Telelogic Publishing Engine Examples Guide Release 1.0 Before using this information, be sure to read the general information under Appendix, "Notices" on page 53.

This edition applies to **VERSION 1.0, Telelogic Publishing Engine** and to all subsequent releases and modifications until otherwise indicated in new editions.

# © Copyright IBM Corporation 2008

US Government Users Restricted Rights—Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

# Table of contents

# **Telelogic Publishing Engine**

xamples	. 1
A template for DOORS data	. 1
Basic setup	. 1
Advanced template options	12
More advanced settings	
Generate the document	25
A template for Tau data	30
Define the template content	
Generate the document	48
Appendix A: Notices	53

ademarks
----------

1

*iv* Telelogic Publishing Engine Examples Guide

# **Telelogic Publishing Engine**

# **Examples**

# A template for DOORS data

### **Basic setup**

Create a new Document Template

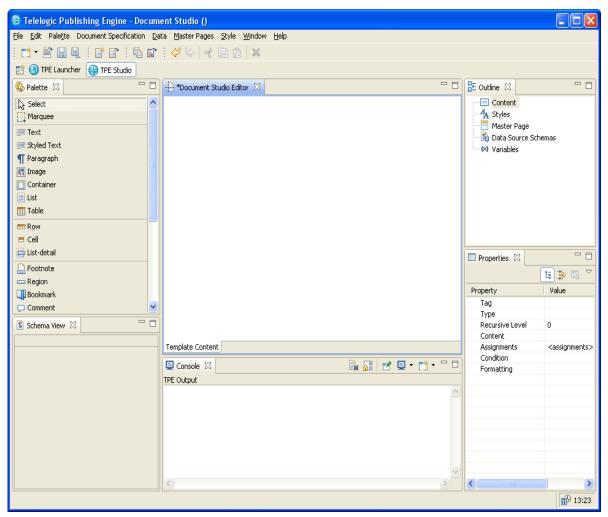


Figure 1

Telelogic Publishing Engine Examples Guide 1

Add a data source schema using the Add Data Source Wizard

🖶 Add Data Source Wizard	×
Add Data Source Wizard Welcome to the Add Data Source Wizard	
This wizard allows you to add a new Data Source Schema to the currently open template.	
< <u>B</u> ack <u>Mext</u> Einish Cancel	

Figure 2

2 Telelogic Publishing Engine Examples Guide

Select the schema provided in the TPE installation

🖶 Add Data So	purce Wizard	
	Data Source Schema urce Schema file, its type and an identifier	
Schema File:	Publishing Engine\source\DOORS\schema\simpleDOORSSchema.xsc	Browse
Schema Type:	DOORS 💌	
Data Source ID:	D51	
	< <u>B</u> ack Next > Einish	Cancel

### **Review selection**

🖀 Add Data Source Wizard	×
Summary page Review your selections before pressing Finish.	
Schema File: C:\Program Files\Telelogic\Telelogic Publishing Engine\source\DOORS\schema\sim Data Source Type: DOORS Data Source ID: DS1	
< <u>B</u> ack <u>N</u> ext > <u>Finish</u> Cancel	]

Figure 4

4 Telelogic Publishing Engine Examples Guide

Add a container element in the template

🕆 Telelogic Publishing Engine - Docu	ment Studio ()	
File Edit Palette Document Specification	<u>D</u> ata <u>M</u> aster Pages <u>S</u> tyle <u>Wi</u> ndow <u>H</u> elp	
📬 🕶 🔛 🔛 🗎 😭 🖬 🗎 🔍 🛛	L 🖬 🗄 🖑 🐃 👘 💼 🛛 🗶	
😭 🕒 TPE Launcher   🌐 TPE Studio		
🗞 Palette 🛛 🗖 🗖	🕀 *Document Studio Editor 🛛 🗖 🗖	🗄 Outline 🛛 🗖 🗖
💫 Select 🔼	III	
[]] Marquee	Container	
📰 Text		A Styles     Master Pages
📰 Styled Text		
¶ Paragraph		(X) Variables
🕅 Image		
Container		
📃 List		
Table		
I Row		
E Cell		
S Schema View 🛛 🗸 🖾 🗖 🗖		
D51		Properties 🛛 🗌 🗖
🖃 💯 Module (DOORSModule) 🗾 🔼		[≒] 🐉 💀 🎽
So Description (string)		Property Value
So <i>Name (string)</i> ⊡SP Attribute (DOORSAttribute)		Tag container
		Туре
🖃 💯 Object (DOORSObject)		Recursive Le 0 Content
- 🜆 Object Identifier (string, 🗏	Template Content	Assignments <assignments></assignments>
	🖳 Console 💥 📑 🔁 👘 🕆 🖓 🗖	Condition
	Telelogic Publishing Engine	Formatting
Sa Object Heading (string) Sa Object Text (string)	<u>^</u>	
Attribute (DOORSAttribu		
🗉 👂 Column (DOORSColumn)		
Tabla (DOODSTabla) ×		
	13:40	

Figure 5

#### Assign a query to the element

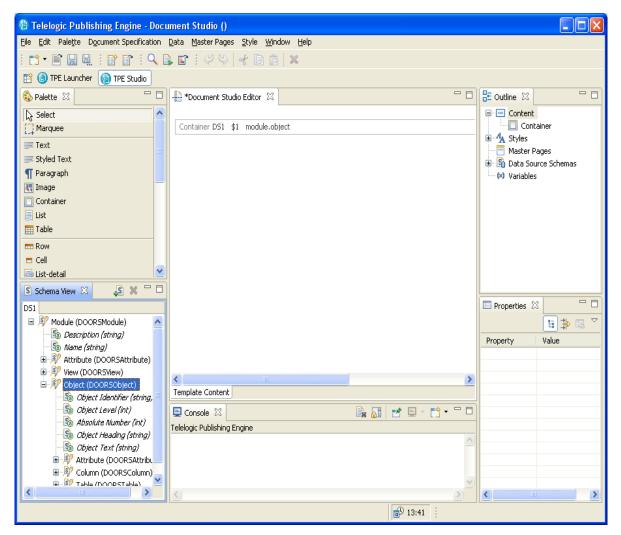


Figure 6

Add the "Object Heading" attribute

🖶 Telelogic Publishing Engine - Docu		
File Edit Palette Document Specification		
📬 • 🖺 🖩 🖳   📴 🔐   🔍 🛛		
😭 🕓 TPE Launcher 🛛 🛞 TPE Studio		
🚯 Palette 🛛 🗖 🗖	🗄 *Document Studio Editor 🛛	E Outline 🛛 🗖 🗖
Rest         Image	Container DS1 \$1 module.object Text module.object.Object Heading	Content Container Container Text - module.object. A Styles Master Pages Data Source Schemas Variables
Cell     List-detail     Schema View      S     Schema View      S     Schema View      D51     S     Produle (DOORSModule)     S     Description (string)		Properties X  Property Value
So Name (string) Solution: Sol	Template Content	
So Object Level (int) So Absolute Number (int) So Object Heading (string) So Object Text (string) So	E Console X E Cons	
	<u>د</u> ۱3:42	

