Active-HDL 7.1 sp2 Release Notes

This page is intentionally left blank.

Table of Contents

What's New in Service Pack 2 for Version 7.1?

The following is a brief overview of new features and changes introduced to Active-HDL 7.1.SP2 (BUILD 1583.SP2.09, 05/23/2006):

Licensing

- Since Service Pack 1 for version 7.1, a valid maintenance contract has been required in order to run Active-HDL. Only users who have a valid maintenance contract will be able to apply Service Pack 2 for the release build of Active-HDL 7.1 (BUILD 1583, 10/11/2005) or upgrade from Service Pack 1 for version 7.1 (BUILD 1583.SP1.23, 03/08/2006). Users whose maintenance contract has expired should not install Service Pack 2 as Active-HDL will not run with their existing license and report an error. For more information refer to the License Maintenance chapter.

Compiler and Simulator

VHDL and Verilog Compilation

- Service Pack 2 for Active-HDL 7.1 provides an improvement in HDL compilation. Now, designs that contain a number of units can be compiled faster (with the comparison to the previous version of Active-HDL) since the time of HDL compilation has been reduced twice (approximately). The optimization significantly improves the process of building system and vendor libraries.

C Compilation

- The "-Wl,--noinhibit-exec," arguments have been removed from the options of the compilation performed by using the DLM C/C++ Configuration (**File | New | C/C++ Configuration**). This change improved detection of compilation errors. Now, errors found during the compilation are displayed in red in the **Console** window.
- The template for PLI applications in the C/C++ Configuration has been changed. Now, PLI applications are linked with the *aldecpli.dll* library instead of *aldecpli.lib*.

Design Flow Manager

- **New flowcharts**

  HDL Synthesis
  1. Altera Quartus II 6.0
  2. Synplicity FPGA Synthesis 8.5/8.5.1
  3. Synplicity Synplify 8.5 for Actel
  4. Synplicity Synplify 8.5 for Lattice

  Physical Synthesis
  1. Magma PALACE 3.2 for Actel
  2. Xilinx PlanAhead 8.1.5

  Implementation

1. Actel Designer 7.1; the Design Flow Manager also provides the separated flowcharts for previous versions of the Actel implementation tool i.e. Actel Designer ver. 6.0, 6.1, 6.2, and 7.0. The flowcharts support all available service packs.
2. Altera Quartus II 6.0; the Design Flow Manager also supports the flowcharts for previous versions of Quartus II, i.e. Quartus II 3.x, 4.0, 4.1, 4.2, 5.0, and 5.1 along with available service packs.
3. Lattice ispLEVER 6.0; the Design Flow Manager also supports the flowcharts for previous versions of the Lattice implementation tools, i.e. Lattice ispLEVER ver. 4.1, 4.2, 5.0, and 5.1.
4. Xilinx ISE/WebPack 8.1 (supports Service Pack 3 for Xilinx ISE/WebPack 8.1i); the Design Flow Manager also supports the flowcharts for previous versions of the Xilinx synthesis/implementation tools, i.e. ISE/WebPack 4.x, 5.1, 5.2, 6.1, 6.2/6.3, and 7.1.

- **Updated flowcharts**

HDL Synthesis
1. Altera Quartus II 5.1 SP2 Synthesis & Implementation
2. Mentor Graphics Precision RTL Synthesis (based on Precision RTL Synthesis 2005c.79 for Lattice)

Implementation
1. Actel Designer 7.0 SP1
2. Altera Quartus II 5.1 SP2
3. Lattice ispLEVER 5.1 SP2
QuickLogic QuickWorks 9.8

- **Flowchart improvements and changes**

Altera flowchart
1. The layout of the flowchart has been changed and supplemented with a new group of auxiliary tools. Now, there are two pop-up windows available: **Design Entry Tools** and **Post-Layout Tools**. Additionally, the **Megafunctions Wizard** button has been relocated and now it can be found in the **Design Entry Tools** window.
2. Service Pack 2 allows running TimeQuest Timing Analyzer and Design Space Explorer directly from the Active-HDL environment. These new tools are available in the **Post-Layout Tools** and **Design Entry Tools** menu, respectively.

