

# **Web Services Integration Toolkit for OpenVMS**

---

## **Installation Guide and Release Notes**

September 2005

This document contains information that will help you install and use this release of WSIT for OpenVMS.

### **Software Version**

Web Services Integration Toolkit Version 1.0

Hewlett-Packard Company  
Palo Alto, Calif.

---

**© 2005 Hewlett-Packard Development Company, L.P.**

Intel, Intel Inside, and Itanium are trademarks of Intel Corporation in the U.S. and/or other countries.

Microsoft, Windows, Windows XP, Visual Basic, Visual C++, and Win32 are trademarks of Microsoft Corporation in the U.S. and/or other countries.

Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

All other product names mentioned herein may be trademarks of their respective companies.

Confidential computer software. Valid license from HP and/or its subsidiaries required for possession, use, or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Neither HP nor any of its subsidiaries shall be liable for technical or editorial errors or omissions contained herein. The information in this document is provided "as is" without warranty of any kind and is subject to change without notice. The warranties for HP products are set forth in the express limited warranty statements accompanying such products. Nothing herein should be construed as constituting an additional warranty.

## CONTENTS

---

### 1. REQUIREMENTS

- 1.1 Hardware Requirements
- 1.2 Software Installation, Development, and Runtime Requirements

### 2. INSTALLATION

- 2.1 Installing the Web Services Integration Toolkit for OpenVMS
- 2.2 After Installing the Web Services Integration Toolkit for OpenVMS
- 2.3 Starting and Stopping the Web Services Integration Toolkit
- 2.4 Uninstalling the Web Services Integration Toolkit
- 2.5 Web Services Integration Toolkit Directory Structure
- 2.6 Performance Recommendations

### 3. RELEASE NOTES

- 3.1 Support for Web Services Integration Toolkit Version 1.0
- 3.2 Languages Tested with Web Services Integration Toolkit
- 3.3 Design Restrictions
- 3.4 Instantiated WSIT JavaBeans Cannot Be Shared Among Clients
- 3.5 Tools and Files Renamed in Field Test Kit
- 3.6 Tips and Hints for Supported Languages

### 4. CONTACT AND SUPPORT INFORMATION

#### APPENDIX

- A. Sample FTP File Transfer
- B. Sample OpenVMS Installation Log

### About Web Services Integration Toolkit for OpenVMS Documentation

This *Installation Guide and Release Notes* includes system requirements and installation instructions for OpenVMS, as well as release notes for the current release of the Web Services Integration Toolkit for OpenVMS.

The *Developer's Guide* contains information about how to use the tools in the Web Services Integration Toolkit for OpenVMS, and things to consider as you prepare your legacy application.

For the latest release information, refer to the Web Services Toolkit for OpenVMS web site at <http://www.hp.com/products/openvms/webservices/>.

## 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.

### 1.1 Hardware Requirements

The Web Services Toolkit for OpenVMS is available on OpenVMS I64 and OpenVMS Alpha.

The Web Services Toolkit for OpenVMS **I64** EXE file requires approximately 48,500 blocks of disk space. Expanding the EXE file requires an additional 68,000 blocks.

The Web Services Toolkit for OpenVMS **Alpha** EXE file requires approximately 26,500 blocks of disk space. Expanding the EXE file requires an additional 60,000 blocks.

### 1.2 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.

**Make sure you have the latest MUPs, ECOs, and patches for your configuration. Search for these on the Patch Download page at:**

**<http://www2.itrc.hp.com/service/patch/mainPage.do>**

#### **Always Check Cross-Product Requirements!**

Always check for cross-product software requirements, which can change with new releases. These product requirements may be greater than for the Web Services Integration Toolkit itself.

For example, to run a Web Services Integration Toolkit-generated JavaBean component requires OpenVMS Version 8.2 or higher and a Java 2 Runtime Environment 1.4.2-1 or higher. Each Java SDK and RTE has specific ECO requirements.

#### **Installation Requirements**

- OpenVMS I64 Version 8.2 or higher - or -
- OpenVMS Alpha Version 7.3-2 or higher

#### **Development Requirements**

The C Compiler is required to build all server components.

