# UNIVERSAL APP INSTALLER - QUICK START GUIDE

## 🚀 Get Started in 5 Minutes

This guide will help you quickly integrate the Universal App Installer into your orchestration framework.

---

## Step 1: Copy Files to Your Deployment Structure

```
Your-Deployment-Folder/
├── Orchestration-Master.ps1
├── Orchestration-Config.ps1
├── Scripts/
│   └── Universal-AppInstaller.ps1        ← Copy here
└── Installers/
    └── Apps/
        ├── 7z2408-x64.msi                ← Place your installers here
        ├── npp.8.6.9.Installer.x64.exe
        ├── vlc-3.0.21-win64.exe
        └── ... (more installers)
```

**File Locations:**

1. **Universal-AppInstaller.ps1** → `Scripts\Universal-AppInstaller.ps1`
2. **Application Installers** → `Installers\Apps\`
3. **Configuration** → Add to `Orchestration-Config.ps1` (see Step 2)

---

## Step 2: Add Applications to Configuration

Open your `Orchestration-Config.ps1` file and find the `$Phase4_Applications` section.

### A. First Application (7-Zip Example)

Add this to your Phase 4 tasks array:

```
powershell
```

```
$Phase4_Applications = @{
    Enabled = $true
    PhaseName = "Application Installation"
    PhaseDescription = "Install standard enterprise applications"
    StopOnPhaseFailure = $false
    Tasks = @(

        # Your existing tasks (if any)...

        # NEW: Universal Installer Apps
        @{
            TaskID = "APP-010"
            TaskName = "Install 7-Zip"
            ScriptPath = "Scripts\Universal-AppInstaller.ps1"
            Enabled = $true
            Timeout = 600
            RunAs = "SYSTEM"
            RequiresReboot = $false
            AllowRetry = $true
            Critical = $false
            Description = "Installs 7-Zip file compression utility"
            Parameters = @{
                AppName = "7-Zip"
                InstallerFileName = "7z2408-x64.msi"
                InstallerType = "MSI"
                InstallArguments = "/quiet /norestart"
                DetectionMethod = "Registry"
                DetectionPath = "HKLM:\SOFTWARE\7-Zip"
                DetectionValue = "Path"
            }
        }

        # Add more apps here...
    )
}
```

### B. Add More Applications

Use the template from `Phase4-Applications-Sample.ps1` or create your own:

```
powershell
```

```
@{
    TaskID = "APP-0XX"                          # Unique ID
    TaskName = "Install [App Name]"             # Display name
    ScriptPath = "Scripts\Universal-AppInstaller.ps1"   # Always this
    Enabled = $true                             # Enable it
    Timeout = 600                               # 10 minutes
    RunAs = "SYSTEM"                            # Run as SYSTEM
    RequiresReboot = $false                     # Usually false
    AllowRetry = $true                          # Enable retry
    Critical = $false                           # Not critical
    Description = "Installs [App Name]"         # Description
    Parameters = @{
        AppName = "[Application Display Name]"
        InstallerFileName = "[installer-file.exe/msi]"
        InstallerType = "AUTO"                  # AUTO, MSI, EXE
        InstallArguments = "[/S or /quiet]"
        DetectionMethod = "[Registry/File/Package]"
        DetectionPath = "[path or registry key]"
        DetectionValue = ""                     # Optional
    }
}
```

---

## Step 3: Test Your First Application

### A. Test Installation Manually

Before adding to orchestration, test the installer:

```powershell
# Navigate to installer location
cd C:\Deploy\Installers\Apps

# Test the silent install
.\7z2408-x64.msi /quiet /norestart

# Verify installation
Test-Path "C:\Program Files\7-Zip"
Get-ItemProperty "HKLM:\SOFTWARE\7-Zip"
```

### B. Test with Universal Installer Script

```powershell
# Test the Universal Installer script directly
cd C:\Deploy\Scripts

.\Universal-AppInstaller.ps1 `
    -AppName "7-Zip" `
    -InstallerFileName "7z2408-x64.msi" `
    -InstallerType "MSI" `
    -InstallArguments "/quiet /norestart" `
    -DetectionMethod "Registry" `
    -DetectionPath "HKLM:\SOFTWARE\7-Zip" `
    -DetectionValue "Path"

# Check exit code
$LASTEXITCODE
# 0 = Success, 10 = Already installed, other = error
```

### C. Check the Log

```powershell
# View the installation log
Get-Content "C:\ProgramData\OrchestrationLogs\Apps\Install-7-Zip_*.log" -Tail 50
```

