SDx Development Environment

Release Notes, Installation, and Licensing Guide

UG1238 (v2017.1) June 20, 2017

XILINX
ALL PROGRAMMABLE™

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
06/20/17	2017.1	Revised to reflect changes for 2017.1 SDx software.

Table of Contents

XILINX
ALL PROGRAMMABLE™

Chapter 1

Release Notes and Supported Hardware

SDSoC - SDAccel Development Environment Common Infrastructure

The common infrastructure includes:

- Integrated installer to download and install SDSoC™ and SDAccel™ Development environments.
  - Complete installation environment containing the required tools and data files for supported devices and platforms.
  - Web-based installer support.
  - Option to install SDSoC or SDAccel environments.
- Supported Operating Systems.
  - Windows 7 and 7 SP1 Professional (64-bit) (SDSoC only).
  - Windows 10 Professional (64-bit) (SDSoC only).
  - Linux Support.
    - Red Hat Enterprise Workstation/Server 7.2 and 7.3 (64-bit).
    - Red Hat Enterprise Workstation 6.7 and 6.8 (64-bit).
    - CentOS 6.8, CentOS 7.3 (64-bit) (SDAccel only).
    - Ubuntu Linux 16.04.1 LTS (64-bit).
- Integrated GUI supporting both SDSoC and SDAccel Development environments.
  - Eclipse based IDE with support for project creation, emulation, performance estimation, implementation and debug.
  - One-stop Reports View to access all reports.
- Stability improvements of SDx GUI around project creation, emulation, and license checking.

What's New in 2017.1

SDSoC Development Environment Features

The SDSoC™ development environment updates included in this release:

- ARM® compiler tool chain support
  - Linaro-based gcc 6.2-2016.11 32-bit and 64-bit tool chains
- Target OS support
  - Linux (kernel 4.9, Xilinx branch xilinx-v2017.1_sdsoc), bare-metal and FreeRTOS 8.2.3
  - Example PetaLinux BSP for ZC702 platform with documentation in the *SDSoC Environment Platform Development Guide* (UG1164)
- Device support
  - Zynq®-7000 support
  - Enhanced Zynq UltraScale+™ MPSoC support, including:
    - Support for 128-bit HP and HPC ports.
    - QEMU / RTL co-simulation based.
    - zcu102_es1_ocl and zcu102_es2_ocl platforms supporting OpenCL™ and C/C++ applications on ES1 and ES2 silicon.
    - Support PMU firmware updates for power management.
    - ZCU102_axis_io (direct I/O) example platform
    - Supports Zynq UltraScale™ MPSoC reVISION™ and Base Targeted Reference Design 2017.1 (embedded video processing platform).
- Emulator based on QEMU and RTL co-simulation. Supports Zynq UltraScale+ MPSoC and Zynq-7000 SoC platforms.
  - Linux and Windows 64-bit host support (Beta release for Windows).
  - Support for all ZCU102 platforms (C/C++ applications only), ZC702, ZC706, Zybo, ZedBoard, MicroZed platforms.
  - Command line interface with graphical waveform viewer.
  - Integrated flows within the Eclipse IDE.
- OpenCL™ compilation flows supported for Zynq UltraScale+ MPSoC ZCU102_es1_ocl and ZCU102_es2_ocl platforms.
  - Build, software profiling, and debugging flows only (no emulation, performance estimation, or event tracing in this release).
  - Linux host OS support only.
  - Performance analysis is not supported.

- SDx Eclipse UI
  - Platform setup and project creation.
    - Add custom platform(s) without creating an SDx project.
    - SDx example store: access to GitHub SDSoC examples.
  - Acceleration
    - Support for accelerating C function templates.
    - Launch HLS for specified hardware functions.
  - Reporting
    - New post-implementation utilization reports.
    - Collapsible headers on all SDx reports.
- System compiler enhancements.
  - Support for xFAST reVISION libraries.
  - Support for multiple accelerator AXI clocks for zero_copy datamovers (i.e., all data movers for overclocked hardware functions).
  - Support of packed `structs` and `scalar` widths up to 1024-bits (was 32-bits).
  - Support for class members in SDS copy pragmas.
  - Improved scheduling for standalone and FreeRTOS.
  - Support for SG-DMA on MIG accessible DDR.
  - Bug fixes.
- C/C++ runtime enhancements
  - sds_alloc support for up to 4K buffers
  - Support for up to 64 UIO devices
  - Performance improvements
  - Bug fixes
- Platform updates
  - zcu102_es1_ocl, zcu102_es2_ocl platforms supporting Zynq UltraScale+ and OpenCL applications.
- Enhanced user-defined platform support
  - Enhanced sdspfm utility to assist in SDSoC platform creation
  - Enhanced error handling in the sdspfm utility
- New and updated sample applications
  - Source code examples for basic hardware optimizations, including loop unrolling, pipelining, array partitioning.
  - Accessible from github.com repository
  - xFAST libraries for reVISION applications
- Bug fixes and infrastructure updates
  - Improved error checking and handling
  - Updated and enhanced accelerator driver API software

SDAccel Development Environment Features

- RTL Kernel Support

  2017.1 provides significant improvement to usability and performance for importing and optimizing RTL kernels in SDx™.

  - New RTL Kernel wizard is added to provide user a template for importing RTL IP into SDx.
  - Provides scripts to build the RTL kernel into `.xo` (Xilinx Object) avoiding error-prone creation of `kernel.xml` files that were required in earlier releases.
  - For Vivado® experts working on optimizing performance of RTL kernels, `xocc` has been enhanced to include Vivado script through `-custom_script` during `xocc --link`.
  - Utilization reports are automatically generated during compilation. These reports show utilization of LUTS, Registers, BRAMs, and DSPs for the platform region, as well as for each of the kernels.
- The SDAccel™ compiler employs mathematical techniques to identify statically the access pattern of inputs and outputs to perform better memory coalescing and burst inferencing. This is an early access feature and can be enabled with both of the following `xocc` options:
  - `--xp param:compiler.version=39`
  - `--xp param:compiler.advancedLoopOptimizations=true`
- SDx/HLS should be able to infer a shift register pattern automatically for OpenCL™ kernel. OpenCL Shifter Design Pattern is inferred if:
  - The shift logic is described simply using for-loop.
  - The array is initiated with 0, with assignment to the begin or to the end for new value.
  - The access points should be constant offset.
- The compiler automatically decides if a shift-register is implemented in SRL or BRAM to improve better resource use and improve timing closure.
- Dataflow support for OpenCL is now a production feature.
- The compiler can support dataflow on functions with arbitrary sized parameters, sub-functions, or loops. Dataflow can also apply to loop statements.
- The `xocc` command to define OpenCL Compiler dataflow FIFO size is:

  ```
  --xp param:compiler.xclDataflowFifoDepth = 4
  ```
- The following warning message might appear during compile:

  ```
  kernel.cl:28:17: warning: unknown attribute 'xcl_dataflow'
  ignored __attribute__ ((xcl_dataflow))
  ```

  Ignore the warning or use `-k kernel_name` to avoid the warning.
- SDx introduces the OpenCL 2.0 image data type, which provides the ability to read and write to images in kernels through OpenCL 2.0 image built-ins.

