



# OSD APP TREE

version 2.6.0

document version 1.6.0

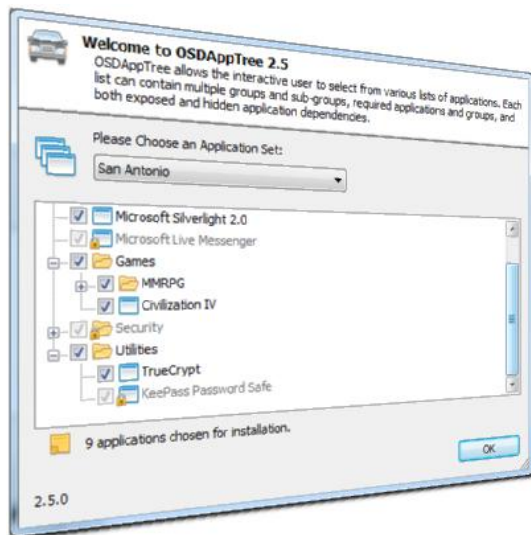
Jan 01, 2010

Jason Sandys

## WHAT IS IT

OSD App Tree is a configurable way to allow user interaction during an Configuration Manager (ConfigMgr) Operating System Deployment (OSD) Task Sequence (TS). It presents a tree of applications to the interactive user allowing them to choose which ones to install during the TS. Configuration is based on an XML file that contains all possible applications, their settings, and their organization in the tree presented to the end user.

OSD App Tree was previously called OSD App Chooser. I changed the name because it was shorter and sounded better (to me at least).



## WHERE CAN I GET IT

The latest version OSD App Tree is no longer available via Microsoft Codeplex. It has moved to <http://myitforum.com/cs2/blogs/jsandys/pages/OSDAppTree.aspx>.

Please send all bugs, features requests, comments, etc. using the contact form on my blog.

## CHANGE LOG

### 2.6

- Added conditions (page 14)
- Added save to and load from data file ability; aka persistence (page 6)
- Added creation of “is laptop”, “is desktop”, and “is server” TS variables (page 14)
- Added creation of ts variable for the chose AppSet (page 10)
- Created 64-bit version

### 2.5.2

- Fixed return code bug where pushing **Enter** to invoke the OK button resulted in a return code of 0xD or tabbing to the OK button and pressing space resulted in 0x20.

### 2.5.1

- Bug fix

### 2.5.0

- Added customizable header
- Added customizable help text
- Added application dependencies
- Added customizable application tooltips
- Changed window resizing code
- Removed title bar
- Updated tree and dialog icons

### 2.0.0.5 Release

- Fixed memory leak
- Added logging info

### 2.0.0.3b

- Added graceful handling of comments in the XML configuration file
- Removed dependence on oledlg.dll
- Made the window appear on top of all others including the TS progress bar

## USAGE

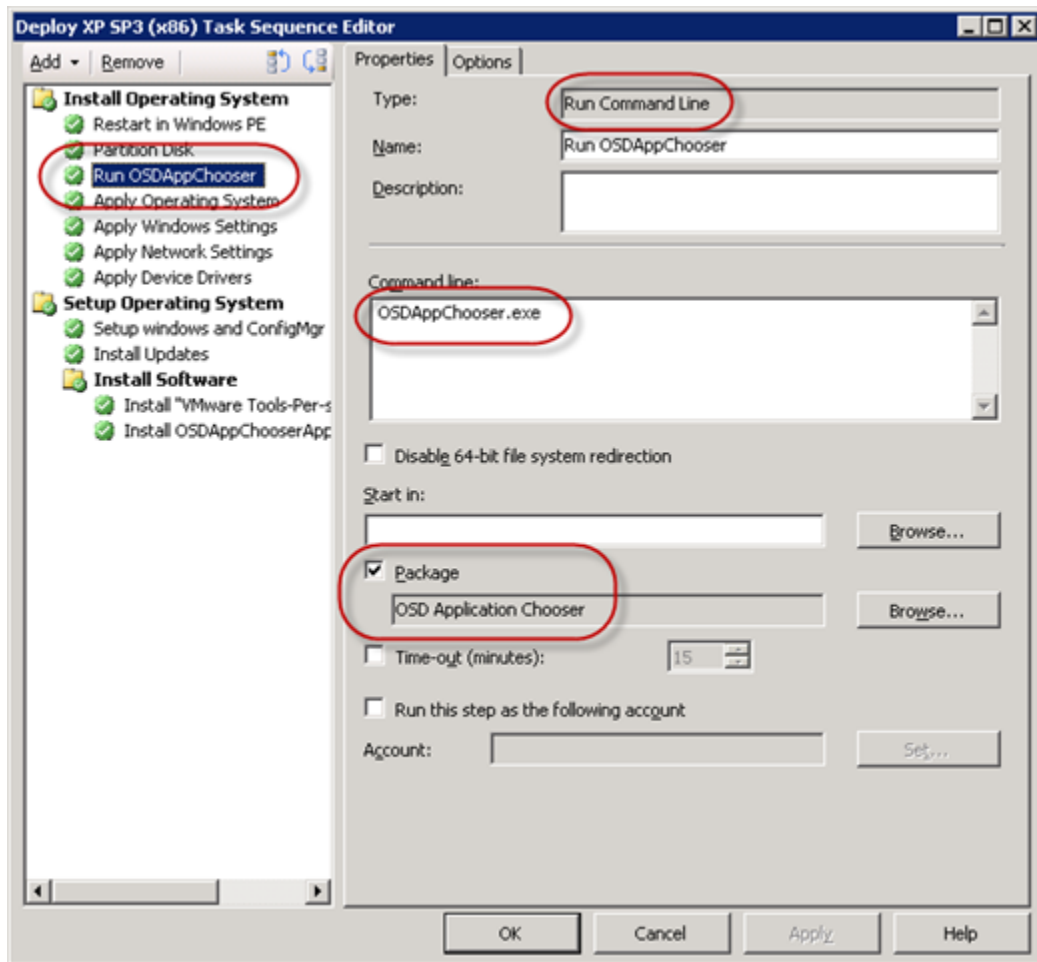
OSD App Tree must be run from a command-line task inside an OSD Task Sequence. Because of "by design" limitations, a Software Install task cannot show any UI or be interactive in any way. Additionally, command-line tasks must be run while in Windows PE to show any UI thus OSD App Tree cannot be used in a pure software distribution task sequence or any part of the task sequence not running in Windows PE.