#### Figure 7

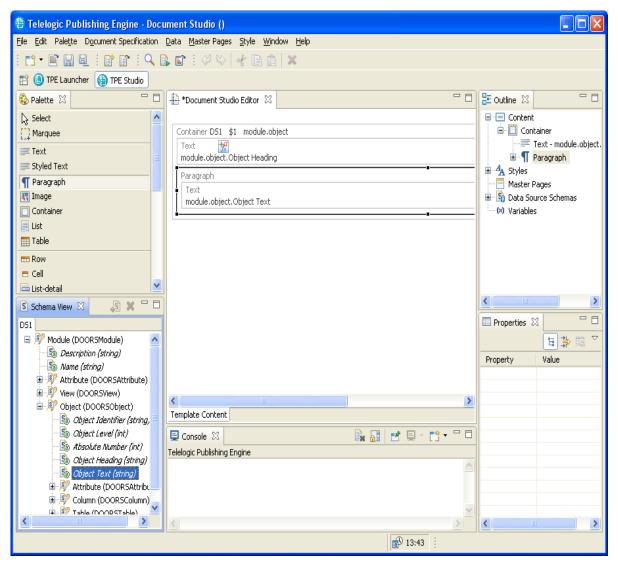
Apply a Condition to the display of the element

The text element containing the object heading should be processed only if the Object Heading is not empty.

🚇 Set condition for current element 🛛 🛛 🗙		
Script expression		
Variables Attributes S1 Object_Headir Object_Text id level		
	Test	Ok Cancel

#### Add the Object text

The Object Text is added in its own paragraph to avoid collating multiple object texts on the same line.





Condition the display of Object Text

🖶 Set condition for curr	ent element	
Script expression		
Variables Attributes Standard Absolute_Num Object_Headir Object_Text id Ievel	Object_Text != ""	×
	<	2
	Test	Ok Cancel

Save the template.

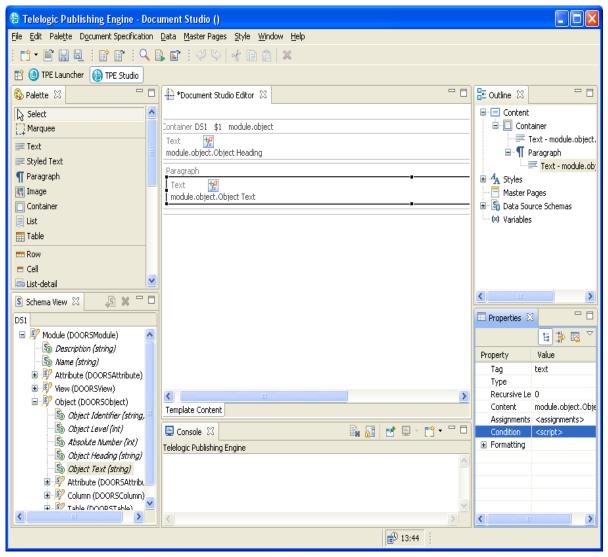


Figure 11

#### Advanced template options

Add a Table of contents

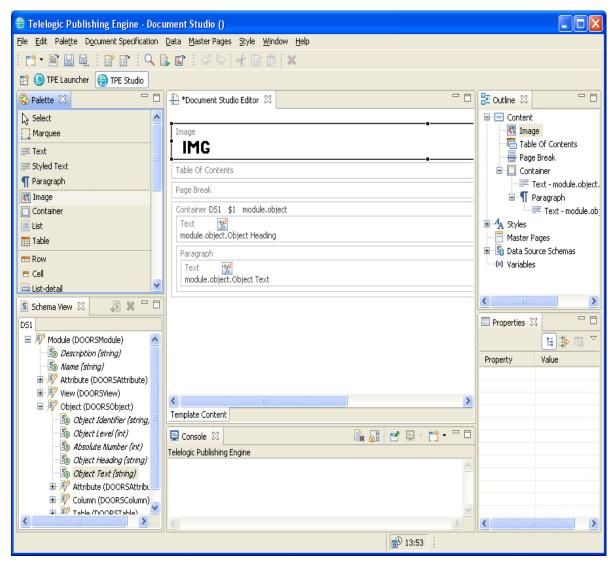
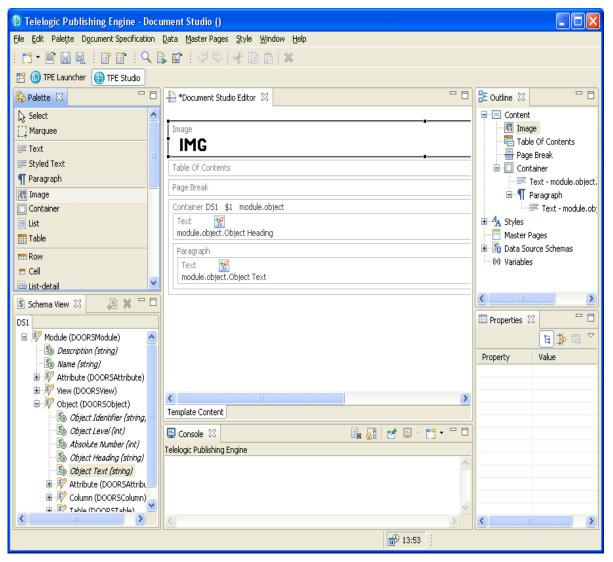


Figure 12

12 Telelogic Publishing Engine Examples Guide

Add an image at the top of the document





#### Load an image from the file system

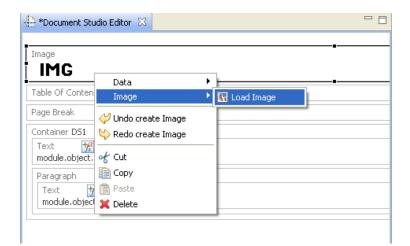


Figure 14

14 Telelogic Publishing Engine Examples Guide

NOTE The selected image is copied in the template.