- HP C Version 6.5 or higher for OpenVMS
- HP C++ Version 6.5 or higher for OpenVMS
- Java 2 SDK ("JDK") Standard Edition 1.4.2-1 or higher

Java Technology for OpenVMS Download Page  
<http://h18012.www1.hp.com/java/download/>

## **Runtime Requirements**

- Java 2 RTE (JRE) Standard Edition 1.4.2-1 or higher

Java Technology for OpenVMS Download Page  
<http://h18012.www1.hp.com/java/download/>

- Java Software Patches (ECOs) (if required)

Java Technology for OpenVMS v 1.4.2 Software Patch Page  
[http://h18010.www1.hp.com/java/download/ovms/1.4.2/sdk1.4.2\\_patches.html](http://h18010.www1.hp.com/java/download/ovms/1.4.2/sdk1.4.2_patches.html)

# INSTALLATION

---

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

## 2.1 Installing the Web Services Integration Toolkit for OpenVMS

### Before You Begin

- Log into the SYSTEM account.
- If you have previously installed a field test 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
```

- HP recommends that you perform a system disk backup before proceeding.

### Web Download Instructions

1. Save the Web Services Toolkit self-extracting archive (`WSIT-V0100-I64.EXE` or `WSIT-V0100-AXP.EXE`) to a temporary location by choosing the option to save the program to your disk.
2. If you are downloading to a Windows machine, move the downloaded file to your OpenVMS machine using either a shared network drive or FTP. (See Appendix A: Sample FTP File Transfer.)

### Installation Instructions

1. Expand the Web Services Integration Toolkit compressed EXE file by entering one of the following command, depending on the platform on which you are installing the kit:

```
$ RUN WSIT-V0100-I64.EXE ! for OpenVMS I64
$ RUN WSIT-V0100-AXP.EXE ! for OpenVMS Alpha
```

The EXE file contains the file `HP-IA64VMS-WSIT-V0100--1.PCSI$COMPRESSED` (for OpenVMS I64) or `HP-AXPVMS-WSIT-V0100--1.PCSI$COMPRESSED` (for OpenVMS Alpha) and the WSIT documentation.

2. Start PCSI, the utility that installs the Web Services Toolkit component, by entering the following command. (PCSI installs the `PCSI$COMPRESSED` file directly; you do not need to expand it.)

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

The Web Services Toolkit requires a destination directory to be specified on the PCSI install command line. This location is used as the root directory for the WSIT product.

**The destination directory must be an ODS5 volume.**

3. Follow the on-screen installation instructions. (See Appendix B: Sample OpenVMS Installation Log.)

To proceed with a default installation, press Enter/Return in response to any other installation questions.

## 2.2 After Installing the Web Services Integration Toolkit for OpenVMS

ALWAYS restart J2EE environments, such as BEA WebLogic Server, or other Bean servers, such as Tomcat, after a Web Services Toolkit installation. Such environments do not automatically refresh shareable images and libraries without being restarted.

1. Modify your system startup command procedure (`SYS$MANAGER:SYSTARTUP_VMS.COM`) to include the Web Services Toolkit startup command:

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

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

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

3. 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: If you opt not to use the SYSTEM account**

The Web Services Toolkit startup procedure must still be run every time your system starts. It can be run as part of your system startup or in another account.

**Important:** You are recommended to 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
```

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

### Starting the Web Services Integration Toolkit

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

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

### Stopping the Web Services Integration Toolkit

Stopping the Web Services Integration Toolkit requires the same privileges as starting it. Ensure that all client connections have been closed.

1. Log into the SYSTEM account.

2. Use the following command to stop the Web Services Integration Toolkit:

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

## 2.4 Uninstalling the Web Services Integration Toolkit

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

```
$ PRODUCT REMOVE WSIT
```

## 2.5 Web Services Integration Toolkit Directory Structure

After you install the Web Services Integration Toolkit, look in the following directories to see the location of the tools and sample programs.

