terminal number
`desc` -
description of the event
`percent` -
percent of session initialization complete

Methods

getTerminalNumber

```
public java.lang.String getTerminalNumber()
```

Returns the session terminal number.

Returns:

terminal number

getPercentComplete

```
public int getPercentComplete()
```

Returns the percentage of session initialazation that is complete.

Returns:

percentage of session initialization that has completed.

getDescription

```
public java.lang.String getDescription()
```

Returns the description of the status event.

Returns:

description of the event

setTerminalNumber

```
public void setTerminalNumber(java.lang.String term)
```

Sets the session terminal number.

(continued from last page)

Parameters:

`term -`
terminal number

setPercentComplete

`public void` **`setPercentComplete`**`(int percent)`

Set the percent complete.

Parameters:

`percent -`
percentage complete of session initialization

setDescription

`public void` **`setDescription`**`(java.lang.String desc)`

Set the description of the status event.

Parameters:

`desc -`
description of the event

com.ibm.retail.AEF.session

Interface SessionStatusListener

All Known Implementing Classes:
SessionStatusListenerProxy

public interface **SessionStatusListener**
extends java.rmi.Remote

A `SessionStatusListener`provides a listener interface for `AEFSession SessionStatusEvents`. `SessionStatusEvents` inform listeners of the key events during session initialization and allow a client to determine when the session is ready for commands.

The `SessionStatusEvent`includes an indication of the "percent complete" of the initialization process.

See Also:
com.ibm.retail.AEF.session.AEFSession#addSessionStatusListener addSessionStatusListener

Method Summary

void	`sessionEnded(SessionStatusEvent evt)` Indicates that the session has ended.
void	`sessionReady(SessionStatusEvent evt)` Indicates that the session is ready for commands.
void	`sessionStatusChanged(SessionStatusEvent evt)` Indicates a session status event has occurred.

Methods

sessionStatusChanged

```
public void sessionStatusChanged(SessionStatusEvent evt)
                          throws java.rmi.RemoteException
```

Indicates a session status event has occurred.

Parameters:
`evt -`
contains details of SessionStatusEvent

Exceptions:
`RemoteException`

sessionReady

```
public void sessionReady(SessionStatusEvent evt)
                  throws java.rmi.RemoteException
```

Indicates that the session is ready for commands.

Parameters:

(continued from last page)

`evt` -
contains details of SessionStatusEvent

Exceptions:
`RemoteException`

sessionEnded

```
public void sessionEnded(SessionStatusEvent evt)
                  throws java.rmi.RemoteException
```

Indicates that the session has ended.

Parameters:
`evt` -
contains details of SessionStatusEvent

Exceptions:
`RemoteException`

Package
com.ibm.retail.AEF.thread

Provides thread pooling and locking implementations used by the AEF.

com.ibm.retail.AEF.thread

Class ConditionLock

```
java.lang.Object
   |
   +-com.ibm.retail.AEF.thread.ConditionLock
```

Direct Known Subclasses:
ObjectDetectorLock

public class **ConditionLock**
extends java.lang.Object

An synchronization mechanism which can use an external lock.

Field Summary

com.ibm.retail.AEF.action.AEFAction	action
com.ibm.retail.AEF.thread.MutexLock	lock
com.ibm.retail.AEF.util.AEFMessage	msg
java.lang.String	name
int	returnIndex
com.ibm.retail.AEF.session.AEFSession	session
int	timeout
com.ibm.retail.si.util.AEFException	unblockException
java.lang.RuntimeException	unblockRuntimeException

Constructor Summary

`ConditionLock()`

Constructs a ConditionLock with an internal lock object.

Method Summary

MutexLock	`getLock()` Gets the lock object for this condition variable.
java.lang.Thread	`getLockOwner()` Gets the thread which owns the lock object for this condition variable.

void	`notify(AEFException e)` Unblock a thread waiting on this condition variable by throwing an exception on the blocked thread.
void	`notify(int index)` Notify any threads waiting on this condition variable to execute.
void	`notify(java.lang.RuntimeException e)` Unblock a thread waiting on this condition variable by throwing an exception on the blocked thread.
int	`performActionAndWait(java.lang.String name,AEFAction name,Condition[] name,Condition[] name,int name)` Blocks the calling thread on the condition variable or a timeout
int	`performActionAndWait(java.lang.String name,AEFAction name,Condition[] name,Condition name,int name)` Blocks the calling thread on the condition variable or a timeout
int	`performActionAndWait(java.lang.String name,AEFAction name,Condition[] name,int name)` Blocks the calling thread on the condition variable or a timeout
int	`performActionAndWait(java.lang.String name,AEFAction name,Condition name,int name)` Blocks the calling thread on the condition variable or a timeout
int	`wait(java.lang.String name,Condition[] name,Condition[] name,int name)` Blocks the calling thread on the condition variable or a timeout
int	`wait(java.lang.String name,Condition[] name,Condition[] name,int name,boolean name)` Blocks the calling thread on the condition variable or a timeout
int	`wait(java.lang.String name,Condition[] name,Condition name,int name)` Blocks the calling thread on the condition variable or a timeout
int	`wait(java.lang.String name,Condition[] name,Condition name,int name,boolean name)` Blocks the calling thread on the condition variable or a timeout
int	`wait(java.lang.String name,Condition[] name,int name)` Blocks the calling thread on the condition variable or a timeout
int	`wait(java.lang.String name,Condition[] name,int name,boolean name)` Blocks the calling thread on the condition variable or a timeout
int	`wait(java.lang.String name,Condition name,int name)` Blocks the calling thread on the condition variable or a timeout.
int	`wait(java.lang.String name,Condition name,int name,boolean name)` Blocks the calling thread on the condition variable or a timeout

Methods inherited from : class `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
wait
```

Fields

lock

protected com.ibm.retail.AEF.thread.MutexLock **lock**

action

protected com.ibm.retail.AEF.action.AEFAction **action**

timeout

protected int **timeout**

unblockException

protected com.ibm.retail.si.util.AEFException **unblockException**

unblockRuntimeException

protected java.lang.RuntimeException **unblockRuntimeException**

returnIndex

protected int **returnIndex**

name

protected java.lang.String **name**

msg

protected com.ibm.retail.AEF.util.AEFMessage **msg**

session

protected com.ibm.retail.AEF.session.AEFSession **session**

(continued from last page)

Constructors

ConditionLock

```
public ConditionLock()
```

Constructs a ConditionLock with an internal lock object.

Methods

performActionAndWait

```
public int performActionAndWait(java.lang.String name,
                                AEFAction action,
                                Condition condition,
                                int timeout)
                         throws AEFException
```

Blocks the calling thread on the condition variable or a timeout

Parameters:

`name -`
A debug aid. Give a unique name to the condition lock such as "wait-for-password-state".
`action -`
The action which will be performed which should cause some conditions to change.
`condition -`
The condition to wait for before returning.
`timeout -`
The number of milliseconds to wait for the condition to be true.

Returns:

int The zero based index of the condition that evaluated to true.

Exceptions:

`AEFException -`

Because of the nature of this method just about any error code could be returned. Among the possible AEFException error codes are:
Common Errors

performActionAndWait

```
public int performActionAndWait(java.lang.String name,
                                AEFAction action,
                                Condition[] conditions,
                                int timeout)
                         throws AEFException
```

Blocks the calling thread on the condition variable or a timeout

Parameters:

`name -`
A debug aid. Give a unique name to the condition lock such as "wait-for-password-state".
`action -`
The action which will be performed which should cause some conditions to change.
`conditions -`
An array of conditions, one of which must evaluate to true before returning.
`timeout -`
The number of milliseconds to wait for the condition to be true.

Returns:

int The zero based index of the condition that evaluated to true.

(continued from last page)

Exceptions:

`AEFException -`

Because of the nature of this method just about any error code could be returned. Among the possible AEFException error codes are:
Common Errors

performActionAndWait

```
public int performActionAndWait(java.lang.String name,
                                AEFAction action,
                                Condition[] conditions,
                                Condition errorCondition,
                                int timeout)
                         throws AEFException
```

Blocks the calling thread on the condition variable or a timeout

Parameters:

`name -`
A debug aid. Give a unique name to the condition lock such as "wait-for-password-state".
`action -`
The action which will be performed which should cause some conditions to change.
`conditions -`
An array of conditions, one of which must evaluate to true before returning.
`errorCondition -`
A condition which will cause the calling thread to unblock if it evaluates to true.
`timeout -`
The number of milliseconds to wait for the condition to be true.

Returns:

int The index of the condition that evaluated to true. If zero or positive, the return value represents the zero based index into the "normal" condition array. If negative one (-1) , the error condition evaluated to true.

Exceptions:

`AEFException -`

Because of the nature of this method just about any error code could be returned. Among the possible AEFException error codes are:
Common Errors

performActionAndWait

```
public int performActionAndWait(java.lang.String name,
                                AEFAction action,
                                Condition[] conditions,
                                Condition[] errorConditions,
                                int timeout)
                         throws AEFException
```

Blocks the calling thread on the condition variable or a timeout

Parameters:

`name -`
A debug aid. Give a unique name to the condition lock such as "wait-for-password-state".
`action -`
The action which will be performed which should cause some conditions to change.
`conditions -`
An array of conditions, one of which must evaluate to true before returning.
`errorConditions -`
An array of conditions which will cause the calling thread to unblock if one of the conditions evaluates to true. These conditions are considered "error" conditions.
`timeout -`
The number of milliseconds to wait for the condition to be true.

(continued from last page)

Returns:

int The index of the condition that evaluated to true. If zero or positive, the return value represents the zero based index into the "normal" condition array. If negative, the return value is the negation of the zero based index into the "error" condition array plus 1. To convert a negative return value back into a zero based index into the "error" condition array, take the absolute value, then subtract 1.

Exceptions:

`AEFException -`

Because of the nature of this method just about any error code could be returned. Among the possible AEFException error codes are:
Common Errors

---

wait

```
public int wait(java.lang.String name,
                Condition condition,
                int timeout)
          throws AEFException
```

Blocks the calling thread on the condition variable or a timeout. The property monitored in the condition must change before the condition is evaluated.

Parameters:

`name -`
A debug aid. Give a unique name to the condition lock such as "wait-for-password-state".
`condition -`
The condition must evaluate to true before returning.
`timeout -`
The number of milliseconds to wait for the condition to be true.

Returns:

int The zero based index of the condition that evaluated to true. Will always be zero in this case.

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
Common Errors

---

wait

```
public int wait(java.lang.String name,
                Condition condition,
                int timeout,
                boolean immediateEval)
          throws AEFException
```

Blocks the calling thread on the condition variable or a timeout

Parameters:

`name -`
A debug aid. Give a unique name to the condition lock such as "wait-for-password-state".
`condition -`
The condition must evaluate to true before returning.
`timeout -`
The number of milliseconds to wait for the condition to be true.
`immediateEval -`
If true, the conditions are evaluated immediately without first waiting for a property change.

Returns:

int The zero based index of the condition that evaluated to true.

Exceptions:

(continued from last page)

`AEFException -`
Among the possible AEFException error codes are:
Common Errors

wait

```
public int wait(java.lang.String name,
                Condition[] conditions,
                int timeout)
         throws AEFException
```

Blocks the calling thread on the condition variable or a timeout

Parameters:

`name -`
A debug aid. Give a unique name to the condition lock such as "wait-for-password-state".
`conditions -`
An array of conditions, one of which must evaluate to true before returning.
`timeout -`
The number of milliseconds to wait for the condition to be true.

Returns:

int The zero based index of the condition that evaluated to true.

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
Common Errors

wait

```
public int wait(java.lang.String name,
                Condition[] conditions,
                int timeout,
                boolean immediateEval)
         throws AEFException
```

Blocks the calling thread on the condition variable or a timeout

Parameters:

`name -`
A debug aid. Give a unique name to the condition lock such as "wait-for-password-state".
`conditions -`
An array of conditions, one of which must evaluate to true before returning.
`timeout -`
The number of milliseconds to wait for the condition to be true.
`immediateEval -`
If true, the conditions are evaluated immediately without first waiting for a property change.

Returns:

int The zero based index of the condition that evaluated to true.

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
Common Errors

(continued from last page)

wait

```
public int wait(java.lang.String name,
                Condition[] conditions,
                Condition[] errorConditions,
                int timeout)
          throws AEFException
```

Blocks the calling thread on the condition variable or a timeout

Parameters:

`name -`
A debug aid. Give a unique name to the condition lock such as "wait-for-password-state".
`conditions -`
An array of conditions which will cause the calling thread to unblock if one of the conditions evaluates to true. These conditions are considered "non-error" conditions.
`errorConditions -`
An array of conditions which will cause the calling thread to unblock if one of the conditions evaluates to true. These conditions are considered "error" conditions.
`timeout -`
The number of milliseconds to wait for the condition to be true.

Returns:

int The index of the condition that evaluated to true. If zero or positive, the return value represents the zero based index into the "normal" condition array. If negative, the return value is the negation of the zero based index into the "error" condition array plus 1. To convert a negative return value back into a zero based index into the "error" condition array, take the absolute value, then subtract 1.

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
Common Errors

---

wait

```
public int wait(java.lang.String name,
                Condition[] conditions,
                Condition errorCondition,
                int timeout)
          throws AEFException
```

Blocks the calling thread on the condition variable or a timeout

Parameters:

`name -`
A debug aid. Give a unique name to the condition lock such as "wait-for-password-state".
`conditions -`
An array of conditions which will cause the calling thread to unblock if one of the conditions evaluates to true. These conditions are considered "non-error" conditions.
`errorCondition -`
A condition which will cause the calling thread to unblock if it evaluates to true. This condition is considered and "error" condition.
`timeout -`
The number of milliseconds to wait for the condition to be true.

Returns:

int The index of the condition that evaluated to true. If zero or positive, the return value represents the zero based index into the "normal" condition array. If negative, the return value is the negation of the zero based index into the "error" condition array plus 1. To convert a negative return value back into a zero based index into the "error" condition array, take the absolute value, then subtract 1.

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
Common Errors

wait

```
public int wait(java.lang.String name,
                Condition[] conditions,
                Condition errorCondition,
                int timeout,
                boolean immediateEval)
          throws AEFException
```

Blocks the calling thread on the condition variable or a timeout

Parameters:

`name -`
A debug aid. Give a unique name to the condition lock such as "wait-for-password-state".
`conditions -`
An array of conditions which will cause the calling thread to unblock if one of the conditions evaluates to true. These conditions are considered "non-error" conditions.
`errorCondition -`
A condition which will cause the calling thread to unblock if it evaluates to true. This condition is considered and "error" condition.
`timeout -`
The number of milliseconds to wait for the condition to be true.
`immediateEval -`
If true, the conditions are evaluated immediately without first waiting for a property change.

Returns:

int The index of the condition that evaluated to true. If zero or positive, the return value represents the zero based index into the "normal" condition array. If negative, the return value is the negation of the zero based index into the "error" condition array plus 1. To convert a negative return value back into a zero based index into the "error" condition array, take the absolute value, then subtract 1.

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
Common Errors

wait

```
public int wait(java.lang.String name,
                Condition[] conditions,
                Condition[] errorConditions,
                int timeout,
                boolean immediateEval)
          throws AEFException
```

Blocks the calling thread on the condition variable or a timeout

Parameters:

`name -`
A debug aid. Give a unique name to the condition lock such as "wait-for-password-state".
`conditions -`
An array of conditions which will cause the calling thread to unblock if one of the conditions evaluates to true. These conditions are considered "non-error" conditions.
`errorConditions -`
An array of conditions which will cause the calling thread to unblock if one of the conditions evaluates to true. These conditions are considered "error" conditions.
`timeout -`
The number of milliseconds to wait for the condition to be true.
`immediateEval -`
If true, the conditions are evaluated immediately without first waiting for a property change.

Returns:

(continued from last page)

int The index of the condition that evaluated to true. If zero or positive, the return value represents the zero based index into the "normal" condition array. If negative, the return value is the negation of the zero based index into the "error" condition array plus 1. To convert a negative return value back into a zero based index into the "error" condition array, take the absolute value, then subtract 1.

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
Common Errors

notify

`public void **notify**(AEFException e)`

Unblock a thread waiting on this condition variable by throwing an exception on the blocked thread.

Parameters:

`e -`
The exception to throw on the blocked thread.

notify

`public void **notify**(java.lang.RuntimeException e)`

Unblock a thread waiting on this condition variable by throwing an exception on the blocked thread.

Parameters:

`e -`
The exception to throw on the blocked thread.

notify

`public void **notify**(int index)`

Notify any threads waiting on this condition variable to execute.

Parameters:

`sv -`
An external lock to signal.

getLock

`public MutexLock **getLock**()`

Gets the lock object for this condition variable.

Returns:

MutexLock

getLockOwner

`public java.lang.Thread **getLockOwner**()`

Gets the thread which owns the lock object for this condition variable.

Returns:

Thread

com.ibm.retail.AEF.thread

Class ObjectDetectorLock

```
java.lang.Object
  |-com.ibm.retail.AEF.thread.ConditionLock
    |-com.ibm.retail.AEF.thread.ObjectDetectorLock
```

All Implemented interfaces:
ObjectDetectorListener

---

public class **ObjectDetectorLock**
extends ConditionLock
implements ObjectDetectorListener

An synchronization mechanism for use with ObjectDetectors.

Field Summary

com.ibm.retail.AEF.automation.ObjectDetector	detector
com.ibm.retail.AEF.util.AEFMessage	msg
boolean	objectDetected
com.ibm.retail.AEF.session.AEFSession	session
com.ibm.retail.si.util.AEFException	unblockException

Fields inherited from : class com.ibm.retail.AEF.thread.ConditionLock
action, lock, msg, name, returnIndex, session, timeout, unblockException, unblockRuntimeException

Constructor Summary

ObjectDetectorLock() Constructs a ObjectDetectorLock.

Method Summary

void	exceptionDetected(AEFException e) Called by the ObjectDetector when it throws an AEFException.
void	objectDetected() Called by the ObjectDetector when an object has been created.

java.lang.Object	waitForNewObject(java.lang.String name,ObjectDetector name,int name,int name) Blocks the calling thread on the condition variable or a timeout
java.lang.Object	waitForNewObject(java.lang.String name,ObjectDetector name,int name,int name,boolean name) Blocks the calling thread on the condition variable or a timeout
java.lang.Object	waitForNewObjectOrError(java.lang.String name,ObjectDetector name,int name,Condition[] name,boolean name,int name) Blocks the calling thread on the condition variable or a timeout
java.lang.Object	waitForNewObjectOrError(java.lang.String name,ObjectDetector name,int name,Condition[] name,boolean name,int name,boolean name) Blocks the calling thread on the condition variable or a timeout

Methods inherited from : class com.ibm.retail.AEF.thread.ConditionLock
getLock, getLockOwner, notify, notify, notify, performActionAndWait, performActionAndWait, performActionAndWait, performActionAndWait, wait, wait, wait, wait, wait, wait, wait, wait

Methods inherited from : class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

objectDetected

protected boolean **objectDetected**

detector

protected com.ibm.retail.AEF.automation.ObjectDetector **detector**

unblockException

protected com.ibm.retail.si.util.AEFException **unblockException**

msg

protected com.ibm.retail.AEF.util.AEFMessage **msg**

session

protected com.ibm.retail.AEF.session.AEFSession **session**

Constructors

ObjectDetectorLock

```
public ObjectDetectorLock()
```

Constructs a ObjectDetectorLock.

Methods

waitForNewObject

```
public java.lang.Object waitForNewObject(java.lang.String name,
                                         ObjectDetector detector,
                                         int oldInstanceNumber,
                                         int timeout)
                                  throws AEFException
```

Blocks the calling thread on the condition variable or a timeout

Parameters:

`name -`
A debug aid. Give a unique name to the detector lock such as "wait-for-operator".
`detector -`
If this detector senses an object, the calling thread will be unblocked.
`oldInstanceNumber -`
Each object detected by the detector has an instance number. The caller should pass in the prior instance number from the detector so that if the new object has been detected prior to entry into this routine, it can be returned without waiting.
`timeout -`
The number of milliseconds to wait for the object.

Exceptions:

`AEFException -`
Because of the nature of this method, just about any error code could be returned.
Among the possible AEFException error codes are:
Common Errors

---

waitForNewObject

```
public java.lang.Object waitForNewObject(java.lang.String name,
                                         ObjectDetector detector,
                                         int oldInstanceNumber,
                                         int timeout,
                                         boolean ignoreTimeout)
                                  throws AEFException
```

Blocks the calling thread on the condition variable or a timeout

Parameters:

`name -`
A debug aid. Give a unique name to the detector lock such as "wait-for-operator".
`detector -`
If this detector senses an object, the calling thread will be unblocked.
`oldInstanceNumber -`
Each object detected by the detector has an instance number. The caller should pass in the prior instance number from the detector so that if the new object has been detected prior to entry into this routine, it can be returned without waiting.
`timeout -`
The number of milliseconds to wait for the object.
`ignoreTimeout -`
If true, do not log error message if timeout situation occurs. In some conditions (e.g., in the ACEApplyDelayedCoupons.wait ForCoupons() method), the timeout is a nominal value since there may or may not be any object created

(continued from last page)

Exceptions:

`AEFException -`
Because of the nature of this method, just about any error code could be returned.
Among the possible AEFException error codes are:
Common Errors

waitForNewObjectOrError

```
public java.lang.Object waitForNewObjectOrError(java.lang.String name,
                                                ObjectDetector detector,
                                                int oldInstanceNumber,
                                                Condition[] errorConditions,
                                                boolean immediateEval,
                                                int timeout)
                                         throws AEFException
```

Blocks the calling thread on the condition variable or a timeout

Parameters:

`name -`
A debug aid. Give a unique name to the detector lock such as "wait-for-operator".
`detector -`
If this detector senses an object, the calling thread will be unblocked.
`oldInstanceNumber -`
Each object detected by the detector has an instance number. The caller should pass in the prior instance number from the detector so that if the new object has been detected prior to entry into this routine, it can be returned without waiting.
`errorConditions -`
An array of conditions which will cause the calling thread to unblock if one of the conditions evaluates to true. These conditions are considered "error" conditions.
`immediateEval -`
If true, the conditions are evaluated immediately without first waiting for a property change.
`timeout -`
The number of milliseconds to wait for the object.

Returns:

Object If the object is an Integer, it is the index of the condition that ocurred. Otherwise, the object is the object which was detected.

Exceptions:

`AEFException -`
Because of the nature of this method, just about any error code could be returned.
Among the possible AEFException error codes are:
Common Errors

waitForNewObjectOrError

```
public java.lang.Object waitForNewObjectOrError(java.lang.String name,
                                                ObjectDetector detector,
                                                int oldInstanceNumber,
                                                Condition[] errorConditions,
                                                boolean immediateEval,
                                                int timeout,
                                                boolean ignoreTimeout)
                                         throws AEFException
```

Blocks the calling thread on the condition variable or a timeout

Parameters:

`name -`
A debug aid. Give a unique name to the detector lock such as "wait-for-operator".
`detector -`
If this detector senses an object, the calling thread will be unblocked.

(continued from last page)

`oldInstanceNumber -`
Each object detected by the detector has an instance number. The caller should pass in the prior instance number from the detector so that if the new object has been detected prior to entry into this routine, it can be returned without waiting.
`errorConditions -`
An array of conditions which will cause the calling thread to unblock if one of the conditions evaluates to true. These conditions are considered "error" conditions.
`immediateEval -`
If true, the conditions are evaluated immediately without first waiting for a property change.
`timeout -`
The number of milliseconds to wait for the object.
`ignoreTimeout -`
If true, do not log error message if timeout situation occurs. In some conditions (e.g., in the ACEApplyDelayedCoupons.wait ForCoupons() method), the timeout is a nominal value since there may or may not be any object created

Returns:

Object If the object is an Integer, it is the index of the condition that ocurred. Otherwise, the object is the object which was detected.

Exceptions:
`AEFException -`
Because of the nature of this method, just about any error code could be returned.
Among the possible AEFException error codes are:
Common Errors

objectDetected

`public void` **`objectDetected`**`()`

Called by the ObjectDetector when an object has been created.

exceptionDetected

`public void` **`exceptionDetected`**`(AEFException e)`

Called by the ObjectDetector when it throws an AEFException.

Package
com.ibm.retail.AEF.util

Provides an assortment of utility classes used by the AEF.

com.ibm.retail.AEF.util

Class AEFErrorHandler

```
java.lang.Object
   |
   +-com.ibm.retail.AEF.util.AEFErrorHandler
```

public class **AEFErrorHandler**
extends java.lang.Object

ErrorHandler is a class which the AEF applications can use to handle application errors.

Field Summary

java.lang.Object	additionalData
com.ibm.retail.si.util.AEFException	allExceptions
java.lang.String	applicationErrorText
java.util.ArrayList	errorHistory
boolean	ignoreMaxErrorLevels
static org.apache.commons.logging.Log	log
int	maxErrorLevels
int	maxSameError
com.ibm.retail.AEF.util.AEFMessage	msg
int	numSameError
int	sleepInterval

Constructor Summary

AEFErrorHandler(int errorLevels,java.lang.String errorLevels) Constructor
AEFErrorHandler(int errorLevels,java.lang.String errorLevels,java.lang.Object errorLevels) Constructor
AEFErrorHandler(java.lang.String errorText,java.lang.Object errorText) Constructor

`AEFErrorHandler(java.lang.String errorText)` Constructor

Method Summary

`java.lang.Object`	`getAdditionalData()` Gets the additional data pased by the calling application or a NULL.
`AEFException`	`getAllExceptions()` Returns the allExceptions variable which contains the exceptions encountered while trying to handle the error.
`java.lang.String`	`getAllFormattedErrorText(boolean includeApplicationText)` Returns a string which is the formatted error information from all the error handled.
`int`	`getErrorHandlingMode()` Gets the error handling mode from the Automation Provider.
`AEFError[]`	`getErrors()` Returns the array of all the errors handled so far.
`java.lang.String`	`getFormattedErrorText(int index,boolean index)` Returns a string which is the formatted error information for a specific error.
`int`	`getSleepInterval()` Returns the number of milliseconds to wait before continuing with the next operation while handling errors.
`int`	`handleError(ConditionLock lock,AEFAction lock,Condition[] lock,Condition[] lock,int lock,int lock)` Returns the index of the condition that satisfied the wait.
`int`	`handleSpecificError(ConditionLock lock,AEFAction lock,Condition[] lock,Condition[] lock,int lock,int lock,int lock)` Returns the index of the condition that satisfied the wait.
`void`	`sleep()` Causes the Error Helper to wait before continuing with the next operation.
`void`	`sleep(int userInterval)` Causes the Error Helper to wait before continuing with the next operation.

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

maxSameError

`protected int` **`maxSameError`**

(continued from last page)

numSameError

protected int **numSameError**

maxErrorLevels

protected int **maxErrorLevels**

ignoreMaxErrorLevels

protected boolean **ignoreMaxErrorLevels**

applicationErrorText

protected java.lang.String **applicationErrorText**

allExceptions

protected com.ibm.retail.si.util.AEFException **allExceptions**

errorHistory

protected java.util.ArrayList **errorHistory**

sleepInterval

protected int **sleepInterval**

additionalData

protected java.lang.Object **additionalData**

log

protected static org.apache.commons.logging.Log **log**

msg

protected com.ibm.retail.AEF.util.AEFMessage **msg**

Constructors

(continued from last page)

AEFErrorHandler

```
public AEFErrorHandler(int errorLevels,
                       java.lang.String errorText)
```

Constructor

Parameters:

`errorLevels -`
Determines the maximum number of errors that the handler will try to clear. 0 means to continue to try to correct the error forever.
`errorText -`
Includes any additional text that the application wants to include in the exception if any is thrown by the handler.

AEFErrorHandler

```
public AEFErrorHandler(int errorLevels,
                       java.lang.String errorText,
                       java.lang.Object data)
```

Constructor

Parameters:

`errorLevels -`
Determines the maximum number of errors that the handler will try to clear. 0 means to continue to try to correct the error forever.
`errorText -`
Includes any additional text that the application wants to include in the exception if any is thrown by the handler.
`data -`
Includes any additional data that the Error Handler or Helpers may need.

AEFErrorHandler

```
public AEFErrorHandler(java.lang.String errorText,
                       java.lang.Object data)
```

Constructor

Parameters:

`errorText -`
Includes any additional text that the application wants to include in the exception if any is thrown by the handler.
`data -`
Includes any additional data that the Error Handler or Helpers may need.

AEFErrorHandler

```
public AEFErrorHandler(java.lang.String errorText)
```

Constructor

Parameters:

`errorText -`
Includes any additional text that the application wants to include in the exception if any is thrown by the handler.

Methods

getErrors

```
public AEFError[] getErrors()
```

Returns the array of all the errors handled so far.

Returns:

(continued from last page)

AEFError[] The array of all the errors handled so far.

getAllFormattedErrorText

```
public java.lang.String getAllFormattedErrorText(boolean includeApplicationText)
```

Returns a string which is the formatted error information from all the error handled.

Parameters:

`boolean -`
A flag used to indicate if the application text should be included.

Returns:

String The additional error text.

getFormattedErrorText

```
public java.lang.String getFormattedErrorText(int index,
                                              boolean includeApplicationText)
```

Returns a string which is the formatted error information for a specific error.

Parameters:

`int -`
The index of the error to return the text on.
`boolean -`
A flag used to indicate if the application text should be included.

Returns:

String The formatted error text.

handleError

```
public int handleError(ConditionLock lock,
                       AEFAction keySequenceAction,
                       Condition[] goodConditions,
                       Condition[] badConditions,
                       int timeout,
                       int satisfiedCondition)
                 throws AEFException
```

Returns the index of the condition that satisfied the wait.

Parameters:

`lock -`
The ConditionLock that returned the error.
`keySequenceAction -`
The key sequence that generated the error.
`goodConditions -`
An array of the conditions that would indicate success after the key sequence is sent.
`badConditions -`
An array of the conditions that would indicate a failure after the key sequence is sent. For the most part, these should be states and not individual errors. The assumption is that most errors would put the application in some sort of error state. The handler would then figure out which error put the application in this state and handle the error.
`timeout -`
An integer used to determine how long to wait for one of the listed conditions before a timeout exception is thrown.
`satisfiedCondition -`
The index of the condition in the badConditions array that was satisfied (needs to be adjusted as it is a negative number).

Returns:

int The index of the condition in the goodConditions array that was satisfied after the error was handled.

(continued from last page)

Exceptions:

`AEFException -`
Because of the nature of this method, just about any error code could be returned.
Among the possible AEFException error codes are:
Common Errors

handleSpecificError

```
public int handleSpecificError(ConditionLock lock,
                               AEFAction keySequenceAction,
                               Condition[] goodConditions,
                               Condition[] badConditions,
                               int timeout,
                               int satisfiedCondition,
                               int errorLevel)
                        throws AEFException
```

Returns the index of the condition that satisfied the wait.

Parameters:

`lock -`
The ConditionLock that returned the error.
`keySequenceAction -`
The key sequence that generated the error.
`goodConditions -`
An array of the conditions that would indicate success after the key sequence is sent.
`badConditions -`
An array of the conditions that would indicate a failure after the key sequence is sent.
`timeout -`
An integer used to determine how long to wait for one of the listed conditions before a timeout exception is thrown.
`satisfiedCondition -`
The index of the condition in the badConditions array that was satisfied (needs to be adjusted as it is a negative number).
`errorLevel -`
The number of times that this function has been called.

Returns:

int The index of the condition in the goodConditions array that was satisfied after the error was handled.

Exceptions:

`AEFException -`
Because of the nature of this method, just about any error code could be returned.
Among the possible AEFException error codes are:
Common Errors

getAllExceptions

```
public AEFException getAllExceptions()
```

Returns the allExceptions variable which contains the exceptions encountered while trying to handle the error.

Returns:

AEFException the allExceptions variable

getSleepInterval

```
public int getSleepInterval()
```

Returns the number of milliseconds to wait before continuing with the next operation while handling errors. During testing, we found that differences in hardware, and software configuration could lead to timing issues. In many cases, invalid key sequence errors were erroneously generated. In order to compensate for this, JAVA sleep calls were added to the AEFErrorHandler and ErrorHelper classes. The Sleep Interval is configured in config.properties.

(continued from last page)

Returns:

int The Sleep Interval in milliseconds.

sleep

```
public void sleep()
```

Causes the Error Helper to wait before continuing with the next operation. During testing, we found that differences in hardware, and software configuration could lead to timing issues. In many cases, invalid key sequence errors were erroneously generated. In order to compensate for this, sleep calls (using this method) were added to the AEFErrorHandler and ErrorHelper classes. The default Sleep Interval is configured in config.properties.

sleep

```
public void sleep(int userInterval)
```

Causes the Error Helper to wait before continuing with the next operation. During testing, we found that differences in hardware, and software configuration could lead to timing issues. In many cases, invalid key sequence errors were erroneously generated. In order to compensate for this, sleep calls (using this method) were added to the AEFErrorHandler and ErrorHelper classes. The default Sleep Interval is configured in config.properties.

Parameters:

`userInterval -`
A user specified interval to be used instead of the default sleep value.

getErrorHandlingMode

```
public int getErrorHandlingMode()
                          throws AEFException
```

Gets the error handling mode from the Automation Provider.

Returns:

Int The error handling mode.

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
AEFConst.PROPERTY_NOT_SUPPORTED
Common Errors

getAdditionalData

```
public java.lang.Object getAdditionalData()
```

Gets the additional data pased by the calling application or a NULL.

Returns:

Object The additional data pased by the calling application or a NULL.

com.ibm.retail.AEF.util

Class AEFMessage

```
java.lang.Object
   |
   +-com.ibm.retail.AEF.util.AEFMessage
```

public class **AEFMessage**
extends java.lang.Object

AEFMessage is a class which the AEF applications use to form a message to log.

Field Summary

`java.lang.String`	`message`
`java.lang.String`	`sessionID`

Constructor Summary

`AEFMessage()` Constructors
`AEFMessage(java.lang.String msg)`
`AEFMessage(java.lang.String sID,java.lang.String sID)`

Method Summary

`java.lang.String`	`getMessage()` Gets the message text
`java.lang.String`	`getSessionID()` Gets the session ID
`void`	`setMessage(java.lang.String msg)` Sets the message text
`void`	`setSessionID(java.lang.String id)` Sets the session ID
`java.lang.String`	`toString()` Returns a string which is the formatted message.

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

sessionID

```
protected java.lang.String sessionID
```

message

```
protected java.lang.String message
```

Constructors

AEFMessage

```
public AEFMessage()
```

Constructors

AEFMessage

```
public AEFMessage(java.lang.String msg)
```

AEFMessage

```
public AEFMessage(java.lang.String sID,
                  java.lang.String msg)
```

Methods

toString

```
public java.lang.String toString()
```

Returns a string which is the formatted message.

Returns:

String The formatted message.

setMessage

```
public void setMessage(java.lang.String msg)
```

Sets the message text

Parameters:

msg

(continued from last page)

getMessage

`public java.lang.String **getMessage**()`

Gets the message text

Returns:

String msg

setSessionID

`public void **setSessionID**(java.lang.String id)`

Sets the session ID

Parameters:

`id` - sessionID

getSessionID

`public java.lang.String **getSessionID**()`

Gets the session ID

Returns:

String

com.ibm.retail.AEF.util

Interface ErrorHelper

public interface **ErrorHelper**

ErrorHelper is an interface which defines the methods needed for the AEF Error Handler to handle application errors.

Field Summary

static int	ERROR_HANDLED
static int	ERROR_IS_FATAL

Method Summary

java.lang.String	getCurrentState() Gets the current state.
AEFError	getError() Gets the error that we are working on.
AEFErrorHandler	getErrorHandler() Gets the Error Handler that created this Error Helper.
AEFException	getOuterException() Returns the outer most Exception (if any) that was thrown while handling this error.
int	getSleepInterval() Returns the number of milliseconds to wait before continuing with the next operation while handling errors.
int	handleError(ConditionLock lock,AEFAction lock,Condition[] lock,Condition[] lock,int lock,int lock) Tries to resolve the error condition if possible.
boolean	isEmulatedDevice(com.ibm.retail.AEF.io.DeviceRegistry deviceRegistry,java.lang.String deviceRegistry) IsEmulatedDevice Used to determine if a device is real or emulated.
boolean	isEmulatedDevice(java.lang.String logicalName) IsEmulatedDevice Used to determine if a device is real or emulated.
void	resetInputDevices() Reset the input devices
void	setError(AEFError theError) Sets the error that we are working on.
void	setErrorHandler(AEFErrorHandler theErrorHandler) Sets the Error Handler that created this Error Helper.

void	`setInputDevicesLocked(boolean flag)` Lock the input devices
void	`setOuterException(AEFException theException)` Sets the outer most Exception that was thrown while handling this error.
void	`sleep()` Causes the Error Helper to wait before continuing with the next operation.
void	`sleep(int userInterval)` Causes the Error Helper to wait before continuing with the next operation.

Fields

ERROR_HANDLED

`public static final int` **`ERROR_HANDLED`**

ERROR_IS_FATAL

`public static final int` **`ERROR_IS_FATAL`**

Methods

getError

`public AEFError` **`getError`**`()`

Gets the error that we are working on.

Returns:

theError The AEFError object that contains the information about the error we are trying to handle.

setError

`public void` **`setError`**`(AEFError theError)`

Sets the error that we are working on.

Parameters:

`theError -`
The AEFError object that contains the information about the error we are trying to handle.

handleError

```
public int handleError(ConditionLock lock,
                       AEFAction keySequenceAction,
                       Condition[] goodConditions,
                       Condition[] badConditions,
                       int timeout,
                       int satisfiedCondition)
```

Tries to resolve the error condition if possible.

(continued from last page)

Parameters:

`lock -`
The ConditionLock that returned the error.
`keySequenceAction -`
The key sequence that generated the error.
`goodConditions -`
An array of the conditions that would indicate success after the key sequence is sent.
`badConditions -`
An array of the conditions that would indicate a failure after the key sequence is sent.
`timeout -`
An integer used to determine how long to wait for one of the listed conditions before a timeout exception is thrown.
`satisfiedCondition -`
The index of the condition in the badConditions array that was satisfied on the last wait(needs to be adjusted as it is a negative number).

Returns:

int Returns ErrorHelper.ERROR_HANDLED if the error was cleared or dealt with. Returns ErrorHelper.ERROR_IS_FATAL if the error was not cleared or dealt with.

getOuterException

`public AEFException` **`getOuterException`**`()`

Returns the outer most Exception (if any) that was thrown while handling this error.

Returns:

AEFException The outer most Exception that was thrown while handling this error or null.

setOuterException

`public void` **`setOuterException`**`(AEFException theException)`

Sets the outer most Exception that was thrown while handling this error.

Parameters:

`The -`
outer most exception.

getSleepInterval

`public int` **`getSleepInterval`**`()`

Returns the number of milliseconds to wait before continuing with the next operation while handling errors. During testing, we found that differences in hardware, and software configuration could lead to timing issues. In many cases, invalid key sequence errors were erroneously generated. In order to compensate for this, JAVA sleep calls were added to the AEFErrorHandler and ErrorHelper classes. The Sleep Interval is configured in config.properties.

Returns:

int The Sleep Interval in milliseconds.

sleep

`public void` **`sleep`**`()`

Causes the Error Helper to wait before continuing with the next operation. During testing, we found that differences in hardware, and software configuration could lead to timing issues. In many cases, invalid key sequence errors were erroneously generated. In order to compensate for this, sleep calls (using this method) were added to the AEFErrorHandler and ErrorHelper classes. The default Sleep Interval is configured in config.properties.

(continued from last page)

sleep

```
public void sleep(int userInterval)
```

Causes the Error Helper to wait before continuing with the next operation. During testing, we found that differences in hardware, and software configuration could lead to timing issues. In many cases, invalid key sequence errors were erroneously generated. In order to compensate for this, sleep calls (using this method) were added to the AEFErrorHandler and ErrorHelper classes. The default Sleep Interval is configured in config.properties.

Parameters:

`userInterval -`
A user specified interval to be used instead of the default sleep value.

isEmulatedDevice

```
public boolean isEmulatedDevice(java.lang.String logicalName)
                          throws AEFException
```

IsEmulatedDevice Used to determine if a device is real or emulated.

Parameters:

`logicalName -`
The JPOS logical name for the device.

Returns:

boolean True if the device is emulated otherwise False.

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
Common Errors

isEmulatedDevice

```
public boolean isEmulatedDevice(com.ibm.retail.AEF.io.DeviceRegistry deviceRegistry,
                                java.lang.String logicalName)
                          throws jpos.JposException
```

IsEmulatedDevice Used to determine if a device is real or emulated.

Parameters:

`deviceRegistry -`
The device registry for this session.
`logicalName -`
The JPOS logical name for the device.

Returns:

boolean True if the device is emulated otherwise False.

Exceptions:

`JposException`

getCurrentState

```
public java.lang.String getCurrentState()
```

Gets the current state.

Returns:

String The current application state.

setInputDevicesLocked

```
public void setInputDevicesLocked(boolean flag)
                                  throws AEFException
```

Lock the input devices

Parameters:

`flag -`
A boolean that indicates if the are locking or unlocking the input devices.

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
Common Errors

resetInputDevices

```
public void resetInputDevices()
                               throws AEFException
```

Reset the input devices

Exceptions:

`AEFException -`
Among the possible AEFException error codes are:
Common Errors

getErrorHandler

```
public AEFErrorHandler getErrorHandler()
```

Gets the Error Handler that created this Error Helper.

Returns:

AEFErrorHandler The Error Handler that created this Error Helper.

setErrorHandler

```
public void setErrorHandler(AEFErrorHandler theErrorHandler)
```

Sets the Error Handler that created this Error Helper.

Parameters:

`theErrorHandler -`
The Error Handler that created this Error Helper.

com.ibm.retail.AEF.util

Interface LoggerControl

public interface **LoggerControl**
extends java.rmi.Remote

LoggerControl is an interface for enumerating the list of loggers, and for setting the level of a logger.

Method Summary

void	`addSocketHandler(java.lang.String loggerName,java.lang.String loggerName,int loggerName)` Adds a SocketHandler for a logger
java.util.ArrayList	`getLoggerNames()` Returns an array of the current logger names.
void	`readConfiguration()` Reloads the configuration file for logger properties
void	`removeSocketHandler(java.lang.String loggerName,java.lang.String loggerName,int loggerName)` Removes a SocketHandler for a logger
void	`setLoggerLevel(java.lang.String loggerName,java.lang.String loggerName)` Sets the level of a logger.

Methods

setLoggerLevel

```
public void setLoggerLevel(java.lang.String loggerName,
                           java.lang.String level)
                    throws java.rmi.RemoteException,
                           AEFException
```

Sets the level of a logger.

Parameters:

`loggerName -`
The name of the logger.
`level -`
Valid levels are: SEVERE, WARNING, INFO, CONFIG, FINE, FINER, FINEST

Exceptions:

`java.rmi.RemoteException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.INVALID_ARGUMENT, AEFConst.NO_SUCH_LOGGER
AEFConst.INVALID_ARGUMENT, AEFConst.INVALID_LOGGER_LEVEL
Common Errors

(continued from last page)

readConfiguration

```
public void readConfiguration()
                        throws java.rmi.RemoteException,
                               java.io.IOException
```

Reloads the configuration file for logger properties

Exceptions:

`java.rmi.RemoteException`
`java.rmi.IOException`

---

addSocketHandler

```
public void addSocketHandler(java.lang.String loggerName,
                             java.lang.String host,
                             int port)
                      throws java.rmi.RemoteException,
                             AEFException,
                             java.io.IOException
```

Adds a SocketHandler for a logger

Parameters:

`loggerName -`
The name of the logger.
`host -`
The host to log to.
`port -`
The port to log to.

Exceptions:

`java.rmi.RemoteException`
`java.rmi.IOException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.INVALID_ARGUMENT, AEFConst.NO_SUCH_LOGGER
Common Errors

---

removeSocketHandler

```
public void removeSocketHandler(java.lang.String loggerName,
                                java.lang.String host,
                                int port)
                         throws java.rmi.RemoteException,
                                AEFException,
                                java.io.IOException
```

Removes a SocketHandler for a logger

Parameters:

`loggerName -`
The name of the logger.
`host -`
The host to log to.
`port -`
The port to log to.

Exceptions:

`java.rmi.RemoteException`
`java.rmi.IOException`
`AEFException -`
Among the possible AEFException error codes are:
AEFConst.INVALID_ARGUMENT, AEFConst.NO_SUCH_LOGGER
Common Errors

getLoggerNames

```
public java.util.ArrayList getLoggerNames()
                                  throws java.rmi.RemoteException
```

Returns an array of the current logger names.

Returns:

ArrayList

Exceptions:

`java.rmi.RemoteException`

Package
com.ibm.retail.AEF.workstation

Provides an interface for accessing the physical or virtual workstation devices such as keyboard and line display.

com.ibm.retail.AEF.workstation

Class KeyCode

```
java.lang.Object
  |
  +-com.ibm.retail.AEF.workstation.KeyCode
```

All Implemented interfaces:
java.io.Serializable

public class **KeyCode**
extends java.lang.Object
implements java.io.Serializable

KeyCode encapsulates data for a POS function or data key.

Field Summary

java.lang.String	dataString
boolean	doubleKey
boolean	functionKey
int	logicalKeyCode
int	rawScanCode

Constructor Summary

KeyCode() Default constructor
KeyCode(int rawScanCode,int rawScanCode,java.lang.String rawScanCode,boolean rawScanCode,boolean rawScanCode) Constructor

Method Summary

java.lang.String	getDataString() Get the string value of this key For double zeroes returns "00" For triple zeroes returns "000"
int	getLogicalKeyCode() Get the logical key code
int	getRawScanCode() Get the raw scan code

boolean	`isDoubleKey()` Is this a double key (Two keys physically joined)
boolean	`isFunctionKey()` Is this a function key
void	`setDataString(java.lang.String value)` Set the string value of this key
void	`setDoubleKey(boolean value)` Set Is this a double key (Two keys physically joined)
void	`setFunctionKey(boolean value)` Set Is this a function key
void	`setLogicalKeyCode(int value)` Set the logical key code
void	`setRawScanCode(int value)` Set the raw scan code

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

rawScanCode

`protected int` **`rawScanCode`**

logicalKeyCode

`protected int` **`logicalKeyCode`**

dataString

`protected java.lang.String` **`dataString`**

doubleKey

`protected boolean` **`doubleKey`**

functionKey

`protected boolean` **`functionKey`**

(continued from last page)

Constructors

KeyCode

```
public KeyCode()
```

Default constructor

KeyCode

```
public KeyCode(int rawScanCode,
               int logicalKeyCode,
               java.lang.String dataString,
               boolean doubleKey,
               boolean functionKey)
```

Constructor

Methods

getLogicalKeyCode

```
public int getLogicalKeyCode()
```

Get the logical key code

Returns:

int logical key code

getRawScanCode

```
public int getRawScanCode()
```

Get the raw scan code

Returns:

int raw scan code

getDataString

```
public java.lang.String getDataString()
```

Get the string value of this key For double zeroes returns "00" For triple zeroes returns "000"

Returns:

String value of this key

isFunctionKey

```
public boolean isFunctionKey()
```

Is this a function key

Returns:

true if this is a function key

isDoubleKey

`public boolean **isDoubleKey**()`

Is this a double key (Two keys physically joined)

Returns:

true if this is a double key

setLogicalKeyCode

`public void **setLogicalKeyCode**(int value)`

Set the logical key code

Parameters:

`int -`
logical key code

setRawScanCode

`public void **setRawScanCode**(int value)`

Set the raw scan code

Parameters:

`int -`
raw scan code

setDataString

`public void **setDataString**(java.lang.String value)`

Set the string value of this key

Parameters:

`String -`
value of this key

setFunctionKey

`public void **setFunctionKey**(boolean value)`

Set Is this a function key

Parameters:

`true -`
if this is a function key

setDoubleKey

`public void **setDoubleKey**(boolean value)`

Set Is this a double key (Two keys physically joined)

Parameters:

`true -`
if this is a double key

com.ibm.retail.AEF.workstation

Interface KeyConsumer

All Known Implementing Classes:
KeyConsumerProxy

---

public interface **KeyConsumer**
extends java.rmi.Remote

Consumer/Listener interface for receiving POS keyboard events from the terminal workstation. An event is fired whenever a function key or data key is pressed on the POS keyboard.

WARNING: *This is a synchronous call which allows keyboard events to be consumed if the listener returns a value of "true". As a result, the POS sales application is blocked waiting until the listener event processing completes. For best performance it is recommended that any lengthy event processing (e.g. GUI updates) be performed after a Thread context switch. SwingUtilities.invokeLater() may be used for this purpose.*

Method Summary

`boolean`	`keyPress(KeyCode key)` A function or data key has been pressed.

Methods

keyPress

