Template version: proc-RFC_template-1_01_011016.dot

RFC 33 − Wiring Service

Confidential, Draft
rfc-0033-Wiring

26 Pages

Abstract

Service that connects Producer and Consumer services and provides a conduit through which data flows.

0 Document Information

0.1 Table of Contents

0 Document Information..... 1

0.1 Table of Contents 1
0.2 Status 3
0.3 Acknowledgement 3
0.4 Terminology and Document Conventions 3
0.5 Revision History 3

1 Introduction 4

2 Motivation and Rationale 5

3 Technical Discussion 5
3.1 Package org.osgi.service.wireadmin Description 5
3.2 org.osgi.service.wireadmin Interface Constants 5
3.2.1 WIREADMIN_PID 6
3.2.2 WIREADMIN_PRODUCER_PID 7
3.2.3 WIREADMIN_CONSUMER_PID 7
3.2.4 WIREADMIN_FILTER 7
3.2.5 WIREVALUE_CURRENT 8
3.2.6 WIREVALUE_PREVIOUS 8
3.2.7 WIREVALUE_DELTA_ABSOLUTE 8
3.2.8 WIREVALUE_DELTA_RELATIVE 8
3.2.9 WIREVALUE_ELAPSED 8
3.2.10 WIREADMIN_PRODUCER_FILTERS 8
3.2.11 WIREADMIN_CONSUMER_FLAVORS 8
3.2.12 WIREADMIN_PRODUCER_FLAVORS 9
3.2.13 WIREADMIN_EVENTS 9
3.3 org.osgi.service.wireadmin Interface Consumer 9
3.3.1 updated 10
3.3.2 producersConnected 10
3.4 org.osgi.service.wireadmin Interface Producer 11
3.4.1 polled 12
3.4.2 consumersConnected 12
3.5 org.osgi.service.wireadmin Interface Wire 12
3.5.1 isValid 13
3.5.2 isConnected 14
3.5.3 getFlavors 14
3.5.4 update 14
3.5.5 poll 15
3.5.6 getLastValue 15
3.5.7 getProperties 15
3.6 org.osgi.service.wireadmin Interface WireAdmin 16
3.6.1 createWire 16
3.6.2 deleteWire 17
3.6.3 updateWire 18
3.6.4 getWires 18
3.7 org.osgi.service.wireadmin Class WireAdminEvent 19
3.7.1 PRODUCER_EXCEPTION 21
3.7.2 CONSUMER_EXCEPTION 21
3.7.3 WIRE_CREATED 21
3.7.4 WIRE_UPDATED 21

3.7.5 WIRE_DELETED 21
3.7.6 WIRE_CONNECTED 22
3.7.7 WIRE_DISCONNECTED 22
3.7.8 WIRE_TRACE 22
3.7.9 WireAdminEvent 22
3.7.10 getServiceReference 23
3.7.11 getWire 23
3.7.12 getType 23
3.7.13 getThrowable 24
3.8 org.osgi.service.wireadmin Interface WireAdminListener 24
3.8.1 wireAdminEvent 25

4 Security Considerations 25

5 Document Support 25
5.1 References 25
5.2 Author's Address 25
5.3 Acronyms and Abbreviations 26
5.4 End of Document 26

0.2 Status

This document specifies a Wiring Service API for the Open Services Gateway Initiative, and requests discussion and suggestions for improvements. Distribution of this document is unlimited within OSGi.

0.3 Acknowledgement

This work began in VEG which performed the lion share of the work while CPEG was busy completing SPR2.

0.4 Terminology and Document Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY" and "OPTIONAL" in this document are to be interpreted as described in [1].

```
Source code is shown in this typeface.
```

0.5 Revision History

The last named individual in this history is currently responsible for this document.

Revision	Date	Comments

All Page Within This Box

Initial	2001-10-22	First draft based upon prior work by VEG. Substantially modified design. Cleaned up API and javadoc comments. The following areas were modified/clarified: • Removed WirePermission. It was too fine grained. Rearrangement of methods and use of ServicePermission should be sufficient. • Removed WireEvent and listener. These seem to be of little use. If some one is interested in a Producer changed the value, they can be a Consumer and wire themselves to the Producer. • Moved wire properties to Wire and reduced the number of defined property names. BJ Hargrave, IBM, hargrave@us.ibm.com
Second CPEG draft	2001-10-24	Revised API based upon items at today's CPEG meeting call. • Removed getFlavor method from Consumer and added new required service property for both Consumer and Producer. The property will specify the flavors for each side. • Added comment that the update/polled methods may be called before the *Connected methods. • Renamed connected methods to that both interfaces can be implemented on the same class. • Added WireAdminListener and WireAdminEvent back. Used to notify interested parties of Producer and Consumer exceptions. • WireAdmin.connect replaces the properties when the wire already exists. • Added "wire.pid" wire property and removed the Wire.get*PID methods since the values are available from the wire properties. BJ Hargrave, IBM, hargrave@us.ibm.com
Review Draft	2002-01-17	Revised API based upon CPEG meeting. Numerous changes were made since the last draft (see meeting minutes). BJ Hargrave, IBM, hargrave@us.ibm.com

1 Introduction

<Need details>

All Page Within This Box

2 Motivation and Rationale

<Need details>

3 Technical Discussion

3.1 Package org.osgi.service.wireadmin Description

The OSGi Wire Admin service Package. Specification Version 1.0.