---

## Step 4: Run Full Orchestration

### A. Dry Run (Recommended First)

Test without making changes:

```powershell
.\Orchestration-Master.ps1 -Phase Phase4 -DryRun
```

### B. Run Phase 4 Only

Install just the applications:

```powershell
```

```
.\Orchestration-Master.ps1 -Phase Phase4
```

### C. Run Full Orchestration

Run all phases including applications:

```
powershell

.\Orchestration-Master.ps1
```

---

## Step 5: Monitor and Review

### A. Watch Real-Time Progress

The orchestration will show:

- ✅ Green for success
- ⚠️ Yellow for warnings
- ❌ Red for errors

### B. Review Logs

```
powershell

# Main orchestration log
Get-Content "C:\ProgramData\OrchestrationLogs\Orchestration_*.log"

# Individual app logs
Get-ChildItem "C:\ProgramData\OrchestrationLogs\Apps\"

# View specific app log
Get-Content "C:\ProgramData\OrchestrationLogs\Apps\Install-7-Zip_*.log"
```

### C. Check Installation Status

```
powershell
```

```
# Verify app was installed
Get-Package -Name "7-Zip*"

# Or check file existence
Test-Path "C:\Program Files\7-Zip\7z.exe"
```

---

## Common Scenarios

### Scenario 1: Simple MSI Application

### Example: 7-Zip, WinRAR, PuTTY

```
powershell

Parameters = @{
    AppName = "7-Zip"
    InstallerFileName = "7z2408-x64.msi"
    InstallerType = "MSI"
    InstallArguments = "/quiet /norestart"
    DetectionMethod = "Registry"
    DetectionPath = "HKLM:\SOFTWARE\7-Zip"
}
```

### Scenario 2: Simple EXE Application

### Example: Notepad++, VLC, FileZilla

```
powershell

Parameters = @{
    AppName = "Notepad++"
    InstallerFileName = "npp.8.6.9.Installer.x64.exe"
    InstallerType = "EXE"
    InstallArguments = "/S"
    DetectionMethod = "File"
    DetectionPath = "C:\Program Files\Notepad++\notepad++.exe"
}
```

### Scenario 3: Application with Version Check

### Example: VLC with minimum version requirement

```powershell
Parameters = @{
    AppName = "VLC Media Player"
    InstallerFileName = "vlc-3.0.21-win64.exe"
    InstallerType = "EXE"
    InstallArguments = "/L=1033 /S"
    DetectionMethod = "File"
    DetectionPath = "C:\Program Files\VideoLAN\VLC\vlc.exe"
    RequiredVersion = "3.0.21"  # Will upgrade if older version found
}
```

**Scenario 4: Modern MSIX Application**

**Example: Windows Terminal, modern apps**

```powershell
Parameters = @{
    AppName = "Windows Terminal"
    InstallerFileName = "Microsoft.WindowsTerminal.msixbundle"
    InstallerType = "MSIX"
    InstallArguments = ""
    DetectionMethod = "AppX"
    DetectionPath = "Microsoft.WindowsTerminal"
}
```

**Scenario 5: Application Already Installed**

When the Universal Installer detects an app is already installed:

- ✅ Returns exit code 10 (success - already installed)
- ⏭️ Skips installation
- ✅ Orchestration continues to next task
- 📊 Logged as successful (no action needed)

---

## Troubleshooting Quick Reference

| Issue | Quick Fix |
| --- | --- |
| **Installer not found** | Check file name and location in `Installers\Apps\` |

| Issue | Quick Fix |
| --- | --- |
| **Installation fails** | Test installer manually with silent switches |
| **Detection fails** | Verify detection path exists after manual install |
| **Wrong exit code** | Review app-specific log in `OrchestrationLogs\Apps\` |
| **Timeout** | Increase `Timeout` value in configuration |
| **Already installed but re-runs** | Check detection method is finding the app |

### Quick Diagnostic Commands

```powershell
# Check if installer file exists
Test-Path "C:\Deploy\Installers\Apps\7z2408-x64.msi"

# Find where app installed
Get-Package -Name "7-Zip*"
Get-ItemProperty "HKLM:\SOFTWARE\7-Zip"

# View recent log
Get-Content "C:\ProgramData\OrchestrationLogs\Apps\Install-7-Zip_*.log" | Select-Object -Last 30

# Check orchestration checkpoint
Get-Content "C:\ProgramData\OrchestrationLogs\Checkpoint.xml"
```

