

# Getting Started with the VIX API for VMware Player

VMware Workstation 7.0 and VMware Player 3.0

---

This release of the VIX API, used in conjunction with Workstation 7.0 and Player 3.0, allows automation of virtual machine operations in Player using standard VIX API function calls.

## How to Use the VIX API with Player

You can use the VIX API with VMware Player in almost the same way that you use it with Workstation, with several exceptions:

- Some API functions are supported only for Workstation, not for Player.
- To use Player, the `VixHost_Connect` function must be called with different arguments.
- If Player is already open on the console when you power on a virtual machine, VIX launches a new instance of Player, rather than creating a new tab as in Workstation.

The combination of `vmrun` and VIX with Player gives users and developers the ability to build on award-winning VMware virtualization technology using the free Player application!

To learn more, see the online documentation at <http://www.vmware.com/support/developer/vix-api>.

## Supported and Unsupported API Functions

Player supports all VIX API functions except the following:

**Table 1.** VIX API Functions Not Supported in Player

<b>Registration</b>	<code>VixHost_RegisterVM</code> <code>VixHost_UnregisterVM</code>
<b>Power Operations</b>	<code>VixVM_Pause</code> <code>VixVM_Unpause</code>
<b>Snapshot Operations</b>	<code>VixSnapshot_GetChild</code> <code>VixSnapshot_GetNumChildren</code> <code>VixSnapshot_GetParent</code> <code>VixVM_CreateSnapshot</code> <code>VixVM_GetCurrentSnapshot</code> <code>VixVM_GetNamedSnapshot</code> <code>VixVM_GetNumRootSnapshots</code> <code>VixVM_GetRootSnapshot</code> <code>VixVM_RemoveSnapshot</code> <code>VixVM_RevertToSnapshot</code>

**Table 1.** VIX API Functions Not Supported in Player (Continued)

<b>Record/Replay Operations</b>	VixVM_BeginRecording
	VixVM_BeginReplay
	VixVM_EndRecording
	VixVM_EndReplay
<b>Virtual Machine Operations</b>	VixVM_Clone
	VixVM_UpgradeVirtualHardware

For a free product, it was a marketing decision to omit support for cloning, snapshots, record and replay, and pause/unpause. Registration is server-centric so Player does not support it.

## Connecting To and Powering On a Virtual Machine

Among the supported operations, only two API functions differ between Player and Workstation.

The only function whose arguments you need to change is `VixHost_Connect`. This function is typically one of the first VIX API functions you call in a VIX program. It obtains the host handle, which is then used to find or open Virtual Machines. As noted in the VIX API Reference, the signature for this function is as follows:

```
VixHandle
VixHost_Connect(int apiVersion,
                VixServiceProvider hostType,
                const char *hostName,
                int hostPort,
                const char *userName,
                const char *password,
                VixHostOptions options,
                VixHandle propertyListHandle,
                VixEventProc *callbackProc,
                void *clientData);
```

When using the VIX API with Workstation, clients call `VixHost_Connect` passing `VIX_SERVICEPROVIDER_VMWARE_WORKSTATION` as the `hostType` argument.

When using the VIX API with Player, simply call `VixHost_Connect` passing `VIX_SERVICEPROVIDER_VMWARE_PLAYER` instead as the `hostType` argument. For example:

```
jobHandle = VixHost_Connect(VIX_API_VERSION,
                            VIX_SERVICEPROVIDER_VMWARE_PLAYER,
                            NULL, //hostName
                            0, //hostPort
                            NULL, //userName
                            NULL, //password
                            0, //options
                            VIX_INVALID_HANDLE, //propertyListHandle
                            NULL, //callbackProc
                            NULL); //clientData
```

The only other function whose behavior changes noticeably when using VIX with Player is `VixVM_PowerOn`. Choosing Player as the `hostType` changes the behavior of `VixVM_PowerOn` when it is called with the option `VIX_VMPOWEROP_LAUNCH_GUI`. For example:

```
jobHandle = VixVM_PowerOn(vmHandle,
                          VIX_VMPOWEROP_LAUNCH_GUI, //powerOnOption
                          VIX_INVALID_HANDLE, //propertyListHandle
                          NULL, //callbackProc
                          NULL); //clientData
```

When the `hostType` is Player, this function launches the Player user interface instead of the Workstation user interface.

