



# Web Services Integration Toolkit for OpenVMS

Installation Guide and Release Notes

November 2016

## **Software Version:**

Web Services Integration Toolkit  
Version 3.4-2

## Introduction

VMS Software Inc. is pleased to provide you with this new version of the Web Services Integration Toolkit for OpenVMS on Integrity servers.

The Web Service Integration Toolkit for OpenVMS (WSIT) contains a collection of integration tools. These tools are easy to use, highly extensible, based on standards, and built on Open Source technology. The toolkit can be used to call OpenVMS applications written in 3GL languages, such as C, BASIC, COBOL, FORTRAN, and ACMS from newer technologies and languages such as Java, Microsoft .NET, Java -RMI, JMS, and web services.

## What's new in this release

This release includes support for Java™ 8. The release is otherwise the same as the 3.4-1 release; however this new 3.4-2 release is not compatible with Java™ 6, and similarly WSIT 3.4.-1 is not compatible with Java™ 8.

This kit should only be used by users developing applications using the Java™ Development Kit (JDK) 8 for the OpenVMS Integrity servers.

## Requirements

The Web Services Integration Toolkit-generated server component is a native OpenVMS image installed on the system running the wrapped application. The generated source code must be built on the OpenVMS system that hosts the application using command-line driven development tools.

## Hardware Requirements

The Web Services Toolkit for OpenVMS I64 requires approximately 80,000 blocks of disk space.

## Software Installation, Development, and Runtime Requirements

Following are the minimum system requirements needed to install the Web Services Integration Toolkit and to build and run the generated components. Optional requirements represent requirements that are not needed for every type of application. However, several such requirements will apply to your particular platform/component selection.

- **Installation requirements**

- OpenVMS I64 Version 8.4-1H1 or higher
- ODS-5 disk

- **Development requirements**

- VSI C V7.4-001 or higher for OpenVMS (the C compiler is required to build all server components)
- Java™ Development Kit (JDK) 8 for the OpenVMS Integrity servers

- **Runtime requirements**

- Java™ Development Kit (JDK) 8 for the OpenVMS Integrity servers

- **Out-of-process account preparation and requirements**

If you are specifying an account in which to run out-of-process servers, you may want the account to have a minimal amount of privileges. You can specify an account to run out-of-process servers that has only the `NETMBX` and `TMPMBX` privileges. To use an account with these privileges, perform the following steps:

1. Create an identifier within the system UAF with the name `WSI$SERVER`. Perform this step one time only.
2. Grant the `WSI$SERVER` identifier to each account used to run a WSIT out-of-process server.

If you do not perform these steps, the privileges required by the account are as follows:

- `BYPASS`
- `SYSNAM`
- `SYSPRV`
- `IMPERSONATE`
- `DETACH`
- `TMPMBX`

## Installation

Installation involves installing and then starting the Web Services Integration Toolkit on OpenVMS.

- **Before you begin**

If you have previously installed a version of the Web Services Toolkit, make sure the Web Services Toolkit is shut down by entering this command from the `SYSTEM` account:

```
$ @SYS$STARTUP:WSI$SHUTDOWN.COM
```

VMS Software Inc. recommends that you perform a system disk backup before proceeding.

- **Installation**

The kit is provided as an OpenVMS PCSI kit (`VSI-I64VMS-WSIT-V0304-2-1.PCSI`) that can be installed by a suitably privileged user using the following command:

```
$ PRODUCT INSTALL WSIT/DESTINATION=ods5disk:[directory]
```

The Web Services Toolkit installation procedure accepts a destination directory specified on the PCSI install command line. This location is used as the root directory for the WSIT product. If the destination is not specified, the default is `SYS$COMMON:[000000]` and the directory created is `SYS$COMMON:[WSIT]`.

Follow the on-screen installation instructions to complete the installation. To proceed with a default installation, press Enter/Return in response to any installation questions.

- **Installing the WSIT runtime only**

During the WSIT installation procedure, you are asked if you want to install only the runtime file. If you answer YES, the WSIT tools, documentation and sources will not be installed.

```
Do you want the defaults for all options? [YES] n
```

```
    Install the full WSIT kit? [YES] n
```

```
    Install the WSIT runtime only? [YES] y
```

```
Do you want to review the options? [NO] n
```

```
Execution phase starting ...
```

## After Installing the Web Services Integration Toolkit for OpenVMS

To complete the installation, perform the following steps:

- **Modify your system start-up command procedure (`SYS$MANAGER:SYSTARTUP_VMS.COM`) to include the Web Services Toolkit start-up command:**

```
$ @SYS$STARTUP:WSI$STARTUP.COM
```

- **Modify your system shutdown command procedure (`SYS$MANAGER:SYSHUTDWN.COM`) to include the Web Services Toolkit shutdown command:**

```
$ @SYS$STARTUP:WSI$SHUTDOWN.COM
```

- **Finally, start the Web Services Integration Toolkit with this command:**

```
$ @SYS$STARTUP:WSI$STARTUP.COM
```

This completes the Web Services Integration Toolkit for OpenVMS installation.

Note that if you opt not to use the `SYSTEM` account, the Web Services Toolkit start-up procedure must still be run every time your system starts. It can be run as part of your system start-up or in another account.