OSD App Tree takes advantage of the install multiple applications feature of the software install task and pre-populates a series of TS variables based upon applications chosen from the tree.

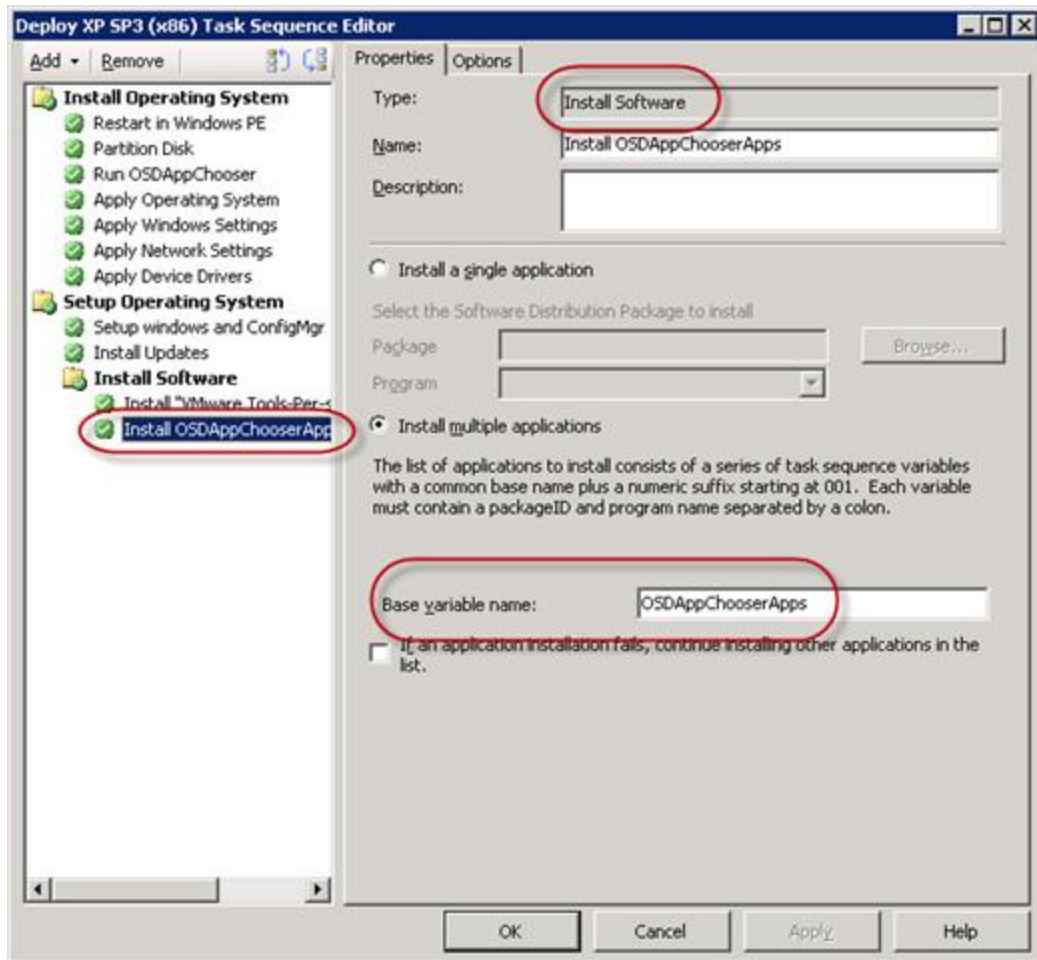
Create a configuration file named *OSDAppTreeConfig.xml*. To load an alternately named file, use the command-line parameter discussed below. See the "Configuration File" section for details.

- 1) Testing
  - a) Simply run OSD App Tree by double-clicking on the icon or from the command-line. The command-line parameters specified below are valid.
  - b) No visible output will be generated, check the log file for results. See the "Log File" section for details.
- 2) Production
  - a) Make the OSD App Tree.exe files available to the task sequence. This includes *OSD App Tree.exe* and your configuration file -- the default name for the configuration file is *OSDAppTreeConfig.xml* but this can be changed using a command-line parameter (see Optional command-line parameters below).


There are multiple ways to do this but I prefer to put them into a software distribution package -- no program is necessary. This has the advantage in larger distributed environments of ensuring that the client accesses the files from the closest DP. The disadvantage is that you have to remember to update the DP anytime you update the configuration file. An alternative is to place the files in a shared folder and use a map folder task to make them available. This has the advantage of not having to update the DP every time you make a change. It also has an advantage when your TS is set to download and execute because the command-line task to run OSD App Tree (see the next step) can be run before the partition and format task. A hybrid approach is also possible where you put *OSDAppTree.exe* in a package and make the configuration file available using a shared folder.



