1. How did you compile spi\_test\_lib\_pkg? Packages must be compiled before they can be imported.
2. For cadence, incdir is prefixed by “-“ , for vcs & modelsim it’s prefixed by “+”
3. Interface is compiled separately instead of adding it to a package since it’s not allowed to be a part of a package or a module
4. Always include “incdir” before each and every directory, both in cadence and in vcs,questasim
5. Find out the stuff to be randomized and include it in seq\_item under uvm\_object\_utils\_begin/end
6. Mistake made :: instead of using uvm\_sequence\_item, used uvm\_object !!!!
7. Forgot how to create random items under uvm\_object\_utils\_begin/end.
   1. Order of creation of sequence\_item is :
      1. Declare the class as txn extends “uvm\_sequence\_item”
      2. Declare the variables under the class before registering them
      3. Register the variables under “uvm\_object\_utils\_begin/end”
      4. Use “uvm\_field\_int” for bits as well as integers. Others max utilized are:
         1. “uvm\_field\_enum”
         2. “uvm\_field\_real”
         3. Uvm\_field\_string
8. Uvm\_sequence\_item doesn’t involve usage of build\_phase or connect\_phase as it is purely a “data” package only
9. Missed out on “uvm\_field\_array\_int” and used “uvm\_field\_int” for usage of dynamic array which is to be randomized for data packet generation
10. Commented out the rest of the packages and instead currently using only Yapp\_pkg & common\_pkg alone for first round of data\_generation
11. Only “new” function is non-virtual, rest of the functions seem to be virtual
12. Mistake made : - instead of build\_phase , added new in its place
13. It is illegal to import a package inside a class. It’s possible to import a package only inside a module
14. Can’t include an “interface” within a package as “Interface” is a design element
15. Task & functions don’t need “begin” statement when they’re used in SV unlike Verilog
16. Warning: “Treating stand-alone use of function 'randomize' as an implicit VOID cast.”
    1. Ans : Randomize function returns a value of ‘1’ if successful and ‘0’ if unsuccessful. You should always check the return value.
17. Setting a dynamic array constraint:
    1. Ans :   
       constraint l1\_array {

payload.size==length;

}  
where, payload is “uvm\_field\_array\_int” and length is “uvm\_field\_int”

1. Error : \*\* Fatal: (SIGSEGV) Bad handle or reference.
   1. Fatal error in Function yapp\_pkg/yapp\_agent::connect\_phase at ../agents/yapp/yapp\_agent.sv
      1. Ans : Made a mistake when creating build by creating uvm\_component parent=”null”, instead it should’ve been parent=”this” so that there is an agent handle which gets created.
2. Only UVM\_DRIVER and UVM\_SEQUENCER are parameterized with uvm\_sequence\_item. UVM\_monitor isn’t. Instead it gets the sequence item from analysis port.
3. Since UVM\_agent itself contains UVM\_active if statement, yet to figure out why agent\_cfg is used.
4. Always use null in top level module when virtual interface is set using uvm\_config\_db. When it comes to context,
5. Uvm\_config\_db::set() seems to be a function without return, whereas uvm\_config\_db::get() is a function which returns a bit value
6. Difference between run\_phase & main\_phase :
   1. Both are parallel and occur simultaneously.
   2. Main\_phase contains the following phase[[2]](http://forums.accellera.org/topic/1820-the-difference-between-run_phase-and-main_phase-in-uvm_component/):
      1. Pre-reset-phase -> Reset-phase -> Post-reset\_phase
      2. Pre-config-phase -> config-phase -> post-config-phase
      3. Pre-main-phase -> main-phase -> post-main-phase
      4. Pre-shutdown-phase -> shutdown -> post-shutdown-phase
7. Running the sequence needs the following :
   1. Creation of sequence’s object in testcase[[3](https://www.verificationguide.com/p/uvm-sequence.html?m=1)]
   2. Usage of “start” function inside run\_phase of test which points to the sequencer
      1. Eg:- seq.start(seqr\_name);
   3. `uvm\_do and the steps that it covers are covered in [[3](https://www.verificationguide.com/p/uvm-sequence.html?m=1)]

