Park|Consulting
Our business is improving yours

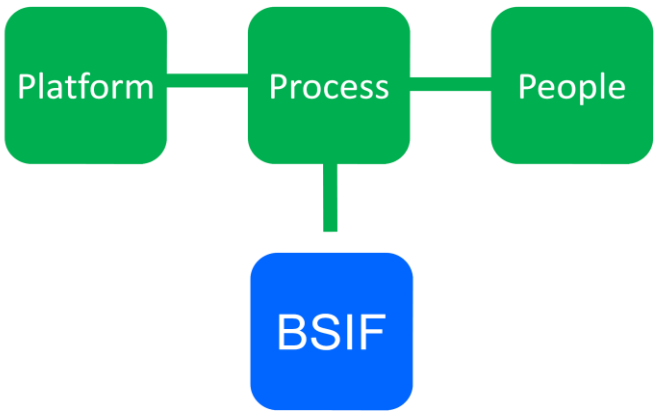
33 Chester Rd
Tawa
5028
Wellington
New Zealand

P: (+64) 4 232 - 2092
m: (+64) 21 322 091
e: ross@parkconsulting.co.nz

The Master Class Series

Using Enterprise Architect for

Project Documentation



Powered by BSIF

The Business Solution Integrated Framework

Ross Park

Park | Consulting
Our business is improving yours

CONTENTS

Copyright

Park | Consulting
Our business is improving yours

1 Introduction

Enterprise Architect as a modelling tool has grown from strength to strength. The addition of BPMN, Enterprise Architecture frameworks and other modelling notations, into a single affordable tool, have seen Enterprise Architect adopted as the tool of choice for a wider range of the Business Solution delivery lifecycle.

Having now implemented EA as the tool for analysis and design for, over 11 organisations, I have found that a critical part of its adoption is the transition from using word to create the project document deliverables, to generating them directly from EA in the form of reports.

In the course of getting there I had to address a number of challenges and the transition was not plain sailing by any means. The good news is if you are aware of the challenges and are willing to put in the effort you can produce high quality documents directly from your project models.

This whitepaper looks at these challenges and how they were overcome using approaches that have worked in large and small organisations enabling them to use EA as the core tool for the production of the various documents that are required during the product development lifecycle.

The Need for Automated Project Documentation

For those that believe that there is no need for documentation - I have to differ. I have yet to find an organisation that can deliver solutions to it's business without documentation. The adoption of Agile practices doesn't change this either. Agile is not about removing documentation, it's about focusing on the more important deliverables. I have, and would continue to argue, that the key to supporting agile approaches is the automatic generation of project documentation from within the modelling tool. Having analysts spending more time working on what goes into the models, rather than what comes out is an important step to agile approaches that are sustainable .

How Important is Project Documentation?

Once EA has been fully implemented, including document generation for projects, the ability to reliably produce these documents became the single most important feature of the tool for most of the clients that I have worked with. Some of this relates to the pride people take in producing quality deliverables and some related to strict corporate standards that need to be maintained.

If you want to commit to using EA to provide the project documentation required for analysis and design phases of the product development lifecycle - its ability to do so faultlessly becomes paramount.

What are Projects Documents?

When I talk about project documents I mean complete ready to go reports that contain all of the sections formatted and styled as required by the organisation for its projects. This includes all the requirements, use cases, process flows and diagrams, that EA is so good at creating, wrapped in all the preamble and other sections, that consumers and corporate standards demand.

This typically includes additional sections such as:

- Cover page details
- Version details
- Related documents

- Document sign offs
- Glossary terms
- Table of Contents
- General information

If you want to create documents directly from your models then you will need to find an effective way of including all this content in the model.

What are the Options

There are three main options you have for document generation from EA

- Use Code to generate the documents
- Generate sections using RTF templates and then put the document together in Word
- Generate the entire document in EA using Document Models

I was looking for a simple approach that enabled organisations to adapt, modify and evolve the documents as they needed. This started out by using an early edition of EA that required generating each section as a separate RTF document and then linking then all into a word document. The results worked but it took around 28 steps to produce the result and so was messy to say the least.