## What Happens if I Install Both Player and Workstation?

When both Workstation and Player are installed on a single host computer, both graphical user interfaces are available. However, it is important to realize that, under the covers, there is only one instance of the virtualization runtime. Workstation and Player act as skins over the same runtime.

For example, if you power on a virtual machine using Player when a computer also has Workstation installed, the virtual machine appears as powered-on in the Workstation user interface as well. Even though you see two separate user interfaces, only one instance of the virtual machine is running.

The VIX API behaves similarly. If you power on a virtual machine using `VixVM_PowerOn` with the `VIX_VMPOWEROP_LAUNCH_GUI` option, the Player or Workstation GUI is launched, depending on which `hostType` you initially passed to `VixHost_Connect`. However, even if you launch the Player user interface, when you subsequently open the virtual machine in Workstation, it appears powered-on.

If a virtual machine is already open in the Workstation user interface, the virtual machine's files are locked so you cannot power it on in Player.

The initial call to `VixHost_Connect` from a specific client-to-host pair controls the value of the connection options for subsequent calls. Also, if you logged into a guest through Workstation, logins are locked and guest operations from Player are prohibited.

## Example: Powering On a Virtual Machine and Launching a Player Window

The following sample code shows how to open and power on a virtual machine using Player.

```
//
VixError err = VIX_OK;
VixHandle hostHandle = VIX_INVALID_HANDLE;
VixHandle jobHandle = VIX_INVALID_HANDLE;
VixHandle vmHandle = VIX_INVALID_HANDLE;
//
// open hostType Player
jobHandle = VixHost_Connect(VIX_API_VERSION,
                           VIX_SERVICEPROVIDER_VMWARE_PLAYER,
                           NULL, // hostName
                           0, // hostPort
                           NULL, // userName
                           NULL, // password
                           0, // options
                           VIX_INVALID_HANDLE, // propertyListHandle
                           NULL, // callbackProc
                           NULL); // clientData
err = VixJob_Wait(jobHandle,
                 VIX_PROPERTY_JOB_RESULT_HANDLE,
                 &hostHandle,
                 VIX_PROPERTY_NONE);
if (VIX_OK != err) {
    // Handle the error...
    goto abort;
}
Vix_ReleaseHandle(jobHandle);

jobHandle = VixVM_Open(hostHandle,
                      "c:\\Virtual Machines\\vm1\\win2000.vmx",
                      NULL, // callbackProc
                      NULL); // clientData
err = VixJob_Wait(jobHandle,
                 VIX_PROPERTY_JOB_RESULT_HANDLE,
                 &vmHandle,
                 VIX_PROPERTY_NONE);
if (VIX_OK != err) {
    // Handle the error...
    goto abort;
}
Vix_ReleaseHandle(jobHandle);
```

```
// This will launch the Virtual Machine in the Player GUI
jobHandle = VixVM_PowerOn(vmHandle,
    VIX_VMPowerOP_LAUNCH_GUI, // powerOnOptions,
    VIX_INVALID_HANDLE, // propertyListHandle,
    NULL, // callbackProc,
    NULL); // clientData
err = VixJob_Wait(jobHandle,
    VIX_PROPERTY_NONE);

abort:
    Vix_ReleaseHandle(jobHandle);
    Vix_ReleaseHandle(vmHandle);
    VixHost_Disconnect(hostHandle);
```

---

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 © 2009 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-000238-00

15 OCTOBER 2009

---