- b) Create a command-line task to run *OSD App Tree.exe*. If you used a package for step 1, reference the package you placed the files in this task. If you used a shared folder, make sure you used a map folder task before this one and prefix *OSDAppTree.exe* with the drive letter you mapped in that task.
- c) Create a Software Install task later in the task sequence; I usually put this at the very end. This task must run in the deployed OS and not Windows PE. Use the Install multiple applications option and set the base variable name to *OSDAppTreeApps* (see Optional command-line parameters next to change this).



### 3) Optional command-line parameters:

- **/appset:<default appset>** The default appset to appset to use. This appset will be chosen automatically and is the only appset available. If not specified, all appsets will be available but none will be chosen.
- **/var:<var name>** The prefix to use for the sequence of TS variables. If not specified, the prefix *OSDAppTreeApps* will be used.
- **/conf:<filename>** The filename of the configuration file to use. This can include a path if necessary. If not specified, *OSDAppTreeConfig.xml* will be used.
-  - **/persist:<filename>** The filename to persist the variables to so that can be loaded later during a task sequence using the LoadEm helper applet.

The advantage of using command-line arguments is that you can use other TS variables to set their values. For example, if you have multiple configuration files, you can populate a TS variable named MyConfig with the filename and then use the following command-line: `OSD App Tree /conf:%MyVar%`

Be sure to enable the following settings for any and all programs that will be installed using this method on the **Advanced** tab of the program's properties: **Allow this program to be installed from the Install Software task sequence without being advertised.**

## LOG FILE

OSD App Tree will generate or append to a log file name *OSDAppTree.log* every time it runs. This log records major events including error events. If you launch OSD App Tree and nothing happens, check this log file for details. To prevent any ugliness from being presented to an end user, everything is sent to this file without notifying the interactive user.

The log file is a standard ConfigMgr log file and is best viewed using Trace32 from the toolkit. If run inside a TS, the log file is located with the standard TS log file SMSTS.log: if you place the task at or near the beginning of the TS as described in the usage section, this will be *X:\Windows\Temp\SMSTS*. If you run OSD App Tree outside of a TS, the log file will be located in the current user's temp directory which can easily be located using the environment variable *%TEMP%*.

## PERSISTENCE

In refresh scenarios, it may be desirable to display OSD App Tree before the system is rebooted into WinPE. This poses a problem because task sequence actions cannot have any user interaction unless they are in WinPE. To solve this, OSD App Tree can be run from a normal ConfigMgr package/program that the task sequence depends on; OSD App Tree will then save all variables to a data file which can then be loaded by the helper applet LoadEm which is called during the task sequence. Use the */file* command-line option to LoadEm to specify the data file to load. A sample walkthrough is in order:

1. Create a package to contain the OSD App Tree source files including *OSDAppTree.exe*, the configuration file, and any supporting icon files.
2. Create a program in the package to run OSD App Tree. Use a command similar to *OSDAppTree.exe /config:OSDAppTreeConfig.xml /persist:c:\OSDAppTree.dat*. Ensure that the **Allow users to interact with this program** checkbox is selected and that Program can run **Only when a user is logged on** is also chosen on the program properties **Environment** tab.
3. Distribute the package to the applicable DPs.
4. On the **Advanced** tab of the desired task sequence's dialog box, check **Run another program first** and choose the package and program that you just created. Also choose the **Always run this program first** checkbox.
5. Create a package to contain the *LoadEm.exe* source file and distribute it to your applicable DPs. No program is necessary.
6. Create a command-line task at the very beginning of the task sequence with the following command line:  
*LoadEm /file:c:\OSDAppTree.dat*






That's it. OSD App Tree will run before the task sequence using a normal software distribution and all of the variables and their values created during the interactive application selection process of OSD App Tree will be saved to *C:\OSDAppTree.dat*. The variables and their values will then be loaded by **LoadEm** when it runs during the TS and **LoadEm** will create the task sequence variables. Be sure to also include a Software Install task as described above in the usage section as this task actually initiates the software installation.


## CONFIGURATION FILE


The configuration file has changed significantly since the version 1.1 Autolt based release. It is still XML based but has been reorganized and enhanced. The following table describes the valid elements. Note that OSD App Tree does not check the configuration file against an XSD schema so will most likely ignore additional or out of place elements. This is not guaranteed though, so definitely test your configuration file outside of OSD before using it.

There are a few additions since OSDAppChooser 2.0.0.5 (marked below) but these are not required; thus your OSDAppChooser 2.0.0.5 configuration files should still work fine.