The code option was trialled but it was rejected simply because it was too complex for most users to maintain and modify. It is still a valid option and it allows a level of control that the RTF editor doesn't provide - but you need to have the right skill set in house to manage it.

I needed a better way - simple, intuitive and repeatable and the solution came in version 7.1 in the form of document models.

Document Models	Document models provide the simplest and most effective way to automate documentation in EA.

The Challenges

I could see the potential of document models but it still took many hours working with EA before I gained a good enough understanding of the best approach to project document generation in EA . While it's easy enough to create RTF templates - generating complete project documents was more of a challenge. It's a case of the 80/20 rule where the first 80% of project document generation is relatively easy - the last 20% is the hard part.

It's a case of needing to spend time learning to work with the tool to get the best result and there were two key challenges

- Localizing information capture to the project level
- Structuring Information for document generation

Localising Information Capture

The first challenge was to learn to localise the capture of project information to the project.

One of the greatest features of EA is the ability to combine all your architectural and project information into a single shared repository. So projects now become models within the main model and when you take this approach - you need to find a way of localising information to the project level.

This includes:

- Glossary Terms
- Project resources
- Clients

I worked with an organisation that had over 5,000 glossary terms! Only a small sub set was ever used for any particular project so I needed an approach that enabled any sub set of these to be localised to a project. Project resources are project specific and the same resource may have different roles in different projects. Many of the key items that one would expect to use in documents need to be the model.

Structuring Information

The second challenge was to learn to structure information in a manner that lends itself to using the RTF editor to produce project documents.

The RTF editor is very complex and provides some powerful features to control the way the model information is rendered in the output. Selecting the content to include is very much a case of point and shoot and so you need to use your project structure to manage the content in a way that works with the selector.

Once you understand these two key factors - project documentation starts to make sense.

The Three Ingredients for Project Documents

Generating project documents is a three part process.

1. You need a suite of RTF templates. Depending on the complexity of your documents you will need to produce around 30 to 60 RTF templates to cover all the variations that will be required to produce a typical range of project documents.
2. You need to have a project template that stores elements in a defined package structure that supports document generation.
3. You need to build document models into your project templates. Once you have the RTF templates and project structure in place you then need to integrate these using document models - one for each document you need.

We will not be looking at project templates here. Instead we will focus on RTF templates and document models.

2 RTF Templates Basics

RTF templates allow you to access information on your models and control its format in Rich Text Format rendered output. RTF is a vendor independent format that can be imported into most word processers with the formatting in place. So once created the RTF output can then be imported and converted into Word documents.

Getting Your styles into EA

It is important that before you start creating RTF templates that you establish a base set of styles that have all of the corporate styles included. This ensures that outputs from EA will be acceptable to all consumers. New templates can be copied from the base template that has these styles providing consistency of styles across all templates.

If you don't do this and then decide you don't like the styles you will have to change them for every template you have created!

If you have corporate styles in a word document then it's a good idea to import these into your RTF templates before you create any templates. Some formatting can only be managed in Word so you need to make sure that the styles are correct before you import then.

How do you do this?

- Create a document that has your styles in it
- Remove all content from the document including headers and footers.
- Ensure styles are correctly named Heading1, Heading2 , TOC1, TOC2 etc.

It is common to use a tilde (~) to identify custom styles e.g. ~Heading1 or ~TOC1

The RTF template editor uses style names to control indenting of headers so these styles must be named correctly for them to work.

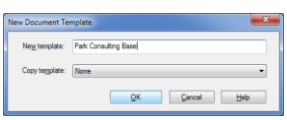
- Save the file in an .rtf format

You are now ready to create your base template in EA and to import the styles your have created.

All other templates you will then create will be copied from this template so it is important that you get it right.

How do you do this?

- Select the Project →Documentation → Rich Text Format Report (or use F8).
- Select the Template Tab
- Create a new template by selecting the New… button.



- Give the template a name such as Base template, select the OK button.

⚠ Do not copy any of the existing templates - they don't have your styles.