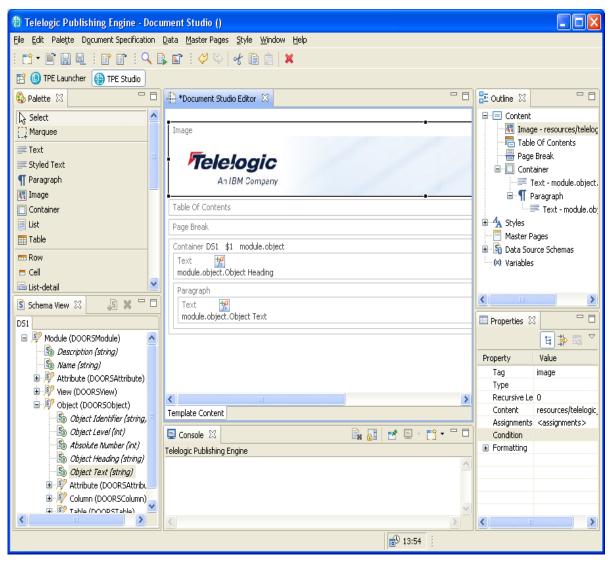
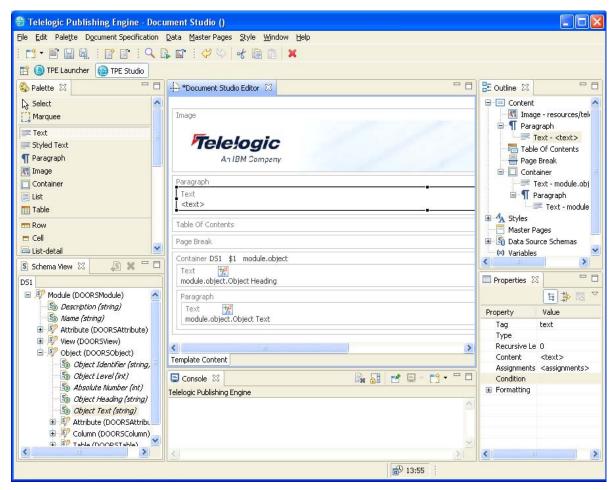


Figure 15

Add a title to the document

Add a paragraph with a text element



#### Set the text

Double click the text element or click the "Content" in the property page.

A Set content for current element	×
Simple value Data expression Script expression	
TPE - DOORS Data	
<u>&lt;</u>	
	Ok Cancel

#### Figure 17

Define a title style

The newly created style is named "TPE\_Title"

🖶 Create a New Style		
Set property values Edit the selected properties with approp	riate values	
Property Name	Property Value true 0000F0 16	
	Back Next >	Einish Cancel

#### Set the style to the paragraph

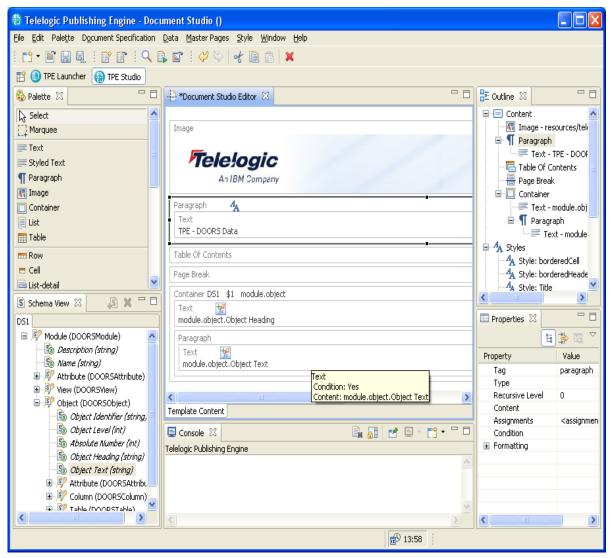


Figure 19

Create a master page for the data

File Edit Palette Dogoment Specification Data Master Pages Style Window Help	🕆 Telelogic Publishing Engine - Doc	iment Studio ()	
Platte Werk CooksAttruke Schema View X Object Level (nt) Schema View X Object	File Edit Palette Document Specification	<u>D</u> ata <u>M</u> aster Pages <u>S</u> tyle <u>W</u> indow <u>H</u> elp	
Palette 23 Palette 23 Palette 23 Palette 23 Palette 23 Palette 23 Palette 24 Palette 24 Palette 25 Palette	📬 • 🛍 🔛 🖳   😭 🔐   🔍	🕽 🖬 🗄 🖑 😂   💰 🗈 👔   🗙	
Select   Marquee   Text   Syled Text   Syled Text   Text and de   A Syle bodered lead   B Decores Stribute   B Decores Stribute <tr< td=""><td>😭 🔘 TPE Launcher   🌐 TPE Studio</td><td></td><td></td></tr<>	😭 🔘 TPE Launcher   🌐 TPE Studio		
Marquee     Text     Styled Text     Footer       <	🚯 Palette 🛛 🗖 🗖	+ *Document Studio Editor 🛛	🗄 Outline 🛛 🗖 🗖
Itext   Text   Styled Text   Paragraph   Style Text   Paragraph   Image   Container   List   Itable   Row   Cell   List-detail   Sthem View   Sthem View <t< td=""><td>💫 Select 🔼</td><td></td><td></td></t<>	💫 Select 🔼		
Footer   Styled Text   Faragraph   Container   List   Table   Container   List   Boschem Wew X   Schem Keing)   Schem View (DOORSModule)   Schem View (DOORSSModule)   Schem Vi	[]] Marquee	Header	
Syled Text   Paragraph   Image   Container   List   IT Table   Row   C Cell   Schema View 23   Schema View 23   Made (DOORSModule)   Schema View 23   Module (DOORSModule)   Schema View 23   Module (DOORSModule)   Schema View 23   Schema View 24   Schema View 25   Schema View 26   Schema View 27   Schema View 28   Schema View 29   Schema View 2000 Sc		Footer	
Image       A Syle: Idebitatic         Container       Syle: Idebitatic         Itable       A Syle: Idebitatic         Table       A Syle: Idebitatic         Row       Cell         Cell       Master pages         Schema View       Image         Schema View       Image         Schema View       Image         Schema View       Image         Master pages       Image         Master pages       Image         Itable       Master pages         Itable       Image         Schema View       Image         Schobiet Lowof Note       Image     <	📰 Styled Text		
Container Lust Table Row Cell Lust-detail S Schema View X S Schema View X S Schema View X S Schema View X Made (DOORSModule) S Schema View X Made (DOORSModule) Made (DOORSModule) S Schema View X Made (String) Module (DOORSModule) S Schema View X Mane (String) S Object Identifier (string) S Object Text (string			🐴 Style: Title 👘
Ust   Table   Row   Cell   Uist-detail   S Schema View X   S Schema View S			
Table   Table   Row   Cell   List-detail   S Schema View X   S Schema View X   DS1   Module (DOORSModule)   B Description (string)   A Mame (string)   Wiew (DOORSNiew)   Diget Level (nt)   Cobject Level (nt)   Cobject Tevel (nt)			
Image: Name   Interview   Interview <t< td=""><td></td><td></td><td></td></t<>			
Row   Cell   Bit List-detail     S Schema View     S Schema View     D51     Properties     Properties     Properties     Property     Value     Name     Property     Value     Value <t< td=""><td></td><td></td><td></td></t<>			
List-detail     S Schema View X        DS1        DS1        Properties X           Property        Property        Property        Property        Property        Property           Property           Property                 DS1 </td <td></td> <td></td> <td></td>			
Schema View X   Schema View X Solution (string) Solution (DOORSAttribute) Solution (string) </td <td></td> <td></td> <td></td>			
Schema view     DS1     DS1     DS1     DS1     DS1     DS1        DS1        DS1        DS1        DS1              DS1 <b>Properties &amp; Property Value Name MPData Property Value Name MPData Property Value Name MPData Property Colonary Property Colonary Property Value Prope</b>			
Image: String	S Schema View 🛛 👃 🗙 🗖 🗆		
Sexription (string)   Mame (string)   Attribute (DOORSAttribute)   Property   View (DOORSView)   Object (DOORSObject)   Sobject Identifier (string)   Object Level (int)   Object Heading (string)   Object Text (string)   Object Text (string)   Property   View (DOORSColumn)   Object Text (string)   Object Text (string)   Object Text (string)   View (DOORSColumn)			
Mame (string)   Attribute (DOORSAttribute)   Property   View (DOORSView)   Object (DOORSObject)   So Object Identifier (string)   So Object Level (int)   So Object Heading (string)   Object Text (string)   So Object Text (string)			🗄 🍄 🖾 🎽
Image: Provide and the second sec			Property Value
Wiew (DOORSView)     Wew (DOORSView)     Object (DOORSObject)     Gobject Identifier (string)     Gobject Level (int)     Gobject Level (int)     Gobject Heading (string)     Gobject Text (string			
Image: Polyact (DOORSObject)     Template Content Master page: MPData       Image: Polyact Identifier (string)     Image: Polyact Identifier (string)       Image: Polyact Identifier (string)<			E Format
Cobject Identifier (string)       Cobject Level (int)       Cobject Level (int)       Cobject Heading (string)       Cobject Text (string)       Cobject Text (string)       Column (DOORSColumn)       Column (DOORSColumn)       Column (DOORSColumn)			
Absolute Number (int)     Telelogic Publishing Engine       Object Text (string)     Object Text (string)       P     Attribute (DOORSAttribu       P     Column (DOORSColumn)       P     Table (DOORSTable)			
Image: Solution (DOORSAttribute)     Image: Solution (DOORSColumn)       Image: Solution (DOORSColumn)     Image: Solution (DOORSColumn)       Image: Solution (DOORSColumn)     Image: Solution (DOORSColumn)       Image: Solution (DOORSColumn)     Image: Solution (DOORSColumn)			
Solution     Solution       Solution     Solution       Solution     Solution       Solution     Solution       Solution     Solution       Solution     Solution			
Column (DOORSColumn)		A	
	🗉 👰 Attribute (DOORSAttribu		
	🗄 🖓 Column (DOORSColumn)		l
	HIMPY Table (DOOUSTable) 💳	< ×	<
13:58	<u></u>	<b>B</b> 13:58	