### 2.5.1 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.

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

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

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

```
Total of 7 files.
```

```
$
```

### 2.5.2 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.

```
$ dir [deploy]
```

```
Directory wsi$root:[deploy]
```

```
wsi-default.wsi;1
```

### 2.5.3 Docs Subdirectory

The Docs directory contains Web Services Integration Toolkit documentation.

```
$ dir [docs]
```

```
Directory wsi$root:[docs]
```

```
wsit-develop.doc;1  wsit-develop_files.dir;1  wsit-develop.html;1
wsit-develop.pdf;1  wsit-develop.ps;1  wsit-install.doc;1  wsit-install.html;1
wsit-install.pdf;1  wsit-install.ps;1
```



### 2.5.4 LIB Subdirectory

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

```
$ dir [lib]

Directory wsi$root:[lib]

velocity-dep-1_4.jar;1      wsi$ana2wsi.exe;1      wsi$stdl2wsi.exe;1
wsirtl.jar;1
```

### 2.5.5 Samples Subdirectory

The Samples subdirectory contains example programs. This kit includes a sample application written in ACMS, a sample written in C called STOCK.C, a sample written in COBOL called COBOLADD.CBL, and a sample written in FORTRAN called SWAP.FOR.

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.

```
$ dir [samples.acms]

Directory wsi$root:[samples.acms]

acms-sample.readme;1      acmscaller.java;1
acmsexample_setup.com;1   employee_info_appl_wsi.adf;1
wsi_add_empl_info.tdf;1   wsi_emp_info_task_group.gdf;1
wsi_get_empl_info.tdf;1   wsi_put_empl_info.tdf;1

$ dir [samples.c]

Directory wsi$root:[samples.c]

stock.c;1      stock.xml;1      stockcaller.java;1
stock-sample.readme;1

$ dir [samples.cobol]

coboladd.cbl;1      coboladd.xml;1      coboladdcaller.java;1

$ dir [samples.fortran]

Directory wsi$root:[samples.fortran]

swap.for;1      swap.xml;1      swapcaller.java;1
```

### 2.5.6 Tools Subdirectory

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

```
$ dir [tools]

Directory wsi$root:[tools]

bw2id1.readme;1      idl2code.jar;1      openvms-basetypes.xml;1
openvms-integration.xsd;1      stdl2idl.jar;1      templates.dir;1
```

```

validate.jar;1          wsi-setenv.com;1

Directory wsi$root:[tools.templates]

javabean.dir;1          master.vm;1          serverwrapper.dir;1
velocity.properties;1

Directory wsi$root:[tools.templates.javabean]

build-interface-jb-com.vm;1          iinterface-java.vm;1
interfaceimpl-java.vm;1             structure-java.vm;1
ws-iinterface-java.vm;1             ws-interfaceimpl-java.vm;1
ws-structure-java.vm;1

Directory wsi$root:[tools.templates.serverwrapper]

appname-opt.vm;1          appname-server-c.vm;1
appname-server-h.vm;1     appname-wsi.vm;1
build-appname-server-com.vm;1      methids-h.vm;1      structkeys-h.vm;1

```

### 2.5.7 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.

```

$ dir [src]

Directory wsi$root:[src]

unpack-wsisrc.com;1          wsisrc.jar;1

```

## 2.6 Performance Recommendations

### 2.6.1 Operating System Requirements

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

### 2.6.2 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 behavior 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*
bytlim	2,000,000
biolm	150

diolm	150
tqelm	100

\*A good number for `Pgflquo` is (2 x heap-size), for example, 128 MB  $(2 \times 128 \times 1024 \times 1024) / 512 = 524288$ . Recall that the recommended minimum `Pgflquo` is 96 MB when using the 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:** If you receive a `WSIConnectException` without a specific message attached to it, the most likely cause is a value for `bytln` that is too low. HP recommends that you set this value to at least 2,000,000.

There are many other optimizations you can make to your Java environment to improve performance, such as setting the compacting garbage collector with the Fast VM.

`-Xgc:compacting`

This will benefit some but not all Java applications. All such optimizations should be tested in a systematic fashion. You can try out the techniques and tips available from the following source.

