t2

Report Templates

Version 2023.1: 13.5.2023

1 Introduction

PILAR generates reports in RTF or CSV formats. In order to specify the desired reports, you may use RTF templates or XML scripts.

1.1 RTF templates

Report templates generate RTF reports using a template written in RTF and merging the text with information provided by PILAR.

The template processor copies most of the text from the template into the report without modification. However, there may be placeholders to call PILAR for data.

The placeholders have the following syntax

<pilar> command(s) </pilar>

The format is XML, with a number of tags, to describe the action, and zero or more xml attributes.

attr

means

attr="value"

When the attribute is optional, it is presented within brackets

[attr]

and it means that it may be omitted in the placeholder, and PILAR will use a default value.

Most frequently, the attribute value comes between double quotes; but sometimes word is nor very friendly with those characters and tries to replace with left- and right- double quotes. In those cases, you may use whatever character you wish as attribute delimiter. For instance

attr=$value$

attr=%value%

attr=:value:

attr=(value)

attr=[value]

Please, avoid the following characters that are heavily used by RTF to encode formatting:

{ } </w:t>

and these ones, open and close quotes, that use to have special typographic variants:

“ ” " ‘ ’ ′ `

1.2 XML scripts

The reporting tags may be provided using an XML file. Here normal XML language is used.

Here follows an example that generates RTF:

```
<?xml version="1.0" encoding="UTF-8" ?>
<pilar format="rtf" >

<ask.domains />
<ask.phases />

<paragraph>
Valuation of risk per asset.
</paragraph>

<graph hw="accumulated risk/asset" />
<nl />
<legend />

<paragraph>
Top accumulated risks.
</paragraph>

<foreach_phase>
  <phase>Phase: [code] name</phase><nl/>
  <foreach_domain>
    <domain /><nl/>
    <accumulated top_R="20" sort="r" cols="tdir" />
    <nl/><nl/>
  </foreach_domain>
</foreach_phase>

</pilar>
```

You may change the first lines to generate a CSV file:

```
<?xml version="1.0" encoding="UTF-8" ?>
<pilar format="csv" >

<ask.domains />
… … …
```

PILAR silently ignores those commands that do not fit in a CSV output file.

2 XML tags

2.1 accumulated

```
<accumulated phase1 [asset] [domain] [domain_family] [dimension]
      [top_I] [top_L] [top_R] [sort] [cols]
      />
```

Generates a table with the columns selected, presenting the accumulated risk values in the given phase.

attr	default	meaning
phase	mandatory	the project phase
asset	all	presents only some assets
domain	base	presents only some domains
domain_family	empty	selects only those domains that are qualified with one of the mentioned domain classes
dimension	all	presents only some dimensions
threat	all	presents only some threats version: 7
cols columns	"ATDILR"	presents the selected columns (see codes below)
sort	"RIAT"	sorts data by the given criteria; default is first risk (R), then impact (I), then asset (A), then threat (T).
top_I	100	selects the entries with an impact in the top percent
top_L	100	selects the entries with a likelihood in the top percent
top_R	100	selects the entries with a risk in the top percent

See "*text formatting*".

To select columns to sort, and to print, the following codes are used:

A	asset
A.G	asset group
A.L	asset layer
B	
D	dimension
I0	potential impact
I	residual impact
L0	potential likelihood
L	residual likelihood
R0	potential risk

R	residual risk
S	sequence number
T	threat
T.T	threat group (top level)
V	value (accumulated)

Furthermore, you may set column headers as in the following example

```
<accumulated
  asset-text=(code)
  threat-text=(code)
  columns=(
    A: Asset,
    D: Dimension,
    T: Threat,
    I0: Impact,
    L0: Likelihood,
    R0: Risk,
    I: Residual impact,
    L: Residual Likelihood,
    R: Residual risk)
/>
```

2.2 Actions (security actions)

2.2.1 action

<action [id] [elements] />

Version 7.

Id selects a security action by identifier. It can be ignored within a for each frame.

Elements is a list of pieces to be listed. Default is

> “id, start, end, domain, measures, description, source, resources, status, comment”

2.2.2 action.id

<action.id [id] />

2.2.3 action.start

<action.start [id] />

2.2.4 action.end

<action.end [id] />

2.2.5 action.domain

<action.domain [id] />

2.2.6 action.measures
```
<action.measures [id] />
```

2.2.7 action.description
```
<action.description [id] />
```

2.2.8 action.source
```
<action.source [id] />
```

2.2.9 action.resources
```
<action.resources [id] />
```

2.2.10 action.status
```
<action.status [id] />
```

2.2.11 action.comment
```
<action.comment [id] />
```

2.2.12 action.table
<action [elements] [domain] [after] [before] [status] [prefix] [sources] />

Version 7.

Prints a table with the selected actions (rows) and the columns described in “elements”.

Rows are selected as in *foreach_action*

Columns are selected as in *action*

2.3 Ask

2.3.1 ask.domain
<ask.domain name />

Ask the user for 1 security domain. It will be called “name” in the template.

You may specify the title of the popup window

<ask.domain name > title </ask.domain>

2.3.2 ask.domains
<ask.domains [name] />

Ask the user for the security domains to be used in other patterns.

By default, PILAR uses all the domains.

It will be called “name” in the templates; or “” if no name is provided.

You may specify the title of the popup window

<ask.domains [name] > title </ask.domains>

2.3.3 ask.phase

<ask.phase name />

Ask the user for 1 project phase. It will be called “name” in the template.

You may specify the title of the popup window

<ask.phase name > title </ask.phase>

2.3.4 ask.phases

<ask.phases [name] />

Ask the user for the project phases to be used in other patterns.

By default, PILAR uses all the phases.

It will be called “name” in the templates; or “” if no name is provided.

You may specify the title of the popup window

<ask.phases [name] > title </ask.phases>

2.4 Assets

2.4.1 asset

<asset asset [text] />

Prints

[code] name

attr	default	meaning
asset	mandatory	the code of one asset
text	optional	See “*text formatting*

Example

<asset asset=(SERVER) />

2.4.2 asset.above

<asset.above asset />

Prints the list of assets that depend on this asset.

See "*asset*".

2.4.3 asset.below

<asset.below asset />

Prints the list of assets on which this asset depends.

See "*asset*".

2.4.4 asset.category

<asset.category asset />

Prints the "*category*" of the asset.

See "*asset*".

2.4.5 asset.classes

<asset.classes asset />

Prints the classes associated to the asset.

See "*asset*".

2.4.6 asset.code

<asset.code asset />

Prints the code.

See "*asset*".

2.4.7 asset.data

<asset.data asset />

Prints the administrative data of the asset.

See "*asset*".

2.4.8 asset.description

<asset.description asset />

Prints the description of the asset.

See "*asset*".

2.4.9 asset.domain

<asset.domain asset />

Prints the security domain of the asset.

See "*asset*".

2.4.10 asset.group

<asset.group asset />

Prints the code of the group that contains the asset, if any. Otherwise, blank.

See "*asset*".

2.4.11 asset.layer

<asset.layer asset />

Prints the code of the layer that contains the asset.

See "*asset*".

2.4.12 asset.name

<asset.name asset />

Prints the name of the asset.

See "*asset*".

2.4.13 asset.sources

<asset.sources asset />

Prints the list of information sources associated to the asset.

See "*asset*".

2.5 Assets

2.5.1 Assets attributes

The attributes filter the assets that match

- the code
- the layer
- the security domain
- a security domain of a given family
- an information source

attr	default	meaning
asset	all	list of asset codes, or names of groups of assets
domain	all	list of security domain codes
domain_family	none	selects only the domains that are qualified with one or more of the mentioned classes
family	all	list of class codes
layer	all	list of layer codes
source	none	list of sources to match
admdata	any	filters by data in administrative attributes (see *adm data filters*

		version: 7
by	layer	by = (layer) Assets are sorted by layer. by = (domain) Assets are sorted by domain.
under	any	given a collection of assets, retains only those below; that is, those with a dependency relation from 'under' assets

Example

- family=(HW)
  all the assets of class HW
- family=(HW) domain=(base)
  all the assets of class HW in the base domain
- asset=(SVR_01, SVR_02)
  those two assets
- under=(INFO)
  those that INFO depends on

2.5.2 assets.description

```
<assets.description [by]
    [asset] [family] [layer] [domain] [domain_family] [source] [under] >
    <print { attributes } />
</assets.description>
```

Prints the description of the selected assets. It prints a few paragraphs, with all the identification of the asset.