- The supported APIs are:
  - `clCreateImage()` for the image types listed above.
  - `clGetSupportedImageFormats()`
  - `clEnqueueReadImage()`
  - `clEnqueueWriteImage()`
  - `clGetImageInfo()`
- Enhancements to OpenCL math built-in functions to improve performance and reduce resources.
- To provide better control to expert users, xocc allows users to write out the default script as well as apply custom scripts.

```
xocc -c -export_script
xocc -c -custom_script
```

- Changes to SDAccel platform include the following:
  - XDMA enhanced to support two Physical Functions to provide one PF for secure management.
  - AXI Firewall at XDMA Full AXI4 and two AXI Lite interfaces to insulate the platform from hangs caused by AXI protocol violations in kernels.
  - Feature ROM to embed platform data in a ROM in the DSA to enable Run Time checks.
  - Bitstream download through ICAP on all platforms rather than MCAP for faster downloads.
- Introducing support for Kintex® UltraScale™ FPGA KCU1500 Reconfigurable Acceleration PCIe® card.

Table 1: Device Support Archive (DSA)

Board	Device	Supported DSAs	Kernel Clock Frequency MHz	Status	Features
XIL-ACCEL-RD-KU115	KU115	xilinx:xil-accel-rd-pcie3-ku115:4ddr:4.0	300	Production	• PCIe Gen3x8, 4 DDR. • Kernel clock frequency control. • Automatic frequency scaling. • Second kernel clock at a higher frequency (up to 500MHz) is now supported that can be used for user created RTL kernels.

Board	Device	Supported DSAs	Kernel Clock Frequency MHz	Status	Features
					• Increased fabric resources for compute units. Global memory changes to volatile, between binary loads.
KCU1500	KU115	xilinx:kcu1500:4ddr:4.0	300	Production	• PCIe Gen3x8, 4 DDR. • Kernel clock frequency control. • Automatic frequency scaling. • Second kernel clock at a higher frequency (up to 500MHz) is now supported that can be used for user created RTL kernels. • Increased fabric resources for compute units. Global memory changes to volatile, between binary loads.
ADM-PCIE-KU3	KU60	xilinx:adm- pcie-ku3:2ddr-xpr:4.0	250	Production	PCIe Gen3x8, 2 DDR Automatic frequency scaling Increased fabric resources for compute units. Global memory changes to volatile, between binary loads

Board	Device	Supported DSAs	Kernel Clock Frequency MHz	Status	Features
ADM-PCIE-7V3	V7690T	xiilinx:adm-pcie-7v3:1ddr:3.0	200	Production	PCIe Gen3x8, 1DDR Automatic frequency scaling

Table 2: 2017.14.x Platform Driver Changes

DSA	User PF Driver	User Device Node	Management PF Driver	Management PF Node
xilinx:kcu1500:4ddr-xpr:4.0	`xdma`	`/dev/xdmaX_user /dev/xdmaX_c2h_0 /dev/xdmaX_c2h_1 /dev/xdmaX_h2c_0 /dev/xdmaX_h2c_1`	`xclmgmt`	`/dev/xclmgmtX`
xilinx:xil-accel-rd-ku115:4ddr-xpr:4.0	`xdma`	`/dev/xdmaX_user /dev/xdmaX_c2h_0 /dev/xdmaX_c2h_1 /dev/xdmaX_h2c_0 /dev/xdmaX_h2c_1`	`xclmgmt`	`/dev/xclmgmtX`
xilinx:xil-accel-rd-vu9p:4ddr-xpr:4.1	`xdma`	`/dev/xdmaX_user /dev/xdmaX_c2h_0 /dev/xdmaX_c2h_1 /dev/xdmaX_h2c_0 /dev/xdmaX_h2c_1`	`xclmgmt`	`/dev/xclmgmtX`
xilinx:adm-pcie-ku3:2ddr-xpr:4.0	`xdma`	`/dev/xdmaX_user /dev/xdmaX_c2h_0 /dev/xdmaX_c2h_1 /dev/xdmaX_h2c_0 /dev/xdmaX_h2c_1`	`xclmgmt`	`/dev/xclmgmtX`

2017.1 enhancements also include:

- SDx Eclipse UI
  - Platform setup and project creation.
    - Add custom platform(s) without creating an SDx project.
    - SDx example store: access to GitHub SDSoC and SDAccel examples.
    - Wizard for creating RTL kernels.
  - Acceleration
    - Support for accelerating C function templates.
    - Launch HLS for specified hardware functions.
  - Reporting
    - New post-implementation utilization reports.
    - Collapsible headers on all SDx reports.
- Xilinx Runtime
  - Early access `xocl` kernel driver, based on Linux kernel GEM framework for PCIe based DSAs has the following features.
    - Support for host page pinning which improves DMA bandwidth.
    - Uses Linux kernel based memory management for device memory management.
    - Is multi-threading safe and provides a single device node per device for all device operations.
    - `xocl` requires Redhat 6.9 or higher version or Ubuntu 16.04.
    - User can install `xocl`, by invoking `xbinst -gem <other options>`. The rest of the steps (e.g. running ./install.sh) remain the same.
  - Features ROM in device which advertises device configuration to the driver.
  - Migration to `xclbin2` format, which has features like skipping bitstream re-download if the same bitstream is already running.
  - Calling `clReleaseContext()` truly releases exclusive lock on the device so that another concurrently running application can create context with `clCreateContext()`.
  - Several new features in `xbask` including the new commands `scan`, `mem`, and `status`.
  - `xbsak` flash requires root permissions.
  - 4.X DSAs have AXI Firewal IP which protects PCIe from hangs and stalls inside the device. In case of AXI bus errors, AXI Firewall IP would trip which will cause the driver to send a SIGBUS to all applications which have opened the device node.
  - Runtime now includes support for latest version of OpenCL C++ wrappers from Khronos. The `cl2.hpp` header file ships along with standard OpenCL C API header files.
- `xocc` supports set target kernel frequency. Overriding kernel clock frequency with lower frequency might help with designs that fail to meet timing on platform clocks.
  - `--kernel_frequency<arg>` sets a user-defined clock frequency in MHz for kernel, overriding a default value from DSA.
  - For a kernel compilation to change target to 150 MHz, add `--kernel_frequency 150`.

