# OVERVIEW

DocDynamo dynamically generates HTML documentation from XML file(s). Using Vagrant, a VM is built and a set of docker containers are created to act as hosts for the commands to execute as specified in the XML document; this ensures that the commands are correct and execute as anticipated. Displayed commands can be easily copied and pasted; useful for installation and migration documents.

# INSTALLATION

Using the Github CrunchyData/doc-source, crunch/ha/master (<https://github.com/CrunchyData/doc-source/tree/crunchy/ha/master)> as a template, navigate to the cloned root directory and create the vagrant VM using vagrant up. Once the VM is built, which can take several minutes, enter the VM using vagrant ssh.

DocDynamo will be cloned from the pgBackRest repository. DTD files and the engine itself are loaded as part of the installation.

# CAVEATS

Currently, DocDynamo is tied to pgBackRest however, even though all of the pgBackRest code is pulled, it could be packaged such that only the common code that is needed is pulled. So right now, the version displayed is the pgBackRest version which is also considered the DocDynamo version.

# DOCUMENT CREATION

Required files are the manifest.xml which defines the documents to load and render and the XML document(s) to load.

The manifest.xml must be located in the doc directory (one level from the root) and the XML document(s) to render must be located in the doc/xml directory.

To build the documentation, enter the VM and execute the following:

docker rm -f $(docker ps -a -q);/docdynamo/doc/doc.pl --doc-path=/crunchydoc/doc

The first part of the command removes existing docker containers to ensure each is rebuilt from scratch. The doc-path indicates the location of the manifest.xml and XML document(s) to render.

## General Options

The DocDynamo invocation provides the following options.

|  |  |
| --- | --- |
| Option | Description |
| --help | Display usage and exit |
| --version | Display version |
| --quiet | Sets log level to ERROR |
| --log-level | Log level for execution (e.g. ERROR, WARN, INFO, DEBUG) |
| --deploy | Write exe.cache into resource for persistence |
| --no-exe | Should commands be executed when building? (for testing only) |
| --no-cache | Don't use execution cache (the exe.cache file). Useful for ensuring all commands get re-executed after changes to the documentation are made. |
| --cache-only | Only use the execution cache (the exe.cache file). exe.cache is where all dynamically generated sections of the document are stored statically; it is used for reproducibility and deployment. Also useful if a Docker environment is not available since the documentation can be built from the cache. |
| --var | Override variables defined in the XML. This option can be repeated. Useful, for example, for changing the os type or the PostgreSQL version. Invocation: --var=varname=value |
| --doc-path | Document path to render (manifest.xmlshould be located here) |
| --out | Output types (html, pdf, markdown) ONLY HTML IS VALID FOR GENERAL USE. |
| --require | Require only certain sections of the document (to speed testing). The default dependencies for a section are all the proceeding sections. This option can be repeated. NOTE when using this option, the --include option is required. This option is not usually relevant to installations. |
| --include | Include source in generation (links will reference website). The parameter must match the render-source key (or source attribute if specified) in the manifest.xml. This option can be repeated. (include and exclude cannot be provided together) |
| --exclude | Exclude source from generation (links will reference website) The parameter must match the render-source key (or source attribute if specified) in the manifest.xml. This option can be repeated. (include and exclude cannot be provided together) |
| --keyword | Keyword used to filter output. Overwrites keyword ‘default’ with the value provided thereby executing only those elements that are tagged with the new keyword or do not have a keyword at all. |
| --keyword-add | Add keyword without overriding 'default' keyword |
| --dev | Add 'dev' keyword. Used for development only; for example writing plugins. |
| --debug | Add 'debug' keyword**.** Used for development only. |
| --pre | Add 'pre' keyword. Use images with pre-loaded libraries. |

# XML FILES

The following describes the format and options for the XML files. An asterisk (\*) indicates an optional attribute.

## Manifest

The manifest.xml file is required to exist in the doc directory; it defines the documents to load and render. System wide variables are also declared.