```
public boolean keyPress(KeyCode key)
                 throws java.rmi.RemoteException
```

A function or data key has been pressed.

Parameters:
`key -`
KeyCode object contains information about the key press

Returns:

boolean

com.ibm.retail.AEF.workstation

Interface Workstation

public interface **Workstation**
extends java.rmi.Remote

Workstation is an interface which allows interaction with workstation devices.

Field Summary

static java.lang.String	CLASS_KEY
static int	LOCKED_POSITION
static int	NORMAL_POSITION
static int	PHYSICAL_POSITION
static int	PTR_BC_CENTER
static int	PTR_BC_LEFT
static int	PTR_BC_RIGHT
static int	PTR_BC_TEXT_ABOVE
static int	PTR_BC_TEXT_BELOW
static int	PTR_BC_TEXT_NONE
static int	PTR_BCS_Codabar
static int	PTR_BCS_Code128
static int	PTR_BCS_Code39
static int	PTR_BCS_Code93
static int	PTR_BCS_EAN128
static int	PTR_BCS_EAN13
static int	PTR_BCS_EAN13_S
static int	PTR_BCS_EAN8

static int	PTR_BCS_EAN8_S
static int	PTR_BCS_ITF
static int	PTR_BCS_JAN13
static int	PTR_BCS_JAN8
static int	PTR_BCS_MAXICODE
static int	PTR_BCS_OCRA
static int	PTR_BCS_OCRB
static int	PTR_BCS_OTHER
static int	PTR_BCS_PDF417
static int	PTR_BCS_TF
static int	PTR_BCS_UPCA
static int	PTR_BCS_UPCA_S
static int	PTR_BCS_UPCD1
static int	PTR_BCS_UPCD2
static int	PTR_BCS_UPCD3
static int	PTR_BCS_UPCD4
static int	PTR_BCS_UPCD5
static int	PTR_BCS_UPCE
static int	PTR_BCS_UPCE_S
static int	PTR_BM_ASIS
static int	PTR_S_JOURNAL
static int	PTR_S_RECEIPT
static int	PTR_S_SLIP
static int	SUPERVISOR_POSITION

Method Summary

void	`addKeyConsumer(KeyConsumer c)` Add a KeyConsumer.
void	`clearLineDisplay()` Used to clear the line display device (LineDisplay1).
void	`clearLineDisplay(java.lang.String displayLogicalName)` Used to clear the line display device.
boolean	`getInputDevicesLocked()` Indicates whether input devices are allowed to provide input.
KeyCode	`getKeyCode(int rawscancode)` Returns a logical keyboard code object from a raw keyboard scancode.
int	`getKeyLockPosition()` Gets the logical keylock position which may or may not match the physical position.
java.lang.String	`getLastKeySequence()` Returns the last key sequence which was sent to the POS application via a call to the `sendKeySequence`method.
int	`getLogicalKeyCode(int rawscancode)` Returns a logical keyboard code from a raw keyboard scancode.
int	`getRawScanCode(int logicalcode)` Translates a logical keyboard code to a raw keyboard scancode
java.util.Vector	`getRawScanCodes(int logicalcode)` Translates a logical keyboard code to a Vector of raw keyboard scancode
AEFSession	`getSession()` Gets the AEFSession associated with this workstation.
boolean	`hasPendingMessage()` Determine if the workstation has a pending system message
boolean	`isLogicalKeyCodeDoubleKey(int keycode)` Indicates whether a logical keycode is part of a double key pair.
boolean	`isRawScanCodeDoubleKey(int scancode)` Indicates whether a raw scancode is part of a double key pair.
java.lang.String	`lineDisplayPrompt(java.lang.String promptText,int promptText,int promptText,int promptText,int promptText,java.lang.String promptText,java.lang.String promptText,java.lang.String promptText,java.lang.String promptText)` Used to prompt for input on a line display device (LineDisplay1) and keyboard.

java.lang.String	`lineDisplayPrompt(java.lang.String displayLogicalName,java.lang.String displayLogicalName,int displayLogicalName,int displayLogicalName,int displayLogicalName,int displayLogicalName,java.lang.String displayLogicalName,java.lang.String displayLogicalName,java.lang.String displayLogicalName,java.lang.String displayLogicalName)` Used to prompt for input on a line display device and keyboard.
void	`printBarCode(int Station,java.lang.String Station,int Station,int Station,int Station,int Station,int Station)` Used to print a bar code on the POS printer.
void	`printBitmap(int Station,java.lang.String Station,int Station,int Station)` Used to print a bitmap on the POS printer.
void	`printLine(int Station,java.lang.String Station)` Used to print a line on the POS printer.
java.lang.String	`readPendingMessage()` Reads pending system messages for the workstation.
void	`removeKeyConsumer(KeyConsumer c)` Remove a KeyConsumer.
void	`resendLastKeySequence()` Resends the last key sequence which was sent to the POS application.
void	`sendKeySequence(java.lang.String sequence)` Sends a key sequence to the transaction.
void	`setInputDevicesLocked(boolean locked)` Determines whether input devices are allowed to provide input.
void	`setInputPending(boolean pending)` Used to indicate there is input coming, but not yet sent.
void	`setKeyLockPosition(int position)` Overrides the reported keylock position.
void	`setLineDisplayText(java.lang.String text,int text,int text,boolean text)` Used to write text to the line display (LineDisplay1) device.
void	`setLineDisplayText(java.lang.String displayLogicalName,java.lang.String displayLogicalName,int displayLogicalName,int displayLogicalName,boolean displayLogicalName)` Used to write text to the line display device.
void	`setSession(AEFSession session)` Sets the AEFSession associated with this workstation.
void	`swipeMSR(byte[] track1Data,byte[] track1Data,byte[] track1Data)` Used to generate a MSR card swipe containing the specified track data.
void	`swipeMSR(java.lang.String track1Data,java.lang.String track1Data,java.lang.String track1Data)` Used to generate a MSR card swipe containing the specified track data.

Fields

CLASS_KEY

public static final java.lang.String **CLASS_KEY**

PHYSICAL_POSITION

public static final int **PHYSICAL_POSITION**

LOCKED_POSITION

public static final int **LOCKED_POSITION**

NORMAL_POSITION

public static final int **NORMAL_POSITION**

SUPERVISOR_POSITION

public static final int **SUPERVISOR_POSITION**

PTR_S_JOURNAL

public static final int **PTR_S_JOURNAL**

PTR_S_RECEIPT

public static final int **PTR_S_RECEIPT**

PTR_S_SLIP

public static final int **PTR_S_SLIP**

PTR_BCS_UPCA

public static final int **PTR_BCS_UPCA**

PTR_BCS_UPCE

public static final int **PTR_BCS_UPCE**

PTR_BCS_JAN8

public static final int **PTR_BCS_JAN8**

PTR_BCS_EAN8

public static final int **PTR_BCS_EAN8**

PTR_BCS_JAN13

public static final int **PTR_BCS_JAN13**

PTR_BCS_EAN13

public static final int **PTR_BCS_EAN13**

PTR_BCS_TF

public static final int **PTR_BCS_TF**

PTR_BCS_ITF

public static final int **PTR_BCS_ITF**

PTR_BCS_Codabar

public static final int **PTR_BCS_Codabar**

PTR_BCS_Code39

public static final int **PTR_BCS_Code39**

PTR_BCS_Code93

public static final int **PTR_BCS_Code93**

PTR_BCS_Code128

public static final int **PTR_BCS_Code128**

(continued from last page)

PTR_BCS_UPCA_S

public static final int **PTR_BCS_UPCA_S**

PTR_BCS_UPCE_S

public static final int **PTR_BCS_UPCE_S**

PTR_BCS_UPCD1

public static final int **PTR_BCS_UPCD1**

PTR_BCS_UPCD2

public static final int **PTR_BCS_UPCD2**

PTR_BCS_UPCD3

public static final int **PTR_BCS_UPCD3**

PTR_BCS_UPCD4

public static final int **PTR_BCS_UPCD4**

PTR_BCS_UPCD5

public static final int **PTR_BCS_UPCD5**

PTR_BCS_EAN8_S

public static final int **PTR_BCS_EAN8_S**

PTR_BCS_EAN13_S

public static final int **PTR_BCS_EAN13_S**

PTR_BCS_EAN128

public static final int **PTR_BCS_EAN128**

PTR_BCS_OCRA

public static final int **PTR_BCS_OCRA**

PTR_BCS_OCRB

public static final int **PTR_BCS_OCRB**

PTR_BCS_PDF417

public static final int **PTR_BCS_PDF417**

PTR_BCS_MAXICODE

public static final int **PTR_BCS_MAXICODE**

PTR_BCS_OTHER

public static final int **PTR_BCS_OTHER**

PTR_BC_LEFT

public static final int **PTR_BC_LEFT**

PTR_BC_CENTER

public static final int **PTR_BC_CENTER**

PTR_BC_RIGHT

public static final int **PTR_BC_RIGHT**

PTR_BC_TEXT_NONE

public static final int **PTR_BC_TEXT_NONE**

PTR_BC_TEXT_ABOVE

public static final int **PTR_BC_TEXT_ABOVE**

PTR_BC_TEXT_BELOW

public static final int **PTR_BC_TEXT_BELOW**

(continued from last page)

PTR_BM_ASIS

```
public static final int PTR_BM_ASIS
```

Methods

sendKeySequence

```
public void sendKeySequence(java.lang.String sequence)
                     throws AEFException,
                            java.rmi.RemoteException
```

Sends a key sequence to the transaction. A key sequence is composed of function keys designated by <ff> where ff is the function code of the key sequence. The function code is mapped to a keyboard scancode in a system dependent fashion. The key sequence may contain raw numeric and alphabetic characters.

Parameters:

`The -`
sequence string. Example: "<81>5000<95>".

Exceptions:

`java.rmi.RemoteException`
`com.ibm.retail.AEF.util.AEFException -`
Possible return codes are:
AEFConst.CONFIG_ERROR, AEFConst.INVALID_KEYSEQUENCE_EXPRESSION
AEFConst.CONFIG_ERROR, AEFConst.NONNUMERIC_FUNCTIONCODE
AEFConst.SYSTEM_BUSY
AEFConst.INPUT_NOT_ALLOWED

---

resendLastKeySequence

```
public void resendLastKeySequence()
                           throws AEFException,
                                  java.rmi.RemoteException
```

Resends the last key sequence which was sent to the POS application. This is typically used in a situation where there was an error that was cleared, and the key sequence needs to be retried.

Exceptions:

`java.rmi.RemoteException`
`com.ibm.retail.AEF.util.AEFException -`
Possible return codes are:
AEFConst.CONFIG_ERROR, AEFConst.INVALID_KEYSEQUENCE_EXPRESSION
AEFConst.CONFIG_ERROR, AEFConst.NONNUMERIC_FUNCTIONCODE
AEFConst.SYSTEM_BUSY
AEFConst.INPUT_NOT_ALLOWED

---

getLastKeySequence

```
public java.lang.String getLastKeySequence()
                                     throws java.rmi.RemoteException
```

Returns the last key sequence which was sent to the POS application via a call to the `sendKeySequence`method.

Returns:

String The last key sequence sent via the `sendKeySequence`method.

Exceptions:

`java.rmi.RemoteException`

(continued from last page)

getKeyCode

```
public KeyCode getKeyCode(int rawscancode)
                    throws java.rmi.RemoteException
```

Returns a logical keyboard code object from a raw keyboard scancode.

Parameters:

`raw -`
scancode value to translate

Returns:

keycode object for POS application

Exceptions:

`java.rmi.RemoteException`

getLogicalKeyCode

```
public int getLogicalKeyCode(int rawscancode)
                      throws java.rmi.RemoteException
```

Returns a logical keyboard code from a raw keyboard scancode.

Parameters:

`raw -`
scancode value to translate

Returns:

mapped code value for POS application (this is a function code on 4690). Returns zero if there is no keyboard, or there is no mapping for the specified scancode.

Exceptions:

`java.rmi.RemoteException`

getRawScanCode

```
public int getRawScanCode(int logicalcode)
                    throws java.rmi.RemoteException
```

Translates a logical keyboard code to a raw keyboard scancode

Parameters:

`logical -`
keyboard code value to translate (this is a function code on 4690).

Returns:

scancode value for POS application, if the value is -1, there is no scancode mapping for the requested function code.

Exceptions:

`java.rmi.RemoteException`

getRawScanCodes

```
public java.util.Vector getRawScanCodes(int logicalcode)
                                 throws java.rmi.RemoteException
```

Translates a logical keyboard code to a Vector of raw keyboard scancode

Parameters:

(continued from last page)

`logical -`
keyboard code value to translate (this is a function code on 4690).

Returns:

Vector of scancode values for POS application, if the value is null there is no scancode mapping for the requested function code.

Exceptions:
`java.rmi.RemoteException`

isRawScanCodeDoubleKey

```
public boolean isRawScanCodeDoubleKey(int scancode)
                                      throws java.rmi.RemoteException
```

Indicates whether a raw scancode is part of a double key pair.

Parameters:
`scancode -`
value to check.

Returns:

boolean

Exceptions:
`java.rmi.RemoteException`

isLogicalKeyCodeDoubleKey

```
public boolean isLogicalKeyCodeDoubleKey(int keycode)
                                         throws java.rmi.RemoteException
```

Indicates whether a logical keycode is part of a double key pair.

Parameters:
`keycode -`
value to check.

Returns:

boolean

Exceptions:
`java.rmi.RemoteException`

setInputDevicesLocked

```
public void setInputDevicesLocked(boolean locked)
                                  throws AEFException,
                                         java.rmi.RemoteException
```

Determines whether input devices are allowed to provide input.

Parameters:
`locked -`
Set true to lock all input devices, false to unlock all input devices.

Exceptions:
`java.rmi.RemoteException`
`com.ibm.retail.AEF.util.AEFException`

getInputDevicesLocked

```
public boolean getInputDevicesLocked()
                              throws AEFException,
                                     java.rmi.RemoteException
```

Indicates whether input devices are allowed to provide input.

Returns:

True if all input devices are locked, false otherwise.

Exceptions:

`java.rmi.RemoteException`
`com.ibm.retail.AEF.util.AEFException`

lineDisplayPrompt

```
public java.lang.String lineDisplayPrompt(java.lang.String displayLogicalName,
                                          java.lang.String promptText,
                                          int row,
                                          int startCol,
                                          int endCol,
                                          int maxLen,
                                          java.lang.String doneFCode,
                                          java.lang.String clearFCode,
                                          java.lang.String cancelFCode,
                                          java.lang.String maskChar)
                                   throws AEFException,
                                          java.rmi.RemoteException
```

Used to prompt for input on a line display device and keyboard. This method will unlock input devices if they are locked.

Parameters:

`displayLogicalName -`
The logical name of the line display which should be used to display the prompt. Should be LineDisplay1 or LineDisplay2.
`promptText -`
The text which should be displayed on the line display device. The text will be broken into multiple lines as appropriate.
`row -`
The row where input will be displayed (1 based index)
`startCol -`
The starting column where input will be displayed (1 based index)
`endCol -`
The ending column where input will be displayed (1 based index)
`maxLen -`
The maximum length of the input.
`doneFCode -`
The function code which indicates the input is complete. This function code is mapped to a keyboard scancode in an application specific manner.
`clearFCode -`
The function code which clears the input field. This function code is mapped to a keyboard scancode in an application specific manner.
`cancelFCode -`
The function code which cancels the input operation. This function code is mapped to a keyboard scancode in an application specific manner. If this function code is detected, an AEFException is thrown with an INPUT_CANCELLED error code. This value may be null if cancel is not allowed.
`maskChar -`
An optional string containing a single character which will be used as a mask character to display instead of the actual input. This is used for entering sensitive data such as passwords. Use a null string for no masking.

Returns:

String The data as entered by the operator on the keyboard.

(continued from last page)

Exceptions:

`java.rmi.RemoteException`

`com.ibm.retail.AEF.util.AEFException -`
Possible return codes are:
AEFConst.INVALID_ARGUMENT
AEFConst.INPUT_CANCELLED
AEFConst.DEVICE_HOOK_ERROR

lineDisplayPrompt

```
public java.lang.String lineDisplayPrompt(java.lang.String promptText,
                                          int row,
                                          int startCol,
                                          int endCol,
                                          int maxLen,
                                          java.lang.String doneFCode,
                                          java.lang.String clearFCode,
                                          java.lang.String cancelFCode,
                                          java.lang.String maskChar)
                                   throws AEFException,
                                          java.rmi.RemoteException
```

Used to prompt for input on a line display device (LineDisplay1) and keyboard. This method will unlock input devices if they are locked.

Parameters:

`promptText -`
The text which should be displayed on the line display device. The text will be broken into multiple lines as appropriate.

`row -`
The row where input will be displayed (1 based index)

`startCol -`
The starting column where input will be displayed (1 based index)

`endCol -`
The ending column where input will be displayed (1 based index)

`maxLen -`
The maximum length of the input.

`doneFCode -`
The function code which indicates the input is complete. This function code is mapped to a keyboard scancode in an application specific manner.

`clearFCode -`
The function code which clears the input field. This function code is mapped to a keyboard scancode in an application specific manner.

`cancelFCode -`
The function code which cancels the input operation. This function code is mapped to a keyboard scancode in an application specific manner. If this function code is detected, an AEFException is thrown with an INPUT_CANCELLED error code. This value may be null if cancel is not allowed.

`maskChar -`
An optional string containing a single character which will be used as a mask character to display instead of the actual input. This is used for entering sensitive data such as passwords. Use a null string for no masking.

Returns:

String The data as entered by the operator on the keyboard.

Exceptions:

`java.rmi.RemoteException`

`com.ibm.retail.AEF.util.AEFException -`
Possible return codes are:
AEFConst.INVALID_ARGUMENT
AEFConst.INPUT_CANCELLED
AEFConst.DEVICE_HOOK_ERROR

(continued from last page)

setLineDisplayText

```
public void setLineDisplayText(java.lang.String displayLogicalName,
                               java.lang.String text,
                               int row,
                               int col,
                               boolean clear)
                        throws AEFException,
                               java.rmi.RemoteException
```

Used to write text to the line display device.

Parameters:

`displayLogicalName -`
The logical name of the line display which should be used to display the text. Should be LineDisplay1 or LineDisplay2.
`text -`
The text which should be displayed on the line display device. The text will be broken into multiple lines as appropriate.
`row -`
The row where input will be displayed (1 based index)
`col -`
The starting column where input will be displayed (1 based index)
`clear -`
boolean indicating whether the line display should be cleared prior to displaying the specified text.

Exceptions:

`java.rmi.RemoteException`
`com.ibm.retail.AEF.util.AEFException -`
Possible return codes are:
AEFConst.INVALID_ARGUMENT
AEFConst.DEVICE_HOOK_ERROR

setLineDisplayText

```
public void setLineDisplayText(java.lang.String text,
                               int row,
                               int col,
                               boolean clear)
                        throws AEFException,
                               java.rmi.RemoteException
```

Used to write text to the line display (LineDisplay1) device.

Parameters:

`text -`
The text which should be displayed on the line display device. The text will be broken into multiple lines as appropriate.
`row -`
The row where input will be displayed (1 based index)
`col -`
The starting column where input will be displayed (1 based index)
`clear -`
boolean indicating whether the line display should be cleared prior to displaying the specified text.

Exceptions:

`java.rmi.RemoteException`
`com.ibm.retail.AEF.util.AEFException -`
Possible return codes are:
AEFConst.INVALID_ARGUMENT

clearLineDisplay

```
public void clearLineDisplay(java.lang.String displayLogicalName)
                     throws AEFException,
                            java.rmi.RemoteException
```

(continued from last page)

Used to clear the line display device.

Parameters:

`displayLogicalName -`
The logical name of the line display which should be cleared. Should be LineDisplay1 or LineDisplay2.

Exceptions:

`java.rmi.RemoteException`
`com.ibm.retail.AEF.util.AEFException -`
Possible return codes are:
AEFConst.INVALID_ARGUMENT
AEFConst.DEVICE_HOOK_ERROR

clearLineDisplay

```
public void clearLineDisplay()
                      throws AEFException,
                             java.rmi.RemoteException
```

Used to clear the line display device (LineDisplay1).

Exceptions:

`java.rmi.RemoteException`
`com.ibm.retail.AEF.util.AEFException -`
Possible return codes are:
AEFConst.INVALID_ARGUMENT
AEFConst.DEVICE_HOOK_ERROR

printBarCode

```
public void printBarCode(int Station,
                         java.lang.String data,
                         int symbology,
                         int width,
                         int height,
                         int alignment,
                         int textPosition)
                  throws java.rmi.RemoteException,
                         AEFException
```

Used to print a bar code on the POS printer.

Parameters:

`Station -`
PTR_S_RECEIPT, PTR_S_SLIP
`data -`
The bar code data
`symbology -`
PTR_BCS_UPCA, etc.
`height -`
Height of the barcode (defaults to dots)
`width -`
Width of the barcode (defaults to dots)
`alignment -`
PTR_BC_LEFT, PTR_BC_CENTER, PTR_BC_RIGHT
`textPositoin -`
PTR_BC_TEXT_NONE, PTR_BC_TEXT_ABOVE, PTR_BC_TEXT_BELOW

Exceptions:

`java.rmi.RemoteException`
`AEFException -`
Possible return codes are:
AEFConst.DEVICE_NOT_REDIRECTED if the printer is not redirected.
AEFConst.JAVA_POS_EXCEPTION, see exception cause() for actual javaPos exception.

(continued from last page)

printBitmap

```
public void printBitmap(int Station,
                        java.lang.String filename,
                        int width,
                        int alignment)
                 throws java.rmi.RemoteException,
                        AEFException
```

Used to print a bitmap on the POS printer.

Parameters:

`Station -`
PTR_S_JOURNAL, PTR_S_RECEIPT, PTR_S_SLIP
`filename -`
The filename of the image to print
`width -`
Width of the image (defaults to dots)
`alignment -`
PTR_BC_LEFT, PTR_BC_CENTER, PTR_BC_RIGHT

Exceptions:

`java.rmi.RemoteException`
`AEFException -`
Possible return codes are:
AEFConst.DEVICE_NOT_REDIRECTED if the printer is not redirected.
AEFConst.JAVA_POS_EXCEPTION, see exception cause() for actual javaPos exception.

printLine

```
public void printLine(int Station,
                      java.lang.String data)
               throws java.rmi.RemoteException,
                      AEFException
```

Used to print a line on the POS printer.

Parameters:

`Station -`
PTR_S_RECEIPT, PTR_S_SLIP
`data -`
The data to print.

Exceptions:

`java.rmi.RemoteException`
`AEFException -`
Possible return codes are:
AEFConst.DEVICE_NOT_REDIRECTED if the printer is not redirected.
AEFConst.JAVA_POS_EXCEPTION, see exception cause() for actual javaPos exception.

swipeMSR

```
public void swipeMSR(byte[] track1Data,
                     byte[] track2Data,
                     byte[] track3Data)
              throws java.rmi.RemoteException,
                     AEFException
```

Used to generate a MSR card swipe containing the specified track data. Note that the track data is not validated, and will be passed to the POS application as is.

Parameters:

`track1Data -`
A byte array containing the raw track 1 data (do not include sentinels). Byte values should fall within the range 0x00 - 0x3F. Use null or a zero length array if no track 1 data.

(continued from last page)

`track2Data -`
A byte array containing the raw track 2 data (do not include sentinels). Byte values should fall within the range 0x00 - 0x0F. Use null or a zero length array if no track 2 data.
`track3Data -`
A byte array containing the raw track 3 data (do not include sentinels). Byte values should fall within the range 0x00 - 0x0F. Use null or a zero length array if no track 3 data.

Exceptions:
`java.rmi.RemoteException`
`AEFException -`
Possible return codes are:
AEFConst.MSR_HOOK_SWIPE_ERROR, AEFConst.MSR_SET_TO_DECODE
AEFConst.MSR_HOOK_SWIPE_ERROR, AEFConst.MSR_NOT_ENABLED
AEFConst.JAVA_POS_EXCEPTION, see exception cause() for actual javaPos exception.
AEFConst.DEVICE_NOT_REDIRECTED if the MSR is not redirected.

swipeMSR

```
public void swipeMSR(java.lang.String track1Data,
                     java.lang.String track2Data,
                     java.lang.String track3Data)
              throws java.rmi.RemoteException,
                     AEFException
```

Used to generate a MSR card swipe containing the specified track data. Note that the track data is not validated. This method is provided as a convenience so that track data can be specified using characters rather than bytes. The actual byte data will be passed to the application. To convert the track 1 data to bytes, 0x20 will be subtracted from each character to come up with the byte value. For track 2 and track 3 data, 0x30 will be subtracted from each character to come up with the byte value.

Parameters:
`track1Data -`
A String containing the raw track 1 data (do not include sentinels). Character values should fall within the range ' ' to '_' (0x20 to 0x5F). Use null or an empty string if no track 1 data.
`track2Data -`
A byte array containing the raw track 2 data (do not include sentinels). Character values should fall within the range '0' to '?' (0x30 to 0x3F). Use null or an empty string if no track 2 data.
`track3Data -`
A byte array containing the raw track 3 data (do not include sentinels). Character values should fall within the range '0' - '?' (0x30 to 0x3F). Use null or an empty string if no track 3 data.

Exceptions:
`java.rmi.RemoteException`
`AEFException -`
Possible return codes are:
AEFConst.MSR_HOOK_SWIPE_ERROR, AEFConst.MSR_SET_TO_DECODE
AEFConst.MSR_HOOK_SWIPE_ERROR, AEFConst.MSR_NOT_ENABLED
AEFConst.JAVA_POS_EXCEPTION, see exception cause() for actual javaPos exception.
AEFConst.DEVICE_NOT_REDIRECTED if the MSR is not redirected.

setSession

```
public void setSession(AEFSession session)
                 throws java.rmi.RemoteException,
                        AEFException
```

Sets the AEFSession associated with this workstation.

Parameters:
`session`

Exceptions:
`java.rmi.RemoteException`
`AEFException -`
Possible return codes are:
AEFConst.JAVA_POS_EXCEPTION

getSession

```
public AEFSession getSession()
                        throws java.rmi.RemoteException
```

Gets the AEFSession associated with this workstation.

Returns:

AEFSession

Exceptions:

`java.rmi.RemoteException`

setKeyLockPosition

```
public void setKeyLockPosition(int position)
                         throws java.rmi.RemoteException,
                                AEFException
```

Overrides the reported keylock position. This method is used when the AEF is in automatic mode with automatic manager override to set the keylock position to the desired position. This method will cause the application to think that the keylock is in the requested position. This applies to simulated or physical keylock devices.

Parameters:

`position -`
Must be either LOCKED_POSITION, NORMAL_POSITION, SUPERVISOR_POSITION, PHYSICAL_POSITION

Exceptions:

`java.rmi.RemoteException`
`com.ibm.retail.AEF.util.AEFException -`
Possible return codes are:
AEFConst.INVALID_ARGUMENT
AEFConst.JAVA_POS_EXCEPTION
AEFConst.DEVICE_HOOK_ERROR

getKeyLockPosition

```
public int getKeyLockPosition()
                         throws java.rmi.RemoteException,
                                AEFException
```

Gets the logical keylock position which may or may not match the physical position. This applies to simulated or physical keylock devices.

Returns:

int Must be either LOCKED_POSITION, NORMAL_POSITION, SUPERVISOR_POSITION, PHYSICAL_POSITION

Exceptions:

`java.rmi.RemoteException`
`com.ibm.retail.AEF.util.AEFException -`
Possible return codes are:
AEFConst.JAVA_POS_EXCEPTION
AEFConst.DEVICE_HOOK_ERROR

readPendingMessage

```
public java.lang.String readPendingMessage()
                                   throws java.rmi.RemoteException
```

Reads pending system messages for the workstation.

(continued from last page)

Returns:

String text of the message (null if no messages)

Exceptions:
`java.rmi.RemoteException`

hasPendingMessage

```
public boolean hasPendingMessage()
                            throws java.rmi.RemoteException
```

Determine if the workstation has a pending system message

Returns:

true if the workstation has a pending system message

Exceptions:
`java.rmi.RemoteException`

addKeyConsumer

```
public void addKeyConsumer(KeyConsumer c)
                     throws java.rmi.RemoteException
```

Add a KeyConsumer. KeyConsumers receive synchronous notification for each keyboard event generated and can elect to consume the event.

Exceptions:
`java.rmi.RemoteException`

See Also:

com.ibm.retail.AEF.workstation.KeyConsumer

removeKeyConsumer

```
public void removeKeyConsumer(KeyConsumer c)
                        throws java.rmi.RemoteException
```

Remove a KeyConsumer. KeyConsumers receive synchronous notification for each keyboard event generated and can elect to consume the event.

Exceptions:
`java.rmi.RemoteException`

See Also:

com.ibm.retail.AEF.workstation.KeyConsumer

setInputPending

```
public void setInputPending(boolean pending)
                      throws java.rmi.RemoteException
```

Used to indicate there is input coming, but not yet sent. Used by the GUI during input (such as quantity input) so that an error will be generated if a scan is done while there is input pending.

Parameters:
`boolean`

Exceptions:

(continued from last page)

```
java.rmi.RemoteException
```

Package
com.ibm.retail.AEF.xml

Provides the XML parser and interfaces used by the AEF to parse event data from the POS application.

com.ibm.retail.AEF.xml

Interface SAXElementHandler

public interface **SAXElementHandler**

An interface for objects which receive data from the SAXEventParser. This is used by Event classes.

Method Summary

void	`endElement(java.lang.String name,java.lang.String name)` Called by the parser at the end of each element.
void	`startElement(java.lang.String name,org.xml.sax.Attributes name)` Parser calls this for each element in a document.

Methods

startElement

```
public void startElement(java.lang.String name,
                         org.xml.sax.Attributes atts)
```

Parser calls this for each element in a document.

Parameters:

`name` -
The element tag
`attrs` -
The element attributes

endElement

```
public void endElement(java.lang.String name,
                       java.lang.String value)
```

Called by the parser at the end of each element. If the element contained a character value, it is included.

Parameters:

`name` -
The element tag
`value` -
An optional character value of the element.

Package
com.ibm.retail.si.mgmt

This provides the core classes and interfaces used in the SIF Management infrastructure. Also included are:
- MBean Interfaces for common functions like inventory, JVM statistics, and extended control.
- Classes and interfaces used for General Agent Discovery

com.ibm.retail.si.mgmt

Interface MgmtAgent

public interface **MgmtAgent**

A MgmtAgent provides an interface to a JMX MBean server in addition to loading a set of common mbeans that provide management services

Method Summary

MgmtAgentConfiguration	getAgentConfiguration()
MgmtDeviceInfo	getDeviceInfo()
javax.management.MBeanServer	getMBeanServer()
boolean	isAgentStarted()
void	shutdown() Shutdown the agent completely, including its MBeanServer.

Methods

getMBeanServer

```
public javax.management.MBeanServer getMBeanServer()
```

Returns:

Instance of the JMX MBean Server

getDeviceInfo

```
public MgmtDeviceInfo getDeviceInfo()
```

Returns:

Device information object for this device

getAgentConfiguration

```
public MgmtAgentConfiguration getAgentConfiguration()
```

Returns:

A view of the agent's configuration information

isAgentStarted

```
public boolean isAgentStarted()
```

Returns:

True if the agent has started, false otherwise

shutdown

```
public void shutdown()
```

Shutdown the agent completely, including its MBeanServer.

com.ibm.retail.si.mgmt

Interface MgmtAgentConfiguration

public interface **MgmtAgentConfiguration**
extends java.io.Serializable

Method Summary

boolean	containsKey(java.lang.String key)
java.lang.String	getProperty(java.lang.String key)
java.lang.String	getProperty(java.lang.String key,java.lang.String key)
void	setProperty(java.lang.String key,java.lang.String key)

Methods

getProperty

```
public java.lang.String getProperty(java.lang.String key)
```

getProperty

```
public java.lang.String getProperty(java.lang.String key,
                                    java.lang.String defaultVal)
```

containsKey

```
public boolean containsKey(java.lang.String key)
```

setProperty

```
public void setProperty(java.lang.String key,
                        java.lang.String value)
```

com.ibm.retail.si.mgmt

Class MgmtAgentFactory

```
java.lang.Object
   |
   +-com.ibm.retail.si.mgmt.MgmtAgentFactory
```

public class **MgmtAgentFactory**
extends java.lang.Object

Factory class for obtaining the singleton instances of the Master or General Management agents. The agents are initialized and started upon the first call to either of the factory methods.

Constructor Summary

`MgmtAgentFactory()`

Method Summary

`static MgmtAgent`	`getGeneralAgent()` Returns the singleton instance of the general agent for the current JVM.
`static MgmtAgent`	`getGeneralAgent(int deviceType)` Creates a general agent instance with the supplied device type.
`static MgmtAgent`	`getMasterAgent()` Returns the singleton instance of the master agent for the current JVM.
`static boolean`	`isGeneralAgentRunning()`
`static boolean`	`isMasterAgentRunning()`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

MgmtAgentFactory

`public` **`MgmtAgentFactory`**`()`

Methods

(continued from last page)

getGeneralAgent

```
public static MgmtAgent getGeneralAgent()
                                 throws MgmtException
```

Returns the singleton instance of the general agent for the current JVM. The agent instance will be created upon the first call to this method.

Returns:

MgmtAgent Singleton general agent instance

Exceptions:

`MgmtException -`
Error initializing General Agent instance, possibly due to missing configuration file(s) or errors reading the configuration

---

getGeneralAgent

```
public static MgmtAgent getGeneralAgent(int deviceType)
                                 throws MgmtException
```

Creates a general agent instance with the supplied device type. Before calling this method, a check should be made to isGeneralAgentRunning() to see if there is an existing instance. Device type constants are specified in `MgmtConst`

Parameters:

`deviceType -`
Valid device type to use for this agent

Returns:

Singleton general agent instance, or null if the agent instance cannot be initialized

Exceptions:

`MgmtException -`
General Agent instance is already running, or invalid parameter specified

---

isGeneralAgentRunning

```
public static boolean isGeneralAgentRunning()
```

Returns:

boolean True if there is a general agent running in this JVM. False otherwise

---

isMasterAgentRunning

```
public static boolean isMasterAgentRunning()
```

Returns:

boolean True if there is a master agent running in this JVM. False otherwise

---

getMasterAgent

```
public static MgmtAgent getMasterAgent()
                                 throws MgmtException
```

Returns the singleton instance of the master agent for the current JVM. The agent instance will be created upon the first call to this method.

(continued from last page)

Returns:

Singleton master agent instance, or null if the agent instance cannot be initialized

Exceptions:

`MgmtException -`
Error initializing MasterAgent instance, possibly due to missing configuration file(s) or errors reading the configuration

com.ibm.retail.si.mgmt

Interface MgmtClientHealthMBean

All Superinterfaces:
MgmtHealthMBean, MgmtExtendedControlMBean

---

public interface **MgmtClientHealthMBean**
extends MgmtHealthMBean

Management interface for the MgmtClientHealth MBean. It represents the client side discovery and health checking done on behalf of the General Agent ONLY. The MBean emits UDP discovery packets at a configured interval for discovery by a Master Agent.

The `ObjectName`of this MBean includes the following attributes, in addition to the SIF attribute of `DeviceID`:
- SIFComponent=MGMT
- Id=discovery

This management interface includes the following attributes. These attributes are described in more detail in the accessor methods, and in the `MgmtHealthMBean`and `MgmtExtendedControlMBean`interfaces.

- Interval

The following operations are included in this management interface. These are described in more detail in the corresponding method documentation.
- No Operations Defined

An `AgentShutdownNotification`is emitted by classes implementing this interface when the General Agent is cleanly shut down. The notification `userData`will contain the `MgmtDeviceInfo`for the agent.

See Also:
com.ibm.retail.si.mgmt.MgmtHealthMBean

Field Summary

static java.lang.String	mbQueryDefault
static java.lang.String	mbQueryWAS
static java.lang.String	OBJECT_NAME

Method Summary

int	getInterval() Returns the number of seconds between the transmission of discovery UDP packets
void	setInterval(int seconds)

Fields

(continued from last page)

OBJECT_NAME

```
public static final java.lang.String OBJECT_NAME
```

mbQueryWAS

```
public static final java.lang.String mbQueryWAS
```

mbQueryDefault

```
public static final java.lang.String mbQueryDefault
```

Methods

setInterval

```
public void setInterval(int seconds)
                 throws javax.management.InvalidAttributeValueException
```

Parameters:

`seconds` -
The new number of seconds between discovery transmissions

getInterval

```
public int getInterval()
```

Returns the number of seconds between the transmission of discovery UDP packets

Returns:

The number of seconds between the transmission of discovery UDP packets

com.ibm.retail.si.mgmt

Interface MgmtConst

public interface **MgmtConst**

Management Constants

Field Summary

static int	`BASE_GA_MGMT_PORT` When an RMI Registry is created by the general agent, the first port that the general agent tries to use
static java.lang.String	`DEF_AGENT_PROP_FILE` Name of the Master Agent properties file
static int	`DefaultPingInterval` Number of seconds between the sending of discovery packets by a general agent
static int	`dType4690`
static int	`dTypeAssociate`
static int	`dTypeConsumer`
static int	`dTypePOSTerm`
static int	`dTypeROLO`
static int	`dTypeUnknown`
static int	`dTypeWAS5`
static java.lang.String	`GENERAL_AGENT_DOMAIN` Domain for the General Agent MBeanServer
static int	`MA_MGMT_PORT` Port used for remote connection to the Master Agent
static java.lang.String	`MASTER_AGENT_DOMAIN` Domain for the Master Agent MBeanServer
static java.lang.String	`MASTER_AGENT_PROXY_DOMAIN` Domain for Master Agent Proxy MBeans
static java.lang.String	`OBJ_NAME_DEV_MAJ_KEY`
static java.lang.String	`OBJ_NAME_DEV_MIN_KEY`

static java.lang.String	OBJ_NAME_DEVICEID_KEY
static java.lang.String	OBJ_NAME_ID_KEY
static java.lang.String	OBJ_NAME_MGMT_SIF_COMPONENT Value SIF Management Infrastructure MBeans are to use for the OBJ_NAME_SIFMBEAN_KEYObjectName key
static java.lang.String	OBJ_NAME_SIF_COMP_KEY
static java.lang.String	OBJ_NAME_SIFMBEAN_KEY
static java.lang.String	OBJ_NAME_STOREID_KEY
static java.lang.String	OBJ_NAME_SYSTEMID_KEY
static java.lang.String	PACKAGE
static int	SWDIST_RC_CANCELLED The distribution was cancelled
static int	SWDIST_RC_ERR_BUSY Busy threshold reached on software distribution client
static int	SWDIST_RC_ERR_GENERAL General error
static int	SWDIST_RC_ERR_NO_SD_CLIENT No MgmtSoftwareDistClientMBean found on target device
static int	SWDIST_RC_ERR_PASSWD FTP authentication error
static int	SWDIST_RC_ERR_SERVER Unable to reach the FTP Server
static int	SWDIST_RC_ERR_USERID Invalid FTP user ID
static int	SWDIST_RC_OK Good return code
static int	SWDIST_RC_OK_DEFFERED The distribution was deferred by the client
static java.lang.String	SYS_PROP_SIF_MGMT_HOME_KEY Property specifying the SIFHOME path, where all management jars and libraries are kept
static java.lang.String	SYS_PROP_STOREID_KEY Used only by master agents, this property specifies an identifier for the store

static java.lang.String	SYS_PROP_WAS_HOME_KEY Property specifying the base directory for WAS.
static java.lang.String	TMX4J_CONNECTOR_ADDRESS
static java.lang.String	TMX4J_CONNECTOR_CLASS_LOADER
static java.lang.String	TMX4J_CONNECTOR_IMPL_CLASS

Fields

PACKAGE

`public static final java.lang.String` **`PACKAGE`**

MASTER_AGENT_DOMAIN

`public static final java.lang.String` **`MASTER_AGENT_DOMAIN`**

Domain for the Master Agent MBeanServer

MASTER_AGENT_PROXY_DOMAIN

`public static final java.lang.String` **`MASTER_AGENT_PROXY_DOMAIN`**

Domain for Master Agent Proxy MBeans

GENERAL_AGENT_DOMAIN

`public static final java.lang.String` **`GENERAL_AGENT_DOMAIN`**

Domain for the General Agent MBeanServer

DEF_AGENT_PROP_FILE

`public static final java.lang.String` **`DEF_AGENT_PROP_FILE`**

Name of the Master Agent properties file

SYS_PROP_STOREID_KEY

`public static final java.lang.String` **`SYS_PROP_STOREID_KEY`**

Used only by master agents, this property specifies an identifier for the store

SYS_PROP_WAS_HOME_KEY

`public static final java.lang.String` **`SYS_PROP_WAS_HOME_KEY`**

Property specifying the base directory for WAS. Used only by master agents and general agents running on WAS

SYS_PROP_SIF_MGMT_HOME_KEY

`public static final java.lang.String` **`SYS_PROP_SIF_MGMT_HOME_KEY`**

Property specifying the SIFHOME path, where all management jars and libraries are kept

OBJ_NAME_STOREID_KEY

public static final java.lang.String **OBJ_NAME_STOREID_KEY**

OBJ_NAME_DEVICEID_KEY

public static final java.lang.String **OBJ_NAME_DEVICEID_KEY**

OBJ_NAME_SYSTEMID_KEY

public static final java.lang.String **OBJ_NAME_SYSTEMID_KEY**

OBJ_NAME_SIFMBEAN_KEY

public static final java.lang.String **OBJ_NAME_SIFMBEAN_KEY**

OBJ_NAME_SIF_COMP_KEY

public static final java.lang.String **OBJ_NAME_SIF_COMP_KEY**

OBJ_NAME_DEV_MAJ_KEY

public static final java.lang.String **OBJ_NAME_DEV_MAJ_KEY**

OBJ_NAME_DEV_MIN_KEY

public static final java.lang.String **OBJ_NAME_DEV_MIN_KEY**

OBJ_NAME_ID_KEY

public static final java.lang.String **OBJ_NAME_ID_KEY**

OBJ_NAME_MGMT_SIF_COMPONENT

public static final java.lang.String **OBJ_NAME_MGMT_SIF_COMPONENT**

Value SIF Management Infrastructure MBeans are to use for the `OBJ_NAME_SIFMBEAN_KEY`ObjectName key

BASE_GA_MGMT_PORT

public static final int **BASE_GA_MGMT_PORT**

When an RMI Registry is created by the general agent, the first port that the general agent tries to use

(continued from last page)

MA_MGMT_PORT

`public static final int` **`MA_MGMT_PORT`**

Port used for remote connection to the Master Agent

TMX4J_CONNECTOR_CLASS_LOADER

`public static final java.lang.String` **`TMX4J_CONNECTOR_CLASS_LOADER`**

TMX4J_CONNECTOR_ADDRESS

`public static final java.lang.String` **`TMX4J_CONNECTOR_ADDRESS`**

TMX4J_CONNECTOR_IMPL_CLASS

`public static final java.lang.String` **`TMX4J_CONNECTOR_IMPL_CLASS`**

dTypeUnknown

`public static final int` **`dTypeUnknown`**

dType4690

`public static final int` **`dType4690`**

dTypeROLO

`public static final int` **`dTypeROLO`**

dTypePOSTerm

`public static final int` **`dTypePOSTerm`**

dTypeConsumer

`public static final int` **`dTypeConsumer`**

dTypeAssociate

`public static final int` **`dTypeAssociate`**

dTypeWAS5

`public static final int` **`dTypeWAS5`**

DefaultPingInterval

`public static final int` **`DefaultPingInterval`**

Number of seconds between the sending of discovery packets by a general agent

SWDIST_RC_OK

`public static final int` **`SWDIST_RC_OK`**

Good return code

SWDIST_RC_OK_DEFFERED

`public static final int` **`SWDIST_RC_OK_DEFFERED`**

The distribution was deferred by the client

SWDIST_RC_CANCELLED

`public static final int` **`SWDIST_RC_CANCELLED`**

The distribution was cancelled

SWDIST_RC_ERR_GENERAL

`public static final int` **`SWDIST_RC_ERR_GENERAL`**

General error

SWDIST_RC_ERR_NO_SD_CLIENT

`public static final int` **`SWDIST_RC_ERR_NO_SD_CLIENT`**

No MgmtSoftwareDistClientMBean found on target device

SWDIST_RC_ERR_USERID

`public static final int` **`SWDIST_RC_ERR_USERID`**

Invalid FTP user ID

SWDIST_RC_ERR_SERVER

`public static final int` **`SWDIST_RC_ERR_SERVER`**

Unable to reach the FTP Server

SWDIST_RC_ERR_PASSWD

`public static final int` **`SWDIST_RC_ERR_PASSWD`**

FTP authentication error

SWDIST_RC_ERR_BUSY

`public static final int` **`SWDIST_RC_ERR_BUSY`**

Busy threshold reached on software distribution client

com.ibm.retail.si.mgmt

Class MgmtDeviceInfo

```
java.lang.Object
   |
   +-com.ibm.retail.si.mgmt.MgmtDeviceInfo
```

All Implemented interfaces:
java.io.Serializable

---

public class **MgmtDeviceInfo**
extends java.lang.Object
implements java.io.Serializable

A Collection of information pertaining to a device that has been discovered, and is being tracked by the discovery MBean: MgmtMasterHealth.

Field Summary

long	agentStartTime
int	agentType
boolean	connectionAttempted
int	connectionTics
java.lang.String	mbeanQueryString
java.lang.String	mgmtProtocol
int	MissedTics

Constructor Summary

MgmtDeviceInfo(java.lang.String systemID,int systemID,java.lang.String systemID,java.net.InetAddress systemID,int systemID,java.lang.String systemID,int systemID) Constructs a new instance with all needed values
MgmtDeviceInfo(java.lang.String systemId,int systemId,java.lang.String systemId,int systemId) Constructor to obtain an instance containing the minumum values required to determine equality

Method Summary

java.lang.String	BuildKey()
void	decrementConnectionTics()

boolean	equals(java.lang.Object o) Determines equality based on device id, device type, system id, and management port
java.net.InetAddress	getAddress()
long	getAgentStartTime()
int	getAgentType()
boolean	getConnectionAttempted()
int	getConnectionTics()
java.lang.String	getDeviceId()
int	getDeviceType()
java.lang.String	getMbeanQueryString()
int	getMgmtPort()
java.lang.String	getMgmtProtocol()
int	getMissedTics()
java.lang.String	getSystemId()
void	setAgentStartTime(long agentStartTime)
void	setAgentType(int agentType)
void	setConnectionAttempted(boolean connectionAttempted)
void	setMbeanQueryString(java.lang.String string)
void	setMissedTics(int i)
java.lang.String	toString()

Methods inherited from : class `java.lang.Object`
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

(continued from last page)

MissedTics

`protected int` **`MissedTics`**

agentType

`protected int` **`agentType`**

mbeanQueryString

`protected java.lang.String` **`mbeanQueryString`**

agentStartTime

`protected long` **`agentStartTime`**

connectionAttempted

`protected boolean` **`connectionAttempted`**

connectionTics

`protected int` **`connectionTics`**

mgmtProtocol

`protected java.lang.String` **`mgmtProtocol`**

Constructors

MgmtDeviceInfo

```
public MgmtDeviceInfo(java.lang.String systemID,
                      int deviceType,
                      java.lang.String deviceId,
                      java.net.InetAddress address,
                      int mgmtPort,
                      java.lang.String mgmtProtocol,
                      int configuredInterval)
```

Constructs a new instance with all needed values

Parameters:

`systemID` -
System Id of the remote agent

`deviceType` -
Device type of the agent device, as defined in `MgmtConst`

`deviceId` -
Device Id of the agent device

(continued from last page)

`address` -
Address of the interface to use for managment connectivity
`mgmtPort` -
Management port number
`mgmtProtocol` -
Name of the protocol used by the agent's `JMXConnectorServer`
`configuredInterval` -
For a general agent, the time interval between discovery broadcasts

MgmtDeviceInfo

```
public MgmtDeviceInfo(java.lang.String systemId,
                      int deviceType,
                      java.lang.String deviceId,
                      int mgmtPort)
```

Constructor to obtain an instance containing the minumum values required to determine equality

See Also:

com.ibm.retail.si.mgmt.MgmtDeviceInfo#MgmtDeviceInfo(String, int, String, InetAddress, int, String, int)

Methods

BuildKey

```
public java.lang.String BuildKey()
```

Returns:

A unique String key based on the agent information

toString

```
public java.lang.String toString()
```

See Also:

java.lang.Object#toString()

equals