- Usability
  - GDB extension to provide visibility into OpenCL data structures `cl_queue`, `cl_event`, and `cl_mem` to debug host application hangs.
  - Application timeline trace refinements.
    - Multicolor support for better visualization of OpenCL API calls.
    - Support for additional OpenCL APIs:
      - `clCreateContext`
      - `clCreateImage`
      - `clEnqueueTask`
      - `clEnqueueMigrateMemObjects`
      - `clEnqueueReadImage`
      - `clEnqueueWriteImage`
      - `clEnqueueMigrateMem`
      - `clEnqueueMapBuffer`
      - `clEnqueueUnmapMemObject`
  - Detailed Kernel Trace
    - Support for RTL kernels.
    - Reporting loop pipeline activity in waveform.
- Enhanced debug checks in HW Emulation Flow covering:
  - Kernel or System transactions hangs.
  - Uninitialized memory read by kernel.
  - Out of DDR Range access.
  - Out of Bounds array access.
  - Periodic aliveness status during long HW Emulation runs.
- User needs to have a g++ compiler available in the system to compile C++ host code. In the past, SDx tool had a version of g++, but no longer does.

XILINX
ALL PROGRAMMABLE

Chapter 2

Introduction to the SDx Environments

The 2017.1 SDx™ Environment software release consists of the SDSoC™ Development Environment for Zynq® Ultrascale+ MPSoC and Zynq-7000 SoC families, and the SDAccel™ Development Environment for Data Center and PCI-e based accelerator systems. These environments share a common installer, but are licensed individually. All SDx Environments include the Vivado® Design Suite for programming the target devices and for developing custom hardware platforms.

SDSoC Overview

The SDSoC™ (Software-Defined System On Chip) environment is an Eclipse-based Integrated Development Environment (IDE) for implementing heterogeneous embedded systems using the Zynq®-7000 All Programmable SoC and Zynq UltraScale+™ MPSoC platforms. The SDSoC environment provides an embedded C/C++ application development experience with an easy-to-use Eclipse IDE, and comprehensive design tools for heterogeneous Zynq-7000 AP SoC and Zynq UltraScale+ MPSoC development to software engineers and system architects.

New in 2017.1, the SDSoC Environment introduces support for OpenCL™ applications with hardware kernels that target Zynq Ultrascale+ MPSoC devices.

The SDSoC environment includes a full-system optimizing C/C++ compiler that provides automated software acceleration in programmable logic combined with automated system connectivity generation. The application programming model within the SDSoC environment should be intuitive to software engineers. An application is written as C/C++ code, with the programmer identifying a target platform and a subset of the functions within the application to be compiled into hardware. The SDSoC system compiler then compiles the application into hardware and software to realize the complete embedded system implemented on a Zynq device, including a complete boot image with firmware, operating system, and application executable.

The SDSoC environment abstracts hardware through increasing layers of software abstraction that includes cross-compilation and linking of C/C++ functions into programmable logic fabric as well as the ARM CPUs within a Zynq device. Based on a user specification of program functions to run in programmable hardware, the SDSoC environment performs program analysis, task scheduling and binding onto programmable logic and embedded CPUs, as well as hardware and software code generation that automatically orchestrates communication and cooperation among hardware and software components.

SDAccel Overview

SDAccel™ is a development environment for OpenCL™ applications targeting Xilinx FPGA based accelerator cards. This environment enables concurrent programming of the in-system processor and the FPGA fabric without the need for extensive FPGA design experience. The application is captured as a host program written in OpenCL C and a set of computation kernels expressed in C, C++, OpenCL C, or RTL.

Hardware Requirements

SDSoC Hardware Requirements

The 2017.1 SDSoC™ environment release includes support for the following development boards:

- ZC702, ZC706, MicroZed, ZedBoard, and Zybo development boards featuring the Zynq-7000 AP SoC
- ZCU102 development board featuring the Zynq UltraScale+ MPSoC.

Additional platforms are available from partners. Also, the SDSoC Platform Utility enables you to target any custom Zynq and Zynq UltraScale+ board. For more information, visit the SDSoC Developer Zone: https://www.xilinx.com/products/design-tools/software-zone/sdsoc.html.

You also need a mini-USB cable to observe the UART output from the board.

SDAccel Hardware Requirements

The SDAccel™ environment requires the following hardware:

- Acceleration Card. Use one of the following:
  - Alpha Data ADM-PCIE-KU3 card. The card is based on the Kintex® UltraScale™ XCKU060T-2FFVA1156E FPGA.
  - Alpha Data ADM-PCIE-7V3 card. The card is based on the Virtex®-7 XC7VX690T-2FFG1157C FPGA.
  - Xilinx Xil-ACCEL-RD-KU115 card. The card is based on the Kintex UltraScale XCKU115-FLVB2104-2-E FPGA.
  - Xilinx Kintex UltraScale FPGA KCU1500 Reconfigurable Acceleration card based on XCKU115-FLVB2104-2-E FPGA.

- Host computer: Desktop computer for hosting the acceleration card. The host computer must provide the following.
  - Motherboard with a PCIe Gen3 X8 slot
  - 16 GB RAM
  - 100GB free disk space
  - DVD drive
- Programming computer: Laptop or desktop computer running the supplied Vivado Design Suite 2016.4 for programming the FPGA.
- Xilinx® Platform Cable USB 2, part number HW-USB-II-G for connecting the programming computer to the acceleration card. See the *Platform Cable USB II Data Sheet*, (DS593).
- Additional platforms are available from partners. For more information, visit the SDAccel Developer Zone: http://www.xilinx.com/products/design-tools/software-zone/sdaccel.html.

Known Issues

Xilinx performs substantial testing across of a number of different systems. The following are issues that we have identified on some common systems.

Table 3: Known Issues

System	Known Issues
ASUS P8Z77 WS	Receiver errors (DLLP Errors) are seen sometimes. They are benign and can be safely ignored.
ASUS Z170-A	This system will not boot with CentOS/RHEL 6.8. Please use CentOS/RHEL 7 on these systems.
Gigabyte AX370-Gaming 5	The VU9P card links at Gen1x8 on the middle slot which is a x8 slot. Please use a different PCIe slot.
SuperMicro X10 SDV-TLN4F	No known issues.
SuperMicro X10SRi-F	All slots have some issues (Link happens on this card but DMA fails). Please find a different system to use for SDAccel Development.

Software Requirements

The SDx™ Development Environment runs on both Linux and Windows operating systems. The supported operating systems are listed below.

- Windows 7 and 7 SP1 Professional (64-bit) (SDSoC only)
- Windows 10 Professional (64-bit) (SDSoC only)

- Linux Support
  - Red Hat Enterprise Workstation/Server 7.2-7.3 (64-bit)
  - Red Hat Enterprise Workstation 6.7 and 6.8 (64-bit)
  - CentOS 6.8, CentOS 7.3 (64-bit) (SDAccel)
  - Ubuntu Linux 16.04.1 LTS (64-bit)

About the SDSoC Installation

The installation of SDSoC™ includes the following:

- SDSoC environment, including an Eclipse/CDT-based GUI, high-level system compiler, and ARM GNU toolchain
- Vivado® Design Suite System Edition, with Vivado High-Level Synthesis (HLS) and the Xilinx® Software Development Kit (SDK)

The SDSoC environment includes the same GNU ARM toolchain included with the Xilinx Software Development Kit (SDK), which also provides additional tools used by the SDSoC environment. The SDSoC environment setup script sets PATH variables to use this toolchain.