Add some static text in the page's header

🕆 Telelogic Publishing Engine - Docu	ment Studio ()		
File Edit Palette Document Specification	<u>D</u> ata <u>M</u> aster Pages <u>S</u> tyle <u>W</u> indow <u>H</u> elp		
i 🗗 • 🖹 🔛 🖳 i 😭 🗑 i 🔍 🛛	L 🖬 🗄 💛   💰 🗎 💼 🖡 🗶		
😰 🕒 TPE Launcher 🚇 TPE Studio			
🚯 Palette 🛛 🗖 🗖	🕀 *Document Studio Editor 🙁 🗖 🗖		
Yalette ≥      Yalette ≥	Header Paragraph Text TPE - DOORS Data Footer Footer Console X Console X Telelogic Publishing Engine	Content Content Table Of Co Paragraph Text - 1 Table Of Co Page Break Container Text - n Text - n Paragra Text - n Style: bord Text - n Properties S Property Tag Type Recursive Level Content Assignments	PE - DOOF IPE - DOOF Intents innodule.obj aph tt - module eredCell
Column (DOORSColumn)	8		
	▲ 13:59 ÷		
	13:39		

Figure 21

Add a page number field to the footer

🗄 Telelogic Publishing Engine - Docu	ment Studio ()		
File Edit Palette Document Specification	<u>D</u> ata <u>M</u> aster Pages <u>S</u> tyle <u>W</u> indow <u>H</u> elp		
📬 • 🗈 🔛 🖳   😭 🗊   🔍 🛛	🕽 🖬 🗄 🖑 😒 🛛 🐮 🖹 💼 🗍 🗰		
😭 🕒 TPE Launcher 🔮 TPE Studio			
🚯 Palette 🛛 🗖 🗖	🕀 *Document Studio Editor 🛛 🗖	🗄 Outline 🖾	
💭 Comment 📉		E Content	~
急 Hyperlink	Header	🔣 Image - resour	ces/tel
🖶 Page Break	Paragraph	😑 👖 Paragraph	
🚔 Section Break	Text	Text - TPE	
🔚 Table Of Contents	TPE - DOORS Data	Table Of Conte	ents 📃
🕞 Table Of Tables		E Container	
🛃 Table Of Figures	Footer	Text - mod	ule.obi
🌐 Field	Page Number	□ ¶ Paragraph	
📄 Page Number		Text - r	module
(  ] Table Caption		🖻 🐴 Styles	
[8] Figure Caption		A Style: bordered	
🕒 Include File 🤍		A Style: bordered	dHeade
S Schema View 🛛 🔊 🗶 🗖 🗖		<ul> <li>A Style: Title</li> </ul>	>
		Properties 🛛	
DS1 Module (DOORSModule)			
Module (DOORSModule)     Secription (string)			
Name (string)		Property V	/alue
I → I → I → I → I → I → I → I → I → I →		-	age numbe
🗈 🔊 View (DOORSView)		Type Recursive Level 0	
🖃 💯 Object (DOORSObject)	Template Content Master page: MPData	Content	
🛛 🦓 Object Identifier (string,			assignmen
🛛 📓 Object Level (int)	🖳 Console 💥 📄 👘 👘 🖓 🖛 🗖	Condition	
Sia Absolute Number (int) Sia Object Heading (string)	Telelogic Publishing Engine	Formatting	
Object Heading (string)     Object Text (string)	A		
E I Attribute (DOORSAttribu			
😥 👰 Column (DOORSColumn)			
		<	>
	<b>E</b> 14:00		



#### Assign the page

The master page is assigned to the container element.