"*Assets / attributes*" are used to select the assets to process.

For each selected asset, you may control the information that is printed:

attribute	default	meaning
assets_below	false	prints the set of assets directly or indirectly below, according to dependencies
category	false	prints the category See "*category*
classes	true	prints the classes associated
data	true	prints the administrative data
dependencies	false	prints the dependencies between assets
description	true	prints the description

sources	true	prints the sources of information
values	false	prints the values

2.5.3 assets.group

<assets.group name

[asset] [family] [layer] [domain] [domain_family] [source] [under]

/>

Defines a group of assets, and it may be later identified by the given name.

See "*Assets / attributes*".

2.5.4 assets.list

<assets.list [by]

[asset] [family] [layer] [domain] [domain_family] [source] [under]

/>

Prints the selected assets. It prints one line per asset: [code] name.

See "*Assets / attributes*".

assets

2.5.5 assets.valuation

<assets.valuation [by]

[asset] [family] [layer] [domain] [domain_family] [source] [under]

/>

Prints the valuation of the selected assets. It prints the value and the valuation criteria. You may skip criteria using the attribute

criteria= (off)

See "*Assets / attributes*".

2.6 category

<category [domain] [domain_family] />

The category of an information system is defined as the highest valuation of its assets in any dimension of security.

The following table shows the algorithm:

category	**if …**
HIGH	if valuation of some asset in the domain is higher than [5]
MEDIUM	if valuation of some asset in the domain is higher than [2]
LOW	if valuation of some asset in the domain is higher than [0]

attr	**default**	**meaning**

domain	all	list of security domain codes
domain_family	none	selects only the domains that are qualified with one or more of the mentioned classes

2.7 comment

<comment> anything </comment>

<comment.xxx> anything </comment.xxx>

Content is ignored.

2.8 comments

<comments slice > foreach </>

Foreach may be *foreach_control* or *foreach_safeguard*. There may be several iterators.

Prints the comments associated to each safeguard or control.

attr	default	meaning
slice	domain	**domain** first, sort by security domains, then by control, then by phase **phase** first, sort by phase, then by control, then by security domains. **control** first by control, then phase, then security domain **safeguard** first by safeguard, then phase, then security domain

Example

```
<comments slice=(domain) >
  <foreach_control evl=(ens:2015)
    pattern=(org.*)
    applies=(all) />
</comments>
```

2.9 control

<control evl control [text] />

Prints, for the specified control in the specified security profile

[code] name

attr	default	meaning
evl	mandatory	the code of one security profile
control	mandatory	the code of one control in the security profile
text	optional	See "*text formatting*

2.9.1 control.code

<control.code evl control />

Prints the code of the specified control in the specified security profile

See "*control*".

2.9.2 control.name

<control.name evl control />

Prints the name of the specified control in the specified security profile

See "*control*".

2.9.3 control.applies

<control.applies evl control [domain] [stage] />

Prints whether the control applies or not in the given security domain.

See "*control*".

2.9.4 control.comment

<control.comment evl control [domain] [phase]/>

Prints the comment associated to the specified control in the given security domain end project phase.

See "*control*".

2.10 criticality.list

<criticality.list />

Prints a list of criticality values. These values are used to valuate risk in qualitative assessments.

2.11 date

<date [format] />

Prints the current date.

attr	default	meaning
format	"day.month.year"	You may specify a format as in http://java.sun.com/javase/6/docs/api/java/util/Formatter.html PILAR will pass a single argument that is the current date.

Example:

pattern	output

`<pilar><date /></pilar>`	4.12.2009
`<pilar><date format=(%tc) /></pilar>`	Fri Dec 04 16:34:01 CET 2009
`<pilar><date format=(%tD) /></pilar>`	12/04/09
`<pilar><date format=(%tY/%<tm/%<td) /></pilar>`	2009/12/04

2.12 define

<define kw > value </define>

It associates the name "kw" with the "value".

attr	default	meaning
kw	mandatory	term defined

In the "*replace* command, the "kw" will be replaced.

Example:

<pilar><define kw=(WS)>William Shakespeare</define></pilar>

2.12.1 define macro

You may associate a name to a piece of text, to be replaced later. It is useful for repetitive text, and for long strings that may be given a short and meaningful name.

Example:

<pilar><define key=(t2)>control.name evl=(29151:2017)</></pilar>

Later:

<pilar><[[t2]] control=(A.1) /></pilar>

The text is copied as is, before normal parsing of the rule, that is, the last line is expanded into:

<pilar>< control.name evl=(29151:2017) control=(A.1) /></pilar>

2.13 deflected

<deflected phase1 [asset] [domain] [domain_family] [dimension]
[top_I] [top_L] [top_R] [sort] [cols] />

Generates a table with the columns selected, presenting the deflected values in the given phase.

attr	default	meaning
phase	mandatory	the project phase
asset	all	presents only some assets
domain	base	presents only some domains

domain_family	none	selects only the domains that are qualified with one or more of the mentioned classes
dimension	all	presents only some dimensions
cols columns	"ABTDILR"	presents the selected columns (see codes below)
sort	"RIABT"	sorts data by the given criteria; default is first risk (R), then impact (I), then asset (A), then threat (T).
top_I	100	selects the entries with an impact in the top percent
top_L	100	selects the entries with a likelihood in the top percent
top_R	100	selects the entries with a risk in the top percent

See "*text formatting*".

To select columns to sort, and to print, the following codes are used:

A	asset above
B	asset below
D	dimension
I	impact
I0	potential impact
L	likelihood
L0	potential likelihood
R	risk
R0	potential risk
S	sequential number
T	threat

Furthermore, you may set column headers as in the following example

```
<deflected
  asset-text=(code)
  threat-text=(code)
  columns=(
    A: Business asset,
    D: Dimension,
    B: Supporting asset,
    T: Threat,
    I0: Impact,
    L0: Likelihood,
    R0: Risk,
    I: Residual impact,
    L: Residual likelihood,
```

```
        R: Residual risk)
/>
```

2.14 Dimension

2.14.1 Dimension codes

Dimensions receive different names in different languages.

When looking for the dimension matching a code, PILAR uses the codes presented below, either the universal code or the language.code forms.

Dimension codes						
universal	English		Español		Italiano	
D	en.A	Availability	es.D	Disponibilidad	it.D	Disponibilità
I	en.I	Integrity	es.I	Integridad	it.I	Integrità
C	en.C	Confidentiality	es.C	Confidencialidad	it.R	Riservatezza
A	en.Auth	Authenticity	es.A	Autenticidad	it.A	Authenticità
T	en.Acc	Accountability	es.T	Trazabilidad	it.T	Tracciabilità

English code is always available; while language specific codes are load according to the language specified in the .CAR file when PILAR starts.

2.14.2 dimension

<dimension dimension [text] />

Prints

[code] name

attr	**default**	**meaning**
dimension	mandatory	the code of one dimension See "*Dimension / code*
text	optional	See "*text formatting*

Example

<dimension dimension=(D) />

2.14.3 dimension.code

<dimension.code dimension />

Prints the code of the dimension.

See "*dimension*".

2.14.4 dimension.name

<dimension.name dimension />

Prints the name of the dimension.

See "*dimension*".

2.14.5 dimensions.list

<dimensions.list />

Prints the list of dimensions that are not OFF.

2.15 Domains, security domains

2.15.1 domain

<domain domain [text] />

Prints

[code] domain

attr	default	meaning
domain	mandatory	the code of one security domain
text	optional	See "*text formatting*

2.15.2 domain.category

<domain.category domain />

Prints the category of the domain. It is the highest "*category*" of the assets in the domain.

See "*domain*".

2.15.3 domain.code

<domain.code domain />

Prints the code of the domain.

See "*domain*".

2.15.4 domain.description

<domain.description domain />

Prints the description of the domain.

See "*domain*".

2.15.5 domain.families

<model.families domain [families] />

Prints a list of the family types in the domain. Optionally, provide an argument with a comma-separated list of types; then only subtypes of those are listed.

2.15.6 domain.name

<domain.name domain />

Prints the name of the domain.

See “*domain*”.

2.15.7 domain.operation_mode

<domain.operation_mode domain />

Prints the operation mode of the domain.

See “*domain*”.

2.15.8 domains.description

<domains.description [domain] [domain_family] />

Lists the description of the security domains.

attr	default	meaning
domain	all	list of security domain codes
domain_family	none	selects only the domains that are qualified with one or more of the mentioned classes

2.15.9 domains.group

<domains.group [name] [domain] [domain_family] />

Defines a group of security domains, and it may be later identified by the given name.

attr	default	meaning
name	“”	the defined group
domain	all	list of security domain codes
domain_family	none	selects only the domains that are qualified with one or more of the mentioned classes

There is an empty name, that is used by default by every tag that requires one or more domains.