More information about the SDSoC installation:

- The provided toolchains contain 32-bit executables, requiring your Linux host OS installation to include 32-bit compatibility libraries.
- RHEL 6 and 7 64-bit x86 Linux installations might not include the 32-bit compatibility libraries, and might need to be added separately; see https://access.redhat.com/site/solutions/36238.
- On RHEL, 32-bit compatibility libraries can be installed by becoming a superuser (or root) with root access privileges and running the yum install glibc.i686 command.
- On Ubuntu, 32-bit compatibility libraries can be installed by becoming a superuser (or root) with root access privileges and running the following commands. Refer to SDSoC Development Environment Features for additional information.

```
sudo dpkg --add-architecture i386
sudo apt-get update
sudo apt-get install libc6:i386 libncurses5:i386 libstdc++6:i386
sudo apt-get install libgtk2.0-0:i386 dpkg-dev:i386
sudo ln -s /usr/bin/make /usr/bin/gmake
```

- The version of the toolchain can be displayed by running the `arm-linux-gnueabihf-g++ -v` command.
- The last line of the output printed in the shell window should be GCC version 4.9.2 20140904 (pre-release)(crosstool-NG linaro-1.13.1-4.9-2014.09 - Linaro GCC 4.9-2014.09).

About the SDAccel Installation

The SDAccel™ Environment runs on the Linux operating systems only with no support for Windows. It supports RedHat Enterprise Linux, CentOS 6.9 and 7.3 64-bit, as well as Ubuntu 16.04 64-bit.

CentOS/RHEL 7.3 (6.9 also supported) Package List

Please install the EPEL repository using the instructions at https://fedoraproject.org/wiki/EPEL. In addtion, the following packages should be installed with the `yum` install command.

- `ocl-icd`
- `ocl-icd-devel`
- `opencl-headers`
- `kernel-headers`
- `kernel-devel`
- `gcc-c++`
- `gcc`
- `gdb`
- `make`
- `opencv`
- `libjpeg-turbo-devel`
- `libpng12-devel`
- `python`
- `git`
- `unzip`
- `dmidecode`
- `redhat-lsb`
- `kernel-headers-$(uname -r)`

Ubuntu 16.04 Package List

The following packages should be installed with `apt-get install` command.

- `ocl-icd-libopencl1`
- `opencl-headers`
- `ocl-icd-opencl-dev`
- `linux-headers`
- `linux-libc-dev`
- `g++`
- `gcc`
- `gdb`
- `make`
- `libopencv-core`
- `libjpeg-dev`
- `libpeng-dev`
- `python`
- `git`
- `dmidecode`
- `lsb`
- `unzip`
- `linux-headers-$(uname -r)`

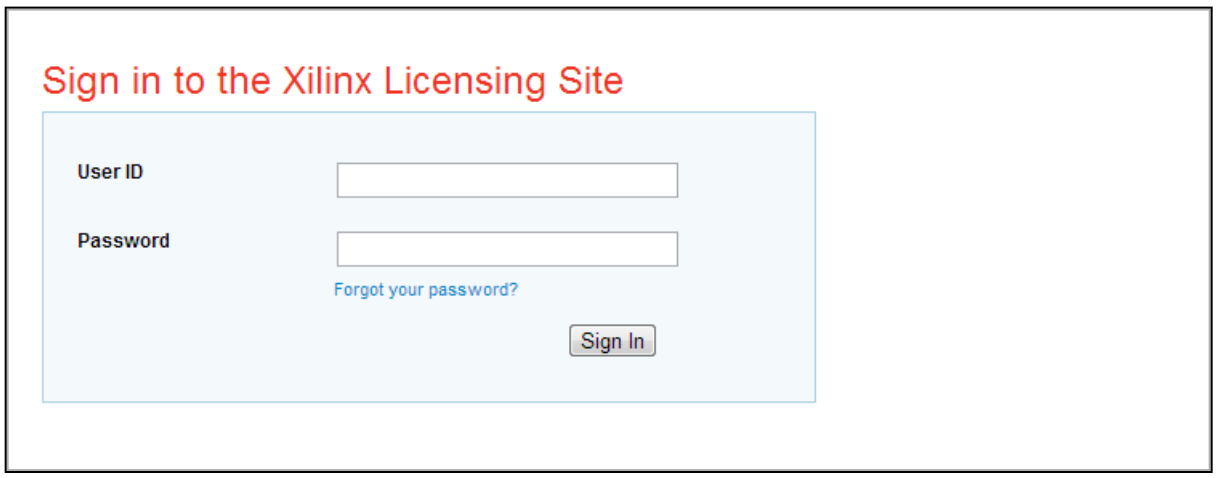
Obtaining a License

The steps to obtain a license for the SDx™ development environment are described below.

Generating a License on the Xilinx Licensing Site

1. Sign in to the Xilinx® licensing website: https://www.xilinx.com/getproduct. See the following figure.

   Table 4: Xilinx Licensing Site Sign-in Screen



   If this is your first time generating a license for the SDAccel™ - Xilinx OpenCL™ Design Environment, contact your Xilinx representative to enable your access to the SDAccel licensing website.

   SDSoC comes with a 60-day evaluation license, so you should be able to see it in your available license list.

