

# The project-v1.0 DTD

## Table of contents

1 project.....	2
2 node.....	2
3 setting.....	3
4 package.....	4
5 deployment.....	6
6 resource.....	7
7 resources.....	8
8 referrers.....	8

The 'project' XML document declares object data suitable for Workbench.

**Note:**

This is a demonstration document using all possible elements in the current ControlTier project-v10.dtd.

## 1. project

The root (aka "top-level") element of the project.xml file. Contains a set of resource elements. There is an explicit ordering of subsequent resource definitions.

element	description
<a href="#">node</a>	node resource
<a href="#">setting</a>	setting resource
<a href="#">package</a>	package resource
<a href="#">deployment</a>	deployment resource

**Table 1: elements**

### Example

```
<!DOCTYPE project PUBLIC
    "-//ControlTier Software Inc.//DTD Project Document 1.0//EN"
    "project.dtd">
<project>
  <node .../>
  <setting ... />
  <package ... />
  <deployment ... />
</project>
```

## 2. node

The node element defines a Node resource object.

attribute	description	values
type	node type	Node
name	node name	a name
description	node description	a character string
hostname	administrative hostname	a valid hostname

osArch	operating system architecture	(eg uname -p)
osFamily	operating system family	windows or unix
osName	operating system name	(eg uname -s)
osVersion	operating system version	(eg uname -r)
ctlbase	CTL framework instance directory	CT_BASE
ctlhome	CTL software installation root directory	CTL_HOME
ctlusername	CTL remote connection username	(eg \${framework.ssh.user})
ctlpassword	CTL remote connection password	Unceccessary when using key-based authentication

**Table 1: attributes**

element	description
<a href="#">resources</a>	Child dependencies
<a href="#">referrers</a>	Parent dependencies

**Table 2: elements**

### Example

```
<node name="strongbad" type="Node"
  description="a development host"
  hostname="strongbad.local"
  osArch="i386" osFamily="unix" osName="Darwin" osVersion="9.2.2"
  ctlbase="/ctier/ctl" ctlhome="/ctier/pkgs/ctl-1.1"
  ctlusername="${framework.ssh.user}"/>
```

## 3. setting

The setting element defines a Setting resource object.

attribute	description	values
type	setting type	Setting type
name	setting name	a name
description	setting description	a character string

settingType	the setting value	a string
settingValue	the value type	a string

**Table 1: attributes**

element	description
<a href="#">referrers</a>	Parent dependencies

**Table 2: elements****Example**

```
<setting type="PackageFinishScript" name="apache-tomcat-5.5.26.zip"
  description="The script used during prepare"
  settingValue="$HOME/bin/tomcat-finish.sh" settingType="script" />
```

**4. package**

The package element defines a Package resource object.

attribute	description	values
type	package type	Package type
name	package name	a name
description	package description	a character string
arch	platform architecture	a string
base	the package base name	a string
buildtime	the package buildtime	a string
filename	the package filename name	a string
filetype	the package archive type	(eg, zip,tgz,jar,rpm)
installroot	the package installation root directory	a path
installrank	the package relative install rank	(eg, an integer)
release	the package release	a string
releasetag	the package release tag	a string
repoUrl	the URL to the package	a URL

restart	Boolean specifying if a restart is required	true/false
vendor	the package vendor	a string
version	the package version	a string

**Table 1: attributes**

element	description
<a href="#">resources</a>	Child dependencies
<a href="#">referrers</a>	Parent dependencies

**Table 2: elements**

### Example

```
<package
  arch="noarch"
  base="apache-tomcat-5.5.26"
  buildtime="2008061570109"
  description="The Tomcat application server."
  filename="apache-tomcat-5.5.26.zip"
  filetype="zip"
  installroot="/demo/apache-tomcat-5.5.26"
  installrank=""
  name="apache-tomcat-5.5.26.zip"
  release=""
  releasetag=""
  repoUrl="{framework.pkgRepo.upload-url}/default/zip/zips/apache-tomcat-5.5.26.zip"
  restart="false"
  type="zip"
  vendor=""
  version="5.5.26"
/>
```