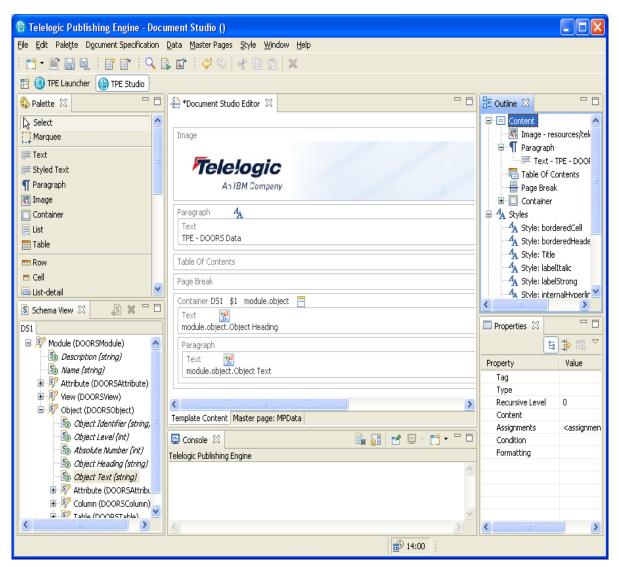
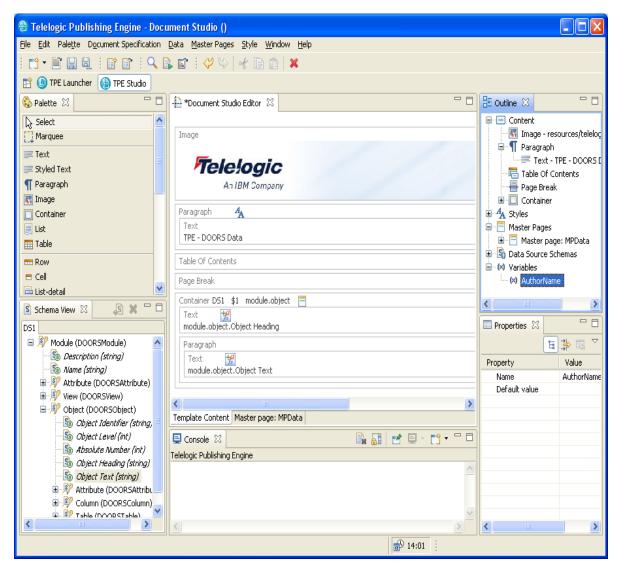


Figure 23

#### More advanced settings

#### Define a variable

The variable name is AuthorName.





Use the variable in the template after the title

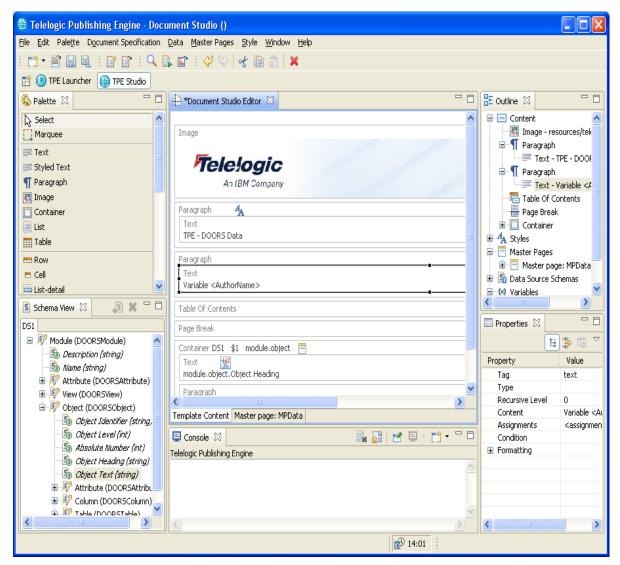


Figure 25

#### Generate the document

Before generating a document make sure you have saved the template.

Open the TPE Launcher Perspective

Use the button located in the top left part of the screen, right bellow the toolbar.

🖶 Open Perspective	
TPE Launcher	
💮 TPE Studio (default)	
ОК	Cancel

🗄 Telelogic Publishing En	gine - Document Studio (	file:/D:/TPE/Demo/demo.dta)				X
<u>File</u> Edit Document Specificatio	on <u>D</u> ata <u>M</u> asterPages <u>S</u> tyl	e <u>W</u> indow <u>H</u> elp				
i 📬 🕈 🛍 🔛 🖳 i 😭 🛛	r I 🔍 🖡 🖬					
🖹 🕓 TPE Launcher 🔮 TPE	Studio					
📄 *Document Specification 🛛	- 8	🖳 Console 🛛		🔓 🚮 📑 🖃 × 📬 🔹		
🖃 📄 Runtime*		Telelogic Publishing Engine				8
Metadata					<u>^</u>	
Output*						
Target: Word*						
Target: PDF*						
in Templates*	E\Demo\demo_dta*					
Data sources						
Data sou	irce: DS1 - DOORS*					
Variables*	(A. 16 - 11 14					
	(Authorivame)"					
Properties 🛛	₩ ₩ 2					
Property Value						
		<			V	
			14:32			
			14:32			

#### Figure 27

NOTE When the launcher is embedded in TPE Document Studio and the "keep document specification synchronized" option is checked in the preference page, the document specification is synchronized with the edited template at every load/close/save operation. Otherwise you need to manually synchronize the document specification to ensure all the data source schemas from the template have a data source equivalent in the document specification.

#### Configure the DOORS Data source

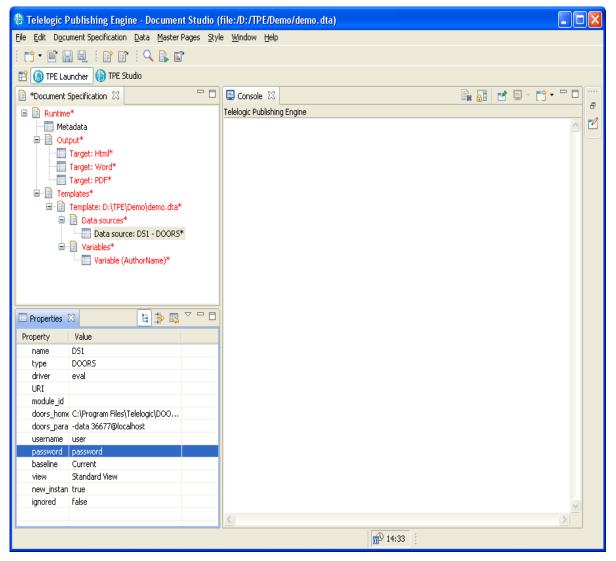
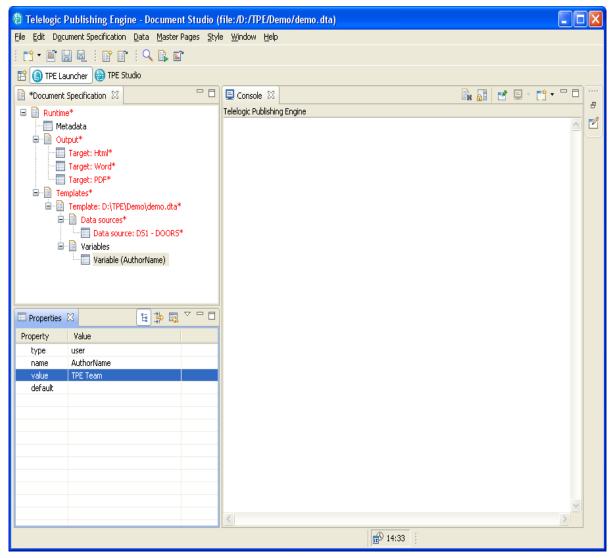


Figure 28

#### Set a value for the AuthorName variable





#### Generate the document

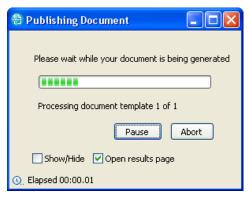


Figure 30

#### View the results

🛱 Telelogic Publishing Engine - Results	×
<b>Tele!ogic</b> Publishing Engine	
Click on a link to open or save an output document	
D:\output.html D:\output.dod D:\output.pdf	
	~
	Close
	Close

#### Figure 31

NOTE When you open the generated Word document for the first time you need to manually update the table of contents. You can use the provided "updateTOCs" macro to automate this task.