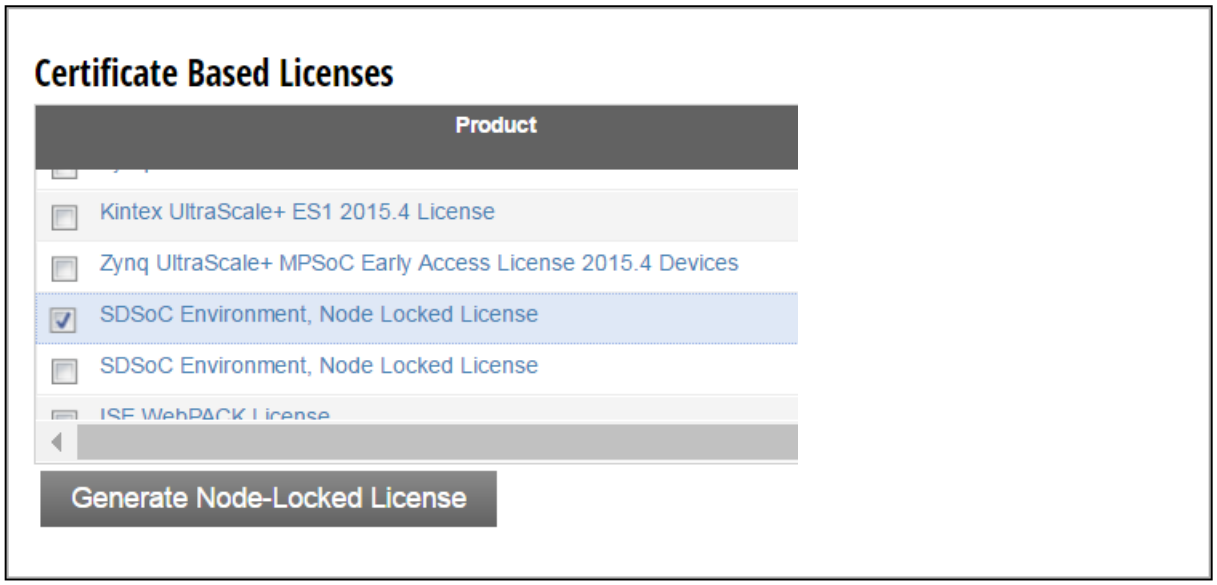
2. In the account drop-down menu, select **XILINX - SDSoC Environment** or **SDAccel Environment**.

   NOTE: *This only shows up if you have purchased or redeemed an SDSoC™ or SDAccel license.*

IMPORTANT: *If you are interested in SDSoC, you should also see a "SDSoC 60-day evaluation license" for the first time use.*

3. From the Certificate Based Licenses menu, select **SDSoC Environment, Node-Locked License** or **SDAccel Environment, Node-Locked License**.

Table 5: Certificate Based Licenses Menu



4. Click **Generate node-locked license**.
5. Enter a Host ID in the **License Generation** screen and click **Next**.
6. Verify that the Host ID for the license is correct and click **Next**.
7. Accept the licensing agreement by clicking **Accept**.

   You will receive an email from xilinx.notification@entitlenow.com with the license file.

8. Set the `XILINXD_LICENSE_FILE` environment variable to point to the location of the license file on your system.

XILINX
ALL PROGRAMMABLE™

Chapter 4

Installing the SDx Environments

This chapter explains the installation process for either the SDSoC™ environment or the SDAccel™ environment.

Preparing to Install the Tools

NOTE: *Before starting installation, you must complete the following steps.*

1. Make sure your system meets the requirements described in Hardware Requirements and Software Requirements.
2. Disable anti-virus software to reduce installation time.
3. Close all open programs before you begin installation.

Installing SDSoC and SDAccel

You have two options for installation of SDSoC™ and SDAccel™. Both installation types are available on the Xilinx® Downloads Website.

NOTE: *There are separate installers for SDSoC and SDAccel. When you launch the installer for the product you want to use, the devices are preselected for you.*

Using the Web Installer

Using the web installer is recommended.

Using the web installer you can pick and choose what you would like to install up front and that is the only data that will need to get downloaded for installation. Also, in the case of a network failure, you can resume from where you last stopped, instead of starting from the beginning again.

NOTE: *The following devices are pre-selected in the individual installers: - For the SDAccel™-specific web installer, 7 Series, UltraScale™, and UltraScale+™ are pre-selected. - For the SDSoC-specific web installer, Zynq®-7000 and UltraScale+ MPSoC are pre-selected. - For the combined SDx SFD (single file download) image, no devices are pre-selected.*

Downloading and Installing the Full Installation File

If you downloaded the full product installation, decompress the file and run xsetup (for Linux) or xsetup.exe (for Windows, not available for SDAccel™) to launch the installation.

If you downloaded the web installer client, launch the downloaded file.You are prompted to log in and use your regular Xilinx login credentials to continue with the installation process.

Xilinx recommends the use of 7-zip or WinZip (v.15.0 or newer) to decompress the downloaded `tar.gz` file.

- The **Download and Install Now** choice allows you to select specific tools and device families on following screens, downloads only the files required to install those selections, and then installs them for you. After entering your login credentials, you can select between a traditional web-based installation or a full image download.
- The **Download Full Image** requires you to select a download destination and to choose whether you want a Windows only, Linux only, or an install that supports both operating systems. There are no further options to choose with the **Download Full Image** selection, and installation needs to be done separately by running the xsetup application from the download directory.

Batch Mode Installation Flow

The installer can run in an unattended batch process. To run unattended, a standard edition and install location must be specified or a configuration file must be present which tells the installer the install location and which of the tools, devices and options you wish to install. The installer has a mode in which it can generate a reference option file for you based on common configurations, which you can further edit to customize your installation.

Xilinx recommends that you generate this reference for each new quarterly release, so that new devices, tools, options or other changes will be accounted for in your options file.

To begin using batch mode, open a command shell and change to the directory where you have stored your extracted installer.

For Windows, open the command window with administrator privileges and run the xsetup.bat file, found in the `\bin` directory, and not `xsetup.exe` with the options below.

Generating a Configuration File

1. Run: `xsetup -b ConfigGen`

   This will put you in an interactive mode where you will see the following menu. Choose the SDx™ IDE for SDSoC™ and SDAccel™ development environments edition.

2. After you make a selection, you will be prompted with the location/filename for your configuration file and the interactive mode exits.

   Below is a sample configuration file:

```
#### SDx IDE for SDSoC and SDAccel development environments Install
Configuration
####
Edition=SDx IDE for SDSoC and SDAccel development environments

# Path where Xilinx software will be installed.
Destination=/opt/Xilinx

# Choose the Products/Devices the you would like to install.
Modules=DocNav:1,UltraScale+:0,7 Series:0,Zynq UltraScale+
MPSoC:0,Zynq-7000:0,UltraScale:0

# Choose the post install scripts you'd like to run as part of the
finalization step.
Note that some of these scripts may require user interaction during
runtime.
InstallOptions=Enable WebTalk for SDx to send usage statistics to Xilinx:1

## Shortcuts and File associations ##
# Choose whether Start menu/Application menu shortcuts will be created or
not.
CreateProgramGroupShortcuts=1

# Choose the name of the Start menu/Application menu shortcut. This
setting will be
ignored if you choose NOT to create shortcuts.
ProgramGroupFolder=Xilinx Design Tools

# Choose whether shortcuts will be created for All users or just the
Current user.
Shortcuts can be created for all users only if you run the installer as
administrator.
CreateShortcutsForAllUsers=0

# Choose whether shortcuts will be created on the desktop or not.
CreateDesktopShortcuts=1

# Choose whether file associations will be created or not.
CreateFileAssociation=1
```

Each option in the configuration file matches a corresponding option in the GUI. A value of `1` means that option is selected; a value of `0` means the option is unselected.

