

<b>Slidename</b>	<b>Text</b>
Welcome	Welcome to the Creating a new model viewlet of the Getting Started Guide for IBM Rational Rhapsody.
Slide 3	This viewlet uses some of the content of the Rhapsody C++ Tutorial, but these steps are valid for any of the Rhapsody editions. You can read the tutorial by clicking the Rhapsody Help menu &gt; List of Books, then in the document that opens, below the heading Getting Started, find the link called Rhapsody in C++ Tutorial.
Slide 4	For this viewlet we have created a directory called C:\Rhapsody specifically to hold the new project we create.
Slide 5	Let's create a new project called Handset.
Slide 5	Click the New Project button
Slide 6	Type &quot;Handset&quot; here
Slide 7	We need to change the folder the project will be stored in, to use the folder C:\Rhapsody that we created for this viewlet.
Slide 7	Click here
Slide 7	Drag to here
Slide 8	Type &quot;Rhapsody&quot; here
Slide 9	Click OK button
Slide 10	Save the project.
Slide 10	Click Save button
Slide 11	A new project is always created with a package called Default that we want to rename to AnalysisPkg, because it will hold the first part of our analysis of the problem.
Slide 11	Click the + to expand Packages
Slide 12	Right click the Default package
Slide 13	Click Properties
Slide 14	Type &quot;AnalysisPkg&quot; here
Slide 15	Click OK
Slide 16	Now we want to create a use case diagram in the AnalysisPkg.
Slide 16	Right click AnalysisPkg
Slide 17	Click Add New
Slide 18	Click Diagrams
Slide 19	Click Use Case diagram
Slide 20	Type &quot;Functional Overview&quot; here
Slide 21	Click OK button
Slide 22	The use case diagram has been added to the Model View, and the diagram is open. Let's make the drawing toolbar visible so that we can add some symbols to the diagram.
Slide 22	Click View menu
Slide 23	Click Toolbars
Slide 24	Click Drawing
Slide 25	The first thing to put on the diagram is a Boundary Box to

	represent the border of the system.To draw this, select the symbol, and then click and drag to show the area that the symbol must cover.
Slide 25	Click Boundary Box
Slide 26	Click here
Slide 26	Drag to here
Slide 27	Rename the symbol to “Handset protocol system”
Slide 27	Type “Handset Protocol System” here
Slide 28	To make the symbol a bit smaller click here
Slide 29	Drag to here
Slide 30	Let's add the first Actor to represent the Human-Machine Interface of the handset.
Slide 30	Click Actor
Slide 31	Click here
Slide 32	Type “MMI” here
Slide 33	Now let's add the second Actor to represent the network.
Slide 33	Click Actor
Slide 34	Click here
Slide 35	Type “Network” here
Slide 36	Click here
Slide 37	Drag to here
Slide 38	Now let's add the four use cases, naming them as we go...
Slide 38	Click Use Case
Slide 39	Click here
Slide 40	Type “Place Call” here
Slide 41	Click Use Case
Slide 42	Click here
Slide 43	Type “Supplementary Service” here
Slide 45	Click Use Case
Slide 46	Click here
Slide 47	Type “Receive Call” here
Slide 48	Click Use Case
Slide 49	Click here
Slide 50	Type “Provide Status” here
Slide 51	Next we'll give the Place Call use case a description. In a real project you must do this for all the use cases.We'll open thefeatures dialog fromthe model, but you can also open the features dialog by double-clicking the symbol.
Slide 51	Click here
Slide 52	Right click Place Call
Slide 53	Click Features
Slide 54	Click Description
Slide 55	Type “General function of the system is that it must be able to place various types of calls.” here
Slide 56	Click OK

Slide 57	Next we'll associate the Actors with the appropriate use cases.
Slide 57	Click Association
Slide 58	Click on actor MMI
Slide 59	Click on use case Place Call
Slide 60	Click Association
Slide 61	Click on actor MMI
Slide 62	Click on use case Receive Call
Slide 63	Click Association
Slide 64	Click on actor Network
Slide 65	Click on use case Receive Call
Slide 66	Click Association
Slide 67	Click on actor Network
Slide 68	Click on use case Provide Status
Slide 69	In this example the Supplementary Service use case is a specialisation of both the Place Call and the Receive Call use cases. We show this by creating generalization lines.
Slide 69	Click Generalization
Slide 69	Click on use case Supplementary Service
Slide 70	Click on use case Place Call
Slide 71	Click Generalization
Slide 72	Click on use case Supplementary Service
Slide 73	Click on use case Receive Call
Slide 74	Finally, we want to add a comment on the diagram to give some information to the reader.
Slide 74	Click Common
Slide 75	Click Comment
Slide 76	Click here
Slide 77	Type "This is a mock-up solution of a generic protocol system" here
Slide 78	Click here
Slide 78	Drag to here
Slide 79	Click here
Slide 80	This use case diagram is now complete. You can create any of the other UML diagrams using exactly the same techniques as you have seen here.
Conclusion	You have completed this viewlet! We hope that it has given you an insight into how to get started with IBM Rational Rhapsody. This example has shown you how to create a new project, rename the Default package to a more meaningful name, and create a use case diagram showing the relationships between use cases and actors. In addition, you have seen how to add a description on one use case and how to put a comment onto the diagram to explain its purpose. The same techniques can be used to create the other

	<p>UML diagrams. The content used in this viewlet is from the Rhapsody C++ Tutorial, but the same sequence can be used with any of the Rhapsody editions. For more information on Rhapsody, Rational software, and IBM please see the resources and links on the next slide.</p>
--	--