## A template for Tau data

This chapter shows how to build a simple TPE Document Template. The output document will contain the list of top level packages, and for each package the list of classes and the diagrams contained in that package. For each class a table will be generated listing all the class's attribute names and types.

- Package
   Di
  - Diagrams
    - Diagram
    - Diagram
    - ...
  - Classes
    - Class
      - Attribute table
      - Class
        - Attribute table
      - ...

Start Document Studio

Add a Tau Data source Schema

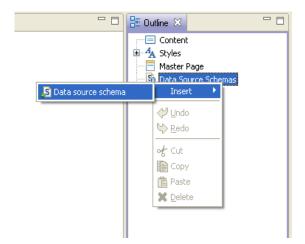


Figure 32 Selecte "Insert schema" from the outline's context menu

The "Add Datasource schema" wizard is started.

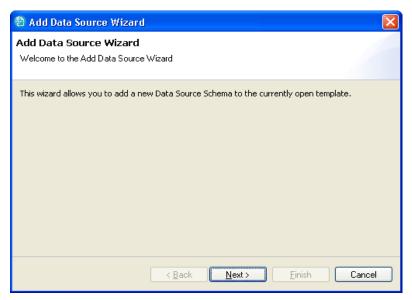


Figure 33

Select the ttdMetamodel.xsd schema provided with TPE. Make sure you set the data source type to Tau. You can use any value for the ID of the schema.

🛞 Add Data S	ource Wizard	
	Data Source Schema urce Schema file, its type and an identifier	
Schema File:	\Telelogic Publishing Engine\source\Tau\schema\ttdMetamodel.xsd	Browse
Schema Type: Data Source ID:		
	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish	Cancel

🖶 Add Data Source Wizard	×
Summary page	
Review your selections before pressing Finish.	
	_
Schema File: C:\Program Files\Telelogic\Telelogic Publishing Engine\source\Tau\schema\ttdMet Data Source Type: Tau Data Source ID: DS1	~
<u>≺B</u> ack Next> <u>F</u> inish Cancel	

🗄 Telelogic Publishing Engine - Document Studio () <u>File Edit</u> Pale<u>t</u>te Document Specification Data Master Pages Style <u>W</u>indow <u>H</u>elp - 💼 • 🖬 🔛 🖳 🗄 😭 🗊 🗄 🔍 🕒 🖬 🔮 💛 😽 🛅 👔 🗙 🖹 🕓 TPE Launcher 🔮 TPE Studio 🖵 🗖 🕂 \*Document Studio Editor 🛛 🖵 🗖 📴 Outline 🛛 🚯 Palette 🛛 ---- Content ^ ₿ Select 🗄 🐴 Styles []] Marquee \_\_\_\_ Text 🚊 🛐 Data Source Schemas 📰 Styled Text 🔊 DS1 ¶ Paragraph (×) Variables 💦 Image Container 🔳 List 🌐 Table 📅 Row 🗖 Cell 🚍 List-detail 📙 Footnote 📼 Region 🛄 Bookmark 🔊 🗙 🗖 🗖 S Schema View 🛛 DS1 🖃 👂 model (Model) Template Content - -Properties 🛛 👰 query(Any Type) 🔊 root(Any Type) 📃 Console 🖾 🖹 🚮 🛃 E - 📬 - 🗆 🗆 1 🛟 🖪 🌣 Image: Image in the second Telelogic Publishing Engine Property Value 🗄 👂 library (Package) INFO [main] - Template location C:\Program Files\Telelogic\Tele ID DS1 🖅 🔊 rootResource (Resource) INFO [main] - Loading file:/D:/cmsynergy/altair/launcher/launch URI schemas/ttdMeta i clientDependency (Dependency) 🗄 👂 stereotypeInstance (StereotypeInstar < > < > < > 21:32

Once the wizard is complete, the new data source schema is added to the template.

Figure 36 Tau data source schema added to the template

Save the document

# Define the template content

Define a query for the top level packages. In order to do that add the "Package" cast under the *model.root* element by clicking the "Cast to type" button.

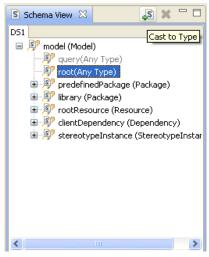
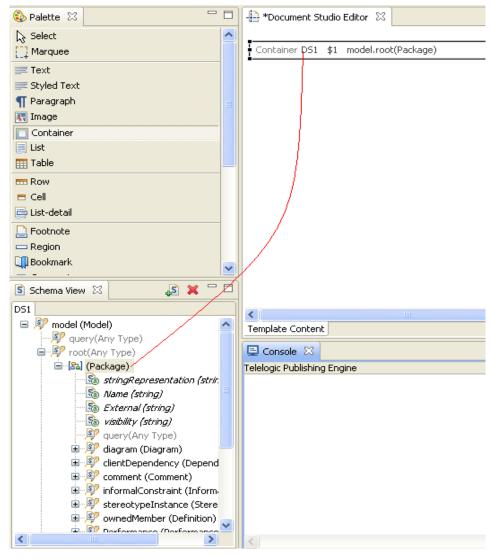


Figure 37

🖶 Select Type 🛛 🔀
LifeLine LifeLineSymbol Line LinkLine Literal Message MessageLine MethodCallLine MethodCallLine MethodSuspensionSymbol Model ModelElement MultiState NamedInstance Namespace Namespace Namespace NatStateActionOccurrenceSymbol ObjectDiagram ObjectNode OperationBody OperationSody OperationSymbol OutputSymbol Package
OK Cancel

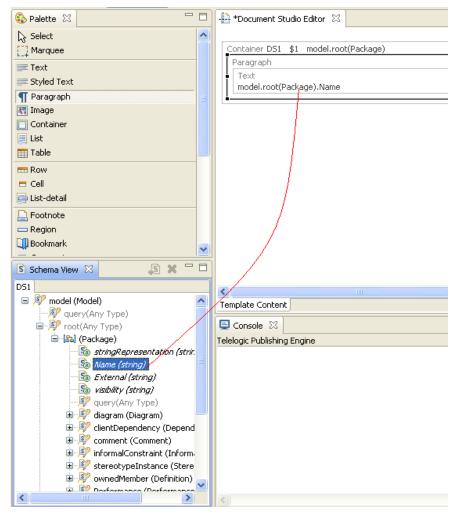
Fi	g	u	r	e	3	8



Create a container element in the template and drag the newly added Package element to it.



