# English (auto-generated) Click • for settings



## CONSTRAINED PROPERTIES

- ► A Bean that has a *constrained* property generates an event when an attempt is made to change its value.
- The event is of type PropertyChangeEvent. It is sent to bjects that previously registered an interest in receiving such tifications.
  - ose other objects have the ability to veto the proposed nge.













# English (auto-generated) Click • for settings





► A Bean that has a *constrained* property generates an event when an attempt is made to change its value.

The event is of type PropertyChangeEvent. It is sent to bjects that previously registered an interest in receiving such tifications.

ose other objects have the ability to veto the proposed nge.

event and it is sent

















## CONSTRAINED PROPERTIES

- ► A Bean that has a *constrained* property generates an event when an attempt is made to change its value.
- The event is of type **PropertyChangeEvent**. It is sent to objects that previously registered an interest in receiving such otifications.

hose other objects have the ability to veto the proposed

other objects already registered an interest to receive the event





















## CONSTRAINED PROPERTIES

- ► A Bean that has a *constrained* property generates an event when an attempt is made to change its value.
- The event is of type PropertyChangeEvent. It is sent to objects that previously registered an interest in receiving such otifications.

hose other objects have the ability to veto the proposed ange.

beam and uh that is the







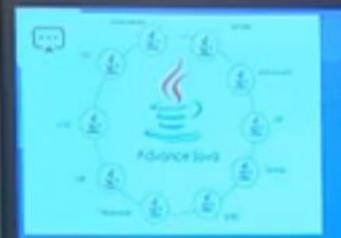














#### PERSISTENCE

- Persistence is the ability to save a Bean to non-volatile storage and retrieve it at a later time. The information that is particularly important are the configuration settings.
  - In the example of both the Colors and TickTock Beans, The rectangular property of the Colors Bean was changed to true, and the interval property of the TickTock Bean was These changes can be saved.

of the property interval property and it has changed its







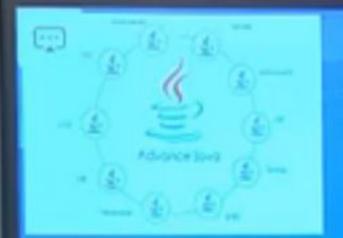














#### **PERSISTENCE**

Persistence is the ability to save a Bean to non-volatile storage and retrieve it at a later time. The information that is particularly important are the configuration retries.

In the example of both the Colors and Tick lock Beans, The rectangular property of the Colors Bean was changed to true, and the interval property of the TickTock Bean was

and its configuration parameters

These changes can be saved.

and we have to supply that file



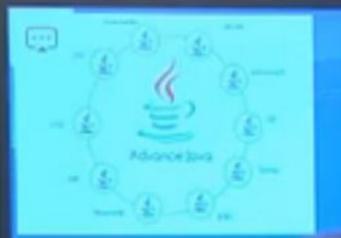














#### PERSISTENCE

- ► Start the BDK again. To restore the application, go to the menu bar of the BeanBox and select File | Load.
- A dialog box should appear, allowing to specify the name of the file from which an application should be restored.
- > Supply the name of the file in which the application was saved, and click the OK button.

specify the name of the v be functioning. file in which the application should be

















#### PERSISTENCE

- ► Start the BDK again. To restore the application, go to the menu bar of the BeanBox and select File | Load.
- A dialog box should appear, allowing to of the file from which an application should be restored.
- > Supply the name of the file in which the application was saved, and click the OK button.

of the colors beam is true and the The application should now be functioning. interval



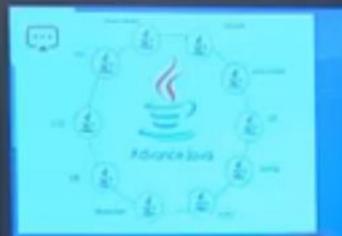














### PERSISTENCE

- ➤ Confirm that the rectangular property of the Colors Bean is true and that the interval property for the TickTock Bean is equal to one second.
- ► The object serialization capabilities provided by the Java class libraries are used to provide persistence for Beans.
- ►If a Bean inherits directly or indirectly from

awt dot component uh proponent, it is automatically serializable, it is automatically serializable it is automatically serializable implements the iava.io.Serializable





