

VSI OpenVMS

VSI x86-64 Cross-tools Kit Installation and Startup Guide

This document describes how to install and setup your VSI x86-64 Cross-tools Kit.

Operating System and Version: VSI OpenVMS x86-64 V9.2

VSI x86-64 Cross-tools Kit Installation and Startup Guide



VMS Software

Copyright © 2022 VMS Software, Inc. (VSI), Boston, Massachusetts, USA

Legal Notice

Confidential computer software. Valid license from VSI 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.

The information contained herein is subject to change without notice. The only warranties for VSI products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. VSI shall not be liable for technical or editorial errors or omissions contained herein.

HPE and HPE Integrity are trademarks or registered trademarks of Hewlett Packard Enterprise.

Intel, Itanium and IA-64 are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Kerberos is a trademark of the Massachusetts Institute of Technology.

Preface	v
1. Intended Audience	v
2. Prerequisites	v
VSI x86-64 Cross-tools Kit Installation and Startup	1
1. Verifying MD5 Checksum	2
2. Unzipping the Kit	2
3. Installing the VSI x86-64 Cross-tools Kit	2
4. Upgrading to the Cross-tools Kit V9.2-XG9N	6
5. Setting Up and Starting the VSI x86-64 Cross-tools Kit	7
6. Showing the Process Logicals	8
7. Running the IVP	9

Preface

1. Intended Audience

This document is intended for all users of VSI OpenVMS x86-64 V9.2.

2. Prerequisites

This kit must be installed on an Itanium system running some version of VSI OpenVMS.

VSI x86-64 Cross-tools Kit Installation and Startup

This section provides instructions for how to unzip, install, and start up the VSI x86-64 cross-tools kit.

1. Verifying MD5 Checksum

VSI recommends that you verify the MD5 checksum of the cross-tools ZIP file after it has been downloaded from the VSI Services Portal to your target Itanium system. The MD5 checksum of the cross-tools **VSI-I64VMS-X86_XTOOLS-V0902-XG9N-1.ZIP** file must correspond to the following value:

```
840673CD4F5D4978005EAF4D4A189903
```

2. Unzipping the Kit

Once you have located and downloaded the ZIP file, enter the following command:

```
$ unzip VSI-I64VMS-X86_XTOOLS-V0902-XG9N-1.ZIP
```

The archive will be unzipped, as shown in the following example:

```
Archive:  STAGED$:[X86_XTOOLS]VSI-I64VMS-X86_XTOOLS-V0902-XG9N-1.ZIP;1
```

```
-----
This ZIP kit contains a PCSI kit with cross-tools for
the VMS Software, Inc. OpenVMS X86-64 platform.
```

```
The following cross-tools are included in this kit:
```

Bliss-32	T1.12-129-50W5U	31-MAY-2022
Bliss-64	T1.12-129-50W5U	31-MAY-2022
XCC\$COMPILER	X7.4-547-50V6F	1-OCT-2021
XMacro	X6.0-111	19-APR-2022
Message	I01-10 (XG7B-T7Y-000000)	4-APR-2022
CDU	I01-12 (XG7B-T7Y-000000)	4-APR-2022
ANALYZOBJ	I01-85 (XG7B-T7Y-000000)	4-APR-2022
LLVM-MC	V1.0	5-FEB-2020
LINKer	I02-89 (XG7B-T7Y-000000)	4-APR-2022
CrfShr	I1.106 (XG7B-T7Y-000000)	4-APR-2022
Librarian	I01-47 (XG7B-T7Y-000000)	4-APR-2022
LbrShr	I01-45 (XG7B-T7Y-000000)	4-APR-2022
F90 (Fortran)	F90 X8.4-104966	31-MAY-2021
Cobol	COBOL x86-64 X3.1-0013 (50V8U)	1-SEP-2021
Pascal	X6.3-136-50VCS	5-MAR-2022

```
LIB and STARLET are supplied from the XG9N result disk
which was built on Thursday, 2-JUN-2022.
```

```
The unzipped size of this kit is 1273627 blocks.
```

```
-----
inflating: VSI-I64VMS-X86_XTOOLS-V0902-XG9N-1.PCSI$COMPRESSED
extracting: VSI-I64VMS-X86_XTOOLS-V0902-XG9N-1.PCSI$COMPRESSED_VNC
$
```

3. Installing the VSI x86-64 Cross-tools Kit

This section provides instructions for how to install the VSI x86-64 Cross-tools Kit using PCSI on an IA64 server.