NOTE: *In this configuration file, by default there are no devices selected for installation (all devices have a value of 0). You MUST update a device to a value of 1 in order to install that device.*

Running the Installer

Now that you have edited your configuration file to reflect your installation preferences, you are ready to run the installer. As part of the command line installer, you must indicate your acceptance of the Xilinx End-User License Agreement Xilinx End-User License Agreement and Third Party End-User License Agreement Third Party End-User License Agreement, and confirm you understand the WebTalk Terms and Conditions.

WebTalk Terms and Conditions

The WebTalk Terms and Conditions, which you must agree to when running the installer, reads as follows:

By indicating `I AGREE`, I also confirm that I have read Section 13 of the terms and conditions above concerning WebTalk and have been afforded the opportunity to read the WebTalk FAQ posted at http://www.xilinx.com/webtalk. I understand that I am able to disable WebTalk later if certain criteria described in Section 13(c) apply. If they don't apply, I can disable WebTalk by uninstalling the Software or using the Software on a machine not connected to the internet. If I fail to satisfy the applicable criteria or if I fail to take the applicable steps to prevent such transmission of information, I agree to allow Xilinx to collect the information described in Section 13(a) for the purposes described in Section 13(b).

When using the command line, use the command-line switch, `-a` or `--agree`, to indicate your agreement to each of the above. If one of the above is left out of the list, or the agree switch is not specified, the installer exits with an error and does not install.

Verifying Connectivity

The installer connects to the Internet through the system proxy settings in Windows. These settings can be found under **Control Panel** > **Network and Internet** > **Internet Options**. For Linux users, the installer uses Firefox browser proxy settings (when explicitly set) to determine connectivity.

Figure 1: Vivado Design Suite Installation - Connectivity

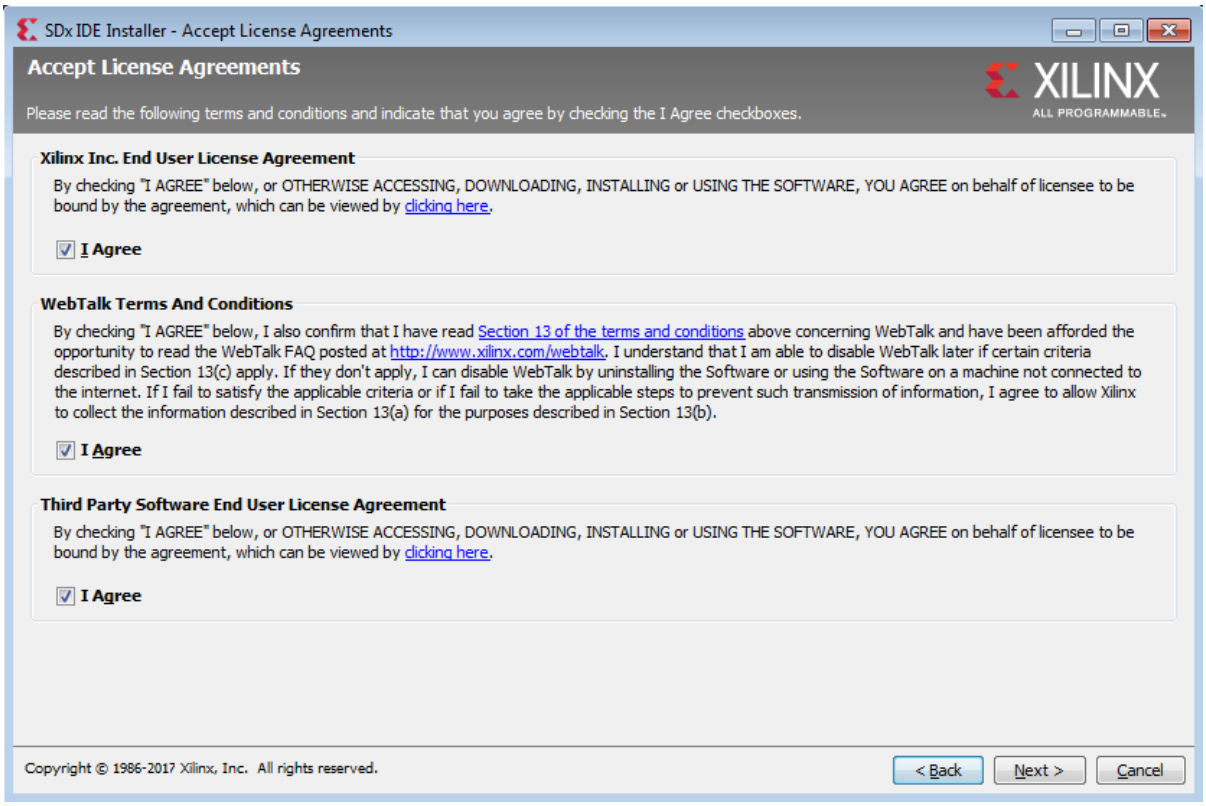
If there are connectivity issues, verify the following:

1. If you are using alternate proxy settings to the ones referred to, select the **Manual Proxy Configuration** option to specify the settings.
2. Check whether your company firewall requires a proxy authentication with a user name and password. If so, select the **Manual Proxy Configuration** option in the dialog box above.
3. For Linux users, if either the **Use System settings** or the **Auto detect settings** option is selected in the Firefox browser, you must manually set the proxy in installer.

Accepting License Agreements

Carefully read the license agreements before continuing with the installation. If you do not agree to the terms and conditions, cancel the installation and contact Xilinx.

