

EntryPoints

ASAP

- INSTALL
 - Sets up the hardware, reads the loadboard calibration values stored either in the EEPROM located on the loadboard or in a file, and loads all the patterns used in the test program. If the installation process is not successful, the next step (the INIT flow) cannot be run. Any errors must be fixed before the INIT flow can be run.
- INIT
 - The INIT button initializes the hardware and all system variables, clears all counters, and loads the system calibration values into the hardware. If the installation is not successful (does not pass), the INIT button can not be pressed.
 - START_LOT (segment – aka flownode – not an entry point)
 - Include a START_LOT segment in the INIT flow to enable Start Lot records to be written to the datalog stream.
 - START_WAFER (segment – aka flownode – not an entry point)
 - Include a START_WAFER segment in the INIT flow to enable Start Wafer records to be written to the datalog stream.
- BEGIN
 - Starts execution of the test program. The PASS and FAIL indicators display the status of the test execution.
- RESET
 - Resets the test program any time the push-button is pressed. The purpose of this button is to provide a means to interrupt an active test flow, and once interrupted, returns the system to the default state. When there is no active test, there is nothing to interrupt and no action is taken. However, when a test or test flow is executing, the RESET terminates the testing and executes the default RESET segment. The RESET segment sets the device to bin 0 and applies the user defined power down levels block if defined. Afterwards, the system power down function is executed.
- END_WAFER
 - Executes the END_WAFER flow, which includes the PRINT_WAFER system function and any other user-specific end-of-wafer actions.
 - PRINT_WAFER
 - System function. Included in the END_WAFER entry point flow.
- END_LOT
 - Executes the END_WAFER flow, which includes the PRINT_WAFER system function and any other user-specific end-of-wafer actions.
 - PRINT_LOT
 - System function. Included in the END_LOT entry point flow.

Stylus

- Main
- Start
- Finish
- Load
- Unload

Envision

- OnLoad
 - Executed when the test program is loaded, and generally used to perform such functions as Adapterboard diagnostics. It takes place after all objects are loaded and as the last step in the loading process.
- OnStart
 - The OnStart subflow executes immediately after the OnInitFlow when the TEST/Start button is clicked. This initiates the execution of the test program. The OnPowerDown subflow executes following the completion of the OnStart subflow.
- OnRestart
 - The OnRestart subflow executes immediately after the OnInitFlow subflow when the RETEST button is clicked. This initiates the execution of the test program for retest conditions. The OnPowerDown subflow executes following the completion of the OnRestart subflow.
- OnUnload
 - Executed when the test program is released from the shared memory.
- OnReset
 - Executed immediately after the RESET button is clicked. The default behavior is to bin the device to evResetBin. This entry point is also called when a “halt” command is encountered in the Cadence portion of the program. After the OnReset subflow is executed all clients are notified that the test has been completed. The device will have a bin result of evResetBin. The OnPowerDown subflow will then execute.
- OnRunTimeError
 - Executed immediately after a run time error occurs, generally a test program error, such as a parameter out of range. The default behavior is to bin the device to evRunTimeErrorBin. After the OnRunTimeError subflow is executed, all clients are notified that the test completed. The device will have a bin result of evRunTimeErrorBin. The OnPowerDown subflow will then execute.
- OnFault
 - Executed immediately after a Cadence fault is detected. The default behavior is to call the on_fault cadence routine. After the OnFault subflow is executed all clients will be notified that the test has been completed. The device will have an invalid bin result of 0. The OnPowerDown subflow will then execute.