Bundles wishing to use this package must list the package in the Import-Package header of the bundle's manifest. For example:

Import-Package: org.osgi.service.wireadmin; specification-version=1.0

3.2 org.osgi.service.wireadmin Interface Constants

public interface **Constants**

Defines standard names for `Wire` properties, wire filter attributes, `Consumer` and `Producer` service properties.

Version:
$Revision: 1.2 $

Author:
Open Services Gateway Initiative

Field Summary

`static java.lang.String`	**`WIREADMIN_CONSUMER_FLAVORS`** Service Registration property (named "wireadmin.consumer.flavors") specifying the list of data types understood by this `Consumer` service.
`static java.lang.String`	**`WIREADMIN_CONSUMER_PID`** `Wire` property key (named "wireadmin.consumer.pid") specifying the "service.pid" of the associated `Consumer` service.
`static java.lang.String`	**`WIREADMIN_EVENTS`** Service Registration property (named "wireadmin.events") specifying the

All Page Within This Box

	`WireAdminEvent` type of interest to a `WireAdminListener` service.
`static java.lang.String`	**`WIREADMIN_FILTER`** `Wire` property key (named "wireadmin.filter") specifying a filter used to control the delivery rate of data between the `Producer` and the `Consumer`.
`static java.lang.String`	**`WIREADMIN_PID`** `Wire` property key (named "wireadmin.pid") specifying the persistent identity (PID) of this `Wire` object.
`static java.lang.String`	**`WIREADMIN_PRODUCER_FILTERS`** Service Registration property (named "wireadmin.producer.filters").
`static java.lang.String`	**`WIREADMIN_PRODUCER_FLAVORS`** Service Registration property (named "wireadmin.producer.flavors") specifying the list of data types available from this `Producer` service.
`static java.lang.String`	**`WIREADMIN_PRODUCER_PID`** `Wire` property key (named "wireadmin.producer.pid") specifying the "service.pid" of the associated `Producer` service.
`static java.lang.String`	**`WIREVALUE_CURRENT`** Wire filter attribute (named "wirevalue.current") representing the current value.
`static java.lang.String`	**`WIREVALUE_DELTA_ABSOLUTE`** Wire filter attribute (named "wirevalue.delta.absolute") representing the absolute delta.
`static java.lang.String`	**`WIREVALUE_DELTA_RELATIVE`** Wire filter attribute (named "wirevalue.delta.relative") representing the relative delta.
`static java.lang.String`	**`WIREVALUE_ELAPSED`** Wire filter attribute (named "wirevalue.elapsed") representing the elapsed time, in ms, between this filter evaluation and the last update of the `Consumer` service.
`static java.lang.String`	**`WIREVALUE_PREVIOUS`** Wire filter attribute (named "wirevalue.previous") representing the previous value.

Field Detail

3.2.1 WIREADMIN_PID

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

`Wire` property key (named "wireadmin.pid") specifying the persistent identity (PID) of this `Wire` object.

Each `Wire` object has a PID to allow unique and persistent identification of a specific `Wire` object. The PID must be generated by the `WireAdmin` service when the `Wire` object is created.

This wire property is automatically set by the `WireAdmin` service. The value of the property must be of type `java.lang.String`.

All Page Within This Box

3.2.2 WIREADMIN_PRODUCER_PID

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

`Wire` property key (named "wireadmin.producer.pid") specifying the "service.pid" of the associated `Producer` service.

This wire property is automatically set by the `WireAdmin` service. The value of the property must be of type `java.lang.String`.

3.2.3 WIREADMIN_CONSUMER_PID

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

`Wire` property key (named "wireadmin.consumer.pid") specifying the "service.pid" of the associated `Consumer` service.

This wire property is automatically set by the `WireAdmin` service. The value of the property must be of type `java.lang.String`.

3.2.4 WIREADMIN_FILTER

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

`Wire` property key (named "wireadmin.filter") specifying a filter used to control the delivery rate of data between the `Producer` and the `Consumer`.

This property should contain a filter as described in the org.osgi.framework.Filter class. The filter can be used to specify when an updated value from the `Producer` should be delivered to the `Consumer`. In many cases the `Consumer` service does not need to receive the data with the same rate that the `Producer` service can generate data. This property can be used to control the delivery rate.

The filter can use a number of pre-defined attributes that can be used to control the delivery of new data values. If the filter produces a match upon the wire filter attributes, the `Consumer` service should be notifed of the updated data value.

If the `Producer` service was registered with the `WIREADMIN_PRODUCER_FILTERS` service property indicating that the `Producer` service will perform the data filtering then the `Wire` object will not perform data filtering. Otherwise, the `Wire` object must perform basic filtering. Basic filtering includes supporting the following standard wire filter attributes:

- `WIREVALUE_CURRENT` - Current value
- `WIREVALUE_PREVIOUS` - Previous value
- `WIREVALUE_DELTA_ABSOLUTE` - Absolute delta
- `WIREVALUE_DELTA_RELATIVE` - Relative delta
- `WIREVALUE_ELAPSED` - Elapsed time

See Also:

All Page Within This Box

"org.osgi.framework.Filter"

3.2.5 WIREVALUE_CURRENT

`public static final java.lang.String` **`WIREVALUE_CURRENT`**
Wire filter attribute (named "wirevalue.current") representing the current value.