**Guide to Optimizing Java on OpenVMS**

<http://h71000.www7.hp.com/ebusiness/OptimizingSDKGuide/OptimizingSDKGuide.html>

## RELEASE NOTES

---

### 3.1 Support for Web Services Integration Toolkit Version 1.0

Support for **Version 1.0** of the Web Services Integration Toolkit is provided by [HP Support](#). If you want to send technical feedback or comments to the Web Services for OpenVMS engineering team, please send mail to [OpenVMS.WebServices@hp.com](mailto:OpenVMS.WebServices@hp.com).

### 3.2 Languages Tested with Web Services Integration Toolkit

The Web Services Integration Toolkit has been tested with the following languages:

- ACMS
- BASIC
- C
- COBOL

Other languages (including FORTRAN) may work, but have had little or no testing in Version 1.0.

### 3.3 Design Restrictions

The Web Services Integration Toolkit runtime has the following restrictions. The descriptions are expressed in C syntax but are relevant for all languages.

**Note:** If a structure contains a pointer to a type, WSIT passes the pointer by value without any regard for the type. The application must handle the memory appropriately. This memory is only valid in the context of the application being wrapped and is not valid in the generated JavaBean.

<u>Restriction</u>	<u>Description</u>
Routine parameters cannot be defined as “pointer to pointer”.  However, a single pointer is supported.	<pre>// This is not supported void myfunction( int **p)  // This is supported void myfunction( int *p)</pre>
User-defined structures cannot contain pointers to other user-defined structures.  However, nested structures are supported.	<pre>// For the structure buyerData: typedef struct _buyerData {     char buyer_name[MAX_STRING] ;; } buyerData;  // This is not supported typedef struct _customerData {     buyerData *pbuyer; } customerData;  // This is supported typedef struct _customerData {     buyerData buyer; } customerData;</pre>

<p>Routine return values must be returned by value and cannot be user-defined structures.</p> <p>However, the interface can add a special parameter to the routine to return the same structure.</p>	<pre>// For the structure buyerData: typedef struct _buyerData {     char buyer_name[MAX_STRING]; } buyerData;  // This is not supported buyerData * buy()  // This is supported int buy(buyerData *return_value)</pre>
--	---

### 3.4 Instantiated WSIT JavaBeans Cannot Be Shared Among Clients

The WSIT runtime supports a many-to-one relationship between clients and backend servers. However, **each client must have its own copy of the JavaBean object** (unless you manually add synchronization code to the generated JavaBean class). The JavaBean object acts as the client's personal interface into the backend server. If a single JavaBean instantiation is shared among clients, unexpected results may occur, including incorrect call data and memory management exceptions.

See the Modifying an Existing Template section in the *Developer's Guide* for information about how to manually add synchronization code.

### 3.5 Tools and Files Renamed in Field Test Kit

The following tools and files were renamed in a pre-Version 1.0 field test kit, as shown in the following table.

Directory	Old Name	New Name
[wsit.tools]	idlgen.exe	obj2idl.jar
[wsit.tools]	wsigen.jar	idl2code.jar
[wsit.tools]	validateIDL.jar	validate.jar
[wsit.tools]	wsisetenv.com	wsisetenv.com
[wsit.tools]	OpenVMSbasetypes.xml	openvms-basetypes.xml
[wsit.tools.templates.javabeen]	structure_java.vm	structure-java.vm
[wsit.tools.templates.javabeen]	viewname_java.vm	interface-appname-java.vm
[wsit.tools.templates.javabeen]	build_jb_com.vm	build-appname-jb-com.vm
[wsit.tools.templates.javabeen]	viewnameImpl_java.vm	appname-impl-java.vm
[wsit.tools.templates.serverwrapper]	build_server_com.	build-appname-server-com.vm
[wsit.tools.templates.serverwrapper]	conn-name_opt.vm	appname-opt.vm
[wsit.tools.templates.serverwrapper]	conn-name_server_c.	appname-server-c.vm
[wsit.tools.templates.serverwrapper]	conn-name_server_h.	appname-server-h.vm
[wsit.tools.templates.serverwrapper]	conn-name_wsi.	appname-wsi.vm
[wsit.tools.templates.serverwrapper]	methIDs_h.vm	methlds-h.vm
[wsit.tools.templates.serverwrapper]	structkeys_h.vm	structkeys-h.vm
[wsit.deploy]	wsidefault.wsi	wsi-default.wsi

### 3.6 Tips and Hints for Supported Languages

#### 3.6.1 All Languages

The following issues apply to all WSIT-supported languages (ACMS, BASIC , C, and COBOL).

