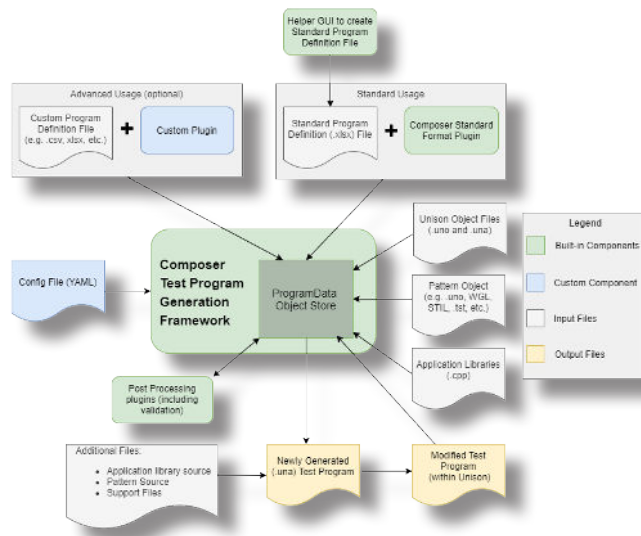


Composer

Unison Test Program Generator Tool



Automotive



Mobility



IoT/IIoV & Optoelectronics



Computing & Network



Industrial & Medical



Consumer

Course Description

This eLearning material introduces the student to Composer – a test program generator tool that can be used to create a working Unison test program using spreadsheet-based test conditions, device data and other input files. On completion of this course, the student will be able to describe the purposes and elements of Composer and recognize how using this tool and its associated framework, can lead to improved test program Time-To-Yield (TTY). This is accomplished by a combination of multimedia presentations and interactive software demonstrations.

Course Outline

- Introduction and Overview
- Features and Framework
- Installation and Licensing
- Help and Examples
- Standard Program Definition File
- Configuration File and Usage
- Importing Application Libraries
- Generating a Test Program
- Incremental Program Updates
- Excel™ Spreadsheet Addin

Course Length

- Self-paced – 8-10 hours depending on skill level

Prerequisites

- Six months test program experience
- Successful completion of Unison Applications Programming course

Recommended

- Familiarity with Linux Operating System
- English - written and spoken

- Reduce Time-To-Yield (TTY) program development
- Standard program definition file format provided
- Customizable and extendable framework
- Incremental program updates supported
- Generate Unison objects and patterns
- Python-based plugin modules

Composer

Unison Test Program Generator Tool

Course Modules

1 - Introduction and Overview

On completion of this module the student will be able to:

- Describe the purpose and elements of Composer
- Recognize how tool can lead to improved Time-To-Yield (TTY)

2 - Features and Framework

On completion of this module the student will be able to:

- Recognize the main features of Composer
- Recognize its customizable and extendable framework

3 - Installation and Licensing

On completion of this module the student will be able to:

- Recognize how to install Composer on a Linux workstation
- Check that the necessary license is installed

4 - Help and Examples

On completion of this module the student will be able to:

- Recognize how to launch and access Composer help using several methods
- Install and review some built-in Composer examples

5 - Standard Program Definition File (SPDF)

On completion of the module the student will be able to:

- Describe the purpose for the SPDF spreadsheet file and its pre-defined formats
- Be able to summarize its sections, both visible and hidden
- Summarize the usage of the Global Info, Device Info, Test Cell Info, Test Block, Test List, Time Sets, Levels and Device Setup sheets

6 - Configuration File and Usage

On completion of this module the student will be able to:

- Describe the overall purpose of the config file in generating a test program using Composer

7 - Importing Application Libraries

On completion of this module the student will be able to:

- Summarize the different techniques used by Composer for importing Application Libraries

(Note – focus is on using the Standard Program Definition file and Composer Standard Format plugin)

8 - Generating a Test Program

On completion of this module the student will be able to:

- Recognize the files involved in generating a device test program using ComposerStdFormat
- Summarize the options available when using this script
- Complete a Standard Program Definition (SPDF) file
- Execute the generated program

9 - Incremental Program Updates

On completion of this module the student will be able to:

- Summarize and use the Composer *Incremental Update* feature

10 - Microsoft Excel™ Spreadsheet Addin

On completion of this module the student will be able to:

- Describe the Composer Standard Program Definition File (SPDF) Excel™ Add-in
- Recognize the features and benefits of using this Excel™ Add-in

- Reduce Time-To-Yield (TTY) program development
- Standard program definition file format provided
- Customizable and extendable framework

- Incremental program updates supported
- Generate Unison objects and patterns
- Python-based plugin modules

Composer

Unison Test Program Generator Tool

Who Should Attend

- Test program development and support engineers

Course Viewing Requirements

To view the course, you must have:

- Browser supporting HTML5
- Audio-listening capabilities (such as a headset or speakers)
- Connection speed of at least 600 kbps

Course Cost

- Access is free of charge for all Cohu Semiconductor Tester Customers