Element Name	Valid Attributes	Valid Parents	Valid Children	Comments
<b>OSD App Tree</b>	<ul style="list-style-type: none"> <li>* <b>title</b> – The title of the dialog box, displayed in the header</li> <li>* <b>description</b> – A description to present to the interactive user, displayed in the header</li> <li>* <b>icon</b> – The icon to display in the header</li> <li>* <b>info</b> – Initial prompt text to display under the tree when the dialog is first started</li> </ul>	N/A	Apps AppSets	This is the root element of the file
<b>Apps</b>	N/A	OSD App Tree	App	Only the first occurrence will be used
<b>App</b>	<ul style="list-style-type: none"> <li>+ <b>id</b> - a unique id, typically a GUID</li> <li>Name - a display name to display to the user</li> <li>+ <b>pkgID</b> - the ConfigMgr package ID of the package</li> <li>+ <b>progID</b> - the ConfigMgr program ID</li> <li>* <b>icon</b> – The icon to display in the application’s tooltip</li> <li>* <b>info</b> – Detailed information about the application to display in the application’s tooltip</li> <li>* <b>requires</b> – The ids of any application that will also be selected if this application is selected. Multiple ids can be separated by a comma.</li> </ul>	Apps	N/A	The order that apps are listed in determines their installation order
<b>AppSets</b>	<ul style="list-style-type: none"> <li>* <b>prompt</b> - The prompt to display in the dialog box above the appset combo box</li> </ul>	OSD App Tree	AppSet	Only the first occurrence will be used
<b>AppSet</b>	<ul style="list-style-type: none"> <li>+ <b>Name</b> - the name of the AppSet</li> <li>* <b>defaultAll</b> - (0 or 1) All children (groups and apps) are selected by default</li> </ul>	AppSets	AppRef AppGroup	These will be presented to the user for selection in the combo box
<b>AppRef</b>	<ul style="list-style-type: none"> <li>+ <b>appid</b> - an valid id from an App element</li> <li>* <b>mandatory</b> - (0 or 1) sets an application to mandatory preventing a user from unselecting it</li> <li>* <b>default</b> - (0 or 1) selects an application by default</li> </ul>	AppSet AppGroup	N/A	These determine which AppSet an application will be available under in the tree.

<b>AppGroup</b>	<ul style="list-style-type: none"> <li> <b>groupName</b> - The name of the group</li> <li> <b>mandatory</b> - (0 or 1) sets a group and all of its children to mandatory preventing a user from unselecting them</li> <li> <b>default</b> - (0 or 1) selects a group and all of its children by default</li> </ul>	AppSet AppGroup	AppRef	AppGroups can be nested.
<b>Condition</b>	<ul style="list-style-type: none"> <li> <b>type</b> - The type of conditional statement to perform. There are currently two types: <b>wmi</b> and <b>tsvar</b>.</li> <li> <b>query</b> - The definition of the query to perform. For wmi conditions, this should be a proper WMI query against the root\cimv2 namespace and for tsvr conditions it should be in the form of <i>tsvariable</i> &lt;operator&gt; <i>value</i>. There are currently two operators available: = and !=.</li> </ul>	AppSet AppGroup Condition	AppGroup AppRef Condition	Conditions can be nested to create a logical "And".

 These attributes are optional. Lack of one of these will result in default functionality.

 These attributes are required. Lack of one of these will result in undefined behavior.

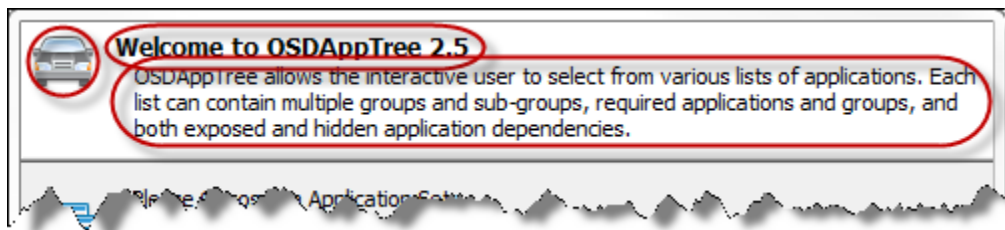
Note that all element and attribute names are case sensitive. Also note that you can omit attributes that take a zero or one value, they will be set to zero by default.