Note

The time needed to perform an installation will vary. VSI has observed that installations performed on an rx2660 that is booted from a disk that is a locally attached SCSI drive takes about 10 minutes. When performed on that same rx2660 that was booted from a Fibre Channel disk, installation time is substantially shorter taking a little less than 4 minutes.

Enter the following command on your IA64 system:

```
$ PRODUCT INSTALL X86_XTOOLS
```

The installation of the cross-tools kit begins, as shown in this example:

```
Performing product kit validation of signed kits ...
%PCSI-I-VSIVALPASSED, validation of $1$DGA452:[X86_XTOOLS]VSI-I64VMS-
X86_XTOOLS-V0902-XG9N-1.PCSI$COMPRESSED;1 succeeded
```

The following product has been selected:

```
VSI I64VMS X86_XTOOLS V9.2-XG9N          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 X86_XTOOLS V9.2-XG9N: X86 cross-architecture toolkit for IA64 VMS (base level XG9N)

Copyright 2022 VMS Software, Inc.

VMS Software, Inc.

Some components of this kit require a Product Authorization Key (PAK)

* This product does not have any configuration options.

The following cross-tools will be installed:

Bliss-32	T1.12-129-50W5U	31-MAY-2022
Bliss-64	T1.12-129-50W5U	31-MAY-2022
XCC\$COMPILER	X7.4-547-50V6F	1-OCT-2021
XMacro	X6.0-111	19-APR-2022
Message	I01-10 (XG7B-T7Y-000000)	4-APR-2022
CDU	I01-12 (XG7B-T7Y-000000)	4-APR-2022
ANALYZOBJ	I01-85 (XG7B-T7Y-000000)	4-APR-2022
LLVM-MC	V1.0	5-FEB-2020
LINKer	I02-89 (XG7B-T7Y-000000)	4-APR-2022
CrfShr	I1.106 (XG7B-T7Y-000000)	4-APR-2022
Librarian	I01-47 (XG7B-T7Y-000000)	4-APR-2022
LbrShr	I01-45 (XG7B-T7Y-000000)	4-APR-2022
F90 (Fortran)	F90 X8.4-104966	31-MAY-2021
Cobol	COBOL x86-64 X3.1-0013 (50V8U)	1-SEP-2021
Pascal	X6.3-136-50VCS	5-MAR-2022

LIB and STARLET are supplied from the XG9N result disk which was built on Thursday, 2-JUN-2022.

Components requiring Product Authorization Keys (PAKs):

The F90 (Fortran-90) and COBOL components of this toolkit each require a Product Authorization Key (PAK).

Licenses are not required for any other components of this toolkit.

Execution phase starting ...

The following product will be installed to destination:

VSI I64VMS X86_XTOOLS V9.2-XG9N DISK\$FRED_V842L1:[VMS\$COMMON.]

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

Extracting modules from the PASCAL STARLET text library and creating .PEN files...

Executing X86_XTOOLS\$STARTUP.COM

Executing X86_XTOOLS\$SYLOGIN.COM

Source text library: X86_XTOOLS\$ROOT:[SYSLIB]STARLETPAS.TLB;1

...using X86_XTOOLS\$ROOT:[SYSEXEC]PASCAL.EXE X6.3-136 (LINK'd 5-MAR-2022 22:54:24.89)

Generating STARLET.PAS/PEN...
Generating PASCAL\$ACLEDIT_ROUTINES.PAS/PEN...
Generating PASCAL\$CLI_ROUTINES.PAS/PEN...
Generating PASCAL\$CMA_ROUTINES.PAS/PEN...
Generating PASCAL\$CONV_ROUTINES.PAS/PEN...
Generating PASCAL\$CVT_ROUTINES.PAS/PEN...
Generating PASCAL\$DCX_ROUTINES.PAS/PEN...
Generating PASCAL\$DTK_ROUTINES.PAS/PEN...
Generating PASCAL\$EDT_ROUTINES.PAS/PEN...
Generating PASCAL\$FDL_ROUTINES.PAS/PEN...
Generating PASCAL\$LBR_ROUTINES.PAS/PEN...
Generating PASCAL\$LIB_ROUTINES.PAS/PEN...
Generating PASCAL\$LIB_ROUTINES_IEEE_FLOAT.PAS/PEN...
Generating PASCAL\$MAIL_ROUTINES.PAS/PEN...
Generating PASCAL\$MTH_ROUTINES.PAS/PEN...
Generating PASCAL\$NCS_ROUTINES.PAS/PEN...
Generating PASCAL\$OTS_ROUTINES.PAS/PEN...
Generating PASCAL\$PPL_ROUTINES.PAS/PEN...
Generating PASCAL\$PSM_ROUTINES.PAS/PEN...
Generating PASCAL\$SMB_ROUTINES.PAS/PEN...
Generating PASCAL\$SMG_ROUTINES.PAS/PEN...
Generating PASCAL\$SOR_ROUTINES.PAS/PEN...
Generating PASCAL\$STR_ROUTINES.PAS/PEN...
Generating PASCAL\$TBK_ROUTINES.PAS/PEN...
Generating PASCAL\$TPU_ROUTINES.PAS/PEN...

...100%

The following product has been installed:

VSI I64VMS X86_XTOOLS V9.2-XG9N Layered Product

%PCSI-I-IVPEXECUTE, executing test procedure for VSI I64VMS X86_XTOOLS V9.2-XG9N ...

Executing X86_XTOOLS\$SYLOGIN.COM

Maps, listings, sources, etc. can be found in X86\$IVP: (X86_XTOOLS \$ROOT:[SYS\$IVP])

Compiling DEC C, Bliss-32, Bliss-64, XMACRO & CDU objects...
(& running ANALYZE/OBJECT after compilation)

.
.

.

Linking images (compilation warnings are possible but not expected)...

