

# Dynamic Environment Manager Application Migration

OmniSSA Dynamic Environment Manager



# Creating migration XML files

The Dynamic Environment Manager download package contains **Migrate10.xsd**, an XML schema definition for the migration XML vocabulary. Use this file to validate your migration XML file or to provide context-sensitive help.

The download package also contains a sample migration XML file (**Application Migration sample.xml**) containing examples for each migration action.

## 1 General structure

The general structure of a migration XML file is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<migrate xmlns="http://www.immidio.com/Schemas/Flex/Migrate10/">
  <registry>
    ... registry actions ...
  </registry>
  <filesystem>
    ... file system actions ...
  </filesystem>
</migrate>
```

**NOTE:** By default, all registry and file system information is left intact. You only need to specify actions for information that needs to change.

Section 2 describes the actions that can be used within the `<registry>` element, and section 3 describes the `<filesystem>` actions.

Registry actions are performed first, followed by file system actions. The registry and file system actions are performed in the order they are listed in in the XML file, which is something to keep in mind when performing related actions. For instance, to rename a folder and delete a file from that folder, you can either first rename the folder and then delete the file using the *new* folder name, or first delete the file using the *old* folder name and then rename the folder. Changing the order won't work, as the file will no longer exist with the old name or won't yet exist with the new name, respectively.

## 2 Registry actions

Registry actions can be performed on values and keys (note that most key-related actions also affect sub keys, so in effect are applied to a registry *tree*).

All registry key references must start with **HKEY\_CURRENT\_USER** (or the **HKCU** alias) or **HKEY\_LOCAL\_MACHINE** (or the **HKLM** alias), specified in all capitals.

To specify the default registry value, use the empty string (two quotes with no content in between) as the value name.

### 2.1 Registry value actions

Registry value actions are specified as

```
<value action="..." key="..." source="..." ... />
```

child elements of the `<registry>` element, with the `action` attribute indicating the action



to perform, and the **key** and **source** attributes specifying the registry key and value to operate on.

Depending on the action, other attributes are needed as well, as described below.

## 2.2 Set registry value

To create or update a registry value:

```
<value action="set"
      key="HKCU\Software\Vendorw"
      source=""
      type="string">Some text</value>
```

This sets the default registry value (as indicated by the empty **source** attribute) of key **HKCU\Software\Vendorw** to the string (as specified by the **type** attribute) *Some text* (i.e. the content of the **<value>** element).

The **type** attribute determines the type of the registry value, and also affects the format of the element content:

| Type              | Description                                     | Element content example  |
|-------------------|---|--|
| <b>integer</b>    | A 32-bit number                                 | <code>&lt;value ... type="integer"&gt;42&lt;/value&gt;</code>        |
| <b>string</b>     | A simple string                                 | <code>&lt;value ... type="string"&gt;Some text&lt;/value&gt;</code>  |
| <b>multi</b>      | Multiple strings, separated by <b>&amp;#10;</b> | <code>&lt;value ... type="multi"&gt;A&amp;#10;B&lt;/value&gt;</code> |
| <b>expandable</b> | A string that can contain environment variables | <code>&lt;value ... type="expandable"&gt;%PATH%&lt;/value&gt;</code> |
| <b>binary</b>     | Base64-encoded binary data                      | <code>&lt;value ... type="binary"&gt;Rmx1eCs=&lt;/value&gt;</code>   |

## 2.3 Rename registry value

To rename a registry value:

```
<value action="rename"
      key="HKCU\Software\VendorX"
      source="oldName"
      destination="newName" />
```

This renames the **oldName** registry value of key **HKCU\Software\VendorX** to **newName**. Note that this action can only rename values within the same key – to move a value to another key, use the **move\_value** action of the **<key>** element (cf. 2.10).

## 2.4 Delete registry value

To delete a registry value:

```
<value action="delete"
      key="HKCU\Software\VendorY"
      source="removeMe" />
```

This deletes the **removeMe** registry value of key **HKCU\Software\VendorY**.



## 2.5 Convert registry value to other type

To convert a registry value to another type:

```
<value action="convert"
       key="HKCU\Software\VendorZ"
       source="v"
       type="string" />
```

This changes the type of registry value `v` of key `HKCU\Software\VendorZ` to `string`. Note that the original data is not converted for this; the `convert` action just flips the type.

The `type` attribute accepts the same values as for the `set` action: `integer`, `string`, `multi`, `expandable`, and `binary`.

**NOTE:** You will hardly ever need to use this action. It is mainly provided for applications that store strings as binary data in one version and use actual strings for the next version.

## 2.6 Registry key actions

Registry key actions are specified as