## FEATURE CONFIGURATION

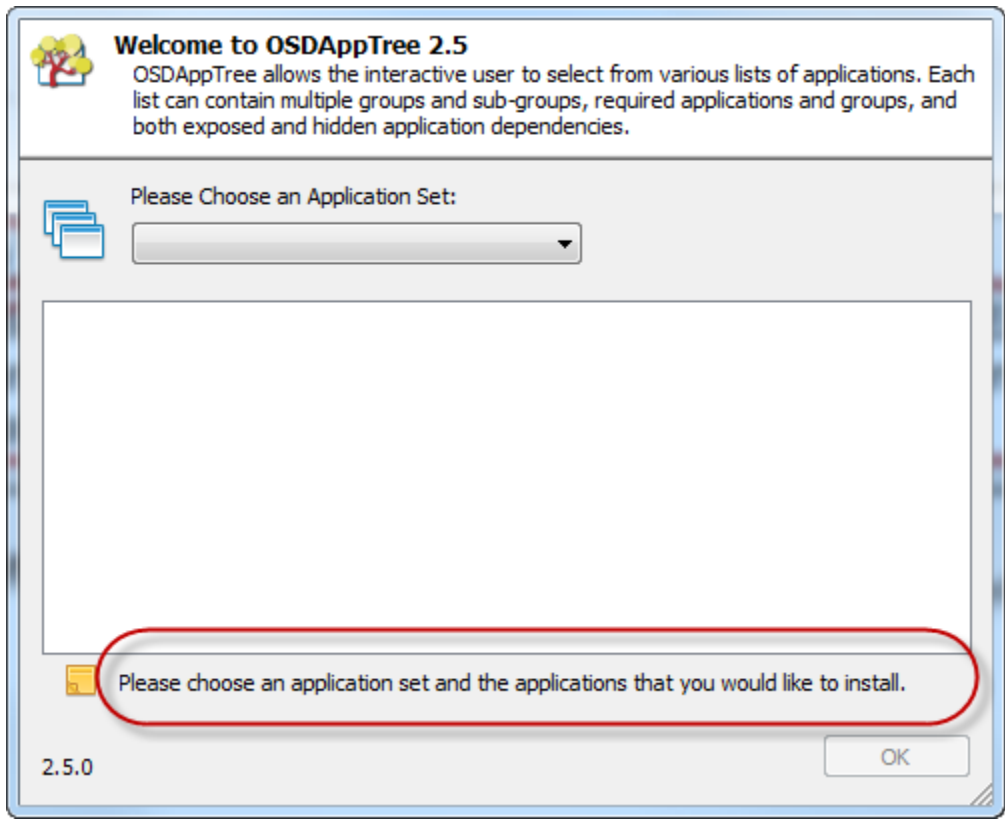
### HEADER

A major new element, the header is purely aesthetic but gives you the ability to provide a lot more information to the interactive user. There are three elements that can be customized: the title, the description, and the icon. Lack of any of these attributes results in default text of the default icon. Referenced icons should be placed in the package source files so that they are available.

```
<OSDAppTree title="Welcome to OSD App Tree 2.5" description=" OSD App Tree allows the interactive user to select from various lists of applications. Each list can contain multiple groups and sub-groups, required applications and groups, and both exposed and hidden application dependencies." info="Please choose an application set and the applications that you would like to install." icon="car.ico">
```



Additionally, the initial text displayed below the application tree can also be customized using the **info** attribute of the OSD App Tree element. Note that this text is not permanent and will be replaced by the count of applications currently chosen in the tree once an application set has been chosen.

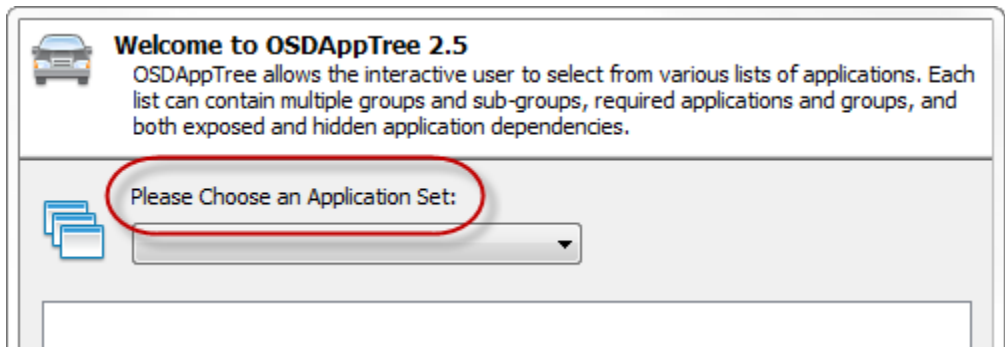


---

## APPSET PROMPT

Use the **prompt** attribute of the **AppSets** element to set the prompt displayed.

```
<AppSets prompt="Please Choose an Application Set:">
```

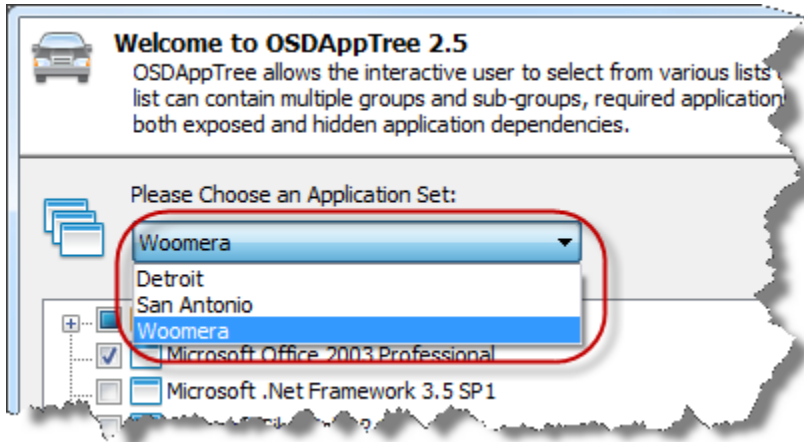


---

## APPSETS

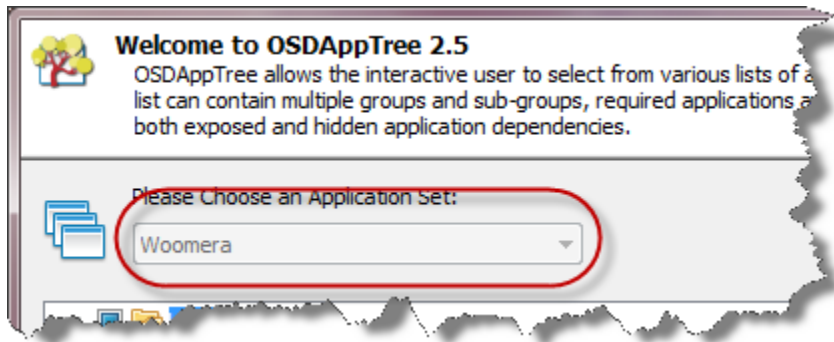
AppSets shown in this combo box are defined by adding **AppSet** elements to the **AppSet** element. No nesting is possible.

```
<AppSet name="San Antonio" defaultAll="1">
<AppSet name="Detroit" defaultAll="0">
<AppSet name="Woomera">
```



If you pass a default AppSet on the command-line using the **/appset** parameter, then the specified AppSet will be chosen and no change will be possible; e.g., `OSDAppTree /appset:Woomera`.