This gives you a blank RTF template and the next step is to import the styles you have created into your base template.

How do you do this?

- Open the base template
- Right click and Select import from the popup menu

- Browse to the RTF document you created and select ok

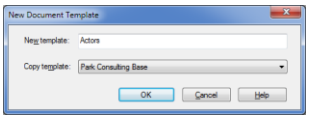
The styles in your RTF Document will be imported into the base template and you should be ready to start making your own custom RTF Templates.

Using the RTF Template Editor

Once you have your styles setup you can start to create your own custom RTF templates to render sections of your model

How do you do this?

- Select the Project →Documentation → Rich Text Format Report or use F8.
- Select the Template Tab and select new.



- Select the base template from to copy from and the new template will have the same style sheet as the base template.

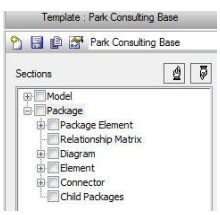
The RTF editor will be then be displayed

The RTF editor allows you to select content form your models and to format and style it.

The Section Selector

The first step is to select the content to add to your RTF Template. The Sections selector on the left hand side allows you to access information from different areas of the model.

This includes:

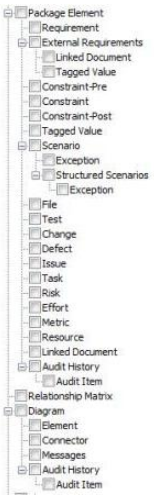


Model – General information that is not presented in the project browser including Glossary, Issues and Tasks

Package – Information that is shown in the project browser that can be accessed from the following selections:

- Package elements
- Relationship Matrix
- Diagram
- Elements.
- Connectors
- Child Packages

Each of the sections can be expanded to access sub sections containing more detailed information by clicking on the – and + nodes. They can also be reordered by selecting the section or sub section and clicking on the up and down hands.



The type and amount of information available depends on the section that is selected:

Package section

This section provides access to the information associated with a package.

Package Element Section

In this section you have access to some but not all of the available information for an element. Excluded are attribute, method and connector information.

It can be used for templates where the additional information is not required.

Diagram Section

This section provides access to field information for a element. That means that any information sorted in a list associated with the element cannot be accessed.

In a collaborative environment you may want to reduce duplication of entities such as actors by using a central shared collection in multiple project models. This can be achieved by simply adding links to the elements on a diagram. You then need to document the elements from the diagram to render these in the RTF output.

Element Section

This section provides full access to all of the information for an element.

In addition to this you can access child diagram and element information from this section.

Adding Element Information

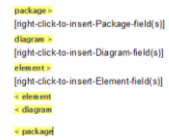
Once the RTF style template editor is displayed you can select the information you would like to add to the template..

How do you do this?

- Check required box in the sections selector

Note When you select a section all parent sections will be selected as well

This adds sections with yellow markers to the RTF template where fields can be added



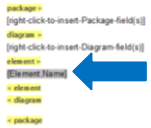
Adding a Element Field

You can now add field information to your template

- Right click on the RTF editor page in the required location
- Select Insert Field

A list of the available fields relating to the section will be displayed for you to select.

Select the required field and it will be added to the template enclosed in a bracket.



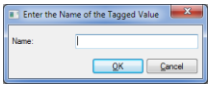
Adding Tagged values

You can also selectively add tagged values to your document using the valueOf as the field.

How do you do this?

- Right click on the RTF editor page in the required location
- Select Insert Field
- Select valueOf

You will then be prompted for the name of the tagged value to include



- Type in the name of the tag
- Click on OK

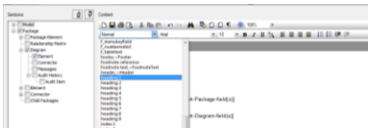
Formatting Text

You can now format the text and paragraph

How do you do this?

If you have the toolbox visible

- Select the section you want to format
- Select the style and other formatting from the toolbox



To turn this on:

- Right click on the RTF template and select view and then toolbar

If you don't want the toolbox displayed :

- Select the section you want to format
- Right click and select Font and then style and other formatting from the menu

Tables

You can also add tables to format data in the document. The most common way to use tables is to create a tabular section for a list of elements.