```
$ !
$ link x86_xtools$ivp.olb/include=x86_xtools$ivp_decc /
executable=x86_xtools$ivp_decc /map=x86_xtools$ivp_decc
$ link x86_xtools$ivp.olb/include=x86_xtools$ivp_bliss32 /
executable=x86_xtools$ivp_bliss-32 /map=x86_xtools$ivp_bliss-32
$ link x86_xtools$ivp.olb/include=x86_xtools$ivp_bliss64 /
executable=x86_xtools$ivp_bliss-64 /map=x86_xtools$ivp_bliss-64
$ link x86_xtools$ivp.olb/include=x86_xtools$ivp_xmacro /
executable=x86_xtools$ivp_xmacro /map=x86_xtools$ivp_xmacro
$ link x86_xtools$ivp.olb/include=x86_xtools$ivp_f90 /
executable=x86_xtools$ivp_f90 /map=x86_xtools$ivp_f90
$ link x86_xtools$ivp.olb/include=x86_xtools$ivp_cobol /
executable=x86_xtools$ivp_cobol /map=x86_xtools$ivp_cobol
$ link x86_xtools$ivp.olb/include=x86_xtools$ivp_pascal /
executable=x86_xtools$ivp_pascal /map=x86_xtools$ivp_pascal
$ !
```

%PCSI-I-IVPSUCCESS, test procedure completed successfully

VSI I64VMS X86_XTOOLS V9.2-XG9N: X86 cross-architecture toolkit for IA64 VMS (base level XG9N)

The following startup, login and verification files are supplied by this kit:

```
SYS$STARTUP:X86_XTOOLS$STARTUP.COM
SYS$MANAGER:X86_XTOOLS$SYLOGIN.COM
SYS$TEST:X86_XTOOLS$IVP.COM
```

Release notes from this kit are available in SYS\$HELP

The PCSI kit release notes are copied to the system disk during kit

installation and are available as a standard text file which may be read using the TYPE command or from an editor:

```
SYSS$HELP:X86_XTOOLS-V0902-XG9N.RELEASE_NOTES
```

```
$
```

The V9.2-XG9N cross-tools kit release notes are located in SYSS\$SYSROOT:SYSS\$HELP:X86_XTOOLS-V0902-XG9N.RELEASE_NOTES.

4. Upgrading to the Cross-tools Kit V9.2-XG9N

If you have an earlier version of the cross-tools kit installed on your system, you can easily upgrade to the V9.2-XG9N cross-tools kit by performing the standard installation procedure described in the Section 3 section. The installation procedure will remove the earlier version of the cross-tools kit and install the V9.2-XG9N kit, as shown in this example:

```

.
.
.
Execution phase starting ...

The following product will be installed to destination:
  VSI I64VMS X86_XTOOLS V9.2-XG9N      DISK$FRED_V842L1:[VMS$COMMON.]
The following product will be removed from destination:
  VSI I64VMS X86_XTOOLS E9.2-XG6F      DISK$FRED_V842L1:[VMS$COMMON.]

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

  Extracting modules from the PASCAL STARLET text library
  and creating .PEN files...

  Executing X86_XTOOLS$SYLOGIN.COM

  Source text library: X86_XTOOLS$ROOT:[SYSLIB]STARLETPAS.TLB;1

  ...using X86_XTOOLS$ROOT:[SYSEXE]PASCAL.EXE X6.3-136 (LINK'd 5-
MAR-2022 22:54:24.89)

  Generating STARLET.PAS/PEN...
  Generating PASCAL$ACLEDIT_ROUTINES.PAS/PEN...
  Generating PASCAL$CLI_ROUTINES.PAS/PEN...
  Generating PASCAL$CMA_ROUTINES.PAS/PEN...
  Generating PASCAL$CONV_ROUTINES.PAS/PEN...
  Generating PASCAL$CVT_ROUTINES.PAS/PEN...
  Generating PASCAL$DCX_ROUTINES.PAS/PEN...
  Generating PASCAL$DTK_ROUTINES.PAS/PEN...
  Generating PASCAL$EDT_ROUTINES.PAS/PEN...
  Generating PASCAL$FDL_ROUTINES.PAS/PEN...
  Generating PASCAL$LBR_ROUTINES.PAS/PEN...
  Generating PASCAL$LIB_ROUTINES.PAS/PEN...
  Generating PASCAL$LIB_ROUTINES_IEEE_FLOAT.PAS/PEN...
  Generating PASCAL$MAIL_ROUTINES.PAS/PEN...
  Generating PASCAL$MTH_ROUTINES.PAS/PEN...
  Generating PASCAL$NCS_ROUTINES.PAS/PEN...
  Generating PASCAL$OTS_ROUTINES.PAS/PEN...

```

```

Generating PASCAL$PPL_ROUTINES.PAS/PEN...
Generating PASCAL$PSM_ROUTINES.PAS/PEN...
Generating PASCAL$SMB_ROUTINES.PAS/PEN...
Generating PASCAL$SMG_ROUTINES.PAS/PEN...
Generating PASCAL$SOR_ROUTINES.PAS/PEN...
Generating PASCAL$STR_ROUTINES.PAS/PEN...
Generating PASCAL$TBK_ROUTINES.PAS/PEN...
Generating PASCAL$TPU_ROUTINES.PAS/PEN...

```

...100%

The following product has been installed:

```
VSI I64VMS X86_XTOOLS V9.2-XG9N      Layered Product
```

The following product has been removed:

```
VSI I64VMS X86_XTOOLS E9.2-XG6F      Layered Product
```

```
%PCSI-I-IVPEXECUTE, executing test procedure for VSI I64VMS X86_XTOOLS
V9.2-XG9N ...
```

