vms Software

XML Java Technology for OpenVMS Installation Guide and Release Notes

September 2015

Version 4.0, based on
Apache Xerces-Java Version 2.11.0 and
Apache Xalan-Java Version 2.7.1

Contents:

Before Installing XML Java Technology

Hardware Prerequisites

XML Java Technology for OpenVMS is available on OpenVMS Industry Standard 64 (I64). XML Java is based on Apache Xerces-Java Version 2.11.0 and Apache Xalan-Java Version 2.7.1.

The XML Java Technology for OpenVMS V4.0 compressed PCSI kit for OpenVMS I64 requires approximately 190,000 blocks of disk space. To install the product, a minimum of 350,000 blocks of disk space is required.

Software Prerequisites

- OpenVMS I64 Version 8.4-1H1 or higher
- HP Software Development Kit (SDK) for the Java™ Platform Version 1.6 (or higher)
- All patches required for the HP SDK for the Java™ Platform. Check the SDK documentation for the version of Java you are installing to be sure that you have all prerequisite OpenVMS patches.

Installing on an ODS-5 enabled disk is required. Because of long file names and directory depth issues, the installed code base and the accompanying documentation cannot be guaranteed to function properly in a non-ODS-5 environment.

Installing XML Java Technology

To install the XML Java Technology for OpenVMS, enter a command similar to the following:

```
$ product install xml_j

The following product has been selected:
    VSI I64VMS XML_J V4.0                  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.

Configuring VSI I64VMS XML_J V4.0: XML Java Technology for
OpenVMS is based on Apache Xerces-Java Version 2.11.0 and
Apache Xalan-Java Version 2.7.1

    © Copyright 2015 VMS Software Inc.

    VSI Software Inc.

* This product does not have any configuration options.

Execution phase starting ...

The following product will be installed to destination:
    VSI I64VMS XML_J V4.0
DISK$I64SYS:[VMS$COMMON.]

Portion done: 0%...10%...20%...30%...40%...50%...60%...70%...80%...90%

XML-J$ROOT, XERCES-J$ROOT and XALAN-J$ROOT have been defined.
The following lines must be added to SYS$MANAGER:SYLOGICALS.COM so
that it will be defined each time the system is rebooted.

define/system/nolog/trans=concealed XML-J$ROOT
DISK$I64SYS:[SYS0.SYSCOMMON.xml.]
define/system/nolog/trans=concealed XERCES-J$ROOT
DISK$I64SYS:[SYS0.SYSCOMMON.xml.xerces-2_11_0.]
define/system/nolog/trans=concealed XALAN-J$ROOT
DISK$I64SYS:[SYS0.SYSCOMMON.xml.xalan-j_2_7_1.]

Verification of the installation can be performed using the XML-J
Test Procedure.  To run the XML-J Test Procedure, enter the
following command:

        $ @XML-J$ROOT:[xml-j-4_0]xml-j-4_0-tp

The file XML-J$ROOT:[xml-j-4_0]xml-j-4_0_setclasspath.com has been
provided to set up the Java class path for the XML-J Test Procedure.
It may also be useful to retain for your own XML applications.

...100%
```

```
The following product has been installed:
    VSI I64VMS XML_J V4.0                  Layered Product
$
```

After Installing the XML Java Technology for OpenVMS

After the installation is complete, optionally run the XML Java Test Procedure. To run the test procedure, enter the following command:

```
$ @XML-J$ROOT:[xml-j-4_0]xml-j-4_0-tp
```

Interpreting the results of the XML Java Test Procedure

The XML Test Procedure compares the output from the tests with a set of benchmarks. Because of the nature of the tests, some differences between the results and the benchmarks are to be expected. The following describes the common differences, which are expected to occur. These differences can be ignored because they do not indicate a potential problem with the XML installation:

Xerces dom.Counter

The output of this program shows the time and count of elements, attributes, ignorable whitespaces, and characters appearing in the document. Three times are shown: the parse time, the first traversal of the document, and the second traversal of the tree. The times are not likely to match the times in the benchmark; however, the element counts should match.

Note: The results produced by this program should never be accepted as true performance measurements.

