

Deploying the UI and the UI Service

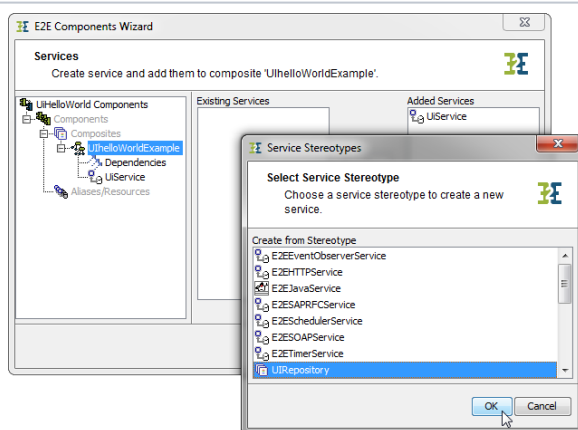
To be able to view the UI with the web browser, the service as well as the UI components need to be deployed and running on the Bridge. The next steps will introduce how the UI components will be added to the component diagram and how the UI can be viewed within the Bridge and web browser.

The xUML UI service can be deployed without any functionality implemented. This could be the case in e.g. early UI development stages where the different screens need to be reviewed. To be able to deploy the service you at least need to have a SOAP Port Type defined.

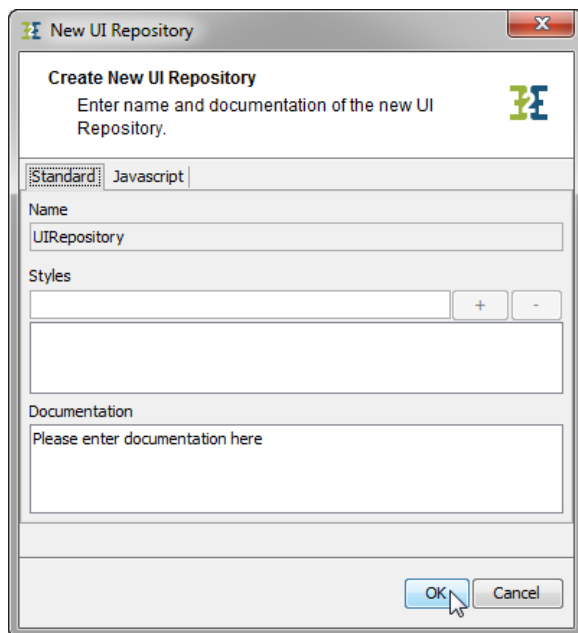


Running the UI Application

Draw the component diagram by the help of the xUML Components Wizard (🔧). For detailed help on how to use the Components Wizard refer to the [Builder User's Guide](#).

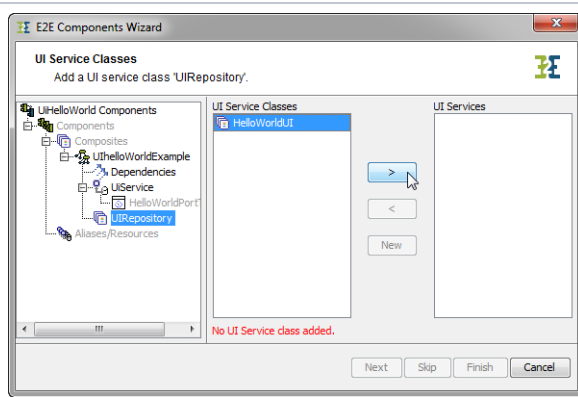


Define a composite and, in addition to the SOAP service, add a UIRepository to it.



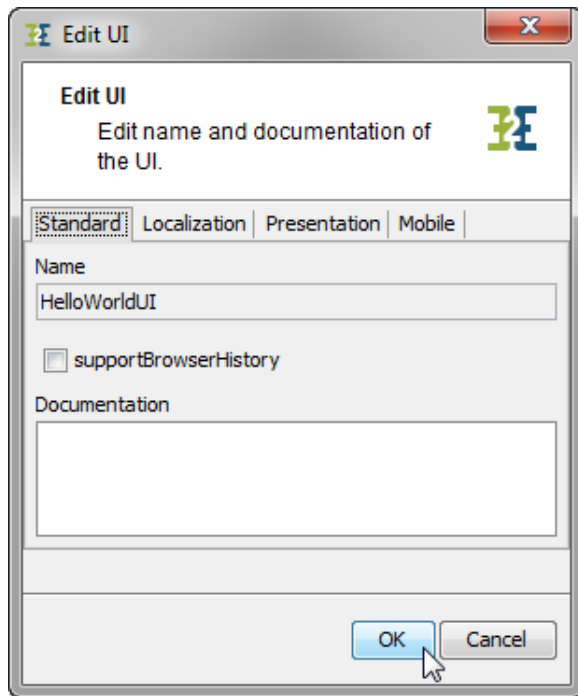
A dialog pops up where you can adjust some settings of the UI repository.

Just click **OK**.



Clicking **Next** several times will bring you to the point that the wizard complains about a missing UI service class.

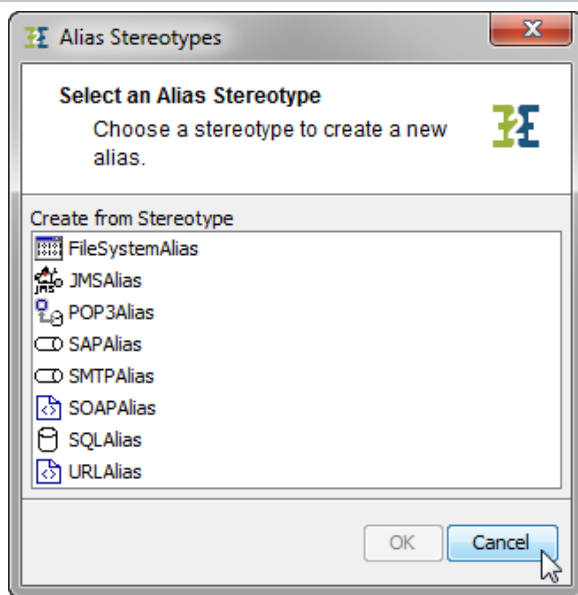
Add the UI Service class **HelloWorldUI** to the UI Services list by selecting it from the left classes list and clicking on **>**.