Tabular Tables

These are tables containing any number of columns, but with only *two* rows.

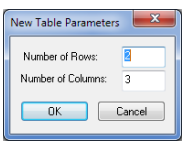
- The first row is the headings of the columns, which you type in and format yourself.
- The second row contains the output, which you do by right-clicking in each cell and selecting from the field list.

The second row is then rendered iteratively for every element creating a tabular table.

How do you do this?

- Select the section you want to add the table to
- Right click and select Table-> Insert Table from the menu

The New Table Parameters Dialog will appear



- Select the number of columns your require but leave the number of rows to 2

Once the table has been created you can access additional functions from the same menu

The options available include:

- Inserting rows and columns
- Splitting and Merging cells
- Deleting Cells
- Setting Header Rows
- Formatting layout of rows and Cells
- Formatting styling of rows and Cells

By default an extra return is added after the table. You need to remove this space for it to work as a tabular table.

Documenting Elements from Diagrams

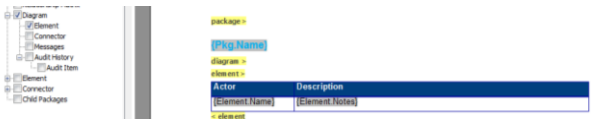
Elements can only exist in one place in the project browser. - Links to these elements can be added to as many diagrams as is required. So if you want to localise shared elements - adding them to diagrams is a great option. The actor template above requires the actor elements to be in a particular location in the project browser. Changing the template to document the elements form a diagram allows the same actors to be used in multiple projects.

When you want to document information from a diagram there are two steps that you need to take.

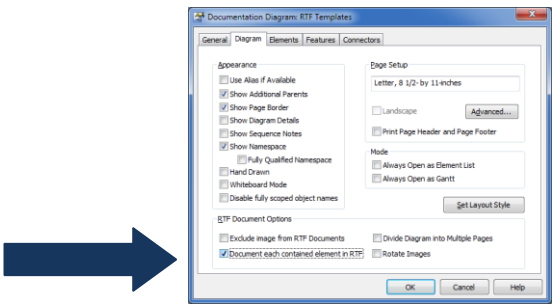
1. You need to create an RTF template that references the elements on the diagram as shown below.

   Note that the diagram – element is selected and a

   table has been created to place the element name and notes in.



2. You need to select the properties of the diagram that has the elements on it and in the diagram tab under RTF Document options click “Document each contained element in RTF”



The amount of information that can be accessed from elements on a diagram is limited - but includes field and tagged values information making it suitable for generating tables.

Linked Documents

Linked documents provide a means of adding rich content to your model by linking a document to a package or element.

These hold an internal RTF template that can be used to store formatted documents in EA. The difference is that these documents are linked to packages, or elements, and you can include the contents in reports produced using Model Documents

An example of output generated using a linked document associated with an element. The RTF Template extracts the following information:


Professional Level

RTF Templates

Delivering quality document outputs to stakeholders is always an important consideration and EA Rich Text Format templates enables you produce these outputs

Authored by an innovator in RTF template techniques who will show you how to create quality documentation.

This module teaches you how to create professional looking output from your models using the RTF Template Editor.

Topics

RTF Templates
- Creating a Base Template
- Importing Styles from Word
- Templates

Accessing Content
- The Sections Selector
- Package Elements
- Diagram Elements
- Accessing Field Information
- Accessing Tagged Values
- Tabular List Information
- Linked Documents
- Images

Formatting Output
- Sections
- Headers & Footers
- List Numbering
- Text Formatting
- Paragraph Formatting
- Table of Contents
- Page Counts
- RTF Options

Document Modelling
- Master Documents
- Model documents

Package Name: (Professional Level)

Element Name: (RTF Templates)

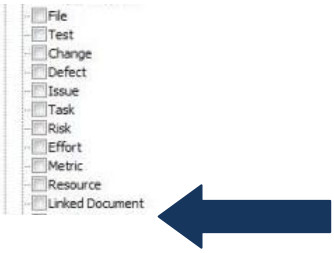
Element Notes: Delivering.....