3.2.6 WIREVALUE_PREVIOUS

`public static final java.lang.String` **`WIREVALUE_PREVIOUS`**
Wire filter attribute (named "wirevalue.previous") representing the previous value.

3.2.7 WIREVALUE_DELTA_ABSOLUTE

`public static final java.lang.String` **`WIREVALUE_DELTA_ABSOLUTE`**
Wire filter attribute (named "wirevalue.delta.absolute") representing the absolute delta. The absolute (always positive) difference between the last update and the current value (only when numeric). This attribute must not be used when the values are not numeric.

3.2.8 WIREVALUE_DELTA_RELATIVE

`public static final java.lang.String` **`WIREVALUE_DELTA_RELATIVE`**
Wire filter attribute (named "wirevalue.delta.relative") representing the relative delta. The relative difference is (`current` - `previous`) / `current` (only when numeric). This attribute must not be used when the values are not numeric.

3.2.9 WIREVALUE_ELAPSED

`public static final java.lang.String` **`WIREVALUE_ELAPSED`**
Wire filter attribute (named "wirevalue.elapsed") representing the elapsed time, in ms, between this filter evaluation and the last update of the `Consumer` service.

3.2.10 WIREADMIN_PRODUCER_FILTERS

`public static final java.lang.String` **`WIREADMIN_PRODUCER_FILTERS`**
Service Registration property (named "wireadmin.producer.filters"). A `Producer` service registered with this property indicates to the `WireAdmin` service that the `Producer` service implements at least the filtering as described for the `WIREADMIN_FILTER` property. If the `Producer` service is is not registered with this property, the `Wire` object must perform the basic filtering as described in `WIREADMIN_FILTER`.

The type of the property value is not relavent. Only its presence is relavent.

3.2.11 WIREADMIN_CONSUMER_FLAVORS

`public static final java.lang.String` **`WIREADMIN_CONSUMER_FLAVORS`**
Service Registration property (named "wireadmin.consumer.flavors") specifying the list of data types understood by this `Consumer` service.

All Page Within This Box

The `Consumer` service object must be registered with this service property.

The list must be in the order of preference with the first type being the most preferred. The value of the property must be of type `java.lang.Class[]`.

3.2.12 WIREADMIN_PRODUCER_FLAVORS

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

Service Registration property (named "wireadmin.producer.flavors") specifying the list of data types available from this `Producer` service.

The `Producer` service object should be registered with this service property.

The value of the property must be of type `java.lang.Class[]`.

3.2.13 WIREADMIN_EVENTS

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

Service Registration property (named "wireadmin.events") specifying the `WireAdminEvent` type of interest to a `WireAdminListener` service. The value of the property is a bitwise OR of all the `WireAdminEvent` types the `WireAdminListener` service wishes to receive and must be of type `java.lang.Integer`.

See Also:
`WireAdminEvent`

3.3 org.osgi.service.wireadmin Interface Consumer

---

public interface **Consumer**

Data Consumer. A service that can receive udpated values from `Producer` services.

Service objects registered under the `Consumer` interface are expected to consume values from a `Producer` service via a `Wire` object. A `Consumer` service may poll the `Producer` service by calling the `Wire.poll()` method. The `Consumer` service will also receive an updated value when called at it's `updated(org.osgi.service.wireadmin.Wire, java.lang.Object)` method. The `Producer` service should have coerced the value to be an instance of one of the types specified by the `Wire.getFlavors()` method, or one of their subclasses.

`Consumer` service objects must register with a "`service.pid`" and a `Constants.WIREADMIN_CONSUMER_FLAVORS` property. It is recommended that `Consumer` service objects also register with a "`service.description`" property.

If an exception is thrown by a `Consumer` object method, a `WireAdminEvent` of type `WireAdminEvent.CONSUMER_EXCEPTION` is broadcast by the Wire Admin service.

Security Considerations. Data consuming bundles will require ServicePermission to register a `Consumer` service. In general, only the `WireAdmin` service bundle should have ServicePermission to get a

Consumer service. Thus only the WireAdmin service bundle may directly call a Consumer service. Care must be taken in the sharing of Wire objects with other code.

Version:
$Revision: 1.4 $

Author:
Open Services Gateway Initiative

Method Summary

void	**producersConnected**(Wire[] wires) Update the list of Wire objects to which this Consumer service is connected.
void	**updated**(Wire wire, java.lang.Object value) Update the value.

Method Detail

3.3.1 updated

```
public void updated(Wire wire,
                    java.lang.Object value)
```

Update the value. This Consumer service is called by the Wire object with an updated value from the Producer service.

Note: This method may be called by a Wire object prior to this object being notified that it is connected to that Wire object (via the producersConnected(org.osgi.service.wireadmin.Wire[]) method).

Parameters:
wire - The Wire object which is delivering the updated value.
value - The updated value. The value is an instance of one of the types specified by the Wire.getFlavors() method.

3.3.2 producersConnected

```
public void producersConnected(Wire[] wires)
```

Update the list of Wire objects to which this Consumer service is connected.

This method is called when the Consumer is first registered and subsequently whenever a Wire associated with this Consumer becomes connected, is modified or becomes disconnected.