Tool based debugs:

Questasim:

1. Didn’t create vlib work first under sim directory
2. Didn’t map the work directory using “vmap” command, which copies modelsim.ini from elsewhere(Tool’s home) to the work directory
3. Didn’t know to use “-l <file\_name>” for compile log
4. Error: ../agents/yapp/yapp\_seq\_item.sv(2): near "uvm\_sequence\_item": syntax error, unexpected IDENTIFIER
   1. Your pakmx\_transaction class should be inside a package that imports uvm\_pkg.
   2. Ans : Imported uvm\_pkg::\* ; under the package to avoid this error
5. \*\* Error: ../agents/yapp/yapp\_seq\_item.sv(9): (vlog-2163) Macro `uvm\_object\_utils\_begin is undefined.
   1. Ans : Imported the `include “uvm\_macros.svh” to avoid this error.
6. \*\* Error: ../agents/yapp/yapp\_seq\_item.sv(21): Unresolved reference to 'build\_phase'
   1. Ans : uvm\_sequence\_item doesn’t involve any build\_phase or connect\_phase
7. \*\* at ../agents/yapp/yapp\_seq\_item.sv(2): Typedef 'yapp\_seq\_item' multiply defined.
   1. Ans : uvm\_sequence\_item declared both in package and in yapp\_env.sv
8. tools/mentor/qsim10.3b/questasim/bin/../linux/vish: error while loading shared libraries: libXft.so.2: cannot open shared object file: No such file or directory
   1. Ans : Forgot to include –L ${PRJ\_HOME}/sim/work as library before doing simulation
   2. Forgot to add work.top as the module to be simulated and instead only used “vsim –L work top” as the vsim command
9. \*\* Warning: (vlog-2240) : Treating stand-alone use of function 'randomize' as an implicit VOID cast.
   1. Ans: Check “16” above.
10. When it came to using Questasim, missed out on UVM\_DPI related command to vlog before vsim was run:
    1. +define+UVM\_NO\_DPI
    2. Instead of using the above, uvm\_dpi.dll[[1](https://blogs.mentor.com/verificationhorizons/blog/2011/03/08/using-the-uvm-10-release-with-questa/)] has to be added
11. UVM\_FATAL: [THISPARENT] uvm\_component.svh(1729)
    1. cannot set the parent of a component to itself
    2. Ans: Made sure that in build\_phase, uvm\_component parent!= this in uvm\_agent to ensure that this error is not coming.

IRUN:

1. ncvlog: \*W,FUNTSK: function called as a task without void'().
   1. Ans: Dave\_59:
      1. if you want to call a function that normally returns a value and ignore it, you can use an explicit cast to void  
         “**void**'( my function\_with\_return\_value() )”  
         The reason for the warning is that many people either forget to check the return value of a function that they should have (like randomize), or they forget that functions declared with no return type have an implict 1-bit return type
2. /usr/include/gnu/stubs.h:7:27: fatal error: gnu/stubs-32.h: No such file or directory

# include <gnu/stubs-32.h>

^

compilation terminated.

ncelab: \*W,DPIEXP: DPI export function in \_sv\_export.so not available.

gcc: error: ./INCA\_libs/worklib/hbus\_if/sv/\_sv\_export.o: No such file or directory

ncelab: \*W,DPIEXP: DPI export function in \_sv\_export.so not available.

* 1. Ans: Broken OS issue, this is a
     1. Should use “-64bit” for Irun if it’s available.

References:

1. <https://blogs.mentor.com/verificationhorizons/blog/2011/03/08/using-the-uvm-10-release-with-questa/>
2. <http://forums.accellera.org/topic/1820-the-difference-between-run_phase-and-main_phase-in-uvm_component/>
3. <https://www.verificationguide.com/p/uvm-sequence.html?m=1>