Atmel flowchart
1. The name of the *A*tmel flowchart has been changed from *Figaro IDS 7.5* to *Figaro IDS 7.x*. The flowchart supports both the version 7.5 and 7.6 of the Atmel implementation tool.

Lattice flowchart
1. Service Pack 2 enhances the set of the available design entry tools for the Lattice ispLEVER 6.0 flowchart. Now Active-HDL allows running Memory Initialization Tool directly from the Design Flow Manager window. The tool is available in the **Design Entry Tools** menu.

Libraries

- **Updated vendor libraries**
HDL Synthesis
1. Synplicity FPGA Synthesis 8.5.1

Implementation
1. Actel Designer 7.1
2. Altera Quartus II 6.0
3. Lattice ispLEVER 6.0
4. QuickLogic QuickWorks 9.8
5. Xilinx ISE 8.1i SP3

- **Updated schematic libraries**
  1. Xilinx ISE 8.1i SP3
- **New schematic libraries**
  1. The unisim and ovi_unisim libraries have been supplemented with symbols that can be used on block diagrams. Similarly to other libraries, the symbols stored in these libraries can be seen in the **Symbols Toolbox** window.

Block Diagram Editor

- The **Update Complex Types** option has been introduced to the Block Diagram Editor. The new option available in the **Diagram** main menu updates all signals of the complex type (arrays or records) declared on the current block diagram sheet. The option reads the contents of the library and checks whether a package containing type declarations has been changed. If so, the type and all block diagram objects using this type are automatically updated. These objects are also updated when the object properties dialog box is opened.

State Diagram Editor

- The State Diagram Editor has been enhanced with a new object that allows specifying an internal delay for transitions between states of a state machine. The **Delay** state (**FSM | Delay**) defines a period of time (expressed as an integer multiple of a synchronization signal (Clock) cycle) that must elapse before a transition from one state to another occurs on the active clock edge. The **Delay Properties** dialog box allows specifying all parameters of the internal delay object. The **Delay** object can be used on both VHDL and Verilog state diagrams.
- Service Pack 2 introduces a new property in the **Machine Properties** dialog box. The **State Actions** tab allows selecting a type of actions (*Entry, Exit*, or *State Actions*) that will be available while working with the State Diagram Editor. The editor will automatically generate warnings to the **Console** window if it detects the use of an illegal type of the state diagram actions during the DRC process (always run prior to generating an HDL source code). As the allowed type of the actions used on a state diagram specifies also the kind of the state machine, this tab may directly point whether you intend to build a Mealy or Moore finite state machine.
- The State Diagram Editor supports user-defined types and specification of an initial value for the ports, signals, or variables. Active-HDL 7.1 with Service Pack 2 installed allows not only manual specification of the type and initial value but it also provides ready-to-use templates for most frequently used standard types. You can specify the object type in the **General** tab of the **Port/Signal/Variable Properties** dialog box or select the predefined type template from the **Type and Initial Value Templates** window.
- The **State Codes Visible** option has been added to the properties of the state object. It enables the display of the state encoding value (specified in the **Code** edit box) on the state diagram. The new option can be set in three different states: checked (state codes are displayed), unchecked

(the display of the state encoding value is disabled locally), and grayed (the current setting for this option is inherited from the **State Diagram Editor** category of the **Preferences** dialog box). The **State Codes Visible** option can be found on the **General** tab of the **State Properties** dialog box.

- The **View/Sort Objects** window (**FSM | View/Sort Objects**) has been supplemented with the **Declarations** tab that lists object declarations appropriate to the selection made to the **Select Machine** list box. Additionally, the order of the specified declarations, i.e. Diagram Declarations, Diagram Declarations after Signal Declarations, Machine Declarations, and Machine Declarations after Variable Declarations) can be individually set for each state machine of the state diagram.
- Generation of Verilog source code has been enhanced. Now, the declaration of FSM states (values of state register) and their encoding (state code) can be specified by using the `localparam` construct. The State Diagram Editor still supports the `parameter` and `'define` construct. The construct used to define machine states and state encoding can be selected in the **Code Generation Settings** dialog box (**FSM | Code Generation Settings**).