Create a paragraph in the container element and add the "Name" attribute.



#### Figure 40

NOTE If you would generate the document using the template in its current form, the output will contain the name of all the top level packages in the model.

- Package
- Package
- Package

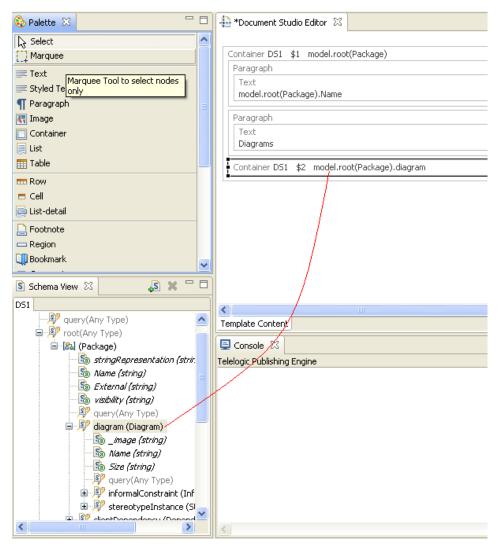
Save the document.

36 Telelogic Publishing Engine Examples Guide

To continue, add the static text heading Diagrams for the list of diagrams by adding a Paragraph into the Container, adding Text into the paragraph, then double-clcik the text and set the value to Diagrams.

+ *Document Studio Editor 🛛	
Container DS1 \$1 model.root(Package)	
Paragraph	
Text model.root(Package).Name	
Paragraph	
Text	
Diagrams	

Add the package diagram element to the template.



# Figure 42

NOTE If you are interested in a specific diagram type you can add a cast to that diagram type below the diagram(Diagram) node and use it instead of using the generic "Diagram" type.

Add an image element in the container. Set its content to be the "\_image" attribute of *model.root(Package).diagram(Diagram)*.

🗞 Palette 🛛 🗖 🗖	-🗎 *Docu	ument Studio Editor 🕺 🗖 🗖
💫 Select		
Marquee	Container	r DS1 \$1 model.root(Package)
Text	Paragrap	ph
	Text	
Styled Text	model.r	root(Package).Name
¶ Paragraph		1
	Paragrap Text	pn
Container	Diagram	ms
📃 List	Diagram	
Table	Containe	er DS1 \$2 model.root(Package).diagram
📅 Row	Tmage	· · · · · · · · · · · · · · · · · · ·
🗖 Cell	i IM	IG
🚍 List-detail	<u>i</u>	
Bookmark		
		🚢 Set value for current image element 🛛 🛛 🔀
🖻 Schema View 🔀 🔒 🧏 🗖 🗖		Simple value Data expression Script expression
DS1		Variables     Selected data item
	Template	
🖨 🔊 root(Any Type)		÷1
🖻 🕞 [82] (Package)	📮 Consi	
stringRepresentation (strin	Telelogic F	Name
🧐 Name (string)		Size
External (string)		···· image
🦾 visibility (string)		
🖓 query(Any Type)		
🖃 🦻 diagram (Diagram)		
Sa _image (string) Sa Name (string)		
Size (string)		
I query(Any Type)		Ok Cancel
🕀 👂 informalConstraint (Inf		
🗉 👂 stereotypeInstance (SI		
🖈 🕅 diant Dependency (Depend		
	<	>



Add the name of the diagram bellow its image. Additionally you can add a figure caption field so you can build a Table of Figures.

Container DS1 \$2 model.root(P	ackage).diagram	
Image		
IMG		
Paragraph		
Text Diagram	Figure Caption	Text model.root(Package).diagram.Name

## Figure 44

NOTE If you were to generate the document using the template in its current form, the output will contain the name of all the top level packages in the model, and all the diagrams in each package:

- Package
  - Diagrams
    - Diagram (image and name)
    - Diagram (image and name)
    - ...
- Package
  - Diagrams
    - Diagram (image and name)
    - Diagram (image and name)
    - ...

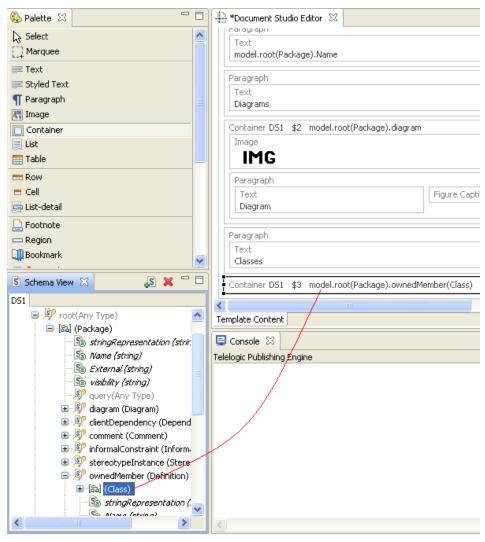
## • ...

Save the document.

Add the static text for the class list.

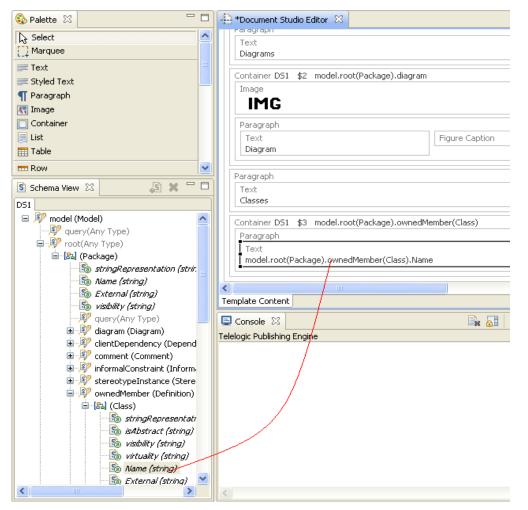
*Document Studio Editor 🛛		
Container DS1 \$1 model.root(Packag	je)	
Paragraph		
Text		
model.root(Package).Name		
Paragraph		
Text		
Diagrams		
Container DS1 \$2 model.root(Pack	age).diagram	
Image		
IMG		
Paragraph		
Text	Figure Caption	Text
Diagram		model.root(Package).diagram.Name
Paragraph		
Text		
Classes		-
I (10000)		

Add the "Class" type cast for the "ownedMember" element of the package and use it to create the query.