### General outline

The content of the manifest.xml file normally consist of the following:

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE doc SYSTEM "manifest.dtd">

<doc>

<variable-list>…</variable-list>\*

<source-list>…</source-list>

<render-list>…</render-list>

</doc>

Source-list and render-list are required but only one of each may exist.

### Elements and Attributes

#### variable-list

Declares system-wide variables (most suitable for multi-page html output or multiple document types).

|  |  |  |  |
| --- | --- | --- | --- |
| Element | Description | Attribute | Description |
| variable | Identifies a variable to use during document creation. | key | The name of the variable. |
| eval\* | * Indicates whether code will need to be run. * Default “n” |

#### source-list

The XML document(s) to load and parse to be passed to the render(s).

|  |  |  |  |
| --- | --- | --- | --- |
| Element | Description | Attribute | Description |
| source | Identifies the source file (without extension) to use during document creation. | key | The name of the source file (without extension). |
| type\* | For specifying custom handling of a document. NOT FOR GENERAL USE. |

#### render-list

Specifies a list of renderers to use.

|  |  |  |  |
| --- | --- | --- | --- |
| Element | Description | Attribute | Description |
| render | Identifies the renderer to use. | type | html, pdf or markdown. NOTE: Only html is valid for general use. |
| compact\* | * Should additional html meta data be generated? * Default “n” |
| pretty\* | * Add additional spacing and line feeds? * Default “n” |
| file\* | Used for PDF generation. NOT FOR GENERAL USE. |

##### render-source

Indicates the source files to use for the renderer.

|  |  |  |  |
| --- | --- | --- | --- |
| Element | Description | Attribute | Description |
| render-source | Specifies the source for the renderer. | key | Usually the name of the source file but can be different for pgBackRest. |
| source\* | Special indicator for split reference data. NOT FOR GENERAL USE. |
| file\* | * The output filename which may include a path * Default “” meaning the key will be used as the name and the location will be the doc/output directory for the specific renderer. |
| menu | Used for multi-webpage output. NOT FOR GENERAL USE. |

## Source Files

Source XML files to render must be located in the doc/xml directory. A source file contains the XML to display and execute.

### General outline

A general outline of a source XML is described below.

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE doc SYSTEM "doc.dtd">

<doc title=”*doctitle*”>

<description>*description*</description>

*<!-- Optional variables to use in the document -->*

<variable-list>

<variable key=”var1”>variable1</variable>

</variable-list>

*<!-- Optional repeatable code block -->*

<block-define id=”*id1*”>…</block-define>

*<!-- Use of repeatable code block with replacement variable -->*

<block id="id1">

<block-variable-replace key="*varkey*">*newvalue*

</block-variable-replace>

</block>

*<!-- Document section -->*

<section id=”*sectionid1*”>

<title>*sectiontitle*</title>

<p>*text*</p>

*<!-- Initialize a Docker container -->*

<host-add name=”*hname*” user=”*huser*” image=”*himage*”>…</host-add>

*<!-- Configure Postgres settings -->*

<postgres-config host=”*hname*” file=”*pgfile*”>

<title>*pgtitl*e”</title>

<postgres-config-option key=”*pgkey*”>*pgvalue*

</postgres-config-option>

</postgres-config>

*<!-- Configure pgBackRest settings -->*

<backrest-config host=”*hname*” file=”*brfile*” owner=”*browner*”>

<title>*brtitl*e”</title>

<backrest-config-option section=”*brsection*”

key=”*brkey*”>*brvalue*</backrest-config-option>

</backrest-config>

*<!-- Execute commands in the host container -->*

<execute-list host=”*hname*”>

<title>*eltitle*</>

<execute>

<exe-cmd>*some command*</exe-cmd>

<exe-highlight>*highlight text*</exe-highlight>  
 </execute>

</execute-list>

</section>

</doc>

### Elements and Attributes

#### doc

The doc element is the outermost container.

|  |  |  |  |
| --- | --- | --- | --- |
| Element | Description | Attribute | Description |
| doc | Container element. | title | Title of the document to be displayed |
| subtitle\* | Document subtitle to be displayed. |
| toc\* | * Display table of contents? * Default “y” |
| toc-number\* | * Display number for headings? * Default “y” |
| cmd-line-len\* | * Allowable length of executable code lines * Default “80” |

For most documents, the following elements are contained within the doc element.