The Wire Admin service must call this method on a thread other than the thread which initiated the call. This implies that implementors of Consumer can be assured that the callback will not take place during registration when they execute the registration in a synchronized method.

Parameters:
wires - An array of the current and complete list of Wire objects to which this Consumer service is connected. May be null if the Consumer service is not currently connected to any Wire objects.

All Page Within This Box

3.4 org.osgi.service.wireadmin Interface Producer

public interface **Producer**

Data Producer. A service that can generate values to be used by `Consumer` services.

Service objects registered under the `Producer` interface are expected to produce values (internally generated or from external sensors). The value can be of different types. When delivering a value to a `Wire` object, the `Producer` service should coerce the value to be an instance of one of the types specified by `Wire.getFlavors()`. The classes are specified in order of preference.

When the data represented by the `Producer` object changes, this object should send the updated value by calling the `update` method on each of `Wire` objects passed in the most recent call to this object's `consumersConnected(org.osgi.service.wireadmin.Wire[])` method. These `Wire` objects will pass the value on to the associated `Consumer` service object.

The `Producer` service may use the information in the `Wire` object's properties to schedule the delivery of values to the `Wire` object.

`Producer` service objects must register with a "`service.pid`" and a `Constants.WIREADMIN_PRODUCER_FLAVORS` property. It is recommended that a `Producer` service object also registers with a "`service.description`" property. `Producer` service objects must register with a `Constants.WIREADMIN_PRODUCER_FILTERS` property if the `Producer` service will be performing filtering instead of the `Wire` object.

If an exception is thrown by a `Producer` object method, a `WireAdminEvent` of type `WireAdminEvent.PRODUCER_EXCEPTION` is broadcast by the Wire Admin service.

Security Considerations. Data producing bundles will require ServicePermission to register a `Producer` service. In general, only the `WireAdmin` service bundle should have ServicePermission to get a `Producer` service. Thus only the `WireAdmin` service bundle may directly call a `Producer` service. Care must be taken in the sharing of `Wire` objects with other code.

Version:
$Revision: 1.4 $

Author:
Open Services Gateway Initiative

Method Summary

void	**consumersConnected**(Wire[] wires) Update the list of `Wire`s to which this `Producer` is connected.
java.lang.Object	**polled**(Wire wire) Return the current value of this `Producer`.

All Page Within This Box

Method Detail

3.4.1 polled

`public java.lang.Object` **`polled`**`(Wire wire)`

Return the current value of this `Producer`.

This method is called by a `Wire` object in response to the `Consumer` calling the `Wire` object's `poll` method. The `Producer` should coerce the value to be an instance of one of the types specified by `Wire.getFlavors()`. The types are specified in order of of preference. The returned value should be as new or newer than the last value furnished by this object.

Note: This method may be called by a `Wire` object prior to this object being notified that it is connected to that `Wire` object (via the `consumersConnected(org.osgi.service.wireadmin.Wire[])` method).

Parameters:
`wire` - The `Wire` object which is polling this service.
Returns:
The current value of the `Producer` service or `null` if the value cannot be coerced into a compatible type.

3.4.2 consumersConnected

`public void` **`consumersConnected`**`(Wire[] wires)`

Update the list of `Wire`s to which this `Producer` is connected.

This method is called when the `Producer` is first registered and subsequently whenever a `Wire` associated with this `Producer` becomes connected, is modified or becomes disconnected.

The Wire Admin service must call this method on a thread other than the thread which initiated the call. This implies that implementors of `Producer` can be assured that the callback will not take place during registration when they execute the registration in a synchronized method.

Parameters:
`wires` - An array of the current and complete list of `Wire` objects to which this `Producer` is connected. May be `null` if the `Producer` is not currently connected to any `Wire`s.

3.5 org.osgi.service.wireadmin Interface Wire

public interface **Wire**

A connection between a `Producer` service and a `Consumer` service.

A `Wire` object connects a `Producer` service to a `Consumer` service. Both the `Producer` and `Consumer` services are identified by their unique "service.pid" values. The `Producer` and `Consumer` services may communicate with each other via `Wire` objects that connect them. The `Producer` service may send updated values to the `Consumer` service by calling the `update(java.lang.Object)` method. The

All Page Within This Box

`Consumer` service may request an updated value from the `Producer` service by calling the `poll()` method.

A `Producer` service and a `Consumer` service may be connected through multiple `Wire` objects.

Security Considerations. `Wire` objects are available to `Producer` and `Consumer` services connected to a given `Wire` object and to bundles which can access the `WireAdmin` service. A bundle must have ServicePermission to get the `WireAdmin` service to access all `Wire` objects. A bundle registering a `Producer` service or a `Consumer` service must have the appropriate ServicePermission to register the service and will be passed `Wire` objects when the service object's `consumersConnected` or `producersConnected` method is called.

Version:
$Revision: 1.3 $

Author:
Open Services Gateway Initiative

Method Summary

`java.lang.Class[]`	**`getFlavors`**`()` Return the list of data types understood by the `Consumer` service connected to this `Wire` object.
`java.lang.Object`	**`getLastValue`**`()` Return the last value sent through this `Wire` object.
`java.util.Dictionary`	**`getProperties`**`()` Return the wire properties for this `Wire` object.
`boolean`	**`isConnected`**`()` Return the connection state of this `Wire` object.
`boolean`	**`isValid`**`()` Return the state of this `Wire` object.
`java.lang.Object`	**`poll`**`()` Poll for an updated value.
`void`	**`update`**`(java.lang.Object value)` Update the value.