- OnPowerDown
 - Executed at the end of the OnStart execution, independent of how it terminated that subflow. This entry point is also called when a “stop” command is encountered in the Cadence portion of the program.
- OnBeginLot
 - When controlled by an External interface, if the “Begin of Lot” signal is generated, then this subflow will be executed.
- OnEndLot
 - When controlled by an External interface, if the “End of Lot” signal is generated, then this subflow will be executed.
- OnGpibSrq
 - When communicating with an external instrument using the GPIB bus, if an instrument generates an interrupt, then this subflow is executed.
- OnInitFlow
 - This subflow is used to setup conditions which are required no matter where the TEST button is clicked. External interface, OperatorTool, Flow tool, C++ TestTool, or TestTool. It is executed prior to the calling tool sequence (the OnStart subflow, if from the External interface or OperatorTool; the displayed subflow, if from the FlowTool; and the displayed test, if from the C++ TestTool or TestTool).
- OnAfterBin
 - This Subflow is invoked when all bins have been resolved, in a Multisite application.
- OnWaferStart
 - When controlled by an External interface, if the “Wafer Start” signal is generated, then this subflow will be executed.
- OnWaferEnd
 - When controlled by an External interface, if the “End of Wafer” signal is generated, then this subflow will be executed.
- OnBinOverflow
 - If a Bin object reaches the Max Count value, then this subflow will be executed, allowing the user to take the desired action.
- UserCal
 - Executed the first time the TEST button is clicked, after the program is loaded, and it is intended to perform program focused adapter board calibration. This flow is also called by enVision when a Focus Calibration needs to take place. This is where all Focus Calibration Test Methods are placed. The user can place as many Focus Calibration methods depending on the different conditions the user wishes to calibrate. The user can also place other Test Methods here; for instance, the user may want an Opens and Shorts test before Focus Calibration, to be sure the Adapterboard is properly connected and without a device installed.
- OnUsr0-OnUsr9
 - Invoked by an External Interface (or mapped Function key) signal to perform a user defined special sequence. Ten different entry points are available.

OTPL

- CfgPLLoadFlow
 - This Flow is executed on the System Controller when a Test Plan is loaded onto one or more Site Controllers. It is executed prior to the actual loading of the Test Plan on any Site Controller. This flow allows the Test Plan developer to define actions that should originate from the System Controller. Such actions include broadcast load of Pattern files, Calibration actions, etc.
- SiteLoadFlow
 - This Flow is executed on the Site Controller after a Test Plan has been loaded onto the Site Controller and initialized. This allows any Site Controller specific initialization to occur.
- InitFlow
 - The user invokes this flow, as required. The typical scenario is to perform some common or routine initialization necessary to bring the TestPlan back into a known state after operation and/or modification.
- LotStartFlow, LotEndFlow
 - These Flows execute on the Site Controllers when the Test Plan Server is notified of a start of a new lot. This is typically used in production environments to annotate datalog streams with lot-specific information.
- DUTChangeFlow
 - This Flow executes on the Site Controller when its DUT information changes. Again, this is typically used in production environments to update datalog streams.
- TestPlanStartFlow, TestPlanEndFlow
 - These Flows execute on the Site Controller when the Test Plan Server is instructed to start executing the current Test Flow and when that flow finishes its execution.
- TestStartFlow, TestEndFlow
 - These Flows execute on the Site Controller when the Test Flow is starting to run a new Test and when that Test finishes its execution.
- MainFlow
 - This Flow is the main Flow object executed when the Test Plan Server receives an “Execute Test Plan” message.

Image

- No predefined entry points. User can define any number of SEQUENCER() functions, and call them from test program main(). Execution of these functions within main() can be controlled conditionally any way the user desires. Loading the job (program) includes loading all vectors and initializes everything (load board calibrations, for instance). Unlike some other languages, the order in which these actions occur is not under user control.

IG900

- .t file (ATL programming only)
 - tt_job_init()
 - job_main()
- Uses flowtable file to specify the order of the flow, and what happens when a particular flow element or test element fails. As with Image, loading a job (program) causes many things (vector loads, loadboard calcs, etc.) to occur as part of the load process. Unlike some other languages, the order in which these actions occur is not under user control.

IGXL:

XTOS: Need to get information from Daniel Fan, but believe it to be similar to ASAP.

Diamond: Need to get information from Daniel Fan.

SmarTest:

- Only device test flow is represented in test flow file. All other events are handled either in the application model (user-customizable), or via the system software when a testprogram file is loaded. Vector load is an example of the latter – the vectors are loaded the first time a flow is run after a test program file is loaded. On subsequent flow executions, the vectors are not (re)loaded (since they are already loaded).