2.15.10 domains.list

<domains.list [domain] [domain_family] />

Lists the security domains in the model.

attr	default	meaning
domain	all	list of security domain codes
domain_family	none	selects only the domains that are qualified with one or more of the mentioned classes

2.15.11 domains.valuation

<domains.valuation [domain] [domain_family] />

Lists the valuation of the domains.

attr	default	meaning
domain	all	list of security domain codes
domain_family	none	selects only the domains that are qualified with one or more of the mentioned classes

2.15.12 domains.vulnerability

```
<domains.vulnerability [domain] [domain_family] />
```

List the aggravating | mitigating factors identified per domain.

attr	default	meaning
domain	all	list of security domain codes
domain_family	none	selects only the domains that are qualified with one or more of the mentioned classes

2.16 ENS – Esquema Nacional de Seguridad (España)

Specific for the Spanish ENS security profile.

2.16.1 ens.compliance

```
<ens.compliance evl pattern [source] [applies]
    [slice] [comments]
    [domain] [domain_family] [phase] [mode] />

<ens.compliance evl [control] [source] [applies]
    [depth] [questions] [safeguards]
    [slice] [comments]
    [domain] [domain_family] [phase] [mode] />

<ens.compliance evl [source] [applies]
    [slice] [comments]
    [domain] [domain_family] [phase] [mode] >
  path_expression
</ens.compliance>
```

See "*evl.valuation*".

2.17 EVL – Security profiles

2.17.1 EVL attributes

The rows that are printed are those for controls selected by one of the following means using either the "rows" tag or directly in the "evl…" tag:

	attribute	description
option 1	pattern	selects by code pattern
	applies	filters those that do [not] apply
	sources	filters those associated to the given sources
option 2	path	selects by code paths
	applies	filters those that do [not] apply
	sources	filters those associated to the given sources
option 3	control	selects my control code
	depth	selects down to a given depth in the evl tree
	expertise	selects down to a given expertise in the evl tree
	questions	follows questions in the tree
	safeguards	follows safeguards in the tree

Examples

```
<evl.valuation evl=(27002:2013) >
  <rows pattern=(6.1.*) />
</>

<evl.list evl=(ens:2015) sources=(info_owner, service_owner)  />
```

attribute	**default**	**meaning**
evl	mandatory	identifies the security profile by its code
applies	yes	only prints the controls that apply, or not: yes – only those that apply no – only those that do not apply all – all
control	all	list of control codes
depth	all	prunes the tree to the selected depth
pattern	none	selects controls that match the pattern
questions	no	determines whether questions are listed or not

		boolean: yes / no, true / false
safeguards	no	determines whether safeguards are listed or not boolean: yes / no, true / false
source	none	only those associated to one of the given sources

Other attributes drive report generation:

attr	**default**	**meaning**
comments	2	to print comments along the controls 0 – comments are not printed 1 – comments are printed in-line, close to the control 2 – comments are printed afterwards as a hierarchy 3 – comments are printed using calls
domain	all	list of security domain codes
domain_family	none	selects only the domains that are qualified with one or more of the mentioned classes
light	all	you may specify two phases; e.g. light=(phase1, phase2) or you may specify none, and PILAR will choose first and last phases light=() PILAR presents a coloured square with the difference between phase1 and phase2: • RED: far below • YELLOW: below • GREEN: equals • BLUE: above
mode	percent	selects the format to print control and safeguard values (see below)
phase	all	list of phases

Mode parameter controls the presentation of maturity values:

mode value	prints	example
0 | M | maturity	control maturity [range]	L1-L3
1 | M_M	control and safeguard maturity [range]	L1-L3 (L2-L3)
2 | _M	safeguard maturity [range]	(L2-L3)
3 | MA	control approximate maturity	L2+
4 | MA_MA	control and safeguard approximate maturity	L2+ (L3-)
5 | _MA	safeguard approximate maturity	(L3-)
6 | P | percent	safeguard percent (in ENS; compliance percent)	66%
7 | P_P	control percent (safeguard percent)	45% (66%)
8 | P_	control percent	45%
9 | C	control compliance for some profiles with formal compliance statements	PASS

2.17.2 evl

`<evl evl [text] />`

Prints

[code] name

attr	default	meaning
evl	mandatory	the code of one security profile
text	optional	See "*text formatting*

2.17.3 evl.code

`<evl.code evl />`

Prints the code of the security profile.

See "*evl*".

2.17.4 evl.name

`<evl.name evl />`

Prints the name of the security profile.

See "*evl*".

2.17.5 evl.compliance

See *evl.valuation*

2.17.6 evl.group

```
<evl.group evl name [control] />
```

Groups some controls under a single name.

attr	default	meaning
evl	mandatory	the code of one security profile
name	mandatory	the name of the group
control	all	a list of control codes

2.17.7 evl.list

```
<evl.list evl pattern [source] [applies]
    [domain] [domain_family] [comments] />

<evl.list evl [control] [source] [applies]
    [depth] [questions] [safeguards]
    [domain] [domain_family] [comments] />

<evl.list evl [source] [applies]
    [domain] [domain_family] [comments] >
  path_expression
</evl.list>
```

You may use the *evl/rows*tag to select rows.

Inserts a list with the controls that apply from the evaluation profile "code", in the requested domains.

You may select some controls, and dig downwards (see "*EVL / attributes*").

You may select by pattern (see "*Patterns*")

You may select by paths (see "*Paths*")

2.17.8 evl.required

```
<evl.required evl pattern [source] [applies]
    [domain] [domain_family] [comments] />

<evl.required evl [control] [source] [applies]
    [depth] [questions] [safeguards]
    [domain] [domain_family] [comments] />
```

```
<evl.required evl [source] [applies]
    [domain] [domain_family] [comments] >
  path_expression
</evl.required>
```

You may use the *evl/rows* and *evl/cols* tags to select rows and columns.

Inserts a table with the applicability of the controls.

You may select some controls and dig downwards (see "*EVL / attributes*").

You may select by pattern (see "*Patterns*").

You may select by paths (see "*Paths*").

2.17.9 evl.valuation

```
<evl.valuation evl pattern [source] [applies]
    [slice] [comments]
    [domain] [domain_family] [phase] [mode] />

<evl.valuation evl [control] [source] [applies]
    [depth] [questions] [safeguards]
    [slice] [comments]
    [domain] [domain_family] [phase] [mode] />

<evl.valuation evl [source] [applies]
    [slice]
    [domain] [domain_family] [phase] [mode] [comments] >
  path_expression
</evl.valuation>
```

You may use the *evl/rows* and *evl/cols* tags to select rows and columns.

Inserts a table with the valuation of the controls from the evaluation profile "evl", in the requested domains, down to the requested depth.

"slice" is used to select one table by security domain or one table by project phase.

attr	default	meaning
slice	domain	**domain** there are as many tables as security domains, and in each table, as many columns as project phases. **phase**

		there are as many tables as project phases, and in each table, as many columns as security domains.

You may select some controls, and dig downwards (see "*EVL / attributes*").

You may select by pattern (see "*Patterns*").

You may select by path (see "*Paths*").

2.17.10 rows

In "evl…" tags, you may select controls (to appear as rows) as explained in *EVL / attributes*

2.17.11 cols

In "evl…" tags that render a table, you may select the columns to show. Please, note that the first column, "control", is always present.