The rest is in the linked document

These can then be included in RTF documents by Select the Linked Document section in the selector.

No additional detail is required.

This is incredibly useful - there is often information that comes from an external source that you want to include in documents. It could be a data sheet for a device or just a table that can't be generated within EA



The complete contents of the linked document will be rendered in the final output

Formatting

The RTF editor also provides additional features that allow you to further format documents.

- Add document models
- Add images
- Add OLE objects
- Create sections
- Add reference material such as
- Headers and Footers
- Dates

- Page number
- Page Count
- TOC
- Format fonts and paragraphs.

Note A complete discussion of formatting is beyond this article but here are some useful tricks on formatting.

Tables of Contents

You can add a table of contents (TOC) to an RTF template however it needs to encounter at least one header style when rendering before EA will generate the TOC.

I have found the best way to do this is to add the TOC to the RTF template used to create the section prior to the TOC. The header for the section then causes the TOC to be generated.

You need to refresh the TOC once you have generated the complete document.

Headers and Footers

While the RTF editor allows you to put headers and footers in you can't access model information from them or put some of the typical references, such as file name, in them so they do have limitations. If you just want your logo and standard inclusions such have I have in this document it works fine and you can have a separate first page header and footer.

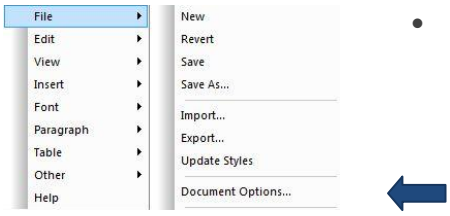
Once you have finished updating the template use the File > Close menu item to close and exit from the template editor window.

You can now use these RTF template as part of a model document or for ad hoc generation of documentation from your model.

Document Options

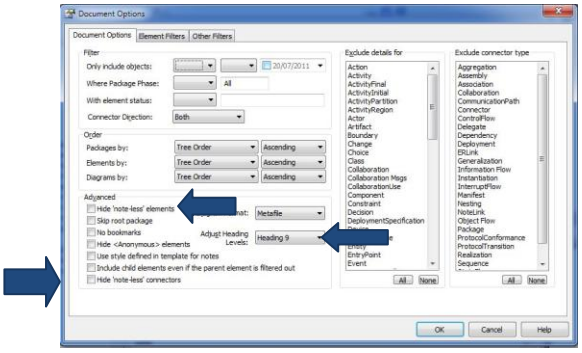
You can also set document options at the template level by selecting the Document options. This saves having to make these selections when running the document generator.

To do this:



- Right click on the RTF template and select File -> Document Options

The Document options can now be set



The key selections you may want to set are in the advanced section:

- Restrict the header levels to be used to match the TOC styles that you have
- Hide elements/connectors with no notes
- Hide unnamed (Anonymous) elements
- Select use Template Styles to override Rich notes formatting

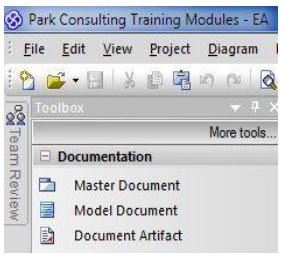
3 Document Models & BSIF

BSIF was designed to make modelling business solutions easier and so enhancements to support the generation of complex documents are incorporated into the BSIF MDG and supporting processes. Our goal for generating documents was to make it as simple as running a report, requiring no significant additional work to produce the finished result.

This section looks at the documentation elements we have provided in BSIF, using the RTF templates and approaches that allow entire project documents to be generated out of EA.

Document Models

EA provides a special document toolbox that allows you to model the virtual document and to create any number of sections within the virtual document. Each section has an associated RTF template that controls the formatting of the sections output.



There are two elements used to create the complex documents Master Document and Model Documents.

Master Documents

The Master Document is the shell that allows you store the model documents. They can have content included or simply be used as a blank holder to the Model Documents that have all the content.