☀️ The chosen AppSet will also be saved to the task sequence variable **varChoice** where **var** is the variable name specified using the **/var** command-line option or the default **OSDAppTreeApps**; thus if **/var** is not specified on the command-line, **OSDAppTreeAppsChoice** will be the variable name set with the chosen AppSet name.



## GROUPS

Use the **appGroup** element to create groupings of applications inside of an **AppSet**. These can be nested, disabled (using the **mandatory** attribute), and selected by default (using the **default** attribute). The mandatory and selected by default state of groups are inherited by all of a groups children. Mandatory groups are shown with grey text and checkbox and with a lock on their tree icon.

```
<AppGroup groupName="Anti-Virus" mandatory="1">
<AppGroup groupName="Security" mandatory="0">
<AppGroup groupName="Utilities" default="1">
```



## APPLICATIONS

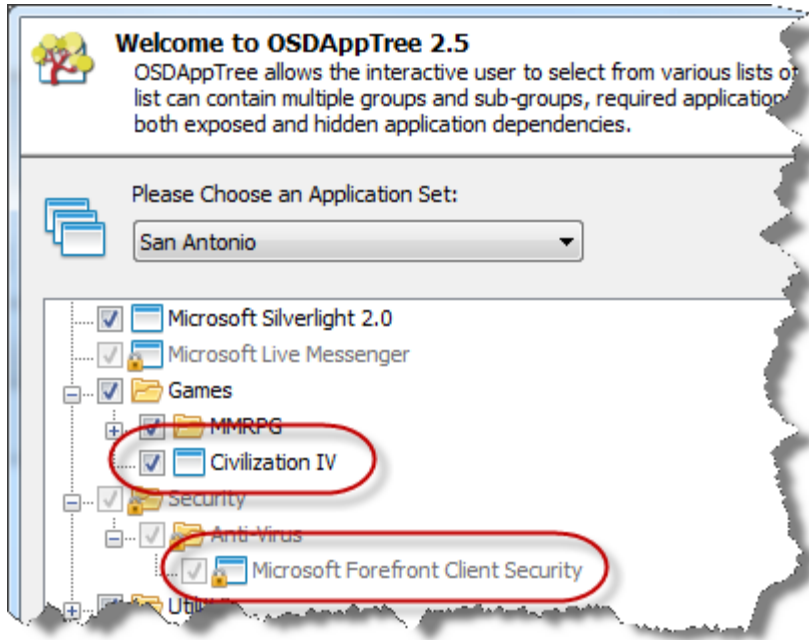
Applications are defined in two phases. The first is using an **App** element in the **Apps** section. These define the absolute properties for a specific application including the display name and the ConfigMgr package and program IDs. These IDs must match exactly what is defined in ConfigMgr. The order of the **App** elements also defines the order in which the TS variables are created and thus their installation order. The id must be unique for each **App** element and GUIDs are recommended. Visual Studio includes a GUID generator and there are also multiple available on the web.

```
<App id="A95EBE8A-F2E8-45ce-8E96-DA90DC7FC5A4" name="Microsoft Office 2007
Enterprise" icon="office.ico" pkgID="XYZ00009" progID="Per-system unattended"
info="Microsoft Office 2007 (officially called 2007 Microsoft Office System) is
the most recent Windows version of the Microsoft Office System, Microsoft's
productivity suite."/>
```

The second part is to define **AppRef** elements inside either **AppSet** or **AppGroup** elements. These are references to **App** elements and matched using the **appId** attributes. The order of **App** elements determines display order only and not application installation order.

Applications can be disabled (using the **mandatory** attribute), and selected by default (using the **default** attribute). The mandatory and selected by default state of applications is inherited from any parent groups. Mandatory applications are shown with grey text and checkbox and with a lock on their tree icon.

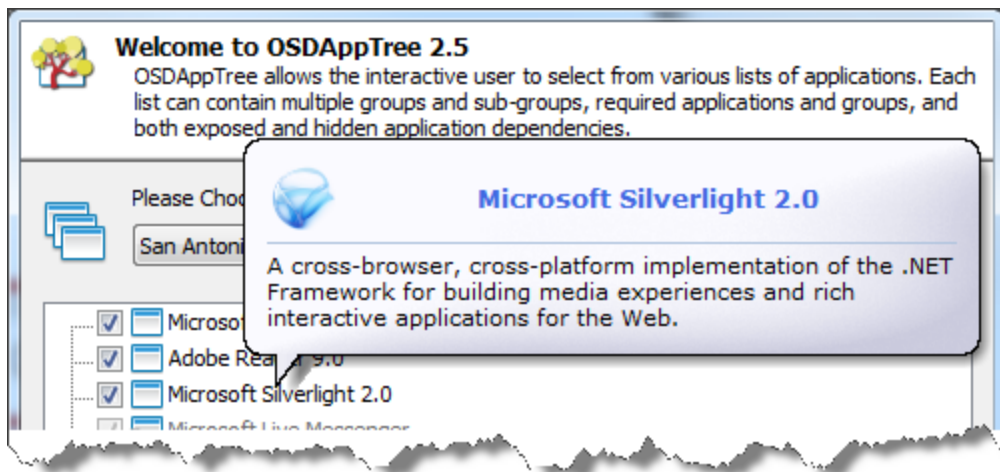
```
<AppRef appId="1B3FFF8B-C044-4340-9898-D25091D61FE2" mandatory="0"
default="1"/>
<AppRef appId="58E6BFAF-6D15-42fc-B37D-964264B8FF91" mandatory="1"
default="0"/>
<AppRef appId="8236C6F6-75D8-43d4-88F0-1A7685668471" mandatory="0"/>
```