Another example showing resources:

```
<!--
  **
  ** Defines path to the prepare and finish scripts
  **
-->
<setting type="PackagePrepareScript" name="apache-tomcat-5.5.26.zip"
  description="The script used during finish"
  settingValue="$HOME/bin/tomcat-prepare.sh"
settingType="script"/>
<setting type="PackageFinishScript" name="apache-tomcat-5.5.26.zip"
  description="The script used during prepare"
  settingValue="$HOME/bin/tomcat-finish.sh" settingType="script"/>
```

```

<!--
**
** Defines package resource
**
-->
<package
  arch="noarch"
  base="apache-tomcat-5.5.26"
  buildtime="2008061570109"
  description="The Tomcat application server."
  filename="apache-tomcat-5.5.26.zip"
  filetype="zip"
  installroot="/demo/apache-tomcat-5.5.26"
  installrank=""
  name="apache-tomcat-5.5.26.zip"
  release=""
  releasetag=""
  repoUrl="{framework.pkgRepo.upload-url}/default/zip/zips/apache-tomcat-5.5.26.zip"
  restart="false"
  type="zip"
  vendor=""
  version="5.5.26"
>
  <!--
  **
  ** References the prepare and finish scripts as dependencies to the
  package
  **
  -->
  <resources>
    <resource type="PackagePrepareScript"
name="apache-tomcat-5.5.26.zip"/>
    <resource type="PackageFinishScript"
name="apache-tomcat-5.5.26.zip"/>
  </resources>
</package>

```

## 5. deployment

The deployment element defines a Deployment resource object.

attribute	description	values
type	deployment type	Deployment type
name	deployment name	a name
description	deployment description	a character string
basedir	the base directory	a path
installRoot	the install directory	a path

**Table 1: attributes**

element	description
<a href="#">resources</a>	Child dependencies
<a href="#">referrers</a>	Parent dependencies

**Table 2: elements**

**Example**

```
<deployment
  type="Deployment "
  name="tomcat1"
  description="The Tomcat deployment."
  installRoot="/demo/apache-tomcat-5.5.26 "
  basedir="/demo/apache-tomcat-5.5.26" />
```

Another example showing resources and referrers:

```
<deployment
  type="Deployment "
  name="tomcat1"
  description="The Tomcat deployment."
  installRoot="/demo/apache-tomcat-5.5.26 "
  basedir="/demo/apache-tomcat-5.5.26">

  <resources>
    <resource name="apache-tomcat-5.5.26.zip" type="zip" />
    <resource name="extras.jar" type="jar" />
  </resources>

  <referrers>
    <resource name="centos" type="Node" />
    <resource name="development" type="Node" />
  </referrers>
</deployment>
```

**6. resource**

A resource object. A resource element is used to reference another object in the project model by type and name.

attribute	description	values
type	resource type	a type
name	resource name	a name

**Table 1: attributes****Example**

Refrences an instance of Deployment named "tomcat1".

```
<resource type="Deployment" name="tomcat1" />
```

**7. resources**

The resources tag defines a set of child dependencies. Each dependency is referenced via a [resource](#) element.

attribute	description
<a href="#">resource</a>	A child dependency resource

**Table 1: elements****Example**

```
<resources>
  <resource name="apache-tomcat-5.5.26.zip" type="zip" />
  <resource name="extras.jar" type="jar" />
</resources>
```

**8. referrers**

The referrers tag defines a set of parent dependencies. Each dependency is referenced via a [resource](#) element.

attribute	description
<a href="#">resource</a>	A parent dependency resource

**Table 1: elements****Example**

```
<referrers>
  <resource name="centos" type="Node" />
  <resource name="development" type="Node" />
</referrers>
```