They are actually stereotyped packages which when added to a diagram have an internal diagram that allows you to add sections by using Model Documents.

Each Master Document has its own appearance defined by an associated RTF template.

Tip I use a completely blank RTF template for my Master Document.

Content is only added using Model Documents.

Model Documents

Document models allow selected packages of your model to be rendered using a specified RTF Template. The are used to add content to your documents that comes from information stored in your model.

This can include:

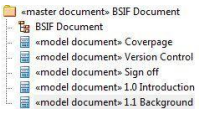
- Cover Pages

- Table of contents
- Section headers
- Custom rendering for each of the sections included in the master document.

Once you have a Master Document you can then add Model Documents as required by dragging the Model Document element onto the Master Documents diagram.

- When prompted supply a name that identifies the section and click on OK.
- Open the Tagged Values window, or element tab, and click on the Model Document element.
- Select the RTF Template to use for the Model Document element.

The elements are then created in the project browser

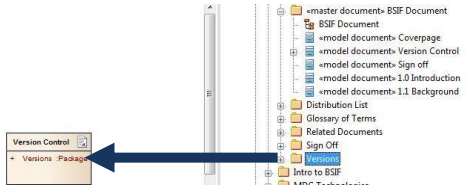


Tip The order of the elements in the project browser determines the order in which sections will be rendered. The diagram is only used to create the elements.

The next step is to add the source package or packages to the model document

How do you do this?

- Drag and drop packages from the project browser onto the Model Document element on the diagram.



Each package is added as an attribute for the Document Model. Once you have added the required packages you can change the order that the packages will be documented in the section by opening the Model Document elements and reordering the attributes.

It's then a matter of repeating the process for each section that is required

I have constructed some very complex documents using this process and it works well and is simple to modify and maintain.

Approaches to Localising Information

When it comes to producing complete project documents we need to localise the information to the project and I have only seen two approaches, so far, that have been successful.

Linked Document Approach

A linked document can be used to hold the front-page and the tables for versions as well as the Table of Contents. This then becomes the first document model to be called

An example of a cover page using a Linked Document

While this approach works you have to open up the linked document each time you generate to see if the details are correct and any duplicated information will have to be updated in each linked documents for such information making them harder to maintain.

It also limits sharing between documents as the information is specific to each document.

The BSIF Approach

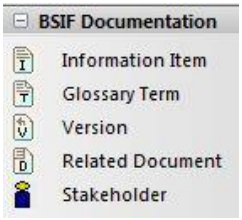
The Document Toolbox

The other approach to localising information is simply to add the information to your model using elements and packages that can be edited in the model. In other words to have a model for sections of the project document.

This approach is supported the BSIF Document toolbox.

This toolbox provides all the elements that are required to generate the sections on the fly by adding them to packages or diagrams in the project template. I pioneered this approach in my early days as an EA consultant and have yet to find an easier or more effective method.

The toolbox is relatively simple providing a selection of elements that allow you to model the sections of the document that are used to wrap the main content.



It includes:

- **Information Item**s - Paragraphs of text such as a problem statement
- **Glossary terms** - to create the glossary as elements in the model
- **Version elements** - elements for the version control table
- **Related Documents** - elements for the related documents table
- **Stakeholders** - These represent people involved in the project

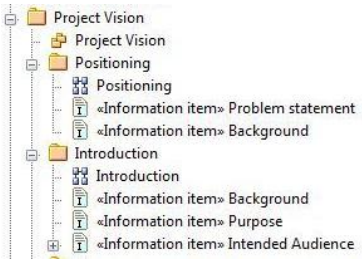
To use the elements you just need to add the BSIF MDG and create a package with the BSIF documentation diagram in it that will provide the toolbox.

All use the Name as the name of the element and the notes for any details.

The BSIF document toolbox provides elements with the associated attributes required to generate the tables you need for documentation. These attributes are captured in tagged values for each of the elements.

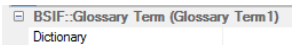
Information Items

Information items are used to capture paragraphs of text and are used all over the place in complex documents. An example is shown here where typical inclusions have been added using these elements.