It is recommended that you use the `SYSTEM` account. However, the account in which you run the Web Services Toolkit must have, minimally, the following authorized account privileges:

- `BYPASS`

- SYSPRV
- TMPMBX
- SYSNAM
- CMKRNL
- DETACH
- LOG\_IO

## Starting and Stopping the Web Services Integration Toolkit for OpenVMS

- Use the following command to start the Web Services Integration Toolkit:

```
$ @SYS$STARTUP:WSI$STARTUP.COM
```

- Stopping the Web Services Integration Toolkit requires the same privileges as starting it. Ensure that all client connections have been closed and from the `SYSTEM` account run the following command:

```
$ @SYS$STARTUP:WSI$SHUTDOWN.COM
```

## Uninstalling the Web Services Integration Toolkit

To remove the Web Services Toolkit for OpenVMS from your system, enter the following command:

```
$ PRODUCT REMOVE WSIT
```

## Web Services Integration Toolkit Directory Structure

After you install the Web Services Integration Toolkit, browse the `WSIT` directories to see the location of the tools and sample programs.

- **WSIT root directory**

The root directory is represented by the logical `WSI$ROOT`. This logical is created when the `WSI$STARTUP.COM` procedure is executed.

```
$ directory wsi$root:[000000]
```

```
Directory WSI$ROOT:[000000]
```

```
Apache_Software_License_V1_1.txt;1      Apache_Software_License_V2_0.txt;1
deploy.dir;1          docs.dir;1          lib.dir;1          logs.dir;1
samples.dir;1        src.dir;1          tools.dir;1       wsi-version.txt;1
```

```
Total of 10 files.
```

- **Deploy subdirectory**

The `[.deploy]` subdirectory contains the shareable image for the application's server wrapper. The generated JavaBean calls the server wrapper's shareable image at runtime.

- **Docs subdirectory**

The `[.docs]` directory contains Web Services Integration Toolkit documentation.

- **LIB subdirectory**

The `[.lib]` subdirectory is where the Web Services Integration Toolkit stores the files it uses internally; for example, Velocity template files.

- **Samples subdirectory**

The [.samples] subdirectory contains example programs. This kit includes sample applications written in ACMS, C, COBOL, and FORTRAN.

The .C, .CBL and .FOR programs are the wrappers for the original application. The .JAVA files illustrate calling the WSIT-generated JavaBean for each wrapper. See the Developer's Guide Appendix for the source files of the C sample.

- **Tools subdirectory**

The tools in this directory are used to develop code that wraps 3GL applications.

- **Source subdirectory**

The [.source] subdirectory contains objects and classes used by the generated code, including Java exceptions and holder classes. The files in this subdirectory are provided for your reference.

## Performance Recommendations

- **Operating system requirements**

Increase the PRCLM quota to be high enough to run the maximum number of expected concurrent servers.

- **Java requirements**

The Java runtime environment was designed to perform optimally on UNIX systems, where each process is given large quotas by default. On OpenVMS, the default behaviour gives each process lower quotas so that many processes can co-exist on a system. To get the best Java performance on OpenVMS, you are recommended to set process quotas to match a typical UNIX system. These are also the recommended minimum quota settings (except where noted).

These are the recommended J2SDK for OpenVMS quotas:

UAF FILLM	4096
CHANNELCNT	4096
WSDEF	2048
WSQUOTA	4096
WSEXTENT and WSMAX	16384
PGFILQUO	2,097,152 (see comments below)
BYTLM	2,000,000
BIOLM	150
DIOLM	150
TQELM	100

A good number for PGFLQUO is (2 x heap-size), for example, 128 MB (2\*128\*1024\*1024)/512 = 524288. The recommended minimum PGFLQUO is 96 MB when using the WSIT RTE. When you increase the PGFLQUO parameter, you should always increase the system's page file size to accommodate the new PGFLQUO parameter, if needed.

Note that if you receive a `WSIConnectException` without a specific message attached to it, the most likely cause is a value for `BYTLM` that is too low. VMS Software Inc. recommends that you set this value to at least 2,000,000.

## Known problems and issues

- **Known problem in AXIS2 java2WSDL tool affecting case**

The AXIS2 `java2WSDL` tool modifies the name of all properties to begin with a lowercase letter. For example, if the IDL XML has a structure with a field named `EMPLOYEE_NAME`, the AXIS2 `java2WSDL` tool generates WSDL with the element `eMPLOYEE_NAME`. Until this matter is resolved, you must choose between the following:

- Using lowercase first letters in the WSIT IDL where appropriate; or
- Including your own WSDL file in your AXIS2 archive file (`.aar`). In this scenario, you can run the tool by hand and then modify the WSDL and add it to the archive.

The elements that need to be addressed include structure field names and parameter element names when the usage is specified as `IN/OUT` or `OUT`. The parameters are included because the SOAP response class is a composite of the returned parameters.

- **Copying WSIRTL.JAR to server classpath (if required)**

If you plan to call a Web Services Integration Toolkit application from a web server or application server, the file `wsi$root:[lib]wsirtl.jar` must be in the servers classpath. The details on how to configure the server to include this jar file are specific to each server.

For example, if you are using `CSWS_JAVA` (Apache Tomcat for OpenVMS), you should copy the file `wsi$root:[lib]wsirtl.jar` to the `common/lib` directory. To ensure that the server becomes aware of the file, a restart is often required. However, this is also server-specific.