```
public boolean equals(java.lang.Object o)
```

Determines equality based on device id, device type, system id, and management port

See Also:

java.lang.Object#equals(java.lang.Object)

getMissedTics

```
public int getMissedTics()
```

Returns:

The number of seconds since a discovery packet has been received by the `MgmtMasterHealthMBean`

setMissedTics

`public void **setMissedTics**(int i)`

Parameters:

`i -`
New MissedTics value

See Also:

com.ibm.retail.si.mgmt.MgmtDeviceInfo#getMissedTics()

getDeviceType

`public int **getDeviceType**()`

Returns:

The numerical device type for the device, as defined in `MgmtConst`

getSystemId

`public java.lang.String **getSystemId**()`

Returns:

The system identifier for this agent, which is a combination of device ID and management port

getDeviceId

`public java.lang.String **getDeviceId**()`

Returns:

The device identifier for this device

getAddress

`public java.net.InetAddress **getAddress**()`

Returns:

The address of the interface used for management

getMgmtPort

`public int **getMgmtPort**()`

Returns:

(continued from last page)

Port number used for management

getMgmtProtocol

`public java.lang.String getMgmtProtocol()`

Returns:

Name of the protocol used by the agent's `JMXConnectorServer`

getAgentType

`public int getAgentType()`

Returns:

The type of agent, either MASTER_AGENT, or GENERAL_AGENT

setAgentType

`protected void setAgentType(int agentType)`

Parameters:

`agentType -`
The new agent type

See Also:

com.ibm.retail.si.mgmt.MgmtDeviceInfo#getAgentType()

decrementConnectionTics

`protected void decrementConnectionTics()`

See Also:

com.ibm.retail.si.mgmt.MgmtDeviceInfo#getConnectionTics()

getConnectionTics

`protected int getConnectionTics()`

Returns:

The number of seconds since the last connection attempt to this agent by the Master Agent

getAgentStartTime

`public long getAgentStartTime()`

(continued from last page)

Returns:

The time in milliseconds when this agent was started

setAgentStartTime

```
protected void setAgentStartTime(long agentStartTime)
```

Parameters:

`agentStartTime -`
New start time

See Also:

com.ibm.retail.si.mgmt.MgmtDeviceInfo#getAgentStartTime()

setMbeanQueryString

```
public void setMbeanQueryString(java.lang.String string)
                        throws javax.management.MalformedObjectNameException
```

Parameters:

`string -`
New value

Exceptions:

`MalformedObjectNameException -`
Invalid query string

See Also:

com.ibm.retail.si.mgmt.MgmtDeviceInfo#getMbeanQueryString()

getMbeanQueryString

```
public java.lang.String getMbeanQueryString()
```

Returns:

The default MBean query string used by the Master Agent to determine the MBeans to proxy

getConnectionAttempted

```
public boolean getConnectionAttempted()
```

Returns:

True if an attempt has been made to contact the remote `MgmtAgent`

setConnectionAttempted

```
public void setConnectionAttempted(boolean connectionAttempted)
```

(continued from last page)

Parameters:

`connectionAttempted` -
New connection attemp status

See Also:

#getConnectionAttempted()

com.ibm.retail.si.mgmt

Class MgmtException

```
java.lang.Object
  +-java.lang.Throwable
      +-java.lang.Exception
          +-com.ibm.retail.si.mgmt.MgmtException
```

public class **MgmtException**
extends java.lang.Exception

Constructor Summary

`MgmtException()`
`MgmtException(java.lang.String message)`
`MgmtException(java.lang.Throwable rootCause)`
`MgmtException(java.lang.String message,java.lang.Throwable message)`

Method Summary

`java.lang.Throwable`	`getCause()`

Methods inherited from : class `java.lang.Throwable`
`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

MgmtException

`public` **`MgmtException`**`()`

(continued from last page)

MgmtException

```
public MgmtException(java.lang.String message)
```

MgmtException

```
public MgmtException(java.lang.Throwable rootCause)
```

MgmtException

```
public MgmtException(java.lang.String message,
                     java.lang.Throwable rootCause)
```

Methods

getCause

```
public java.lang.Throwable getCause()
```

com.ibm.retail.si.mgmt

Interface MgmtExtendedAgentControlMBean

public interface **MgmtExtendedAgentControlMBean**

This interface specifies a management interface for extended control functions for `MgmtAgent`objects. This MBean is created on demand by the `MgmtClientHealthMBean`and the `MgmtMasterHealthMBean`as a way of enabling detailed control.

The ObjectName of this MBean includes the following attributes, in addition to the required SIF attributes:
- SIFComponent=MGMT
- Id=ExtendedMgmtAgentControl

This management interface does not include any attributes.

The following operations are included in this management interface:
- startAgent
- shutdownAgent
- unloadMgmtMBeans
- loadMgmtMBeans
- deregisterAllMBeans

No notifications are generated by classes implementing this interface.

Field Summary

static java.lang.String	`OBJECT_NAME_BASE`

Method Summary

void	`deregisterAllMBeans()` Unloads and unregisters all Management MBeans defined in the agent's `MgmtAgentSetup`instance
void	`loadMgmtMBeans()` Loads all Management MBeans, if they are not loaded
void	`shutdownAgent()` Completely shuts down the `MgmtAgent`, if it has been started.
void	`startDiscovery()` This method does one of the following, depending on the agent type: **General Agents:**Starts the transmission of discovery packets, if it has been stopped**Master Agents:**Starts receiving discovery packets on all network interfaces, if any interfaces had been stopped
void	`stopDiscovery()` This method does one of the following, depending on the agent type: **General Agents:**Stops the transmission of discovery packets if started**Master Agents:**Stops receiving discovery packets on all network interfaces that are receiving, causing all agents to be eventually lost

(continued from last page)

Fields

OBJECT_NAME_BASE

`public static final java.lang.String` **`OBJECT_NAME_BASE`**

Methods

shutdownAgent

`public void` **`shutdownAgent`**`()`

Completely shuts down the `MgmtAgent`, if it has been started. This operation cannot be recovered from.

See Also:

com.ibm.retail.si.mgmt.AbstractMgmtAgent#shutdown()

deregisterAllMBeans

`public void` **`deregisterAllMBeans`**`()`

Unloads and unregisters all Management MBeans defined in the agent's `MgmtAgentSetup`instance

loadMgmtMBeans

```
public void loadMgmtMBeans()
                    throws MgmtException
```

Loads all Management MBeans, if they are not loaded

Exceptions:

`MgmtException -`
Error loading Management MBeans

startDiscovery

`public void` **`startDiscovery`**`()`

This method does one of the following, depending on the agent type: **General Agents:**Starts the transmission of discovery packets, if it has been stopped**Master Agents:**Starts receiving discovery packets on all network interfaces, if any interfaces had been stopped

stopDiscovery

`public void` **`stopDiscovery`**`()`

This method does one of the following, depending on the agent type: **General Agents:**Stops the transmission of discovery packets if started**Master Agents:**Stops receiving discovery packets on all network interfaces that are receiving, causing all agents to be eventually lost

com.ibm.retail.si.mgmt

Interface MgmtExtendedControlMBean

All Subinterfaces:
MgmtHealthMBean, MgmtMasterHealthMBean, MgmtClientHealthMBean, SessionServerMBean, SessionMBean

---

public interface **MgmtExtendedControlMBean**

Management interface for controlling extended management functions. This interface allows the remote creation and removal of extended capabilities that are not normally instanciated by the component's MBeans. Examples include extended debugging, extended logging,...

This interface should be implemented by any device/component that wishes to provide very detailed command/control/monitoring functions, but does not wish to expose those functions all the time. This interface would be implemented within an MBean that is always present, and the use of it will either create an instance of the extended function MBean or remove it. By using this functionality a management application can enable very granular control over a component ONLY when explicitly required.

This management interface includes the following attributes. These attributes are described in more detail in the accessor methods.

- ExtendedCapabilities
- CurrentlyActiveCapabilities

The following operations are included in this management interface. These are described in more detail in the corresponding method documentation.
- enableDebug
- destroyDebug
- enableDetailedLogControl
- destroyDetailedLogControl
- enableDetailedControl
- destroyDetailedControl

Field Summary

static int	EXTENDED_CTRL
static int	EXTENDED_DEBUG
static int	EXTENDED_LOG

Method Summary

int	destroyDebug() Destroy the instance of the extended debug MBean.
int	destroyDetailedControl() Destroy the instance of the extended debug MBean.
int	destroyDetailedLogControl() Destroy the instance of the extended debug MBean.

javax.management.ObjectName	enableDebug() Enable, if available, the extended Debug facilities of this component.
javax.management.ObjectName	enableDetailedControl() Enable, if available, the extended Control facilities of this component.
javax.management.ObjectName	enableDetailedLogControl() Enable, if available, the extended Logging control facilities of this component.
int	getCurrentlyActiveCapabilities() Query the which extened functions are currenly active.
int	getExtendedCapabilities() Query the capabilities of this implemetation of this interface.

Fields

EXTENDED_DEBUG

`public static final int` **`EXTENDED_DEBUG`**

EXTENDED_LOG

`public static final int` **`EXTENDED_LOG`**

EXTENDED_CTRL

`public static final int` **`EXTENDED_CTRL`**

Methods

getExtendedCapabilities

`public int` **`getExtendedCapabilities`**`()`

Query the capabilities of this implemetation of this interface.

Returns:

int - A mask indicating the functions that are available for control by this interface.

getCurrentlyActiveCapabilities

`public int` **`getCurrentlyActiveCapabilities`**`()`

Query the which extened functions are currenly active.

Returns:

int - A mask indicating the functions that are currenly active.

enableDebug

`public javax.management.ObjectName **enableDebug**()`

Enable, if available, the extended Debug facilities of this component.

Returns:

ObjectName - The ObjectName of the newly created MBean.

destroyDebug

`public int **destroyDebug**()`

Destroy the instance of the extended debug MBean.

Returns:

int - General return code.

enableDetailedLogControl

`public javax.management.ObjectName **enableDetailedLogControl**()`

Enable, if available, the extended Logging control facilities of this component.

Returns:

ObjectName - The ObjectName of the newly created MBean.

destroyDetailedLogControl

`public int **destroyDetailedLogControl**()`

Destroy the instance of the extended debug MBean.

Returns:

int - General return code.

enableDetailedControl

`public javax.management.ObjectName **enableDetailedControl**()`

Enable, if available, the extended Control facilities of this component.

Returns:

ObjectName - The ObjectName of the newly created MBean.

destroyDetailedControl

`public int **destroyDetailedControl**()`

Destroy the instance of the extended debug MBean.

Returns:

int - General return code.

com.ibm.retail.si.mgmt

Interface MgmtHardwareInventoryMBean

All Superinterfaces:
MgmtSimpleInventoryMBean

public interface **MgmtHardwareInventoryMBean**
extends MgmtSimpleInventoryMBean

MBean Interface for hardware inventory. There is no function defined at this point. For now it is acceptable for hardware inventory to be handled by the `MgmtSimpleInventoryMBean`interface.

com.ibm.retail.si.mgmt

Interface MgmtHealthMBean

All Superinterfaces:
MgmtExtendedControlMBean

All Subinterfaces:
MgmtMasterHealthMBean, MgmtClientHealthMBean

---

public interface **MgmtHealthMBean**
extends MgmtExtendedControlMBean

Agent Health MBean interface containing functionality provided by all health MBeans. It extends `MgmtExtendedControlMBean` to expose additional agent functionality.

This management interface includes the following attributes. These attributes are described in more detail in the accessor methods, and in the `MgmtExtendedControlMBean`interface.

- DeviceInfo
- Hostname
- IPAddress
- AgentStarted

The following operations are included in this management interface. These are described in more detail in the corresponding method documentation.
- No Operations Defined

No `Notifications` are emitted by methods in this interface.

See Also:
com.ibm.retail.si.mgmt.MgmtExtendedControlMBean

Method Summary

`MgmtDeviceInfo`	`getDeviceInfo()`
`java.lang.String`	`getHostname()`
`java.lang.String`	`getIPAddress()` On General Agents, the IP Address of the network interface that is sending discovery packets is returned.
`boolean`	`isAgentStarted()`

Methods

getDeviceInfo

`public MgmtDeviceInfo` **`getDeviceInfo`**`()`

(continued from last page)

Returns:

Agent Device information

getHostname

`public java.lang.String` **`getHostname`**`()`

Returns:

Hostname of the machine the agent is running on

getIPAddress

`public java.lang.String` **`getIPAddress`**`()`

On General Agents, the IP Address of the network interface that is sending discovery packets is returned. On Master Agents, the localhost IP address is returned.

Returns:

Discovery interface for general agents, or the localhost address for master agents

isAgentStarted

`public boolean` **`isAgentStarted`**`()`

Returns:

True if the `MgmtAgent`has started, false otherwise

com.ibm.retail.si.mgmt

Interface MgmtJVMEnvironmentMBean

public interface **MgmtJVMEnvironmentMBean**

The purpose of this MBean is to provide specific information regarding the JVM / OS combination that this agent is running within.

The `ObjectName`of this MBean includes the following attributes, in addition to the SIF attribute of `DeviceID`:
- SIFComponent=MGMT
- Id=JVMEnvironment

This management interface includes the following attributes. These attributes are described in more detail in the accessor methods.

- ActiveThreadCount
- ActiveThreadNames
- AvailableProcessors
- ClassPath
- ClassVersion
- EnvSpecName
- EnvSpecVendor
- EnvSpecVersion
- ExtDirs
- FreeMemory
- InstallDirectory
- JITCompilerName
- LibPath
- MaxMemory
- OSArchitecture
- OSName
- OSVersion
- RuntimeVendor
- RuntimeVendorURL
- RuntimeVersion
- TmpPath
- TotalMemory
- VMImplName
- VMImplVendor
- VMImplVersion
- VMSpecName
- VMSpecVendor
- VMSpecVersion

The following operations are included in this management interface. These are described in more detail in the corresponding method documentation.
- executeGC

This MBean emits no `Notifications`

Field Summary

static java.lang.String	`OBJECT_NAME_BASE`

Method Summary

void	`executeGC()` Runs the garbage collector.
int	`getActiveThreadCount()` Returns the number of threads currently active in this JVM.
java.lang.String[]	`getActiveThreadNames()` Returns a list of the names of all of the threads currently active in the JVM.
int	`getAvailableProcessors()` Returns the number of processors that are currently available for use by the JVM.
java.lang.String	`getClassPath()` Returns the Java class path.
java.lang.String	`getClassVersion()` Returns the Java class format version number.
java.lang.String	`getEnvSpecName()` Returns the Java Runtime Environment specification name.
java.lang.String	`getEnvSpecVendor()` Returns the Java Runtime Environment specification vendor.
java.lang.String	`getEnvSpecVersion()` Returns the Java Runtime Environment specification version.
java.lang.String	`getExtDirs()` Returns the Path of extension directory or directories.
long	`getFreeMemory()` Returns the amount of free memory in the Java Virtual Machine.
java.lang.String	`getInstallDirectory()` Returns the Java installation directory.
java.lang.String	`getJITCompilerName()` Returns the Name of JIT compiler to use.
java.lang.String	`getLibPath()` Returns the List of paths to search when loading libraries.
long	`getMaxMemory()` Returns the maximum amount of memory that the Java virtual machine will attempt to use.
java.lang.String	`getOSArchitecture()` Returns the Operating system architecture.
java.lang.String	`getOSName()` Returns the Operating system name.
java.lang.String	`getOSVersion()` Returns the Operating system version.

java.lang.String	getRuntimeVendor() Returns the Java Runtime Environment vendor.
java.lang.String	getRuntimeVendorURL() Returns the Java vendor URL.
java.lang.String	getRuntimeVersion() Returns the Java Runtime Environment version.
java.lang.String	getTmpPath() Returns the Default temp file path.
long	getTotalMemory() Returns the total amount of memory in the Java virtual machine.
java.lang.String	getVMImplName() Returns the Java Virtual Machine implementation name.
java.lang.String	getVMImplVendor() Returns the Java Virtual Machine implementation vendor.
java.lang.String	getVMImplVersion() Returns the Java Virtual Machine implementation version.
java.lang.String	getVMSpecName() Returns the Java Virtual Machine specification name.
java.lang.String	getVMSpecVendor() Returns the Java Virtual Machine specification vendor.
java.lang.String	getVMSpecVersion() Returns the Java Virtual Machine specification version.

Fields

OBJECT_NAME_BASE

`public static final java.lang.String` **`OBJECT_NAME_BASE`**

Methods

getAvailableProcessors

`public int` **`getAvailableProcessors()`**

Returns the number of processors that are currently available for use by the JVM.

Returns:

int - the number of processors

getFreeMemory

`public long getFreeMemory()`

Returns the amount of free memory in the Java Virtual Machine.

Returns:

long

getMaxMemory

`public long getMaxMemory()`

Returns the maximum amount of memory that the Java virtual machine will attempt to use.

Returns:

long

getTotalMemory

`public long getTotalMemory()`

Returns the total amount of memory in the Java virtual machine.

Returns:

long

executeGC

`public void executeGC()`

Runs the garbage collector.

getActiveThreadCount

`public int getActiveThreadCount()`

Returns the number of threads currently active in this JVM.

Returns:

int

getActiveThreadNames

`public java.lang.String[] getActiveThreadNames()`

Returns a list of the names of all of the threads currently active in the JVM.

Returns:

String[]

getRuntimeVersion

`public java.lang.String getRuntimeVersion()`

Returns the Java Runtime Environment version.

(continued from last page)

Returns:

String

getRuntimeVendor

`public java.lang.String` **`getRuntimeVendor`**`()`

Returns the Java Runtime Environment vendor.

Returns:

String

getRuntimeVendorURL

`public java.lang.String` **`getRuntimeVendorURL`**`()`

Returns the Java vendor URL.

Returns:

String

getInstallDirectory

`public java.lang.String` **`getInstallDirectory`**`()`

Returns the Java installation directory.

Returns:

String

getVMSpecVersion

`public java.lang.String` **`getVMSpecVersion`**`()`

Returns the Java Virtual Machine specification version.

Returns:

String

getVMSpecVendor

`public java.lang.String` **`getVMSpecVendor`**`()`

Returns the Java Virtual Machine specification vendor.

Returns:

String

getVMSpecName

`public java.lang.String` **`getVMSpecName`**`()`

Returns the Java Virtual Machine specification name.

Returns:

String

getVMImplVersion

`public java.lang.String getVMImplVersion()`

Returns the Java Virtual Machine implementation version.

Returns:

String

getVMImplVendor

`public java.lang.String getVMImplVendor()`

Returns the Java Virtual Machine implementation vendor.

Returns:

String

getVMImplName

`public java.lang.String getVMImplName()`

Returns the Java Virtual Machine implementation name.

Returns:

String

getEnvSpecVersion

`public java.lang.String getEnvSpecVersion()`

Returns the Java Runtime Environment specification version.

Returns:

String

getEnvSpecVendor

`public java.lang.String getEnvSpecVendor()`

Returns the Java Runtime Environment specification vendor.

Returns:

String

getEnvSpecName

`public java.lang.String getEnvSpecName()`

Returns the Java Runtime Environment specification name.

Returns:

String

(continued from last page)

getClassVersion

`public java.lang.String` **`getClassVersion`**`()`

Returns the Java class format version number.

Returns:

String

getClassPath

`public java.lang.String` **`getClassPath`**`()`

Returns the Java class path.

Returns:

String

getLibPath

`public java.lang.String` **`getLibPath`**`()`

Returns the List of paths to search when loading libraries.

Returns:

String

getTmpPath

`public java.lang.String` **`getTmpPath`**`()`

Returns the Default temp file path.

Returns:

String

getJITCompilerName

`public java.lang.String` **`getJITCompilerName`**`()`

Returns the Name of JIT compiler to use.

Returns:

String

getExtDirs

`public java.lang.String` **`getExtDirs`**`()`

Returns the Path of extension directory or directories.

Returns:

String

(continued from last page)

getOSName

`public java.lang.String getOSName()`

Returns the Operating system name.

Returns:

String

getOSArchitecture

`public java.lang.String getOSArchitecture()`

Returns the Operating system architecture.

Returns:

String

getOSVersion

`public java.lang.String getOSVersion()`

Returns the Operating system version.

Returns:

String

com.ibm.retail.si.mgmt

Interface MgmtMasterHealthMBean

All Superinterfaces:
MgmtHealthMBean, MgmtExtendedControlMBean

public interface **MgmtMasterHealthMBean**
extends MgmtHealthMBean

This interface represents the Master Agent side of Discovery and health checking. Its job is to monitor the clients for discovery and to make sure they are still there. Management interface for the MgmtMasterHealth MBean

This management interface includes the following attributes. These attributes are described in more detail in the accessor methods, and in the `MgmtHealthMBean`interface.

- KnownDeviceList
- StoreId

This MBean defines no operations

An `AgentLostNotification`is emitted by classes implementing this interface when an agent is discovered to have been lost, either by too many missed discovery packets or by the receipt of a `JMXConnectionNotification`of type `JMXConnectionNotification.FAILED`

Field Summary

static int	`ConnectionAttemptInterval` The default time interval between connection attempts of a newly discovered agent
static int	`MissedIntervalThreshold` The number of missed advertisements allowed before being considered offline
static java.lang.String	`OBJECT_NAME`

Method Summary

java.util.Vector	`getKnownDeviceList()`
java.lang.String	`getStoreId()`

Fields

MissedIntervalThreshold

`public static final int` **`MissedIntervalThreshold`**

The number of missed advertisements allowed before being considered offline

ConnectionAttemptInterval

`public static final int` **`ConnectionAttemptInterval`**

The default time interval between connection attempts of a newly discovered agent

OBJECT_NAME

`public static final java.lang.String` **`OBJECT_NAME`**

Methods

getKnownDeviceList

`public java.util.Vector` **`getKnownDeviceList`**`()`

Returns:

A Vector of known active devices (`MgmtDeviceInfo`)

getStoreId

`public java.lang.String` **`getStoreId`**`()`

Returns:

The StoreId for the MasterAgent this MBean is running in

com.ibm.retail.si.mgmt

Interface MgmtSimpleInventoryMBean

All Subinterfaces:
MgmtSoftwareInventoryMBean, SoftwareInventoryMBean, MgmtHardwareInventoryMBean

---

public interface **MgmtSimpleInventoryMBean**

MBean Interface for MgmtSimpleInventory. This interface represents defines the MBean interface for use as the base level component for both software and hardware inventory. It should never be implemented by itself, but should always be used as a base.

This management interface includes the following attributes. These attributes are described in more detail in the accessor methods.

- BuildNumber
- CurrentState
- Description
- FixLevel
- InstallationDate
- MajorVersion
- MinorVersion
- ProductName
- SerialNumber
- Version

This MBean defines no operations

This MBean emits no `Notifications`

Field Summary

static int	INV_ERROR
static int	INV_INSTALLED_FAILED
static int	INV_INSTALLED_NOT_TESTED
static int	INV_INSTALLED_UNKNOWN
static int	INV_NOEXIST
static int	INV_NOTSUPPORTED
static int	INV_OK

Method Summary

java.lang.String	getBuildNumber() Retrieve the buildnumber (if available) that is associated with this component.

int	`getCurrentState()` Test the current state of this component.
java.lang.String	`getDescription()` Retrieve the Description of this component.
java.lang.String	`getFixLevel()` Retrieve the fix level (if applicable) associated with this component.
java.util.Date	`getInstallationDate()` Retrieve the date that this component was installed on the client system.
int	`getMajorVersion()` Retrieve the Major Version number associcated with this component.
java.lang.String	`getManufacturer()` Retrieve the name of the manufacturer of this component.
int	`getMinorVersion()` Retrieve the Minor Version number associated with this component.
java.lang.String	`getProductName()` Retrieve the Product name of this component.
java.lang.String	`getSerialNumber()` Retrieve the serial number (if available) associated with this component.
java.lang.String	`getVersion()` Retrieve the FULL version for this component as a string.

Fields

INV_ERROR

`public static final int` **`INV_ERROR`**

INV_NOEXIST

`public static final int` **`INV_NOEXIST`**

INV_NOTSUPPORTED

`public static final int` **`INV_NOTSUPPORTED`**

INV_INSTALLED_NOT_TESTED

`public static final int` **`INV_INSTALLED_NOT_TESTED`**

INV_INSTALLED_UNKNOWN

`public static final int` **`INV_INSTALLED_UNKNOWN`**

INV_INSTALLED_FAILED

`public static final int` **`INV_INSTALLED_FAILED`**

INV_OK

`public static final int` **`INV_OK`**

Methods

getManufacturer

`public java.lang.String` **`getManufacturer`**`()`

Retrieve the name of the manufacturer of this component.

Returns:

String

getProductName

`public java.lang.String` **`getProductName`**`()`

Retrieve the Product name of this component.

Returns:

String

getDescription

`public java.lang.String` **`getDescription`**`()`

Retrieve the Description of this component.

Returns:

String

getVersion

`public java.lang.String` **`getVersion`**`()`

Retrieve the FULL version for this component as a string. This is meant to be a human readable string that is built up of all of the component parts.

Returns:

String

getMajorVersion

`public int getMajorVersion()`

Retrieve the Major Version number associcated with this component.

Returns:

int

getMinorVersion

`public int getMinorVersion()`

Retrieve the Minor Version number associated with this component.

Returns:

int

getFixLevel

`public java.lang.String getFixLevel()`

Retrieve the fix level (if applicable) associated with this component.

Returns:

String

getBuildNumber

`public java.lang.String getBuildNumber()`

Retrieve the buildnumber (if available) that is associated with this component.

Returns:

int

getSerialNumber

`public java.lang.String getSerialNumber()`

Retrieve the serial number (if available) associated with this component.

Returns:

String

getInstallationDate

`public java.util.Date getInstallationDate()`

Retrieve the date that this component was installed on the client system.

Returns:

Date

(continued from last page)

getCurrentState

`public int` **`getCurrentState`**`()`

Test the current state of this component.

Returns:

int - see the INV_xxx statics defined within this class for appropriate return states.

com.ibm.retail.si.mgmt

Interface MgmtSoftwareInventoryMBean

All Superinterfaces:
MgmtSimpleInventoryMBean

All Subinterfaces:
SoftwareInventoryMBean

---

public interface **MgmtSoftwareInventoryMBean**
extends MgmtSimpleInventoryMBean

This interface represents defines the MBean interface for use in collecting software inventory.

This management interface includes the following attributes. These attributes are described in more detail in the accessor methods.

- ComponentCount
- Components
- InstalledPath
- PackageSize

This MBean defines no operations

This MBean emits no `Notifications`

Method Summary

`int`	`getComponentCount()` Get the number of individual files that make up this package.
`MgmtSftComponent[]`	`getComponents()` Retrieve an array containing the information about every file that makes up this package.
`java.lang.String`	`getInstalledPath()` Retrieve the installed path for this piece of software.
`long`	`getPackageSize()` Retrieve the total amount of storage on the client consumed by this package.

Methods

getInstalledPath

`public java.lang.String` **`getInstalledPath()`**

Retrieve the installed path for this piece of software.

Returns:

String, the installation path.

getPackageSize

`public long getPackageSize()`

Retrieve the total amount of storage on the client consumed by this package.

Returns:

long - size in bytes.

getComponentCount

`public int getComponentCount()`

Get the number of individual files that make up this package.

Returns:

int

getComponents

`public MgmtSftComponent[] getComponents()`

Retrieve an array containing the information about every file that makes up this package.

Returns:

MgmtSftComponent

com.ibm.retail.si.mgmt

Class ObjectNameFactory

```
java.lang.Object
   |
   +-com.ibm.retail.si.mgmt.ObjectNameFactory
```

public class **ObjectNameFactory**
extends java.lang.Object

Factory for creating custom ObjectNames for Regular and Proxied MBeans. ObjectNames are passed in and copied, adding and/or replacing additional attributes. The static initialize method should be called before getting the singleton instance for the first time in order to pass in configuration information

Constructor Summary

`ObjectNameFactory(java.lang.String deviceId,java.lang.String deviceId,java.lang.String deviceId)`

Method Summary

`javax.management.ObjectName`	`createObjectName(javax.management.ObjectName objName,java.lang.String objName,java.lang.String objName,java.lang.String objName)` **Deprecated.** *The Id key needs to be supplied, use createObjectName(ObjectName, String, String, String) instead*
`javax.management.ObjectName`	`createObjectName(javax.management.ObjectName objName,java.lang.String objName,java.lang.String objName,java.lang.String objName,java.lang.String objName)` Creates a new `ObjectName`from an existing `ObjectName`that conforms to the SI Object Naming conventions, adding the supplied system information in addition to other required properties.
`javax.management.ObjectName`	`createObjectName(java.lang.String domain,java.lang.String domain,java.lang.String domain,java.lang.String domain,java.lang.String domain)` Creates a new `ObjectName`from the supplied information that conforms to the SI Object Naming conventions, having the supplied system information in addition to other required properties.
`javax.management.ObjectName`	`createProxyObjectName(javax.management.ObjectName objName,MgmtDeviceInfo objName)` Create a modified ObjectName for a proxy of the supplied ObjectName
`static ObjectNameFactory`	`getInstance()`
`static void`	`initialize(java.lang.String deviceId,java.lang.String deviceId,java.lang.String deviceId)` Initializes the singleton instance with the supplied attributes.
`javax.management.ObjectName`	`reverseObjectName(javax.management.ObjectName objName)` For usage within a general agent, this method, removes all device and component information added in createObjectName()

Methods inherited from : class `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
wait
```

Constructors

ObjectNameFactory

```
protected ObjectNameFactory(java.lang.String deviceId,
                            java.lang.String systemId,
                            java.lang.String storeId)
```

Parameters:

`deviceId -`
Device ID added only to regular ObjectNames
`systemId -`
System ID added only to regular ObjectNames
`storeId -`
Store ID to add to proxied MBeans, should be null for GA's

Methods

createProxyObjectName

```
public javax.management.ObjectName createProxyObjectName(javax.management.ObjectName
objName,
                                                         MgmtDeviceInfo devInfo)
                                                   throws
javax.management.MalformedObjectNameException
```

Create a modified ObjectName for a proxy of the supplied ObjectName

Parameters:

`objName -`
Source ObjectName
`devInfo -`
MgmtDeviceInfo matching the source device

Returns:

ObjectName for a proxy of the source instance, having new attributes and values based on parameters supplied to this factory

Exceptions:

`MalformedObjectNameException -`
Error creating new ObjectName

createObjectName

```
public javax.management.ObjectName createObjectName(javax.management.ObjectName
objName,
                                                    java.lang.String component,
                                                    java.lang.String devMajor,
                                                    java.lang.String devMinor)
                                              throws
javax.management.MalformedObjectNameException
```

Deprecated. *The Id key needs to be supplied, use createObjectName(ObjectName, String, String, String) instead*

(continued from last page)

For usage within a general agent, this method sreates a new ObjectName for a regular MBean, adding device and system information in addition to the supplied component and version information

Parameters:

`objName -`
Source ObjectName instance
`component -`
Component name the MBean matches
`devMajor -`
Major version of the component being represented
`devMinor -`
Minor version of the component being represented

Returns:

New ObjectName based on the supplied instance with additional information

Exceptions:

`MalformedObjectNameException -`
Invalid parameter values supplied

createObjectName

```
public javax.management.ObjectName createObjectName(javax.management.ObjectName
objName,
                                                    java.lang.String id,
                                                    java.lang.String component,
                                                    java.lang.String devMajor,
                                                    java.lang.String devMinor)
                                             throws
javax.management.MalformedObjectNameException
```

Creates a new `ObjectName`from an existing `ObjectName`that conforms to the SI Object Naming conventions, adding the supplied system information in addition to other required properties.

Parameters:

`objName -`
Source `ObjectName`instance
`id -`
Type/Identifier for the MBean, cannot be null
`component -`
Component name the MBean matches, can be null
`devMajor -`
Major version of the component being represented, can be null
`devMinor -`
Minor version of the component being represented, can be null

Returns:

New `ObjectName`based on the supplied instance with additional information

Exceptions:

`MalformedObjectNameException -`
Invalid parameter values supplied

createObjectName

```
public javax.management.ObjectName createObjectName(java.lang.String domain,
                                                    java.lang.String id,
                                                    java.lang.String component,
                                                    java.lang.String devMajor,
                                                    java.lang.String devMinor)
                                             throws
javax.management.MalformedObjectNameException
```

Creates a new `ObjectName`from the supplied information that conforms to the SI Object Naming conventions, having the supplied system information in addition to other required properties.

(continued from last page)

Parameters:

`domain -`
Domain portion of the new `ObjectName`
`id -`
Type/Identifier for the MBean, cannot be null
`component -`
Component name the MBean matches, can be null
`devMajor -`
Major version of the component being represented, can be null
`devMinor -`
Minor version of the component being represented, can be null

Returns:

New ObjectName based on the supplied instance with additional information

Exceptions:

`MalformedObjectNameException -`
Invalid parameter values supplied

reverseObjectName

```
public javax.management.ObjectName reverseObjectName(javax.management.ObjectName
objName)
                                                    throws
javax.management.MalformedObjectNameException
```

For usage within a general agent, this method, removes all device and component information added in createObjectName()

Parameters:

`objName -`
Source ObjectName

Returns:

ObjectName with device and component information removed

Exceptions:

`MalformedObjectNameException -`
Error creating new ObjectName

initialize

```
protected static void initialize(java.lang.String deviceId,
                                 java.lang.String systemId,
                                 java.lang.String storeId)
```

Initializes the singleton instance with the supplied attributes. Some

Parameters:

`deviceId -`
Device ID added only to regular ObjectNames
`systemId -`
System ID added only to regular ObjectNames
`storeId -`
Store ID to add to proxied MBeans, should be null for GA's

getInstance

```
public static ObjectNameFactory getInstance()
                                      throws MgmtException
```

(continued from last page)

Returns:

Singleton instance of this class. If a singleton hasn't been initialized via the initialize() method, then a MgmtException will be thrown

Exceptions:

`MgmtException -`
Singleton instance has not been initialized

Package
com.ibm.retail.si.mgmt.generalagent

Provides the classes required for a General Agent

com.ibm.retail.si.mgmt.generalagent

Interface ServerSocketConnectionMBean

public interface **ServerSocketConnectionMBean**

Method Summary

void	`close()`
void	`open(java.lang.String jndiName)`

Methods

open

```
public void open(java.lang.String jndiName)
          throws java.lang.Exception
```

close

```
public void close()
           throws java.lang.Exception
```

com.ibm.retail.si.mgmt.generalagent

Interface TestMBean

public interface **TestMBean**

Field Summary

static java.lang.String	OBJECT_NAME

Method Summary

void	decrement()
java.lang.Integer	getCounter()
java.lang.Integer	getTestInt()
java.lang.String	getTestString()
void	increment()
void	setCounter(java.lang.Integer count)
void	setTestInt(java.lang.Integer testInt)
void	setTestString(java.lang.String testString)

Fields

OBJECT_NAME

public static final java.lang.String **OBJECT_NAME**

Methods

increment

public void **increment**()

decrement

public void **decrement**()

(continued from last page)

getCounter

```
public java.lang.Integer getCounter()
```

setCounter

```
public void setCounter(java.lang.Integer count)
```

getTestInt

```
public java.lang.Integer getTestInt()
```

setTestInt

```
public void setTestInt(java.lang.Integer testInt)
```

getTestString

```
public java.lang.String getTestString()
```

setTestString

```
public void setTestString(java.lang.String testString)
```

Package
com.ibm.retail.si.mgmt.logging

MBean interfaces and classes for logging control of JDK, Log4J and Syslog.

The remote logging MBeans forward logging events as `RtlTracePointNotifications`

com.ibm.retail.si.mgmt.logging

Class JDKHandlerMBean

```
java.lang.Object
  |
  +-com.ibm.retail.si.mgmt.logging.JDKHandlerMBean
```

All Implemented interfaces:
javax.management.DynamicMBean

---

public class **JDKHandlerMBean**
extends java.lang.Object
implements javax.management.DynamicMBean

MBean for making non persistent changes to the logging levels on JDK Logging Handlers

The `ObjectName`of this MBean includes the following attributes, in addition to the SIF attribute of `DeviceID`:
- SIFComponent=MGMT
- Id=JDKHandlers

This management interface has a dynamic list of attributes, each of which is the class name of the `Handler`, and whose value is the `Handler`'s level.

This MBean defines no operations

This MBean emits no `Notifications`

Field Summary

static java.util.logging.LogManager	manager
static java.lang.String	OBJECT_NAME_BASE

Constructor Summary

`JDKHandlerMBean()` Constructor

Method Summary

java.lang.Object	getAttribute(java.lang.String attributeName)
javax.management.AttributeList	getAttributes(java.lang.String[] attributeNames)
java.util.logging.Handler	getHandler(java.lang.String handlerName) Obtains the `Handler`instance from the root logger based on the supplied class name

java.util.ArrayList	getHandlerNames() Returns an ArrayList of all current Handlers' classnames.
javax.management.MBeanInfo	getMBeanInfo()
java.lang.Object	invoke(java.lang.String operation,java.lang.Object[] operation,java.lang.String[] operation)
void	setAttribute(javax.management.Attribute newAttribute)
javax.management.AttributeList	setAttributes(javax.management.AttributeList attrs)
void	setHandlerLevel(java.lang.String handlerName,java.lang.String handlerName) Sets the level of a Handler.

Methods inherited from : class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

OBJECT_NAME_BASE

```
public static final java.lang.String OBJECT_NAME_BASE
```

manager

```
protected static java.util.logging.LogManager manager
```

Constructors

JDKHandlerMBean

```
public JDKHandlerMBean()
```

Constructor

Methods

setHandlerLevel

```
protected void setHandlerLevel(java.lang.String handlerName,
                               java.lang.String levelStr)
                        throws MgmtException,
                               javax.management.InvalidAttributeValueException
```

Sets the level of a Handler.

(continued from last page)

Parameters:

`handlerName -`
Class name of the `Handler`to change
`level -`
Valid levels are: SEVERE, WARNING, INFO, CONFIG, FINE, FINER, FINEST

Exceptions:

`MgmtException -`
No `Handler`matching the supplied class name, or a null or invalid level was specified

getHandlerNames

```
protected java.util.ArrayList getHandlerNames()
```

Returns an ArrayList of all current `Handlers`' classnames.

Returns:

ArrayList of the current `Handlers`' classnames.

getHandler

```
protected java.util.logging.Handler getHandler(java.lang.String handlerName)
```

Obtains the `Handler`instance from the root logger based on the supplied class name

Parameters:

`handlerName -`
Class name of the `Handler`

Returns:

Handler from the root logger, or null of none exists

getAttribute

```
public java.lang.Object getAttribute(java.lang.String attributeName)
                              throws javax.management.AttributeNotFoundException,
                                     javax.management.MBeanException,
                                     javax.management.ReflectionException
```

See Also:

DynamicMBean#getAttribute(java.lang.String)

getAttributes

```
public javax.management.AttributeList getAttributes(java.lang.String[] attributeNames)
```

See Also:

DynamicMBean#getAttributes(java.lang.String[])

invoke

```
public java.lang.Object invoke(java.lang.String operation,
                               java.lang.Object[] params,
                               java.lang.String[] signature)
```

(continued from last page)

Returns:

null because this MBean has no operations defined

See Also:

DynamicMBean#invoke(java.lang.String, java.lang.Object[], java.lang.String[])

setAttribute

```
public void setAttribute(javax.management.Attribute newAttribute)
                  throws javax.management.AttributeNotFoundException,
                         javax.management.InvalidAttributeValueException,
                         javax.management.MBeanException,
                         javax.management.ReflectionException
```

See Also:

DynamicMBean#setAttribute(javax.management.Attribute)

setAttributes

```
public javax.management.AttributeList setAttributes(javax.management.AttributeList
attrs)
```

See Also:

DynamicMBean#setAttributes(javax.management.AttributeList)

getMBeanInfo

```
public javax.management.MBeanInfo getMBeanInfo()
```

com.ibm.retail.si.mgmt.logging

Class JDKLoggerMBean

```
java.lang.Object
   |
   +-com.ibm.retail.si.mgmt.logging.JDKLoggerMBean
```

All Implemented interfaces:
javax.management.DynamicMBean

---

public class **JDKLoggerMBean**
extends java.lang.Object
implements javax.management.DynamicMBean

Dynamic MBean that allows for dynamic, non persistent changes to JDK Logger logging levels.

The `ObjectName`of this MBean includes the following attributes, in addition to the SIF attribute of `DeviceID`:
- SIFComponent=MGMT
- Id=JDKLoggers

This management interface has a dynamic list of attributes, each of which is the name of the `Logger`, and whose value is the `Logger`'s level.

The following operations are included in this management interface. These are described in more detail in the corresponding method documentation.
- readConfiguration

This MBean emits no `Notifications`

Field Summary

static java.util.logging.LogManager	manager
static java.lang.String	OBJECT_NAME_BASE

Constructor Summary

JDKLoggerMBean() Constructor

Method Summary

java.lang.Object	getAttribute(java.lang.String attributeName)
javax.management.AttributeList	getAttributes(java.lang.String[] attributeNames)
java.util.ArrayList	getLoggerNames() Returns an ArrayList of the current logger names.

javax.management.MBeanInfo	getMBeanInfo()
java.lang.Object	invoke(java.lang.String operation,java.lang.Object[] operation,java.lang.String[] operation)
void	readConfiguration() Reload the configuration file for logger properties
void	setAttribute(javax.management.Attribute newAttribute)
javax.management.AttributeList	setAttributes(javax.management.AttributeList attrs)
void	setLoggerLevel(java.lang.String loggerName,java.lang.String loggerName) Sets the level of a logger.

Methods inherited from : class `java.lang.Object`
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

OBJECT_NAME_BASE

```
public static final java.lang.String OBJECT_NAME_BASE
```

manager

```
protected static java.util.logging.LogManager manager
```

Constructors

JDKLoggerMBean

```
public JDKLoggerMBean()
```

Constructor

Methods

setLoggerLevel

```
protected void setLoggerLevel(java.lang.String loggerName,
                              java.lang.String levelStr)
                       throws MgmtException,
                              javax.management.InvalidAttributeValueException
```

Sets the level of a logger.

Parameters:

(continued from last page)

`loggerName`
`level -`
Valid levels are: SEVERE, WARNING, INFO, CONFIG, FINE, FINER, FINEST

getLoggerNames

```
protected java.util.ArrayList getLoggerNames()
```

Returns an ArrayList of the current logger names.

Returns:

ArrayList of the current logger names.

readConfiguration

```
public void readConfiguration()
                        throws java.io.IOException
```

Reload the configuration file for logger properties

Exceptions:
`java.io.IOException`

getAttribute

```
public java.lang.Object getAttribute(java.lang.String attributeName)
                              throws javax.management.AttributeNotFoundException,
                                     javax.management.MBeanException,
                                     javax.management.ReflectionException
```

See Also:

DynamicMBean#getAttribute(java.lang.String)

getAttributes

```
public javax.management.AttributeList getAttributes(java.lang.String[] attributeNames)
```

See Also:

DynamicMBean#getAttributes(java.lang.String[])

invoke

```
public java.lang.Object invoke(java.lang.String operation,
                               java.lang.Object[] params,
                               java.lang.String[] signature)
```

See Also:

DynamicMBean#invoke(java.lang.String, java.lang.Object[], java.lang.String[])

(continued from last page)

setAttribute

```
public void setAttribute(javax.management.Attribute newAttribute)
                  throws javax.management.AttributeNotFoundException,
                         javax.management.InvalidAttributeValueException,
                         javax.management.MBeanException,
                         javax.management.ReflectionException
```

See Also:

DynamicMBean#setAttribute(javax.management.Attribute)

setAttributes

```
public javax.management.AttributeList setAttributes(javax.management.AttributeList
attrs)
```

See Also:

DynamicMBean#setAttributes(javax.management.AttributeList)

getMBeanInfo

```
public javax.management.MBeanInfo getMBeanInfo()
```

com.ibm.retail.si.mgmt.logging

Interface JDKLoggingMBean

public interface **JDKLoggingMBean**

Extension of MgmtLoggingCtrlMBean for controlling JDK Logging

com.ibm.retail.si.mgmt.logging

Class Log4JLoggerMBean

```
java.lang.Object
   |
   +-com.ibm.retail.si.mgmt.logging.Log4JLoggerMBean
```

All Implemented interfaces:
javax.management.DynamicMBean

---

public class **Log4JLoggerMBean**
extends java.lang.Object
implements javax.management.DynamicMBean

Dynamic MBean that allows for dynamic, non persistent changes to Log4J Logger logging levels.

The `ObjectName`of this MBean includes the following attributes, in addition to the SIF attribute of `DeviceID`:
- SIFComponent=MGMT
- Id=Log4JLoggers

This management interface has a dynamic list of attributes, each of which is the name of the `Logger`, and whose value is the `Logger`'s level.

The following operations are included in this management interface. These are described in more detail in the corresponding method documentation.
- resetConfiguration

This MBean emits no `Notifications`

Field Summary

static java.lang.String	OBJECT_NAME_BASE

Constructor Summary

Log4JLoggerMBean() Constructor

Method Summary

java.lang.Object	getAttribute(java.lang.String attributeName)
javax.management.AttributeList	getAttributes(java.lang.String[] attributeNames)
java.util.ArrayList	getLoggerNames() Returns an ArrayList of the current logger names.
javax.management.MBeanInfo	getMBeanInfo()

java.lang.Object	invoke(java.lang.String operation,java.lang.Object[] operation,java.lang.String[] operation)
void	resetConfiguration() Reload the configuration file for logger properties
void	setAttribute(javax.management.Attribute newAttribute)
javax.management.AttributeList	setAttributes(javax.management.AttributeList attrs)
void	setLoggerLevel(java.lang.String loggerName,java.lang.String loggerName) Sets the level of a logger.

Methods inherited from : class `java.lang.Object`
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

OBJECT_NAME_BASE

```
public static final java.lang.String OBJECT_NAME_BASE
```

Constructors

Log4JLoggerMBean

```
public Log4JLoggerMBean()
```

Constructor

Methods

setLoggerLevel

```
protected void setLoggerLevel(java.lang.String loggerName,
                              java.lang.String levelStr)
                       throws MgmtException
```

Sets the level of a logger.

Parameters:

`loggerName` -
Name of logger to set level for
`level` -
Valid levels are: FATAL, ERROR, WARN, INFO, DEBUG, ALL, OFF

getLoggerNames

```
protected java.util.ArrayList getLoggerNames()
```

Returns an ArrayList of the current logger names.

(continued from last page)

Returns:

ArrayList of the current logger names.

resetConfiguration

```
public void resetConfiguration()
```

Reload the configuration file for logger properties

getAttribute

```
public java.lang.Object getAttribute(java.lang.String attributeName)
                              throws javax.management.AttributeNotFoundException,
                                     javax.management.MBeanException,
                                     javax.management.ReflectionException
```

See Also:

DynamicMBean#getAttribute(java.lang.String)

getAttributes

```
public javax.management.AttributeList getAttributes(java.lang.String[] attributeNames)
```

See Also:

DynamicMBean#getAttributes(java.lang.String[])

invoke

```
public java.lang.Object invoke(java.lang.String operation,
                               java.lang.Object[] params,
                               java.lang.String[] signature)
```

See Also:

DynamicMBean#invoke(java.lang.String, java.lang.Object[], java.lang.String[])

setAttribute

```
public void setAttribute(javax.management.Attribute newAttribute)
                  throws javax.management.AttributeNotFoundException,
                         javax.management.InvalidAttributeValueException,
                         javax.management.MBeanException,
                         javax.management.ReflectionException
```

See Also:

DynamicMBean#setAttribute(javax.management.Attribute)

(continued from last page)

setAttributes

```
public javax.management.AttributeList setAttributes(javax.management.AttributeList
attrs)
```

See Also:

DynamicMBean#setAttributes(javax.management.AttributeList)

getMBeanInfo

```
public javax.management.MBeanInfo getMBeanInfo()
```

com.ibm.retail.si.mgmt.logging

Interface Log4JLoggingMBean

public interface **Log4JLoggingMBean**

Extension of MgmtLoggingCtrlMBean for controlling Log4J logging

com.ibm.retail.si.mgmt.logging

Class MgmtLoggingCtrlMBean

```
java.lang.Object
  |
  +-com.ibm.retail.si.mgmt.logging.MgmtLoggingCtrlMBean
```

All Implemented interfaces:
javax.management.DynamicMBean

---

public class **MgmtLoggingCtrlMBean**
extends java.lang.Object
implements javax.management.DynamicMBean

Dynamic MBean that instantiates all applicable Logging MBeans for JDK Logging, Log4J, or Syslog. Agents with a device type of `MgmtConst.dTypeConsumer`or `MgmtConst.dTypeROLO`start the RemoteSyslogLoggingCtrlMBean.