- **Using I64 generated XML IDL on Alpha:** Using an XML file generated on I64 is an excellent way to begin using the Web Services Toolkit on OpenVMS Alpha. However, be aware that some compilers may have different default values on I64 than on Alpha. These differences need to be addressed. For example, the C compiler uses a different `VMSTDataTypes` for primitive types *float* and *double*:

<u>C Primitive</u>	<u>Default value on I64</u>	<u>Default value on Alpha</u>
<i>float</i>	DSC\$K_DTYPE_FS	DSC\$K_DTYPE_F
<i>double</i>	DSC\$K_DTYPE_FT	DSC\$K_DTYPE_G

The preceding table lists all known differences in data types mapping for the supported languages. However, there might be other differences that are not listed here.

- **The exception `java.lang.UnsatisfiedLinkError: no WSI$JNISHR` may be generated:** If you have not started WSIT by calling `sys$startup:wsit$startup`, an exception is thrown when you run the generated JavaBean.
- **If a structure contains a pointer to a type:** WSIT passes the pointer by value without any regard for the type. The application must handle the memory appropriately. This memory is only valid in the context of the application being wrapped and is not valid in the generated JavaBean.

#### 3.6.2 BASIC Language

The following issues apply to the BASIC language only.

- **For formal parameters, passed by value, with types other than strings or signed decimal data:** `obj2idl.jar` defaults to a passing mechanism of *reference* and a usage of *IN/OUT*. Modify the XML to specify a passing mechanism of *value* and a usage of *IN*.
- **For formal parameters, passed by descriptor:** `obj2idl.jar` does not recognize the type of the formal parameter. In such cases, the type defaults to a null terminated string. Modify the XML to specify the correct type for the formal parameter. In some cases, you may need to add the type (primitive or structure) to the XML.
- **For formal parameters that are arrays, if the passing mechanism is descriptor:** `obj2idl.jar` does not recognize the arrays. Modify the XML to specify the array. (See the example of syntax in the C language section.)

#### 3.6.3 C Language

The following issues apply to the C language only.

- **When passing a formal parameter of byte (type `char`) by reference:** If a `char` is being used to represent a single byte and is passed by reference as a formal parameter, then the generated XML will

specify the parameter as having a type of `type="AutoGen_NullTermString"`. Modify the XML to specify `type="char"` which will properly resolve to a byte (DSC\$K\_DTYPE\_B).

- **For C applications, obj2idl.jar does not recognize formal parameters that are arrays:** For example, if an array of `userstruct` with size 3 is passed as a parameter, then the XML that is generated will be as follows:

```
<Routine Name = "incrementArrayOfStructures" ReturnType = "signed int">
<Parameter Name = "array" Type = "userstruct" PassingMechanism = "Reference"
Usage = "IN/OUT"/>
</Routine>
```

Modify the XML to specify the array as follows:

```
<Routine Name = "incrementArrayOfStructures" ReturnType = "signed int">
<Parameter Name = "array" Type = "userstruct" PassingMechanism = "Reference"
Usage = "IN/OUT" ArrayDimension = "1">
<Array LowerBound = "0" UpperBound = "2"/>
</Parameter>
</Routine>
```

#### 3.6.4 COBOL Language

The following issues apply to the COBOL language only.

- **For all formal parameters:** `obj2idl.jar` defaults the usage to IN/OUT.
- **For all formal parameters:** `obj2idl.jar` defaults the passing mechanism to pass by reference.

## CONTACT AND SUPPORT INFORMATION

---

### For OpenVMS Alpha

- Support for customer release versions of the Web Services Integration Toolkit for OpenVMS is provided by HP under OpenVMS Alpha service agreements through standard support channels.
- Download is an official distribution mechanism the Web Services Integration Toolkit, including updates in addition to the OpenVMS Alpha media kit.
- Source code kits are provided in the spirit of the open source community, but are not supported by HP.