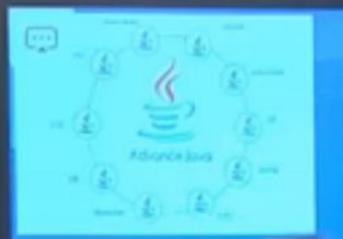














### PERSISTENCE

- Confirm that the rectangular property of the Colors Bean is true and that the interval property for the TickTock Bean is equal to one second.
- ► The object serialization capabilities provided libraries are used to provide persistence for Beans.
  - Bean inherits directly or indirectly from

configuration of the L.Component, it is automatically serializable, parameters configuration class implements the iava.io.Serializable Subtitles/closed captions (c)



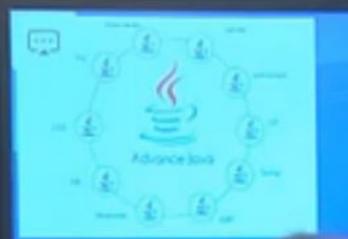














#### **CUSTOMIZERS**

- The Properties window of the BDK allows a developer to modify the properties of a Bean.
- This may not be the best user interface for a complex component with many interrelated properties.
- Therefore, a Bean developer can provide a customizer that helps another developer configure this software.
- A customizer can provide a step-by-step guide through the

on the next customizers must be followed to use the component in a Subtitles/closed captions (c)



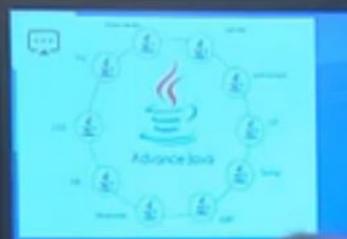














#### **CUSTOMIZERS**

- The Properties window of the BDK allows a developer to modify the properties of a Bean.
- This may not be the best user interfac complex component with many interrelated properties
- Therefore, a Bean developer can provide a customizer that helps another developer configure this software.

these complex components contains the p-by-step guide through the interrelated process that must be followed to use the component in a Subtitles/closed captions (c)



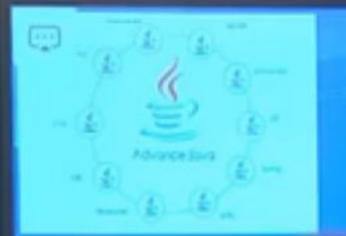














#### **CUSTOMIZERS**

- ► The Properties window of the BDK allows a developer to modify the properties of a Bean.
- This may not be the best user interface for a complex component with many interrelated properties.
- ► Therefore, a Bean developer can provide a <u>customizer</u> that helps another developer configure this software.

that is the step-by-step process to use the component in a particular wed to use the component in a Subtitles/closed captions (c)















#### **CUSTOMIZERS**

- The Properties window of the BDK allows a developer to modify the properties of a Bean.
  - This may not be the best user interfac complex component with many interrelated properties
- Therefore, a Bean developer can provide a customizer that helps another developer configure this software.

customizer is the interface so rovide a step-by-step guide through the it it provides the ess that must be followed to use the component in a Subtitles/closed captions (c)

















#### **CUSTOMIZERS**

- ➤ Since builder tools recognize any customizer that implements the Customizer interface, it can be launched from any builder tool that has the bean installed.
- ► Customizer Interface: The Customizer interface is simple and has only three methods.
- ▶ setObject(Object bean): The builder tool calls this method

and next application builder tool instance to the customizer. It is called can recognize by once, which is before the builder tool launches the Subtitles/closed captions (c)











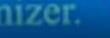


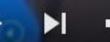


#### **CUSTOMIZERS**

- ▶ Since builder tools recognize any customizer that implements the Customizer interface, it can be launched from any builder tool that has the bean installed.
- Customizer Interface: The Customizer and has only three methods.
- ▶ setObject(Object bean): The builder tool calls this method

customizer contains e target bean instance to the customizer. It is called the contains this set object method are the builder tool launches the Subtitles/closed captions (c)



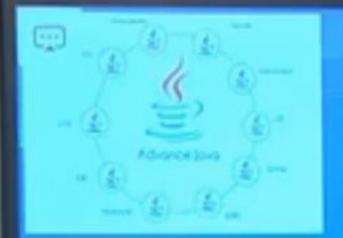














#### **CUSTOMIZERS**

addPropertyChangeListener(PropertyChangeListener

listener): This method adds a propertyChange event listener.