---

## Your First 5 Applications

Here's a recommended starter set that works well with Universal Installer:

### 1. 7-Zip (File Compression)

```powershell
```

```
@{
    TaskID = "APP-010"
    TaskName = "Install 7-Zip"
    Parameters = @{
        AppName = "7-Zip"
        InstallerFileName = "7z2408-x64.msi"
        InstallerType = "MSI"
        InstallArguments = "/quiet /norestart"
        DetectionMethod = "Registry"
        DetectionPath = "HKLM:\SOFTWARE\7-Zip"
    }
}
```

### 2. Notepad++ (Text Editor)

powershell

```
@{
    TaskID = "APP-020"
    TaskName = "Install Notepad++"
    Parameters = @{
        AppName = "Notepad++"
        InstallerFileName = "npp.8.6.9.Installer.x64.exe"
        InstallerType = "EXE"
        InstallArguments = "/S"
        DetectionMethod = "File"
        DetectionPath = "C:\Program Files\Notepad++\notepad++.exe"
    }
}
```

### 3. Adobe Reader (PDF Viewer)

powershell

```
@{
    TaskID = "APP-030"
    TaskName = "Install Adobe Acrobat Reader"
    Parameters = @{
        AppName = "Adobe Acrobat Reader DC"
        InstallerFileName = "AcroRdrDC2400221005_en_US.exe"
        InstallerType = "EXE"
        InstallArguments = "/sAll /rs /msi EULA_ACCEPT=YES"
        DetectionMethod = "Registry"
        DetectionPath = "HKLM:\SOFTWARE\WOW6432Node\Adobe\Acrobat Reader\DC\Installer"
    }
}
```

### 4. VLC (Media Player)

powershell

```
@{
    TaskID = "APP-040"
    TaskName = "Install VLC Media Player"
    Parameters = @{
        AppName = "VLC Media Player"
        InstallerFileName = "vlc-3.0.21-win64.exe"
        InstallerType = "EXE"
        InstallArguments = "/L=1033 /S"
        DetectionMethod = "File"
        DetectionPath = "C:\Program Files\VideoLAN\VLC\vlc.exe"
    }
}
```

### 5. PuTTY (SSH Client)

powershell

```
@{
    TaskID = "APP-050"
    TaskName = "Install PuTTY"
    Parameters = @{
        AppName = "PuTTY"
        InstallerFileName = "putty-64bit-0.81-installer.msi"
        InstallerType = "MSI"
        InstallArguments = "/quiet /norestart"
        DetectionMethod = "File"
        DetectionPath = "C:\Program Files\PuTTY\putty.exe"
    }
}
```

---

## Next Steps

Once your first 5 apps are working:

1. ✅ **Add More Apps** - Use examples from `Phase4-Applications-Sample.ps1`
2. ✅ **Optimize Detection** - Fine-tune detection methods for reliability
3. ✅ **Version Control** - Track app versions in your configuration
4. ✅ **Documentation** - Document any special requirements per app
5. ✅ **Automation** - Schedule regular deployments with SCCM or Task Scheduler

---

## Getting Help

### Check the Logs

```powershell
# Always check the logs first
C:\ProgramData\OrchestrationLogs\Apps\Install-[AppName]_*.log
```

### Review Documentation

- `Universal-AppInstaller-Documentation.md` - Full documentation
- `Phase4-Applications-Sample.ps1` - 20+ working examples

**Test Incrementally**

1. Test installer manually first
2. Test with Universal Installer script directly
3. Test with orchestration Phase 4 only
4. Test with full orchestration

---

## Success Checklist

Before deploying to production:

- [ ] Universal-AppInstaller.ps1 copied to Scripts folder
- [ ] All installer files copied to Installers\Apps folder
- [ ] Each app tested manually with silent switches
- [ ] Each app tested with Universal Installer script
- [ ] Detection methods verified (app found after install)
- [ ] Phase 4 tested with dry run
- [ ] Phase 4 tested with actual installation
- [ ] Logs reviewed for errors
- [ ] Apps verified in Windows Programs & Features
- [ ] Apps verified to launch and work correctly

---

## Ready to Deploy!

You're now ready to use the Universal App Installer in your orchestration framework. This template will handle 80% of your application deployments with minimal configuration effort.

**Remember:** For complex apps like Microsoft Office, SQL Server, or apps requiring extensive configuration, continue using custom scripts. The Universal Installer is perfect for standard applications with straightforward installation requirements.

Happy deploying! 🚀