Active-HDL Interfaces and Wizards

The following changes and improvements have been made to the built-in Active-HDL DSP interfaces:

- **Simulink® Interface**
  1. Service Pack 2 provides support for Xilinx System Generator ver. 8.1. The **HDL Black-Box Manager for System Generator 8.1** block has been added to the Active-HDL Blockset. The implementation of this block results in introduction of a new method of HDL black-box instantiation in Simulink block diagrams (based on the Configuration Subsystem Manager). Active-HDL 7.1 SP2 continues to support previous versions of Xilinx System Generator (based on the Simulation Multiplexer) and still allows co-simulation of designs that instantiate the **HDL Black-Box for System Generator** blocks coming from Xilinx System Generator ver. 6.x/7.x.
  2. An improvement has been introduced to mechanisms that create a workspace used for post co-simulation verification (HDL simulation performed after co-simulation and based on test-vectors created by Simulink during co-simulation). Now, a temporary workspace that is created for co-simulation purposes (after co-simulation is initialized in the Simulink environment) as well as for dumping co-simulation test-vectors can be moved to any location or another machine. It allows running HDL simulation that uses stimulators created directly by Simulink with no need to control the path where the original workspace (containing simulation libraries and source files) is stored.
  3. The on-line documentation of the Simulink Interface has been enhanced and supplemented with more information on how use the Active-HDL Blockset, instantiate blocks of this blockset in Simulink diagrams, specify block parameters (*Using Active-HDL Blockset*), prepare a design for stand-alone simulation and verify HDL model (*Post Co-simulation Design Verification*).

The following changes and improvements have been made to the import of third-party projects:

- **Import of Altera SOPC Simulation Scripts**
  Service Pack 2 for Active-HDL 7.1 provides the update to the import of simulation scripts generated by Altera SOPC Builder (ver.6.0 Build 213). Now, the process of the import can also be started when no design is loaded in Active-HDL. The **Altera SOPC Simulation Script** option is available in the **File | Import** menu.
- **Import of Xilinx EDK Simulation Scripts**
  Active-HDL 7.1 SP2 provides the update to the import of simulation scripts coming from the Xilinx EDK 8.1 SP2 design environment. The **Xilinx EDK Simulation Script** option is available in the **File | Import** menu.
- **Import of Synplicity Synplify Projects**
  1. Service Pack 2 for Active-HDL 7.1 continues the support for the import of synthesis projects

created in the design environment of Synplicity Synplify and Synplify Pro ver. 8.5/8.5.1.
2. The Service Pack also provides a new option that allows importing synthesis projects created by Synplicity Synplify Premier ver. 8.5/8.5.1.
The **Synplicity Synplify(R) Project** option is available in the **File | Import** menu.

- **Import of Xilinx ISE Projects**
  Service Pack 2 for Active-HDL 7.1 allows importing Xilinx projects coming from ISE 8.1i SP3 . Now, users who target their designs to the Xilinx technology can import an ISE project and simulate it in the Active-HDL environment. The **Xilinx ISE 8.1 Import** option is available in the **File | Import** menu.

Accelerated Waveform Viewer

- The Accelerated Waveform Viewer supports the Analog mode for vectors. Now, if your design employs signals of the integer, physical, or floating-point type (e.g. co-simulation results created in a DSP development environment), the Waveform Viewer will allow you to present them in the analog format. This new feature may be used while working with waveforms created during co-simulation, e.g. with use of the built-in Simulink Interface.
- The **Browse by** option has been added to the **Search** menu. Selecting this option opens the submenu containing the same set of browsing options that is available in the pop-up menu activated by the **Select Browse Mode/Object** button located in the bottom-right corner of the Accelerated Waveform window.
- The **Find Signal** tab of the **Find** dialog box has been supplemented with a new feature. The **Whole word** option helps you specify more precisely a range or subset of objects to search for in the simulation database (*.asdb).