By registering as a listener, the target object(s) can receive

notification and the modified value of the property. When the

rget object receives a propertyChange event, it retrieves the

tter Method object and the property value from the

those objects are already registered to Using this Method object and receive

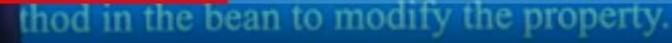
property value, the target object then invokes the setter

Subtitles/closed captions (c)





6:05 / 31:25

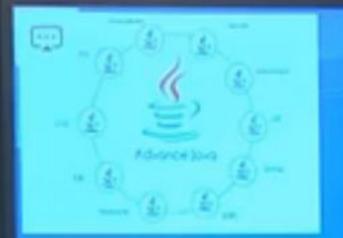














#### **CUSTOMIZERS**

addPropertyChangeListener(PropertyChangeListener

listener): This method adds a propertyChange event listener.

By registering as a listener, the target object(s)

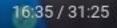
notification and the modified value of the property. When the

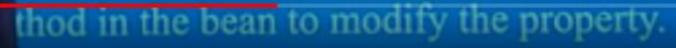
rget object receives a propertyChange event, it retrieves the

tter Method object and the property value from the

object. Using this Method object and mainly which is used to set the set the property property value, the target object then invokes the setter





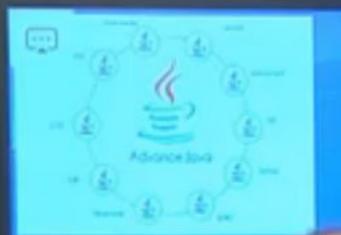














#### **CUSTOMIZERS**

- removePropertyChangeListener (PropertyChangeListener listener): This removes a property change event listener.
- A customizer class has to meet the following requirements to enable builder tools to launch it:
- It must implement the Customizer interface: This is to enable the builder tool to recognize the customizer and to

remove remove property change listener can be passed on to the is to be

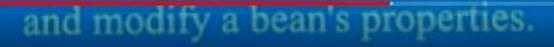
customizer. Using this bean instance, the customizer can fetch

Subtitles/closed captions (c)



18:05 / 31:25



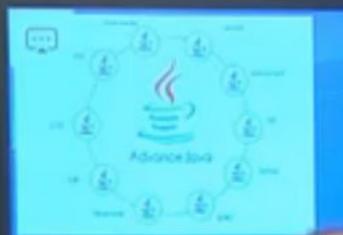














#### **CUSTOMIZERS**

- ► removePropertyChangeListener (PropertyChangeListener listener): This removes a property change event listener.
- A customizer class has to meet the following requirements to enable builder tools to launch it:
- ► It must implement the Customizer interface: This is to enable the builder tool to recognize the customizer and to obtain the bean instance, which can be passed on to the

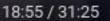
and the next one mizer. Using this bean instance, the customizer can fetch

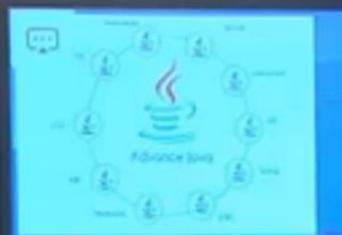












to launch the customizer ters.



#### **CUSTOMIZERS**

- It has to be a subclass (direct or derived) of the java.awt.Component class. This is to enable builder tools to embed the customizer in a dialog box.
- ► It must have a constructor with no arguments. This is to enable the builder tool to launch it. If constructors with arguments are allowed, builder tools wouldn't know which builder tool constructor to call (in case there are many) or what values to





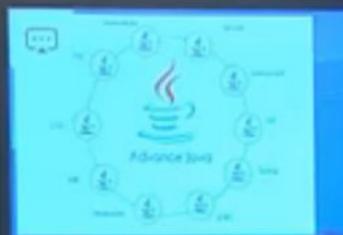














#### **CUSTOMIZERS**

- It has to be a subclass (direct or derived) of the java.awt.Component class. This is to enable builder tools to embed the customizer in a dialog box.
- enable the builder tool to launch it. If constructors with arguments are allowed, builder tools wouldn't know which

interfaces first one is applet initializer as parameters.