Method Detail

3.5.1 isValid

`public boolean` **`isValid`**`()`

Return the state of this `Wire` object.

A connected `Wire` must always be disconnected before becoming invalid.

Returns:
`false` if this `Wire` is invalid because it has been deleted via `WireAdmin.deleteWire(org.osgi.service.wireadmin.Wire)`; `true` otherwise.

All Page Within This Box

3.5.2 isConnected

`public boolean` **`isConnected`**`()`

Return the connection state of this `Wire` object.

A `Wire` is connected after the Wire Admin service receives notification that the `Producer` service and the `Consumer` service for this `Wire` object are both registered. This method will return `true` prior to notifying the `Producer` and `Consumer` services via calls to their respective `consumersConnected` and `producersConnected` methods.

A `WireAdminEvent` of type `WireAdminEvent.WIRE_CONNECTED` must be broadcast by the Wire Admin service when the `Wire` becomes connected.

A `Wire` object is disconnected when either the `Consumer` or `Producer` service is unregistered or the `Wire` object is deleted.

A `WireAdminEvent` of type `WireAdminEvent.WIRE_DISCONNECTED` must be broadcast by the Wire Admin service when the `Wire` becomes disconnected.

Returns:
`true` if both the `Producer` and `Consumer` for this `Wire` object are connected to the `Wire` object; `false` otherwise.

3.5.3 getFlavors

`public java.lang.Class[]` **`getFlavors`**`()`

Return the list of data types understood by the `Consumer` service connected to this `Wire` object. Note that subclasses of the classes in this list are acceptable data types as well.

The list is the value of the `Constants.WIREADMIN_CONSUMER_FLAVORS` service property of the `Consumer` service object connected to this object. If no such property was registered or the type of the property value is not `Class[]`, this method must return `null`.

Returns:
An array containing the list of classes understood by the `Consumer` service or `null` if the `Wire` is not connected, or the consumer did not register a `Constants.WIREADMIN_CONSUMER_FLAVORS` property or the value of the property is not of type `Class[]`.

3.5.4 update

`public void` **`update`**`(java.lang.Object value)`

Update the value.

This methods is called by the `Producer` service to notify the `Consumer` service connected to this `Wire` object of an updated value.

If the properties of this `Wire` object contain a `Constants.WIREADMIN_FILTER` property, then filtering is performed on this `Wire`. If the `Producer` service connected to this `Wire` object was registered with the service property `Constants.WIREADMIN_PRODUCER_FILTERS`, the `Producer`

All Page Within This Box

service will perform the filter the value according to the rules specified for the filter. Otherwise, this Wire object will filter the value according to the rules specified for the filter.

If no filtering is done, or the filter indicates the updated value should be delivered to the Consumer service, then this Wire object must call the Consumer.updated(org.osgi.service.wireadmin.Wire, java.lang.Object) method with the updated value. If this Wire object is not connected, then the Consumer service must not be called.

A WireAdminEvent of type WireAdminEvent.WIRE_TRACE must be broadcast by the Wire Admin service after the Consumer service has been successfully called.

Parameters:
value - The updated value. The value should be an instance of one of the types returned by getFlavors().
See Also:
Constants.WIREADMIN_FILTER

3.5.5 poll

public java.lang.Object **poll**()
Poll for an updated value.

This methods is normally called by the Consumer service to request an updated value from the Producer service connected to this Wire object. This Wire object will call the Producer.polled(org.osgi.service.wireadmin.Wire) method to obtain an updated value. If this Wire object is not connected, then the Producer service must not be called.

A WireAdminEvent of type WireAdminEvent.WIRE_TRACE must be broadcast by the Wire Admin service after the Producer service has been successfully called.

Returns:
An updated value whose type should be one of the types returned by getFlavors() or null if the Wire object is not connected, the Producer service threw an exception, or the Producer service returned a value which is not an instance of one of the types returned by getFlavors().

3.5.6 getLastValue

public java.lang.Object **getLastValue**()
Return the last value sent through this Wire object.

The returned value is the most recent, valid value passed to the update(java.lang.Object) method or returned by the poll() method of this object. If filtering is performed by this Wire object, this methods returns the last value provided by the Producer service.

Returns:
The last value passed though this Wire object or null if no valid values have been passed.

3.5.7 getProperties

`public java.util.Dictionary` **`getProperties`**`()`

Return the wire properties for this `Wire` object.

Returns:

The properties for this `Wire` object. The returned `Dictionary` must be read only.

3.6 org.osgi.service.wireadmin Interface WireAdmin

public interface **WireAdmin**

Wire Administration service.

This service can be used to create `Wire` objects connecting a `Producer` services and a `Consumer` services. `Wire` objects also have wire properties that may be specified when a `Wire` object is created. The `Producer` and `Consumer` services may use the `Wire` object's properties to manage or control their interaction. The use of `Wire` object's properties by a `Producer` or `Consumer` services is optional.