* description
* variable-list
* block-define
* section

#### description

Text to describe the document.

#### variable-list

Element in which variables are defined. The variable-list element must contain one or more variable elements.

#### variable

Variables used throughout the document are defined in this element. Once a variable is defined, it can be used in defining other variables, in command execution or other elements of the document by wrapping the variable key name in curly and square brackets, e.g. {[myvariable]}.

|  |  |  |  |
| --- | --- | --- | --- |
| Element | Description | Attribute | Description |
| variable | Identifies a variable to use during document creation. | key | The name of the variable. |
| eval\* | * Indicates whether code will need to be run. * Default “n” |
| keyword\* | * Indicates whether the variable is applicable when a keyword is specified. For example, if --keyword=xyz is passed on the command line when creating the document, then only variables that have specified keyword=xyz will be applicable. ??? default is no longer a keyword if a keyword is specified * Default “default” |

#### block-define

The block-define describes a repeatable set of elements. For example, if an installation requires the same steps to be performed on different machines, then a block-define can be used as a reference rather than copy and pasting the same set of elements in different sections of the document.

|  |  |  |  |
| --- | --- | --- | --- |
| Element | Description | Attribute | Description |
| block-define | Defines a a set of elements that can be referenced as a block. | id | Unique identifier. |

The block-define element can contain the following elements.

* p
* list
* table
* host-add
* execute-list
* backrest-config
* postgres-config
* cmd-description
* option-description
* code-block
* block

#### section

A document should contain at least one section. The section title is intended to be displayed in the Table of Contents. Sections can be nested.

|  |  |  |  |
| --- | --- | --- | --- |
| Element | Description | Attribute | Description |
| section | Contains a distinct group of elements. | id | Unique identifier. |
| depend\* | Id of the section that must be executed before this section. If none provided, defaults to previous section. |
| keyword\* | Keyword indicating the possibility of the section only being executed when the keyword is set. |
| source\* | ??? I don’t see this used |

The following elements are valid within a section.

* title
* p
* list
* table
* host-add
* execute-list
* backrest-config
* postgres-config
* cmd-description
* option-description
* code-block -- used to display code that is not executed
* block
* section

#### host-add

Hosts are Docker containers used to setup the environment for which the documentation will run on. Docker files are located in the docker directory and are used in the Vagrantfile to build the docker images from which containers will be created. Examples are co7-backup.docker (Centos 7 pgBackRest container), co7-db.docker (Centos 7 PostgreSQL container), etc. When setting up images and containers, it is recommended that you mimic as closely as possible to the environment for which the generated document will be applied.

The host-add indicates to Docker how to initialize the container. Host-add will connect to the image as the supplied user; this user is the default user for the container under which commands will run unless another user is specified for the command.

|  |  |  |  |
| --- | --- | --- | --- |
| Element | Description | Attribute | Description |
| host-add | Creates a Docker container. | name | Unique name. |
| user | Main user of the image. |
| image | Name of the image to connect to. |
| option\* | ??? |
| os\* | ??? these always appear to be used so do we know what would happen if they were not provided? |
| mount\* | ??? can this really be optional? |

#### execute-list

Defines a section in which commands will be executed on a specific host.

|  |  |  |  |
| --- | --- | --- | --- |
| Element | Description | Attribute | Description |
| execute-list | Defines a section in which commands will be executed on the specified host. | host | Host name. |
| keyword\* | Keyword indicating the possibility of the section only being executed when the keyword is set. |

The following required elements are valid within execute-list.