Example

```
<evl.valuation evl=(27002:2013) comments=(0) >
      <rows pattern=(6.1.*) />
      <cols>
            <applies />
            <phases />
            <sources />
            <sources sources=(info_owner) />
      </>
</>
```

It shows several columns: the first one, "controls", plus

- one column that shows the applicability of the control
- one column for each phase
- one column for information sources
- one column for information sources, but only if the information source is the given one

Column options

applies	it does not accept any other attribute
light	accepts attributes as in *EVL / attributes*
phases	generates as many columns as selected phases accepts one attribute "phases" that is a comma separated list of phase codes use null to refer to the potential values if there is no refining attribute, all phases are shown

domains	attributes • domains: a comma-separated list of security domain codes • domain-family: a list of domain family codes if there is no refining attribute, all domains are shown
source	shows a column with the codes of the information sources associated to the control if there is no attribute, any information source is listed if there is some source specified in the “sources” attribute, only these are reported

2.17.12 Examples

2.17.12.1 Example

<evl.valuation evl=(27002:2013) pattern=(6.1.*) slice=(domain) mode=(maturity) />

control	current	target	PILAR
[6] ORGANIZATION OF INFORMATION SECURITY	L0-L5	L4-L5	L2-L4
[6.1] INTERNAL ORGANIZATION	L0-L5	L4-L5	L2-L4
[6.1.1] Information security roles and responsibilities	L0-L5	L4-L5	L2-L3
[6.1.2] Segregation of duties	L2	L5	L4
[6.1.3] Contact with authorities	L5	L5	L3
[6.1.4] Contact with special interest groups	L5	L5	L3
[6.1.5] Information security in project management	L1	L4	L2

2.17.12.2 Example

<evl.valuation evl=(27002:2013) pattern=(6.1.*) slice=(phase) mode=(maturity) />

control	base	bps
[6] ORGANIZATION OF INFORMATION SECURITY	L0-L5	L0-L5
[6.1] INTERNAL ORGANIZATION	L0-L5	L0-L5
[6.1.1] Information security roles and responsibilities	L0-L5	L0-L5
[6.1.2] Segregation of duties	L2	L2
[6.1.3] Contact with authorities	L5	L5
[6.1.4] Contact with special interest groups	L5	L5
[6.1.5] Information security in project management	L1	L1

2.17.12.3 Example

<evl.valuation evl=(27002:2013) control=(6.1.1, 6.1.3) slice=(domain) mode=(maturity) />

control	current	target	PILAR
[6] ORGANIZATION OF INFORMATION SECURITY	L0-L5	L4-L5	L2-L4
[6.1] INTERNAL ORGANIZATION	L0-L5	L4-L5	L2-L4
[6.1.1] Information security roles and responsibilities	L0-L5	L4-L5	L2-L3
[6.1.3] Contact with authorities	L5	L5	L3

2.17.12.4 Example

```
<evl.valuation evl=(27002:2013) slice=(domain) mode=(maturity) >
  <path><name>6</name><name>6.1</name><pattern>*.*.*</pattern></path>
</>
```

control	current	target	PILAR
[6] ORGANIZATION OF INFORMATION SECURITY	L0-L5	L4-L5	L2-L4
[6.1] INTERNAL ORGANIZATION	L0-L5	L4-L5	L2-L4
[6.1.1] Information security roles and responsibilities	L0-L5	L4-L5	L2-L3
[6.1.2] Segregation of duties	L2	L5	L4
[6.1.3] Contact with authorities	L5	L5	L3
[6.1.4] Contact with special interest groups	L5	L5	L3
[6.1.5] Information security in project management	L1	L4	L2

2.17.12.5 Example

```
<evl.valuation evl=(27002:2013) slice=(domain) mode=(maturity) >
  <rows pattern=(6.1.*) />
  <cols>
    <applies />
    <sources />
  </cols>
</>
```

control	applies	sources
[6] ORGANIZATION OF INFORMATION SECURITY	yes	
[6.1] INTERNAL ORGANIZATION	yes	
[6.1.1] Information security roles and responsibilities	yes	info_owner
[6.1.2] Segregation of duties	yes	info_owner, service_owner
[6.1.3] Contact with authorities	yes	service_owner
[6.1.4] Contact with special interest groups	yes	

2.18 foreach

There are several <foreach_XXX> tags to iterate over some elements of the risk assessment.

These iterations are used in ***table***, in ***graph***, and simply to loop over some other tags.

The item in each iteration may have an implicit name (hidden) or its name made explicit to be sure what is PILAR referring to in each iteration.

Example

anonymous item	named item

<foreach_asset >
  code: <asset.code /><nl/>
  name: <asset.name /><nl/>
  <nl/>
</foreach_asset>
	<foreach_asset id=($A) >
	  code: <asset.code name=($A) /><nl/>
	  name: <asset.name name=($A) /><nl/>
	  <nl/>
	</foreach_asset>

Example

```
<foreach_asset family=(COM) >
  <asset.code/>: <asset.name /> <nl/>
</foreach_asset>
```

Example

```
<foreach_asset id=($A) family=(essential) >
  <foreach_dimension id=($D) dimension=(I, C, es:D) >
    <asset name=($A) />
     value: <asset.value name=($A) />
    (<dimension.code name=($D) />)<nl/>
  </foreach_dimension>
</foreach_asset>
```

2.18.1 foreach_action

Version 7.

<foreach_action [domain] [after] [before] [status] [prefix] [sources] >

** action**

</foreach_action>

Filters:

attr	**default**	**meaning**
domain	all	presents only some security domains
after	-	presents only actions alive after a given date
before	-	presents only actions alive before a given date
status	all	presents only actions in some status { Planned, Ongoing, Suspended, Done }
prefix	-	presents only those actions with a given prefix in its identifier.
sources	-	presents only those actions labelled with one of the sources

An action is alive in the period between its start, and its end. therefore, after-before define a period that must match the start-end period for the action to be selected.

See *foreach*

2.18.2 foreach_asset

```
<foreach_asset
    [asset] [domain] [domain_family] [family] [layer] [source] [under]
    [text] >
  action
</foreach_asset>
```

Selects some assets. See "*Assets / attributes*".

See *foreach*

See "*text formatting*".

2.18.3 foreach_control

```
<foreach_control evl [applies] [source] [text]
    [control] [expertise] [depth]
    [pattern] >
  action
</foreach_control>

<foreach_control evl [applies] [source] [text] >
  action
  <path>...</path>
</foreach_control>
```

Selects some controls from a security profile.

attr	default	meaning
evl	mandatory	selects the security profile
applies	yes	only prints those that apply, or not: yes – only those that apply no – only those that do not apply all – all
control	all	selects a list of control codes
depth	all	depth to list
expertise	project expertise	expertise may be a number or a keyword 0 – basic 1 – medium

		2 - expert By specifying an expertise, you determine de depth of the tree presented.
pattern	none	Filters only those that match any of a set of "*Patterns*
path	none	Filters only those that match any of a set of "*Paths*
sources	none	Filters only those associated to some of the given information sources

See *foreach*

See "*text formatting*".

2.18.4 foreach_dimension

```
<foreach_dimension [dimension] [text] >
  action
</foreach_dimension>
```

Selects some dimensions.

attr	default	meaning
dimension	all	a list of dimension codes See "*Dimension / code*

See *foreach*

See "*text formatting*".

2.18.5 foreach_domain

```
<foreach_domain [domain] [domain_family] [text] >
  action
</foreach_domain>
```

Selects some security domains.

attr	default	meaning
domain	all	a list of security domains
domain_family	none	selects only the domains that are qualified with one or more of the mentioned classes

See *foreach*

See "*text formatting*".

~~2.18.6 foreach_element~~

~~It is useful in tables to list some explicit attributes.~~

~~Example:~~

```
<table>
  <foreach_asset />
  <foreach_element headers=(code name group layer) >
    <asset.code />
    <asset.name />
    <asset.group />
    <asset.layer />
  </>
</table>
```

~~The headers attribute is optional, if none is provided, or there are less headers than elements in the list, it is left empty.~~

~~Within foreach_element you may refer to assets, controls, domains, phases, safeguards, sources, and threats.~~

2.18.7 foreach_evl

```
<foreach_evl [evl] [text] >
  action
</foreach_evl>
```

Selects some security profiles.

attr	default	meaning
evl	all	a list of security profiles

See *foreach*

See "*text formatting*".

2.18.8 foreach_phase

```
<foreach_phase [phase] [text] >
  action
</foreach_phase>
```

Selects some project phases.

attr	default	meaning
phase	all	a list of phases

See *foreach*

See "*text formatting*".

2.18.9 foreach_policy

```
<foreach_policy [safeguard] [expertise] [depth] [applies] [pattern]
    [text] >
```

```
  action
</foreach_policy>
```

See "*foreach_safeguard*".

See *foreach*

2.18.10 foreach_procedure

```
<foreach_procedure [safeguard] [expertise] [depth]
                   [applies] [pattern] [text] >
  action
</foreach_procedure>
```

See "*foreach_safeguard*".

See *foreach*

2.18.11 foreach_safeguard

```
<foreach_safeguard [safeguard] [expertise] [depth]
                   [applies] [pattern] [text] >
  action
</foreach_safeguard>
```

Selects some safeguards.

attr	default	meaning
applies	yes	only prints those that apply, or not: yes – only those that apply no – only those that do not apply all – all
depth	all	depth to list
expertise	project expertise	expertise may be a number or a keyword 0 – basic 1 – medium 2 - expert By specifying an expertise, you determine de depth of the tree presented.
pattern	none	Filters only those that match any of a set of "*Patterns*
safeguards	all	selects a list of safeguard codes

See *foreach*

See "*text formatting*".