The `ObjectName`of this MBean includes the following attributes, in addition to the SIF attribute of `DeviceID`:
- SIFComponent=MGMT
- Id=MgmtLoggingCtrl

This MBean has no attributes.

The following operations are included in this management interface. These are described in more detail in the corresponding method documentation.
- start
- stop

This MBean emits no `Notifications`

Field Summary

static java.lang.String	OBJECT_NAME_BASE

Constructor Summary

MgmtLoggingCtrlMBean(MgmtAgent mgmtAgent)

Method Summary

java.lang.Object	getAttribute(java.lang.String attributeName)
javax.management.AttributeList	getAttributes(java.lang.String[] attributeNames)
javax.management.MBeanInfo	getMBeanInfo()
java.lang.Object	invoke(java.lang.String operation,java.lang.Object[] operation,java.lang.String[] operation)

void	setAttribute(javax.management.Attribute newAttribute)
javax.management.AttributeList	setAttributes(javax.management.AttributeList attrs)
void	start()
void	stop()

Methods inherited from : class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

OBJECT_NAME_BASE

```
public static final java.lang.String OBJECT_NAME_BASE
```

Constructors

MgmtLoggingCtrlMBean

```
public MgmtLoggingCtrlMBean(MgmtAgent mgmtAgent)
```

Methods

start

```
public void start()
           throws MgmtException
```

stop

```
public void stop()
```

getAttribute

```
public java.lang.Object getAttribute(java.lang.String attributeName)
                                throws javax.management.AttributeNotFoundException,
                                       javax.management.MBeanException,
                                       javax.management.ReflectionException
```

See Also:

DynamicMBean#getAttribute(java.lang.String)

getAttributes

```
public javax.management.AttributeList getAttributes(java.lang.String[] attributeNames)
```

See Also:

DynamicMBean#getAttributes(java.lang.String[])

invoke

```
public java.lang.Object invoke(java.lang.String operation,
                               java.lang.Object[] params,
                               java.lang.String[] signature)
```

See Also:

DynamicMBean#invoke(java.lang.String, java.lang.Object[], java.lang.String[])

setAttribute

```
public void setAttribute(javax.management.Attribute newAttribute)
                  throws javax.management.AttributeNotFoundException,
                         javax.management.InvalidAttributeValueException,
                         javax.management.MBeanException,
                         javax.management.ReflectionException
```

See Also:

DynamicMBean#setAttribute(javax.management.Attribute)

setAttributes

```
public javax.management.AttributeList setAttributes(javax.management.AttributeList
attrs)
```

See Also:

DynamicMBean#setAttributes(javax.management.AttributeList)

getMBeanInfo

```
public javax.management.MBeanInfo getMBeanInfo()
```

com.ibm.retail.si.mgmt.logging

Interface RemoteJDKLoggingCtrlMBean

All Superinterfaces:
RemoteLoggingCtrlMBean

public interface **RemoteJDKLoggingCtrlMBean**
extends RemoteLoggingCtrlMBean

Extension of `RemoteLoggingCtrlMBean`for controlling the forwarding of JDK logging events as `RtlTracePointNotifications`

The `ObjectName`of this MBean includes the following attributes, in addition to the SIF attribute of `DeviceID`:
- SIFComponent=MGMT
- Id=RemoteJDKLogCtrl

This management interface includes the following attributes. These attributes are described in more detail in the accessor methods, as well as others defined in the `RemoteLoggingCtrlMBean`interface.
- pushInterval

There are no operations are included in this management interface, but there are operations defined in the `RemoteLoggingCtrlMBean`interface.

This MBean emits a `RtlTracePointNotification`for each logging event it is configured to forward

See Also:
com.ibm.retail.si.mgmt.logging.RemoteLoggingCtrlMBean

Field Summary

static java.lang.String	OBJECT_NAME_BASE

Method Summary

int	getPushInterval()
void	setPushInterval(int pushInterval)

Fields

OBJECT_NAME_BASE

`public static final java.lang.String` **`OBJECT_NAME_BASE`**

Methods

(continued from last page)

getPushInterval

```
public int getPushInterval()
```

Returns:

The time in seconds between pushes of the remote logging `MemoryHandler`. This value should be set to a higher value before setting the remote logging level to `LEVEL_MASK_ALL`, `LEVEL_DEBUG`or `LEVEL_INFO`

See Also:

java.util.logging.MemoryHandler

setPushInterval

```
public void setPushInterval(int pushInterval)
                     throws javax.management.InvalidAttributeValueException
```

Parameters:

`pushInterval -`
New push interval value, in seconds

Exceptions:

`InvalidAttributeValueException -`
If the new value is less than or equal to 0

See Also:

#getPushInterval()

com.ibm.retail.si.mgmt.logging

Interface RemoteLog4JLoggingCtrlMBean

All Superinterfaces:
RemoteLoggingCtrlMBean

public interface **RemoteLog4JLoggingCtrlMBean**
extends RemoteLoggingCtrlMBean

Extension of `RemoteLoggingCtrlMBean`for controlling the forwarding of Log4J logging events as `RtlTracePointNotifications`

The `ObjectName`of this MBean includes the following attributes, in addition to the SIF attribute of `DeviceID`:
- SIFComponent=MGMT
- Id=RemoteLog4JLogCtrl

This management interface defines no attributes.

There are no operations are included in this management interface, but there are operations defined in the `RemoteLoggingCtrlMBean`interface.

This MBean emits a `RtlTracePointNotification`for each logging event it is configured to forward

See Also:
com.ibm.retail.si.mgmt.logging.RemoteLoggingCtrlMBean

Field Summary

static java.lang.String	OBJECT_NAME_BASE

Fields

OBJECT_NAME_BASE

`public static final java.lang.String` **`OBJECT_NAME_BASE`**

com.ibm.retail.si.mgmt.logging

Interface RemoteLoggingCtrlMBean

All Subinterfaces:
RemoteSyslogLoggingCtrlMBean, RemoteLog4JLoggingCtrlMBean, RemoteJDKLoggingCtrlMBean

---

public interface **RemoteLoggingCtrlMBean**

Management interface for controlling the forwarding of logging events from a particular logging implementation as `RtlTracePointNotifications`. This interface should not be instantiated but instead be extended for each logging implementation. Each implementation should map each of the remote logging levels listed below to a level in their implementation.

This management interface includes the following attributes. These attributes are described in more detail in the accessor methods.

- Level
- LogType
- Active

The following operations are included in this management interface. These are described in more detail in the corresponding method documentation.
- start
- stop

Classes that implement this interface emit a `RtlTracePointNotification`for each logging event it is configured to forward.

Field Summary

static int	LEVEL_ALERT
static int	LEVEL_CRIT
static int	LEVEL_DEBUG
static int	LEVEL_EMERGENCY
static int	LEVEL_ERR
static int	LEVEL_INFO
static int	LEVEL_MASK_ALL
static int	LEVEL_MASK_OFF
static int	LEVEL_MASK_SEVERE
static int	LEVEL_NOTICE
static int	LEVEL_WARNING

static int[]	LEVELS

Method Summary

java.lang.Integer	getLevel() Returns a mask indicating the current level of logging that is enabled.
java.lang.String	getLogType() Returns a sting to indicate the type of logger that is managed by this interface.
boolean	isActive() Returns a string to indicate the type of logger that is managed by this interface.
void	setLevel(java.lang.Integer LevelMask) Returns a mask indicating the current level of logging that is enabled.
boolean	start() Causes the logger that is managed by this MBean to start logging.
boolean	stop() Causes the logger that is managed by this MBean to stop logging.

Fields

LEVEL_EMERGENCY

public static final int **LEVEL_EMERGENCY**

LEVEL_ALERT

public static final int **LEVEL_ALERT**

LEVEL_CRIT

public static final int **LEVEL_CRIT**

LEVEL_ERR

public static final int **LEVEL_ERR**

LEVEL_WARNING

public static final int **LEVEL_WARNING**

(continued from last page)

LEVEL_NOTICE

```
public static final int LEVEL_NOTICE
```

LEVEL_INFO

```
public static final int LEVEL_INFO
```

LEVEL_DEBUG

```
public static final int LEVEL_DEBUG
```

LEVEL_MASK_ALL

```
public static final int LEVEL_MASK_ALL
```

LEVEL_MASK_OFF

```
public static final int LEVEL_MASK_OFF
```

LEVEL_MASK_SEVERE

```
public static final int LEVEL_MASK_SEVERE
```

LEVELS

```
public static final int LEVELS
```

Methods

setLevel

```
public void setLevel(java.lang.Integer LevelMask)
              throws javax.management.InvalidAttributeValueException
```

Returns a mask indicating the current level of logging that is enabled.

Parameters:

`Integer -`
LevelMask - The new logging level mask to apply to this logger. The definitions for these values are defined as part of this interface.

Exceptions:

`InvalidAttributeValueException -`
An attempt is made to set an undefined level or mask

getLevel

```
public java.lang.Integer getLevel()
```

(continued from last page)

Returns a mask indicating the current level of logging that is enabled.

Returns:

Integer RC - A mask indicating the current level of logging. See the Mask values defined for this interface.

isActive

`public boolean **isActive**()`

Returns a string to indicate the type of logger that is managed by this interface.

Returns:

boolean - Return code indicating that the logging is or is not currenly forwarding information to the JMX infrastructure.

getLogType

`public java.lang.String **getLogType**()`

Returns a sting to indicate the type of logger that is managed by this interface.

Returns:

String - The name of the logger.

start

`public boolean **start**()`

Causes the logger that is managed by this MBean to start logging. The result is that assuming the entries pass the set level filter, they are forwarded as notifications.

Returns:

boolean - The result of the operation. True if the MBean was started successfully, false if there was an error starting or the MBean was already started

stop

`public boolean **stop**()`

Causes the logger that is managed by this MBean to stop logging. The result is that NO log entries are forwarded as notifications.

Returns:

boolean - The result of the operation. True if the MBean was stopped successfully, false if there was an error stopping or the MBean was already stopped.

com.ibm.retail.si.mgmt.logging

Interface RemoteSyslogLoggingCtrlMBean

All Superinterfaces:
RemoteLoggingCtrlMBean

public interface **RemoteSyslogLoggingCtrlMBean**
extends RemoteLoggingCtrlMBean

Extension of `RemoteLoggingCtrlMBean`for controlling the forwarding of Syslog logging events as `RtlTracePointNotifications`

The `ObjectName`of this MBean includes the following attributes, in addition to the SIF attribute of `DeviceID`:
- SIFComponent=MGMT
- Id=RemoteSysLogCtrl

This management interface defines no attributes.

There are no operations are included in this management interface, but there are operations defined in the `RemoteLoggingCtrlMBean`interface.

This MBean emits a `RtlTracePointNotification`for each logging event it is configured to forward

See Also:
com.ibm.retail.si.mgmt.logging.RemoteLoggingCtrlMBean

Field Summary

static java.lang.String	`OBJECT_NAME_BASE`
static int	`SYSL_FAC_CLOCK_DAEMON_0` Clock daemon.
static int	`SYSL_FAC_CLOCK_DAEMON_1` Clock daemon.
static int	`SYSL_FAC_FTPD` FTP Daemon
static int	`SYSL_FAC_KERNEL` Kernel messages
static int	`SYSL_FAC_LCL_USE_0` Local use 0
static int	`SYSL_FAC_LCL_USE_1` Local use 1
static int	`SYSL_FAC_LCL_USE_2` Local use 2

static int	`SYSL_FAC_LCL_USE_3` Local use 3
static int	`SYSL_FAC_LCL_USE_4` Local use 4
static int	`SYSL_FAC_LCL_USE_5` Local use 5
static int	`SYSL_FAC_LCL_USE_6` Local use 6
static int	`SYSL_FAC_LCL_USE_7` Local use 7
static int	`SYSL_FAC_LINE_PTR_SYS` Line printer subsystem
static int	`SYSL_FAC_LOG_ALERT` Security/authorization messages.
static int	`SYSL_FAC_LOG_AUDIT` Log audit.
static int	`SYSL_FAC_MAIL_SYS` Mail system
static int	`SYSL_FAC_NNTP_SYS` Network news subsystem
static int	`SYSL_FAC_NTP_SYS` NTP subsystem
static int	`SYSL_FAC_SEC_AUTH_0` Security/authorization messages.
static int	`SYSL_FAC_SEC_AUTH_1` Security/authorization messages.
static int	`SYSL_FAC_SYS_DAEMONS` System daemons
static int	`SYSL_FAC_SYSLOGD` Messages generated internally by syslogd
static int	`SYSL_FAC_USR` User-level messages
static int	`SYSL_FAC_UUCP_SYS` UUCP subsystem
static int	`SYSL_LOG_ALERT` Condition needing immediate attention

static int	SYSL_LOG_CRIT Critical conditions
static int	SYSL_LOG_DEBUG Level for when debugging a system
static int	SYSL_LOG_EMERG Kernel panic
static int	SYSL_LOG_ERR Errors
static int	SYSL_LOG_INFO Informational messages
static int	SYSL_LOG_NOTICE Not an error, but may need attention
static int	SYSL_LOG_WARNING Warning messages

Fields

OBJECT_NAME_BASE

`public static final java.lang.String` **`OBJECT_NAME_BASE`**

SYSL_LOG_EMERG

`public static final int` **`SYSL_LOG_EMERG`**

Kernel panic

SYSL_LOG_ALERT

`public static final int` **`SYSL_LOG_ALERT`**

Condition needing immediate attention

SYSL_LOG_CRIT

`public static final int` **`SYSL_LOG_CRIT`**

Critical conditions

SYSL_LOG_ERR

`public static final int` **`SYSL_LOG_ERR`**

Errors

SYSL_LOG_WARNING

`public static final int` **`SYSL_LOG_WARNING`**

(continued from last page)

Warning messages

SYSL_LOG_NOTICE

`public static final int` **`SYSL_LOG_NOTICE`**

Not an error, but may need attention

SYSL_LOG_INFO

`public static final int` **`SYSL_LOG_INFO`**

Informational messages

SYSL_LOG_DEBUG

`public static final int` **`SYSL_LOG_DEBUG`**

Level for when debugging a system

SYSL_FAC_KERNEL

`public static final int` **`SYSL_FAC_KERNEL`**

Kernel messages

SYSL_FAC_USR

`public static final int` **`SYSL_FAC_USR`**

User-level messages

SYSL_FAC_MAIL_SYS

`public static final int` **`SYSL_FAC_MAIL_SYS`**

Mail system

SYSL_FAC_SYS_DAEMONS

`public static final int` **`SYSL_FAC_SYS_DAEMONS`**

System daemons

SYSL_FAC_SEC_AUTH_0

`public static final int` **`SYSL_FAC_SEC_AUTH_0`**

Security/authorization messages. RFC 3164 includes this note: Various operating systems have been found to utilize Facilities 4, 10, 13 and 14 for security/authorization,audit, and alert messages which seem to be similar.

SYSL_FAC_SYSLOGD

`public static final int` **`SYSL_FAC_SYSLOGD`**

Messages generated internally by syslogd

SYSL_FAC_LINE_PTR_SYS

`public static final int` **`SYSL_FAC_LINE_PTR_SYS`**

Line printer subsystem

(continued from last page)

SYSL_FAC_NNTP_SYS

`public static final int` **`SYSL_FAC_NNTP_SYS`**

Network news subsystem

SYSL_FAC_UUCP_SYS

`public static final int` **`SYSL_FAC_UUCP_SYS`**

UUCP subsystem

SYSL_FAC_CLOCK_DAEMON_0

`public static final int` **`SYSL_FAC_CLOCK_DAEMON_0`**

Clock daemon. RFC 3164 includes this note: Various operating systems have been found to utilize both Facilities 9 and 15 for clock (cron/at) messages

SYSL_FAC_SEC_AUTH_1

`public static final int` **`SYSL_FAC_SEC_AUTH_1`**

Security/authorization messages. RFC 3164 includes this note: Various operating systems have been found to utilize Facilities 4, 10, 13 and 14 for security/authorization,audit, and alert messages which seem to be similar.

SYSL_FAC_FTPD

`public static final int` **`SYSL_FAC_FTPD`**

FTP Daemon

SYSL_FAC_NTP_SYS

`public static final int` **`SYSL_FAC_NTP_SYS`**

NTP subsystem

SYSL_FAC_LOG_AUDIT

`public static final int` **`SYSL_FAC_LOG_AUDIT`**

Log audit. RFC 3164 includes this note: Various operating systems have been found to utilize Facilities 4, 10, 13 and 14 for security/authorization,audit, and alert messages which seem to be similar.

SYSL_FAC_LOG_ALERT

`public static final int` **`SYSL_FAC_LOG_ALERT`**

Security/authorization messages. RFC 3164 includes this note: Various operating systems have been found to utilize Facilities 4, 10, 13 and 14 for security/authorization,audit, and alert messages which seem to be similar.

SYSL_FAC_CLOCK_DAEMON_1

`public static final int` **`SYSL_FAC_CLOCK_DAEMON_1`**

Clock daemon. RFC 3164 includes this note: Various operating systems have been found to utilize both Facilities 9 and 15 for clock (cron/at) messages

SYSL_FAC_LCL_USE_0

`public static final int` **`SYSL_FAC_LCL_USE_0`**

Local use 0

SYSL_FAC_LCL_USE_1

`public static final int` **`SYSL_FAC_LCL_USE_1`**

Local use 1

SYSL_FAC_LCL_USE_2

`public static final int` **`SYSL_FAC_LCL_USE_2`**

Local use 2

SYSL_FAC_LCL_USE_3

`public static final int` **`SYSL_FAC_LCL_USE_3`**

Local use 3

SYSL_FAC_LCL_USE_4

`public static final int` **`SYSL_FAC_LCL_USE_4`**

Local use 4

SYSL_FAC_LCL_USE_5

`public static final int` **`SYSL_FAC_LCL_USE_5`**

Local use 5

SYSL_FAC_LCL_USE_6

`public static final int` **`SYSL_FAC_LCL_USE_6`**

Local use 6

SYSL_FAC_LCL_USE_7

`public static final int` **`SYSL_FAC_LCL_USE_7`**

Local use 7

com.ibm.retail.si.mgmt.logging

Interface SyslogLoggingMBean

public interface **SyslogLoggingMBean**

Extension of MgmtLoggingCtrlMBean for controlling the forwarding of Syslog data as notifications

Field Summary

static int	`SYSL_FAC_CLOCK_DAEMON_0` Clock daemon.
static int	`SYSL_FAC_CLOCK_DAEMON_1` Clock daemon.
static int	`SYSL_FAC_FTPD` FTP Daemon
static int	`SYSL_FAC_KERNEL` Kernel messages
static int	`SYSL_FAC_LCL_USE_0` Local use 0
static int	`SYSL_FAC_LCL_USE_1` Local use 1
static int	`SYSL_FAC_LCL_USE_2` Local use 2
static int	`SYSL_FAC_LCL_USE_3` Local use 3
static int	`SYSL_FAC_LCL_USE_4` Local use 4
static int	`SYSL_FAC_LCL_USE_5` Local use 5
static int	`SYSL_FAC_LCL_USE_6` Local use 6
static int	`SYSL_FAC_LCL_USE_7` Local use 7
static int	`SYSL_FAC_LINE_PTR_SYS` Line printer subsystem
static int	`SYSL_FAC_LOG_ALERT` Security/authorization messages.

static int	`SYSL_FAC_LOG_AUDIT` Log audit.
static int	`SYSL_FAC_MAIL_SYS` Mail system
static int	`SYSL_FAC_NNTP_SYS` Network news subsystem
static int	`SYSL_FAC_NTP_SYS` NTP subsystem
static int	`SYSL_FAC_SEC_AUTH_0` Security/authorization messages.
static int	`SYSL_FAC_SEC_AUTH_1` Security/authorization messages.
static int	`SYSL_FAC_SYS_DAEMONS` System daemons
static int	`SYSL_FAC_SYSLOGD` Messages generated internally by syslogd
static int	`SYSL_FAC_USR` User-level messages
static int	`SYSL_FAC_UUCP_SYS` UUCP subsystem
static int	`SYSL_LOG_ALERT` Condition needing immediate attention
static int	`SYSL_LOG_CRIT` Critical conditions
static int	`SYSL_LOG_DEBUG` Level for when debugging a system
static int	`SYSL_LOG_EMERG` Kernel panic
static int	`SYSL_LOG_ERR` Errors
static int	`SYSL_LOG_INFO` Informational messages
static int	`SYSL_LOG_NOTICE` Not an error, but may need attention
static int	`SYSL_LOG_WARNING` Warning messages

Fields

SYSL_LOG_EMERG

`public static final int` **`SYSL_LOG_EMERG`**

Kernel panic

SYSL_LOG_ALERT

`public static final int` **`SYSL_LOG_ALERT`**

Condition needing immediate attention

SYSL_LOG_CRIT

`public static final int` **`SYSL_LOG_CRIT`**

Critical conditions

SYSL_LOG_ERR

`public static final int` **`SYSL_LOG_ERR`**

Errors

SYSL_LOG_WARNING

`public static final int` **`SYSL_LOG_WARNING`**

Warning messages

SYSL_LOG_NOTICE

`public static final int` **`SYSL_LOG_NOTICE`**

Not an error, but may need attention

SYSL_LOG_INFO

`public static final int` **`SYSL_LOG_INFO`**

Informational messages

SYSL_LOG_DEBUG

`public static final int` **`SYSL_LOG_DEBUG`**

Level for when debugging a system

SYSL_FAC_KERNEL

`public static final int` **`SYSL_FAC_KERNEL`**

Kernel messages

SYSL_FAC_USR

`public static final int` **`SYSL_FAC_USR`**

User-level messages

SYSL_FAC_MAIL_SYS

`public static final int` **`SYSL_FAC_MAIL_SYS`**

Mail system

SYSL_FAC_SYS_DAEMONS

`public static final int` **`SYSL_FAC_SYS_DAEMONS`**

System daemons

SYSL_FAC_SEC_AUTH_0

`public static final int` **`SYSL_FAC_SEC_AUTH_0`**

Security/authorization messages. RFC 3164 includes this note: Various operating systems have been found to utilize Facilities 4, 10, 13 and 14 for security/authorization,audit, and alert messages which seem to be similar.

SYSL_FAC_SYSLOGD

`public static final int` **`SYSL_FAC_SYSLOGD`**

Messages generated internally by syslogd

SYSL_FAC_LINE_PTR_SYS

`public static final int` **`SYSL_FAC_LINE_PTR_SYS`**

Line printer subsystem

SYSL_FAC_NNTP_SYS

`public static final int` **`SYSL_FAC_NNTP_SYS`**

Network news subsystem

SYSL_FAC_UUCP_SYS

`public static final int` **`SYSL_FAC_UUCP_SYS`**

UUCP subsystem

SYSL_FAC_CLOCK_DAEMON_0

`public static final int` **`SYSL_FAC_CLOCK_DAEMON_0`**

Clock daemon. RFC 3164 includes this note: Various operating systems have been found to utilize both Facilities 9 and 15 for clock (cron/at) messages

SYSL_FAC_SEC_AUTH_1

`public static final int` **`SYSL_FAC_SEC_AUTH_1`**

Security/authorization messages. RFC 3164 includes this note: Various operating systems have been found to utilize Facilities 4, 10, 13 and 14 for security/authorization,audit, and alert messages which seem to be similar.

SYSL_FAC_FTPD

`public static final int` **`SYSL_FAC_FTPD`**

FTP Daemon

SYSL_FAC_NTP_SYS

`public static final int` **`SYSL_FAC_NTP_SYS`**

NTP subsystem

SYSL_FAC_LOG_AUDIT

`public static final int` **`SYSL_FAC_LOG_AUDIT`**

Log audit. RFC 3164 includes this note: Various operating systems have been found to utilize Facilities 4, 10, 13 and 14 for security/authorization,audit, and alert messages which seem to be similar.

SYSL_FAC_LOG_ALERT

`public static final int` **`SYSL_FAC_LOG_ALERT`**

Security/authorization messages. RFC 3164 includes this note: Various operating systems have been found to utilize Facilities 4, 10, 13 and 14 for security/authorization,audit, and alert messages which seem to be similar.

SYSL_FAC_CLOCK_DAEMON_1

`public static final int` **`SYSL_FAC_CLOCK_DAEMON_1`**

Clock daemon. RFC 3164 includes this note: Various operating systems have been found to utilize both Facilities 9 and 15 for clock (cron/at) messages

SYSL_FAC_LCL_USE_0

`public static final int` **`SYSL_FAC_LCL_USE_0`**

Local use 0

SYSL_FAC_LCL_USE_1

`public static final int` **`SYSL_FAC_LCL_USE_1`**

Local use 1

SYSL_FAC_LCL_USE_2

`public static final int` **`SYSL_FAC_LCL_USE_2`**

Local use 2

SYSL_FAC_LCL_USE_3

`public static final int` **`SYSL_FAC_LCL_USE_3`**

Local use 3

SYSL_FAC_LCL_USE_4

`public static final int` **`SYSL_FAC_LCL_USE_4`**

Local use 4

SYSL_FAC_LCL_USE_5

`public static final int` **`SYSL_FAC_LCL_USE_5`**

Local use 5

SYSL_FAC_LCL_USE_6

`public static final int` **`SYSL_FAC_LCL_USE_6`**

Local use 6

SYSL_FAC_LCL_USE_7

`public static final int` **`SYSL_FAC_LCL_USE_7`**

Local use 7

Package
com.ibm.retail.si.mgmt.masteragent

Provides the classes required for a Master Agent, including its core MBeans:
- `ProxyManagerMBean`: Manages remote General Agent MBean Proxies.
- `RemoteServerPoolMBean`: Manages the `MBeanServerConnection`s to remote general agents.

com.ibm.retail.si.mgmt.masteragent

Interface MasterAgentClientMBean

public interface **MasterAgentClientMBean**

com.ibm.retail.si.mgmt.masteragent

Interface MasterAgentMBean

public interface **MasterAgentMBean**

com.ibm.retail.si.mgmt.masteragent

Interface ProxyFactoryMBean

public interface **ProxyFactoryMBean**

MBean interface definition for ProxyFactory objects to be managed

Field Summary

static java.lang.String	OBJECT_NAME

Method Summary

RemoteMBeanProxy	createProxy(MgmtDeviceInfo devInfo,javax.management.ObjectName devInfo,javax.management.MBeanInfo devInfo) Creates a Remote MBean proxy of the supplied ObjectInstance
void	initialize() Initializes factory by getting a reference to the active MasterAgent

Fields

OBJECT_NAME

```
public static final java.lang.String OBJECT_NAME
```

Methods

createProxy

```
public RemoteMBeanProxy createProxy(MgmtDeviceInfo devInfo,
                                    javax.management.ObjectName objName,
                                    javax.management.MBeanInfo mBeanInfo)
```

Creates a Remote MBean proxy of the supplied ObjectInstance

Parameters:

`devInfo` -
Device Information object about the source device
`objName` -
ObjectName of the source MBean
`mBeanInfo` -
MBeanInfo object about the source MBean

Returns:

Proxied MBean instance of the original ObjectInstance

(continued from last page)

initialize

`public void **initialize**()`

Initializes factory by getting a reference to the active MasterAgent

com.ibm.retail.si.mgmt.masteragent

Interface ProxyManagerMBean

public interface **ProxyManagerMBean**

The purpose of this MBean is to create, destroy, and manage the Master Agent MBean proxies. Proxies for a remote agent upon the receipt of a `AgentDiscoveredNotification`and are torn down upon receipt of a `AgentLostNotification`.

When a remote agent is proxied, this MBean registers as a listener to the MBeanServerDelegate on the remote `MBeanServer`. Individual MBean proxies are created and destroyed upon receipt of a `MBeanServerNotification`

The `ObjectName`of this MBean includes the following attributes, in addition to the SIF attribute of `DeviceID`:
- SIFComponent=MGMT
- Id=ProxyManager

This management interface defines no attributes.

The following operations are included in this management interface. These are described in more detail in the corresponding method documentation.
- initialize
- obtainProxyObjectName
- obtainRemoteObjectName
- shutdown

This MBean emits a `Notification`with type `ProxyManager.MA_PRXY_UPDATED_TYPE`when a new proxy is added or removed in response to an `MBeanServerNotification`. `Notifications` of type `ProxyManager.MA_PRXY_SVR_ADDED_NOT_TYPE` and `ProxyManager.MA_PRXY_SRVR_REMOVED_NOT_TYPE`are emitted after processing a newly discovered or lost agent.

Field Summary

static java.lang.String	`OBJECT_NAME`

Method Summary

void	`initialize()` Initializes the ProxyManager
javax.management.ObjectName	`obtainProxyObjectName(javax.management.ObjectName remoteName)` Returns the ObjectName of the local proxy MBean, given the remote proxy's ObjectName
javax.management.ObjectName	`obtainRemoteObjectName(javax.management.ObjectName proxyName)` Returns the ObjectName of the remote proxied MBean, given the local proxy's ObjectName
void	`shutdown()`

Fields

(continued from last page)

OBJECT_NAME

`public static final java.lang.String` **`OBJECT_NAME`**

Methods

initialize

```
public void initialize()
                throws MgmtException
```

Initializes the ProxyManager

Exceptions:

`MgmtException`

See Also:

com.ibm.retail.si.mgmt.masteragent.ProxyManager#initialize()

shutdown

```
public void shutdown()
```

See Also:

com.ibm.retail.si.mgmt.masteragent.ProxyManager#shutdown()

obtainRemoteObjectName

```
public javax.management.ObjectName obtainRemoteObjectName(javax.management.ObjectName
proxyName)
```

Returns the ObjectName of the remote proxied MBean, given the local proxy's ObjectName

Parameters:

`proxyName -`
Local MBean Proxy ObjectName

Returns:

ObjectName of the remote proxied MBean

obtainProxyObjectName

```
public javax.management.ObjectName obtainProxyObjectName(javax.management.ObjectName
remoteName)
```

Returns the ObjectName of the local proxy MBean, given the remote proxy's ObjectName

Parameters:

`remoteName -`
ObjectName of the remote MBean, as it exists on the remote MBeanServer

Returns:

ObjectName of the local MBean proxy

com.ibm.retail.si.mgmt.masteragent

Interface RemoteConnectorMBean

public interface **RemoteConnectorMBean**

The purpose of this MBean is to make `MBeanServerConnection`s to newly discovered agents, and to trigger the creation of `MgmtRemoteMBeanProxy`s.

The `ObjectName`of this MBean includes the following attributes, in addition to the SIF attribute of `DeviceID`:
- SIFComponent=MGMT
- Id=RemoteConnector

This management interface defines no attributes

The following operations are included in this management interface. These are described in more detail in the corresponding method documentation.
- initialize
- shutdown

An `AgentDiscoveredNotification`is emitted by classes implementing this interface when a connection has been established to a newly discovered agent and its corresponding `MBeanServerConnection`has been stored in the `RemoteServerPool`.

An `AgentConnectionFailedNotification`is emmited by classes implementing this interface upon a failed connection attempt to a newly discovered agent.

Method Summary

`void`	`initialize()` Initializes the MBean
`void`	`shutdown()` Performs shutdown of the MBean

Methods

initialize

```
public void initialize()
                 throws MgmtException
```

Initializes the MBean

shutdown

```
public void shutdown()
```

Performs shutdown of the MBean

com.ibm.retail.si.mgmt.masteragent

Interface RemoteServerPoolMBean

public interface **RemoteServerPoolMBean**

MBean interface definition for the RemoteServerPool, contains methods to add, get and remove `MBeanServerConnection`s from the pool.

The `ObjectName`of this MBean includes the following attributes, in addition to the SIF attribute of `DeviceID`:
- SIFComponent=MGMT
- Id=RemoteServerPool

This management interface includes the following attributes. These attributes are described in more detail in the accessor methods.

- AllDevInfo
- Servers
- Size

The following operations are included in this management interface. These are described in more detail in the corresponding method documentation.
- getDevInfo
- devDevInfoByDevice
- getDevInfoByType
- getServer
- getServers
- removeRemoteServer
- shutdown

This MBean emits no `Notification`s.

Field Summary

static java.lang.String	`OBJECT_NAME`

Method Summary

void	`addRemoteServer(MgmtDeviceInfo devInfo,javax.management.remote.JMXConnector devInfo,javax.management.MBeanServerConnection devInfo)` Adds a MBeanServerConnection reference to the pool.
MgmtDeviceInfo[]	`getAllDevInfo()`
MgmtDeviceInfo	`getDevInfo(java.lang.String jndiName)` Returns the MgmtDeviceInfo object for a remote agent
MgmtDeviceInfo[]	`getDevInfoByDevice(java.lang.String deviceId)` Returns an array of currently known MgmtDeviceInfo running the supplied device
MgmtDeviceInfo[]	`getDevInfoByType(java.lang.Integer deviceType)` Returns a List of currently known MgmtDeviceInfo that are of the supplied device type

javax.management.MBeanServerConnection	`getServer(java.lang.String jndiName)` Returns a reference to a MBeanServerConnection that has been stored in the pool, which is associated by the JNDI name of the remote agent
javax.management.MBeanServerConnection[]	`getServers()` Returns references to all MBeanServerConnections that have been stored in the pool
javax.management.MBeanServerConnection[]	`getServers(java.lang.Integer deviceType)` Returns references to all MBeanServerConnections that have been stored in the pool that are associated to devices with the supplied deviceType
java.lang.Integer	`getSize()`
void	`removeRemoteServer(MgmtDeviceInfo devInfo)` Removes a MBeanServerConnection reference from the pool, usually in response to an agent being lost or disconnected from the monitoring agent.
void	`shutdown()` Shuts down the server pool, closing all remote JMXConnectors

Fields

OBJECT_NAME

```
public static final java.lang.String OBJECT_NAME
```

Methods

addRemoteServer

```
public void addRemoteServer(MgmtDeviceInfo devInfo,
                            javax.management.remote.JMXConnector connector,
                            javax.management.MBeanServerConnection server)
                     throws java.io.IOException
```

Adds a MBeanServerConnection reference to the pool. It uses the MgmtDeviceInfo object from the agent which contains the MBeanServerConnection reference to use as a key for storing the servers in the pool.

Parameters:

`devInfo` -
MgmtDeviceInfo object corresponding to the remote agent.
`connector` -
JMXConnector used to obtain this MBeanServerConnection
`server` -
MBeanServerConnection reference

Exceptions:

`IOException` -
if there is a problem communicating with the remote reference.

removeRemoteServer

```
public void removeRemoteServer(MgmtDeviceInfo devInfo)
                        throws java.io.IOException
```

(continued from last page)

Removes a MBeanServerConnection reference from the pool, usually in response to an agent being lost or disconnected from the monitoring agent.

Parameters:

`devInfo -`
MgmtDeviceInfo corresponding to the remote agent.

Exceptions:

`IOException -`
if there is a problem communicating with the remote reference.

getServer

`public javax.management.MBeanServerConnection` **`getServer`**`(java.lang.String jndiName)`

Returns a reference to a MBeanServerConnection that has been stored in the pool, which is associated by the JNDI name of the remote agent

Parameters:

`jndiName -`
JNDI name associated to a remote agent

Returns:

MBeanServerConnection if one exists for the supplied name, or null

getServers

`public javax.management.MBeanServerConnection[]` **`getServers`**`()`

Returns references to all MBeanServerConnections that have been stored in the pool

Returns:

MBeanServerConnection[]

getServers

`public javax.management.MBeanServerConnection[]` **`getServers`**`(java.lang.Integer deviceType)`

Returns references to all MBeanServerConnections that have been stored in the pool that are associated to devices with the supplied deviceType

Parameters:

`deviceType -`
Device type associated with the connections

Returns:

MBeanServerConnection[]

getAllDevInfo

`public MgmtDeviceInfo[]` **`getAllDevInfo`**`()`

Returns:

Array containing each `MgmtDeviceInfo`corresponding to each stored `MBeanServerConnection`.

(continued from last page)

getDevInfo

`public MgmtDeviceInfo **getDevInfo**(java.lang.String jndiName)`

Returns the MgmtDeviceInfo object for a remote agent

Parameters:

`jndiName -`
JNDI Name of the remote agent connection

Returns:

MgmtDeviceInfo object for a remote agent, or null if none exists

getDevInfoByDevice

`public MgmtDeviceInfo[] **getDevInfoByDevice**(java.lang.String deviceId)`

Returns an array of currently known MgmtDeviceInfo running the supplied device

Parameters:

`deviceId -`
Device to search for

Returns:

MgmtDeviceInfo[] of information about each running agent on the supplied device, or an empty array if no agents are known

getDevInfoByType

`public MgmtDeviceInfo[] **getDevInfoByType**(java.lang.Integer deviceType)`

Returns a List of currently known MgmtDeviceInfo that are of the supplied device type

Parameters:

`deviceType -`
Integer device type to search for

Returns:

Array of currently known MgmtDeviceInfo that are of the supplied device type, or an empty array if no devices of that type are known

getSize

`public java.lang.Integer **getSize**()`

Returns:

Integer size of the server pool

shutdown

`public void **shutdown**()`

Shuts down the server pool, closing all remote JMXConnectors

Package
com.ibm.retail.si.mgmt.monitor

Classes for MBean Monitor policy control. The `MonitorManagerMBean`is the primary interface for controlling `MonitorPolicy`, and for applying policies on newly discovered agents.

com.ibm.retail.si.mgmt.monitor

Class DeviceMonitorPolicy

```
java.lang.Object
  |
  +-com.ibm.retail.si.mgmt.monitor.AbstractMonitorPolicy
      |
      +-com.ibm.retail.si.mgmt.monitor.DeviceMonitorPolicy
```

public class **DeviceMonitorPolicy**
extends AbstractMonitorPolicy

Fields inherited from : class `com.ibm.retail.si.mgmt.monitor.AbstractMonitorPolicy`
`attributeList`, `monitorClassName`, `policyOwner`

Constructor Summary

`DeviceMonitorPolicy(java.lang.String monitorClassName,javax.management.NotificationListener monitorClassName,javax.management.AttributeList monitorClassName,java.lang.String monitorClassName)`

Method Summary

`java.lang.String`	`getDeviceId()`

Methods inherited from : class `com.ibm.retail.si.mgmt.monitor.AbstractMonitorPolicy`
`equals, getAttribute, getAttributeList, getMonitorClassName, getPolicyOwner`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

DeviceMonitorPolicy

```
public DeviceMonitorPolicy(java.lang.String monitorClassName,
                           javax.management.NotificationListener listener,
                           javax.management.AttributeList attributes,
                           java.lang.String deviceId)
```

Parameters:
`deviceId`

(continued from last page)

Methods

getDeviceId

```
public java.lang.String getDeviceId()
```

Returns:

deviceId

com.ibm.retail.si.mgmt.monitor

Class DeviceTypeMonitorPolicy

```
java.lang.Object
  |
  +-com.ibm.retail.si.mgmt.monitor.AbstractMonitorPolicy
      |
      +-com.ibm.retail.si.mgmt.monitor.DeviceTypeMonitorPolicy
```

public class **DeviceTypeMonitorPolicy**
extends AbstractMonitorPolicy

Fields inherited from : class `com.ibm.retail.si.mgmt.monitor.AbstractMonitorPolicy`
`attributeList`, `monitorClassName`, `policyOwner`

Constructor Summary

`DeviceTypeMonitorPolicy(java.lang.String monitorClassName,javax.management.NotificationListener monitorClassName,javax.management.AttributeList monitorClassName,int monitorClassName)`

Method Summary

`int`	`getDeviceType()`

Methods inherited from : class `com.ibm.retail.si.mgmt.monitor.AbstractMonitorPolicy`
`equals, getAttribute, getAttributeList, getMonitorClassName, getPolicyOwner`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

DeviceTypeMonitorPolicy

```
public DeviceTypeMonitorPolicy(java.lang.String monitorClassName,
                               javax.management.NotificationListener listener,
                               javax.management.AttributeList attributes,
                               int deviceType)
```

Parameters:
`deviceType`

Methods

(continued from last page)

getDeviceType

```
public int getDeviceType()
```

Returns:

deviceType

com.ibm.retail.si.mgmt.monitor

Interface MonitorManagerMBean

public interface **MonitorManagerMBean**

This MBean is responsible for applying JMX Monitors to agents based on `MonitorPolicy`s added to this MBean. Policies are added/removed using the `addMonitorPolicy`, `removeMonitorPolicy`methods. Once MonitorPolicy objects have been added, they can then be mapped or unmapped to a device type or a specific device using oneof the `registerMonitor()`or `deregisterMonitor()`methods. When a device is discovered, any applicable policies and their corresponding Monitors will be applied.

The `ObjectName`of this MBean includes the following attributes, in addition to the SIF attribute of `DeviceID`:
- SIFComponent=MGMT
- Id=MonitorManager

This management interface includes the following attributes. These attributes are described in more detail in the accessor methods.

- AllMonitorPolicies
- AllMonitorPolicyActions

The following operations are included in this management interface. These are described in more detail in the corresponding method documentation.
- addMonitorPolicy
- containsPolicy
- deregisterAllMonitors
- deregisterMonitor
- getMonitorPolicyAction
- getMonitorPolicy
- initialize
- isMonitorRegistered
- registerMonitor
- removeAllPolicies
- removeMonitorPolicy
- shutdown

This MBean emits no `Notifications`, although each instantiated `Monitor`will emit `MonitorNotifications`.

Field Summary

static java.lang.String	`OBJECT_NAME`
static java.lang.String	`SYS_PROP_MONITOR_POLICY_STORE_CLASSNAME` Property specifying the class name to use for storing monitor policies

Method Summary

boolean	`addMonitorPolicy(MonitorPolicy policy)` Adds a `MonitorPolicy`from the registry.
boolean	`containsPolicy(MonitorPolicy policy)` Returns `true`if the supplied `MonitorPolicy`has been added, otherwise `false`.

boolean	`containsPolicy(java.lang.String policyId)` Returns `true`if a policy matching the supplied identifier has been added, otherwise `false`.
void	`deregisterAllMonitors()` Removes all monitor registrations for all policies
boolean	`deregisterMonitor(MonitorPolicyAction action)` Removes the registration for the supplied `MonitorPolicyAction`, and, if the action was enabled, unregisters any created `MonitorMBean`s.
MonitorPolicy[]	`getAllMonitorPolicies()` Return an array of all stored monitor policies.
MonitorPolicyAction[]	`getAllMonitorPolicyActions()` Return an array of all stored monitor policy actions.
MonitorPolicy	`getMonitorPolicy(java.lang.String policyId)` Retrieves the `MonitorPolicy`matching the supplied policy identifier, or `null`if it cannot be found.
MonitorPolicyAction[]	`getMonitorPolicyActions(java.lang.String policyId)` Retrieves all `MonitorPolicyAction`associated with a matching the supplied policy identifier, or `null`if it cannot be found.
void	`initialize()` Initializes the MonitorManager.
boolean	`isMonitorRegistered(MonitorPolicyAction action)` Returns whether or not the supplied `MonitorPolicyAction`is registered
boolean	`registerMonitor(MonitorPolicyAction action)` Registers the supplied `MonitorPolicyAction`and, if the action is enabled, applies it to all active and newly discovered devices that match the `MonitorPolicyAction`'s target identifier (device type, device id, or system id).
void	`removeAllPolicies()` Removes all `MonitorPolicy`objects and any registrations
MonitorPolicy	`removeMonitorPolicy(MonitorPolicy policy)` Removes the `MonitorPolicy`object equal to that supplied, plus any registrations of those policies.
MonitorPolicy	`removeMonitorPolicy(java.lang.String policyId)` Removes the `MonitorPolicy`matching the supplied policy identifier, and any registrations of that policy.
void	`shutdown()` Shutdown the MBean, attempting to save currently stored policies in persistent storage

Fields

(continued from last page)

OBJECT_NAME

`public static final java.lang.String` **`OBJECT_NAME`**

SYS_PROP_MONITOR_POLICY_STORE_CLASSNAME

`public static final java.lang.String` **`SYS_PROP_MONITOR_POLICY_STORE_CLASSNAME`**

Property specifying the class name to use for storing monitor policies

See Also:

com.ibm.retail.si.mgmt.monitor.MonitorPolicyRegistry

Methods

initialize

```
public void initialize()
                throws MgmtException
```

Initializes the MonitorManager.

Exceptions:

`MgmtException` -
Error initializing the MBean

shutdown

`public void` **`shutdown`**`()`

Shutdown the MBean, attempting to save currently stored policies in persistent storage

addMonitorPolicy

`public boolean` **`addMonitorPolicy`**`(MonitorPolicy policy)`

Adds a `MonitorPolicy`from the registry. Returns `true`if the policy was added, or `false`if the policy has already been added.

Parameters:

`policy` -
MonitorPolicy to add

Returns:

true if the policy was successfully added, or false if the policy has already been added.

removeMonitorPolicy

`public MonitorPolicy` **`removeMonitorPolicy`**`(MonitorPolicy policy)`

Removes the `MonitorPolicy`object equal to that supplied, plus any registrations of those policies. This method returns the `MonitorPolicy`removed or null if it was not removed.

Parameters:

`policy` -
MonitorPolicy to remove

Returns:

(continued from last page)

The MonitorPolicy that was removed, or null if it was not removed

removeMonitorPolicy

`public MonitorPolicy **removeMonitorPolicy**(java.lang.String policyId)`

Removes the `MonitorPolicy`matching the supplied policy identifier, and any registrations of that policy. This method returns the `MonitorPolicy`removed, or null if it was not removed.

Parameters:

`policyId -`
Identifier of the MonitorPolicy to remove

Returns:

The MonitorPolicy that was removed, or null if it was not added

removeAllPolicies

`public void **removeAllPolicies**()`

Removes all `MonitorPolicy`objects and any registrations

containsPolicy

`public boolean **containsPolicy**(MonitorPolicy policy)`

Returns `true`if the supplied `MonitorPolicy`has been added, otherwise `false`.

Parameters:

`policy -`
MonitorPolicy to search for

Returns:

true if a policy matching the supplied policy identifier has been added, `false`otherwise.

containsPolicy

`public boolean **containsPolicy**(java.lang.String policyId)`

Returns `true`if a policy matching the supplied identifier has been added, otherwise `false`.

Parameters:

`policyId -`
MonitorPolicy identifier to search for

Returns:

true if a policy matching the supplied policy identifier has been added, `false`otherwise.

getMonitorPolicy

`public MonitorPolicy **getMonitorPolicy**(java.lang.String policyId)`

Retrieves the `MonitorPolicy`matching the supplied policy identifier, or `null`if it cannot be found.

Parameters:

`policyId -`
Policy identifier of the `MonitorPolicy`to retrieve

Returns:

(continued from last page)

`MonitorPolicy`
matching the supplied policy identifier, or null if it cannot be found.

getAllMonitorPolicies

```
public MonitorPolicy[] getAllMonitorPolicies()
```

Return an array of all stored monitor policies.

Returns:

MonitorPolicy[] An array of monitor policies.

registerMonitor

```
public boolean registerMonitor(MonitorPolicyAction action)
                        throws MgmtException
```

Registers the supplied `MonitorPolicyAction`and, if the action is enabled, applies it to all active and newly discovered devices that match the `MonitorPolicyAction`'s target identifier (device type, device id, or system id).

Parameters:

`action -`
The MonitorPolicyAction to register

Returns:

`true`
if the registration was added and applied successfully, or `false`if a registration for the supplied action already exists.

Exceptions:

`MgmtException`

See Also:

com.ibm.retail.si.mgmt.monitor.MonitorPolicyAction

deregisterAllMonitors

```
public void deregisterAllMonitors()
```

Removes all monitor registrations for all policies

deregisterMonitor

```
public boolean deregisterMonitor(MonitorPolicyAction action)
                          throws MgmtException
```

Removes the registration for the supplied `MonitorPolicyAction`, and, if the action was enabled, unregisters any created `MonitorMBean`s.

Parameters:

`action -`
The MonitorPolicyAction to deregister

Returns:

boolean `true`if the action was deregistered successfully, `false`otherwise

Exceptions:

`MgmtException -`
Error unregistering the general agent's MonitorMBean

(continued from last page)

getAllMonitorPolicyActions

`public MonitorPolicyAction[] getAllMonitorPolicyActions()`

Return an array of all stored monitor policy actions.

Returns:

MonitorPolicyAction[] An array of monitor policy actions.

getMonitorPolicyActions

`public MonitorPolicyAction[] getMonitorPolicyActions(java.lang.String policyId)`

Retrieves all `MonitorPolicyAction`associated with a matching the supplied policy identifier, or `null`if it cannot be found.