```

.
.
.

```

5. Setting Up and Starting the VSI x86-64 Cross-tools Kit

This section provides instructions for starting and setting up your cross-tools environment. The following list describes the three login and verification files that are needed.

- `SY$STARTUP:X86_XTOOLS$STARTUP.COM`: This command procedure defines `/SYSTEM` logical names and needs to be run by a privileged user once. Optionally, it can be added to the system startup file (invoked within `SY$STARTUP_VMS.COM`). It needs to be run at least once to define the `X86$` logical names, which are listed later in this section.
- `SY$MANAGER:X86_XTOOLS$SYLOGIN.COM`: This command procedure performs the following functions:
 1. Defines `/PROCESS` logical names
 2. Adds the cross-tools specific command definitions to the CLI table of a process
 3. Sets up DCL global symbols (which shows how someone might define their own DCL symbols to run any of the tools)

The file can be run by anyone who wants to use the tools. Each process is required to have these items set up in order to use the cross-tools. It assumes that the `/SYSTEM` logicals are defined. The command procedure can be added to the system-wide sylogin file (invoked within `SYLOGIN.COM`), or added by any user to a personal login file.

- `SY$TEST:X86_XTOOLS$IVP.COM`: This command procedure can be run at any time by a privileged user. Its primary purpose is to be run as part of the installation to verify that the installation completed as expected. This command file runs automatically and unconditionally **when the cross-tools kit is installed**, although it can be run again at any time after the installation.

Enter the following command to list the system logicals defined by the startup file:

```
$ SHOW LOGICAL X86* /SYSTEM
```

```
(LNM$SYSTEM_TABLE)
```

```
"X86$ETC" = "X86_XTOOLS$ROOT:[SYS$ETC]"
"X86$HELP" = "X86_XTOOLS$ROOT:[SYSHLP]"
"X86$IVP" = "X86_XTOOLS$ROOT:[SYS$IVP]"
"X86$LIBRARY" = "X86_XTOOLS$ROOT:[SYSLIB]"
"X86$LOADABLE_IMAGES" = "X86_XTOOLS$ROOT:[SYS$LDR]"
"X86$MESSAGE" = "X86_XTOOLS$ROOT:[SYSMSG]"
"X86$SYSTEM" = "X86_XTOOLS$ROOT:[SYSEXE]"
"X86$UPDATE" = "X86_XTOOLS$ROOT:[SYSUPD]"
"X86_XTOOLS$ROOT" = "FRED$DKA400:[SYS0.SYSCOMMON.X86_XTOOLS$ROOT.]"
```

```
(LNM$SYSCLUSTER_TABLE)
```

```
$
```

6. Showing the Process Logicals

Use the following command to show the process logicals and DCL global symbols by setting P1 in the command line to 1:

```
$ @SYS$MANAGER:X86_XTOOLS$SYLOGIN 1
```

```
%X86_XTOOLS$SYLOGIN-I-XXTSETBEG, X86 VMS XTools setup starting
```

```
$ Define /NoLog XCC$COMPILER X86_XTOOLS$ROOT:[SYSEXE]XCC$COMPILER.EXE
$ Set Command X86_XTOOLS$ROOT:[SYSUPD]XCC.CLD
$ CC == "XCC"
```

```
$ Define /NoLog BLISS32X X86_XTOOLS$ROOT:[SYSEXE]BLISS32X.EXE
$ Set Command X86_XTOOLS$ROOT:[SYSUPD]BLISS_IN.CLD
$ BLISS == "Bliss /X32"
```

```
$ Define /NoLog BLISS64X X86_XTOOLS$ROOT:[SYSEXE]BLISS64X.EXE
$ BLISS64 == "Bliss /X64"
```

```
$ Define /NoLog MACRO X86_XTOOLS$ROOT:[SYSEXE]XMACRO.EXE
$ Set Command X86_XTOOLS$ROOT:[SYSUPD]X86_MACRO.CLD
```

```
$ Define /NoLog ANALYZOBJ X86_XTOOLS$ROOT:[SYSEXE]X86_ANALYZOBJ.EXE
$ Set Command X86_XTOOLS$ROOT:[SYSUPD]ANALYZE.CLD
$ Set Message X86_XTOOLS$ROOT:[SYSMSG]X86_FILMNTMSG.EXE
```

```
$ Define /NoLog IA64_LINK X86_XTOOLS$ROOT:[SYSEXE]X86_LINK.EXE
$ Define /NoLog IBUILD_CRFSHR X86_XTOOLS$ROOT:[SYSLIB]X86_CRFSHR.EXE
$ Set Command X86_XTOOLS$ROOT:[SYSUPD]X86_LINK.CLD
```