2.18.12 foreach_scenario

Version 7.

```
<foreach_scenario [asset] [dimension] [threat] [prefix] >
  action
</foreach_scenario>
```

Selects some assets. See "*Assets / attributes*".

Filters:

attr	default	meaning
dimension	all	presents only some dimensions
threat	all	presents only some threats
prefix	-	presents only those risks with a given prefix in its identifier.

See *foreach*

2.18.13 foreach_stage

```
<foreach_stage [stage] >
  action
</foreach_stage >
```

Selects some applicability stages.

attr	default	meaning
stage	base	a list of applicability stages

See *foreach*

2.18.14 foreach_step

```
<foreach_step [step] >
  action
</foreach_step>
```

Selects some time intervals.

attr	default	meaning
step	those defined in the project	a list of time intervals

See *foreach*

2.19 graph

A graph is a collection of series. Each series provides a mapping of an X value to a Y value.

For instance, if we have the valuation of the assets in a project:

asset	[D]	[I]	[C]	[A]	[T]
[EI_info]		[5]	[6]	[5]	[5]
[ES_local]	[5]			[7]	[6]
[ES_remote]	[3]			[7]	[6]

There we have 3 series, one per asset. The X-axis shows dimensions, while the Y-axis shows qualitative valuation. The first series, for asset [EI_INFO] provides the following mapping

D → []; I → [5]; C → [6]; A → [5]; T → [5]

In PILAR we need to specify

```
<graph graph what [width] [height] >
  X-axis
  series
</graph>
```

where X-axis and the series to show are described as foreach iterations.

Attribute “graph” determines the type of graph:

graph	meaning
radar.lines	
radar.areas	
radar.sectors	
lines.vertical	
lines.horizontal	
bars.vertical	
bars.horizontal	
stack.vertical	
stack.horizontal	
pareto	

Attribute “what” specifies the contents; that is the value for Y-axis

what	meaning
value own	the explicit value for an asset
valua.accumulated	the accumulated value for an asset
safeguard.valuation	the maturity value for the safeguard; you need to determine

	domain – one security domain phase – one phase safeguard – one safeguard
evl.valuation	the valuation of the controls in a security profile you need to determine you need to determine evl – one security profile domain – one security domain phase – one phase safeguard – one safeguard
impact.accumulated	the accumulated impact
impact.deflected	the deflected impact
risk.accumulated	the accumulated risk
risk.deflected	the deflected risk

"width" adjusts the width of the graph.

"height" adjusts the height of the graph.

units	**example**
centimetres	width=(15cm)
millimetres	width=(150mm)
inches	width=(5.9in)
points	width=(425.19pt)

Lastly, for the series and the X-axis, you may provide a format for printing the text. For the series, the text is printed in the "*legend*".

E.g.

```
<foreach_dimension text=(code – name)/>
<foreach_asset text=(code: name)/>
```

See "*text formatting*".

Example

```
<graph what=(value.own) graph=(bars.vertical) width=(15cm) >
  <foreach_dimension />
  <foreach_asset text=(code: name)/>
</graph>
```

Each asset creates a numerical series, mapping a dimension (X) to a value (Y).

2.19.1 Hardwired graphs

<graph hw graph [width] [height]/>

Hw	
value / domain	Valuation of security domains.
value / asset	Valuation of individual assets.
accumulated impact / asset	Shows the evolution of impact along phases, asset by asset.
accumulated impact / dimension	Shows the evolution of impact along phases, asset by asset.
accumulated risk / asset	Shows the evolution of risk along phases, asset by asset.
accumulated risk / dimension	Shows the evolution of risk along phases, asset by asset. The tree-map displays an area that is proportional to the risk on the asset shown on the label.
accumulated risk / dimension / phase	Shows the distribution of risk in one dimension in one phase, asset by asset.
deflected impact	Shows the evolution of impact along phases, asset by asset.
deflected risk	Shows the evolution of risk along phases, asset by asset.

graph	selects the type of graph; see *graph*

You may select security domains, assets, dimensions, and phases as need for the graph.

Example

```
<graph hw="accumulated risk/asset" phases="null, target" />
<nl />
<legend />
```

2.19.2 heatmap

<heatmap [phases]
[asset] [domain] [domain_family] [dimension] [threat] [under]
[reference] [top_I] [top_L] [top_R]
[width] [height]
/>

Generates a table with the columns selected, presenting the accumulated risk values in the given phase.

attr	**default**	**meaning**
phases	all	project phases

asset	all	presents only some assets
domain	base	presents only some domains
domain_family	empty	selects only those domains that are qualified with one of the mentioned domain classes
dimension	all	presents only some dimensions
threat	all	presents only some threats version: 7
under	any	assets that are under those selected in this attribute; under means that there is dependency from the assets above and the asset selected for the map
reference	none	for top_xxx attributes, a reference phase is needed
top_I	100	selects the entries with an impact in the top percent
top_L	100	selects the entries with a likelihood in the top percent
top_R	100	selects the entries with a risk in the top percent
width		see *graph*
height		see *graph*

Example

```
<heatmap asset=(SVR) reference=(current) top_R=(10)/>
```

Example

```
<heatmap under=(INFO) reference=(target) top_R=(10)/>
```

2.20 legend

<legend />

Prints the legend of the last graph.

The legend is the name of the series in the graph.

The legend presents the colour used, along with the code and name of the element in the series.

Series are created by means of “foreach” tags. In this tag you may specify a “text” attribute that is used to print the textual form, replacing

‘code’ for the code of the element

‘name’ for the name of the element

E.g.

```
<foreach_dimension text=(code – name)/>
<foreach_asset text=(code: name)/>
```

2.21 marking

<marking />

Print the classification marking of the document.

2.22 maturity.list

<maturity.list />

Prints the list of maturity levels.

2.23 Models, projects

2.23.1 model

<model [text] />

Prints

[project code] project name

See "*text formatting*".

2.23.2 model.code

<model.code />

Prints the code of the proyect.

2.23.3 model.data

<model.data />

Prints a table with the administrative data of the project.

2.23.4 model.description

<model.description />

Prints the description of the project.

2.23.5 model.families

<model.families [families] />

Prints a list of the family types in the model. Optionally, provide an argument with a comma-separated list of types; then only subtypes of those are listed.

2.23.6 model.name

<model.name />

Prints the name of the project.

2.23.7 model.operation_mode

<model.operation_mode />

Prints the mode of operation of the project.

2.24 nl

<nl />

Starts a new line.

2.25 page

<page />

Starts a new page.

2.26 pattern

<pattern name … **/>**

Inlines the pattern with the given name. See *patterns* to load patterns.

The attributes given in this command are copied onto the pattern.

Example:

```
<pilar>
<pattern name=(asset.valuation.graph) />
</pilar>
```

2.27 patterns

<patterns file />

Loads a collection of patterns from an XML file. The file has the following format:

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

  <pattern name="asset.valuation.graph">
    <graph what="value.own"
           graph="bars.horizontal"
           width="15cm" >
      <foreach_asset family="essential" />
      <foreach_dimension />
    </graph>
    <legend />
  </pattern>

  <pattern name="asset.valuation.table">
    <asset.valuation />
```

```
  </pattern>

etc.
```

Each pattern has a unique name, which may be referenced from *pattern* commands.

2.28 Phases, project phases

2.28.1 phase

<phase phase [text] />

Prints

[code] name

attr	default	meaning
phase	mandatory	the code of one project phase
text	optional	See "*text formatting*

2.28.2 phase.code

<phase.code phase />

Prints the code of the phase.

See "*phase*".

2.28.3 phase.date

<phase.date phase />

Prints the date of the phase.

See "*phase*".

2.28.4 phase.description

<phase.description phase />

Prints the description of the phase.

See "*phase*".

2.28.5 phase.name

<phase.name phase />

Prints the name of the phase.

See "*phase*".

2.28.6 phase.sources

<phase.sources phase />

Prints the list of information sources associated to the phase.

See "*phase*".

2.29 Phases

Phases may be referenced by code, or by relative position.

When using a relative position

0 – stands for potential

1 – stands for first user phase

2 – stands for second user phase

-1 – stands for last user phase

-2 – stands for next to last user phase

etc

2.29.1 phases.description

<phases.description [phase] />

Lists the description of the project phases.

attr	default	meaning
phase	all	list of project phase codes

2.29.2 phases.group

<phases.group [name] [phase] />

Defines a group of phases, and it may be later identified by the given name.

attr	default	meaning
name	""	the defined group
phase	all	list of project phase codes

There is an empty name, that is used by default by every tag that requires one or more phases.

2.29.3 phases.list

<phases.list />

Lists the project phases.

2.30 Policies, security policies

2.30.1 Policies attributes

attr	default	meaning
domain	all	list of security domain codes
domain_family	none	selects only the domains that are qualified with one or more of the mentioned classes
expertise	project expertise	expertise may be a number or a keyword 0 – basic 1 – medium 2 - expert By specifying an expertise, you determine de depth of the tree presented.
phase	all	list of project phases codes
safeguard	all	list of safeguard codes
source	all	list of sources of information codes

2.30.2 policies.list