```
<key action="..." source="..." ... />
```

child elements of the `<registry>` element, with the `action` attribute indicating the action to perform, and the `source` attribute specifying the registry key to operate on. Depending on the action, other attributes are needed as well, as described below.

## 2.7 Create registry key

To create a registry key:

```
<key action="create"
      source="HKCU\Software\VendorW" />
```

This creates registry key `HKCU\Software\VendorW`.

## 2.8 Rename registry key

To rename a registry key:

```
<key action="rename"
      source="HKCU\Software\VendorX\1.0"
      destination="HKCU\Software\VendorX\2.0" />
```

This renames registry key `HKCU\Software\VendorX\1.0` to `HKCU\Software\VendorX\2.0`.

## 2.9 Delete registry key

To delete a registry key:

```
<key action="delete"
      source="HKCU\Software\VendorY" />
```

This deletes registry key `HKCU\Software\VendorY`.



## 2.10 Move registry value to other key

To move a registry value to another key:

```
<key action="move_value"
      source="HKCU\Software\VendorZ"
      destination="HKCU\Software\VendorZ\AppData"
      value="Path" />
```

This moves registry value `Path` from key `HKCU\Software\VendorZ` to key `HKCU\Software\VendorZ\AppData`.

## 3 File system actions

All paths for file system actions must start with one of the following folder tokens:

- AppData
- Cookies
- Desktop
- Favorites
- LocalAppData
- NetHood
- Personal
- PrintHood
- ProgramsMenu
- RecentFiles
- SendTo
- StartMenu
- StartupMenu
- UserProfile

These folder tokens are case-sensitive.

### 3.1 File actions

File actions are specified as

```
<file action="..." source="..." ... />
```

child elements of the `<filesystem>` element, with the `action` attribute indicating the action to perform, and the `source` attribute specifying the file to operate on.

Depending on the action, the `destination` attribute is needed as well, as described below.

### 3.2 Create file

To create a file:

```
<file action="create"
      source="AppData\Vendorw\config.xml"></file>
```

This creates a `config.xml` file in the `AppData\Vendorw` directory.



To create an empty file, leave the `<file>` element empty. Otherwise, provide Base64-encoded data as the element content.

### 3.3 Copy file

To copy a file:

```
<file action="copy"
      source="AppData\VendorX\App1\settings.xml"
      destination="AppData\VendorX\App2\settings.xml" />
```

This copies the `settings.xml` file from the `AppData\VendorX\App1` directory to the `AppData\VendorX\App2` directory.

Note that it is also possible to rename the file while copying it: just specify a different file name in the `destination` attribute.

### 3.4 Move file

To move a file:

```
<file action="move"
      source="AppData\VendorY\App1\settings.xml"
      destination="AppData\VendorY\App2\settings.xml" />
```

This moves the `settings.xml` file from the `AppData\VendorY\App1` directory to the `AppData\VendorY\App2` directory.

Note that it is also possible to rename the file while moving it: just specify a different file name in the `destination` attribute.

Similarly, to just rename a file (without moving it to another folder) keep the folder part of the `destination` attribute identical to that of the `source`.

### 3.5 Delete file

To delete a file:

```
<file action="delete"
      source="AppData\VendorZ\obsolete.txt" />
```

This deletes the `obsolete.txt` file from the `AppData\VendorZ` directory.



### 3.6 Directory actions

Directory actions are specified as

```
<directory action="..." source="..." ... />
```

child elements of the `<filesystem>` element, with the `action` attribute indicating the action to perform, and the `source` attribute specifying the directory to operate on. Depending on the action, the `destination` attribute is needed as well, as described below.

### 3.7 Create directory

To create a directory:

```
<directory action="create"  
  source="AppData\VendorW\Data" />
```

This creates the `AppData\VendorW\Data` directory.

### 3.8 Copy directory

To copy a directory (with all its contents, i.e. recursively):

```
<directory action="copy"  
  source="AppData\VendorX\App1\Data"  
  destination="AppData\VendorX\App2\Data" />
```

This copies the `AppData\VendorX\App1\Data` directory to `AppData\VendorX\App2\Data`. All files and folders within `AppData\VendorX\App1\Data` will be copied.

### 3.9 Move directory

To move a directory:

```
<directory action="move"  
  source="AppData\VendorY\App1\Data"  
  destination="AppData\VendorY\App2\Data" />
```

This moves the `AppData\VendorY\App1\Data` directory to `AppData\VendorY\App2\Data`.

### 3.10 Delete directory

To delete a directory (with all its contents, i.e. recursively):

```
<directory action="delete"  
  source="AppData\VendorZ"
```

This deletes the `AppData\VendorZ` directory, including all files and folders within.