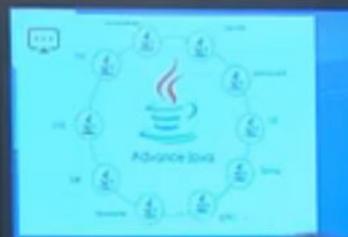














## THE JAVA BEANS API

The Java Beans provides a set of classes and interfaces in the java.beans package.

The state of the s			
	INTERFACE	DESCRIPTION	
	AppletInitializer	Methods in this interface are used to initialize beans that are also applets.	
	BeanInfo	This interface allows a designer to specify information about the properties, events and methods of a bean.	
	Customizer	This interface allows a designer to provide a	
uh which	allows to provid	e a graphical interface through which a bean	
user inter	face	may be configured.	



Methods in this interface



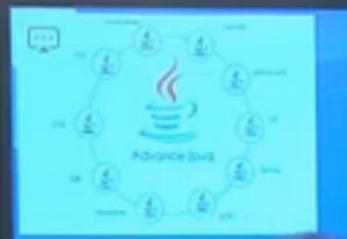


22:05 / 31:25











### THE JAVA BEANS API

► The Java Beans provides a set of classes and interfaces in the java.beans package.

INTERFACE	DESCRIPTION	
AppletInitializer	Methods in this interface are been initialize beans that are also applets.	
BeanInfo	This interface allows a designer to specify information about the properties, events and methods of a bean.	
Customizer	This interface allows a designer to provide a graphical user interface through which a bean	
stener are invoked ay be configured.  Subtitles/closed captions (c)		



property

change I

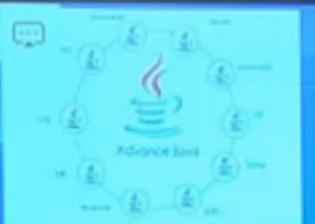
Methods in this interface determin







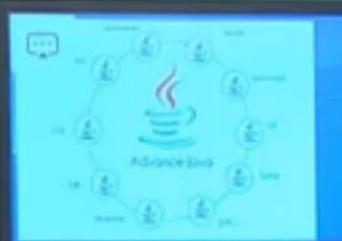






INTIDIRE	ACE DESCRIPTION
ExceptionList	ener A method in this interface is invoked when an exception has occurred.
PropertyChang	geListener A method in this interface is invoked when a bound property is changed.
Property Editor	Objects that implement this interface allow designers to change and display property values.
	geListener A method in this interface is invoked when a constraint property is changed.
will change listener so with change	Methods in this interface allow a bean to execu Subtitles/closed captions (c) nments

24:05 / 31:25

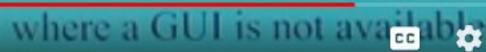


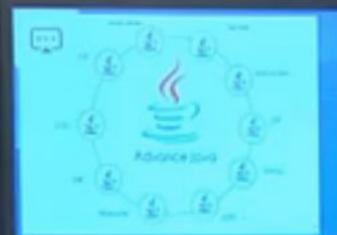


	INTERFACE	DESCRIPTION
	ExceptionListener	A method in this interface is invoked when an exception has occurred.
Co-y	PropertyChangeListener	A method in this interface is invoked when a bound property is changed.
	PropertyEditor	Objects that implement this interface allow designers to change and display property values.
	VetoableChangeListener	A method in this interface is invoked when a constraint property is
class prov	ides the information al	Dout Dout Dout Dout Dout Dout Dout Dout



25:05 / 31:25







#### THE JAVA BEANS API

The following table lists the classes in java.beans.

CLASS	DESCRIPTION	
EventHandler	Supports dynamic event listener creation.	
EventSetDescriptor	Instances of this class describe an event that can be generated by a Bean.	
Expression	Encapsulates a call to a method that returns a result.	

handler which supports the dynamic





















#### THE JAVA BEANS API

► The following table lists the classes in java.beans.