Glossary Terms

These include a dictionary tag that can be configured as a lookup or left as free text.

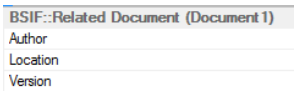


Versions

This includes tags for the author, Revision Date and Version

Related Documents

These represent external documents - such as other specifications that are related to the current document. It includes Author, Location and version tags.



Stakeholders

I used to have people and stakeholders but finally realised that the are the same thing!

The elements have the tagged values needed to capture the persons details

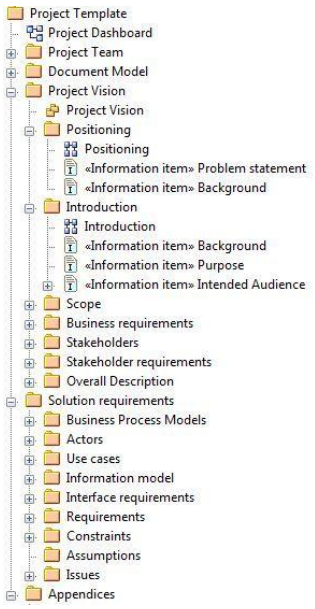
While people may have different roles in different projects they usually have the same role within a project. I use a Project team folder to create all my project stakeholders including team members. I then add them to diagrams to create the lists needed for each document - dragging them from this folder onto the diagrams which saves a lot of time.

Structuring Information for Project Document using BSIF

Lets now look at the steps to producing our complete document using information stored in our model.

The Project Template

The following shows a sample project template structure.



The project template is important as you need to structure the way you capture information if you want to create documents based on it.

This sample is broken up into the following key sections

- Project Team
- Documentation Model
- Project vision
- Solution requirements
- Appendices

The benefits of using a template approach are many:

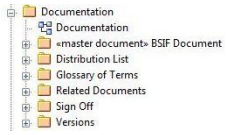
- You can overlay documentation
- You can preload standard elements
- You can produce user guides
- You can deploy them using MDG technologies
- You can share information

We are interested in overlaying documentation onto the model template.

Documentation Model

If it hasn't become obvious yet we are actually modelling the document and you will need a section in our model template to put all these elements and diagrams.

I have a separate documentation model section in my templates similar to below.

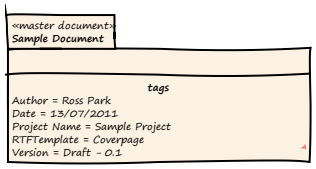


These are documentation models that include the Master and Model document and elements for all documents.

Cover Pages

You will need a template that can produce the cover page for your document and you need to store the information displayed somewhere in your model.

I use the Master Document element to store the cover page details for each document as tagged values and this works well making the details apparent each time you generate the document.



In this case the following tags have been added to the Master Document

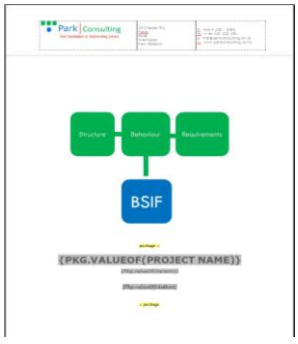
- Author
- Date
- Project Name
- Version

The RTF Template tag shows the template I am using to render the content in this case my cover page template.

The Associated RTF Template

As before, it is a case of coping my base RTF template and then adding the information for the package section. Most of it is tagged values so I just add "valueOf" selections for each.