Following is a session log containing a run of the XML Test Procedure followed by a section showing the differences between the run log and supplied benchmark:

```
$ @XML-J$ROOT:[xml-j-4_0]xml-j-4_0-tp
No test specified.  All applicable tests will be run.


Starting Xerces-J tests...
%DCL-S-SPAWNED, process BIGGLES_40688 spawned
%DCL-S-ATTACHED, terminal now attached to process BIGGLES_40688
%DCL-S-RETURNED, control returned to process BIGGLES

Starting Xalan-J tests...
%DCL-S-SPAWNED, process BIGGLES_53406 spawned
%DCL-S-ATTACHED, terminal now attached to process BIGGLES_53406
%DCL-S-RETURNED, control returned to process BIGGLES

Tests complete.  Check XML-J$ROOT:[xml-j-4_0]XML-J-TP.LOG for errors.

$ type XML-J$ROOT:[xml-j-4_0]XML-J-TP.LOG
************
File XML-J$ROOT:[xerces-2_11_0.samples.RESULTS]XML-XERCES-TP.OUT;1
   91   personal.xml: 112;24;0 ms (37 elems, 18 attrs, 140 spaces, 128
chars)
   92   $ java "dom.Counter" -x 5 personal.xml
   93   personal.xml: 135/5=27;24;1 ms (37 elems, 18 attrs, 140 spaces,
128 chars)
   94   $ java "dom.Counter" -n personal.xml
   95   personal.xml: 100;25;0 ms (37 elems, 18 attrs, 140 spaces, 128
chars)
   96   $ java "dom.Counter" -N personal.xml
```

```
   97   personal.xml: 100;24;0 ms (37 elems, 18 attrs, 140 spaces, 128
chars)
   98   $ java "dom.Counter" -s personal-schema.xml
   99   personal-schema.xml: 454;20;0 ms (37 elems, 20 attrs, 0 spaces,
268 chars)
  100   $ java "dom.Counter" -S personal-schema.xml
******
File XML-J$ROOT:[xerces-2_11_0.samples.benchmarks]xml-xerces-tp.bmk;1
   91   personal.xml: 100;24;0 ms (37 elems, 18 attrs, 140 spaces, 128
chars)
   92   $ java "dom.Counter" -x 5 personal.xml
   93   personal.xml: 135/5=27;25;0 ms (37 elems, 18 attrs, 140 spaces,
128 chars)
   94   $ java "dom.Counter" -n personal.xml
   95   personal.xml: 100;24;1 ms (37 elems, 18 attrs, 140 spaces, 128
chars)
   96   $ java "dom.Counter" -N personal.xml
   97   personal.xml: 100;25;0 ms (37 elems, 18 attrs, 140 spaces, 128
chars)
   98   $ java "dom.Counter" -s personal-schema.xml
   99   personal-schema.xml: 430;20;0 ms (37 elems, 20 attrs, 0 spaces,
268 chars)
  100   $ java "dom.Counter" -S personal-schema.xml
************
************
File XML-J$ROOT:[xerces-2_11_0.samples.RESULTS]XML-XERCES-TP.OUT;1
  105   personal.xml: 100;25;0 ms (37 elems, 18 attrs, 140 spaces, 128
chars)
  106   $ java "dom.Counter" -v personal.xml
  107   personal.xml: 109;24;0 ms (37 elems, 18 attrs, 140 spaces, 128
chars)
  108   $ java "dom.Counter" -V personal.xml
  109   personal.xml: 109;24;0 ms (37 elems, 18 attrs, 140 spaces, 128
chars)
  110   $ set nover
******
File XML-J$ROOT:[xerces-2_11_0.samples.benchmarks]xml-xerces-tp.bmk;1
  105   personal.xml: 100;24;1 ms (37 elems, 18 attrs, 140 spaces, 128
chars)
  106   $ java "dom.Counter" -v personal.xml
  107   personal.xml: 108;24;0 ms (37 elems, 18 attrs, 140 spaces, 128
chars)
  108   $ java "dom.Counter" -V personal.xml
  109   personal.xml: 108;24;1 ms (37 elems, 18 attrs, 140 spaces, 128
chars)
  110   $ set nover
************

Number of difference sections found: 2
Number of difference records found: 14

DIFFERENCES
/IGNORE=(SPACING,TRAILING_SPACES,FORM_FEEDS,CASE,BLANK_LINES)/MERGED=1-
    XML-J$ROOT:[xerces-2_11_0.samples.RESULTS]XML-XERCES-TP.OUT;1-
    XML-J$ROOT:[xerces-2_11_0.samples.benchmarks]xml-xerces-tp.bmk;1
Number of difference sections found: 0
Number of difference records found: 0

DIFFERENCES
/IGNORE=(SPACING,TRAILING_SPACES,FORM_FEEDS,CASE,BLANK_LINES)/MERGED=1-
    XML-J$ROOT:[xalan-j_2_7_1.samples.RESULTS]XML-XALAN-TP.OUT;1-
    XML-J$ROOT:[xalan-j_2_7_1.samples.benchmarks]xml-xalan-tp.bmk;1
```

Installing the sources

The sources are provided in the form a backup saveset. The “sources” restore operation requires approximately 70,000 blocks of disk space. Execute the following command to restore the sources.

```
      $ set def XML-J$ROOT:[000000]
      $ @xmlj_restore_backups.com

*** THIS PROCEDURE WILL LET YOU RESTORE SOURCES ON TO XML-
J$ROOT:[000000...] ***

               1. RESTORE SOURCES
               2. EXIT

TYPE 1 OR 2 : 1

BACKUP RESTORE OPERATION STARTS...

SOURCE IS RESTORED
```

Removing the Kit

To remove the XML Java Technology for OpenVMS, enter the following command:

```
$ PRODUCT REMOVE XML_J
```

The sample output of a product remove operation is shown below.

```
The following product has been selected:
    VSI I64VMS XML_J V4.0                 Layered Product

Do you want to continue? [YES]

The following product will be removed from destination:
    VSI I64VMS XML_J V4.0                 DISK$I64SYS:[VMS$COMMON.]

Portion done:
0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%

The following product has been removed:
    VSI I64VMS XML_J V4.0                 Layered Product
$
```

Release Notes

There are no release notes for the current release of XML Java Technology for OpenVMS.