```
<policies.list [domain] [domain_family] [safeguard]
               [expertise] [source] />
```

Prints the safeguards related to policy in one or more security domains.

See "*Policies / attributes*".

2.30.3 policies.required

```
<policies.required [domain] [domain_family] [safeguard]
                   [expertise] [source] />
```

Prints a table for the policy safeguards stating whether it applies o not in the domain.

See "*Policies / attributes*".

2.30.4 policies.valuation

```
<policies.valuation [domain] [domain_family] [phase] [safeguard]
                    [expertise] [source]/>
```

Prints a table for the policy safeguards stating the maturity of the implementation for the domain and phase.

See "*Policies / attributes*".

2.31 Procedures, POS, security procedures

2.31.1 Procedures attributes

attr	default	meaning
domain	all	list of security domain codes
domain_family	none	selects only the domains that are qualified with one or more of the mentioned classes
expertise	project expertise	expertise may be a number or a keyword 0 – basic 1 – medium 2 - expert By specifying an expertise, you determine de depth of the tree presented.
phase	all	list of project phases codes
safeguard	all	list of safeguard codes
source	all	list of sources of information codes

2.31.2 procedures.list

```
<procedures.list [domain] [domain_family] [safeguard]
                 [expertise] [source] />
```

Prints the safeguards related to procedures in one or more security domains.

See "*Procedures / attributes*".

2.31.3 procedures.required

```
<procedures.required [domain] [domain_family] [safeguard]
                     [expertise] [source] />
```

Prints a table for the procedure safeguards stating whether it applies o not in the domain.

See "*Procedures / attributes*".

2.31.4 procedures.valuation

```
<procedures.valuation [domain] [domain_family] [phase] [safeguard]
                      [expertise] [source]/>
```

Prints a table for the procedure safeguards stating the maturity of the implementation for the domain and phase.

See "*Procedures / attributes*".

2.32 replace

```
<replace kw />
```

Prints the value associated to the "kw".

attr	default	meaning
kw	mandatory	term defined

See "*define*.

Example

<pilar><replace kw="WS" /></pilar>

2.32.1 Predefined names

kw	value
`library`	[code] name
`library.code`	code
`library.name`	name
`marking`	classification mark
`model`	[code] name
`model.code`	code
`model.name`	name
`model.description`	description
`model.kw.KEY`	value assigned to KEY
`operation_mode`	operation mode

2.33 Risks

2.33.1 risk

`<risk id />`

Prints the information related to a single risk scenario.

2.33.2 risk.accumulated

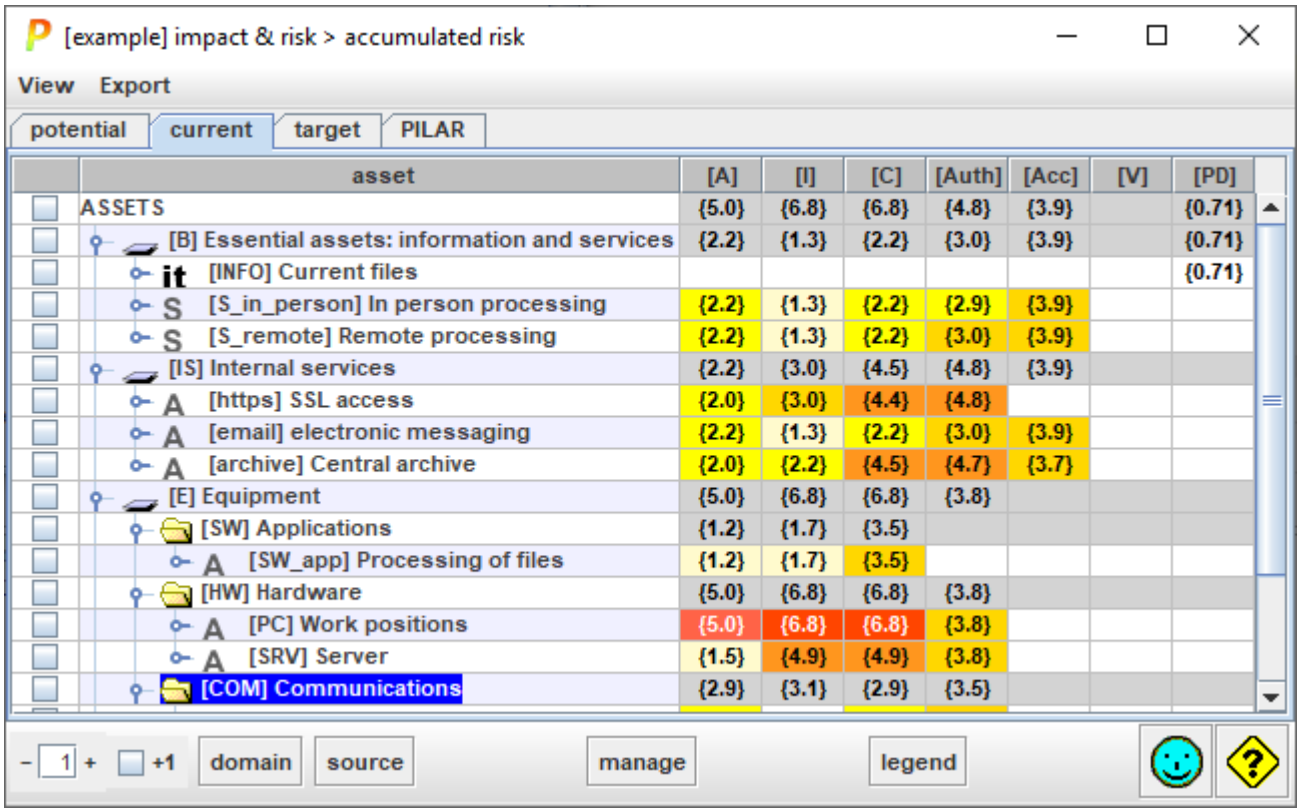
`<risk.accumulated [asset] phase1 />`

Inserts a table showing the accumulated risk in the selected phase.

attr	default	meaning
phase1	mandatory	the code of one project phase. Use phase "null" for potential risk.
asset	all	list of asset codes (see *Assets / attributes*

dimension	all	list of dimensions (not for bcm)
phases	all	list of phases (only for bcm)

The output is like this screen



2.33.3 risk.aggregated

Version 7.

`<risk.aggregated [layer] [asset] [dimension] [source] [threat] [phase] />`

Generates a table with as many columns as phases.

- For each layer and each asset, presents the aggregated risk,
  - aggregating all the assets in the layer or the asset group, if any
  - A layer must be explicitly mentioned to be presented. Use * to refer to all layers.
  - An asset may be directly mentioned, or via an asset group. Use * to refer to all assets.
- For each source, aggregates all risks affecting assets labelled for that source.
- for each threat, aggregates all risks from that threat

If no dimension is mentioned, all the dimensions are summed up. Otherwise, aggregates the mentioned dimensions.

If no phase is mentioned, all the phases are listed.

2.33.4 risk.aggregated.1phase

Version 7.