```
$ Define /NoLog LIBRARIAN X86_XTOOLS$ROOT:[SYSEXE]X86_LIBRARIAN.EXE
$ Define /NoLog IBUILD_LBRSHR X86_XTOOLS$ROOT:[SYSLIB]X86_LBRSHR.EXE
$ Set Command X86_XTOOLS$ROOT:[SYSUPD]LIBRARIAN.CLD
```

```
$ Define /NoLog MESSAGE X86_XTOOLS$ROOT:[SYSEXE]X86_MESSAGE.EXE
$ Set Command X86_XTOOLS$ROOT:[SYSUPD]X86_MESSAGE.CLD
```

```
$ Define /NoLog LLVM_MC X86_XTOOLS$ROOT:[SYSEXE]LLVM-MC.EXE
$ LLVM_MC == "$LLVM_MC"
```

```

$ Define /NoLog F90$MAIN X86_XTOOLS$ROOT:[SYSEXE]F90$MAIN.EXE
$ Define /NoLog F90$MESSG X86_XTOOLS$ROOT:[SYSMSG]F90$MESSG.EXE
$ Set Command X86_XTOOLS$ROOT:[SYSUPD]F90.CLD

$ Define /NoLog PASCAL X86_XTOOLS$ROOT:[SYSEXE]PASCAL.EXE
$ Define /NoLog PASCALER1 X86_XTOOLS$ROOT:[SYSMSG]PASCALER1.EXE
$ Define /NoLog PASCALER2 X86_XTOOLS$ROOT:[SYSMSG]PASCALER2.EXE
$ Set Command X86_XTOOLS$ROOT:[SYSUPD]PASCAL.CLD

$ Define /NoLog COBOL X86_XTOOLS$ROOT:[SYSEXE]COBOL.EXE
$ Define /NoLog COBOL$MESSG X86_XTOOLS$ROOT:[SYSMSG]COBOL$MESSG.EXE
$ Set Command X86_XTOOLS$ROOT:[SYSUPD]COBOL_CLD.CLD

$ Define /NoLog CDU X86_XTOOLS$ROOT:[SYSEXE]X86_CDU.EXE
$ Set Command X86_XTOOLS$ROOT:[SYSUPD]X86_SET.CLD
$ CLD == "Set Command"

$
%X86_XTOOLS$SYLOGIN-I-XXTSETDON, X86 VMS XTools setup done

```

7. Running the IVP

Enter the following command to display the results of the IVP, as shown in the example that follows:

```
$ DIRECTORY X86$IVP.; /SIZE /DATE /WIDTH=FILE=40
```

```

Directory X86_XTOOLS$ROOT:[SYS$IVP]