Parameters:

`policyId -`
Policy identifier of the `MonitorPolicy`to retrieve

Returns:

`MonitorPolicyAction[]`
containing all actions whose policy matches the supplied policy identifier, or an empty array if none can be found.

isMonitorRegistered

`public boolean isMonitorRegistered(MonitorPolicyAction action)`

Returns whether or not the supplied `MonitorPolicyAction`is registered

Parameters:

`action -`
MonitorPolicyAction to search for

Returns:

`true`
if the supplied action is registered, `false`otherwise

com.ibm.retail.si.mgmt.monitor

Class MonitorPolicy

```
java.lang.Object
   |
   +-com.ibm.retail.si.mgmt.monitor.MonitorPolicy
```

All Implemented interfaces:
java.io.Serializable

---

public class **MonitorPolicy**
extends java.lang.Object
implements java.io.Serializable

Monitor Policy Object. Contains the information required to manage a monitor, system wide, and is used in conjunction with `MonitorManagerMBean`.

Constructor Summary

`MonitorPolicy(java.lang.String monitorClassName,javax.management.AttributeList monitorClassName)`

Method Summary

`boolean`	`equals(java.lang.Object o)` Determines equality by comparing the policies' identifiers
`javax.management.AttributeList`	`getAttributeList()`
`java.lang.String`	`getId()` Unique identifier for this policy, automatically generated in the constructor of this object.
`java.lang.String`	`getMonitorClassName()`
`java.lang.String`	`toString()`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

MonitorPolicy

```
public MonitorPolicy(java.lang.String monitorClassName,
                     javax.management.AttributeList attributeList)
```

(continued from last page)

Parameters:

`monitorClassName -`
- The name of the monitor class to be created.

`attributeList -`
- The attribute list with all the attributes.

Methods

getMonitorClassName

`public java.lang.String` **`getMonitorClassName`**`()`

Returns:

String the monitor class name.

getAttributeList

`public javax.management.AttributeList` **`getAttributeList`**`()`

Returns:

AttributeList the list with all the attributes.

getId

`public java.lang.String` **`getId`**`()`

Unique identifier for this policy, automatically generated in the constructor of this object. One of it's uses is to link this policy to the instantiated Monitor MBean, by adding it as a key in the Monitor's ObjectName.

Returns:

Unique String identifier for this policy

equals

`public boolean` **`equals`**`(java.lang.Object o)`

Determines equality by comparing the policies' identifiers

See Also:

java.lang.Object#equals(java.lang.Object)

toString

`public java.lang.String` **`toString`**`()`

Package
com.ibm.retail.si.mgmt.notifications

This package contains the SIF Notification classes, a filter for these Notification classes (`RtlNotificationFilter`and `MgmtNotificationControlMBean`, which manages the Notifications emitted from all General Agents in the store.

com.ibm.retail.si.mgmt.notifications

Class AgentConnectionFailedNotification

```
java.lang.Object
  |-java.util.EventObject
    |-javax.management.Notification
      |-com.ibm.retail.si.mgmt.notifications.RtlNotification
        |-com.ibm.retail.si.mgmt.notifications.RtlInformationNotification
          |-com.ibm.retail.si.mgmt.notifications.AgentConnectionFailedNotification
```

public class **AgentConnectionFailedNotification**
extends RtlInformationNotification

Notification emitted indicating a set of failed connection attempts to a GeneralAgent has occurred

Fields inherited from : class `com.ibm.retail.si.mgmt.notifications.RtlInformationNotification`
`NOTIFICATION_TYPE`

Fields inherited from : class `javax.management.Notification`
`source`

Fields inherited from : class `java.util.EventObject`
`source`

Constructor Summary

`AgentConnectionFailedNotification(java.lang.Object source,MgmtDeviceInfo source)`

Method Summary

`MgmtDeviceInfo`	`getMgmtDeviceInfo()`

Methods inherited from : class `com.ibm.retail.si.mgmt.notifications.RtlNotification`
`getOriginatingDevice, GetSystemSequenceNo, GetSystemTimeStamp, setOriginatingDevice, SetSystemSequenceNo, SetSystemTimeStamp`

Methods inherited from : class `javax.management.Notification`
`getMessage, getSequenceNumber, getSource, getTimeStamp, getType, getUserData, setSequenceNumber, setSource, setTimeStamp, setUserData, toString`

Methods inherited from : class `java.util.EventObject`
`getSource, toString`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

AgentConnectionFailedNotification

```
public AgentConnectionFailedNotification(java.lang.Object source,
                                         MgmtDeviceInfo devInfo)
```

Methods

getMgmtDeviceInfo

```
public MgmtDeviceInfo getMgmtDeviceInfo()
```

Returns:

MgmtDeviceInfo the encapsulated object with the remote agent's information

com.ibm.retail.si.mgmt.notifications

Class AgentDiscoveredNotification

```
java.lang.Object
  +-java.util.EventObject
    +-javax.management.Notification
      +-com.ibm.retail.si.mgmt.notifications.RtlNotification
        +-com.ibm.retail.si.mgmt.notifications.RtlInformationNotification
          +-com.ibm.retail.si.mgmt.notifications.AgentDiscoveredNotification
```

public class **AgentDiscoveredNotification**
extends RtlInformationNotification

`RtlInformationNotification`
for discovery of a new `GeneralAgent`, it encapsulates a `MgmtDeviceInfo`object which has the necessary information to handle and work with the remote agents.

Fields inherited from : class `com.ibm.retail.si.mgmt.notifications.RtlInformationNotification`
`NOTIFICATION_TYPE`

Fields inherited from : class `javax.management.Notification`
`source`

Fields inherited from : class `java.util.EventObject`
`source`

Constructor Summary

`AgentDiscoveredNotification(java.lang.Object source,MgmtDeviceInfo source)`

Method Summary

`MgmtDeviceInfo`	`getMgmtDeviceInfo()`

Methods inherited from : class `com.ibm.retail.si.mgmt.notifications.RtlNotification`
`getOriginatingDevice, GetSystemSequenceNo, GetSystemTimeStamp, setOriginatingDevice, SetSystemSequenceNo, SetSystemTimeStamp`

Methods inherited from : class `javax.management.Notification`
`getMessage, getSequenceNumber, getSource, getTimeStamp, getType, getUserData, setSequenceNumber, setSource, setTimeStamp, setUserData, toString`

Methods inherited from : class `java.util.EventObject`

`getSource, toString`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

AgentDiscoveredNotification

```
public AgentDiscoveredNotification(java.lang.Object source,
                                   MgmtDeviceInfo devInfo)
```

Methods

getMgmtDeviceInfo

```
public MgmtDeviceInfo getMgmtDeviceInfo()
```

Returns:

MgmtDeviceInfo the encapsulated object with the remote agent's information

com.ibm.retail.si.mgmt.notifications

Class AgentLostNotification

```
java.lang.Object
  +-java.util.EventObject
    +-javax.management.Notification
      +-com.ibm.retail.si.mgmt.notifications.RtlNotification
        +-com.ibm.retail.si.mgmt.notifications.RtlInformationNotification
          +-com.ibm.retail.si.mgmt.notifications.AgentLostNotification
```

public class **AgentLostNotification**
extends RtlInformationNotification

`RtlInformationNotification`
for release of a `GeneralAgent`that is being managed, it encapsulates a `MgmtDeviceInfo`object which has the necessary information to remote the remote agent from the pool.

Fields inherited from : class `com.ibm.retail.si.mgmt.notifications.RtlInformationNotification`
`NOTIFICATION_TYPE`

Fields inherited from : class `javax.management.Notification`
`source`

Fields inherited from : class `java.util.EventObject`
`source`

Constructor Summary

`AgentLostNotification(java.lang.Object source,MgmtDeviceInfo source)`

Method Summary

`MgmtDeviceInfo`	`getMgmtDeviceInfo()`

Methods inherited from : class `com.ibm.retail.si.mgmt.notifications.RtlNotification`
`getOriginatingDevice, GetSystemSequenceNo, GetSystemTimeStamp, setOriginatingDevice, SetSystemSequenceNo, SetSystemTimeStamp`

Methods inherited from : class `javax.management.Notification`
`getMessage, getSequenceNumber, getSource, getTimeStamp, getType, getUserData, setSequenceNumber, setSource, setTimeStamp, setUserData, toString`

Methods inherited from : class `java.util.EventObject`

`getSource, toString`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

AgentLostNotification

```
public AgentLostNotification(java.lang.Object source,
                             MgmtDeviceInfo devInfo)
```

Methods

getMgmtDeviceInfo

```
public MgmtDeviceInfo getMgmtDeviceInfo()
```

Returns:

MgmtDeviceInfo the encapsulated object with the remote agent's information

com.ibm.retail.si.mgmt.notifications

Class AgentShutdownNotification

```
java.lang.Object
  +-java.util.EventObject
    +-javax.management.Notification
      +-com.ibm.retail.si.mgmt.notifications.RtlNotification
        +-com.ibm.retail.si.mgmt.notifications.RtlInformationNotification
          +-com.ibm.retail.si.mgmt.notifications.AgentShutdownNotification
```

public class **AgentShutdownNotification**
extends RtlInformationNotification

Fields inherited from : class `com.ibm.retail.si.mgmt.notifications.RtlInformationNotification`
`NOTIFICATION_TYPE`

Fields inherited from : class `javax.management.Notification`
`source`

Fields inherited from : class `java.util.EventObject`
`source`

Constructor Summary

`AgentShutdownNotification(java.lang.Object source,MgmtDeviceInfo source)`

Method Summary

`MgmtDeviceInfo`	`getMgmtDeviceInfo()`

Methods inherited from : class `com.ibm.retail.si.mgmt.notifications.RtlNotification`
`getOriginatingDevice, GetSystemSequenceNo, GetSystemTimeStamp, setOriginatingDevice, SetSystemSequenceNo, SetSystemTimeStamp`

Methods inherited from : class `javax.management.Notification`
`getMessage, getSequenceNumber, getSource, getTimeStamp, getType, getUserData, setSequenceNumber, setSource, setTimeStamp, setUserData, toString`

Methods inherited from : class `java.util.EventObject`
`getSource, toString`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

AgentShutdownNotification

```
public AgentShutdownNotification(java.lang.Object source,
                                 MgmtDeviceInfo devInfo)
```

Methods

getMgmtDeviceInfo

```
public MgmtDeviceInfo getMgmtDeviceInfo()
```

Returns:

MgmtDeviceInfo the encapsulated object with the remote agent's information

com.ibm.retail.si.mgmt.notifications

Interface MgmtNotificationControlMBean

public interface **MgmtNotificationControlMBean**

This interface describes the Notification Collection & Control function to be implemented on the Master Agent. Its job is to register as a Notification listener on all General Agents and the local Mastera Agent for all `Notification`s. When notifications are received, they are resent, and those that pass through a local filter are stored in a log, which is implemented as a circular queue of a finite size. When the log reaches the count specified as the maximum size, the oldest stored `Notification`s are deleted in order to make space for new ones.

The `RtlNotificationFilter`each instance carries is comprised of one of more fully qualified class names (based on defined Notifications) that is used to control the flow of incoming `Notification`s that are actually stored in the log. These classnames, as well as the size of the log itself, are configurable, and can be modified at any time during the life of the log as needed by the user. The stored contents of the log can be retrieved by a caller using a variety of additional filtering as detailed by the definition of the interface.

More than one of these logs can be created on the Master Agent, with each one being configured with a different set of filter classnames. For example, it might be desirable to create a log that listens only to type `RtlDebugNotification`, or `RtlTracePointNotification`. To make this easier, there are several pre-defined filters listed below, that can be used as starting places for this.

To filter based on criteria other than classname (like type), the class name should be entered in the filter and filtering should be done as Notifications are retrieved from the log.

The `ObjectName`of this MBean includes the following attributes, in addition to the SIF attribute of `DeviceID`:
- SIFComponent=MGMT
- Id=NotificationControl

This management interface includes the following attributes. These attributes are described in more detail in the accessor methods.

- ActiveFilter
- CurrentNotificationCount
- FirstStoredNotification
- LogName
- QMaxSize

The following operations are included in this management interface. These are described in more detail in the corresponding method documentation.
- getCurrentNotificationCount
- getFirstStoredNotification
- getNextStoredNotification
- initialize
- removeActiveFilterElements
- resetLog
- shutdown

This MBean emits all `Notification`s of various types which are emitted from each General Agent.

See Also:

com.ibm.retail.si.mgmt.notifications.RtlNotification, com.ibm.retail.si.mgmt.notifications.RtlCriticalNotification, com.ibm.retail.si.mgmt.notifications.RtlEmergencyNotification, com.ibm.retail.si.mgmt.notifications.RtlAlertNotification, com.ibm.retail.si.mgmt.notifications.RtlErrorNotification, com.ibm.retail.si.mgmt.notifications.RtlWarningNotification, com.ibm.retail.si.mgmt.notifications.RtlNoticeNotification, com.ibm.retail.si.mgmt.notifications.RtlInformationNotification, com.ibm.retail.si.mgmt.notifications.RtlDebugNotification, com.ibm.retail.si.mgmt.notifications.RtlTracePointNotification, com.ibm.retail.si.mgmt.notifications.RtlConsumerNotification

Field Summary

static java.lang.String[]	ALL_FILTER Notification filter list that will cause ALL notifications to be passed into the log.
static java.lang.String[]	ALL_RTL_FILTER Notification filter list that will cause ALL RSS notifications to be passed into the log.
static java.lang.String[]	CONSUMER_FILTER Notification filter list that will cause ALL Consumer notifications to be passed into the log.
static java.lang.String[]	CRITICAL_FILTER Notification filter list that will cause ALL Critical notifications to be passed into the log.
static java.lang.String[]	DEBUG_FILTER Notification filter list that will cause ALL Debug notifications to be passed into the log.
static java.lang.String[]	DEFAULT_FILTER Default Notification filter list.
static java.lang.String	DEFAULT_LOG_NAME
static int	DEFAULT_Q_SIZE
static java.lang.String[]	INFORMATION_FILTER Notification filter list that will cause ALL Informational notifications to be passed into the log.
static java.lang.String[]	LOG_FILTER Notification filter list that will cause ALL Logging notifications to be passed into the log.
static java.lang.String	OBJECT_NAME

Method Summary

void	addActiveFilterElement(java.lang.String[] objects) Adds Notification Class(s) to the currently active filter.
java.lang.String[]	getActiveFilter() Returns the array of Notification Class names that comprise the currently active filter.
int	getCurrentNotificationCount() Returns a count that represents the number of Notifications currenly stored in this log.
int	getCurrentNotificationCount(int deviceType) Returns a count that represents the number of Notifications currenly stored in this log that were generated by a particular device type.

int	`getCurrentNotificationCount(java.lang.String systemId)` Returns a count that represents the number of Notifications currenly stored in this log that were generated by a particular agent.
int	`getCurrentNotificationCount(java.lang.String[] filter)` Returns a count that represents the number of Notifications currenly stored in this log that are of the type specified in the passed filter.
int	`getCurrentNotificationCount(java.lang.String[] filter,int filter)` Returns a count that represents the number of Notifications currenly stored in this log that are of the type specified in the passed Notication filter, and were emitted by the passed device type.
int	`getCurrentNotificationCount(java.lang.String[] filter,java.lang.String filter)` Returns a count that represents the number of Notifications currenly stored in this log that are of the type specified in the passed filter, and were generated by a particular agent.
javax.management.Notification	`getFirstStoredNotification()` Returns the first stored Notification that matches the search criteria.
javax.management.Notification	`getFirstStoredNotification(int deviceType)` Returns the first stored Notification that matches the search criteria.
javax.management.Notification	`getFirstStoredNotification(java.lang.String systemId)` Returns the first stored Notification that matches the search criteria.
javax.management.Notification	`getFirstStoredNotification(java.lang.String[] filter)` Returns the first stored Notification that matches the search criteria.
javax.management.Notification	`getFirstStoredNotification(java.lang.String[] filter,int filter)` Returns the first stored Notification that matches the search criteria.
javax.management.Notification	`getFirstStoredNotification(java.lang.String[] filter,java.lang.String filter)` Returns the first stored Notification that matches the search criteria.
java.lang.String	`getLogName()` Returns the filename in use by this log.
javax.management.Notification	`getNextStoredNotification(long sequenceNo)` Returns the next stored Notification (based on SequenceNo) that matches the search criteria.
javax.management.Notification	`getNextStoredNotification(long sequenceNo,int sequenceNo)` Returns the next stored Notification (based on SequenceNo) that matches the search criteria.
javax.management.Notification	`getNextStoredNotification(long sequenceNo,java.lang.String sequenceNo)` Returns the next stored Notification (based on SequenceNo) that matches the search criteria.
javax.management.Notification	`getNextStoredNotification(long sequenceNo,java.lang.String[] sequenceNo)` Returns the next stored Notification (based on SequenceNo) that matches the search criteria.

javax.management.Notification	`getNextStoredNotification(long sequenceNo,java.lang.String[] sequenceNo,int sequenceNo)` Returns the next stored Notification (based on SequenceNo) that matches the search criteria.
javax.management.Notification	`getNextStoredNotification(long sequenceNo,java.lang.String[] sequenceNo,java.lang.String sequenceNo)` Returns the next stored Notification (based on SequenceNo) that matches the search criteria.
java.lang.Integer	`getQMaxSize()` Get the number of notifications that the persistent store is currently configured to store.
void	`initialize()` Initializes the MBean, setting up communication with the `MasterAgent`and registers NotificationListeners with all `MasterAgent`MBeans
void	`removeActiveFilterElements(java.lang.String[] objects)` Removes Notification Class(s) from the currently active filter.
void	`resetLog()` Clears the log of all stored entires.
void	`setActiveFilter(java.lang.String[] objects)` Sets the filter to be used when considering a notification for submition to this log file.
void	`setQMaxSize(java.lang.Integer q_Size)` Set the number of notifications that the persistent store is currently configured to store.
void	`shutdown()` Shutdown of notification control, for sync of registry and file storage.

Fields

DEFAULT_Q_SIZE

`public static final int` **`DEFAULT_Q_SIZE`**

DEFAULT_LOG_NAME

`public static final java.lang.String` **`DEFAULT_LOG_NAME`**

OBJECT_NAME

`public static final java.lang.String` **`OBJECT_NAME`**

ALL_FILTER

`public static final java.lang.String` **`ALL_FILTER`**

(continued from last page)

Notification filter list that will cause ALL notifications to be passed into the log.

ALL_RTL_FILTER

`public static final java.lang.String` **`ALL_RTL_FILTER`**

Notification filter list that will cause ALL RSS notifications to be passed into the log.

CRITICAL_FILTER

`public static final java.lang.String` **`CRITICAL_FILTER`**

Notification filter list that will cause ALL Critical notifications to be passed into the log. Those include: RtlCriticalNotification, RtlEmergencyNotification, RtlAlertNotification and all notifications derived from them.

DEBUG_FILTER

`public static final java.lang.String` **`DEBUG_FILTER`**

Notification filter list that will cause ALL Debug notifications to be passed into the log. Those include: RtlDebugNotification, RtlTracePointNotification and all notifications derived from them.

LOG_FILTER

`public static final java.lang.String` **`LOG_FILTER`**

Notification filter list that will cause ALL Logging notifications to be passed into the log. Those include: RtlTracePointNotification and all notifications derived from it.

CONSUMER_FILTER

`public static final java.lang.String` **`CONSUMER_FILTER`**

Notification filter list that will cause ALL Consumer notifications to be passed into the log. Those include: RtlConsumerNotification and all notifications derived from it.

INFORMATION_FILTER

`public static final java.lang.String` **`INFORMATION_FILTER`**

Notification filter list that will cause ALL Informational notifications to be passed into the log. Those include: RtlInformationNotification and all notifications derived from it.

DEFAULT_FILTER

`public static final java.lang.String` **`DEFAULT_FILTER`**

Default Notification filter list. The default list include: RtlCriticalNotification, RtlEmergencyNotification, RtlAlertNotification, RtlErrorNotification, RtlInformationNotification, RtlNoticeNotification, RtlWarningNotification, and all notifications derived from them.

Methods

initialize

```
public void initialize()
                 throws MgmtException
```

Initializes the MBean, setting up communication with the `MasterAgent`and registers NotificationListeners with all `MasterAgent`MBeans

Exceptions:

`MgmtException` -
Error communicating with the `MasterAgent`

shutdown

```
public void shutdown()
```

Shutdown of notification control, for sync of registry and file storage.

setQMaxSize

```
public void setQMaxSize(java.lang.Integer q_Size)
                 throws javax.management.InvalidAttributeValueException
```

Set the number of notifications that the persistent store is currently configured to store. This persistent store is a circular queue, and once the number of stored notifications exceeds this number, the oldest in the queue is removed to allow space for new notificaitons. By default, this value is set to 500. If attempt is made to set this size to a number that is smaller then the currently configured size, then the oldest notifications will be deleted and the log resized accordingly.

Parameters:

`q_Size -`
The number of notifications that the log should be configured to hold.

getQMaxSize

```
public java.lang.Integer getQMaxSize()
```

Get the number of notifications that the persistent store is currently configured to store.

Returns:

Integer, The number of notifications that this log is currently configured to hold.

getLogName

```
public java.lang.String getLogName()
```

Returns the filename in use by this log. Since in any Master Agent there can be one or more instances of this interface (based on filter types), the name is what uniquely identifies this log.

Returns:

String The filename.

setActiveFilter

```
public void setActiveFilter(java.lang.String[] objects)
                     throws javax.management.InvalidAttributeValueException
```

Sets the filter to be used when considering a notification for submition to this log file. The log file represented by this interface can be configured to only store notifications of a particular class or classes, and it is this function that sets that. NOTE: that this is not a modification to the filter, but rather a replacement, which means that as a consequence of this call, the implemenation of the log will reset the log on completion of this call.

Parameters:

`objects -`
An array of Notification Class names to be included in the log.

addActiveFilterElement

```
public void addActiveFilterElement(java.lang.String[] objects)
```

Adds Notification Class(s) to the currently active filter. This is used to append additional Class types to the list of classes that are to be passed by the filter into the log.

Parameters:

(continued from last page)

`objects -`
An array of Notification Class names to be included in the log.

removeActiveFilterElements

`public void` **`removeActiveFilterElements`**`(java.lang.String[] objects)`

Removes Notification Class(s) from the currently active filter. This is used to remove Class types from the list of classes that are to be passed by the filter into the log. The side effect of this call is that Notifications currently in the log that match these class name(s) will be removed from the log.

Parameters:

`objects -`
An array of Notification Class names to be removed from the log.

getActiveFilter

`public java.lang.String[]` **`getActiveFilter`**`()`

Returns the array of Notification Class names that comprise the currently active filter.

Returns:

String[] An array of Notification Class names.

getCurrentNotificationCount

`public int` **`getCurrentNotificationCount`**`()`

Returns a count that represents the number of Notifications currenly stored in this log.

Returns:

int Number of stored Notifications.

getCurrentNotificationCount

`public int` **`getCurrentNotificationCount`**`(java.lang.String[] filter)`

Returns a count that represents the number of Notifications currenly stored in this log that are of the type specified in the passed filter. As an example, an applicaiton might only be interested in RtlCriticalNotification, and is looking to determine how many of that type have been logged.

Parameters:

`filter -`
List of Class names to be used to filter the log for a count.

Returns:

int Number of stored Notifications that match the filter.

getCurrentNotificationCount

`public int` **`getCurrentNotificationCount`**`(int deviceType)`

Returns a count that represents the number of Notifications currenly stored in this log that were generated by a particular device type. For a list of Device Types currenly supported, see the common constants class.

Parameters:

`deviceType -`
Device type

Returns:

(continued from last page)

int Number of stored Notifications that match the device type.

getCurrentNotificationCount

```
public int getCurrentNotificationCount(java.lang.String[] filter,
                                       int deviceType)
```

Returns a count that represents the number of Notifications currenly stored in this log that are of the type specified in the passed Notication filter, and were emitted by the passed device type. As an example, an applicaiton might only be interested in RtlCriticalNotification that were generated by a POS controller, and is looking to determine how many of that type have been logged.

Parameters:

`filter` -
List of Class names to be used to filter the log for a count.
`deviceType` -
Device type

Returns:

int Number of stored Notifications that match the filter.

getCurrentNotificationCount

```
public int getCurrentNotificationCount(java.lang.String systemId)
```

Returns a count that represents the number of Notifications currenly stored in this log that were generated by a particular agent.

Parameters:

`systemId` -
The agent's system ID

Returns:

int Number of stored Notifications that match the device type.

getCurrentNotificationCount

```
public int getCurrentNotificationCount(java.lang.String[] filter,
                                       java.lang.String systemId)
```

Returns a count that represents the number of Notifications currenly stored in this log that are of the type specified in the passed filter, and were generated by a particular agent. As an example, an application might only be interested in `RtlCriticalNotification`s that were generated by a particular mobile device, and is looking to determine how many of that type have been logged.

Parameters:

`filter` -
List of Class names to be used to filter the log for a count.
`systemId` -
The agent's system ID

Returns:

int Number of stored Notifications that match the filter.

resetLog

```
public void resetLog()
```

Clears the log of all stored entires. This should be used carefully, as it will remove any history that this log was storing.

(continued from last page)

getFirstStoredNotification

`public javax.management.Notification` **`getFirstStoredNotification`**`()`

Returns the first stored Notification that matches the search criteria.

Returns:

Notification, The first Notification that matches the filter criteria, or null if there are no more that match.

---

getNextStoredNotification

`public javax.management.Notification` **`getNextStoredNotification`**`(long sequenceNo)`

Returns the next stored Notification (based on SequenceNo) that matches the search criteria.

Parameters:

`sequenceNo -`
- Sequence number from the previous notification

Returns:

Notification - The next Notification that matches the filter criteria, or null if there are no more that match.

---

getFirstStoredNotification

`public javax.management.Notification` **`getFirstStoredNotification`**`(java.lang.String[] filter)`

Returns the first stored Notification that matches the search criteria. As an example, an applicaiton might only be interested in RtlCriticalNotification, and is looking to determine how many of that type have been logged.

Parameters:

`filter -`
- List of Class names to be used to filter the log.

Returns:

Notification - The first Notification that matches the filter criteria, or null if there are no more that match.

---

getNextStoredNotification

`public javax.management.Notification` **`getNextStoredNotification`**`(long sequenceNo, java.lang.String[] filter)`

Returns the next stored Notification (based on SequenceNo) that matches the search criteria.

Parameters:

`sequenceNo -`
- The listener that is to be the target of the re-emitted Notifications.

`filter -`
- List of Class names to be used to filter the log.

Returns:

Notification - The next Notification that matches the filter criteria, or null if there are no more that match.

---

getFirstStoredNotification

`public javax.management.Notification` **`getFirstStoredNotification`**`(int deviceType)`

Returns the first stored Notification that matches the search criteria. For a list of Device Types currenly supported, see the common constants class.

(continued from last page)

Parameters:

`deviceType -`
- Device type

Returns:

Notification - The first Notification that matches the filter criteria, or null if there are no more that match.

getNextStoredNotification

```
public javax.management.Notification getNextStoredNotification(long sequenceNo,
                                                               int deviceType)
```

Returns the next stored Notification (based on SequenceNo) that matches the search criteria.

Parameters:

`sequenceNo -`
- The listener that is to be the target of the re-emitted Notifications.

`deviceType -`
- Device type

Returns:

Notification - The next Notification that matches the filter criteria, or null if there are no more that match.

getFirstStoredNotification

```
public javax.management.Notification getFirstStoredNotification(java.lang.String[]
filter,
                                                                int deviceType)
```

Returns the first stored Notification that matches the search criteria. As an example, an applicaiton might only be interested in RtlCriticalNotification that were generated by a POS controller, and is looking to determine how many of that type have been logged.

Parameters:

`filter -`
- List of Class names to be used to filter the log.

`deviceType -`
- Device type

Returns:

Notification - The first Notification that matches the filter criteria, or null if there are no more that match.

getNextStoredNotification

```
public javax.management.Notification getNextStoredNotification(long sequenceNo,
                                                               java.lang.String[]
filter,
                                                               int deviceType)
```

Returns the next stored Notification (based on SequenceNo) that matches the search criteria. As an example, an applicaiton might only be interested in RtlCriticalNotification that were generated by a POS controller, and is looking to determine how many of that type have been logged.

Parameters:

`sequenceNo -`
- The listener that is to be the target of the re-emitted Notifications.

`filter -`
- List of Class names to be used to filter the log.

`deviceType -`
- Device type

(continued from last page)

Returns:

Notification - The next Notification that matches the filter criteria, or null if there are no more that match.

getFirstStoredNotification

```
public javax.management.Notification getFirstStoredNotification(java.lang.String
systemId)
```

Returns the first stored Notification that matches the search criteria.

Parameters:

`systemId -`
- The agent system ID to match

Returns:

Notification - The first Notification that matches the filter criteria, or null if there are no more that match.

getNextStoredNotification

```
public javax.management.Notification getNextStoredNotification(long sequenceNo,
                                                               java.lang.String
systemId)
```

Returns the next stored Notification (based on SequenceNo) that matches the search criteria.

Parameters:

`sequenceNo -`
- The listener that is to be the target of the re-emitted Notifications.

`systemId -`
- The agent system ID to match

Returns:

Notification - The next Notification that matches the filter criteria, or null if there are no more that match.

getFirstStoredNotification

```
public javax.management.Notification getFirstStoredNotification(java.lang.String[]
filter,
                                                                java.lang.String
systemId)
```

Returns the first stored Notification that matches the search criteria. As an example, an application might only be interested in RtlCriticalNotification(s) that were generated by a particular mobile device, and is looking to determine how many of that type have been logged.

Parameters:

`filter -`
- List of Class names to be used to filter the log.

`systemId -`
- The agent system ID to match

Returns:

Notification - The first Notification that matches the filter criteria, or null if there are no more that match.

(continued from last page)

getNextStoredNotification

```
public javax.management.Notification getNextStoredNotification(long sequenceNo,
                                                               java.lang.String[]
filter,
                                                               java.lang.String
systemId)
```

Returns the next stored Notification (based on SequenceNo) that matches the search criteria.

Parameters:

`sequenceNo` -
- The listener that is to be the target of the re-emitted Notifications.

`filter` -
- List of Class names to be used to filter the log.

`systemId` -
- The agent system ID to match

Returns:

Notification - The next Notification that matches the filter criteria, or null if there are no more that match.

com.ibm.retail.si.mgmt.notifications

Class MgmtSDCompletionNotification

```
java.lang.Object
  |-java.util.EventObject
    |-javax.management.Notification
      |-com.ibm.retail.si.mgmt.notifications.RtlNotification
        |-com.ibm.retail.si.mgmt.notifications.RtlInformationNotification
          |-com.ibm.retail.si.mgmt.notifications.MgmtSDCompletionNotification
```

public class **MgmtSDCompletionNotification**
extends RtlInformationNotification

Software Distribution completion Notification class

See Also:
RtlInformationNotification

Field Summary

boolean	isInstall
int	RC
com.ibm.retail.si.mgmt.swdist.MgmtSftPackage	swPackage

Fields inherited from : class `com.ibm.retail.si.mgmt.notifications.RtlInformationNotification`

NOTIFICATION_TYPE

Fields inherited from : class `javax.management.Notification`

source

Fields inherited from : class `java.util.EventObject`

source

Constructor Summary

`MgmtSDCompletionNotification(java.lang.Object source,java.lang.String source,MgmtSftPackage source,boolean source,int source)`

Construct a MgmtSDCompletionNotification for submission to the agent.

Method Summary

int	`getCompletionCode()` Retrieve the completion code for this distribution.
MgmtSftPackage	`getSwPackage()` Retrieve the Package
boolean	`isInstall()`

Methods inherited from : class `com.ibm.retail.si.mgmt.notifications.RtlNotification`
`getOriginatingDevice, GetSystemSequenceNo, GetSystemTimeStamp, setOriginatingDevice, SetSystemSequenceNo, SetSystemTimeStamp`

Methods inherited from : class `javax.management.Notification`
`getMessage, getSequenceNumber, getSource, getTimeStamp, getType, getUserData, setSequenceNumber, setSource, setTimeStamp, setUserData, toString`

Methods inherited from : class `java.util.EventObject`
`getSource, toString`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

swPackage

`protected com.ibm.retail.si.mgmt.swdist.MgmtSftPackage` **`swPackage`**

RC

`protected int` **`RC`**

isInstall

`protected boolean` **`isInstall`**

Constructors

(continued from last page)

MgmtSDCompletionNotification

```
public MgmtSDCompletionNotification(java.lang.Object source,
                                    java.lang.String message,
                                    MgmtSftPackage swPackage,
                                    boolean isInstall,
                                    int RC)
```

Construct a MgmtSDCompletionNotification for submission to the agent.

Parameters:

`Object, -`
source The object that generated this notification
`String, -`
message Caller provided message
`String, -`
swPackage - The software package this Notification is associated with.
`int, -`
RC - the return or completion code associated with this Distribution. These codes are defined in: MgmtSoftwareDistClientMBean;

See Also:

com.ibm.retail.si.mgmt.swdist.MgmtSoftwareDistClientMBean

Methods

getSwPackage

```
public MgmtSftPackage getSwPackage()
```

Retrieve the Package

Returns:

MgmtSftPackage

getCompletionCode

```
public int getCompletionCode()
```

Retrieve the completion code for this distribution.

Returns:

int

isInstall

```
public boolean isInstall()
```

Returns:

True if the completed distribution was an installation, false if an uninstallation

com.ibm.retail.si.mgmt.notifications

Class MgmtSDProgressNotification

```
java.lang.Object
  |-java.util.EventObject
    |-javax.management.Notification
      |-com.ibm.retail.si.mgmt.notifications.RtlNotification
        |-com.ibm.retail.si.mgmt.notifications.RtlInformationNotification
          |-com.ibm.retail.si.mgmt.notifications.MgmtSDProgressNotification
```

public class **MgmtSDProgressNotification**
extends RtlInformationNotification

Software Distribution Progress Notification class

See Also:
RtlInformationNotification

Fields inherited from : class `com.ibm.retail.si.mgmt.notifications.RtlInformationNotification`
`NOTIFICATION_TYPE`

Fields inherited from : class `javax.management.Notification`
`source`

Fields inherited from : class `java.util.EventObject`
`source`

Constructor Summary

`MgmtSDProgressNotification(java.lang.Object source,MgmtSftPackage source,int source,java.lang.String source)` Construct a MgmtSDProgressNotification for submition to the agent.

Method Summary

`java.lang.String`	`getMessage()` Retrieve the Message associated with this notification.
`int`	`getPercentage()` Retrieve the complete percentage indicated by this notification.
`MgmtSftPackage`	`getSwPackage()` Retrieve the Package associated with this notification.

Methods inherited from : class `com.ibm.retail.si.mgmt.notifications.RtlNotification`
`getOriginatingDevice, GetSystemSequenceNo, GetSystemTimeStamp, setOriginatingDevice, SetSystemSequenceNo, SetSystemTimeStamp`

Methods inherited from : class `javax.management.Notification`
`getMessage, getSequenceNumber, getSource, getTimeStamp, getType, getUserData, setSequenceNumber, setSource, setTimeStamp, setUserData, toString`

Methods inherited from : class `java.util.EventObject`
`getSource, toString`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

MgmtSDProgressNotification

```
public MgmtSDProgressNotification(java.lang.Object source,
                                  MgmtSftPackage swPackage,
                                  int percentage,
                                  java.lang.String Message)
```

Construct a MgmtSDProgressNotification for submition to the agent.

Parameters:

`Object, -`
source The object that generated this notification
`MgmtSftPackage, -`
swPackage - The software package this Notification is associated with.
`int, -`
percentage - the percentage complete that this notification is announcing.
`String, -`
Message - a text message that 'may' be included if needed. NOTE: that given NLS requirements, this may in fact be a translation tag, based on implementation.

Methods

getSwPackage

```
public MgmtSftPackage getSwPackage()
```

Retrieve the Package associated with this notification.

Returns:

MgmtSftPackage

getPercentage

```
public int getPercentage()
```

(continued from last page)

Retrieve the complete percentage indicated by this notification.

Returns:

int

getMessage

`public java.lang.String` **`getMessage`**`()`

Retrieve the Message associated with this notification. This can be NULL.

Returns:

String

com.ibm.retail.si.mgmt.notifications

Class MgmtSDStartedNotification

```
java.lang.Object
  |-java.util.EventObject
    |-javax.management.Notification
      |-com.ibm.retail.si.mgmt.notifications.RtlNotification
        |-com.ibm.retail.si.mgmt.notifications.RtlInformationNotification
          |-com.ibm.retail.si.mgmt.notifications.MgmtSDStartedNotification
```

public class **MgmtSDStartedNotification**
extends RtlInformationNotification

Software Distribution Started Notification class. Sent when a software distribution has been successfully signaled

See Also:
RtlInformationNotification

Field Summary

java.lang.String	clientSystemId
boolean	isInstall
int	RC
com.ibm.retail.si.mgmt.swdist.MgmtSftPackage	swPackage

Fields inherited from : class com.ibm.retail.si.mgmt.notifications.RtlInformationNotification
NOTIFICATION_TYPE

Fields inherited from : class javax.management.Notification
source

Fields inherited from : class java.util.EventObject
source

Constructor Summary

MgmtSDStartedNotification(java.lang.Object source,java.lang.String source,MgmtSftPackage source,java.lang.String source,boolean source,int source)

Construct a MgmtSDCompletionNotification for submission to the agent.

Method Summary

java.lang.String	getClientSystemId()
int	getCompletionCode() Retrieve the completion code returned by the client
MgmtSftPackage	getSwPackage() Retrieve the Package
boolean	isInstall()

Methods inherited from : class com.ibm.retail.si.mgmt.notifications.RtlNotification

getOriginatingDevice, GetSystemSequenceNo, GetSystemTimeStamp, setOriginatingDevice, SetSystemSequenceNo, SetSystemTimeStamp

Methods inherited from : class javax.management.Notification

getMessage, getSequenceNumber, getSource, getTimeStamp, getType, getUserData, setSequenceNumber, setSource, setTimeStamp, setUserData, toString

Methods inherited from : class java.util.EventObject

getSource, toString

Methods inherited from : class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

swPackage

protected com.ibm.retail.si.mgmt.swdist.MgmtSftPackage **swPackage**

RC

protected int **RC**

clientSystemId

protected java.lang.String **clientSystemId**

(continued from last page)

isInstall

```
protected boolean isInstall
```

Constructors

MgmtSDStartedNotification

```
public MgmtSDStartedNotification(java.lang.Object source,
                                 java.lang.String message,
                                 MgmtSftPackage swPackage,
                                 java.lang.String clientSystemId,
                                 boolean isInstall,
                                 int RC)
```

Construct a MgmtSDCompletionNotification for submission to the agent.

Parameters:

`Object, -`
source The object that generated this notification
`String, -`
message Caller provided message
`String, -`
swPackage - The software package this Notification is associated with.
`String, -`
clientSystemId - The system ID of the client where the distribution was triggered
`int, -`
RC - The return code from the client. These codes are defined in: MgmtSoftwareDistClientMBean;

See Also:

com.ibm.retail.si.mgmt.swdist.MgmtSoftwareDistClientMBean

Methods

getSwPackage

```
public MgmtSftPackage getSwPackage()
```

Retrieve the Package

Returns:

MgmtSftPackage

---

getCompletionCode

```
public int getCompletionCode()
```

Retrieve the completion code returned by the client

Returns:

int

---

isInstall

```
public boolean isInstall()
```

(continued from last page)

Returns:

True if the distribution is an installation, false if an uninstallation

getClientSystemId

```
public java.lang.String getClientSystemId()
```

Returns:

String System ID of the client where the distribution was triggered

com.ibm.retail.si.mgmt.notifications

Class RtlAlertNotification

```
java.lang.Object
  +-java.util.EventObject
    +-javax.management.Notification
      +-com.ibm.retail.si.mgmt.notifications.RtlNotification
        +-com.ibm.retail.si.mgmt.notifications.RtlAlertNotification
```

public class **RtlAlertNotification**
extends RtlNotification

This is used to represent an Alertable condition that has occurred in the system. Alert events are defined as those things that very important but not terminal to the operation of the system. It is advisable that devices wishing to define Alert Notifications that are unique to themselves, sub-class this Class to make the resulting Notification filterable at a finer granularity, and more usable by a management application.

See Also:
javax.management.Notification, com.ibm.retail.si.mgmt.notifications.RtlNotification

Field Summary

static java.lang.String	NOTIFICATION_TYPE

Fields inherited from : class `javax.management.Notification`

source

Fields inherited from : class `java.util.EventObject`

source

Constructor Summary

`RtlAlertNotification(java.lang.Object source,java.lang.String source)`

Creates a new instance of this Notificaiton where the caller supplies both a Message string, and a source Object.

`RtlAlertNotification(java.lang.Object source,java.lang.String source,java.lang.Object source)`

Creates a new instance of this Notificaiton where the caller supplies a Message string, a source object, and a user data Object.

Methods inherited from : class `com.ibm.retail.si.mgmt.notifications.RtlNotification`

getOriginatingDevice, GetSystemSequenceNo, GetSystemTimeStamp, setOriginatingDevice, SetSystemSequenceNo, SetSystemTimeStamp

Methods inherited from : class `javax.management.Notification`

`getMessage, getSequenceNumber, getSource, getTimeStamp, getType, getUserData, setSequenceNumber, setSource, setTimeStamp, setUserData, toString`

Methods inherited from : class `java.util.EventObject`
`getSource, toString`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

NOTIFICATION_TYPE

```
public static final java.lang.String NOTIFICATION_TYPE
```

Constructors

RtlAlertNotification

```
public RtlAlertNotification(java.lang.Object source,
                            java.lang.String Message)
```

Creates a new instance of this Notificaiton where the caller supplies both a Message string, and a source Object. While extrememly useful, care should be taken in using this form of the contructor since the object passed in will be serialized and passed through the Notification sub-system.

Parameters:

`source` -
- the object that has generated this notification.

`Message` -
- a caller provided message.

RtlAlertNotification

```
public RtlAlertNotification(java.lang.Object source,
                            java.lang.String Message,
                            java.lang.Object userData)
```

Creates a new instance of this Notificaiton where the caller supplies a Message string, a source object, and a user data Object. While extrememly useful, care should be taken in using this form of the contructor since the objects passed in will be serialized and passed through the Notification sub-system.

Parameters:

`source` -
- the object that has generated this notification.

`Message` -
- a caller provided message.

`userData` -
- a caller supplied data object.

com.ibm.retail.si.mgmt.notifications

Class RtlConsumerNotification

```
java.lang.Object
  +-java.util.EventObject
    +-javax.management.Notification
      +-com.ibm.retail.si.mgmt.notifications.RtlNotification
        +-com.ibm.retail.si.mgmt.notifications.RtlConsumerNotification
```

public class **RtlConsumerNotification**
extends RtlNotification

This Notificaiton type is used by application components that wish to record entries relevant to a consumers activity, in a consumer log.

See Also:
javax.management.Notification, com.ibm.retail.si.mgmt.notifications.RtlNotification

Field Summary

static java.lang.String	NOTIFICATION_TYPE

Fields inherited from : class `javax.management.Notification`

source

Fields inherited from : class `java.util.EventObject`

source

Constructor Summary

`RtlConsumerNotification(java.lang.Object source,java.lang.String source)`

Creates a new instance of this Notification where the caller supplies both a Message string, and a source Object.

`RtlConsumerNotification(java.lang.Object source,java.lang.String source,java.lang.Object source)`

Creates a new instance of this Notification where the caller supplies a Message string, a source object, and a user data Object.

Methods inherited from : class `com.ibm.retail.si.mgmt.notifications.RtlNotification`

getOriginatingDevice, GetSystemSequenceNo, GetSystemTimeStamp, setOriginatingDevice, SetSystemSequenceNo, SetSystemTimeStamp

Methods inherited from : class `javax.management.Notification`

getMessage, getSequenceNumber, getSource, getTimeStamp, getType, getUserData, setSequenceNumber, setSource, setTimeStamp, setUserData, toString

Methods inherited from : class `java.util.EventObject`
`getSource, toString`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

NOTIFICATION_TYPE

```
public static final java.lang.String NOTIFICATION_TYPE
```

Constructors

RtlConsumerNotification

```
public RtlConsumerNotification(java.lang.Object source,
                               java.lang.String Message)
```

Creates a new instance of this Notification where the caller supplies both a Message string, and a source Object. While extremely useful, care should be taken in using this form of the constructor since the object passed in will be serialized and passed through the Notification sub-system.

Parameters:

`source -`
- the object that has generated this notification.

`Message -`
- a caller provided message.

RtlConsumerNotification

```
public RtlConsumerNotification(java.lang.Object source,
                               java.lang.String Message,
                               java.lang.Object userData)
```

Creates a new instance of this Notification where the caller supplies a Message string, a source object, and a user data Object. While extremely useful, care should be taken in using this form of the constructor since the objects passed in will be serialized and passed through the Notification sub-system.

Parameters:

`source -`
- the object that has generated this notification.

`Message -`
- a caller provided message.

`userData -`
- a caller supplied data object.

com.ibm.retail.si.mgmt.notifications

Class RtlCriticalNotification

```
java.lang.Object
  +-java.util.EventObject
    +-javax.management.Notification
      +-com.ibm.retail.si.mgmt.notifications.RtlNotification
        +-com.ibm.retail.si.mgmt.notifications.RtlCriticalNotification
```

public class **RtlCriticalNotification**
extends RtlNotification

This is used to represent a Critical event that has occurred in the system. Critical events are defined as those things that need immediate attention, and that represent a serious compromise to the integrity of the device. It is advisable that devices wishing to define Critical Notifications that are unique to themselves, sub-class this Class to make the resulting Notification filterable at a finer granularity, and more usable by a management application.

See Also:
javax.management.Notification, com.ibm.retail.si.mgmt.notifications.RtlNotification

Field Summary

static java.lang.String	NOTIFICATION_TYPE

Fields inherited from : class `javax.management.Notification`

source

Fields inherited from : class `java.util.EventObject`

source

Constructor Summary

`RtlCriticalNotification(java.lang.Object source,java.lang.String source)`

Creates a new instance of this Notificaiton where the caller supplies both a Message string, and a source Object.

`RtlCriticalNotification(java.lang.Object source,java.lang.String source,java.lang.Object source)`

Creates a new instance of this Notificaiton where the caller supplies a Message string, a source object, and a user data Object.

Methods inherited from : class `com.ibm.retail.si.mgmt.notifications.RtlNotification`

getOriginatingDevice, GetSystemSequenceNo, GetSystemTimeStamp, setOriginatingDevice, SetSystemSequenceNo, SetSystemTimeStamp

Methods inherited from : class `javax.management.Notification`

`getMessage, getSequenceNumber, getSource, getTimeStamp, getType, getUserData, setSequenceNumber, setSource, setTimeStamp, setUserData, toString`

Methods inherited from : class `java.util.EventObject`
`getSource, toString`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

NOTIFICATION_TYPE

```
public static final java.lang.String NOTIFICATION_TYPE
```

Constructors

RtlCriticalNotification

```
public RtlCriticalNotification(java.lang.Object source,
                               java.lang.String Message)
```

Creates a new instance of this Notificaiton where the caller supplies both a Message string, and a source Object. While extrememly useful, care should be taken in using this form of the contructor since the object passed in will be serialized and passed through the Notification sub-system.

Parameters:

`source` -
- the object that has generated this notification.

`Message` -
- a caller provided message.

RtlCriticalNotification

```
public RtlCriticalNotification(java.lang.Object source,
                               java.lang.String Message,
                               java.lang.Object userData)
```

Creates a new instance of this Notificaiton where the caller supplies a Message string, a source object, and a user data Object. While extrememly useful, care should be taken in using this form of the contructor since the objects passed in will be serialized and passed through the Notification sub-system.

Parameters:

`source` -
- the object that has generated this notification.

`Message` -
- a caller provided message.

`userData` -
- a caller supplied data object.

com.ibm.retail.si.mgmt.notifications

Class RtlDebugNotification

```
java.lang.Object
  +-java.util.EventObject
    +-javax.management.Notification
      +-com.ibm.retail.si.mgmt.notifications.RtlNotification
        +-com.ibm.retail.si.mgmt.notifications.RtlDebugNotification
```

public class **RtlDebugNotification**
extends RtlNotification

This is to be used by development to pass debug information up through the system. It is advisable that devices wishing to define Debug Notifications that are unique to themselves, sub-class this Class to make the resulting Notification filterable at a finer granularity, and more usable by a management application.

See Also:
javax.management.Notification, com.ibm.retail.si.mgmt.notifications.RtlNotification

Field Summary

static java.lang.String	NOTIFICATION_TYPE

Fields inherited from : class `javax.management.Notification`

source

Fields inherited from : class `java.util.EventObject`

source

Constructor Summary

`RtlDebugNotification(java.lang.Object source,java.lang.String source)`

Creates a new instance of this Notification where the caller supplies both a Message string, and a source Object.