The resulting RTF Template looks similar to the linked document but the content is taken from the tagged values that have been added to the Master Document package.



package >

{PKG.VALUEOF(PROJECT NAME)}

{Pkg.valueOf(Version)}

{Pkg.valueOf(Author)}

< package

Version Control and other Sections

The same process is then applied to create the RTF templates for the other required sections

Here is the Version table that includes the element name and notes and tagged values for the author and revision date.

element >

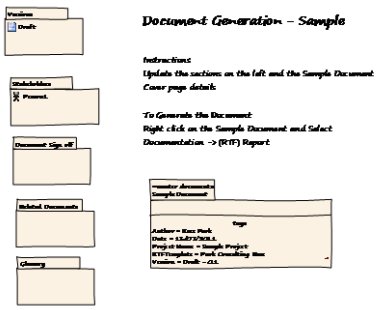
Version	Date	Author	Changes
{Element.Name}	{Element.valueOf(Revision Date)}	{Element.valueOf(Author)}	{Element.Notes}

< element

Documentation Diagrams

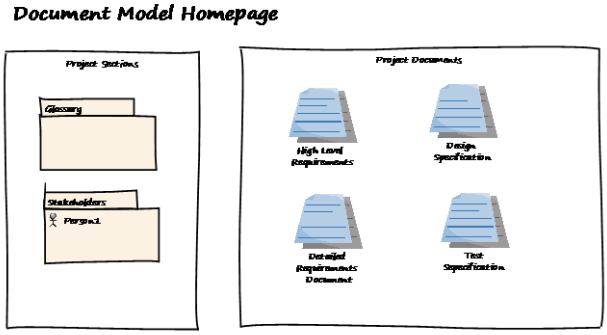
Within my documentation model section I include a documentation diagram for each document I want to create. This provides a single place in your model template that you go to manage all the document related sections and to generate the document.

The following shows an example of a documentation diagram.



Note Some of the sections may be related to a single project document - such as versions and related documents. Others such as the glossary may be shared for a project.

You can then add a documentation homepage to your project template that includes the shared sections and a link to each of the document generation pages as shown.



Summary

I must admit that when I first looked at this approach - making up models for documents- I felt like a bit of a naughty boy. I was not sure that it was the acceptable, or correct thing, to do with a modelling tool. The reception that I received the first time I used it quashed any concerns that I may of had. The approach has proved to be so successful that it became a standard part of my approach - providing a simple, yet effective, method for producing project documentation from within EA.

The BSIF approach enables most project work to be delivered using standard templates and provided documents. The rest can generally be managed by adapting the standard template to produce a custom project model. While maintaining the RTF templates can be complex, using and maintaining the document models is not and with a little training most people are able to copy and modify the document models to suit particular projects.

You can apply the same approach to sprint based agile projects. I have developed project templates that do this by changing to the way the project template works and having mini templates for each sprint. These sprints are created within a main project template that holds all the shared project elements.

Providing a standardised project template with the documentation built in has always been a core part of my implementation strategy. I have found that it typically results in fast tracking wide spread adoption of the tool within organisations I have worked with.

Of course there are still some tips and tricks that I haven't include in this paper - but they build on the material that I have covered here - so it's a good starting point.

I hope that this was helpful to you!

Happy modelling

Ross

4 BSIF

BSIF was developed to support effective Business Solution delivery using modelling notations and so is ideal for organisations that are looking at using EA as the tool of choice for analysis and design.

Because BSIF evolved from years of experience, discovering how to make Enterprise Architect useful for practitioners involved in Business Solution delivery, it is a very pragmatic framework that bridges the gap between theory and practical use of the tool to produce deliverables.

BSIF has three parts to it.

Platform

BSIF is provided in the form of an MDG technology for EA that has all the elements and diagrams required.

Processes

BSIF has been carefully designed to support agile work practices based on collaborative modelling. At the centre of this are modelling approaches that have deliverables that integrate phases of solution delivery and make use of shared reference information and frameworks.

People

We also provide our Master Class training program. This is a complete and integrated range of over 20 training modules designed to support the implementation of Enterprise Architect. It includes optional sections on the use of BSIF.

Portions of this white paper have been reproduced from this innovative training program.

Downloading BSIF

The BSIF MDG can be downloaded free from our website along guides in the use of the framework that will get you started.

www.parkconsulting.co.nz

Park Consulting also provides a complete range of Implementation, Consulting and Training services supporting organisations looking at implementing EA (with or without the use of BSIF).

If you have any comments for questions I can be contacted at:

Ross@ParkConsulting.co.nz