Figure 2: License Agreements

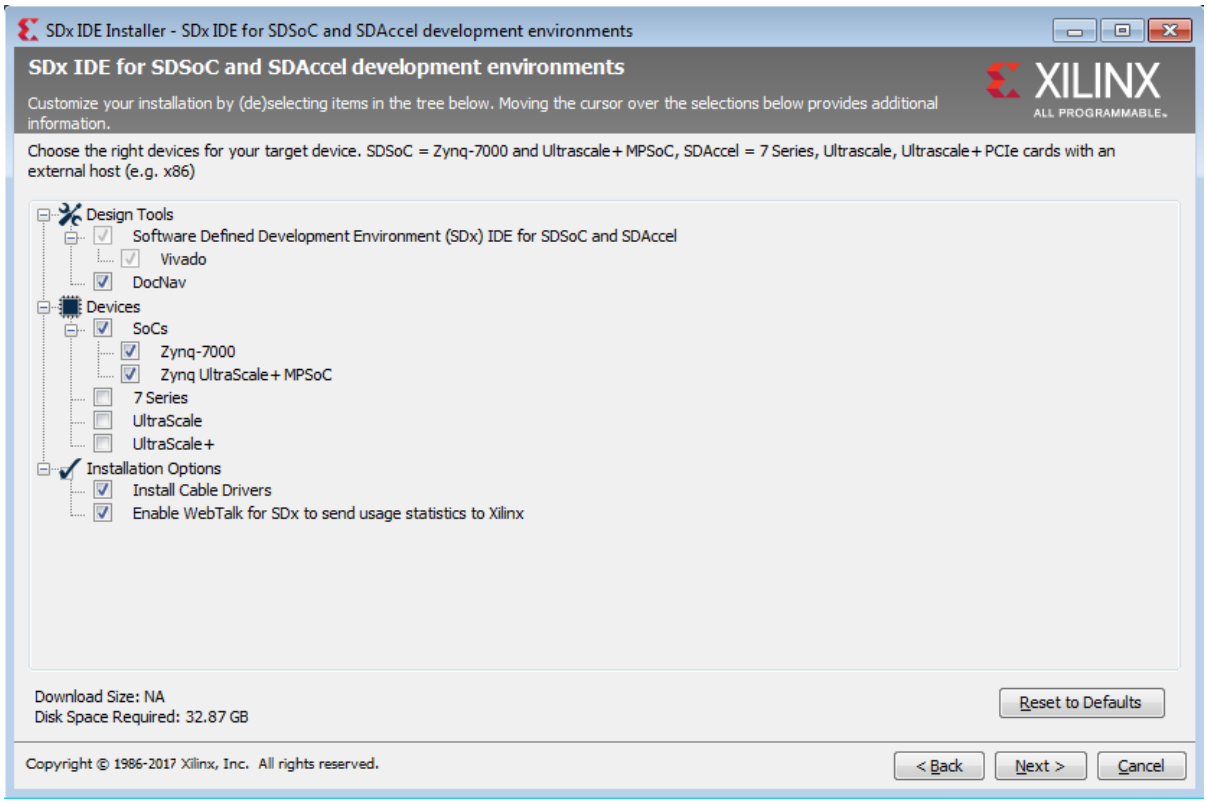


Selecting Tool and Device Options

Customize the installation by choosing the design tools, device families and installation options. Selecting only what you need helps to minimize the time taken to download and install the product. You will be able to add to this installation later by clicking **Add Design Tools or Devices** from either the operating system Start Menu or the **Vivado** > **Help** menu.

When you launch the installer for the product you want to use, the devices are preselected for you.

Figure 3: Design Tools and Device Options



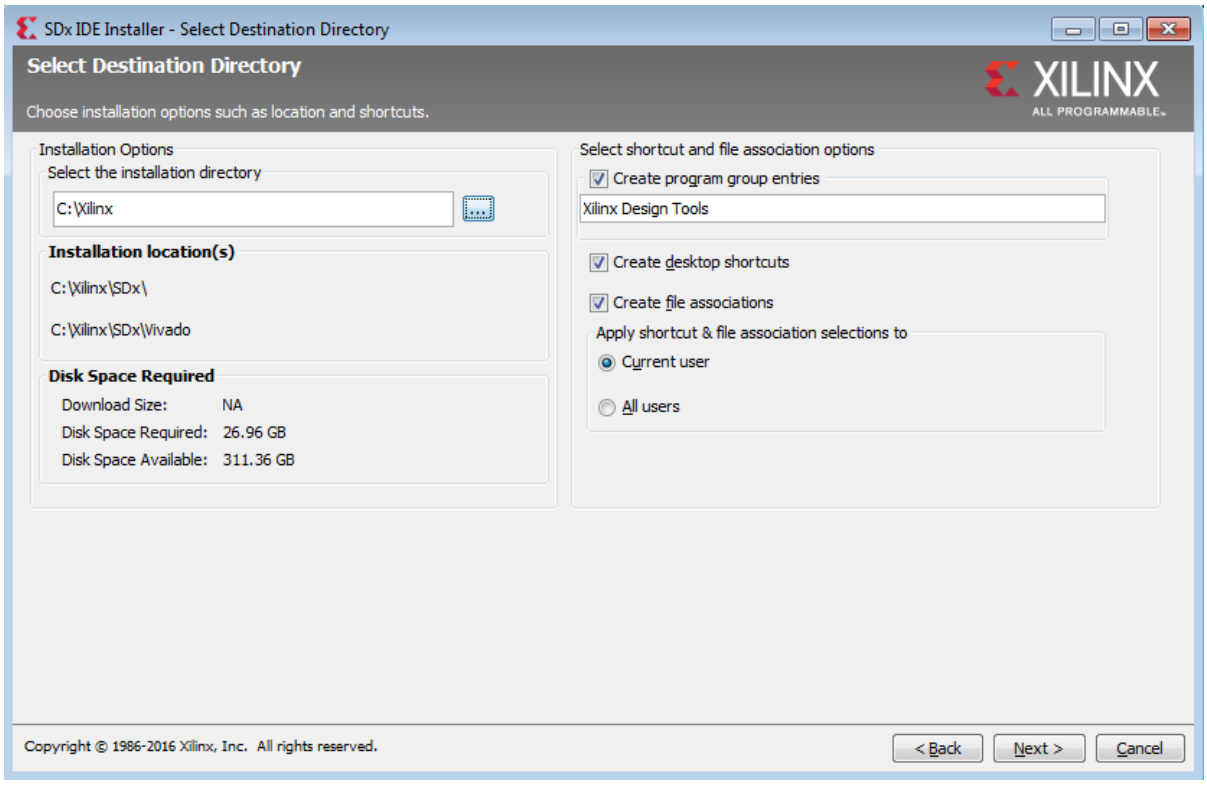
Setting Destination Directory and Installation Options

Define the installation directory for the software, as shown in the following figure.

NOTE: *The installation directory name must not contain any spaces in any part of the directory path.*

You can customize the creation of the program group entries (Start Menu) and the creation of desktop shortcuts. The shortcut creation and file association options can be applied to the current user or all users.

