

# Creating a vSphere Update Manager Depot

ESXi 5.0

---

This technical note describes how to create a depot for vSphere Installation Bundles (VIBs) that VMware vSphere Update Manager 5.0 or ESXCLI can access for ESXi 5.0 host updating.

The following topics are included in this document:

- “Related Documentation” on page 1
- “Setting Up the Publishing Web Site” on page 2
- “Grouping VIBs into metadata.zip Files” on page 6
- “Naming metadata.zip Files” on page 6
- “Understanding How Update Manager Constructs URLs” on page 6
- “Hosting Requirements” on page 7
- “Merge vSphere 5.x and vSphere 4.x Depots” on page 7
- “Creating Depots from Offline Bundles” on page 8

## Related Documentation

This document describes the structure of an online depot. Related documentation provides complete information about the the creation, signing, publishing, and using of VIBs.

- *vSphere VIB Customization Overview for VMware Partners* includes an overview of the VIB creation process for VMware partners.
- *vSphere Image and VIB Customization for VMware OEMs* includes an overview of the VIB creation process for VMware OEM partners.
- VMware Workbench online help explains how to create, sign, and package VIBs from within Workbench.
- The vSphere Documentation Center includes the following documentation:
  - *vSphere Installation and Setup* includes information about Image Builder cmdlets and ESXCLI
  - *Image Builder PowerCLI Reference*
  - *vSphere Command-Line Interface Reference* (includes ESXCLI reference)
  - *Installing and Administering VMware vSphere Update Manager*

## About Vendor Depots

A vSphere Installation Bundle (VIB) is a single software package that may contain drivers, CIM providers, VMkernel modules, or other software. A depot is a repository of VIBs available online to Update Manager users. You create and maintain a depot to control the VIB files that Update Manager users can access.

---

**NOTE** Compared to vSphere 4.x depots, vSphere 5.x depots have a new directory structure and a new schema for top-level index files and vendor-level index files. Update Manager 5.0 supports vSphere 4.x and 5.x depots, including depots that use both formats simultaneously.

---

The depot and Web service portal use a static Web site to deliver a set of files intended for consumption by Update Manager. Before you provide updates to users through a depot, use the Workbench-based developer kit or the VIB Suite to generate the following files:

- A `vendor-index.xml` file. The index lists the product ID and version number, and points to `metadata.zip` files.
- One or more `metadata.zip` files. The `metadata.zip` files point to the location of the VIB files.  
Names of `metadata.zip` files cannot conflict with other filenames within your set of files or with the files of other vendors.
- One or more bulletins describing the VIB file. A bulletin is an XML file that describes the VIB included in a consumable package, `metadata.zip`. At least one bulletin must be included in every `metadata.zip` file.
- One or more VIB files.

From the vendor depot, Update Manager downloads a list of supported vendor Web site URLs. Based on the list of URLs, Update Manager obtains a list of metadata URLs from the publishing Web site, where it downloads the metadata. The metadata, in turn, contains the URLs for the VIB files.

You can create the files required for an update depot by using VIB tools that are part of VMware Workbench. To create an online depot, unzip the contents of an offline bundle created by Workbench to the Web server root directory.

## Setting Up the Publishing Web Site

The publishing Web site provides a list of supported vendor URLs for Update Manager to download. The publishing Web site has the following requirements:

- Supports HTTPS for downloads.
- Provides a top-level URL where Update Manager connects to get list of supported vendors and their URLs (for example, `https://www.company.com/updatemanagement/`). The URL is registered to Update Manager either through the `index.xml` file on the VMware update publishing Web site or through the Update Manager user interface.
- Metadata files are grouped according to platform (product, version, and locale).

A single `metadata.zip` file can contain updates for more than one platform.

## Naming Convention

The following notation is used in the examples in this document.

**Table 1.** Naming Notation

Notation	Definition and Example
[PRODUCT]	esx
[PRODUCTCODE]	NNN (for example, VMC). The code assigned to the vendor's product (not ESX). The product code is determined by VMware in conjunction with the vendor.
[RELTYPE]	Use <code>patch</code> or <code>update</code> . Applicable to vSphere 4.x only.

