

---

# Tivoli Workload Scheduler

## OpenVMS Extended Agent

### Introduction

The VMS Extended Agent makes it possible for Tivoli Workload Scheduler for z/OS and Tivoli Workload Scheduler customers to schedule jobs on VAX/VMS, VAX/OpenVMS or Alpha OpenVMS systems. This provides clients with the ability to coordinate scheduling activity between VMS jobs and jobs scheduled on any platform or application supported by Tivoli Workload Scheduler.

### VMS Extended Agent Overview

The VMS extended agent (XA) extends the functionality of a Tivoli Workload Scheduler End to End or distributed scheduling network to any VMS nodes. Any UNIX in a TWS scheduling network can be designated as the host of VMS XA workstations. Jobs can be scheduled as any VMS user and can be directed to execute in any VMS batch queue. The X-Agent supports defining job or schedule dependencies on the existence of a VMS file.

To accomplish job execution and file checking on a VMS node, the UNIX host Workstation invokes a method program called `vmsssh`. The `vmsssh` method is a program that performs secure shell operations on a VMS system to launch and manage the execution of a VMS jobs. The `vmsssh` method has a supporting DCL command procedure called `vmsssh.wrp` that is installed on each VMS node that will be part of a TWS network.

### Job Documentation

The VMS jobs must be documented with the “Logon” field specifying a UNIX User ID. The UNIX User Id will be used execute the “`vmsssh`” method when launching and managing the job. The Script field for VMS jobs must be defined using the following syntax.

```
[-q {queue name}] [-u {VMS username}] {command or command procedure}
```

The ‘-q’ qualifier is only required if you do not want to submit the job into the default queue `sys$batch`. The ‘-u’ qualifier is required when the UNIX account specified in the “logon” field of the job definition does not exactly match the VMS account name that must be used to execute the job. Note that in any case a trust relationship must be established between the UNIX user and the VMS user. The command or command procedure portion of the “Script” field must be entered as if it were being executed interactively. That is, command procedures must be preceded by the ‘@’ symbol.

### File Dependencies

A File dependency in TWS is used when a job or schedule must wait until a file is available on a specific Workstation. The VMS extended agent supports the OPENS dependency by allowing any job or schedule on any TWS Workstation to wait for the existence of a file on a VMS extended agent Workstation.

When checking for the existence of files on a VMS extended agent Workstation, the vmsssh method is executed check for the existence of a file. The method establishes a secure shell connection into the VMS node to check the status of the file and returns the status to TWS. The method considers the file to exist if any version of the file is present in the specified directory.

## **Non Standard Exit Status**

When the VMS operating system runs a program or DCL procedure, the application has an opportunity to return information about the success or failure of the job via the \$status variable. Encoded within the value assigned to the \$status variable is whether the job ran normally, with warnings, with errors, or with severe errors. The TWS VMS Extended Agent reports any job with warnings, errors or severe as an ABEND.

In some cases, applications always return a normal value regardless of the success or failure of the job. Changing the application, DCL Procedure or program, may be possible but impractical in the short term. In other cases, changing the exit status may not be possible because the application runs within a programming environment that always returns a normal status.

For situations where the exit status of a job is not reliable, the VMS Extended Agent provides the ability to scan a job's output to determine if any error indications can be found. To enable this feature, create the file `SYS_TWS:[000000]VMSSSH.PATTERNS`. Then edit the file and add any text strings that indicate a job ABEND. Each string should be listed on a separate line of the file. Any time a job returns a normal status, the job log is searched for the error strings listed in `VMSSSH.PATTERNS`. If any strings match the job's status is returned to TWS as ABEND.