

---

# Application Packaging Standard - Parallels Virtual Containers Aspect

1.2

Copyright © 1999, 2008 Parallels, Inc

All rights reserved.

## Table of Contents

1. Introduction .....	1
2. Example .....	2
3. Content Processing Method .....	2
3.1. Versioning .....	2
4. Archive Root Export .....	3

## 1. Introduction

RELAX NG schema of PVC aspect:

```
namespace pvc = "http://apstandard.com/ns/1/pvc"

## Content delivery method
div {
  ContentDeliveryMethod |= element pvc:templates {
    attribute class { "std-win" | "std-lin" | "ez-lin" | text }?,

    ## templates may export archive root to filesystem location
    element pvc:archive-root { attribute path { xsd:string { minLength = "1" } }
    }?,

    ## template is either embedded or available as external URL or must
    ## be searched through available repositories
    element pvc:template {
      ( attribute path { xsd:string { minLength = "1" } } |
        attribute url { xsd:anyURI } |
        attribute filename { xsd:string { minLength = "1" } } ),

    ## Redundancy check
    element pvc:checksum { attribute algorithm { "MD5" |
                                                "SHA1" |
                                                "SHA256" |
                                                text },
                          text }?

    }+
  }
}
```

This aspect allows declaring content processing method for applications utilizing Parallels Virtual Containers technology. This aspect uses the `http://apstandard.com/ns/1/pvc` XML namespace.

## 2. Example

### Example 1. PVC Templates Used for Application Content Delivery

```
<content xmlns:pvc="http://apstandard.com/ns/1/pvc">
  <pvc:templates class="std-win">
    <pvc:template path="pvc/application-template-0.1.12.efd">
      <pvc:checksum algorithm="MD5">
        8f199aebac0036c0c1fa2304eccc3d54
      </pvc:checksum>
    </pvc:template>
    <pvc:template path="pvc/webserver-template-2.3.efd">
      <pvc:checksum algorithm="MD5">
        c82ba28cb60b69c6a9c12334fd147c55
      </pvc:checksum>
    </pvc:template>
  </pvc:templates>
</content>
```

The example above describes that the application files are resided in PVC templates `application-template-0.1.12.efd` and `webserver-template-2.3.efd` placed in the `pvc/` directory within package archive.

## 3. Content Processing Method

This processing method instructs Controller to install the specified templates into available environment, following the usual templates installation procedure. Upgrade and removal of template also conform to usual PVC template update procedure.

If the targeted environment already contains specified templates, installation **MUST** be aborted. Upon removal of application, all the application templates **MUST** be uninstalled.

PVC templates may be embedded directly into APS package, or referenced as external source. When embedded into APS package, they must be specified with the `path` attribute pointing to the exact location of template file in application archive. It is possible to reference template available externally. For this, either an exact downloadable URL of the template must be specified with the `url` attribute, or a full filename with extension of the template file must be stated by the `filename` attribute. In the latter case, Controller should search available template repositories in order to find template with the given name. If such template cannot be found, installation must be aborted.

In order to help Controller to recognize exact format of PVC template type, it may be specified with the `class` attribute. The following types of PVC templates are predefined:

- `std-win` - templates format of Virtuozzo for Windows 3.x, Parallels Virtual Containers 4.0 for Windows
- `std-lin` - standard VZ Template for Virtuozzo v 3.x for Linux, Parallels Virtual Containers 4.0 for Linux
- `ez-lin` - EZ Template for Virtuozzo v 3.x for Linux, Parallels Virtual Containers 4.0 for Linux

Controller must treat unknown template type as unspecified.

### 3.1. Versioning

Referenced template name **SHOULD** contain template version in format described in the `Package Version` section. Vendor **SHOULD** update name in accordance with template version changes. Referenced template information **MAY** contain redundancy check in a form of `checksum`. Controller **SHOULD** perform template integrity check using the specified `checksum algorithm`. If Controller encounters unknown checksum algorithm, it must ignore it and treat the checksum as unspecified. Controller **MAY** use a cached version of template with matched version or checksum.

## 4. Archive Root Export

A PVC template may contain application files and directories, as well as APS metadata elements. In this case it is allowed to declare location of application files on filesystem after template installation with help of `archive-root` element. When specified, this element defines path to the directory which contains the application files after templates installation. This directory **MUST** be used by Controller if the specified files are not found in application archive. A Controller **MUST** prefer files existing in application archive to files from exported root. The value of the `path` attribute of this element **MUST** be an absolute path which is owned by one of the specified PVC templates.

A Package **SHOULD** have metadata elements necessary for displaying information about the Package in archive. Therefore a Package **MUST** have `APP-META.xml` in the root of archive.

### Example 2. PVC Template and URL Mapping

```
<content xmlns:pvc="http://apstandard.com/ns/1/pvc">
  <pvc:templates class="std-win">
    <pvc:archive-root path="C:\Program Files\Broobmla\Broobmla-13"/>
    <pvc:template path="broobmla-template-13.0.efd">
      <pvc:checksum algorithm="MD5">
        7bbf7634cea0cafa80c9255ab87d9ee6
      </pvc:checksum>
    </pvc:template>
  </pvc:templates>
</content>
...
<service id="vhost">
  <provision>
    <url-mapping>
      <mapping url="/" path="htdocs"/>
    </url-mapping>
  </provision>
</service>
```

When processing URL Mapping of the example above, Controller **MUST** firstly lookup mapping/@path content in the `htdocs` directory of application archive, and if it is not presented there - read it from `C:\Program Files\Broobmla\Broobmla-13\htdocs\` directory of the filesystem.