## TOOLTIPS

Hovering over any application will pop up a bubble tooltip showing the title of the application, additional specified information about the application, an icon, and any dependent applications (discussed next).

```
<App id="8236C6F6-75D8-43d4-88F0-1A7685668471" name="Microsoft Silverlight 2.0" icon="silverlight.ico" pkgID="XYZ00017" progID="Per-system unattended" info="A cross-browser, cross-platform implementation of the .NET Framework for building media experiences and rich interactive applications for the Web."/>
```

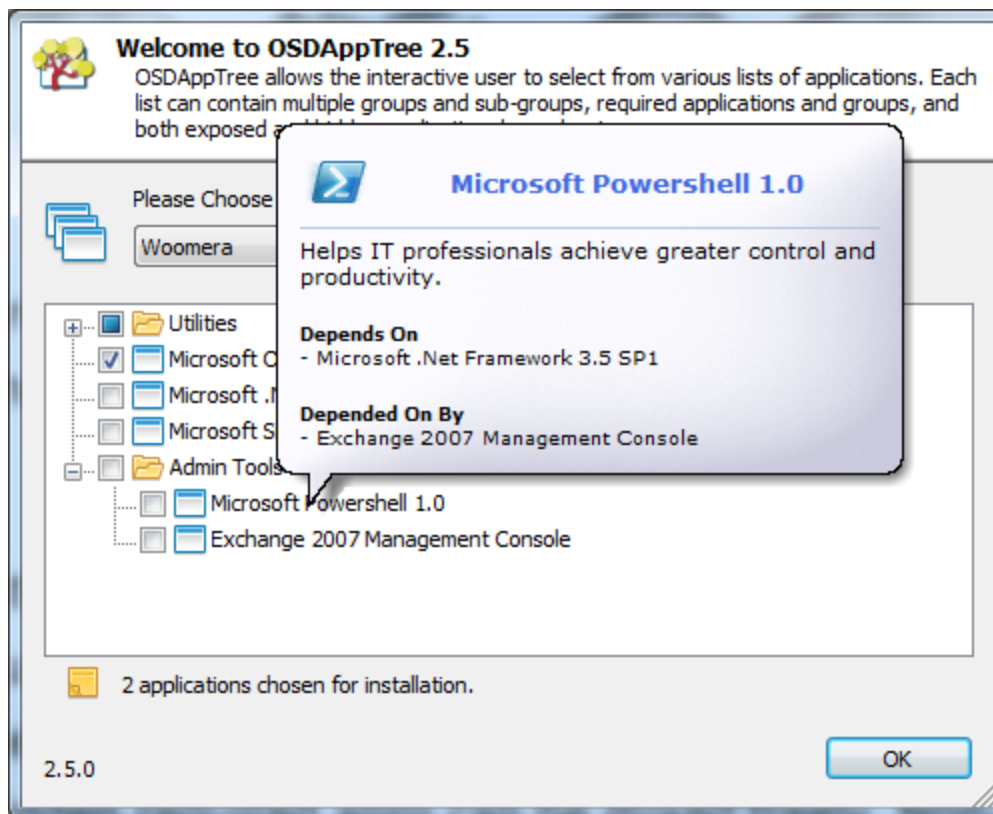


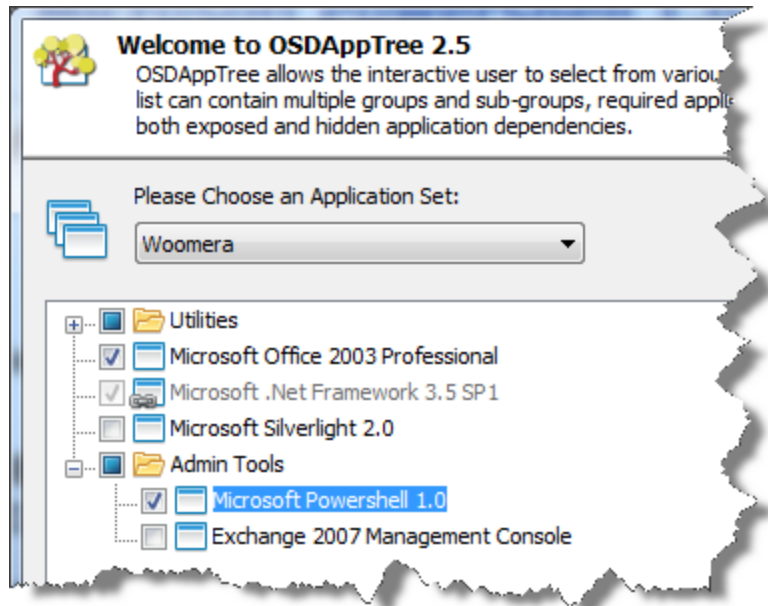
## DEPENDENCIES

Dependencies can be used to model pre-requisites or other required applications based on a selection. Multiple dependencies can be specified and dependencies are transitive; i.e., if app A requires app B and app B requires app C then app A also requires app C.