`RtlDebugNotification(java.lang.Object source,java.lang.String source,java.lang.Object source)`

Creates a new instance of this Notification where the caller supplies a Message string, a source object, and a user data Object.

Methods inherited from : class `com.ibm.retail.si.mgmt.notifications.RtlNotification`

getOriginatingDevice, GetSystemSequenceNo, GetSystemTimeStamp, setOriginatingDevice, SetSystemSequenceNo, SetSystemTimeStamp

Methods inherited from : class `javax.management.Notification`

getMessage, getSequenceNumber, getSource, getTimeStamp, getType, getUserData, setSequenceNumber, setSource, setTimeStamp, setUserData, toString

Methods inherited from : class `java.util.EventObject`
`getSource, toString`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

NOTIFICATION_TYPE

```
public static final java.lang.String NOTIFICATION_TYPE
```

Constructors

RtlDebugNotification

```
public RtlDebugNotification(java.lang.Object source,
                            java.lang.String Message)
```

Creates a new instance of this Notification where the caller supplies both a Message string, and a source Object. While extremely useful, care should be taken in using this form of the constructor since the object passed in will be serialized and passed through the Notification sub-system.

Parameters:

`source` -
- the object that has generated this notification.

`Message` -
- a caller provided message.

RtlDebugNotification

```
public RtlDebugNotification(java.lang.Object source,
                            java.lang.String Message,
                            java.lang.Object userData)
```

Creates a new instance of this Notification where the caller supplies a Message string, a source object, and a user data Object. While extremely useful, care should be taken in using this form of the constructor since the objects passed in will be serialized and passed through the Notification sub-system.

Parameters:

`source` -
- the object that has generated this notification.

`Message` -
- a caller provided message.

`userData` -
- a caller supplied data object.

com.ibm.retail.si.mgmt.notifications

Class RtlEmergencyNotification

```
java.lang.Object
  +-java.util.EventObject
    +-javax.management.Notification
      +-com.ibm.retail.si.mgmt.notifications.RtlNotification
        +-com.ibm.retail.si.mgmt.notifications.RtlEmergencyNotification
```

public class **RtlEmergencyNotification**
extends RtlNotification

This is used to represent an Emergency condition that has occurred in the system. Emergency events are defined as those things that require immediate attention but are not terminal to the operation of the system. It is advisable that devices wishing to define Emergency Notifications that are unique to themselves, sub-class this Class to make the resulting Notification filterable at a finer granularity, and more usable by a management application.

See Also:
javax.management.Notification, com.ibm.retail.si.mgmt.notifications.RtlNotification

Field Summary

static java.lang.String	NOTIFICATION_TYPE

Fields inherited from : class `javax.management.Notification`

source

Fields inherited from : class `java.util.EventObject`

source

Constructor Summary

`RtlEmergencyNotification(java.lang.Object source,java.lang.String source)`

Creates a new instance of this Notificaiton where the caller supplies both a Message string, and a source Object.

`RtlEmergencyNotification(java.lang.Object source,java.lang.String source,java.lang.Object source)`

Creates a new instance of this Notificaiton where the caller supplies a Message string, a source object, and a user data Object.

Methods inherited from : class `com.ibm.retail.si.mgmt.notifications.RtlNotification`

getOriginatingDevice, GetSystemSequenceNo, GetSystemTimeStamp, setOriginatingDevice, SetSystemSequenceNo, SetSystemTimeStamp

Methods inherited from : class `javax.management.Notification`

`getMessage, getSequenceNumber, getSource, getTimeStamp, getType, getUserData, setSequenceNumber, setSource, setTimeStamp, setUserData, toString`

Methods inherited from : class `java.util.EventObject`
`getSource, toString`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

NOTIFICATION_TYPE

```
public static final java.lang.String NOTIFICATION_TYPE
```

Constructors

RtlEmergencyNotification

```
public RtlEmergencyNotification(java.lang.Object source,
                                java.lang.String Message)
```

Creates a new instance of this Notificaiton where the caller supplies both a Message string, and a source Object. While extrememly useful, care should be taken in using this form of the contructor since the object passed in will be serialized and passed through the Notification sub-system.

Parameters:

`source` -
- the object that has generated this notification.

`Message` -
- a caller provided message.

RtlEmergencyNotification

```
public RtlEmergencyNotification(java.lang.Object source,
                                java.lang.String Message,
                                java.lang.Object userData)
```

Creates a new instance of this Notificaiton where the caller supplies a Message string, a source object, and a user data Object. While extrememly useful, care should be taken in using this form of the contructor since the objects passed in will be serialized and passed through the Notification sub-system.

Parameters:

`source` -
- the object that has generated this notification.

`Message` -
- a caller provided message.

`userData` -
- a caller supplied data object.

com.ibm.retail.si.mgmt.notifications

Class RtlErrorNotification

```
java.lang.Object
  +-java.util.EventObject
    +-javax.management.Notification
      +-com.ibm.retail.si.mgmt.notifications.RtlNotification
        +-com.ibm.retail.si.mgmt.notifications.RtlErrorNotification
```

public class **RtlErrorNotification**
extends RtlNotification

This is used to represent an Error condition that has occurred in the system. Error events are defined as an unexpected erroneous condition that the system detected and handled. It is advisable that devices wishing to define Error Notifications that are unique to themselves, sub-class this Class to make the resulting Notification filterable at a finer granularity, and more usable by a management application.

See Also:
javax.management.Notification, com.ibm.retail.si.mgmt.notifications.RtlNotification

Field Summary

static java.lang.String	NOTIFICATION_TYPE

Fields inherited from : class `javax.management.Notification`

source

Fields inherited from : class `java.util.EventObject`

source

Constructor Summary

`RtlErrorNotification(java.lang.Object source,java.lang.String source)`

Creates a new instance of this Notificaiton where the caller supplies both a Message string, and a source Object.

`RtlErrorNotification(java.lang.Object source,java.lang.String source,java.lang.Object source)`

Creates a new instance of this Notificaiton where the caller supplies a Message string, a source object, and a user data Object.

Methods inherited from : class `com.ibm.retail.si.mgmt.notifications.RtlNotification`

getOriginatingDevice, GetSystemSequenceNo, GetSystemTimeStamp, setOriginatingDevice, SetSystemSequenceNo, SetSystemTimeStamp

Methods inherited from : class `javax.management.Notification`

`getMessage, getSequenceNumber, getSource, getTimeStamp, getType, getUserData, setSequenceNumber, setSource, setTimeStamp, setUserData, toString`

Methods inherited from : class `java.util.EventObject`
`getSource, toString`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

NOTIFICATION_TYPE

```
public static final java.lang.String NOTIFICATION_TYPE
```

Constructors

RtlErrorNotification

```
public RtlErrorNotification(java.lang.Object source,
                            java.lang.String Message)
```

Creates a new instance of this Notificaiton where the caller supplies both a Message string, and a source Object. While extrememly useful, care should be taken in using this form of the contructor since the object passed in will be serialized and passed through the Notification sub-system.

Parameters:

`source` -
- the object that has generated this notification.

`Message` -
- a caller provided message.

RtlErrorNotification

```
public RtlErrorNotification(java.lang.Object source,
                            java.lang.String Message,
                            java.lang.Object userData)
```

Creates a new instance of this Notificaiton where the caller supplies a Message string, a source object, and a user data Object. While extrememly useful, care should be taken in using this form of the contructor since the objects passed in will be serialized and passed through the Notification sub-system.

Parameters:

`source` -
- the object that has generated this notification.

`Message` -
- a caller provided message.

`userData` -
- a caller supplied data object.

com.ibm.retail.si.mgmt.notifications

Class RtlInformationNotification

```
java.lang.Object
  +-java.util.EventObject
    +-javax.management.Notification
      +-com.ibm.retail.si.mgmt.notifications.RtlNotification
        +-com.ibm.retail.si.mgmt.notifications.RtlInformationNotification
```

Direct Known Subclasses:
MgmtSDStartedNotification, MgmtSDProgressNotification, MgmtSDCompletionNotification, AgentShutdownNotification, AgentLostNotification, AgentDiscoveredNotification, AgentConnectionFailedNotification

---

public class **RtlInformationNotification**
extends RtlNotification

This is used to represent a condition within the system that an appication might be interested in. Information events are defined as a condition that is not detrimental to the operation of the system but rather that something in the system has changed. That change could be configuration or status related. It is advisable that devices wishing to define Information Notifications that are unique to themselves, sub-class this Class to make the resulting Notification filterable at a finer granularity, and more usable by a management application.

See Also:
javax.management.Notification, com.ibm.retail.si.mgmt.notifications.RtlNotification

---

Field Summary

static java.lang.String	NOTIFICATION_TYPE

Fields inherited from : class `javax.management.Notification`

source

Fields inherited from : class `java.util.EventObject`

source

Constructor Summary

`RtlInformationNotification(java.lang.Object source,java.lang.String source)`

Creates a new instance of this Notification where the caller supplies both a Message string, and a source Object.

`RtlInformationNotification(java.lang.Object source,java.lang.String source,java.lang.Object source)`

Creates a new instance of this Notification where the caller supplies a Message string, a source object, and a user data Object.

Methods inherited from : class `com.ibm.retail.si.mgmt.notifications.RtlNotification`

`getOriginatingDevice, GetSystemSequenceNo, GetSystemTimeStamp, setOriginatingDevice, SetSystemSequenceNo, SetSystemTimeStamp`

Methods inherited from : class `javax.management.Notification`
`getMessage, getSequenceNumber, getSource, getTimeStamp, getType, getUserData, setSequenceNumber, setSource, setTimeStamp, setUserData, toString`

Methods inherited from : class `java.util.EventObject`
`getSource, toString`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

NOTIFICATION_TYPE

```
public static final java.lang.String NOTIFICATION_TYPE
```

Constructors

RtlInformationNotification

```
public RtlInformationNotification(java.lang.Object source,
                                  java.lang.String Message)
```

Creates a new instance of this Notification where the caller supplies both a Message string, and a source Object. While extremely useful, care should be taken in using this form of the constructor since the object passed in will be serialized and passed through the Notification sub-system.

Parameters:

`source -`
- the object that has generated this notification.

`Message -`
- a caller provided message.

RtlInformationNotification

```
public RtlInformationNotification(java.lang.Object source,
                                  java.lang.String Message,
                                  java.lang.Object userData)
```

Creates a new instance of this Notification where the caller supplies a Message string, a source object, and a user data Object. While extremely useful, care should be taken in using this form of the constructor since the objects passed in will be serialized and passed through the Notification sub-system.

Parameters:

`source -`
- the object that has generated this notification.

`Message -`
- a caller provided message.

(continued from last page)

`userData -`
- a caller supplied data object.

com.ibm.retail.si.mgmt.notifications

Class RtlNoticeNotification

```
java.lang.Object
  +-java.util.EventObject
    +-javax.management.Notification
      +-com.ibm.retail.si.mgmt.notifications.RtlNotification
        +-com.ibm.retail.si.mgmt.notifications.RtlNoticeNotification
```

public class **RtlNoticeNotification**
extends RtlNotification

This is used to represent a condition defined as as a less severe form of Warning. Notice events are defined as a condition that is not detrimental to the operation of the system but that rather that a parameter of operation in the system is getting close to a point where it could become a problem. It is advisable that devices wishing to define Notice Notifications that are unique to themselves, sub-class this Class to make the resulting Notification filterable at a finer granularity, and more usable by a management application.

See Also:
javax.management.Notification, com.ibm.retail.si.mgmt.notifications.RtlNotification

Field Summary

static java.lang.String	NOTIFICATION_TYPE

Fields inherited from : class `javax.management.Notification`

source

Fields inherited from : class `java.util.EventObject`

source

Constructor Summary

`RtlNoticeNotification(java.lang.Object source,java.lang.String source)`

Creates a new instance of this Notification where the caller supplies both a Message string, and a source Object.

`RtlNoticeNotification(java.lang.Object source,java.lang.String source,java.lang.Object source)`

Creates a new instance of this Notification where the caller supplies a Message string, a source object, and a user data Object.

Methods inherited from : class `com.ibm.retail.si.mgmt.notifications.RtlNotification`

getOriginatingDevice, GetSystemSequenceNo, GetSystemTimeStamp, setOriginatingDevice, SetSystemSequenceNo, SetSystemTimeStamp

Methods inherited from : class `javax.management.Notification`

`getMessage, getSequenceNumber, getSource, getTimeStamp, getType, getUserData, setSequenceNumber, setSource, setTimeStamp, setUserData, toString`

Methods inherited from : class `java.util.EventObject`
`getSource, toString`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

NOTIFICATION_TYPE

```
public static final java.lang.String NOTIFICATION_TYPE
```

Constructors

RtlNoticeNotification

```
public RtlNoticeNotification(java.lang.Object source,
                             java.lang.String Message)
```

Creates a new instance of this Notification where the caller supplies both a Message string, and a source Object. While extremely useful, care should be taken in using this form of the constructor since the object passed in will be serialized and passed through the Notification sub-system.

Parameters:

`source -`
- the object that has generated this notification.

`Message -`
- a caller provided message.

RtlNoticeNotification

```
public RtlNoticeNotification(java.lang.Object source,
                             java.lang.String Message,
                             java.lang.Object userData)
```

Creates a new instance of this Notification where the caller supplies a Message string, a source object, and a user data Object. While extremely useful, care should be taken in using this form of the constructor since the objects passed in will be serialized and passed through the Notification sub-system.

Parameters:

`source -`
- the object that has generated this notification.

`Message -`
- a caller provided message.

`userData -`
- a caller supplied data object.

com.ibm.retail.si.mgmt.notifications

Class RtlNotification

```
java.lang.Object
  +-java.util.EventObject
    +-javax.management.Notification
      +-com.ibm.retail.si.mgmt.notifications.RtlNotification
```

Direct Known Subclasses:
RtlWarningNotification, RtlTracePointNotification, RtlNoticeNotification, RtlInformationNotification, RtlErrorNotification, RtlEmergencyNotification, RtlDebugNotification, RtlCriticalNotification, RtlConsumerNotification, RtlAlertNotification

---

public abstract class **RtlNotification**
extends javax.management.Notification

This class is intended to be the base class used for all notifcations issued by all RSS components. It should never be instanciated itself, but rather one of it's derived classes should always be used. Base Notification classes are defined by this architecture that have the following order of significance: RtlCriticalNotification RtlEmergencyNotification RtlAlertNotification RtlErrorNotification RtlWarningNotification RtlNoticeNotification RtlInformationNotification RtlDebugNotification RtlTracePointNotification RtlConsumerNotification

See Also:
javax.management.Notification, com.ibm.retail.si.mgmt.notifications.RtlCriticalNotification, com.ibm.retail.si.mgmt.notifications.RtlEmergencyNotification, com.ibm.retail.si.mgmt.notifications.RtlAlertNotification, com.ibm.retail.si.mgmt.notifications.RtlErrorNotification, com.ibm.retail.si.mgmt.notifications.RtlWarningNotification, com.ibm.retail.si.mgmt.notifications.RtlNoticeNotification, com.ibm.retail.si.mgmt.notifications.RtlInformationNotification, com.ibm.retail.si.mgmt.notifications.RtlDebugNotification, com.ibm.retail.si.mgmt.notifications.RtlTracePointNotification, com.ibm.retail.si.mgmt.notifications.RtlConsumerNotification

---

Fields inherited from : class `javax.management.Notification`
`source`

Fields inherited from : class `java.util.EventObject`
`source`

Constructor Summary

`RtlNotification(java.lang.Object source,java.lang.String source)`
`RtlNotification(java.lang.Object source,java.lang.String source,java.lang.String source)`
`RtlNotification(java.lang.Object source,java.lang.String source,java.lang.String source,java.lang.Object source)`

Method Summary

MgmtDeviceInfo	`getOriginatingDevice()` Get information about the device that generated the notification.
long	`GetSystemSequenceNo()` Get the System level assigned sequence number.
long	`GetSystemTimeStamp()` Get the System level assigned Timestamp.
void	`setOriginatingDevice(MgmtDeviceInfo deviceInfo)` Used by the Master Agent to set the originating This is assigned at the time this Notification is received by the Master Agent.
void	`SetSystemSequenceNo(long SeqNo)` Used by the Master Agent to set a system wide sequence number.
void	`SetSystemTimeStamp(long Stamp)` Used by the Master Agent to set a system wide timestamp.

Methods inherited from : class `javax.management.Notification`
`getMessage, getSequenceNumber, getSource, getTimeStamp, getType, getUserData, setSequenceNumber, setSource, setTimeStamp, setUserData, toString`

Methods inherited from : class `java.util.EventObject`
`getSource, toString`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

RtlNotification

```
protected RtlNotification(java.lang.Object source,
                          java.lang.String Type)
```

RtlNotification

```
protected RtlNotification(java.lang.Object source,
                          java.lang.String Type,
                          java.lang.String Message)
```

(continued from last page)

RtlNotification

```
protected RtlNotification(java.lang.Object source,
                          java.lang.String Type,
                          java.lang.String Message,
                          java.lang.Object userData)
```

Methods

GetSystemSequenceNo

```
public long GetSystemSequenceNo()
```

Get the System level assigned sequence number. This number is assigned to this notification by the Master Agent upon receipt.

Returns:

long, the system wide sequence number.

SetSystemSequenceNo

```
protected void SetSystemSequenceNo(long SeqNo)
```

Used by the Master Agent to set a system wide sequence number. This number is assigned at the time this Notification is received by the Master Agent.

Parameters:

`SeqNo` -
- The system wide sequence number.

GetSystemTimeStamp

```
public long GetSystemTimeStamp()
```

Get the System level assigned Timestamp. This timestamp is assigned to this notification by the Master Agent upon receipt. This timestamp is relative to the MasterAgent, and is meant to give system- wide context to all notificaitons.

Returns:

SystemtStamp, the system wide timestamp.

SetSystemTimeStamp

```
protected void SetSystemTimeStamp(long Stamp)
```

Used by the Master Agent to set a system wide timestamp. This is assigned at the time this Notification is received by the Master Agent.

Parameters:

`Stamp` -
- The system wide timestamp.

---

getOriginatingDevice

```
public MgmtDeviceInfo getOriginatingDevice()
```

Get information about the device that generated the notification. This value will be assigned when the notification is received by the MasterAgent

Returns:

(continued from last page)

deviceInfo - Source device information

setOriginatingDevice

`protected void` **`setOriginatingDevice`**`(MgmtDeviceInfo deviceInfo)`

Used by the Master Agent to set the originating This is assigned at the time this Notification is received by the Master Agent.

Parameters:

`deviceInfo -`
- Source device information

com.ibm.retail.si.mgmt.notifications

Class RtlTracePointNotification

```
java.lang.Object
  +-java.util.EventObject
    +-javax.management.Notification
      +-com.ibm.retail.si.mgmt.notifications.RtlNotification
        +-com.ibm.retail.si.mgmt.notifications.RtlTracePointNotification
```

public class **RtlTracePointNotification**
extends RtlNotification

This Notificaiton type is used by the logging component of the General and Master agent for converting locally logged entries to notificaitons that can be forwarded through the centralized notification system. An application should not need to ever instanciate one of these directly. NOTE that timestamp, and sequence number are provided in the base class and should be set too.

See Also:
javax.management.Notification, com.ibm.retail.si.mgmt.notifications.RtlNotification

Fields inherited from : class `javax.management.Notification`
`source`

Fields inherited from : class `java.util.EventObject`
`source`

Constructor Summary

`RtlTracePointNotification(java.lang.Object source,java.lang.String source)` Creates a new instance of this Notification where the caller supplies both a Message string, and a source Object.
`RtlTracePointNotification(java.lang.Object source,java.lang.String source,java.lang.Object source)` Creates a new instance of this Notification where the caller supplies a Message string, a source object, and a user data Object.

Method Summary

`java.lang.String`	`getFileName()`
`int`	`getLevel()`
`java.lang.String`	`getLineNumber()`
`java.lang.String`	`getMethodName()`
`java.lang.String`	`getNdc()`

java.lang.String	getOriginator()
java.lang.String	getSourceClassName()
java.lang.String	getTaskName()
int	getThreadId()
java.lang.Throwable	getThrowable()
void	setFileName(java.lang.String string)
void	setLevel(int Level) Sets the level or severity of the log message.
void	setLineNumber(java.lang.String string)
void	setMethodName(java.lang.String string)
void	setNdc(java.lang.String ndc)
void	setOriginator(java.lang.String originator)
void	setSourceClassName(java.lang.String className)
void	setTaskName(java.lang.String Name)
void	setThreadId(int threadId)
void	setThrowable(java.lang.Throwable Thrown)

Methods inherited from : class com.ibm.retail.si.mgmt.notifications.RtlNotification
getOriginatingDevice, GetSystemSequenceNo, GetSystemTimeStamp, setOriginatingDevice, SetSystemSequenceNo, SetSystemTimeStamp

Methods inherited from : class javax.management.Notification
getMessage, getSequenceNumber, getSource, getTimeStamp, getType, getUserData, setSequenceNumber, setSource, setTimeStamp, setUserData, toString

Methods inherited from : class java.util.EventObject
getSource, toString

Methods inherited from : class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

RtlTracePointNotification

```
public RtlTracePointNotification(java.lang.Object source,
                                 java.lang.String Message)
```

Creates a new instance of this Notification where the caller supplies both a Message string, and a source Object. While extremely useful, care should be taken in using this form of the constructor since the object passed in will be serialized and passed through the Notification sub-system.

Parameters:

`source -`
- the object that has generated this notification.

`Message -`
- a caller provided message.

RtlTracePointNotification

```
public RtlTracePointNotification(java.lang.Object source,
                                 java.lang.String Message,
                                 java.lang.Object userData)
```

Creates a new instance of this Notification where the caller supplies a Message string, a source object, and a user data Object. While extremely useful, care should be taken in using this form of the constructor since the objects passed in will be serialized and passed through the Notification sub-system.

Parameters:

`source -`
- the object that has generated this notification.

`Message -`
- a caller provided message.

`userData -`
- a caller supplied data object.

Methods

getLevel

```
public int getLevel()
```

Returns:

int - The level or severity of the message, as defined by the levels in MgmtLoggingCtrlMBean

setLevel

```
public void setLevel(int Level)
```

Sets the level or severity of the log message. Should be obtained from MgmtLoggingCtrlMBean

Parameters:

`Level -`
- int level, from MgmtLoggingCtrlMBean

getTaskName

```
public java.lang.String getTaskName()
```

(continued from last page)

Returns:

String - The name of the task where the message originated, which correspongs to the thread description from log4j, the tag in the message from syslog, or getsourcemethodname on the JDK 1.4 logger

setTaskName

```
public void setTaskName(java.lang.String Name)
```

Parameters:

`Name -`
- The name of the task where the log message originated

See Also:

#getTaskName()

getOriginator

```
public java.lang.String getOriginator()
```

Returns:

String - Identifier indicating the logging category or entity taht logged the message. This corresponds to the Facility on syslog, getcategory() on log4j, and loggername on the JDK 1.4 logger

setOriginator

```
public void setOriginator(java.lang.String originator)
```

Parameters:

`originator -`
- Sets the originator for this message

See Also:

#getOriginator()

getThrowable

```
public java.lang.Throwable getThrowable()
```

Returns:

throwable - For error messages, the associated Throwable (Not used in Syslog)

setThrowable

```
public void setThrowable(java.lang.Throwable Thrown)
```

Parameters:

(continued from last page)

`Thrown -`
- For error messages, the associated Throwable (Not used in Syslog)

getSourceClassName

`public java.lang.String` **`getSourceClassName`**`()`

Returns:

String - Class that allegedy issued the logging request (JDK and Log4J only)

setSourceClassName

`public void` **`setSourceClassName`**`(java.lang.String className)`

Parameters:

`className -`
- Class that allegedy issued the logging request (JDK and Log4J only)

getFileName

`public java.lang.String` **`getFileName`**`()`

Returns:

File name where the event occurred

getLineNumber

`public java.lang.String` **`getLineNumber`**`()`

Returns:

Line number within the file where the event occurred

getMethodName

`public java.lang.String` **`getMethodName`**`()`

Returns:

Name of the method where the event occurred

setFileName

`public void` **`setFileName`**`(java.lang.String string)`

Parameters:

`string -`
File name where the event occurred

setLineNumber

```
public void setLineNumber(java.lang.String string)
```

Parameters:

`string -`
Line number within the file where the event occurred

setMethodName

```
public void setMethodName(java.lang.String string)
```

Parameters:

`Name -`
of the method where the event occurred

getNdc

```
public java.lang.String getNdc()
```

Returns:

ndc - The NDC (Nested Diagnostic Context) for this record (Log4J only)

setNdc

```
public void setNdc(java.lang.String ndc)
```

Parameters:

`ndc -`
- The NDC (Nested Diagnostic Context) for this record (Log4J only)

getThreadId

```
public int getThreadId()
```

Returns:

threadId - Thread identifier from where the message originated (JDK only)

setThreadId

```
public void setThreadId(int threadId)
```

Parameters:

`threadId -`
- Thread identifier from where the message originated (JDK only)

com.ibm.retail.si.mgmt.notifications

Class RtlWarningNotification

```
java.lang.Object
  +-java.util.EventObject
    +-javax.management.Notification
      +-com.ibm.retail.si.mgmt.notifications.RtlNotification
        +-com.ibm.retail.si.mgmt.notifications.RtlWarningNotification
```

public class **RtlWarningNotification**
extends RtlNotification

This is used to represent a condition that has occured on the system that warrants a Warning. Warning events are defined as a condition that is not detrimental to the operation of the system but that rather that a parameter of operation in the system is getting close to a point where it could become a problem. It is advisable that devices wishing to define Warning Notifications that are unique to themselves, sub-class this Class to make the resulting Notification filterable at a finer granularity, and more usable by a management application.

See Also:
javax.management.Notification, com.ibm.retail.si.mgmt.notifications.RtlNotification

Field Summary

static java.lang.String	NOTIFICATION_TYPE

Fields inherited from : class `javax.management.Notification`

source

Fields inherited from : class `java.util.EventObject`

source

Constructor Summary

`RtlWarningNotification(java.lang.Object source,java.lang.String source)`

Creates a new instance of this Notification where the caller supplies both a Message string, and a source Object.

`RtlWarningNotification(java.lang.Object source,java.lang.String source,java.lang.Object source)`

Creates a new instance of this Notification where the caller supplies a Message string, a source object, and a user data Object.

Methods inherited from : class `com.ibm.retail.si.mgmt.notifications.RtlNotification`

getOriginatingDevice, GetSystemSequenceNo, GetSystemTimeStamp, setOriginatingDevice, SetSystemSequenceNo, SetSystemTimeStamp

Methods inherited from : class `javax.management.Notification`

`getMessage, getSequenceNumber, getSource, getTimeStamp, getType, getUserData, setSequenceNumber, setSource, setTimeStamp, setUserData, toString`

Methods inherited from : class `java.util.EventObject`
`getSource, toString`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

NOTIFICATION_TYPE

```
public static final java.lang.String NOTIFICATION_TYPE
```

Constructors

RtlWarningNotification

```
public RtlWarningNotification(java.lang.Object source,
                              java.lang.String Message)
```

Creates a new instance of this Notification where the caller supplies both a Message string, and a source Object. While extremely useful, care should be taken in using this form of the constructor since the object passed in will be serialized and passed through the Notification sub-system.

Parameters:

`source` -
- the object that has generated this notification.

`Message` -
- a caller provided message.

RtlWarningNotification

```
public RtlWarningNotification(java.lang.Object source,
                              java.lang.String Message,
                              java.lang.Object userData)
```

Creates a new instance of this Notification where the caller supplies a Message string, a source object, and a user data Object. While extremely useful, care should be taken in using this form of the constructor since the objects passed in will be serialized and passed through the Notification sub-system.

Parameters:

`source` -
- the object that has generated this notification.

`Message` -
- a caller provided message.

`userData` -
- a caller supplied data object.

com.ibm.retail.si.mgmt.notifications

Class StoredNotification

```
java.lang.Object
   |
   +-com.ibm.retail.si.mgmt.notifications.StoredNotification
```

All Implemented interfaces:
java.io.Serializable

---

public class **StoredNotification**
extends java.lang.Object
implements java.io.Serializable

Stored Notification represents a notification to be stored persistently

---

Constructor Summary

`StoredNotification(javax.management.Notification notification,MgmtDeviceInfo notification,int notification)`

Method Summary

`MgmtDeviceInfo`	`getDeviceInfo()`
`int`	`getDeviceType()`
`javax.management.Notification`	`getNotification()`
`void`	`setDeviceInfo(MgmtDeviceInfo devInfo)`
`void`	`setDeviceType(int devType)`
`void`	`setNotification(javax.management.Notification notification)`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

StoredNotification

```
public StoredNotification(javax.management.Notification notification,
                          MgmtDeviceInfo devInfo,
                          int devType)
```

(continued from last page)

Parameters:
`notification`
`devInfo`
`devType`

Methods

getDeviceInfo

`public MgmtDeviceInfo **getDeviceInfo**()`

Returns:

MgmtDeviceInfo

getDeviceType

`public int **getDeviceType**()`

Returns:

int

getNotification

`public javax.management.Notification **getNotification**()`

Returns:

Notification

setDeviceInfo

`public void **setDeviceInfo**(MgmtDeviceInfo devInfo)`

Parameters:
`devInfo`

setDeviceType

`public void **setDeviceType**(int devType)`

Parameters:
`devType`

setNotification

`public void **setNotification**(javax.management.Notification notification)`

(continued from last page)

Parameters:

`notification`

Package
com.ibm.retail.si.mgmt.svc

Classes for running the SIF Management Agents as a service.

com.ibm.retail.si.mgmt.svc

Class GeneralAgentService

```
java.lang.Object
   |
   +-com.ibm.retail.si.mgmt.svc.GeneralAgentService
```

public class **GeneralAgentService**
extends java.lang.Object

Constructor Summary

`GeneralAgentService()`

Method Summary

`static void`	`main(java.lang.String[] args)`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

GeneralAgentService

```
public GeneralAgentService()
```

Methods

main

```
public static void main(java.lang.String[] args)
                   throws java.lang.Exception
```

com.ibm.retail.si.mgmt.svc

Class ManagementAgentService

```
java.lang.Object
   |
   +-com.ibm.retail.si.mgmt.svc.ManagementAgentService
```

public class **ManagementAgentService**
extends java.lang.Object

Class that runs a MgmtAgent instance as a service by executing a simple health task in a Timer

Constructor Summary

`ManagementAgentService(MgmtAgent agent)`

Method Summary

`long`	`getTaskFrequency()`
`void`	`setTaskFrequency(long l)`
`void`	`start()` Starts the timer with the agent health task
`void`	`stop()` Cancels the health task timer

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

ManagementAgentService

`public` **`ManagementAgentService`**`(MgmtAgent agent)`

Methods

start

`public void` **`start`**`()`

Starts the timer with the agent health task

stop

`public void stop()`

Cancels the health task timer

getTaskFrequency

`public long getTaskFrequency()`

Returns:

long The time (in ms) between execution of each agent health task

setTaskFrequency

`public void setTaskFrequency(long l)`

Parameters:

`l -`
New task frequency (in ms). Service must be restarted for changes to take effect

See Also:

#getTaskFrequency()

com.ibm.retail.si.mgmt.svc

Class MasterAgentService

```
java.lang.Object
   |
   +-com.ibm.retail.si.mgmt.svc.MasterAgentService
```

public class **MasterAgentService**
extends java.lang.Object

Constructor Summary

`MasterAgentService()`

Method Summary

static void	`main(java.lang.String[] args)`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

MasterAgentService

```
public MasterAgentService()
```

Methods

main

```
public static void main(java.lang.String[] args)
                   throws java.lang.Exception
```

Package
com.ibm.retail.si.mgmt.swdist

Classes for Software Distribution policy control. The `MgmtSoftwareDistMBean`is the primary interface for controlling `SWDistPolicy`, and for triggering installations on general agents.

com.ibm.retail.si.mgmt.swdist

Class DeviceDistributionRecord

```
java.lang.Object
   |
   +-com.ibm.retail.si.mgmt.swdist.DeviceDistributionRecord
```

All Implemented interfaces:
java.io.Serializable

---

public class **DeviceDistributionRecord**
extends java.lang.Object
implements java.io.Serializable

Holds information about the installation/uninstallation of a package on a particular device

---

Constructor Summary

`DeviceDistributionRecord(java.lang.String deviceSystemId,int deviceSystemId,MgmtSftPackage deviceSystemId,boolean deviceSystemId)`
`DeviceDistributionRecord(java.lang.String deviceSystemId,int deviceSystemId,MgmtSftPackage deviceSystemId,boolean deviceSystemId,int deviceSystemId)`
`DeviceDistributionRecord(java.lang.String deviceSystemId,int deviceSystemId,MgmtSftPackage deviceSystemId,boolean deviceSystemId,boolean deviceSystemId,int deviceSystemId,int deviceSystemId)` Constructs a new record

Method Summary

boolean	`equals(java.lang.Object o)` Determines equality based on deviceType, install/uninstall, device system Id, and the target MgmtSftPackage
int	`getBusyCount()`
java.lang.String	`getDeviceSystemId()`
int	`getDeviceType()`
int	`getFailedConnectionCount()`
int	`getIntervalPercentage()`
int	`getRc()`
MgmtSftPackage	`getSwPackage()`
void	`incBusyCount()` Increments the busy count

void	`incFailedConnectionCount()` Increments the failed connection count
boolean	`isCompleted()`
boolean	`isInstall()`
void	`resetBusyCount()` Resets the busy count
void	`resetFailedConnectionCount()` Resets the failed connection count
void	`setCompleted(boolean b)`
void	`setDeviceSystemId(java.lang.String string)`
void	`setDeviceType(int integer)`
void	`setInstall(boolean b)`
void	`setIntervalPercentage(int i)`
void	`setRc(int i)`
void	`setSwPackage(MgmtSftPackage package1)`
java.lang.String	`toString()`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

DeviceDistributionRecord

```
protected DeviceDistributionRecord(java.lang.String deviceSystemId,
                                   int deviceType,
                                   MgmtSftPackage swPackage,
                                   boolean isInstall)
```

(continued from last page)

DeviceDistributionRecord

```
protected DeviceDistributionRecord(java.lang.String deviceSystemId,
                                   int deviceType,
                                   MgmtSftPackage swPackage,
                                   boolean isInstall,
                                   int intervalPercentage)
```

DeviceDistributionRecord

```
protected DeviceDistributionRecord(java.lang.String deviceSystemId,
                                   int deviceType,
                                   MgmtSftPackage swPackage,
                                   boolean isInstall,
                                   boolean completed,
                                   int intervalPercentage,
                                   int rc)
```

Constructs a new record

Parameters:

`deviceSystemId -`
System ID of the target device
`deviceType -`
Device type of target device
`swPackage -`
Package to install
`isInstall -`
True if this is an install, false if an uninstall
`completed -`
Whether or not this distribution has completed
`intervalPercentage -`
Percentage interval for sending notifications. 0 disables them
`rc -`
Return code from the last request.

Methods

toString

```
public java.lang.String toString()
```

equals

```
public boolean equals(java.lang.Object o)
```

Determines equality based on deviceType, install/uninstall, device system Id, and the target MgmtSftPackage

See Also:

Object#equals(java.lang.Object)

MgmtSftPackage#equals(Object)

isCompleted

```
public boolean isCompleted()
```

Returns:

(continued from last page)

boolean True if this distribution has completed, false otherwise

getDeviceSystemId

`public java.lang.String getDeviceSystemId()`

Returns:

String SystemID of the target device

getDeviceType

`public int getDeviceType()`

Returns:

int Device type of the target device

getIntervalPercentage

`public int getIntervalPercentage()`

Returns:

int Interval percentage for status notifications. Setting this to 0 disables status notifications

isInstall

`public boolean isInstall()`

Returns:

boolean True if this is an install, false otherwise

getRc

`public int getRc()`

Returns:

int Return code from the last installation request

getSwPackage

`public MgmtSftPackage getSwPackage()`

Returns:

MgmtSftPackage Target software package for removal/installation

setCompleted

```
public void setCompleted(boolean b)
```

Parameters:
`b`

See Also:

#isCompleted()

setDeviceSystemId

```
public void setDeviceSystemId(java.lang.String string)
```

Parameters:
`string`

See Also:

#getDeviceSystemId()

setDeviceType

```
public void setDeviceType(int integer)
```

Parameters:
`integer`

See Also:

#getDeviceType()

setIntervalPercentage

```
public void setIntervalPercentage(int i)
```

Parameters:
`i`

See Also:

#getIntervalPercentage()

setInstall

```
public void setInstall(boolean b)
```

Parameters:
`b`

(continued from last page)

See Also:

#isInstall()

setRc

```
public void setRc(int i)
```

Parameters:

`i`

See Also:

#getRc()

setSwPackage

```
public void setSwPackage(MgmtSftPackage package1)
```

Parameters:

`package1`

See Also:

#getSwPackage()

getBusyCount

```
public int getBusyCount()
```

Returns:

int The number of times this distribution has been attempted with a busy client response

getFailedConnectionCount

```
public int getFailedConnectionCount()
```

Returns:

int The number of failed connection attempts to the client

incBusyCount

```
public void incBusyCount()
```

Increments the busy count

incFailedConnectionCount

```
public void incFailedConnectionCount()
```

Increments the failed connection count

(continued from last page)

resetBusyCount

`public void` **`resetBusyCount`**`()`

Resets the busy count

resetFailedConnectionCount

`public void` **`resetFailedConnectionCount`**`()`

Resets the failed connection count

com.ibm.retail.si.mgmt.swdist

Class DeviceTypeSWDistPolicy

```
java.lang.Object
   |
   +-com.ibm.retail.si.mgmt.swdist.DeviceTypeSWDistPolicy
```

All Implemented interfaces:
java.io.Serializable

public class **DeviceTypeSWDistPolicy**
extends java.lang.Object
implements java.io.Serializable

Holds policy information for a software distribution based on device type

Constructor Summary

`DeviceTypeSWDistPolicy(int deviceType,MgmtSftPackage deviceType,boolean deviceType)`
`DeviceTypeSWDistPolicy(int deviceType,MgmtSftPackage deviceType,boolean deviceType,int deviceType)`

Method Summary

boolean	`equals(java.lang.Object o)` Determines equality based on install/remove, device type, and the target MgmtSftPackage
int	`getDeviceType()`
int	`getIntervalPercentage()`
MgmtSftPackage	`getSwPackage()`
boolean	`isInstall()`
void	`setDeviceType(int i)`
void	`setInstall(boolean b)`
void	`setIntervalPercentage(int i)`
void	`setSwPackage(MgmtSftPackage package1)`
java.lang.String	`toString()`

Methods inherited from : class `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
wait
```

Constructors

DeviceTypeSWDistPolicy

```
protected DeviceTypeSWDistPolicy(int deviceType,
                                 MgmtSftPackage swPackage,
                                 boolean isInstall)
```

DeviceTypeSWDistPolicy

```
protected DeviceTypeSWDistPolicy(int deviceType,
                                 MgmtSftPackage swPackage,
                                 boolean isInstall,
                                 int intervalPercentage)
```

Parameters:

`deviceType -`
int Device type on which this policy applies
`swPackage -`
MgmtSftPackage Target software package
`isInstall -`
boolean True if the distribution is an install, false if a removal
`intervalPercentage -`
int Percentage interval for status notifications to set on each distribution

Methods

toString

```
public java.lang.String toString()
```

equals

```
public boolean equals(java.lang.Object o)
```

Determines equality based on install/remove, device type, and the target MgmtSftPackage

getDeviceType

```
public int getDeviceType()
```

Returns:

int Device type for this policy

isInstall

```
public boolean isInstall()
```

(continued from last page)

Returns:

boolean True if this is policy is for an install, false for an uninstall

getIntervalPercentage

`public int getIntervalPercentage()`

Returns:

int Interval percentage for status notifications. Setting this to 0 disables status notifications

getSwPackage

`public MgmtSftPackage getSwPackage()`

Returns:

MgmtSftPackage Target software package for removal/installation

setDeviceType

`public void setDeviceType(int i)`

Parameters:
`i`

See Also:

#setDeviceType(int)

setInstall

`public void setInstall(boolean b)`

Parameters:
`b`

See Also:

#setInstall(boolean)

setIntervalPercentage

`public void setIntervalPercentage(int i)`

Parameters:
`i`

See Also:

(continued from last page)

#setIntervalPercentage(int)

setSwPackage

```
public void setSwPackage(MgmtSftPackage package1)
```

Parameters:

`package1`

See Also:

#setSwPackage(MgmtSftPackage)

com.ibm.retail.si.mgmt.swdist

Class MgmtSftComponent

```
java.lang.Object
  |
  +-com.ibm.retail.si.mgmt.swdist.MgmtSftComponent
```

All Implemented interfaces:
java.io.Serializable

public class **MgmtSftComponent**
extends java.lang.Object
implements java.io.Serializable

Class for defining a single file that is part of a software package, and its defining characteristics. This is a data only class with all public elements.

Constructor Summary

`MgmtSftComponent(java.lang.String fileName,java.lang.String fileName,long fileName)`

Method Summary

boolean	`equals(java.lang.Object o)` Determines equality based on size, filename, and path name
java.lang.String	`getFileName()`
java.lang.String	`getPathName()`
long	`getSize()`
void	`setFileName(java.lang.String string)`
void	`setPathName(java.lang.String string)`
void	`setSize(long l)`
java.lang.String	`toString()`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

(continued from last page)

MgmtSftComponent

```
public MgmtSftComponent(java.lang.String fileName,
                        java.lang.String pathName,
                        long size)
```

Methods

toString

```
public java.lang.String toString()
```

equals

```
public boolean equals(java.lang.Object o)
```

Determines equality based on size, filename, and path name

See Also:

Object#equals(java.lang.Object)

getFileName

```
public java.lang.String getFileName()
```

Returns:

String File name of this component

getPathName

```
public java.lang.String getPathName()
```

Returns:

String FTP server path of this component

getSize

```
public long getSize()
```

Returns:

long Size of the file, in bytes

setFileName

```
public void setFileName(java.lang.String string)
```

(continued from last page)

Parameters:
`string`

See Also:

#setFileName(String)

setPathName

```
public void setPathName(java.lang.String string)
```

Parameters:
`string`

See Also:

#setPathName(String)

setSize

```
public void setSize(long l)
```

Parameters:
`l`

See Also:

#setSize(long)

com.ibm.retail.si.mgmt.swdist

Class MgmtSftPackage

```
java.lang.Object
  |
  +-com.ibm.retail.si.mgmt.swdist.MgmtSftPackage
```

All Implemented interfaces:
java.io.Serializable

public class **MgmtSftPackage**
extends java.lang.Object
implements java.io.Serializable

Holds information about a software package, which is a group of MgmtSftComponents installed as one unit

Constructor Summary

`MgmtSftPackage(java.lang.String name)` Creates a package with no components
`MgmtSftPackage(java.lang.String name,MgmtSftComponent[] name)` Creates a package with an initial set of components

Method Summary

`void`	`addComponent(MgmtSftComponent component)`
`boolean`	`equals(java.lang.Object o)` Determines equality based on name, hostPath, clientPath, and each MgmtCftComponent
`java.lang.String`	`getClientPath()` Optional installation parameter specifying a path on the client to install
`MgmtSftComponent[]`	`getComponents()`
`java.lang.String`	`getHostPath()`
`java.lang.String`	`getName()`
`long`	`getSize()` Returns the total size of all MgmtSftComponent in this package
`void`	`setClientPath(java.lang.String clientPath)`
`void`	`setHostPath(java.lang.String hostPath)`
`java.lang.String`	`toString()`

Methods inherited from : class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

MgmtSftPackage

`public` **`MgmtSftPackage`**`(java.lang.String name)`

Creates a package with no components

Parameters:

`name -`
Name of the package

MgmtSftPackage

`public` **`MgmtSftPackage`**`(java.lang.String name, MgmtSftComponent[] comps)`

Creates a package with an initial set of components

Parameters:

`name -`
Name of the package
`comps -`
Array of MgmtSftComponent

Methods

getName

`public java.lang.String` **`getName`**`()`

addComponent

`public void` **`addComponent`**`(MgmtSftComponent component)`

getComponents

`public MgmtSftComponent[]` **`getComponents`**`()`

getClientPath

`public java.lang.String` **`getClientPath`**`()`

Optional installation parameter specifying a path on the client to install

Returns:

Client path for installation

getHostPath

`public java.lang.String **getHostPath**()`

Returns:

Path on the FTP server where this package can be located

setClientPath

`public void **setClientPath**(java.lang.String clientPath)`

Parameters:

`clientPath -`
New local client path

See Also:

#getClientPath()

setHostPath

`public void **setHostPath**(java.lang.String hostPath)`

Parameters:

`hostPath -`
New FTP server path for this package

See Also:

#getHostPath()

getSize

`public long **getSize**()`

Returns the total size of all MgmtSftComponent in this package

Returns:

Sum of all MgmtSftComponent's sizes

toString

`public java.lang.String **toString**()`

equals

`public boolean **equals**(java.lang.Object o)`

Determines equality based on name, hostPath, clientPath, and each MgmtCftComponent

See Also:

(continued from last page)

Object#equals(java.lang.Object)

com.ibm.retail.si.mgmt.swdist

Interface MgmtSoftwareDistClientMBean

public interface **MgmtSoftwareDistClientMBean**

This interface represents defines the MBean interface for use on the client side of Software Distribution within the SIF environment. This intent of this interface is NOT to replicate a full software distribution entity, but rather to, in the controlled world of SIF, provide a way to signal a device, or class of devices that either an update is ready, or a new package is available for installation. In short, this is just a signalling mechanism. It remains the responsibility of the target device to understand how to unpack, and install the packages. Additionally, since this triggering mechanism is predicated on FTP, it is assumed that the target device will act as the FTP client, and that the FTP server is accessible by the client.

The `ObjectName`of this MBean includes the following attributes, in addition to the SIF attribute of `DeviceID`:
- SIFComponent=MGMT
- Id=SWDistClient

This management interface includes the following attributes. These attributes are described in more detail in the accessor methods.

- FreeSpace
- Host
- Pending
- Busy

The following operations are included in this management interface. These are described in more detail in the corresponding method documentation.
- distributePackage
- removePackage
- setFTPServerInfo
- testServer

This MBean emits a `MgmtSDProgressNotification`after each percentage interval increment has passed, and a `MgmtSDCompletionNotification`when an installation has completed.

See Also:
com.ibm.retail.si.mgmt.notifications.MgmtSDProgressNotification,

com.ibm.retail.si.mgmt.notifications.MgmtSDCompletionNotification

Field Summary

static java.lang.String	`OBJECT_NAME`

Method Summary

int	`distributePackage(MgmtSftPackage swPackage,java.lang.String swPackage,java.lang.String swPackage,long swPackage,int swPackage)` Issue the command to distribute a package.
long	`getFreeSpace(java.lang.String Device)` Get the amount of free space currently available on the target device on the client.
java.lang.String	`getHost()` Get the host information currently set for this device.

java.lang.String	getPending() Used to determine if there is a pending (deferred) distribution on this client.
boolean	isBusy() Used to determine if distribution engine is currently performing a distrbution/install.
int	removePackage(MgmtSftPackage swPackage,java.lang.String swPackage,java.lang.String swPackage,long swPackage,int swPackage) Issue the command to remove or uninstal a package.
int	setFTPServerInfo(java.lang.String Host,int Host,java.lang.String Host,java.lang.String Host) Set the FTP Server information to be used by this device.
int	testServer() Tests the FTP parameters, and the path to the server.

Fields

OBJECT_NAME

```
public static final java.lang.String OBJECT_NAME
```

Methods

getHost

```
public java.lang.String getHost()
```

Get the host information currently set for this device. This can be either a host name, or an IP address in dotted decimal notation.

Returns:

String, the currently set host information, wither as a hostname, or dotted decimal IP address.

setFTPServerInfo

```
public int setFTPServerInfo(java.lang.String Host,
                            int Port,
                            java.lang.String UserID,
                            java.lang.String Password)
```

Set the FTP Server information to be used by this device.

Parameters:

`String, -`
Host - as either a hostname, or dotted decimal IP address
`int, -`
Port - the port number to use for this FTP transaction - default should be 21
`String, -`
UserID - the FTP userid that this client should use.
`String, -`
Password - the FTP password that this client should use.

Returns:

(continued from last page)

String, the currently set host information, wither as a hostname, or dotted decimal IP address.

testServer

```
public int testServer()
```

Tests the FTP parameters, and the path to the server.

Returns:

int, return code return codes listed above.

getFreeSpace