X86_XTOOLS$IVP.OLB;2          278  17-JUN-2022  15:40:23.14
X86_XTOOLS$IVP_BLISS-32.B32;2      1  17-JUN-2022  15:40:23.32
X86_XTOOLS$IVP_BLISS-32.EXE;2     19  17-JUN-2022  15:40:28.38
X86_XTOOLS$IVP_BLISS-32.LIS;2      8  17-JUN-2022  15:40:24.49
X86_XTOOLS$IVP_BLISS-32.MAP;2     19  17-JUN-2022  15:40:28.34
X86_XTOOLS$IVP_BLISS-32.OBJ;2      8  17-JUN-2022  15:40:24.65
X86_XTOOLS$IVP_BLISS-32.OBJ$ANALYZE;2  5  17-JUN-2022  15:40:24.86
X86_XTOOLS$IVP_BLISS-64.B64;2      1  17-JUN-2022  15:40:23.38
X86_XTOOLS$IVP_BLISS-64.EXE;2     19  17-JUN-2022  15:40:28.61
X86_XTOOLS$IVP_BLISS-64.LIS;2      8  17-JUN-2022  15:40:25.04
X86_XTOOLS$IVP_BLISS-64.MAP;2     19  17-JUN-2022  15:40:28.57
X86_XTOOLS$IVP_BLISS-64.OBJ;2      8  17-JUN-2022  15:40:25.19
X86_XTOOLS$IVP_BLISS-64.OBJ$ANALYZE;2  5  17-JUN-2022  15:40:25.38
X86_XTOOLS$IVP_CDU.CLD;2          1  17-JUN-2022  15:40:23.51
X86_XTOOLS$IVP_CDU.LIS;2         22  17-JUN-2022  15:40:26.01
X86_XTOOLS$IVP_CDU.OBJ;2          2  17-JUN-2022  15:40:26.05
X86_XTOOLS$IVP_CDU.OBJ$ANALYZE;2    21  17-JUN-2022  15:40:26.14
X86_XTOOLS$IVP_COBOL.COB;2        1  17-JUN-2022  15:40:23.69
X86_XTOOLS$IVP_COBOL.EXE;2       21  17-JUN-2022  15:40:29.30
X86_XTOOLS$IVP_COBOL.LIS;2       16  17-JUN-2022  15:40:26.96
X86_XTOOLS$IVP_COBOL.MAP;2       21  17-JUN-2022  15:40:29.26
X86_XTOOLS$IVP_COBOL.OBJ;2       11  17-JUN-2022  15:40:27.12
X86_XTOOLS$IVP_COBOL.OBJ$ANALYZE;2  8  17-JUN-2022  15:40:27.35
X86_XTOOLS$IVP_DECC.C;2          1  17-JUN-2022  15:40:23.26
X86_XTOOLS$IVP_DECC.EXE;2       20  17-JUN-2022  15:40:28.13
X86_XTOOLS$IVP_DECC.LIS;2       16  17-JUN-2022  15:40:23.79
X86_XTOOLS$IVP_DECC.MAP;2       19  17-JUN-2022  15:40:28.07
X86_XTOOLS$IVP_DECC.OBJ;2       10  17-JUN-2022  15:40:24.05

```

X86_XTOOLS\$IVP_DECC.OBJ\$ANALYZE;2	13	17-JUN-2022	15:40:24.30
X86_XTOOLS\$IVP_F90.EXE;2	20	17-JUN-2022	15:40:29.07
X86_XTOOLS\$IVP_F90.F90;2	1	17-JUN-2022	15:40:23.57
X86_XTOOLS\$IVP_F90.LIS;2	15	17-JUN-2022	15:40:26.34
X86_XTOOLS\$IVP_F90.MAP;2	19	17-JUN-2022	15:40:29.03
X86_XTOOLS\$IVP_F90.OBJ;2	9	17-JUN-2022	15:40:26.51
X86_XTOOLS\$IVP_F90.OBJ\$ANALYZE;2	9	17-JUN-2022	15:40:26.75
X86_XTOOLS\$IVP_PASCAL.EXE;2	19	17-JUN-2022	15:40:29.53
X86_XTOOLS\$IVP_PASCAL.LIS;2	17	17-JUN-2022	15:40:27.52
X86_XTOOLS\$IVP_PASCAL.MAP;2	18	17-JUN-2022	15:40:29.49
X86_XTOOLS\$IVP_PASCAL.OBJ;2	9	17-JUN-2022	15:40:27.67
X86_XTOOLS\$IVP_PASCAL.OBJ\$ANALYZE;2	6	17-JUN-2022	15:40:27.87
X86_XTOOLS\$IVP_PASCAL.PAS;2	1	17-JUN-2022	15:40:23.63
X86_XTOOLS\$IVP_XMACRO.EXE;2	12	17-JUN-2022	15:40:28.85
X86_XTOOLS\$IVP_XMACRO.LIS;2	17	17-JUN-2022	15:40:25.62
X86_XTOOLS\$IVP_XMACRO.MAP;2	19	17-JUN-2022	15:40:28.81
X86_XTOOLS\$IVP_XMACRO.MAR;2	1	17-JUN-2022	15:40:23.44
X86_XTOOLS\$IVP_XMACRO.OBJ;2	5	17-JUN-2022	15:40:25.70
X86_XTOOLS\$IVP_XMACRO.OBJ\$ANALYZE;2	8	17-JUN-2022	15:40:25.84

Total of 47 files, 806 blocks.

\$