**Table 1.** Naming Notation

Notation	Definition and Example
[TOPLEVELURI]	The Web URI for the patch depot. When appended by /index.xml or /depot-index.xml this forms the patch depot URL (for example, https://<sample>.vmware.com/software/VUM/PRODUCTION/).
[VENDOR]	Use the vendor ticker symbol in lowercase. For example, the ticker symbol for VMware is vmw.
[VERSION]	n.n.n (for example, 4.0.0 or 5.0.0). The version number for the product platform (for example, ESX). The version string <b>cannot</b> change unless the platform version changes. For example, if the version string for patch01 is 4.0.0, the version string for patch02 or update01 must also be 4.0.0.
[VERSION#]	nnn (for example, 400 or 500)
[RELEASEID]	vSphere 4.x only. [PRODUCTCODE]-[VERSION]-[RELTYPE]nn (for example, VMC-4.0.0-patch01). The release ID must be unique across all patch releases. If you use VMC-4.0.0-patch01 once, it cannot be used for future patch releases.
[INFRASTRUCTURE]	The infrastructure version of the depot. In vSphere 5.0, this is vib20.

## Structure of the Depot

The structure of vSphere 4.x depots is different from the structure of vSphere 5.x depots.

### Structure of an Update Depot in vSphere 4.x

A typical 4.x vendor update Web site is structured as shown:

```
[TOP-LEVEL-URI]/
  index.xml
  [VENDOR]/
    [VENDOR]-index.xml
    [VENDOR]-[PRODUCTCODE]-[VERSION]-metadata.zip
  [PRODUCT]/
    [VERSION]/
      [RELEASEID]/
        *.vib                                # Use a relative path
                                           # VIB files
```

### Structure of an Update Depot in vSphere 5.x

A typical 5.x vendor update Web site is structured as shown:

```
[TOP-LEVEL-URI]/
  depot-index.xml
  [VENDOR]/
    [VENDOR]-index.xml
    [VENDOR]-[PRODUCTCODE]-[VERSION]-metadata.zip
  [INFRASTRUCTURE]/
    <directory_name>/
      *.vib                                # VIB files
```

In the following example Web site structure, VMware is the vendor and ESXi 5.0 is the product.

```
https://<sample>.vmware.com/software/VUM/PRODUCTION/
  depot-index.xml
  vmw/
    vmw-index.xml      ---- lists product ID, version, and metadata URLs
    vmw-5.0.0-metadata.zip
  vib20/
    esx-base/
      *.vib
```

## Example Top-Level Index File in vSphere 4.x

At the top level of the update Web site is an index.xml file with the following contents.

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

```

<ns:vendorList xmlns:ns="urn:patchmdsvc">
  <vendor>
    <name>vmware, inc.</name>
    <code>vmw</code>
    <indexfile>vmw-index.xml</indexfile>
    <patchUrl>vmw</patchUrl>
  </vendor>
</ns:vendorList>

```

## Example Top-Level Index File in vSphere 5.x

At the top level of the update Web site is a depot-index.xml file with the following contents.

```

<?xml version="1.0" encoding="UTF-8"?>
<vmwdpm:vendorList xmlns:vmwdpm="http://www.vmware.com/depotmanagement"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.vmware.com/depotmanagement/depot-index-xml.xsd">
  <!-- A depot contains a list of vendors that provide the software-->
  <vendor>
    <name>VMware Inc.</name>
    <!-- code is the unique identifier for the vendor, this is used as -->
    <!-- folder name for all software from this vendor -->
    <code>vmw</code>
    <indexfile>vmware-index.xml</indexfile>
    <relativePath>vmw</relativePath>
    <content>
      <!-- readable name of the depot, can be used for UI -->
      <name>VMware ESX</name>
      <!-- Depot content type definition. -->
      <!-- For ESX/ESXi patches uses http://www.vmware.com/depotmanagement/esx -->
      <!-- For virtual appliance updates, use http://www.vmware.com/depotmanagement/vaIm -->
      <type>http://www.vmware.com/depotmanagement/esx</type>
    </content>
  </vendor>
</vmwdpm:vendorList>

```

## Example Vendor-Index.xml File in vSphere 4.x

The vendor-index.xml file lists product patches for the vendor and points to metadata.zip files.

```

<?xml version="1.0" encoding="UTF-8"?>
<ns:metaList xmlns:ns="urn:patchmdsvc">
  <metadata>
    <productId>esx</productId>
    <version>4.0.0</version>
    <locale>INTL</locale>
    <url>vmw-VMC-4.0.0-metadata.zip</url>
  </metadata>
</ns:metaList>

```

## Example Vendor-Index.xml File in vSphere 5.x

The vendor-index.xml file lists product patches for the vendor and points to metadata.zip files.

```

<?xml version="1.0" encoding="UTF-8"?>
<vmwdpm:metaList xmlns:vmwdpm="http://www.vmware.com/depotmanagement"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.vmware.com/depotmanagement/vendor-product-index-xml.xsd">
  <metadata>
    <productId>embeddedEsx</productId>
    <version>5.*</version>
    <locale></locale>
    <url>vmw-VMC-5.0.0-metadata.zip</url>
    <channelName>default</channelName>
  </metadata>
</vmwdpm:metaList>