If a required app is not part of the currently selected appset, then the requirement is a hidden requirement; no feedback will be given to the interactive user that the additional required app(s) will also be selected for installation. If a required app is part of the currently select appset, it will be selected and disabled (the icon will also change to one with a link) at the same time the app requiring it is selected. Hidden and non-hidden required apps can be mixed freely. Non-hidden required apps are listed in an app's tooltip as well as non-hidden apps that require the current app.

```
<App id="58E6BFAF-6D15-42fc-B37D-964264B8FF91" name="Microsoft .Net Framework 3.5 SP1" pkgID="XYZ00044" progID="Per-system unattended" info="A full cumulative update that contains many new features building incrementally upon .NET Framework 2.0, 3.0, 3.5, and includes cumulative servicing updates to the .NET Framework 2.0 and .NET Framework 3.0 subcomponents."/>
<App id="053F88E6-C9E6-47ad-BF3F-54C42295C301" name="Microsoft Powershell 1.0" icon="ps2.ico" pkgID="XYZ000E7" progID="Per-system unattended" requires="58E6BFAF-6D15-42fc-B37D-964264B8FF91" info="Helps IT professionals achieve greater control and productivity."/>
<App id="4F03AAC9-D750-4619-A99F-54B85067FB44" name="Exchange 2007 Management Console" icon="Exchange.ico" pkgID="XYZ000E8" progID="Per-system unattended" requires="053F88E6-C9E6-47ad-BF3F-54C42295C301,FAE1C3A4-C7E7-4268-95AD-31B8B230F016"/>
```





## CONDITIONS

Conditions allow the conditional display of applications (referenced by **AppRef** tags) or groups of applications. To add a condition to an application or group, add an application element as its parent. All child elements of a condition element will only be displayed if the condition evaluates to true or in the case of a WMI query returns at least one result. Conditions can be nested.

**WMI** conditions execute a given WMI query against the root\cimv2 namespace and **tsvar** conditions compare the value of a task sequence variable against a specified value using one of two operators: **equals** and **not equals**.

```
<Condition type="wmi" query="SELECT * FROM Win32_ComputerSystem WHERE
caption='wallace'">
  <AppRef appId="A95EBE8A-F2E8-45ce-8E96-DA90DC7FC5A4" mandatory="0"/>
</Condition>

<Condition type="tsvar" query="myVar != 12">
  <AppGroup groupName="Utilities">
    <AppRef appId="586BE03F-2491-45a6-8A67-88E0F1909336"/>
    <AppRef appId="9838B557-2F2B-4868-8DF0-ACA2BC2AE1C2"
      mandatory="1"/>
  </AppGroup>
</Condition>
```

## TASK SEQUENCE VARIABLES

OSD App Tree also sets three additional (at this time) task sequence variables:

- **LaptopThisIs, DesktopThisIs, ServerThisIs** (think Yoda) – These variables are set to 1 if the current system is a laptop, desktop, or server respectively and 0 if they are not. These are a full-fledged task sequence variables that can be used after OSD App Tree in the task sequence or in an OSD App Tree condition. If OSD App Tree is executed using a normal ConfigMgr package/program (and not in a task sequence) as

described in the **Persistence** section above, they can still be used in a **tsvar** condition even though other task sequence variables cannot.

## CREDIT AND ACKNOWLEDGEMENTS:

- David Sackstein for his great set of articles which I used to read the configuration file: <http://blogs.microsoft.co.il/blogs/davids/archive/2008/12/20/msxml-in-c-but-as-elegant-as-c.aspx>
- Hans Dietrich for his awesome custom tree control: <http://www.codeproject.com/KB/tree/XHtmlTree.aspx>
- Pavel Antonov: <http://www.codeproject.com/KB/cpp/cmdlineparser.aspx>
- Marc Richarme: <http://www.codeproject.com/KB/dialog/easysize.aspx>
- Robert Python: <http://www.codeproject.com/KB/dialog/RPResizeDlg.aspx>

All icons derived from FatCow hosting icons: <http://www.fatcow.com/free-icons/>.

## LICENSE

This application is released subject to the following license (note that the source code is no longer freely available):

Microsoft Public License (Ms-PL)

This license governs use of the accompanying software. If you use the software, you accept this license. If you do not accept the license, do not use the software.

### 1. Definitions

The terms "reproduce," "reproduction," "derivative works," and "distribution" have the same meaning here as under U.S. copyright law.

A "contribution" is the original software, or any additions or changes to the software.

A "contributor" is any person that distributes its contribution under this license.

"Licensed patents" are a contributor's patent claims that read directly on its contribution.

### 2. Grant of Rights

(A) Copyright Grant- Subject to the terms of this license, including the license conditions and limitations in section 3, each contributor grants you a non-exclusive, worldwide, royalty-free copyright license to reproduce its contribution, prepare derivative works of its contribution, and distribute its contribution or any derivative works that you create.

(B) Patent Grant- Subject to the terms of this license, including the license conditions and limitations in section 3, each contributor grants you a non-exclusive, worldwide, royalty-free license under its licensed patents to make, have made, use, sell, offer for sale, import, and/or otherwise dispose of its contribution in the software or derivative works of the contribution in the software.

### 3. Conditions and Limitations

(A) No Trademark License- This license does not grant you rights to use any contributors' name, logo, or trademarks.

(B) If you bring a patent claim against any contributor over patents that you claim are infringed by the software, your patent license from such contributor to the software ends automatically.

(C) If you distribute any portion of the software, you must retain all copyright, patent, trademark, and attribution notices that are present in the software.

(D) If you distribute any portion of the software in source code form, you may do so only under this license by including a complete copy of this license with your distribution. If you distribute any portion of the software in compiled or object code form, you may only do so under a license that complies with this license.

(E) The software is licensed "as-is." You bear the risk of using it. The contributors give no express warranties, guarantees or conditions. You may have additional consumer rights under your local laws which this license cannot change. To the extent permitted under your local laws, the contributors exclude the implied warranties of merchantability, fitness for a particular purpose and non-infringement.