Add the name of the package in its own paragraph:



# Figure 47

NOTE If you would generate the document using the template in its current form, the output will contain the name of all the top level packages in the model, and all the diagrams in each package:

- Package
  - Diagrams
    - Diagram (image and name)
    - Diagram (image and name)
    - •
  - Classes

- Class 1
- Class 2
- Class 3
- ...
- Package

٠

- Diagrams
  - Diagram (image and name)
  - Diagram (image and name)
  - ...
  - Classes
    - Class 1
    - Class 2
    - Class 3
    - ...

• ...

## Save the document.

Create the table with two rows and two cells that will host the class attributes:

Container DS1 \$3 model.root(Package).ownedMember(Class) Paragraph	)	
Text model.root(Package).ownedMember(Class).Name		
Table Row Cell	Cell	Text Content: model.root(Package).ownedMember(Class).Name
Row Cell	Cell	

Add the "Attribute" type cast to the "ownedMember" element of Class and use it as a query on the second row of the table. At the same time apply the borderedCell style to all the table's cells.

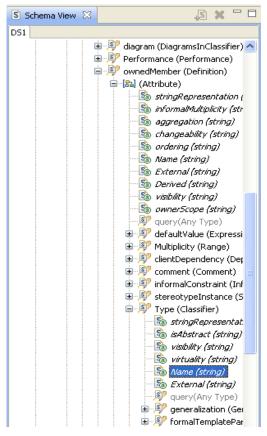
Paragraph		
Text		
model.root(Package).ownedMember(Class).	ame	
Table		
Row		
Cell 🗛	Cell A	
Text	Text	
Name	Туре	
Developed at an add week/Deduced average		
Row DS1 \$4 model.root(Package).owned	ember(Class).ownedMember(Attribute)	
Cell 🗛	Cell 🗛	

#### Figure 49

Use the attribute name in the first cell:

Container DS1 \$3 model.root(Package).owText Paragraph Content: Classes	
Text	
model.root(Package).ownedMember(Class).Name	
Table	
Row	
Cell 4	Cell 4 Text Type
Row DS1 \$4 model.root(Package).ownedMember(Class).ownedMemb	er(Attribute)
Cell <b>4</b> Text model.root(Package).ownedMember(Class).ownedMember(Attribute	Cell 4

Drag the attribute's type element as q query on the  $2^{nd}$  cell of the  $2^{nd}$  row and drag the name attribute in the cell.



agraph	
Brahu	
xt	
del.root(Package).ownedMember(Class).Name	
e	
W	
iell A <sub>A</sub>	Cell 🗛
Text	Text
Name	Туре
w DS1 \$4 model.root(Package).ownedMember(Class).ownedMer	mber(Attribute)
ell 🗛	Cell DS1 \$5 model.root(Package).ownedMember(Class).ownedMembe
Text	Text
model.root(Package).ownedMember(Class).ownedMember(Attribut	te model.root(Package).ownedMember(Class).ownedMember(Attribute

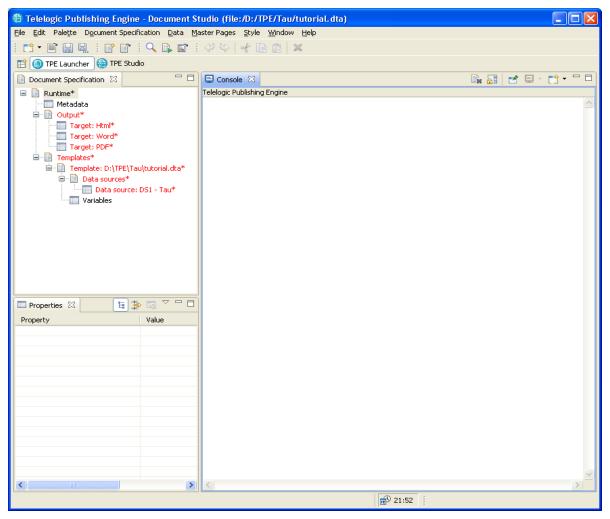
## Figure 52

NOTE Setting a query on a cell/row will generate as many cells/row as the query returns. But since an attribute has only 1 type, the query will return a single element.

# Save the document.

# Generate the document

Open the launcher perspective



#### Figure 53

Select the Tau project (.ttp file) you wish to use as concrete data source for the DS1 schema used in the template. You can use one of the examples provided with Tau, which can be found in the examples directory below the Tau installation directory (on Windows the default location of the examples is C:\Program Files\Telelogic\Tau\_4.2\examples):

*Document Specification X          Runtime*         Metadata         Output*         Target: Html*         Target: PDF*         Templates         Template: D:\TPE\Tau\tutorial.dta         Data sources         Data source: DS1 - Tau         Variables				
🔲 Properties 🖂	E \$ R ∨			
Property	Value			
name	DS1			
type	Tau			
driver	eval			
URI	CoffeeMachine\CMdesign.ttp			
ignored	false			

# Figure 54

NOTE Optionally, you can configure each output target.

Start the document generation process. TPE will ask you to save the changes to the document template, if any unsaved changes exist.

🖶 Save Resource	X
'Document Studio Editor' has been modified. Save changes?	
<u>Y</u> es <u>N</u> o	Cancel

# Figure 55

Unless you cancel the save the document generation process will start

🕆 Publishing Document 📃 🗖 🗙
Please wait while your document is being generated
Starting run
Pause Abort
Show/Hide 🗹 Open results page
()_ Elapsed 00:00.01

Once finished a window will be displayed allowing you to view/save the results

🕆 Telelogic Publishing Engine	- Results	×
Tele!og	<b>gic</b> Publishing Engine	~
Click on a link to o	pen or save an output document	
\Temp\tpe\c \Html_1225 C:\DOCUME- \Temp\tpe\c \Word_1225 C:\DOCUME-	-1\Spurlos\LOCALS~1 put_1225914862955 914862955.htm -1\Spurlos\LOCALS~1 put_1225914863064 9914863064.doc -1\Spurlos\LOCALS~1 put_1225914863174\PDF_1225914863174.pdf	
		~
	Close	

# **Appendix: Notices**

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send written license inquiries to:

IBM Director of Licensing IBM Corporation North Castle Drive Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send written inquiries to:

IBM World Trade Asia Corporation Licensing 2-31 Roppongi 3-chome, Minato-ku Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions. Therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Intellectual Property Dept. for Rational Software IBM Corporation 1 Rogers Street Cambridge, Massachusetts 02142 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

# Trademarks

IBM, the IBM logo, ibm.com, Telelogic, and Telelogic DOORS are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or <sup>™</sup>), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at www.ibm.com/legal/copytrade.html.

Microsoft, Windows, Windows 2003, Windows XP, Windows Vista and/or other Microsoft products referenced herein are either trademarks or registered trademarks of Microsoft Corporation.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product or service names may be trademarks or service marks of others.