```

## Example Index File for a vSphere 4.x Depot Hosting Two Vendors

This index.xml file is an example of one for a depot hosting two vendors' patches.

```
<?xml version="1.0" encoding="UTF-8"?>
<ns:vendorList xmlns:ns="urn:patchmtdsvc">
  <vendor>
    <name>sample company, inc.</name>
    <code>smpl</code>
    <indexfile>smpl-index.xml</indexfile>
    <patchUrl>smpl</patchUrl>
  </vendor>
  <vendor>
    <name>vmware, inc.</name>
    <code>vmw</code>
    <indexfile>vmw-index.xml</indexfile>
    <patchUrl>vmw</patchUrl>
  </vendor>
</ns:vendorList>
```

## Example Index File for a vSphere 5.x Depot Hosting Two Vendors

This depot-index.xml file is an example of one for a depot hosting two vendors' patches.

```
<?xml version="1.0" encoding="UTF-8"?>
<vmwdpm:vendorList xmlns:vmwdpm="http://www.vmware.com/depotmanagement"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.vmware.com/depotmanagement/depot-index-xml.xsd">
  <!-- A depot contains a list of vendors who provide the software-->
  <vendor>
    <name>VMware Inc.</name>
    <!-- code is the unique identifier for the vendor, this is used as -->
    <!-- a folder name for all software from this vendor. -->
    <code>vmw</code>
    <indexfile>vmware-index.xml</indexfile>
    <relativePath>vmw</relativePath>
    <content>
      <!-- readable name of the depot, can be used for UI -->
      <name>VMware ESX</name>
      <!-- This is the internal identifier of the depot schema/types/structure -->
      <!-- and can be used as key to determine whether tools like VUM can consume the
contents. -->
      <type>http://www.vmware.com/depotmanagement/esx</type>
    </content>
  </vendor>
  <vendor>
    <name>sample company, inc.</name>
    <code>smpl</code>
    <indexfile>smpl-update-index.xml</indexfile>
    <relativePath>smpl</relativePath>
    <content>
      <name>sample company module</name>
      <type>http://www.vmware.com/depotmanagement/esx</type>
    </content>
  </vendor>
</vmwdpm:vendorList>
```

## Grouping VIBs into metadata.zip Files

Update Manager downloads a set of `metadata.zip` files to find out if bulletins are installed on the host. The `metadata.zip` files contain all the information Update Manager needs to perform a host scan. Update Manager downloads the VIBs as needed during staging or installation of patches.

Group VIBs into `metadata.zip` files according to product and version. Because some updates belong to more than one `metadata.zip` file, using this organization helps Update Manager locate the file that contains the appropriate update. Grouping files also allows Update Manager to use only the relevant `metadata.zip` files while scanning the hosts.

When you group VIBs into `metadata.zip` files, choose one of these strategies:

- Create one `metadata.zip` file for all VIB files.

For a small number of updates, VMware recommends that you create only one `metadata.zip` file for all VIBs. While the metadata file might be larger when you use this strategy, this is the least complicated way to organize metadata.

- Create one metadata file for each product and version.

You might have to copy the same bulletin into more than one metadata file if there are cross-platform updates present. You cannot further subdivide files within each product/version pair.

## Naming metadata.zip Files

A vendor depot provides all the `metadata.zip` files referenced by the top-level `index.xml` or `depot-index.xml` file. Update Manager downloads these `metadata.zip` files and stores them locally. The relationship between the `metadata.zip` file and its corresponding product version is kept in the Update Manager database.

When you name `metadata.zip` files, follow these guidelines:

- Use the form `[VENDOR]-[PRODUCTCODE]-[VERSION]-metadata.zip`.
- Retain filenames from update to update, because Update Manager interprets changes to filenames to be new or different files. In addition, consistent naming allows Update Manager users to keep track of new bulletins and VIBs.

## Understanding How Update Manager Constructs URLs

Update Manager constructs URLs that it uses to locate metadata and VIB files and access the data. The URLs are listed in `depot-index.xml` and `vendor-index.xml`, and can be relative.