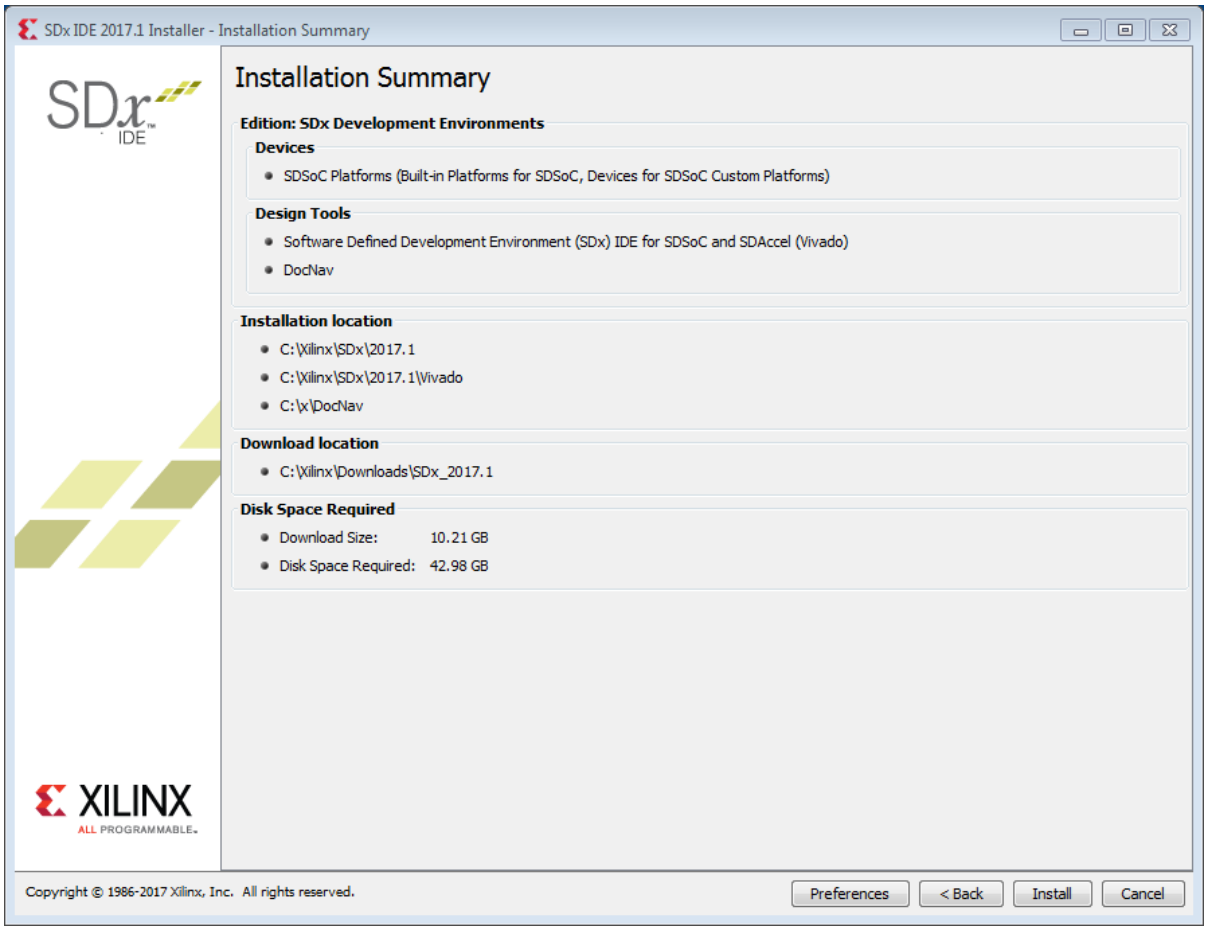
Figure 4: Destination Directory and Installation Options



Reviewing the Installation Details

Review the installation details shown in the **Installation Summary** screen.

Figure 5: Installation Summary



When you click **Install**, the installation process takes several minutes to complete.

Setting Up the Environment to Run SDx

To set up the environment to run SDx™, source the file below so that `sdx` command is in the `PATH`:

```
C Shell: source <SDX_INSTALL_DIR>/settings64.csh
Bash: source <SDX_INSTALL_DIR>/settings64.sh
```

XILINX
ALL PROGRAMMABLE™

Chapter 5

Additional Resources and Legal Notices

Xilinx Resources

For support resources such as Answers, Documentation, Downloads, and Forums, see Xilinx Support.

Solution Centers

See the Xilinx Solution Centers for support on devices, software tools, and intellectual property at all stages of the design cycle. Topics include design assistance, advisories, and troubleshooting tips.

Additional Resources and Legal Notices

Xilinx Resources

For support resources such as Answers, Documentation, Downloads, and Forums, see Xilinx Support.

Solution Centers

See the Xilinx Solution Centers for support on devices, software tools, and intellectual property at all stages of the design cycle. Topics include design assistance, advisories, and troubleshooting tips.

References

SDAccel Documents

SDAccel Environment User Guide (UG1023)

SDAccel Environment Optimization Guide (UG1207)

SDAccel Environment Tutorial: Introduction (UG1021)

SDAccel Environment Platform Development Guide (UG1164)

SDSoC Documents

SDSoC Environment User Guide (UG1027)

SDSoC Environment Optimization Guide (UG1235)

SDSoC Environment Tutorial: Introduction (UG1028)

SDSoC Environment Platform Development Guide (UG1146)

Additional Documents

SDx Pragma Reference Guide (UG1253)

Xilinx OpenCV User Guide (UG1233)

Platform Cable USB II Data Sheet (DS593)

Xilinx® licensing website: https://www.xilinx.com/getproduct

SDSoC Developer Zone: https://www.xilinx.com/products/design-tools/software-zone/sdsoc.html.

SDAccel Developer Zone: http://www.xilinx.com/products/design-tools/software-zone/sdaccel.html

Xilinx End-User License Agreement (UG763)

Third Party End-User License Agreement (UG1254)

Documentation Navigator and Design Hubs

Xilinx® Documentation Navigator provides access to Xilinx documents, videos, and support resources, which you can filter and search to find information. To open the Xilinx Documentation Navigator (DocNav):

- From the Vivado® IDE, select **Help > Documentation and Tutorials**.
- On Windows, select **Start > All Programs > Xilinx Design Tools > DocNav**.
- At the Linux command prompt, enter `docnav`.

Xilinx Design Hubs provide links to documentation organized by design tasks and other topics, which you can use to learn key concepts and address frequently asked questions. To access the Design Hubs:

- In the Xilinx Documentation Navigator, click the **Design Hubs View** tab.
- On the Xilinx website, see the Design Hubs page.

For more information on Documentation Navigator, see the Documentation Navigator page on the Xilinx website.

Please Read: Important Legal Notices

AUTOMOTIVE APPLICATIONS DISCLAIMER

AUTOMOTIVE PRODUCTS (IDENTIFIED AS "XA" IN THE PART NUMBER) ARE NOT WARRANTED FOR USE IN THE DEPLOYMENT OF AIRBAGS OR FOR USE IN APPLICATIONS THAT AFFECT CONTROL OF A VEHICLE ("SAFETY APPLICATION") UNLESS THERE IS A SAFETY CONCEPT OR REDUNDANCY FEATURE CONSISTENT WITH THE ISO 26262 AUTOMOTIVE SAFETY STANDARD ("SAFETY DESIGN"). CUSTOMER SHALL, PRIOR TO USING OR DISTRIBUTING ANY SYSTEMS THAT INCORPORATE PRODUCTS, THOROUGHLY TEST SUCH SYSTEMS FOR SAFETY PURPOSES. USE OF PRODUCTS IN A SAFETY APPLICATION WITHOUT A SAFETY DESIGN IS FULLY AT THE RISK OF CUSTOMER, SUBJECT ONLY TO APPLICABLE LAWS AND REGULATIONS GOVERNING LIMITATIONS ON PRODUCT LIABILITY.