```
public long getFreeSpace(java.lang.String Device)
```

Get the amount of free space currently available on the target device on the client. This is for use by the management tools to determine if there is enough space to carry out a distribution.

Parameters:

`String, -`
Device - a string representing the target device on the client

Returns:

long, the amount of free space in bytes.

distributePackage

```
public int distributePackage(MgmtSftPackage swPackage,
                             java.lang.String HostPath,
                             java.lang.String ClientPath,
                             long size,
                             int intervalPercentage)
```

Issue the command to distribute a package. This will cause the client to create an FTP session with the server based on the Server information previously set. The client will then pull down the package in question, and install it. IF a value other then 0 is set for hte intervalPetcentage, then a MgmtSDProgressNotification will be sent for each "percentage" interval completed. Irregradless of that, the function will send a MgmtSDCompletionNotification when the operation has either completed successfully, or failed. If the client chooses to defer the distribution, then it should indicate that on the return from this function. NOTE: this function ALWAYS returns immediately with either a good return code, deferred code, or error. All subsequent information is passed to the caller by means of notifications.

Parameters:

`MgmtSftPackage, -`
swPackage - The package to be distributed - includes one or more MgmtSftComponents.
`String, -`
HostPath - Path on the FTP server where the package exists.
`String, -`
ClientPath - The path to use on the client for storing the package - a client implementation of this interface may choose to ignore this, and place the package where it deems appropriate.
`long, -`
size - The size of this package.
`int, -`
intervalPercentage - If this is NON zero, then the implementation of this interface should break the total distribution/install process into the fragment based on the percentage, and upon completion of each fragment issue a MgmtSDProgressNotification.

Returns:

int - Immediate return code, as defined in MgmtConst

(continued from last page)

removePackage

```
public int removePackage(MgmtSftPackage swPackage,
                         java.lang.String HostPath,
                         java.lang.String ClientPath,
                         long size,
                         int intervalPercentage)
```

Issue the command to remove or uninstal a package. This will cause the client to create an FTP session with the server - if required, based on the Server information previously set. The client will then pull down the package in question, and remove it. IF a value other then 0 is set for hte intervalPetcentage, then a MgmtSDProgressNotification will be sent for each "percentage" interval completed. Irregradless of that, the function will send a MgmtSDCompletionNotification when the operation has either completed successfully, or failed. If the client chooses to defer the removal, then it should indicate that on the return from this function. NOTE: this function ALWAYS returns immediately with either a good return code, deferred code, or error. All subsequent information is passed to the caller by means of notifications.

Parameters:

`MgmtSftPackage, -`
swPackage - The package to be removed
`String, -`
HostPath - Path on the FTP server where the package exists.
`String, -`
ClientPath - The path to use on the client for storing the package - a client implementation of this interface may choose to ignore this, and place the package where it deems appropriate.
`long, -`
size - The size of this package.
`int, -`
intervalPercentage - If this is NON zero, then the implementation of this interface should break the total distribution/install process into the fragment based on the percentage, and upon completion of each fragment issue a MgmtSDProgressNotification.

Returns:

int - Immediate return code, as defined in MgmtConst

getPending

```
public java.lang.String getPending()
```

Used to determine if there is a pending (deferred) distribution on this client.

Returns:

String, If NON-NULL then this is the name of the package that is pending for distribution on the client.

isBusy

```
public boolean isBusy()
```

Used to determine if distribution engine is currently performing a distrbution/install.

Returns:

boolean

com.ibm.retail.si.mgmt.swdist

Interface MgmtSoftwareDistMasterMBean

public interface **MgmtSoftwareDistMasterMBean**

MBean that manages the signalling of software distributions and keeps a record of distribution policies and distribution records.

The `ObjectName`of this MBean includes the following attributes, in addition to the SIF attribute of `DeviceID`:
- SIFComponent=MGMT
- Id=SWDistMaster

This management interface includes the following attributes. These attributes are described in more detail in the accessor methods.

- AllDeviceDistributionRecords
- AllDeviceTypePolicies
- BusyClientWaitTime
- BusyThreshold
- FailedConnectionThreshold
- FtpAddress
- FtpPassword
- FtpPort
- MaxCurrentJobsPerDevice
- MaxJobsWaitTime
- Paused
- Started

The following operations are included in this management interface. These are described in more detail in the corresponding method documentation.
- cancelDistribution
- cancelDistributionsOnDevice
- getCompleteDistributions
- getDeviceDistributionRecord
- getDeviceDistributionRecords
- getIncompleteDistributions
- installPackageByDeviceType
- installPackageOnDevice
- pause
- removeDeviceInstallRecord
- removeDeviceTypeInstallPolicy
- removeDeviceTypeUninstallPolicy
- removeDeviceUninstallRecord
- resume
- start
- stop
- triggerDevice
- triggerDevicesByType
- uninstallPackageByDeviceType
- uninstallPackageOnDevice

This MBean emits a `MgmtSDStartedNotification`when an installation is triggered. A `MgmtSDCompletionNotification`is emitted when an installation is cancelled, when the failed connection threshold is reached, when the busy client threshold is reached, or when a general error occurs.

See Also:
com.ibm.retail.si.mgmt.swdist.MgmtSoftwareDistMaster

Field Summary

static java.lang.String	OBJECT_NAME ObjectName for this MBean

Method Summary

void	cancelDistribution(MgmtDeviceInfo device,MgmtSftPackage device,boolean device) Cancel the pending distribution matching the supplied information.
void	cancelDistributionsOnDevice(MgmtDeviceInfo device) Cancel all pending distributions for the supplied device
DeviceDistributionRecord[]	getAllDeviceDistributionRecords()
DeviceTypeSWDistPolicy[]	getAllDeviceTypePolicies()
long	getBusyClientWaitTime()
int	getBusyThreshold()
DeviceDistributionRecord[]	getCompletedDistributions(MgmtSftPackage swPackage) Returns a List of all completed distributions for the supplied package
DeviceDistributionRecord	getDeviceDistributionRecord(MgmtDeviceInfo device,MgmtSftPackage device,boolean device) Retrieves the DeviceDistributionRecord associated with the supplied information
DeviceDistributionRecord[]	getDeviceDistributionRecords(int deviceType) Returns an Array of all DeviceDistributionRecords of a particular device type
DeviceDistributionRecord[]	getDeviceDistributionRecords(int deviceType,MgmtSftPackage deviceType) Returns an Array of all DeviceDistributionRecords of a particular device type and package
DeviceDistributionRecord[]	getDeviceDistributionRecords(MgmtDeviceInfo device) Return an Array of all DeviceDistributionRecords for the supplied device
int	getFailedConnectionThreshold()
java.lang.String	getFtpAddress()
java.lang.String	getFtpPassword()
int	getFtpPort()
java.lang.String	getFtpUser()
DeviceDistributionRecord[]	getIncompletedDistributions(MgmtSftPackage swPackage) Returns an Array of all incomplete distributions for the supplied package

DeviceDistributionRecord[]	`getIncompletedDistributions(MgmtSftPackage swPackage,int swPackage)` Returns an of all incomplete distributions for the supplied package and that match the supplied return code
int	`getMaxCurrentJobsPerDevice()`
long	`getMaxJobsWaitTime()`
void	`installPackageByDeviceType(int deviceType,MgmtSftPackage deviceType,boolean deviceType,int deviceType)` Adds a DeviceDistributionRecord to install the supplied package on to devices of the specified type
void	`installPackageOnDevice(MgmtDeviceInfo device,MgmtSftPackage device,boolean device,int device)` Adds a DeviceDistributionRecord to install the supplied package on the specified device
boolean	`isPaused()`
boolean	`isStarted()` Whether or not the distribution engine is currently running
void	`pause()` Pauses the distribution engine.
void	`removeDeviceInstallRecord(MgmtDeviceInfo device,MgmtSftPackage device)` Removes the installation DeviceDistributionRecord for the supplied device and package, if such an entry exists.
void	`removeDeviceTypeInstallPolicy(int deviceType,MgmtSftPackage deviceType)` Removes the installation DeviceTypeSWDistPolicy entry for the supplied device type and package, if such an entry exists.
void	`removeDeviceTypeUninstallPolicy(int deviceType,MgmtSftPackage deviceType)` Removes the uninstallation DeviceTypeSWDistPolicy entry for the supplied device type and package, if such an entry exists.
void	`removeDeviceUninstallRecord(MgmtDeviceInfo device,MgmtSftPackage device)` Removes the uninstallation DeviceDistributionRecord for the supplied device and package, if such an entry exists.
void	`resume()` Resumes the distribution engine after being paused
void	`setBusyClientWaitTime(long l)`
void	`setBusyThreshold(int i)`
void	`setFailedConnectionThreshold(int i)`
void	`setFtpAddress(java.lang.String address)`
void	`setFtpPassword(java.lang.String pw)`

void	`setFtpPort(int port)`
void	`setFtpUser(java.lang.String user)`
void	`setMaxCurrentJobsPerDevice(int i)`
void	`setMaxJobsWaitTime(long l)`
boolean	`start()` Starts the distribution engine
boolean	`stop()` Stops the distribution engine, persisting all policies and `DeviceDistributionRecords`
void	`triggerDevice(MgmtDeviceInfo device)` Trigger a distribution request for all packages on the supplied device
void	`triggerDevicesByType(int deviceType)` Trigger all currently known devices of the supplied type
void	`uninstallPackageByDeviceType(int deviceType,MgmtSftPackage deviceType,boolean deviceType,int deviceType)` Adds a DeviceTypeSWDistPolicy entry to install the supplied package on to devices of the specified type
void	`uninstallPackageOnDevice(MgmtDeviceInfo device,MgmtSftPackage device,boolean device,int device)` Adds a DeviceTypeSWDistPolicy entry to uninstall the supplied package on the specified device

Fields

OBJECT_NAME

`public static final java.lang.String` **`OBJECT_NAME`**

ObjectName for this MBean

Methods

start

`public boolean` **`start`**`()`

Starts the distribution engine

Returns:

boolean The result of the operation. `true`if the MBean was started successfully, or `false`if there was an error starting or the MBean was already started

(continued from last page)

stop

```
public boolean stop()
```

Stops the distribution engine, persisting all policies and `DeviceDistributionRecord`s

Returns:

boolean The result of the operation. `true`if the MBean was stopped successfully, or `false`if there was an error stopping or the MBean was already stopped.

isStarted

```
public boolean isStarted()
```

Whether or not the distribution engine is currently running

Returns:

True if the engine is running, false otherwise

pause

```
public void pause()
```

Pauses the distribution engine. Newly discovered devices will still be processed and their distributions queued up to the maximum number allowed

isPaused

```
public boolean isPaused()
```

Returns:

True if the engine is currently paused, false otherwise

resume

```
public void resume()
```

Resumes the distribution engine after being paused

installPackageOnDevice

```
public void installPackageOnDevice(MgmtDeviceInfo device,
                                   MgmtSftPackage swpkg,
                                   boolean signalNow,
                                   int intervalPercentage)
                            throws MgmtException
```

Adds a DeviceDistributionRecord to install the supplied package on the specified device

Parameters:

`device` -
Destination device
`swpkg` -
Target package
`signalNow` -
Whether or not to trigger an update for all packages on that device immediately after adding the entry
`intervalPercentage` -
A non-zero value indicates the percentage fragment size for which MgmtSDProgressNotifications should be sent during the installation

(continued from last page)

Exceptions:

`MgmtException -`
Attempt to add an existing device installation policy

installPackageByDeviceType

```
public void installPackageByDeviceType(int deviceType,
                                       MgmtSftPackage swpkg,
                                       boolean signalNow,
                                       int intervalPercentage)
```

Adds a DeviceDistributionRecord to install the supplied package on to devices of the specified type

Parameters:

`deviceType -`
Destination device type
`swpkg -`
Target package
`signalNow -`
Whether or not to trigger an update for all packages on devices of that type immediately after adding the entry
`intervalPercentage -`
A non-zero value indicates the percentage fragment size for which MgmtSDProgressNotifications should be sent during the installation

See Also:

com.ibm.retail.si.mgmt.MgmtConst

uninstallPackageOnDevice

```
public void uninstallPackageOnDevice(MgmtDeviceInfo device,
                                     MgmtSftPackage swpkg,
                                     boolean signalNow,
                                     int intervalPercentage)
                                 throws MgmtException
```

Adds a DeviceTypeSWDistPolicy entry to uninstall the supplied package on the specified device

Parameters:

`device -`
Destination device
`swpkg -`
Target package
`signalNow -`
Whether or not to trigger an update for all packages on that device immediately after adding the entry
`intervalPercentage -`
A non-zero value indicates the percentage fragment size for which MgmtSDProgressNotifications should be sent during the uninstallation

uninstallPackageByDeviceType

```
public void uninstallPackageByDeviceType(int deviceType,
                                         MgmtSftPackage swpkg,
                                         boolean signalNow,
                                         int intervalPercentage)
```

Adds a DeviceTypeSWDistPolicy entry to install the supplied package on to devices of the specified type

Parameters:

`deviceType -`
Destination device type
`swpkg -`
Target package
`signalNow -`
Whether or not to trigger an update for all packages on devices of that type immediately after adding the entry

(continued from last page)

`intervalPercentage -`
A non-zero value indicates the percentage fragment size for which MgmtSDProgressNotifications should be sent during the uninstallation

See Also:

com.ibm.retail.si.mgmt.MgmtConst

triggerDevice

```
public void triggerDevice(MgmtDeviceInfo device)
```

Trigger a distribution request for all packages on the supplied device

Parameters:

`device -`
The device to trigger

triggerDevicesByType

```
public void triggerDevicesByType(int deviceType)
```

Trigger all currently known devices of the supplied type

Parameters:

`deviceType -`
int Device type to trigger

removeDeviceInstallRecord

```
public void removeDeviceInstallRecord(MgmtDeviceInfo device,
                                      MgmtSftPackage swpkg)
```

Removes the installation DeviceDistributionRecord for the supplied device and package, if such an entry exists. If no entry exists, then the method will do nothing

Parameters:

`device -`
Target device
`swpkg -`
Target package

removeDeviceUninstallRecord

```
public void removeDeviceUninstallRecord(MgmtDeviceInfo device,
                                        MgmtSftPackage swpkg)
```

Removes the uninstallation DeviceDistributionRecord for the supplied device and package, if such an entry exists. If no entry exists, then the method will do nothing

Parameters:

`device -`
Target device
`swpkg -`
Target package

removeDeviceTypeInstallPolicy

```
public void removeDeviceTypeInstallPolicy(int deviceType,
                                          MgmtSftPackage swpkg)
```

Removes the installation DeviceTypeSWDistPolicy entry for the supplied device type and package, if such an entry exists. Any DeviceDistributionRecords created will not be removed. If no entry exists, then the method will do nothing

Parameters:

(continued from last page)

`deviceType` -
Target device type
`swpkg` -
Target package

See Also:

com.ibm.retail.si.mgmt.MgmtConst

removeDeviceTypeUninstallPolicy

```
public void removeDeviceTypeUninstallPolicy(int deviceType,
                                            MgmtSftPackage swpkg)
```

Removes the uninstallation DeviceTypeSWDistPolicy entry for the supplied device type and package, if such an entry exists. Any DeviceDistributionRecords created will not be removed. If no entry exists, then the method will do nothing

Parameters:

`deviceType` -
Target device type
`swpkg` -
Target package

See Also:

com.ibm.retail.si.mgmt.MgmtConst

cancelDistribution

```
public void cancelDistribution(MgmtDeviceInfo device,
                               MgmtSftPackage swPackage,
                               boolean isInstall)
```

Cancel the pending distribution matching the supplied information. The record matching the distribution will be persisted with a cancelled return code and a MgmtSDCompletion notification sent.

Parameters:

`device` -
Client device
`swPackage` -
Target package
`isInstall` -
True if an installation, false otherwise

cancelDistributionsOnDevice

```
public void cancelDistributionsOnDevice(MgmtDeviceInfo device)
```

Cancel all pending distributions for the supplied device

Parameters:

`device` -
Client device

See Also:

#cancelDistribution(MgmtDeviceInfo, MgmtSftPackage, boolean)

getAllDeviceTypePolicies

```
public DeviceTypeSWDistPolicy[] getAllDeviceTypePolicies()
```

Returns:

(continued from last page)

Array of all DeviceTypeSWDistPolicy entries

getAllDeviceDistributionRecords

```
public DeviceDistributionRecord[] getAllDeviceDistributionRecords()
```

Returns:

Array of all DeviceDistributionRecords for all devices

getDeviceDistributionRecords

```
public DeviceDistributionRecord[] getDeviceDistributionRecords(MgmtDeviceInfo device)
```

Return an Array of all DeviceDistributionRecords for the supplied device

Parameters:

`device -`
MgmtDeviceInfo Target device

Returns:

DeviceDistributionRecord[], or empty array if no records exist

getDeviceDistributionRecord

```
public DeviceDistributionRecord getDeviceDistributionRecord(MgmtDeviceInfo device,
                                                            MgmtSftPackage swPackage,
                                                            boolean isInstall)
```

Retrieves the DeviceDistributionRecord associated with the supplied information

Parameters:

`device -`
MgmtDeviceInfo target device
`swPackage -`
MgmtSftPackage target package
`isInstall -`
boolean True for an installation, false for an uninstall

Returns:

DeviceDistributionRecord matching the supplied information, or null if it doesn't exist

getDeviceDistributionRecords

```
public DeviceDistributionRecord[] getDeviceDistributionRecords(int deviceType)
```

Returns an Array of all DeviceDistributionRecords of a particular device type

Parameters:

`deviceType -`
Target device type

Returns:

DeviceDistributionRecord[] matching the supplied device type, or an empty Array if none exist

(continued from last page)

getDeviceDistributionRecords

```
public DeviceDistributionRecord[] getDeviceDistributionRecords(int deviceType,
                                                               MgmtSftPackage
swPackage)
```

Returns an Array of all DeviceDistributionRecords of a particular device type and package

Parameters:

`deviceType -`
Target device type
`swPackage -`
Target package

Returns:

DeviceDistributionRecord[] matching the supplied device type and package, or an empty Array if none exist

getCompletedDistributions

```
public DeviceDistributionRecord[] getCompletedDistributions(MgmtSftPackage swPackage)
```

Returns a List of all completed distributions for the supplied package

Parameters:

`swPackage -`
Target package

Returns:

DeviceDistributionRecord[] of all completed distributions for the supplied package, or an empty Array if none exist

getIncompletedDistributions

```
public DeviceDistributionRecord[] getIncompletedDistributions(MgmtSftPackage
swPackage)
```

Returns an Array of all incomplete distributions for the supplied package

Parameters:

`swPackage -`
Target package

Returns:

DeviceDistributionRecord[] of all incomplete distributions for the supplied package, or an empty Array if none exist

getIncompletedDistributions

```
public DeviceDistributionRecord[] getIncompletedDistributions(MgmtSftPackage
swPackage,
                                                              int rc)
```

Returns an of all incomplete distributions for the supplied package and that match the supplied return code

Parameters:

`swPackage -`
Target package
`rc -`
Return code to match

Returns:

DeviceDistributionRecord[] of all incomplete distributions for the supplied package and return code, or an empty Array if none exist

getFtpAddress

```
public java.lang.String getFtpAddress()
```

Returns:

Hostname or IP Address to passed to clients for retrieving files

getFtpPort

```
public int getFtpPort()
```

Returns:

FTP Port number passed to clients for retrieving files

setFtpAddress

```
public void setFtpAddress(java.lang.String address)
                   throws javax.management.InvalidAttributeValueException
```

Parameters:

`address -`
String IP Address or hostname to be passed to clients for retrieving files

setFtpPort

```
public void setFtpPort(int port)
                throws javax.management.InvalidAttributeValueException
```

Parameters:

`port -`
FTP Port number passed to clients for retrieving files

getFtpPassword

```
public java.lang.String getFtpPassword()
```

Returns:

FTP password used by clients to retrieve files

getFtpUser

```
public java.lang.String getFtpUser()
```

Returns:

FTP username used by clients to retrieve files

setFtpPassword

```
public void setFtpPassword(java.lang.String pw)
                    throws javax.management.InvalidAttributeValueException
```

Parameters:

pw -
String FTP password used by clients to retrieve files

setFtpUser

```
public void setFtpUser(java.lang.String user)
                throws javax.management.InvalidAttributeValueException
```

Parameters:

String -
FTP username used by clients to retrieve files

getBusyClientWaitTime

```
public long getBusyClientWaitTime()
```

Returns:

long The amount of time to wait, in ms, before retrying a software distribution on a busy client

getBusyThreshold

```
public int getBusyThreshold()
```

Returns:

int The number of consecutive tries performed for a distribution on a busy client before ending the distribution in error

getFailedConnectionThreshold

```
public int getFailedConnectionThreshold()
```

Returns:

int The number of failed connection attempts before a distribution is ended in error

getMaxCurrentJobsPerDevice

```
public int getMaxCurrentJobsPerDevice()
```

Returns:

int The maximum number of distributions per device that are at eny time queued to be signalled

getMaxJobsWaitTime

```
public long getMaxJobsWaitTime()
```

Returns:

long The amount of time to wait, in ms, before attempting to enqueue software distributions for a device that has reached the maxCurrentJobsPerDevice threshold

setBusyClientWaitTime

```
public void setBusyClientWaitTime(long l)
                          throws javax.management.InvalidAttributeValueException
```

Parameters:

`l`

See Also:

#getBusyClientWaitTime()

setBusyThreshold

```
public void setBusyThreshold(int i)
                     throws javax.management.InvalidAttributeValueException
```

Parameters:

`i`

See Also:

#getBusyThreshold()

setFailedConnectionThreshold

```
public void setFailedConnectionThreshold(int i)
                                  throws
javax.management.InvalidAttributeValueException
```

Parameters:

`i`

See Also:

#getFailedConnectionThreshold()

setMaxCurrentJobsPerDevice

```
public void setMaxCurrentJobsPerDevice(int i)
                                throws javax.management.InvalidAttributeValueException
```

Parameters:

(continued from last page)

`i`

See Also:

#getMaxCurrentJobsPerDevice()

setMaxJobsWaitTime