Within this window, some tagged values can be amended which may influence the appearance of the user interface itself. See the table below for a detailed description of all tagged values.

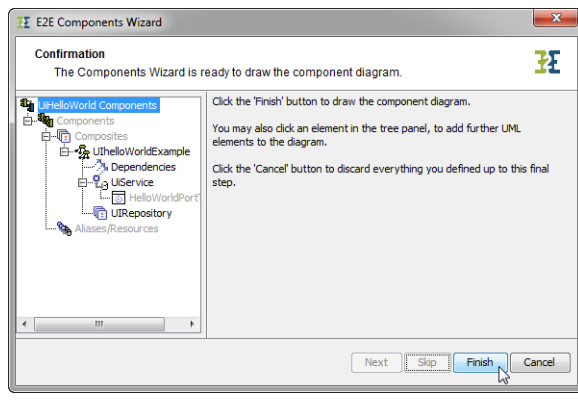
Tagged Value	Description	Allowed Values	
supportBrowserHistory	If true, each UI state is mapped to an anchor. That is, if the browser's back button is pressed, the UI goes back in time. However, if the UI is embedded in another framework (CMS, SharePoint, ...), it is recommended to set this flag to false.	true	Enable browser history.
		false	Disable browser history (default).
Localization			
defaultTheme	Themes are sets of style sheets to define a look and feel of the Ajax application. Choose here one of the visual themes of the selection that can apply to the UI. Details about the theming concept can be found at http://jqueryui.com/themeroller .	None	No theme applied.
		A value from the list of themes.	
defaultLanguage	The language used in the UI templates should be the same as used to configure the UI widgets (tables, calendars, ...) By default, the Bridge takes the operation system language as UI language. However, this default language can be overridden using this tagged value.	None	Use system language.
		A language of the list of languages.	

defaultRegion	The region is used in the UI to format numbers and dates. By default, the Bridge takes the operation system region as UI region. However, this default region can be overridden using this tagged value.	None	Use system region.
		A region of the list of regions.	
Mobile			
platform		None	
		default	
		mobile	
Presentation			
title	Specify here the title of the UI, e.g. HTML page title		
defaultLayout	Specify here in which way the template elements should be organized.	floating	All template elements are floating.
		grid	All template elements get a tabled grid layout.
isConsoleUI	A UI application can be used within the Bridge as an application to e.g. configure a service. If this flag is set, the Bridge will embed this user interface. <div>Only one UI per xUML service is allowed.</div>	true	Embed this UI in the Bridge.
		false	Do not embed this UI (default).
consoleUIName	If isConsoleUI is set to true, this name is taken as tab name of the Bridge tab the UI is embedded to.		
errorPlacement	If referring to a JavaScript operation, errorPlacement overrides the default error placement when validating forms. The operation has to parameters: <ul style="list-style-type: none">error: The jQuery error element to be rendered.element: The jQuery form element causing the error. The default operation looks like: <pre>var left = element.position().left; var top = element.position().top + element. height() + 5; error.css({ "position": "absolute", "left": left + "px", "top": top + "px"}); error.insertAfter(element);</pre>		



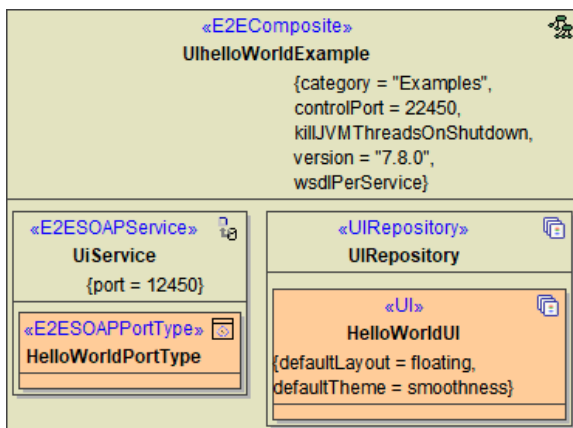
You don't need any aliases for this tutorial service.

Click **Cancel** and then **Next**.

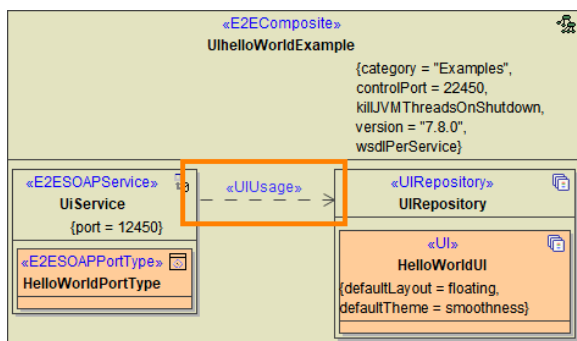


Click **Finish** to finish the setup of the Component Diagram.

The wizard draws a component diagram like the one below:



The service composite contains a SOAP service and a UI repository. The UI repository will connect automatically with the SOAP service but it is deprecated to use this default behavior. As an xUML service may contain different interfaces (multiple SOAP services and or HTTP or REST interface), it is best practice to connect the service interface to the UI it uses via a **<<UIUsage>>** dependency as shown below.



If you do not add this dependency, the xUML Compiler will show a warning.