```
<risk.aggregated.1phase [layer] [asset] [source] [threat]
                        [dimension] [phase] />
```

Generates a table with as many columns as dimensions.

- For each layer and each asset, presents the aggregated risk, aggregating all the assets in the layer or the asset group.
    - A layer must be explicitly mentioned to be presented. Use * to refer to all layers.
    - An asset may be directly mentioned, or via an asset group. Use * to refer to all assets.
- For each source, aggregates all risks on the assets labelled for that source
- For each threat, aggregates all risks from that threat.

If no dimension is mentioned, all the dimensions are listed.

2.33.5 risk.deflected

```
<risk.deflected [asset] phase1 />
```

Inserts a table showing the deflected risk in the selected phase.

attr	default	meaning
phase1	mandatory	the code of one project phase. Use phase "null" for potential risk.
asset	all	list of asset codes
dimension	all	list of dimensions (not for bcm)
phases	all	list of phases (only for bcm)

2.33.6 risk.domain

```
<risk.domain [domain] [domain_family] [dimension] [phase] />
```

Inserts a table showing the risk in the requested domains, in the requested dimensions, in the requested phases.

attr	default	meaning
domain	all	list of security domain codes
domain_family	none	selects only the domains that are qualified with one or more of the mentioned classes
dimension	all	list of dimension codes See "*Dimension / code*
phase	current	the code of one project phase. Use phase "null" for potential risk.

2.33.7 risk.scenario

<risk.scenario [id] [elements] [phase] />

<risk [id] [elements] [phase] />

Id selects a scenario by identifier. It can be ignored within a for each frame.

Elements is a list of pieces to be listed. Default is

"id, asset, dimension, description, threat, measures, risk"

2.33.8 risk.table.down

<risk.table.down [assets] [threats] [dimensions] />

Presents a table like in GUI. Attribute 'assets' may be present to filter assets. Attribute 'threats' may be present to filter threats. Attribute 'dimensions' may be present to filter threats.

The following columns are presented

asset, threat, dimension, impact, likelihood, risk

2.33.9 risk.table.up

<risk.table.up [assets] [threats] [dimensions] />

Presents a table like in GUI. Attribute 'assets' may be present to filter assets. Attribute 'threats' may be present to filter threats. Attribute 'dimensions' may be present to filter threats.

The following columns are presented

asset (above), asset (below), threat (below), dimension (above), impact, likelihood, risk

2.34 Safeguard, countermeasure

2.34.1 safeguard

<safeguard safeguard [text] />

Prints

[code] name

attr	default	meaning
safeguard	mandatory	the code of one safeguard
text	optional	See "*text formatting*

2.34.2 safeguard.code

<safeguard.code safeguard />

Prints the code of the safeguard.

See "*safeguard*".

2.34.3 safeguard.name

<safeguard.name safeguard />

Prints the name of the safeguard.

See "*safeguard*".

2.34.4 safeguard.applies

< safeguard.applies safeguard [domain] [stage] />

Prints whether the **safeguard** applies or not in the given security domain.

See "*safeguard*".

2.34.5 safeguard.comment

< safeguard.comment safeguard [domain] [phase]/>

Prints the comment associated to the specified **safeguard** in the given security domain end project phase.

See "*safeguard*".

2.35 Safeguards, countermeasures

2.35.1 Safeguards attributes

attr	default	meaning
applies	all	only prints the safeguards that apply, or not: yes – only those that apply no – only those that do not apply all – all
comments	2	to print comments along the safeguards 0 – comments are not printed 1 – comments are printed in-line, close to the safeguard 2 – comments are printed afterwards as a hierarchy 3 – comments are printed using calls
depth	full	the depth of the tree presented
dimension	none	if you specify a list of dimensions, PILAR discards those safeguards that do not protect any of those dimensions See "*Dimension / code*".
domain	all	list of security domains
domain_family	none	selects only the domains that are qualified with one or more of the mentioned classes
exclude	none	if you specify a list of excluded safeguards, PILAR discards them

expertise	project expertise	expertise may be a number or a keyword 0 – basic 1 – medium 2 - expert By specifying an expertise, you determine de depth of the tree presented.
light	all	you may specify two phases; e.g. light=(phase1, phase2) or you may specify none, and PILAR will choose first and last phases light=() PILAR presents a coloured square with the difference between phase1 and phase2: • RED: far below • YELLOW: below • GREEN: equals • BLUE: above
name	n.a.	the name being defined
pattern	n.a.	See "*Patterns*".
phase	all	list of phase codes
safeguard	all	list of safeguard codes to be included
source	all	list of information sources codes to filter
threat	none	if you specify a list of threats, PILAR discards those safeguards that do not protect against this threat

2.35.2 safeguards.group

```
<safeguards.group name [pattern] [up_to] [phase] [plane] />

<safeguards.group name [safeguard] [up_to] [phase] [plane] />

<safeguards.group name [up_to] [phase] [plane] > PATH </>
```

Defines a group of safeguards, and it may be later identified by the given name.

attr	default	meaning
name	mandatory	the name being defined
pattern		See "*Patterns*
safeguard	all	list of safeguard codes to be included

up_to		selects only safeguards which valuation is under a given value, for the specified phase
phase		a reference phase for checking up to valuation
plane	base	a reference plane for checking up to valuation

2.35.3 safeguards.list

```
<safeguards.list pattern [domain] [domain_family] [comments] />


<safeguards.list [safeguard] [depth] [expertise] [comments]
                 [domain] [domain_family] [source] [dimension] [threat] />


<safeguards.list [comments]
                 [domain] [domain_family] [source] [dimension] [threat] >
   path_expression
</safeguards.list>
```

Lists safeguards that apply to one or more domains, down to a certain depth in the tree, or to a given expertise expansion.

See "*Safeguards / attributes*".

See *Patterns*.

See *Paths*

2.35.4 safeguards.required

```
<safeguards.required pattern [domain] [domain_family] [comments]
                     [applies] />


<safeguards.required [safeguard] [depth] [expertise] [comments]
                     [domain] [domain_family] [source]
                     [dimension] [threat] [applies] />


<safeguards.list [comments]
                 [domain] [domain_family] [source]
                 [dimension] [threat] >
   path_expression
</safeguards.list>
```

Generates a table showing whether the safeguard applies or not in the domain.

See "*Safeguards / attributes*".

See *Patterns*.

See *Paths*

2.35.5 safeguards.valuation

```
<safeguards.valuation pattern [slice] [comments]
                      [domain] [domain_family] [phase] />


<safeguards.valuation [slice] [comments]
                      [domain] [domain_family] [phase]
                      [safeguard] [depth] [expertise]
                      [source] [dimension] [threat] />


<safeguards.valuation [slice] [comments]
                      [domain] [domain_family] [phase] >
   path_expression
</safeguards.valuation>
```

Generates a table with the valuation of the safeguards in the specified phases.

"slice" is used to select one table by security domain or one table by project phase. See *evl.valuation*

See "*Safeguards / attributes*".

See *Patterns*.

See *Paths*

Attribute "mode" is used to decide the format of the value. The following options are available:

mode value	prints
0 | M | maturity	maturity [range] (default)
3 | MA	approximate maturity
6 | P | percent	percent

2.36 Security actions

See *action*

2.37 Specifications, attribute groups

You may give a name to a group of attributes:

```
<spec sname=s1 n1=v11 n2=v21 />
<spec sname=s2 n2=v22 n3=v32 />
<spec sname=s3 n2=v23 n3=v33 />
```

and refer to them later. PILAR will collect attributes in order:

```
spec=(s1) → n1=v11 n2=v21
spec=(s1, s2) → n1=v11 n2= v22 n3=v32
```

```
spec=(s1, s2, s3) → n1=v11 n2= v23 n3=v33
spec=(s2, s3) → n2= v23 n3=v33
```

You may use spec=(…) as an attribute anywhere, and PILAR will replace it for its expansion.

2.38 Stages, applicability stages

2.38.1 stage

<stage stage [text] />

Prints

[code] domain

attr	default	meaning
stage	base	the code of one applicability stage
text	optional	See "*text formatting*

2.38.2 stage.code

<stage.code stage />

Prints the code of the stage.

See "*stage*".

2.38.3 stage.description

< stage.description stage />

Prints the description of the stage.

See "*stage*".

2.38.4 stage.name

< stage.name stage />

Prints the name of the stage.

See "*stage*".

2.38.5 stages.group

<stages.group [name] [stage] />

Defines a group of applicability stages, and it may be later identified by the given name.

attr	default	meaning
name	""	the defined group
stage	all	list of stage codes

There is an empty name, that is used by default by every tag that requires one stage.

2.38.6 stages.list

< stages.list [domain] />

Lists the applicability stages in the model.

attr	default	meaning
stage	all	list of applicability stage codes

2.39 Steps, interruption steps

2.39.1 step

<step step />

Prints the interruption interval: [time]

attr	default	meaning
step	mandatory	time

2.39.2 steps.list

<steps.list />

List of interruption steps defined in the project.

2.40 table

A table is a collection of rows. Each row provides a mapping of an X value to a Y value.

For instance, if we have the valuation of the assets in a project:

asset	[D]	[I]	[C]	[A]	[T]
[EI_info]		[5]	[6]	[5]	[5]
[ES_local]	[5]			[7]	[6]
[ES_remote]	[3]			[7]	[6]

There we have 3 series, one per asset. The rows show assets, while columns show qualitative valuation.

In PILAR we write

```
<table what >
  rows
  columns
</table>
```

where rows and columns are described as foreach iterations.

Attribute “what” specifies the contents; that is the value for cells

what	meaning
value own	the explicit value for an asset
valua.accumulated	the accumulated value for an asset
safeguard.valuation	the maturity value for the safeguard; you need to determine domain – one security domain phase – one phase safeguard – one safeguard
evl.valuation	the valuation of the controls in a security profile you need to determine you need to determine evl – one security profile domain – one security domain phase – one phase safeguard – one safeguard
impact.accumulated	the accumulated impact
impact.deflected	the deflected impact
risk.accumulated	the accumulated risk
risk.deflected	the deflected risk

For rows and columns, you may provide a format for printing the text.

E.g.