CLASS	DESCRIPTION
EventHandler	Supports dynamic cvent listener creation.
EventSetDescriptor	Instances of this class describe an event that can be generated by a Bean.
Expression	Encapsulates a call to a method that returns a result.

events and descriptor under major descriptor classes







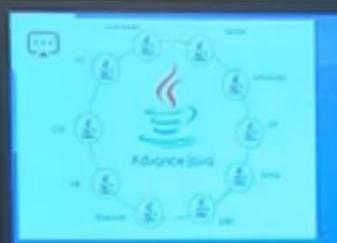














### THE JAVA BEANS API

DESCRIPTION CLASS This class analyses a bean and introspector constructs a BeanInfo object that describes the component. Instances of this class describe a MethodDescriptor method of a Bean. Instances of this class describe a meterDescriptor method parameter Handles the state information of an enceDelegate of a java beam object. and a parameter





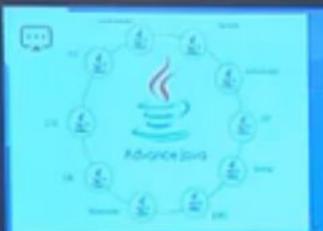














# THE JAVA BEANS API

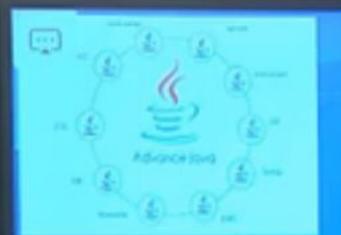
CLASS	DESCRIPTION
introspector	This class analyses a bean and constructs a BeanInfo object that describes the component.
MethodDescriptor	Instances of this 10 seconds describe a method of a Bean.
meterDescriptor	Instances of this class describe a method parameter
reason interface	Handles the state information of an object.
implements whenever the bound	d property  Subtitles/closed captions (c)





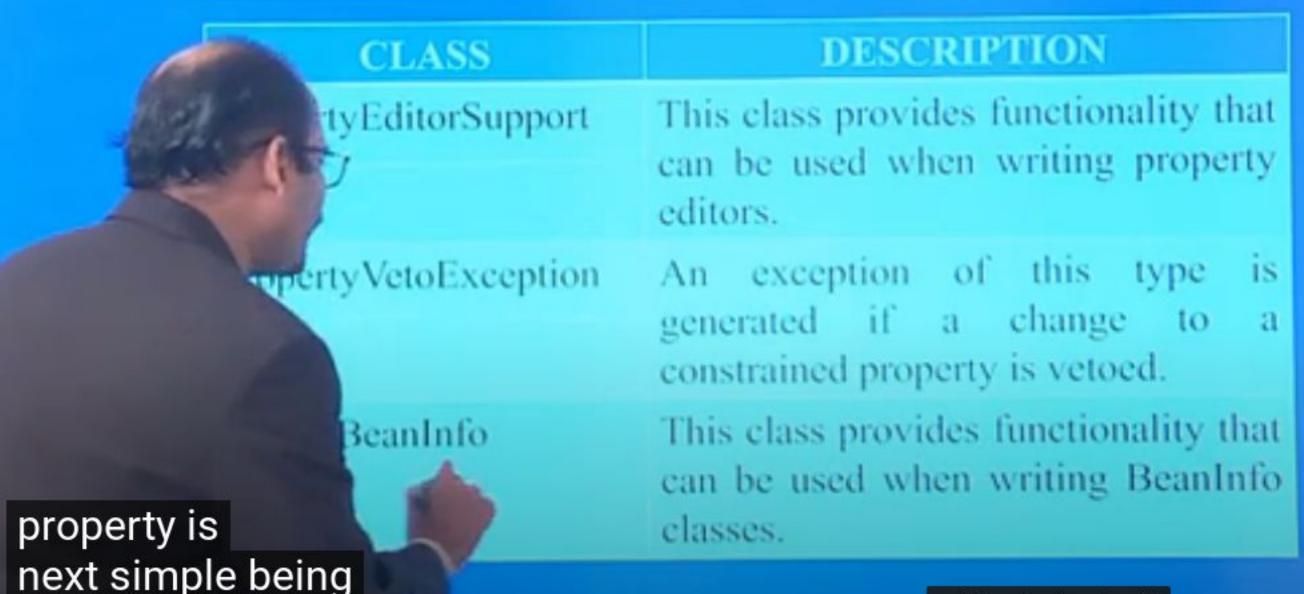








#### THE JAVA BEANS API







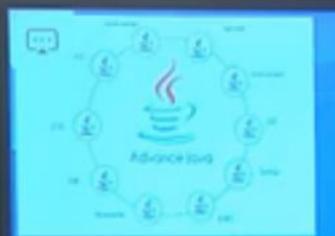














#### THE JAVA BEANS API