* title
* execute

#### execute

Contains the command to be executed. There can be many execute elements within an execute-list or host-add element but each execute element may only have one exe-cmd element.

|  |  |  |  |
| --- | --- | --- | --- |
| Element | Description | Attribute | Description |
| execute | Contains and defines the criteria for how the command will be executed. | keyword\* | Keyword indicating the possibility of the section only being executed when the keyword is set. |
| user\* | * Runs the command as the specified user. * Defaults to user defined when container is created. |
| filter\* | * Used with exe-highlight to display only a few lines surrounding the highlighted item(s). To show all output, set output=”y” filter=”n” * Default “y” |
| filter-context\* | * defines the number of lines displayed around the highlighted line * Default 3 |
| skip\* | Displays the command in the document but skips running the command |
| show\* | * Determines whether to display the command or not. The command will still be executed regardless of this setting. * Default “y” |
| output\* | * Determines if the results of an executed command should be displayed. If exe-highlight is used, it is automatically set to “y”. * Default “n” |
| err-suppress\* | * Indicates if an error should be ignored or not * Default “n” |
| err-expect\* | if there is an error code that is expected to be thrown, then this will look for it and highlight it in red. If an error code is thrown that is not the expected error, the doc builder will abort. If an error code is not known, use <exe-highlight-type>error</exe-highlight-type> prior to exe-highlight tag. |
| retry\* | The number of seconds during which the command will be retried until success or err-expect is returned. |
| variable-key\* | Loads the specified variable with the result of the command. The variable may then be used later in the document. |

The following elements are valid within the execute element. The exe-cmd element is required.

* exe-cmd
* exe-cmd-extra
* exe-highlight
* exe-highlight-type

##### exe-cmd

Specifies the command to execute on the command line. For example:

<exe-cmd>apt-get install libdbd-pg-perl</exe-cmd>

##### exe-cmd-extra

Specifies extra input for the command.

<exe-cmd-extra>-y 2>&amp;1</exe-cmd-extra>

##### exe-highlight

Highlights the line of output from the command containing the specified text. Highlight color is green unless exe-highlight-type is used. Use the vertical bar (|) to indicate highlighting different lines of output. The following will highlight the line containing “stanza” and the line containing “completed successfully”.

<exe-highlight>stanza|completed successfully</exe-highlight>

##### exe-highlight-type

Used prior to exe-highlight to change the highlight color to red when highlighting an error displayed in the output of the command.

<exe-highlight-type>error</exe-highlight-type>

<exe-highlight>could not find the database system</exe-highlight>

#### backrest-config

Creates the pgbackrest.conf file on the specified host.

|  |  |  |  |
| --- | --- | --- | --- |
| Element | Description | Attribute | Description |
| backrest-config | Creates the pgbackrest.conf file with one or more options. | host | Host name. |
| file | File path and name to create. |
| keyword\* | Keyword indicating the possibility of the section only being executed when the keyword is set. |
| owner\* | * Owner of the file. * Default postgres. |
| show\* | * Determines whether to display the command or not. The command will still be executed regardless of this setting. * Default “y” |
| reset\* | Reset all options to previous settings. |

The following elements are required:

* title
* backrest-config-option

#### block

A place holder indicating that the block-define elements identified by the id attribute should be inserted. This is useful for repeating sections, especially if a particular variable may be the only element changing.

|  |  |  |  |
| --- | --- | --- | --- |
| Element | Description | Attribute | Description |
| block | Section that will be replaced by a block-define section. | id | The id of the block-define element to replace this block with in final output of the document. |

The following elements are required:

* block-variable-replace

#### block-variable-replace

Used to temporarily replace a variable within a block.

|  |  |  |  |
| --- | --- | --- | --- |
| Element | Description | Attribute | Description |
| block-variable-replace | Replaces the variable specified by key. | key | The key of the variable element within the block to replace. |