Scripts

- The `abort` command has been enabled in the Tcl and Compatibility mode.
- The `-analogmin` and `-analogmax` arguments have been added to the syntax of the `wave/add wave` command. These options allow setting in the command line a range of analog signals displayed in the Standard Waveform Viewer.
- The `workspace` command has been improved. The `open` and `create` arguments have been substituted for `open <workspace>` and `create <workspace>`, which allows explicit specification of the name of the workspace to be opened or created.

Language Assistant

- The Language Assistant has been supplemented with a new set of VHDL simulation templates. The Stimuli Random section provides the ready-to-use templates of the processes generating stimuli equivalent to the Random stimulator. In order to use these templates, you need to declare the **random_pkg** package of the **aldec** system library as well as selected IEEE packages.

Appendix

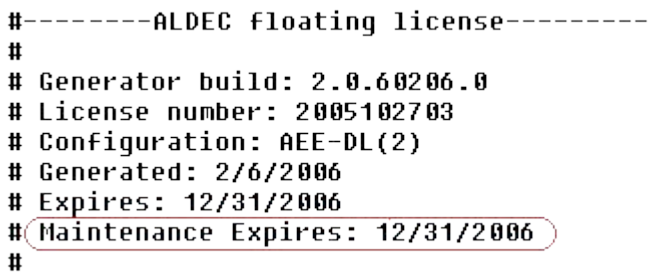
License Maintenance

In order to start working with Active-HDL, you need to have a valid license file. Starting from Active-HDL 7.1, only users who have a valid maintenance contract will be able to apply updates or service packs for a recently released version of Active-HDL. Users who have a maintenance contract that expired should not install any update or service pack as Active-HDL will not run with their existing license and report the following error:

```
Your license file does not support this version of Active-HDL.
```
or
```
(FLEXlm error = -5) You do not have valid license to run Acteve-HDL.
```
or
```
(FLEXlm error = -18) License server system does not support this feature.
Review the License chapter in the Active-HDL Release Notes to check
requirements for software licensing. Contact Aldec for ordering information
- sales@aldec.com
```

In order to find out whether your license allows using a newer version of Active-HDL or applying a service pack for a recently released version of the software, check out your current license file.

In the header section of the *license.dat* file, there is a line that determines the Maintenance Expiration Date. If your maintenance has expired you are not eligible to install a newer build or its update (e.g. service pack). The image below presents the example of the license header with the explicitly specified Maintenance Expiration Date.



You can also find out whether you have a valid maintenance by checking the contents of your license file. It contains a number of lines with a string of information (keyword FEATURE) that define the license features. The license features indicate how long and which version of Active-HDL will run with the license, e.g.:

```
FEATURE ACTIVEHDL_LIC_NUMBER ALDEC 2006.1231 31-dec-2006 uncounted \
```

The argument **2006.1231** (which refers to December 31, 2006) indicates the Maintenance Expiration Date. If your current maintenance contract is valid, you are entitled to the new license with the updated maintenance date. If your maintenance contract has lapsed, you can run versions of Active-HDL released before your maintenance expired or you can renew your maintenance by contacting Aldec at sales@aldec.com.

The table below presents the editions of Active-HDL and the corresponding versions of the maintenance that allow running individual versions of the software.

Active-HDL Version	License Maintenance Date
7.1.SP2 (BUILD 1583.SP2.09, 05/23/2006)	2006.0500 (May 2006)
7.1 SP1 (BUILD 1583.SP1.23, 03/08/2006)	2006.0200 (February 2006)
7.1 (BUILD 1583, 10/11/2005)	2005.0900 (September 2005)

Aldec, Inc.
Corporate Headquarters
2260 Corporate Circle
Henderson, NV 89074

Tel: +1 702 990 4400
Toll Free: +1 800 487 8743
Fax: +1 702 990 4414