```
<foreach_dimension text=(code – name)/>
<foreach_asset text=(code: name)/>
```

See "*text formatting*.

Example

```
<table what=(value.own) >
  <foreach_asset text=([code])/>
  <foreach_dimension text=([code])/>
</table>
```

2.41 Text

<text> xxx </text>

Prints

xxx

2.41.1 Paragraph

<paragraph> xxx </paragraph>

As <text>, adding paragraph marks before and after.

2.42 Text formatting

For many elements, you may specify the format to present.

Default is

…text=([code] name)

where “code” is replaced by the code, and “name” by the name of each element.

The following ones are defined

- source-text
- class-text
- domain-text
- layer-text
- asset-text
- threat-text
- phase-text
- countermeasure-text
- control-text
- stage-text

2.43 Tool

2.43.1 tool

`<tool />`

Prints

name (version)

2.43.2 tool.name

`<tool.name />`

Prints

name

2.43.3 tool.version

`<tool.version />`

Prints

version

2.44 Threats, security threats

2.44.1 threat

`<threat threat [text] />`

Prints

[code] name

attr	default	meaning
threat	mandatory	the code of one threat
text	optional	See "*text formatting*

2.44.2 threat.code

`<threat.code threat />`

Prints the code of the named threat.

See "*threat*".

2.44.3 threat.name

`<threat threat />`

Prints the code of the named threat.

See "*threat*".

2.45 Threats

2.45.1 threats.domain

`<threats.domain [domain] [domain_family] [dimension] [threat] [freq] [deg] />`

Dumps a compact table of threats per domain.

attributes	default	meaning
deg	true	true to list degradation
dimension	all	list of security dimension codes See "*Dimension / code*
domain	all	list of security domain codes
domain_family	none	selects only the domains that are qualified with one or more of the mentioned classes
freq	true	true to list likelihood

threat	all	list of threat codes

There is a row per threat.

If "freq" is true, there is one column showing the likelihood of the threat.

If "deg" is true, there is one column per dimension showing the degradation caused by the threat.

There is one column per dimension showing the risk on the domain caused by this threat.

2.45.2 threats.group

`<threats.group name [threat] />`

Defines a group of threats, and it may be later identified by the given name.

attr	default	meaning
name	mandatory	the name being defined
threat	all	list of threat codes

3 Patterns

Patterns may be used to select a few safeguards or controls.

The user provides an attribute that lists one or more patterns, such as

pattern = (pattern1, pattern2, …)

and PILAR will select the elements which code matches one of the patterns.

All patterns refer to the complete code of the element. That is, PILAR does not look for substrings that match the pattern.

Within a code, it is important to note that the code may be split into pieces separated by dots. For instance

10.6.a

it is formed by three sub patterns: “10”, “6”, and “a”.

PILAR uses two special characters to match

? matches any single character

* matches any sub pattern (between dots)

And there is a special pattern to match anything (version 7.3)

+ matches any string

Let us show some examples, taken from ISO 27002

pattern	selected codes
*	5 6 7 8 9 10 11 12 13 14 15
6.*	6.1 6.2

*.6	10.6 11.6 12.6
?5	15
*5	5 15

Let us show some examples, taken from NIST SP 800-53

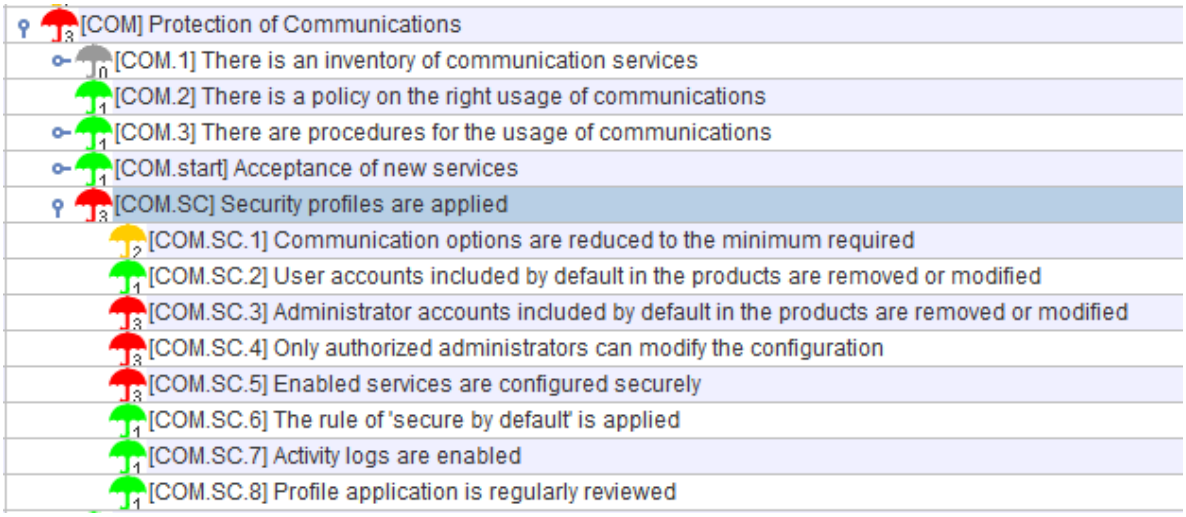
pattern	**selected codes**
SA-?	SA-1 SA-2 SA-3 SA-4 SA-5 SA-6 SA-7 SA-8 SA-9
SA-??	SA-10 SA-11 SA-12 SA-13 SA-14
SA-+	SA-1 SA-2 SA-3 SA-4 SA-5 SA-6 SA-7 SA-8 SA-9 SA-10 SA-11 SA-12 SA-13 SA-14

4 Paths

Paths are useful to select subtrees both on safeguards and on security profiles.

A path is a sequence of filters, starting from the root, and pruning the tree step by step.

Example, on the safeguards tree:




```
<path>
  <name>COM</name>
  <name>COM.SC</name>
  <name></name>
</path>
```

reduces the tree to the nodes expanded in the previous picture.

These are the options in each step:

<name>	Selects the node which name matches the name. There may be several names separated by commas; all that match are selected. If no name is provided, any node matches (that is, selects all the nodes at the level).
<pattern>	Selects the node which name matches the pattern.

	There may be several patterns separated by commas; all that match are selected. If no pattern is provided, any node matches (that is, selects all the nodes at the level). See *Patterns*
<perimeter>	Children are recursively selected until the border of the perimeter is met.

5 Filters for administrative data

Version: 7.

When determining assets, you may apply conditions on administrative data.

admdata = (value)

The value is a simple language:

expression := term

expression := term expression // AND

expression := term | expression // OR

expression := ‘(‘ expression ‘)’

term := name // the key exists and is not empty

term := name ‘=’ text // the key equals the text, case independent

term := name ‘~’ text // the key contains the text, case independent

In order to select attributes, you may specify the visible name, or the key using the notation “k:key”. Keys are visible in INFO files for personalization.

The “text” may be either a simple string, or ‘{‘ a set of strings ‘}’. If the text contains spaces or conflicting characters, it may be surrounded by 'single quotes' or "double quotes" where the terminating character cannot be in the enclosed string.

Some examples:

k:owner	is there is an ‘owner’ key
k:owner = ciso	if ‘ciso’ is the value of key ‘owner’
k:owner ~ smith	if the ‘owner’ value includes ‘smith’
k:owner = [smith, pearson]	if the value is in the set
k:owner = smith location = madrid	both conditions hold
k:owner = smith | location = madrid	one condition holds