Security Considerations. A bundle must have ServicePermission to get the `WireAdmin` service to create, modify and delete `Wire` objects and to find `Wire` objects.

Version:

$Revision: 1.2 $

Author:

Open Services Gateway Initiative

Method Summary

`Wire`	**`createWire`**`(java.lang.String producerPID, java.lang.String consumerPID, java.util.Dictionary properties)` Create a new `Wire` object that connects a `Producer` service to a `Consumer` service.
`void`	**`deleteWire`**`(Wire wire)` Delete a `Wire`.
`Wire[]`	**`getWires`**`(java.lang.String filter)` Return the `Wire` objects that match the given `filter`.
`Wire`	**`updateWire`**`(Wire wire, java.util.Dictionary properties)` Update the properties of a `Wire` object.

Method Detail

3.6.1 createWire

```
public Wire createWire(java.lang.String producerPID,
                       java.lang.String consumerPID,
                       java.util.Dictionary properties)
```

Create a new `Wire` object that connects a `Producer` service to a `Consumer` service. The `Producer` service and `Consumer` service do not have to be registered when the the `Wire` object is created.

All Page Within This Box

The Wire configuration data is persistently stored. All Wire connections are reestablished when the WireAdmin service is registered. A Wire can be permanently removed by using the deleteWire(org.osgi.service.wireadmin.Wire) method.

The Wire object's properties must have case insensitive String objects as keys (like the Framework). However, the case of the key must be preserved. The type of the value of the property must be one of the following:

```
type         = basetype
 | vector | arrays

basetype = String | Integer | Long
 | Float | Double | Byte
 | Short | Character
 | Boolean

primitive    = long | int | short
 | char | byte | double | float

arrays    = primitive '[]' | basetype '[]'

vector    = Vector of basetype
```

The WireAdmin service must automatically add the following Wire properties:

- Constants.WIREADMIN_PID set to the value of the wire's persistent identity (PID). This value is generated by the WireAdmin service when a Wire object is created.
- Constants.WIREADMIN_PRODUCER_PID set to the value of Producer service's PID.
- Constants.WIREADMIN_CONSUMER_PID set to the value of Consumer service's PID.

If the properties argument already contains any of these keys, then the supplied values are replaced with the values assigned by the WireAdmin service.

The Wire Admin service must broadcast a WireAdminEvent of type WireAdminEvent.WIRE_CREATED after the new Wire object becomes available from getWires(java.lang.String).

Parameters:
producerPID - The "service.pid" of the Producer service to be connected to the Wire.
consumerPID - The "service.pid" of the Consumer service to be connected to the Wire.
properties - The Wire object's properties. This argument may be null if the caller does not wish to define any Wire object's properties.
Returns:
The Wire object for this connection.
Throws:
java.lang.IllegalArgumentException - If properties contains case variants of the same key name.

All Page Within This Box

3.6.2 deleteWire

`public void` **`deleteWire`**`(Wire wire)`

Delete a `Wire`.

The `Wire` representing a connection between a `Producer` service and a `Consumer` service is removed. The persistently stored configuration data for the `Wire` object is destroyed. The `Wire` object's method `Wire.isValid()` will return `false` after it is deleted.

The Wire Admin service must broadcast a `WireAdminEvent` of type `WireAdminEvent.WIRE_DELETED` after the `Wire` object becomes invalid.

Parameters:
`wire` - The `Wire` object which is to be deleted.

3.6.3 updateWire

`public Wire` **`updateWire`**`(Wire wire, java.util.Dictionary properties)`

Update the properties of a `Wire` object. The persistently stored configuration data for the `Wire` object is updated with the new properties and then the `Consumer` and `Producer` services will be called at the respective `Consumer.producersConnected(org.osgi.service.wireadmin.Wire[])` and `Producer.consumersConnected(org.osgi.service.wireadmin.Wire[])` methods.

The Wire Admin service must broadcast a `WireAdminEvent` of type `WireAdminEvent.WIRE_UPDATED` after the updated properties are available from the `Wire` object.

Parameters:
`wire` - The `Wire` object which is to be updated.
`properties` - The new `Wire` object's properties or `null` if no properties are required.

3.6.4 getWires

`public Wire[]` **`getWires`**`(java.lang.String filter) throws org.osgi.framework.InvalidSyntaxException`

Return the `Wire` objects that match the given `filter`.

The list of available `Wire` objects is matched against the specified `filter`. `Wire` objects which match the `filter` are returned. These `Wire` objects are not necessarily connected. The Wire Admin service should not return invalid `Wire` objects, but it is possible that a `Wire` object is deleted after it was placed in the list.

The filter matches against the `Wire` object's properties including `Constants.WIREADMIN_PRODUCER_PID`, `Constants.WIREADMIN_CONSUMER_PID` and `Constants.WIREADMIN_PID`.

Parameters:
`filter` - Filter string to select `Wire` objects or `null` to select all `Wire` objects.
Returns:
An array of `Wire` objects which match the `filter` or `null` if no `Wire` objects match the `filter`.
Throws:
`org.osgi.framework.InvalidSyntaxException` - If the specified `filter` has an invalid syntax.
See Also:
"org.osgi.framework.Filter"

All Page Within This Box

3.7 org.osgi.service.wireadmin
Class WireAdminEvent

```
java.lang.Object
  |
  +-org.osgi.service.wireadmin.WireAdminEvent
```

public class **WireAdminEvent**
extends java.lang.Object

A Wire Admin Event.