```
public void setMaxJobsWaitTime(long l)
                        throws javax.management.InvalidAttributeValueException
```

Parameters:

`l`

See Also:

#getMaxJobsWaitTime()

Package
com.ibm.retail.si.util

com.ibm.retail.si.util

Interface AEFConst

public interface **AEFConst**

AEFConst is an interface which defines constants for the Store Integrator.

Field Summary

static int	ACCOUNT_NOT_YET_AVAILABLE
static int	ACCOUNT_NUMBER_ERROR
static int	ACCOUNT_NUMBER_REQUIRED
static int	ACCOUNT_PAID_IN_FULL
static int	ACCOUNT_TENDER_LIMIT
static java.lang.String	AEFPROPS_VAR System property for AEFProps properties filename
static int	AGE_RESTRICTED_ITEM
static int	AMOUNT_REQUIRED
static int	ANOTHER_OPERATOR_SIGNED_ON
static int	API_VALID_FOR_TSS_ONLY
static int	APPLICATION_ALREADY_ACTIVE
static int	APPLICATION_LIMIT_EXCEEDED
static int	APPLICATION_NOT_ACTIVE
static int	APPLICATION_NOT_IN_PROPER_STATE Error Codes
static int	APPLICATION_NOT_IN_PROPER_SUBSTATE
static int	ARGUMENT_REQUIRED
static int	AUTHORIZATION_CODE_REQUIRED
static int	AUTHORIZATION_NUMBER_MISMATCH

static int	AUTHORIZATION_REQUIRED
static java.lang.String	AUTOMATION_PROPS_BUNDLE
static int	BUFFER_FULL
static int	CALL_FOR_AUTHORIZATION
static int	CANCELLED_BY_OPERATOR
static int	CARD_EXPIRED
static int	CARD_ID_REQUIRED
static int	CARD_TYPE_UNKNOWN
static java.lang.String	CASH_COUNT
static java.lang.String	CASH_DOC
static int	CASH_DRAWER_ERROR
static java.lang.String	CASH_SPECIAL
static int	CASHBACK_ON_FINAL_TENDER_ONLY
static int	CHANGE_AMOUNT_LIMIT
static java.lang.String	CHARGE_PLAN_A
static java.lang.String	CHARGE_PLAN_B
static java.lang.String	CHARGE_PLAN_C
static java.lang.String	CHARGE_PLAN_D
static int	CHARGE_PLAN_NOT_ALLOWED
static int	CHECK_CONFIGURATION
static int	CHECK_ITEM_PRICE
static int	CHECK_OVERRIDE_PRICE
static int	CHECK_WEIGHT_PERSONALIZATION
static int	CLASS_NOT_FOUND

static java.lang.String	CLASSNAME_BUNDLE
static int	CLIENT_SESSION_MISMATCH
static int	CLOSE_DRAWER
static int	CLOSING_ACCOUNTING_PERIOD
static java.lang.String	COD
static int	COIN_DISPENSER_ERROR
static int	COMMUNICATION_ERROR
static java.lang.String	CONFIG_BUNDLE
static int	CONFIG_ERROR
static int	COUPON_EXPIRED
static int	COUPON_VALUE_EXCEEDS_ITEM_VALUE
static int	COVER_OPEN
static int	CREDIT_CARD_TYPE_REQUIRED
static int	CREDIT_NOT_AVAILABLE
static int	CUSTOMER_ALTERED_AMOUNT_AT_PINPAD
static int	CUSTOMER_CANCELLED_TENDER_AT_PINPAD
static int	CUSTOMER_CHOSE_ANOTHER_TENDER_AT_PINPAD
static int	CUSTOMER_MUST_SIGN_DOCUMENT
static int	DAILY_LOYALTY_CARD_USAGE_LIMIT
static int	DATA_OUT_OF_RANGE
static int	DEAL_PRICE_AND_DEAL_QUANTITY_REQUIRED
static int	DEPARTMENT_RETURN_LIMIT
static java.lang.String	DEPARTMENT_TOTALS
static java.lang.String	DEPT

static int	DEVICE_HANDLER_FAILURE
static int	DEVICE_HOOK_ERROR
static int	DEVICE_IS_LOADING
static int	DEVICE_MANAGER_ERROR
static int	DEVICE_NOT_REDIRECTED
static int	DEVICE_OFFLINE
static int	DISCOUNT_LIMIT
static int	DISCOUNT_NOT_ALLOWED_FOR_ITEM
static int	DISCOUNT_PERCENTAGE_TOO_LARGE
static int	DISCOUNT_PERCENTAGE_TOO_SMALL
static java.lang.String	DISPLAY
static int	DISPLAY_ERROR
static java.lang.String	EBT_BALANCE_INQUIRY
static int	EBT_NOT_ALLOWED
static boolean	ENABLE_RMI_MULTIHOME_FIX Constants for SocketFactory
static int	ENABLEMENT_ERROR
static int	ERROR_ACCESSING_FACTORY
static int	ERROR_ACCESSING_GENERAL_AGENT
static java.lang.String	ERROR_BUNDLE
static int	ERROR_CREATING_DEVICE_SERVER
static int	ERROR_CREATING_GENERAL_AGENT
static int	ERROR_CREATING_SESSION
static int	ERROR_HANDLER_EXCEPTION_DETECTED
static int	ERROR_HANDLER_FATAL_ERROR

static int	ERROR_HANDLER_OTHER_ERROR
static int	ERROR_HANDLER_OVERRIDE_ERROR
static int	ERROR_STARTING_POS_APPLICATION
static int	ERROR_STOPPING_POS_APPLICATION
static int	EVENT_LISTENER_MISMATCH
static int	EXPIRATION_DATE_REQUIRED
static int	EXTENDED_PRICE_PROHIBITED
static int	EXTENDED_PRICE_TOO_LARGE
static int	EXTERNAL_TENDER_AUTHORIZATION_SUSPENDED
static int	FACTORY_ERROR
static int	FACTORY_NO_SESSIONS_AVAILABLE
static int	FACTORY_SESSION_NOT_READY
static java.lang.String	FCODE_BUNDLE
static int	FCODE_NOT_FOUND
static int	FILE_ACCESS_FAILED
static int	FILE_IO_ERROR
static int	FILE_NOT_FOUND
static int	FOODSTAMP_TENDER_IN_EXCESS_OF_FOODSTAMP_BALANCE
static int	FULLSCREEN_NOT_SUPPORTED
static int	HARDWARE_PROBLEM
static int	HOME_STORE_MUST_REDEEM_LOYALTY_POINTS
static int	HOST_REQUEST_PENDING
static java.lang.String	I18NTEXT_BUNDLE
static int	ID_PROHIBITED

static int	ID_REQUIRED
static int	INPUT_CANCELLED
static int	INPUT_NOT_ALLOWED
static int	INVALID_ACCOUNT_NUMBER
static int	INVALID_ACTION_REQUEST
static int	INVALID_AMOUNT
static int	INVALID_APPLICATION_DATA
static int	INVALID_ARGUMENT
static int	INVALID_AUTHORIZATION_CODE
static int	INVALID_BATCH_NUMBER
static int	INVALID_CARD_DATA
static int	INVALID_CARD_ID
static int	INVALID_CHARGE_PLAN
static int	INVALID_CURRENCY_AMOUNT
static int	INVALID_DATE
static int	INVALID_DEAL_PRICE
static int	INVALID_DEAL_QUANTITY
static int	INVALID_DEPARTMENT
static int	INVALID_DEPARTMENT_TOTALS_LIST_NUMBER
static int	INVALID_DISCOUNT_CODE
static int	INVALID_DISPATCH_QUEUE
static int	INVALID_EXPIRY_DATE
static int	INVALID_FEE_AMOUNT
static int	INVALID_ID

static int	INVALID_INITIALIZE_STATE
static int	INVALID_ITEM_CODE
static int	INVALID_ITEM_IDENTIFIER_TYPE
static int	INVALID_KEY_SEQUENCE
static int	INVALID_KEYSEQUENCE_EXPRESSION
static int	INVALID_LISTENER_TYPE
static int	INVALID_LOGGER_LEVEL
static int	INVALID_LOYALTY_NUMBER
static int	INVALID_MANAGER_OVERRIDE_NUMBER
static int	INVALID_PASSWORD
static int	INVALID_PRICE
static int	INVALID_PROPERTY_VALUE
static int	INVALID_QUANTITY
static int	INVALID_REASON
static int	INVALID_TARE
static int	INVALID_TAX_CODE
static int	INVALID_TENDER_AMOUNT
static int	INVALID_TENDER_TYPE
static int	INVALID_TERMINAL_NUMBER
static int	INVALID_TIMEOUT_VALUE
static int	INVALID_TRANSACTION_TYPE
static int	INVALID_VOLUME
static int	INVALID_VOUCHER_NUMBER
static int	INVALID_WEIGHT

static int	IO_E_FAILURE
static int	IO_E_ILLEGAL
static int	IO_E_INTERNAL Device Accessor error codes.
static int	ITEM_CANNOT_BE_WEIGHED
static int	ITEM_CLASS_PROHIBITED
static int	ITEM_CLASS_REQUIRED
static int	ITEM_CODE_REQUIRED
static int	ITEM_DEPARTMENT_PROHIBITED
static int	ITEM_DEPARTMENT_REQUIRED
static int	ITEM_DISCOUNTS_NOT_ALLOWED
static int	ITEM_IS_ON_THE_SCALE
static int	ITEM_IS_TIME_RESTRICTED
static int	ITEM_LIMIT
static java.lang.String	ITEM_MOVEMENT
static int	ITEM_NOT_FOR_SALE
static int	ITEM_NOT_FOUND
static int	ITEM_NOT_ON_SCALE_CORRECTLY
static int	ITEM_NOT_RETURNABLE
static int	ITEM_PRICE_OR_QUANTITY_REQUIRED
static int	ITEM_PRICE_REQUIRED
static int	ITEM_QUANTITY_PROHIBITED
static int	ITEM_QUANTITY_REQUIRED
static java.lang.String	ITEM_RECORD_PRICE_CHANGE
static java.lang.String	ITEM_RETURN

static int	ITEM_STOCK_PROHIBITED
static int	ITEM_STOCK_REQUIRED
static int	ITEM_WEIGHT_PROHIBITED
static int	ITEM_WEIGHT_REQUIRED
static int	JAVA_INVOCATION_FAILED
static int	JAVA_POS_EXCEPTION
static int	JOURNAL_ERROR
static int	JPOS_NOEXIST
static java.lang.String	KEY_SEQUENCE_BUNDLE Bundle base file names.
static int	KEYBOARD_ERROR
static java.lang.String	KEYED_LABEL
static int	KEYLOCK_ERROR
static int	KEYSEQUENCE_NOT_FOUND
static java.lang.String	LAYAWAY
static java.lang.String	LAYAWAY_CANCEL
static java.lang.String	LAYAWAY_PAYMENT
static int	LOAD_ERROR
static java.lang.String	LOAN
static java.lang.String	LOGON_BUNDLE
static int	LOYALTY_CARD_EXPIRED
static int	LOYALTY_COUPON_NOT_APPLICABLE_TO_CUSTOMER_STATUS_LEVEL
static int	LOYALTY_NUMBER_REQUIRED
static int	LOYALTY_POINTS_LIMIT
static int	MANAGER_KEY_NOT_REMOVED

static int	MANAGER_KEY_REQUIRED
static int	MANAGER_OVERRIDE_REQUIRED
static int	MAX_NUMBER_OF_ITEMS
static int	MAX_NUMBER_OF_PRICE_CHANGES
static int	MICR_DATA_NOT_ALLOWED
static int	MICR_ERROR
static int	MINIMUM_SALE_NOT_SATISFIED
static int	MISSING_CONFIG_VALUE
static int	MISSING_DESCRIPTOR
static int	MISSING_EXCHANGE_RATE
static java.lang.String	MODIFY_DEPT_PRESETS
static int	MONITOR_ALREADY_ACTIVE
static int	MORE_LOYALTY_POINTS_NEEDED
static int	MSR_ERROR
static int	MSR_HOOK_SWIPE_ERROR
static int	MSR_NOT_ENABLED
static int	MSR_SET_TO_DECODE
static int	MSR_TRACK_DATA_NOT_ALLOWED
static int	MSR_TRACK_DATA_REQUIRED
static int	NEGATIVE_TRANSACTION_BALANCE
static int	NETWORK_FAILURE
static int	NEW_PASSWORD_PROHIBITED
static int	NO_DEFAULT_ID_CONFIGURED
static int	NO_DEFAULT_MANAGER_OVERRIDE_PASSWORD

static int	NO_ERROR_HELPER_CONFIGURED
static int	NO_ITEM_MATCH_FOR_COUPON
static int	NO_LISTENER_SUPPORT
static java.lang.String	NO_SALE Transaction types.
static int	NO_SESSION_FACTORIES_FOUND
static int	NO_SUCH_LOGGER
static int	NO_SUCH_PROPERTY
static int	NO_SUCH_SUSPENDED_TRANSACTION
static int	NON_WIC_ITEM
static int	NON_WIC_TENDER
static int	NONE Extended Error Codes
static int	NONE_AVAILABLE
static int	NONNUMERIC_FUNCTION_CODE
static int	NOT_ALLOWED_OFFLINE
static int	NOT_ALLOWED_REENTRY
static int	NOT_ALLOWED_TRAINING
static int	NOT_APPROVED_BY_PAYMENT_SYSTEM
static int	NOT_AUTHORIZED
static int	NOT_IMPLEMENTED
static int	NUMBER_OF_COUPONS_LIMIT
static int	NUMBER_OF_TENDERS_LIMIT
static java.lang.String	OFFLINE_TRANSACTION_REENTRY
static java.lang.String	OPEN_TRANS_REPORT

static int	OPERATION_TIMEOUT
static int	OPERATOR_ALREADY_ACTIVE
static java.lang.String	OPERATOR_TRAINING
static int	OUT_OF_MEMORY
static int	PAPER_LOW
static int	PASSWORD_EXPIRED
static int	PASSWORD_REQUIRED
static int	PAYMENT_ALREADY_ENTERED
static int	PAYMENT_SYSTEM_ERROR
static int	PAYMENT_SYSTEM_OFFLINE
static java.lang.String	PERF_TRACE Performance logger name
static java.lang.String	PICKUP
static int	PICKUP_NEEDED
static int	PIN_COULD_NOT_BE_OBTAINED
static int	PIN_PAD_ERROR
static int	PIN_PAD_PROHIBITED
static int	PIN_PAD_REQUIRED
static int	PINPAD_NOT_AVAILABLE
static int	POS_APP_FAILURE
static int	PREVIOUS_ITEM_ON_SCALE
static int	PRICE_AND_DEAL_PRICE_MUTUALLY_EXCLUSIVE
static java.lang.String	PRICE_CHANGE_TRANS
static int	PRICE_PER_KILOGRAM_REQUIRED
static int	PRICE_PER_POUND_REQUIRED

static java.lang.String	PRICE_VERIFY_CHANGE
static int	PRINTER_ERROR
static int	PROCEDURE_NOT_ALLOWED
static int	PROPERTY_NOT_SUPPORTED
static java.lang.String	QUERY_EXCHANGE_RATE
static int	REASON_CODE_PROHIBITED
static int	REASON_CODE_REQUIRED
static java.lang.String	REGULAR_CASH
static java.lang.String	REGULAR_SALE
static int	REMOTE_EXCEPTION
static int	REMOTE_SESSION_SERVER_ERROR
static int	REPLACE_ITEM
static java.lang.String	REPRINT_PARTIAL_RECEIPT
static java.lang.String	REPRINT_RECEIPT
static int	RETRIEVE_TRANSACTION_WHERE_SUSPENDED
static java.lang.String	RETURN Transaction modifier types.
static int	RETURN_COUPON_BEFORE_ITEM_VOID
static int	REWEIGH_ITEM
static int	RISK_1
static int	RISK_2
static int	RISK_3
static int	RISK_4
static int	RMI_BIND_EXCEPTION
static int	RMI_NAMING_FAILURE

static int	RMI_NAMING_REGISTRY_RETURNED_NULL
static int	RMI_REGISTRY_FAILURE
static int	RMI_REMOTE_EXCEPTION
static int	RMI_URL_EXCEPTION
static int	SAME_ERROR_TWICE
static int	SAME_OPERATOR_MUST_RETRIEVE_TRANSACTION
static int	SCALE_ERROR
static java.lang.String	SCAN_LABEL Item identifier types.
static int	SCANNED_DATA_NOT_ALLOWED
static java.lang.String	SEND
static int	SERVER_NO_SESSION_AVAILABLE
static int	SERVICER_CLOSED
static java.lang.String	SESSION_BUNDLE
static int	SESSION_FACTORY_REMOTE_EXCEPTION
static int	SESSION_NOT_ACTIVE
static int	SESSION_NOT_AVAILABLE
static int	SESSION_READY_WAIT_TIMEOUT
static java.lang.String	SESSIONROLE_BUNDLE
static java.lang.String	SET_TRANSACTION_NUMBER
static int	SIGNATURE_REQUIRED
static java.lang.String	SIPROPS_VAR System property for SIProps properties filename
static java.lang.String	SKU
static int	STAND_IN_AMOUNT_LIMIT

static int	STAND_IN_COUNT_LIMIT
static java.lang.String	STATE_BUNDLE
static int	STATE_NOT_FOUND
static java.lang.String	SUSPENDED_TRANS_REPORT
static int	SUSPENDED_TRANSACTION_LIMIT
static int	SYSTEM_BUSY
static int	TARE_WEIGHT_TOO_LARGE
static int	TAX_CODE_REQUIRED
static int	TAX_EXEMPTION_NOT_ALLOWED_FOR_ITEM
static int	TAX_OPTION_MISMATCH
static int	TAX_TABLE_NOT_FOUND
static int	TENDER_AMOUNT_LIMIT
static java.lang.String	TENDER_CASHING
static java.lang.String	TENDER_COUNT
static java.lang.String	TENDER_EXCHANGE
static int	TENDER_EXPIRED
static java.lang.String	TENDER_FEE_REFUND
static int	TENDER_FLOOR_LIMIT
static java.lang.String	TENDER_LISTING
static java.lang.String	TENDER_MAP_BUNDLE
static int	TENDER_NOT_ACCEPTED
static int	TENDER_NOT_AUTHORIZED
static java.lang.String	TENDER_REMOVAL
static int	TENDER_TYPE_NOT_ALLOWED

static int	TERMINAL_DISABLED
static java.lang.String	TERMINAL_MONITOR
static java.lang.String	TERMINAL_PROGRAM_LOAD
static java.lang.String	TERMINAL_TRANSFER
static int	TILL_EXCHANGE_NEEDED
static int	TIMEOUT_WAITING_FOR_AVAILABLE_SESSION
static int	TONE_ERROR
static int	TOO_LONG_IN_STAND_IN
static int	TOO_MANY_ERRORS
static java.lang.String	TOTAL_READOUT_RESET
static int	TRACK_DATA_PROHIBITED
static int	TRACK_DATA_REQUIRED
static int	TRANSACTION_ALREADY_IN_PROGRESS
static int	TRANSACTION_ALREADY_RETRIEVED
static int	TRANSACTION_CANNOT_BE_VOIDED
static int	TRANSACTION_IN_PROGRESS
static int	TRANSACTION_LIMIT
static int	TRANSACTION_NOT_ACTIVE
static int	TRANSACTION_TOTAL_TOO_LARGE
static int	TRANSACTION_TYPE_REQUIRED
static int	TSS_ERROR
static int	UNABLE_TO_LOAD_PROPERTIES_FILE
static int	UNABLE_TO_RETRIEVE_TRANSACTION
static int	UNCHECKED_EXCEPTION

static int	UNDEFINED_TENDER_VARIETY
static int	UNEXPECTED_STATE
static int	UNKNOWN_MICR_FORMAT
static int	UNKNOWN_SERVICER
static int	UNRECOGNIZED_PRINT_CHARACTERS
static int	UNSUPPORTED_CAPABILITY
static int	UNSUPPORTED_ERROR_HANDLING_MODE
static int	UNSUPPORTED_OPERATION
static int	UNSUPPORTED_TRANSACTION_TYPE
static int	USER_DEFINED
static java.lang.String	VALUE_CARD_BALANCE_INQUIRY
static int	VALUE_CARDS_MUST_BE_VOIDED
static int	VALUE_EXCEEDS_CONNECTION_TIMEOUT
static java.lang.String	VAT
static java.lang.String	VELOCITY
static int	VERIFICATION_TIMEOUT
static int	VERIFY_SIGNATURE
static int	VOID_MUST_MATCH_PREVIOUS
static java.lang.String	VOID_PREVIOUS_BY_LINE_ITEM
static java.lang.String	VOID_PREVIOUS_TRANSACTION
static int	VOLUME_PROHIBITED
static int	VOLUME_REQUIRED
static int	VOUCHER_NOT_YET_VALID
static int	VOUCHER_NUMBER_REQUIRED

static int	WAIT_INTERRUPTED
static int	WEIGH_ITEM
static int	WEIGHT_AND_QUANTITY_MUTUALLY_EXCLUSIVE
static java.lang.String	WIC

Fields

KEY_SEQUENCE_BUNDLE

public static final java.lang.String **KEY_SEQUENCE_BUNDLE**

Bundle base file names.

FCODE_BUNDLE

public static final java.lang.String **FCODE_BUNDLE**

CONFIG_BUNDLE

public static final java.lang.String **CONFIG_BUNDLE**

CLASSNAME_BUNDLE

public static final java.lang.String **CLASSNAME_BUNDLE**

STATE_BUNDLE

public static final java.lang.String **STATE_BUNDLE**

LOGON_BUNDLE

public static final java.lang.String **LOGON_BUNDLE**

AUTOMATION_PROPS_BUNDLE

public static final java.lang.String **AUTOMATION_PROPS_BUNDLE**

TENDER_MAP_BUNDLE

public static final java.lang.String **TENDER_MAP_BUNDLE**

ERROR_BUNDLE

`public static final java.lang.String` **`ERROR_BUNDLE`**

I18NTEXT_BUNDLE

`public static final java.lang.String` **`I18NTEXT_BUNDLE`**

SESSION_BUNDLE

`public static final java.lang.String` **`SESSION_BUNDLE`**

SESSIONROLE_BUNDLE

`public static final java.lang.String` **`SESSIONROLE_BUNDLE`**

AEFPROPS_VAR

`public static final java.lang.String` **`AEFPROPS_VAR`**

System property for AEFProps properties filename

SIPROPS_VAR

`public static final java.lang.String` **`SIPROPS_VAR`**

System property for SIProps properties filename

PERF_TRACE

`public static final java.lang.String` **`PERF_TRACE`**

Performance logger name

SCAN_LABEL

`public static final java.lang.String` **`SCAN_LABEL`**

Item identifier types.

SKU

`public static final java.lang.String` **`SKU`**

VELOCITY

`public static final java.lang.String` **`VELOCITY`**

(continued from last page)

KEYED_LABEL

public static final java.lang.String **KEYED_LABEL**

DEPT

public static final java.lang.String **DEPT**

NO_SALE

public static final java.lang.String **NO_SALE**

Transaction types.

REGULAR_SALE

public static final java.lang.String **REGULAR_SALE**

REGULAR_CASH

public static final java.lang.String **REGULAR_CASH**

CASH_SPECIAL

public static final java.lang.String **CASH_SPECIAL**

CASH_DOC

public static final java.lang.String **CASH_DOC**

COD

public static final java.lang.String **COD**

LAYAWAY

public static final java.lang.String **LAYAWAY**

CHARGE_PLAN_A

public static final java.lang.String **CHARGE_PLAN_A**

CHARGE_PLAN_B

public static final java.lang.String **CHARGE_PLAN_B**

CHARGE_PLAN_C

public static final java.lang.String **CHARGE_PLAN_C**

CHARGE_PLAN_D

public static final java.lang.String **CHARGE_PLAN_D**

LOAN

public static final java.lang.String **LOAN**

PICKUP

public static final java.lang.String **PICKUP**

CASH_COUNT

public static final java.lang.String **CASH_COUNT**

TENDER_COUNT

public static final java.lang.String **TENDER_COUNT**

TOTAL_READOUT_RESET

public static final java.lang.String **TOTAL_READOUT_RESET**

DEPARTMENT_TOTALS

public static final java.lang.String **DEPARTMENT_TOTALS**

ITEM_MOVEMENT

public static final java.lang.String **ITEM_MOVEMENT**

ITEM_RECORD_PRICE_CHANGE

public static final java.lang.String **ITEM_RECORD_PRICE_CHANGE**

(continued from last page)

SET_TRANSACTION_NUMBER

public static final java.lang.String **SET_TRANSACTION_NUMBER**

OFFLINE_TRANSACTION_REENTRY

public static final java.lang.String **OFFLINE_TRANSACTION_REENTRY**

OPERATOR_TRAINING

public static final java.lang.String **OPERATOR_TRAINING**

VOID_PREVIOUS_TRANSACTION

public static final java.lang.String **VOID_PREVIOUS_TRANSACTION**

VOID_PREVIOUS_BY_LINE_ITEM

public static final java.lang.String **VOID_PREVIOUS_BY_LINE_ITEM**

TENDER_LISTING

public static final java.lang.String **TENDER_LISTING**

TENDER_REMOVAL

public static final java.lang.String **TENDER_REMOVAL**

TERMINAL_MONITOR

public static final java.lang.String **TERMINAL_MONITOR**

QUERY_EXCHANGE_RATE

public static final java.lang.String **QUERY_EXCHANGE_RATE**

TENDER_CASHING

public static final java.lang.String **TENDER_CASHING**

TENDER_EXCHANGE

public static final java.lang.String **TENDER_EXCHANGE**

PRICE_VERIFY_CHANGE

public static final java.lang.String **PRICE_VERIFY_CHANGE**

TERMINAL_TRANSFER

public static final java.lang.String **TERMINAL_TRANSFER**

TERMINAL_PROGRAM_LOAD

public static final java.lang.String **TERMINAL_PROGRAM_LOAD**

ITEM_RETURN

public static final java.lang.String **ITEM_RETURN**

WIC

public static final java.lang.String **WIC**

REPRINT_PARTIAL_RECEIPT

public static final java.lang.String **REPRINT_PARTIAL_RECEIPT**

REPRINT_RECEIPT

public static final java.lang.String **REPRINT_RECEIPT**

EBT_BALANCE_INQUIRY

public static final java.lang.String **EBT_BALANCE_INQUIRY**

VALUE_CARD_BALANCE_INQUIRY

public static final java.lang.String **VALUE_CARD_BALANCE_INQUIRY**

LAYAWAY_PAYMENT

public static final java.lang.String **LAYAWAY_PAYMENT**

(continued from last page)

LAYAWAY_CANCEL

public static final java.lang.String **LAYAWAY_CANCEL**

PRICE_CHANGE_TRANS

public static final java.lang.String **PRICE_CHANGE_TRANS**

TENDER_FEE_REFUND

public static final java.lang.String **TENDER_FEE_REFUND**

SUSPENDED_TRANS_REPORT

public static final java.lang.String **SUSPENDED_TRANS_REPORT**

MODIFY_DEPT_PRESETS

public static final java.lang.String **MODIFY_DEPT_PRESETS**

OPEN_TRANS_REPORT

public static final java.lang.String **OPEN_TRANS_REPORT**

RETURN

public static final java.lang.String **RETURN**

Transaction modifier types.

SEND

public static final java.lang.String **SEND**

VAT

public static final java.lang.String **VAT**

DISPLAY

public static final java.lang.String **DISPLAY**

ENABLE_RMI_MULTIHOME_FIX

public static final boolean **ENABLE_RMI_MULTIHOME_FIX**

Constants for SocketFactory

APPLICATION_NOT_IN_PROPER_STATE

public static final int **APPLICATION_NOT_IN_PROPER_STATE**

Error Codes

COUPON_EXPIRED

public static final int **COUPON_EXPIRED**

NO_SUCH_PROPERTY

public static final int **NO_SUCH_PROPERTY**

LOYALTY_CARD_EXPIRED

public static final int **LOYALTY_CARD_EXPIRED**

OPERATION_TIMEOUT

public static final int **OPERATION_TIMEOUT**

NETWORK_FAILURE

public static final int **NETWORK_FAILURE**

UNSUPPORTED_TRANSACTION_TYPE

public static final int **UNSUPPORTED_TRANSACTION_TYPE**

INVALID_ARGUMENT

public static final int **INVALID_ARGUMENT**

SIGNATURE_REQUIRED

public static final int **SIGNATURE_REQUIRED**

MANAGER_OVERRIDE_REQUIRED

public static final int **MANAGER_OVERRIDE_REQUIRED**

(continued from last page)

ITEM_NOT_FOUND

public static final int **ITEM_NOT_FOUND**

FILE_NOT_FOUND

public static final int **FILE_NOT_FOUND**

INVALID_ACTION_REQUEST

public static final int **INVALID_ACTION_REQUEST**

INPUT_NOT_ALLOWED

public static final int **INPUT_NOT_ALLOWED**

SYSTEM_BUSY

public static final int **SYSTEM_BUSY**

MISSING_CONFIG_VALUE

public static final int **MISSING_CONFIG_VALUE**

WAIT_INTERRUPTED

public static final int **WAIT_INTERRUPTED**

STATE_NOT_FOUND

public static final int **STATE_NOT_FOUND**

ANOTHER_OPERATOR_SIGNED_ON

public static final int **ANOTHER_OPERATOR_SIGNED_ON**

AGE_RESTRICTED_ITEM

public static final int **AGE_RESTRICTED_ITEM**

ITEM_IS_TIME_RESTRICTED

public static final int **ITEM_IS_TIME_RESTRICTED**

PROPERTY_NOT_SUPPORTED

public static final int **PROPERTY_NOT_SUPPORTED**

INVALID_PROPERTY_VALUE

public static final int **INVALID_PROPERTY_VALUE**

INVALID_LISTENER_TYPE

public static final int **INVALID_LISTENER_TYPE**

NO_LISTENER_SUPPORT

public static final int **NO_LISTENER_SUPPORT**

EVENT_LISTENER_MISMATCH

public static final int **EVENT_LISTENER_MISMATCH**

UNEXPECTED_STATE

public static final int **UNEXPECTED_STATE**

AUTHORIZATION_REQUIRED

public static final int **AUTHORIZATION_REQUIRED**

TENDER_NOT_ACCEPTED

public static final int **TENDER_NOT_ACCEPTED**

FILE_ACCESS_FAILED

public static final int **FILE_ACCESS_FAILED**

TONE_ERROR

public static final int **TONE_ERROR**

(continued from last page)

MSR_ERROR

public static final int **MSR_ERROR**

APPLICATION_LIMIT_EXCEEDED

public static final int **APPLICATION_LIMIT_EXCEEDED**

ERROR_HANDLER_OTHER_ERROR

public static final int **ERROR_HANDLER_OTHER_ERROR**

ITEM_NOT_RETURNABLE

public static final int **ITEM_NOT_RETURNABLE**

NO_ITEM_MATCH_FOR_COUPON

public static final int **NO_ITEM_MATCH_FOR_COUPON**

COUPON_VALUE_EXCEEDS_ITEM_VALUE

public static final int **COUPON_VALUE_EXCEEDS_ITEM_VALUE**

RETURN_COUPON_BEFORE_ITEM_VOID

public static final int **RETURN_COUPON_BEFORE_ITEM_VOID**

VOID_MUST_MATCH_PREVIOUS

public static final int **VOID_MUST_MATCH_PREVIOUS**

SAME_ERROR_TWICE

public static final int **SAME_ERROR_TWICE**

TOO_MANY_ERRORS

public static final int **TOO_MANY_ERRORS**

ERROR_HANDLER_EXCEPTION_DETECTED

public static final int **ERROR_HANDLER_EXCEPTION_DETECTED**

ERROR_HANDLER_FATAL_ERROR

public static final int **ERROR_HANDLER_FATAL_ERROR**

ERROR_HANDLER_OVERRIDE_ERROR

public static final int **ERROR_HANDLER_OVERRIDE_ERROR**

INVALID_MANAGER_OVERRIDE_NUMBER

public static final int **INVALID_MANAGER_OVERRIDE_NUMBER**

ITEM_NOT_FOR_SALE

public static final int **ITEM_NOT_FOR_SALE**

DISPLAY_ERROR

public static final int **DISPLAY_ERROR**

PASSWORD_EXPIRED

public static final int **PASSWORD_EXPIRED**

CUSTOMER_MUST_SIGN_DOCUMENT

public static final int **CUSTOMER_MUST_SIGN_DOCUMENT**

UNCHECKED_EXCEPTION

public static final int **UNCHECKED_EXCEPTION**

ERROR_ACCESSING_FACTORY

public static final int **ERROR_ACCESSING_FACTORY**

MICR_ERROR

public static final int **MICR_ERROR**

(continued from last page)

SERVER_NO_SESSION_AVAILABLE

public static final int **SERVER_NO_SESSION_AVAILABLE**

FACTORY_NO_SESSIONS_AVAILABLE

public static final int **FACTORY_NO_SESSIONS_AVAILABLE**

FACTORY_SESSION_NOT_READY

public static final int **FACTORY_SESSION_NOT_READY**

UNSUPPORTED_ERROR_HANDLING_MODE

public static final int **UNSUPPORTED_ERROR_HANDLING_MODE**

POS_APP_FAILURE

public static final int **POS_APP_FAILURE**

CONFIG_ERROR

public static final int **CONFIG_ERROR**

ENABLEMENT_ERROR

public static final int **ENABLEMENT_ERROR**

KEYLOCK_ERROR

public static final int **KEYLOCK_ERROR**

PRINTER_ERROR

public static final int **PRINTER_ERROR**

SCALE_ERROR

public static final int **SCALE_ERROR**

UNSUPPORTED_OPERATION

public static final int **UNSUPPORTED_OPERATION**

NO_SESSION_FACTORIES_FOUND

public static final int **NO_SESSION_FACTORIES_FOUND**

COIN_DISPENSER_ERROR

public static final int **COIN_DISPENSER_ERROR**

CASH_DRAWER_ERROR

public static final int **CASH_DRAWER_ERROR**

KEYBOARD_ERROR

public static final int **KEYBOARD_ERROR**

CLIENT_SESSION_MISMATCH

public static final int **CLIENT_SESSION_MISMATCH**

PIN_PAD_ERROR

public static final int **PIN_PAD_ERROR**

USER_DEFINED

public static final int **USER_DEFINED**

RMI_REGISTRY_FAILURE

public static final int **RMI_REGISTRY_FAILURE**

RMI_NAMING_FAILURE

public static final int **RMI_NAMING_FAILURE**

DEVICE_HANDLER_FAILURE

public static final int **DEVICE_HANDLER_FAILURE**

(continued from last page)

DEVICE_HOOK_ERROR

public static final int **DEVICE_HOOK_ERROR**

DEVICE_MANAGER_ERROR

public static final int **DEVICE_MANAGER_ERROR**

VERIFY_SIGNATURE

public static final int **VERIFY_SIGNATURE**

JAVA_POS_EXCEPTION

public static final int **JAVA_POS_EXCEPTION**

INVALID_TIMEOUT_VALUE

public static final int **INVALID_TIMEOUT_VALUE**

JAVA_INVOCATION_FAILED

public static final int **JAVA_INVOCATION_FAILED**

REMOTE_SESSION_SERVER_ERROR

public static final int **REMOTE_SESSION_SERVER_ERROR**

INPUT_CANCELLED

public static final int **INPUT_CANCELLED**

INVALID_DISPATCH_QUEUE

public static final int **INVALID_DISPATCH_QUEUE**

API_VALID_FOR_TSS_ONLY

public static final int **API_VALID_FOR_TSS_ONLY**

ERROR_STARTING_POS_APPLICATION

public static final int **ERROR_STARTING_POS_APPLICATION**

ERROR_STOPPING_POS_APPLICATION

public static final int **ERROR_STOPPING_POS_APPLICATION**

SESSION_NOT_ACTIVE

public static final int **SESSION_NOT_ACTIVE**

ERROR_CREATING_SESSION

public static final int **ERROR_CREATING_SESSION**

ERROR_CREATING_DEVICE_SERVER

public static final int **ERROR_CREATING_DEVICE_SERVER**

MSR_HOOK_SWIPE_ERROR

public static final int **MSR_HOOK_SWIPE_ERROR**

ERROR_CREATING_GENERAL_AGENT

public static final int **ERROR_CREATING_GENERAL_AGENT**

PROCEDURE_NOT_ALLOWED

public static final int **PROCEDURE_NOT_ALLOWED**

JPOS_NOEXIST

public static final int **JPOS_NOEXIST**

TERMINAL_DISABLED

public static final int **TERMINAL_DISABLED**

DEVICE_NOT_REDIRECTED

public static final int **DEVICE_NOT_REDIRECTED**

(continued from last page)

ERROR_ACCESSING_GENERAL_AGENT

`public static final int` **`ERROR_ACCESSING_GENERAL_AGENT`**

TIMEOUT_WAITING_FOR_AVAILABLE_SESSION

`public static final int` **`TIMEOUT_WAITING_FOR_AVAILABLE_SESSION`**

SESSION_NOT_AVAILABLE

`public static final int` **`SESSION_NOT_AVAILABLE`**

SESSION_READY_WAIT_TIMEOUT

`public static final int` **`SESSION_READY_WAIT_TIMEOUT`**

IO_E_INTERNAL

`public static final int` **`IO_E_INTERNAL`**

Device Accessor error codes. Reserved range: -800 through -899

IO_E_FAILURE

`public static final int` **`IO_E_FAILURE`**

IO_E_ILLEGAL

`public static final int` **`IO_E_ILLEGAL`**

NONE

`public static final int` **`NONE`**

Extended Error Codes

RMI_NAMING_REGISTRY_RETURNED_NULL

`public static final int` **`RMI_NAMING_REGISTRY_RETURNED_NULL`**

RMI_REMOTE_EXCEPTION

`public static final int` **`RMI_REMOTE_EXCEPTION`**

RMI_URL_EXCEPTION

`public static final int` **`RMI_URL_EXCEPTION`**

RMI_BIND_EXCEPTION

public static final int **RMI_BIND_EXCEPTION**

REMOTE_EXCEPTION

public static final int **REMOTE_EXCEPTION**

SESSION_FACTORY_REMOTE_EXCEPTION

public static final int **SESSION_FACTORY_REMOTE_EXCEPTION**

APPLICATION_NOT_IN_PROPER_SUBSTATE

public static final int **APPLICATION_NOT_IN_PROPER_SUBSTATE**

NO_SUCH_LOGGER

public static final int **NO_SUCH_LOGGER**

INVALID_LOGGER_LEVEL

public static final int **INVALID_LOGGER_LEVEL**

CLASS_NOT_FOUND

public static final int **CLASS_NOT_FOUND**

APPLICATION_ALREADY_ACTIVE

public static final int **APPLICATION_ALREADY_ACTIVE**

APPLICATION_NOT_ACTIVE

public static final int **APPLICATION_NOT_ACTIVE**

TSS_ERROR

public static final int **TSS_ERROR**

(continued from last page)

MSR_SET_TO_DECODE

public static final int **MSR_SET_TO_DECODE**

MSR_NOT_ENABLED

public static final int **MSR_NOT_ENABLED**

WEIGHT_AND_QUANTITY_MUTUALLY_EXCLUSIVE

public static final int **WEIGHT_AND_QUANTITY_MUTUALLY_EXCLUSIVE**

PRICE_AND_DEAL_PRICE_MUTUALLY_EXCLUSIVE

public static final int **PRICE_AND_DEAL_PRICE_MUTUALLY_EXCLUSIVE**

DEAL_PRICE_AND_DEAL_QUANTITY_REQUIRED

public static final int **DEAL_PRICE_AND_DEAL_QUANTITY_REQUIRED**

INVALID_DEAL_QUANTITY

public static final int **INVALID_DEAL_QUANTITY**

INVALID_QUANTITY

public static final int **INVALID_QUANTITY**

INVALID_PRICE

public static final int **INVALID_PRICE**

INVALID_DEAL_PRICE

public static final int **INVALID_DEAL_PRICE**

INVALID_WEIGHT

public static final int **INVALID_WEIGHT**

INVALID_EXPIRY_DATE

public static final int **INVALID_EXPIRY_DATE**

INVALID_TENDER_AMOUNT

public static final int **INVALID_TENDER_AMOUNT**

INVALID_CURRENCY_AMOUNT

public static final int **INVALID_CURRENCY_AMOUNT**

ARGUMENT_REQUIRED

public static final int **ARGUMENT_REQUIRED**

ACCOUNT_NUMBER_REQUIRED

public static final int **ACCOUNT_NUMBER_REQUIRED**

EXPIRATION_DATE_REQUIRED

public static final int **EXPIRATION_DATE_REQUIRED**

CREDIT_CARD_TYPE_REQUIRED

public static final int **CREDIT_CARD_TYPE_REQUIRED**

INVALID_LOYALTY_NUMBER

public static final int **INVALID_LOYALTY_NUMBER**

ITEM_CLASS_REQUIRED

public static final int **ITEM_CLASS_REQUIRED**

ITEM_CLASS_PROHIBITED

public static final int **ITEM_CLASS_PROHIBITED**

ITEM_STOCK_REQUIRED

public static final int **ITEM_STOCK_REQUIRED**

(continued from last page)

ITEM_STOCK_PROHIBITED

public static final int **ITEM_STOCK_PROHIBITED**

ITEM_DEPARTMENT_REQUIRED

public static final int **ITEM_DEPARTMENT_REQUIRED**

ITEM_DEPARTMENT_PROHIBITED

public static final int **ITEM_DEPARTMENT_PROHIBITED**

ITEM_PRICE_REQUIRED

public static final int **ITEM_PRICE_REQUIRED**

ITEM_WEIGHT_REQUIRED

public static final int **ITEM_WEIGHT_REQUIRED**

ITEM_QUANTITY_REQUIRED

public static final int **ITEM_QUANTITY_REQUIRED**

ITEM_WEIGHT_PROHIBITED

public static final int **ITEM_WEIGHT_PROHIBITED**

DATA_OUT_OF_RANGE

public static final int **DATA_OUT_OF_RANGE**

INVALID_AMOUNT

public static final int **INVALID_AMOUNT**

EXTENDED_PRICE_TOO_LARGE

public static final int **EXTENDED_PRICE_TOO_LARGE**

INVALID_DEPARTMENT

public static final int **INVALID_DEPARTMENT**

INVALID_CHARGE_PLAN

public static final int **INVALID_CHARGE_PLAN**

AMOUNT_REQUIRED

public static final int **AMOUNT_REQUIRED**

INVALID_FEE_AMOUNT

public static final int **INVALID_FEE_AMOUNT**

ITEM_CODE_REQUIRED

public static final int **ITEM_CODE_REQUIRED**

INVALID_TAX_CODE

public static final int **INVALID_TAX_CODE**

TAX_CODE_REQUIRED

public static final int **TAX_CODE_REQUIRED**

ID_REQUIRED

public static final int **ID_REQUIRED**

INVALID_TRANSACTION_TYPE

public static final int **INVALID_TRANSACTION_TYPE**

INVALID_ACCOUNT_NUMBER

public static final int **INVALID_ACCOUNT_NUMBER**

INVALID_PASSWORD

public static final int **INVALID_PASSWORD**

(continued from last page)

PASSWORD_REQUIRED

public static final int **PASSWORD_REQUIRED**

TRANSACTION_TYPE_REQUIRED

public static final int **TRANSACTION_TYPE_REQUIRED**

INVALID_AUTHORIZATION_CODE

public static final int **INVALID_AUTHORIZATION_CODE**

INVALID_DEPARTMENT_TOTALS_LIST_NUMBER

public static final int **INVALID_DEPARTMENT_TOTALS_LIST_NUMBER**

INVALID_BATCH_NUMBER

public static final int **INVALID_BATCH_NUMBER**

INVALID_TERMINAL_NUMBER

public static final int **INVALID_TERMINAL_NUMBER**

INVALID_TENDER_TYPE

public static final int **INVALID_TENDER_TYPE**

ITEM_PRICE_OR_QUANTITY_REQUIRED

public static final int **ITEM_PRICE_OR_QUANTITY_REQUIRED**

INVALID_ITEM_CODE

public static final int **INVALID_ITEM_CODE**

INVALID_TARE

public static final int **INVALID_TARE**

INVALID_DISCOUNT_CODE

public static final int **INVALID_DISCOUNT_CODE**

INVALID_ID

public static final int **INVALID_ID**

AUTHORIZATION_CODE_REQUIRED

public static final int **AUTHORIZATION_CODE_REQUIRED**

INVALID_VOUCHER_NUMBER

public static final int **INVALID_VOUCHER_NUMBER**

VOUCHER_NUMBER_REQUIRED

public static final int **VOUCHER_NUMBER_REQUIRED**

INVALID_REASON

public static final int **INVALID_REASON**

INVALID_DATE

public static final int **INVALID_DATE**

VOLUME_PROHIBITED

public static final int **VOLUME_PROHIBITED**

VOLUME_REQUIRED

public static final int **VOLUME_REQUIRED**

INVALID_VOLUME

public static final int **INVALID_VOLUME**

EXTENDED_PRICE_PROHIBITED

public static final int **EXTENDED_PRICE_PROHIBITED**

(continued from last page)

TRACK_DATA_REQUIRED

public static final int **TRACK_DATA_REQUIRED**

TRACK_DATA_PROHIBITED

public static final int **TRACK_DATA_PROHIBITED**

ID_PROHIBITED

public static final int **ID_PROHIBITED**

INVALID_ITEM_IDENTIFIER_TYPE

public static final int **INVALID_ITEM_IDENTIFIER_TYPE**

ITEM_QUANTITY_PROHIBITED

public static final int **ITEM_QUANTITY_PROHIBITED**

REASON_CODE_PROHIBITED

public static final int **REASON_CODE_PROHIBITED**

INVALID_INITIALIZE_STATE

public static final int **INVALID_INITIALIZE_STATE**

NEW_PASSWORD_PROHIBITED

public static final int **NEW_PASSWORD_PROHIBITED**

REASON_CODE_REQUIRED

public static final int **REASON_CODE_REQUIRED**

INVALID_CARD_ID

public static final int **INVALID_CARD_ID**

CARD_ID_REQUIRED

public static final int **CARD_ID_REQUIRED**

VALUE_EXCEEDS_CONNECTION_TIMEOUT

public static final int **VALUE_EXCEEDS_CONNECTION_TIMEOUT**

TENDER_NOT_AUTHORIZED

public static final int **TENDER_NOT_AUTHORIZED**

TOO_LONG_IN_STAND_IN

public static final int **TOO_LONG_IN_STAND_IN**

CARD_EXPIRED

public static final int **CARD_EXPIRED**

UNKNOWN_SERVICER

public static final int **UNKNOWN_SERVICER**

SERVICER_CLOSED

public static final int **SERVICER_CLOSED**

CARD_TYPE_UNKNOWN

public static final int **CARD_TYPE_UNKNOWN**

RISK_1

public static final int **RISK_1**

RISK_2

public static final int **RISK_2**

RISK_3

public static final int **RISK_3**

(continued from last page)

RISK_4

public static final int **RISK_4**

VERIFICATION_TIMEOUT

public static final int **VERIFICATION_TIMEOUT**

CREDIT_NOT_AVAILABLE

public static final int **CREDIT_NOT_AVAILABLE**

INVALID_CARD_DATA

public static final int **INVALID_CARD_DATA**

CALL_FOR_AUTHORIZATION

public static final int **CALL_FOR_AUTHORIZATION**

PAYMENT_SYSTEM_OFFLINE

public static final int **PAYMENT_SYSTEM_OFFLINE**

CUSTOMER_CHOSE_ANOTHER_TENDER_AT_PINPAD

public static final int **CUSTOMER_CHOSE_ANOTHER_TENDER_AT_PINPAD**

CUSTOMER_CANCELLED_TENDER_AT_PINPAD

public static final int **CUSTOMER_CANCELLED_TENDER_AT_PINPAD**

MSR_TRACK_DATA_REQUIRED

public static final int **MSR_TRACK_DATA_REQUIRED**

MSR_TRACK_DATA_NOT_ALLOWED

public static final int **MSR_TRACK_DATA_NOT_ALLOWED**

SCANNED_DATA_NOT_ALLOWED

public static final int **SCANNED_DATA_NOT_ALLOWED**

CUSTOMER_ALTERED_AMOUNT_AT_PINPAD

public static final int **CUSTOMER_ALTERED_AMOUNT_AT_PINPAD**

CANCELLED_BY_OPERATOR

public static final int **CANCELLED_BY_OPERATOR**

LOYALTY_NUMBER_REQUIRED

public static final int **LOYALTY_NUMBER_REQUIRED**

PIN_PAD_REQUIRED

public static final int **PIN_PAD_REQUIRED**

TENDER_EXPIRED

public static final int **TENDER_EXPIRED**

VOUCHER_NOT_YET_VALID

public static final int **VOUCHER_NOT_YET_VALID**

PIN_PAD_PROHIBITED

public static final int **PIN_PAD_PROHIBITED**

JOURNAL_ERROR

public static final int **JOURNAL_ERROR**

BUFFER_FULL

public static final int **BUFFER_FULL**

COVER_OPEN

public static final int **COVER_OPEN**

(continued from last page)

PAPER_LOW

public static final int **PAPER_LOW**

FILE_IO_ERROR

public static final int **FILE_IO_ERROR**

INVALID_APPLICATION_DATA

public static final int **INVALID_APPLICATION_DATA**

UNRECOGNIZED_PRINT_CHARACTERS

public static final int **UNRECOGNIZED_PRINT_CHARACTERS**

OUT_OF_MEMORY

public static final int **OUT_OF_MEMORY**

PAYMENT_SYSTEM_ERROR

public static final int **PAYMENT_SYSTEM_ERROR**

HARDWARE_PROBLEM

public static final int **HARDWARE_PROBLEM**

MANAGER_KEY_REQUIRED

public static final int **MANAGER_KEY_REQUIRED**

MANAGER_KEY_NOT_REMOVED

public static final int **MANAGER_KEY_NOT_REMOVED**

ITEM_IS_ON_THE_SCALE

public static final int **ITEM_IS_ON_THE_SCALE**

PRICE_PER_POUND_REQUIRED

public static final int **PRICE_PER_POUND_REQUIRED**

PRICE_PER_KILOGRAM_REQUIRED

public static final int **PRICE_PER_KILOGRAM_REQUIRED**

REPLACE_ITEM

public static final int **REPLACE_ITEM**

WEIGH_ITEM

public static final int **WEIGH_ITEM**

CHECK_ITEM_PRICE

public static final int **CHECK_ITEM_PRICE**

ITEM_CANNOT_BE_WEIGHED

public static final int **ITEM_CANNOT_BE_WEIGHED**

CHECK_WEIGHT_PERSONALIZATION

public static final int **CHECK_WEIGHT_PERSONALIZATION**

CHECK_OVERRIDE_PRICE

public static final int **CHECK_OVERRIDE_PRICE**

ITEM_NOT_ON_SCALE_CORRECTLY

public static final int **ITEM_NOT_ON_SCALE_CORRECTLY**

CHECK_CONFIGURATION

public static final int **CHECK_CONFIGURATION**

PREVIOUS_ITEM_ON_SCALE

public static final int **PREVIOUS_ITEM_ON_SCALE**

(continued from last page)

REWEIGH_ITEM

public static final int **REWEIGH_ITEM**

DEVICE_OFFLINE

public static final int **DEVICE_OFFLINE**

KEYSEQUENCE_NOT_FOUND

public static final int **KEYSEQUENCE_NOT_FOUND**

INVALID_KEYSEQUENCE_EXPRESSION

public static final int **INVALID_KEYSEQUENCE_EXPRESSION**

FCODE_NOT_FOUND

public static final int **FCODE_NOT_FOUND**

INVALID_KEY_SEQUENCE

public static final int **INVALID_KEY_SEQUENCE**

NONNUMERIC_FUNCTION_CODE

public static final int **NONNUMERIC_FUNCTION_CODE**

UNABLE_TO_LOAD_PROPERTIES_FILE

public static final int **UNABLE_TO_LOAD_PROPERTIES_FILE**

NO_ERROR_HELPER_CONFIGURED

public static final int **NO_ERROR_HELPER_CONFIGURED**

TAX_TABLE_NOT_FOUND

public static final int **TAX_TABLE_NOT_FOUND**

MISSING_DESCRIPTOR

public static final int **MISSING_DESCRIPTOR**

TAX_OPTION_MISMATCH

public static final int **TAX_OPTION_MISMATCH**

MISSING_EXCHANGE_RATE

public static final int **MISSING_EXCHANGE_RATE**

UNKNOWN_MICR_FORMAT

public static final int **UNKNOWN_MICR_FORMAT**

UNDEFINED_TENDER_VARIETY

public static final int **UNDEFINED_TENDER_VARIETY**

NO_DEFAULT_ID_CONFIGURED

public static final int **NO_DEFAULT_ID_CONFIGURED**

FACTORY_ERROR

public static final int **FACTORY_ERROR**

NO_DEFAULT_MANAGER_OVERRIDE_PASSWORD

public static final int **NO_DEFAULT_MANAGER_OVERRIDE_PASSWORD**

MAX_NUMBER_OF_ITEMS

public static final int **MAX_NUMBER_OF_ITEMS**

MAX_NUMBER_OF_PRICE_CHANGES

public static final int **MAX_NUMBER_OF_PRICE_CHANGES**

TRANSACTION_TOTAL_TOO_LARGE

public static final int **TRANSACTION_TOTAL_TOO_LARGE**

(continued from last page)

DISCOUNT_PERCENTAGE_TOO_SMALL

public static final int **DISCOUNT_PERCENTAGE_TOO_SMALL**

DISCOUNT_PERCENTAGE_TOO_LARGE

public static final int **DISCOUNT_PERCENTAGE_TOO_LARGE**

DEPARTMENT_RETURN_LIMIT

public static final int **DEPARTMENT_RETURN_LIMIT**

CHANGE_AMOUNT_LIMIT

public static final int **CHANGE_AMOUNT_LIMIT**

TRANSACTION_LIMIT

public static final int **TRANSACTION_LIMIT**

ITEM_LIMIT

public static final int **ITEM_LIMIT**

DISCOUNT_LIMIT

public static final int **DISCOUNT_LIMIT**

ACCOUNT_TENDER_LIMIT

public static final int **ACCOUNT_TENDER_LIMIT**

TENDER_AMOUNT_LIMIT

public static final int **TENDER_AMOUNT_LIMIT**

NUMBER_OF_TENDERS_LIMIT

public static final int **NUMBER_OF_TENDERS_LIMIT**

STAND_IN_COUNT_LIMIT

public static final int **STAND_IN_COUNT_LIMIT**

STAND_IN_AMOUNT_LIMIT

public static final int **STAND_IN_AMOUNT_LIMIT**

TENDER_FLOOR_LIMIT

public static final int **TENDER_FLOOR_LIMIT**

LOYALTY_POINTS_LIMIT

public static final int **LOYALTY_POINTS_LIMIT**

NUMBER_OF_COUPONS_LIMIT

public static final int **NUMBER_OF_COUPONS_LIMIT**

SUSPENDED_TRANSACTION_LIMIT

public static final int **SUSPENDED_TRANSACTION_LIMIT**

TARE_WEIGHT_TOO_LARGE

public static final int **TARE_WEIGHT_TOO_LARGE**

NEGATIVE_TRANSACTION_BALANCE

public static final int **NEGATIVE_TRANSACTION_BALANCE**

FOODSTAMP_TENDER_IN_EXCESS_OF_FOODSTAMP_BALANCE

public static final int **FOODSTAMP_TENDER_IN_EXCESS_OF_FOODSTAMP_BALANCE**

DAILY_LOYALTY_CARD_USAGE_LIMIT

public static final int **DAILY_LOYALTY_CARD_USAGE_LIMIT**

TENDER_TYPE_NOT_ALLOWED

public static final int **TENDER_TYPE_NOT_ALLOWED**

(continued from last page)

PAYMENT_ALREADY_ENTERED

public static final int **PAYMENT_ALREADY_ENTERED**

NOT_ALLOWED_OFFLINE

public static final int **NOT_ALLOWED_OFFLINE**

NOT_ALLOWED_REENTRY

public static final int **NOT_ALLOWED_REENTRY**

CHARGE_PLAN_NOT_ALLOWED

public static final int **CHARGE_PLAN_NOT_ALLOWED**

NOT_AUTHORIZED

public static final int **NOT_AUTHORIZED**

OPERATOR_ALREADY_ACTIVE

public static final int **OPERATOR_ALREADY_ACTIVE**

NOT_ALLOWED_TRAINING

public static final int **NOT_ALLOWED_TRAINING**

ACCOUNT_PAID_IN_FULL

public static final int **ACCOUNT_PAID_IN_FULL**

EXTERNAL_TENDER_AUTHORIZATION_SUSPENDED

public static final int **EXTERNAL_TENDER_AUTHORIZATION_SUSPENDED**

ACCOUNT_NOT_YET_AVAILABLE

public static final int **ACCOUNT_NOT_YET_AVAILABLE**

FULLSCREEN_NOT_SUPPORTED

public static final int **FULLSCREEN_NOT_SUPPORTED**

PINPAD_NOT_AVAILABLE

public static final int **PINPAD_NOT_AVAILABLE**

CLOSING_ACCOUNTING_PERIOD

public static final int **CLOSING_ACCOUNTING_PERIOD**

PICKUP_NEEDED

public static final int **PICKUP_NEEDED**

TILL_EXCHANGE_NEEDED

public static final int **TILL_EXCHANGE_NEEDED**

MONITOR_ALREADY_ACTIVE

public static final int **MONITOR_ALREADY_ACTIVE**

HOME_STORE_MUST_REDEEM_LOYALTY_POINTS

public static final int **HOME_STORE_MUST_REDEEM_LOYALTY_POINTS**

MORE_LOYALTY_POINTS_NEEDED

public static final int **MORE_LOYALTY_POINTS_NEEDED**

MINIMUM_SALE_NOT_SATISFIED

public static final int **MINIMUM_SALE_NOT_SATISFIED**

RETRIEVE_TRANSACTION_WHERE_SUSPENDED

public static final int **RETRIEVE_TRANSACTION_WHERE_SUSPENDED**

UNABLE_TO_RETRIEVE_TRANSACTION

public static final int **UNABLE_TO_RETRIEVE_TRANSACTION**

(continued from last page)

TRANSACTION_ALREADY_RETRIEVED

public static final int **TRANSACTION_ALREADY_RETRIEVED**

SAME_OPERATOR_MUST_RETRIEVE_TRANSACTION

public static final int **SAME_OPERATOR_MUST_RETRIEVE_TRANSACTION**

NON_WIC_ITEM

public static final int **NON_WIC_ITEM**

NON_WIC_TENDER

public static final int **NON_WIC_TENDER**

TAX_EXEMPTION_NOT_ALLOWED_FOR_ITEM

public static final int **TAX_EXEMPTION_NOT_ALLOWED_FOR_ITEM**

DISCOUNT_NOT_ALLOWED_FOR_ITEM

public static final int **DISCOUNT_NOT_ALLOWED_FOR_ITEM**

ITEM_DISCOUNTS_NOT_ALLOWED

public static final int **ITEM_DISCOUNTS_NOT_ALLOWED**

MICR_DATA_NOT_ALLOWED

public static final int **MICR_DATA_NOT_ALLOWED**

EBT_NOT_ALLOWED

public static final int **EBT_NOT_ALLOWED**

HOST_REQUEST_PENDING

public static final int **HOST_REQUEST_PENDING**

VALUE_CARDS_MUST_BE_VOIDED

public static final int **VALUE_CARDS_MUST_BE_VOIDED**

NOT_APPROVED_BY_PAYMENT_SYSTEM

public static final int **NOT_APPROVED_BY_PAYMENT_SYSTEM**

CASHBACK_ON_FINAL_TENDER_ONLY

public static final int **CASHBACK_ON_FINAL_TENDER_ONLY**

LOYALTY_COUPON_NOT_APPLICABLE_TO_CUSTOMER_STATUS_LEVEL

public static final int **LOYALTY_COUPON_NOT_APPLICABLE_TO_CUSTOMER_STATUS_LEVEL**

TRANSACTION_NOT_ACTIVE

public static final int **TRANSACTION_NOT_ACTIVE**

TRANSACTION_CANNOT_BE_VOIDED

public static final int **TRANSACTION_CANNOT_BE_VOIDED**

NO_SUCH_SUSPENDED_TRANSACTION

public static final int **NO_SUCH_SUSPENDED_TRANSACTION**

TRANSACTION_IN_PROGRESS

public static final int **TRANSACTION_IN_PROGRESS**

TRANSACTION_ALREADY_IN_PROGRESS

public static final int **TRANSACTION_ALREADY_IN_PROGRESS**

NOT_IMPLEMENTED

public static final int **NOT_IMPLEMENTED**

COMMUNICATION_ERROR

public static final int **COMMUNICATION_ERROR**

(continued from last page)

PIN_COULD_NOT_BE_OBTAINED

public static final int **PIN_COULD_NOT_BE_OBTAINED**

DEVICE_IS_LOADING

public static final int **DEVICE_IS_LOADING**

LOAD_ERROR

public static final int **LOAD_ERROR**

UNSUPPORTED_CAPABILITY

public static final int **UNSUPPORTED_CAPABILITY**

ACCOUNT_NUMBER_ERROR

public static final int **ACCOUNT_NUMBER_ERROR**

AUTHORIZATION_NUMBER_MISMATCH

public static final int **AUTHORIZATION_NUMBER_MISMATCH**

CLOSE_DRAWER

public static final int **CLOSE_DRAWER**

NONE_AVAILABLE

public static final int **NONE_AVAILABLE**

com.ibm.retail.si.util

Class AEFException

```
java.lang.Object
  |-java.lang.Throwable
    |-java.lang.Exception
      |-com.ibm.retail.si.util.AEFException
```

public class **AEFException**
extends java.lang.Exception

Encapsulates a throwable Exception in the Store Integrator.

Field Summary

int	errorCode
int	errorCodeExtended
java.lang.String	errorKey
com.ibm.retail.si.util.AEFError	errorObject
com.ibm.retail.si.util.AEFException	nextException
java.lang.Throwable	originalThrowable
com.ibm.retail.si.util.AEFException	previousException

Constructor Summary

AEFException(int errorCode) Construct an AEFException with an error code
AEFException(int errorCode,int errorCode) Construct an AEFException with an error code and extended error code.
AEFException(int errorCode,java.lang.String errorCode) Construct an AEFException with an error code and error description.
AEFException(int errorCode,int errorCode,java.lang.String errorCode) Construct an AEFException with an error code, extended error code, and error description.
AEFException(int errorCode,int errorCode,java.lang.Throwable errorCode) Construct an AEFException with an error code and description as a wrapper for an original exception.

`AEFException(int errorCode,int errorCode,java.lang.String errorCode,java.lang.Throwable errorCode)` Construct an AEFException with an error code, extended error code, and error description which is a wrapper for an original Exception.
`AEFException(int errorCode,int errorCode,java.lang.String errorCode,java.lang.Throwable errorCode,java.lang.String errorCode)` Construct an AEFException with an error code, extended error code, and error description which is a wrapper for an original Exception.
`AEFException(int errorCode,int errorCode,AEFException errorCode)` Construct an AEFException with an error code and description as a wrapper for an original exception.
`AEFException(int errorCode,int errorCode,java.lang.String errorCode,AEFException errorCode)` Construct an AEFException with an error code, extended error code, and error description which is a wrapper for an original Exception.
`AEFException(int errorCode,int errorCode,java.lang.String errorCode,AEFException errorCode,java.lang.String errorCode)` Construct an AEFException with an error code, extended error code, and error description which is a wrapper for an original Exception.
`AEFException(int errorCode,int errorCode,java.lang.String errorCode,java.lang.String errorCode)` Construct an AEFException with an error code, extended error code, and error description which is a wrapper for an original Exception.
`AEFException(java.lang.String description,java.lang.Throwable description,AEFError description)` Construct an AEFException with an error code, extended error code, and error description which is a wrapper for an original Exception.
`AEFException(java.lang.String description,AEFException description,AEFError description)` Construct an AEFException with an error code, extended error code, and error description which is a wrapper for an original Exception.
`AEFException(java.lang.String description,AEFError description)` Construct an AEFException with an error code, extended error code, and error description which is a wrapper for an original Exception.

Method Summary

`void`	`appendExceptions(AEFException exceptionsToAppend)` This method takes an AEFException and sets the Previous Exception attribute for the last AEFException object on this list (the object that we called the method on) to point to it.
`java.lang.String`	`getAEFExceptionText(AEFException currentAEFException,int currentAEFException)` This method will return the information for this AEFException as a String.
`java.lang.String`	`getDescription()` Returns the description for the exception.
`int`	`getErrorCode()` Returns the error code for the exception.
`int`	`getErrorCodeExtended()` Returns the extended error code for the exception.

java.lang.String	getErrorKey() Returns the AEFErrorKey for this exception.
AEFError	getErrorObject() Gets the AEF error object for this exception.
AEFException	getNextException() Gets the next AEFException on the list of chained exceptions.
AEFException	getOriginalException() Gets the original AEFException (last in the chain) on the list of chained exceptions.
java.lang.Throwable	getOriginalThrowable() Gets the original throwable (the cause in JVM 1.
AEFException	getPreviousException() Gets the previous AEFException on the list of chained exceptions.
java.lang.String	getStackText(java.lang.Throwable currentThrowable,int currentThrowable) This method will return the stack trace for a throwable as a String.
void	printExceptions() This method will print all the exceptions in the chain starting with this exception.
void	setErrorCode(int newErrorCode) Set the error code for this exception.
void	setErrorKey(java.lang.String newErrorKey) Sets the Exception object imbedded within this exception.
void	setErrorObject(AEFError newErrorObject) Set the AEF error object for this exception.
void	setExtendedErrorCode(int newErrorCode) Set the extended error code for this exception.
void	setNextException(AEFException newNextException) Set the next AEFException on the list of chained exceptions.
void	setOriginalThrowable(java.lang.Throwable newOriginalThrowable) Set the original throwable (the cause in JVM 1.
void	setPreviousException(AEFException prevException) Set the previous AEFException on the list of chained exceptions.
java.lang.String	toString() This method will print a summary of the exceptions.

Methods inherited from : class `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from : class `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
wait
```

Fields

errorCode

`protected int` **`errorCode`**

errorCodeExtended

`protected int` **`errorCodeExtended`**

errorKey

`protected java.lang.String` **`errorKey`**

originalThrowable

`protected java.lang.Throwable` **`originalThrowable`**

previousException

`protected com.ibm.retail.si.util.AEFException` **`previousException`**

nextException

`protected com.ibm.retail.si.util.AEFException` **`nextException`**

errorObject

`protected com.ibm.retail.si.util.AEFError` **`errorObject`**

Constructors

AEFException

`public` **`AEFException`**`(int errorCode)`

Construct an AEFException with an error code

Parameters:

`errorCode -`
The error code

AEFException

```
public AEFException(int errorCode,
                    int errorCodeExtended)
```

Construct an AEFException with an error code and extended error code.

Parameters:

`errorCode -`
The error code
`errorCodeExtended -`
An addition extended error code

AEFException

```
public AEFException(int errorCode,
                    java.lang.String description)
```

Construct an AEFException with an error code and error description.

Parameters:

`errorCode -`
The error code.
`description -`
A textual description of the error.

AEFException

```
public AEFException(int errorCode,
                    int errorCodeExtended,
                    java.lang.String description)
```

Construct an AEFException with an error code, extended error code, and error description.

Parameters:

`errorCode -`
The error code.
`errorCodeExtended -`
Additional error code.
`description -`
A textual description of the error.

AEFException

```
public AEFException(int errorCode,
                    int errorCodeExtended,
                    java.lang.Throwable origException)
```

Construct an AEFException with an error code and description as a wrapper for an original exception.

Parameters:

`errorCode -`
The error code.
`description -`
A textual description of the error.
`origException -`
The orignal Exception.

AEFException

```
public AEFException(int errorCode,
                    int errorCodeExtended,
                    java.lang.String description,
                    java.lang.Throwable origException)
```

(continued from last page)

Construct an AEFException with an error code, extended error code, and error description which is a wrapper for an original Exception.

Parameters:

`errorCode -`
The error code.
`errorCodeExtended -`
Additional error code.
`description -`
A textual description of the error.
`origException -`
The orignal Exception.

AEFException

```
public AEFException(int errorCode,
                    int errorCodeExtended,
                    java.lang.String description,
                    java.lang.Throwable origException,
                    java.lang.String theErrorKey)
```

Construct an AEFException with an error code, extended error code, and error description which is a wrapper for an original Exception.

Parameters:

`errorCode -`
The error code.
`errorCodeExtended -`
Additional error code.
`description -`
A textual description of the error.
`origException -`
The orignal Exception.
`theErrorKey -`
The AEFError key for this error.

AEFException

```
public AEFException(int errorCode,
                    int errorCodeExtended,
                    AEFException origException)
```

Construct an AEFException with an error code and description as a wrapper for an original exception.

Parameters:

`errorCode -`
The error code.
`description -`
A textual description of the error.
`origException -`
The orignal Exception.

AEFException

```
public AEFException(int errorCode,
                    int errorCodeExtended,
                    java.lang.String description,
                    AEFException origException)
```

Construct an AEFException with an error code, extended error code, and error description which is a wrapper for an original Exception.

Parameters:

`errorCode -`
The error code.
`errorCodeExtended -`
Additional error code.

(continued from last page)

`description -`
A textual description of the error.
`origException -`
The orignal Exception.

AEFException

```
public AEFException(int errorCode,
                    int errorCodeExtended,
                    java.lang.String description,
                    AEFException origException,
                    java.lang.String theErrorKey)
```

Construct an AEFException with an error code, extended error code, and error description which is a wrapper for an original Exception.

Parameters:

`errorCode -`
The error code.
`errorCodeExtended -`
Additional error code.
`description -`
A textual description of the error.
`origException -`
The orignal Exception.
`theErrorKey -`
The AEFError key for this error.

AEFException

```
public AEFException(int errorCode,
                    int errorCodeExtended,
                    java.lang.String description,
                    java.lang.String theErrorKey)
```

Construct an AEFException with an error code, extended error code, and error description which is a wrapper for an original Exception.

Parameters:

`errorCode -`
The error code.
`errorCodeExtended -`
Additional error code.
`description -`
A textual description of the error.
`theErrorKey -`
The AEFError key for this error.

AEFException

```
public AEFException(java.lang.String description,
                    java.lang.Throwable origException,
                    AEFError errorObjectAEF)
```

Construct an AEFException with an error code, extended error code, and error description which is a wrapper for an original Exception.

Parameters:

`description -`
A textual description of the error.
`origException -`
The orignal Exception.
`errorObject -`
The AEFError object for this error.

(continued from last page)

AEFException

```
public AEFException(java.lang.String description,
                    AEFException origException,
                    AEFError errorObjectAEF)
```

Construct an AEFException with an error code, extended error code, and error description which is a wrapper for an original Exception.

Parameters:

`description -`
A textual description of the error.
`origException -`
The orignal Exception.
`errorObject -`
The AEFError object for this error.

AEFException

```
public AEFException(java.lang.String description,
                    AEFError errorObjectAEF)
```

Construct an AEFException with an error code, extended error code, and error description which is a wrapper for an original Exception.

Parameters:

`description -`
A textual description of the error.
`errorObject -`
The AEFError object for this error.

Methods

getErrorCode

```
public int getErrorCode()
```

Returns the error code for the exception.

Returns:

int The error code.

getErrorCodeExtended

```
public int getErrorCodeExtended()
```

Returns the extended error code for the exception.

Returns:

int The extended error code.

getDescription

```
public java.lang.String getDescription()
```

Returns the description for the exception.

Returns:

String The exception description.

(continued from last page)

getErrorKey

`public java.lang.String **getErrorKey**()`

Returns the AEFErrorKey for this exception.

Returns:

String The AEFErrorKey for this exception.

setErrorKey

`public void **setErrorKey**(java.lang.String newErrorKey)`

Sets the Exception object imbedded within this exception.

Parameters:

`newErrorKey -`
The AEFErrorKey for this exception.

getPreviousException

`public AEFException **getPreviousException**()`

Gets the previous AEFException on the list of chained exceptions.

Returns:

AEFException The previous AEFException on the list of chained exception.

setPreviousException

`public void **setPreviousException**(AEFException prevException)`

Set the previous AEFException on the list of chained exceptions.

Parameters:

`prevException -`
The AEFException to make the next exception on the list.

getNextException

`public AEFException **getNextException**()`

Gets the next AEFException on the list of chained exceptions.

Returns:

AEFException The next AEFException on the list of chained exception.

setNextException

`public void **setNextException**(AEFException newNextException)`

Set the next AEFException on the list of chained exceptions.

Parameters:

`newNextException -`
The AEFException to make the next exception on the list.

getOriginalThrowable

`public java.lang.Throwable **getOriginalThrowable**()`

(continued from last page)

Gets the original throwable (the cause in JVM 1.4 >) for this exception.

Returns:

Throwable The riginal throwable (the cause in JVM 1.4 >) for this exception.

setOriginalThrowable

`public void` **`setOriginalThrowable`**`(java.lang.Throwable newOriginalThrowable)`

Set the original throwable (the cause in JVM 1.4 >) for this exception.

Parameters:

`newOriginalThrowable -`
The original throwable (the cause in JVM 1.4 >) for this exception.

setErrorCode

`public void` **`setErrorCode`**`(int newErrorCode)`

Set the error code for this exception.

Parameters:

`newErrorCode -`
The new error code.

setExtendedErrorCode

`public void` **`setExtendedErrorCode`**`(int newErrorCode)`

Set the extended error code for this exception.

Parameters:

`newErrorCode -`
The new extended error code.

appendExceptions

`public void` **`appendExceptions`**`(AEFException exceptionsToAppend)`

This method takes an AEFException and sets the Previous Exception attribute for the last AEFException object on this list (the object that we called the method on) to point to it. The idea is to have this (the object that we called the method on) object (the newest exception) be a the start of the list of exception. Traversing the list of Previous Exception attributes would lead you back in time until the oldest exception in the chain is reached.

Parameters:

`exceptionsToAppend -`
The list of AEFException to append to this list.

printExceptions

`public void` **`printExceptions`**`()`

This method will print all the exceptions in the chain starting with this exception.

getStackText

```
public java.lang.String getStackText(java.lang.Throwable currentThrowable,
                                     int indentLevel)
```

This method will return the stack trace for a throwable as a String.

(continued from last page)

getAEFExceptionText

```
public java.lang.String getAEFExceptionText(AEFException currentAEFException,
                                            int indentLevel)
```

This method will return the information for this AEFException as a String.

toString

```
public java.lang.String toString()
```

This method will print a summary of the exceptions.

Returns:

String A summary of all the exceptions on the list.

getOriginalException

```
public AEFException getOriginalException()
```

Gets the original AEFException (last in the chain) on the list of chained exceptions.

Returns:

AEFException The last AEFException on the list of chained exception.

setErrorObject

```
public void setErrorObject(AEFError newErrorObject)
```

Set the AEF error object for this exception.

Parameters:

`newNextException -`
The AEF error object for this exception.

getErrorObject

```
public AEFError getErrorObject()
```

Gets the AEF error object for this exception.

Returns:

Throwable The AEF error object for this exception.

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W