### For OpenVMS on Integrity Servers

- Support for customer release versions of the Web Services Integration Toolkit for OpenVMS is provided by HP under OpenVMS I64 Foundation Operating Environment (FOE) service agreements through standard support channels.
- Download is a courtesy distribution mechanism for the Web Services Integration Toolkit for OpenVMS. Kits or updates other than security patch kits obtained via download are unsupported.
- The OpenVMS I64 Operating Environments media kit is the official distribution mechanism for the Web Services Integration Toolkit for OpenVMS. Supported kits and updates must be obtained from this mechanism or from software product update services.

A variety of service options are available from [HP Support](#). For more information, contact your local HP account representative or distributor. Information is also available from [Software Support Services](#).

You can informally exchange information with other users in the OpenVMS newsgroup `comp.os.vms`.

If you would like to provide comments about the Web Services Integration Toolkit for OpenVMS, please send mail to:

[OpenVMS.WebServices@hp.com](mailto:OpenVMS.WebServices@hp.com)

If you would like to provide comments or general feedback about the e-Business offerings on OpenVMS, please send mail to:

[OpenVMS.eBusiness@hp.com](mailto:OpenVMS.eBusiness@hp.com)

### Web Services Toolkit Home Page

<http://www.hp.com/openvms/products/webservices/>



## APPENDIX

---

### A Sample FTP File Transfer

If you downloaded the Web Services Integration Toolkit compressed EXE file to a Windows system, use commands similar to the following to move the kit to your OpenVMS system using command-line FTP (with user input in angle brackets):

```
C:\>cd temp
C:\TEMP>ftp <your.server.name.com>
Connected to your.server.name.com.
220 your.server.name.com FTP Server (Version 5.0) Ready.
User (your.server.name.com:(none)): <user>
331 Username Yourname requires a Password
Password: <password>
230 User logged in.
ftp> binary
200 TYPE set to IMAGE.
ftp> <put WSIT-V0100-I64.EXE>
200 PORT command successful.
150 Opening data connection for disk:[user]WSIT-V0100-I64.EXE; (16.28.0.58,4460)
226 Transfer complete.
335713 bytes sent in 0.40 seconds (837.19 Kbytes/sec)
ftp> <bye>
221 Goodbye.
C:\TEMP>
```

---

## B Sample OpenVMS Installation Log

This is a representative Web Services Integration Toolkit installation log for OpenVMS I64.

### **Step 1: Extract the Kit**

```
$ RUN WSIT-V0100-I64.EXE
UnZipSFX 5.51 of 22 May 2004, by Info-ZIP (http://www.info-zip.org).
  inflating: HP-IA64VMS-WSIT-V0100--1.PCSI$COMPRESSED
$
```

### **Step 2: Run PCSI (POLYCenter Software Installation) Utility**

```
$ PRODUCT INSTALL WSIT/source=dka100:[products.kits] /dest=dka100:[products]
```

```
The following product has been selected:
  HP IA64VMS WSIT V1.0                      Layered Product
```

```
Do you want to continue? [YES]
```

```
Configuration phase starting ...
```

```
You will be asked to choose options, if any, for each selected product and for
any products that may be installed to satisfy software dependency requirements.
```

```
HP IA64VMS WSIT V1.0: Web Services Integration Toolkit (WSIT) V1.0 for OpenVMS
  (C) 2005 Hewlett-Packard Development Company, L.P.
```

```
Do you want the defaults for all options? [YES]
Do you want to review the options? [NO]
```

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

```
The following product will be installed to destination:
  HP IA64VMS WSIT V1.0          I64ARCHIVE:[PRODUCTS.]
```

```
Portion done: 0%...10%...90%
*****
```

```
Insert the following lines in SYS$MANAGER:SYSTARTUP_VMS.COM:
  @sys$startup:wsit$startup.com
```

```
Insert the following lines in SYS$MANAGER:SYSHUTDOWN.COM:
  @sys$startup:wsit$shutdown.com
```

```
*****
Portion done: 100%
```

```
The following product has been installed:
  HP IA64VMS WSIT V1.0                      Layered Product
```

```
$
```

---