`WireAdminEvent` objects are delivered asynchronously to all registered `WireAdminListener` service objects which specify an interest in the `WireAdminEvent` type. However, events must be delivered in chronological order with respect to each listener. For example, a `WireAdminEvent` of type `WIRE_CONNECTED` must be delivered before a `WireAdminEvent` of type `WIRE_DISCONNECTED`.

A type code is used to identify the type of event. The following event types are defined:

- `WIRE_CREATED`
- `WIRE_CONNECTED`
- `WIRE_UPDATED`
- `WIRE_TRACE`
- `WIRE_DISCONNECTED`
- `WIRE_DELETED`
- `PRODUCER_EXCEPTION`
- `CONSUMER_EXCEPTION`

Additional event types may be defined in the future.

Event type values must be unique and disjoint bit values. Event types must be defined as a bit in a 32 bit integer and can thus be bitwise ORed together.

Security Considerations. `WireAdminEvent` objects contain `Wire` objects. Care must be taken in the sharing of `Wire` objects with other code.

Version:
$Revision: 1.3 $
Author:
Open Services Gateway Initiative
See Also:
`WireAdminListener`

Field Summary	
`static int`	**`CONSUMER_EXCEPTION`** A `Consumer` service method has thrown an exception.

All Page Within This Box

static int	**PRODUCER_EXCEPTION** A Producer service method has thrown an exception.
static int	**WIRE_CONNECTED** The WireAdminEvent type that indicates that an existing Wire object has become connected.
static int	**WIRE_CREATED** A Wire has been created.
static int	**WIRE_DELETED** A Wire has been deleted.
static int	**WIRE_DISCONNECTED** The WireAdminEvent type that indicates that an existing Wire object has become disconnected.
static int	**WIRE_TRACE** The WireAdminEvent type that indicates that a new value is transferred over the Wire object.
static int	**WIRE_UPDATED** A Wire has been updated.

Constructor Summary

WireAdminEvent(org.osgi.framework.ServiceReference reference, int type, Wire wire, java.lang.Throwable exception)
Constructs a WireAdminEvent object from the given ServiceReference object, event type, Wire object and exception.

Method Summary

org.osgi.framework.ServiceReference	**getServiceReference**() Return the ServiceReference object of the Wire Admin service that created this event.
java.lang.Throwable	**getThrowable**() Returns the exception associated with the event, if any.
int	**getType**() Return the type of this event.
Wire	**getWire**() Return the Wire object associated with this event.

Methods inherited from class java.lang.Object

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

Field Detail

All Page Within This Box

3.7.1 PRODUCER_EXCEPTION

`public static final int` **`PRODUCER_EXCEPTION`**

A `Producer` service method has thrown an exception.

This `WireAdminEvent` type indicates that a `Producer` service method has thrown an exception. The `getThrowable()` method will return the exception that the `Producer` service method raised.

The value of `PRODUCER_EXCEPTION` is 0x00000001.

3.7.2 CONSUMER_EXCEPTION

`public static final int` **`CONSUMER_EXCEPTION`**

A `Consumer` service method has thrown an exception.

This `WireAdminEvent` type indicates that a `Consumer` service method has thrown an exception. The `getThrowable()` method will return the exception that the `Consumer` service method raised.

The value of `CONSUMER_EXCEPTION` is 0x00000002.

3.7.3 WIRE_CREATED

`public static final int` **`WIRE_CREATED`**

A `Wire` has been created.

This `WireAdminEvent` type that indicates that a new `Wire` object has been created. An event is broadcast when `WireAdmin.createWire(java.lang.String, java.lang.String, java.util.Dictionary)` is called. The `getWire()` method will return the `Wire` object that has just been created.

The value of `WIRE_CREATED` is 0x00000004.

3.7.4 WIRE_UPDATED

`public static final int` **`WIRE_UPDATED`**

A `Wire` has been updated.

This `WireAdminEvent` type that indicates that an existing `Wire` object has been updated with new properties. An event is broadcast when `WireAdmin.updateWire(org.osgi.service.wireadmin.Wire, java.util.Dictionary)` is called with a valid wire. The `getWire()` method will return the `Wire` object that has just been updated.

The value of `WIRE_UPDATED` is 0x00000008.

3.7.5 WIRE_DELETED

All Page Within This Box

`public static final int` **`WIRE_DELETED`**

A `Wire` has been deleted.

This `WireAdminEvent` type that indicates that an existing wire has been deleted. An event is broadcast when `WireAdmin.deleteWire(org.osgi.service.wireadmin.Wire)` is called with a valid wire. `getWire()` will return the `Wire` object that has just been deleted.

The value of `WIRE_DELETED` is 0x00000010.

3.7.6 WIRE_CONNECTED

`public static final int` **`WIRE_CONNECTED`**

The `WireAdminEvent` type that indicates that an existing `Wire` object has become connected. The `Consumer` object and the `Producer` object that are associated with the `Wire` object have both been registered and the `Wire` object is connected. See `Wire.isConnected()` for a description of the connected state. This event may come before the `producersConnected` and `consumersConnected` method have returned or called to allow synchronous delivery of the events. Both methods can cause other `WireAdminEvent`s to take place and requiring this event to be send before these methods are returned would mandate asynchronous delivery.