**CAUTION** URL redirection prevents Update Manager from building metadata and VIB URLs. Do not use redirected URLs.

Update Manager uses the following elements to construct URLs to access `metadata.zip` and VIB files:

- A top-level URL for `depot-index.xml`:  
`https://<sample>.vmware.com/software/VUM/PRODUCTION/depot-index.xml`
- The vendor update URL in `depot-index.xml`:  
`<patchUrl>vmw</patchUrl>`
- The URL of `vendor-index.xml`:  
`https://<sample>.vmware.com/software/VUM/PRODUCTION/vmw/vmw-index.xml`
- The metadata URL in `vendor-index.xml`:  
`<url>vmw-5.0.0-metadata.zip</url>`

- The resulting URL for `vmw-5.0.0-metadata.zip`, for example, would be similar to:  
`https://<sample>.vmware.com/software/VUM/PRODUCTION/vmw/vmw-5.0.0-metadata.zip`
- The relative path to VIB files in `metadata.zip`:  
`<relativePath>/vib20/esx-base/sample.vib</relativePath>`
- The URL for a VIB file named `sample.vib` would be similar to:  
`https://<sample>.vmware.com/software/VUM/PRODUCTION/vmw/vib20/esx-base/sample.vib`

## Hosting Requirements

Update Manager uses a HTTPS/GET request to download `index.xml`, `metadata.zip` files, and VIBs. The vendor Web site must be able to accept HTTPS/GET requests and subsequently return the list of metadata.

The following information is included in the HTTPS/GET parameter list that Update Manager sends:

- Client ID
- Product ID
- Product version

## Merge vSphere 5.x and vSphere 4.x Depots

If 4.x and 5.0 depots must share a Web server, you can merge the two depots. ESX/ESXi hosts of different versions can access the updates from the depot in the corresponding format.

- 1 Copy the 4.x and 5.0 VIBs and related metadata to the Web server root directory, while keeping the directory structure intact.
- 2 Verify that the Web server directory structure contains two sets of depot-level index files and vendor-level index files.
- 3 Edit the 5.0 vendor-level index file to list the 4.x `metadata.zip` files .
- 4 Point Update Manager 5.0 to the 5.0 top-level `depot-index.xml`.

## Creating Depots from Offline Bundles

You can create update depots from existing offline bundles. VMware and third-party vendors might distribute patches and other software as ZIP bundles.

You can export image profiles to an offline bundle ZIP by using ESXi Image Builder CLI. See the *Using vSphere ESXi Image Builder CLI* section in the *vSphere Installation and Setup* documentation.

### Create a Standalone Update Depot from an Offline Bundle

You can create a update depot from an offline bundle by modifying metadata and making the files available online.

- 1 Unzip the offline bundle ZIP.
- 2 Modify the contents of the offline bundle.
  - a In `index.xml`, change the values in the `<name>` and `<code>` tags to the partner company name and its VMware-assigned partner code.
  - b In `vendor-index.xml`, change the version in the `<version>` tag. You might need to edit the `<productId>` tag as well. For ESXi hosts, including ESXi 5.0, the `<productId>` value is `embeddedEsx`.
  - c In `metadata.zip`, modify the bulletin information that is contained in `vmware.xml`. You can modify the `<softwarePlatform>` tag to control on which version of ESXi to install the VIBs. The VIB list of the bulletin contains all VIBs in the image profile. You can modify the list to make only selected VIBs available in the depot.
- 3 Upload the files to a Web server to create a standalone depot.

### Include Files from an Offline Bundle in an Existing Update Depot

You can include files from an offline bundle in existing update depots.

- 1 Unzip the offline bundle ZIP.
- 2 Modify the contents of the offline bundle.
  - a In `index.xml`, change the values in the `<name>` and `<code>` tags to the partner company name and its VMware-assigned partner code.
  - b In `vendor-index.xml`, change the version in the `<version>` tag. You might need to edit the `<productId>` tag as well. For ESXi hosts, including ESXi 5.0, the `<productId>` value is `embeddedEsx`.
  - c In `metadata.zip`, modify the bulletin information that is contained in `vmware.xml`. You can modify the `<softwarePlatform>` tag to control on which version of ESXi to install the VIBs. The VIB list of the bulletin contains all VIBs in the image profile. You can modify the list to make only selected VIBs available in the depot.
- 3 Incorporate the files from the image profile in an existing depot.
  - a Copy the modified `metadata.zip` file in the directory on the Web server where the other `metadata.zip` files are located.
  - b To create `vendor-index.xml`, run the following command:
 

```
vibpublish -I <path_to_vendor-index.xml> -m <metadata_directory>
```
  - c To create the top-level index file, run the following command:
 

```
vibpublish -n <vendor> --vendor-code <code> -I <path_to_depot-index.xml>
          --metadata-baseURL=<URL>
```

---

If you have comments about this documentation, submit your feedback to: [docfeedback@vmware.com](mailto:docfeedback@vmware.com)

**VMware, Inc. 3401 Hillview Ave., Palo Alto, CA 94304 [www.vmware.com](http://www.vmware.com)**

Copyright © 2011 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at <http://www.vmware.com/go/patents>. VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

Item: EN-000631-00

---