The value of `WIRE_CONNECTED` is 0x00000020.

3.7.7 WIRE_DISCONNECTED

`public static final int` **`WIRE_DISCONNECTED`**

The `WireAdminEvent` type that indicates that an existing `Wire` object has become disconnected. The `Consumer` object or/and `Producer` object is/are unregistered breaking the connection between the two. See `Wire.isConnected()` for a description of the connected state.

The value of `WIRE_DISCONNECTED` is 0x00000040.

3.7.8 WIRE_TRACE

`public static final int` **`WIRE_TRACE`**

The `WireAdminEvent` type that indicates that a new value is transferred over the `Wire` object. This event is sent after the Consumer object has been notified by calling the `Consumer.updated(org.osgi.service.wireadmin.Wire, java.lang.Object)` method or the Consumer object requested a new value with the `Wire.poll()` method. This is an advisory event meaning that when this event is received, another update may already have occurred and this the `Wire.getLastValue()` method returns a newer value then the value that was communicated for this event.

The value of `WIRE_TRACE` is 0x00000080.

Constructor Detail

3.7.9 WireAdminEvent

```
public WireAdminEvent(org.osgi.framework.ServiceReference reference,
                      int type,
```

All Page Within This Box

Wire wire,
java.lang.Throwable exception)

Constructs a WireAdminEvent object from the given ServiceReference object, event type, Wire object and exception.

Parameters:

reference - The ServiceReference object of the Wire Admin service that created this event.

type - The event type. See getType().

wire - The Wire object associated with this event.

exception - An exception associated with this event. This may be null if no exception is associated with this event.

Method Detail

3.7.10 getServiceReference

public org.osgi.framework.ServiceReference **getServiceReference**()

Return the ServiceReference object of the Wire Admin service that created this event.

Returns:

The ServiceReference object for the Wire Admin service that created this event.

3.7.11 getWire

public Wire **getWire**()

Return the Wire object associated with this event.

Returns:

The Wire object associated with this event or null when no Wire object is associated with the event.

3.7.12 getType

public int **getType**()

Return the type of this event.

The type values are:

- WIRE_CREATED
- WIRE_CONNECTED
- WIRE_UPDATED
- WIRE_TRACE
- WIRE_DISCONNECTED
- WIRE_DELETED
- PRODUCER_EXCEPTION
- CONSUMER_EXCEPTION

Returns:

The type of this event.

All Page Within This Box

3.7.13 getThrowable

`public java.lang.Throwable` **`getThrowable`**`()`

Returns the exception associated with the event, if any.

Returns:

An exception or `null` if no exception is associated with this event.

3.8 org.osgi.service.wireadmin Interface WireAdminListener

public interface **WireAdminListener**

Listener for Wire Admin Events.

`WireAdminListener` objects are registered with the Framework service registry and are notified with a `WireAdminEvent` object when an event is broadcast.

`WireAdminListener` objects can inspect the received `WireAdminEvent` object to determine its type, the `Wire` object with which it is associated, and the Wire Admin service that broadcast the event.

`WireAdminListener` object must be registered with a service property `Constants.WIREADMIN_EVENTS` whose value is a bitwise OR of all the event types the listener is interested in receiving.

For example:

```
Integer mask = new Integer( WIRE_TRACE
  | WIRE_CONNECTED
  | WIRE_DISCONNECTED );
Hashtable ht = new Hashtable();
ht.put( WIREADMIN_EVENTS, mask );
context.registerService( WireAdminListener.class.getName(), this, ht );
```

If a `WireAdminListener` object is registered without a service property `Constants.WIREADMIN_EVENTS`, then the `WireAdminListener` will receive no events.

Security Considerations. Bundles wishing to monitor `WireAdminEvent` objects will require ServicePermission to register a `WireAdminListener` service. Since `WireAdminEvent` objects contain `Wire` objects, care must be taken in assigning permission to register a `WireAdminListener` service.

Version:

$Revision: 1.4 $

Author:

Open Services Gateway Initiative

See Also:

`WireAdminEvent`

Method Summary

void	**`wireAdminEvent`**`(WireAdminEvent event)` Receives notification of a broadcast `WireAdminEvent` object.

All Page Within This Box

Method Detail

3.8.1 wireAdminEvent

`public void` **`wireAdminEvent`**`(WireAdminEvent event)`

Receives notification of a broadcast `WireAdminEvent` object. The event object will be of an event type specified in this `WireAdminListener` service's `Constants.WIREADMIN_EVENTS` service property.

Parameters:

`event` - The `WireAdminEvent` object.

4 Security Considerations

The security of the wiring topology is addressed by the use of ServicePermissions for the various services: Producer, Consumer and WireAdmin, WireAdminListener.

5 Document Support

5.1 References

[1]. Bradner, S., Key words for use in RFCs to Indicate Requirement Levels, RFC2119, March 1997.

5.2 Author's Address

Name BJ Hargrave
Company IBM
Address 11400 Burnet Road, Austin, TX 78758 USA
Voice +1 521 838 9938
e-mail hargrave@us.ibm.com

All Page Within This Box

5.3 Acronyms and Abbreviations

5.4 End of Document

All Page Within This Box