Build environment

Duc Duong

January 2, 2014

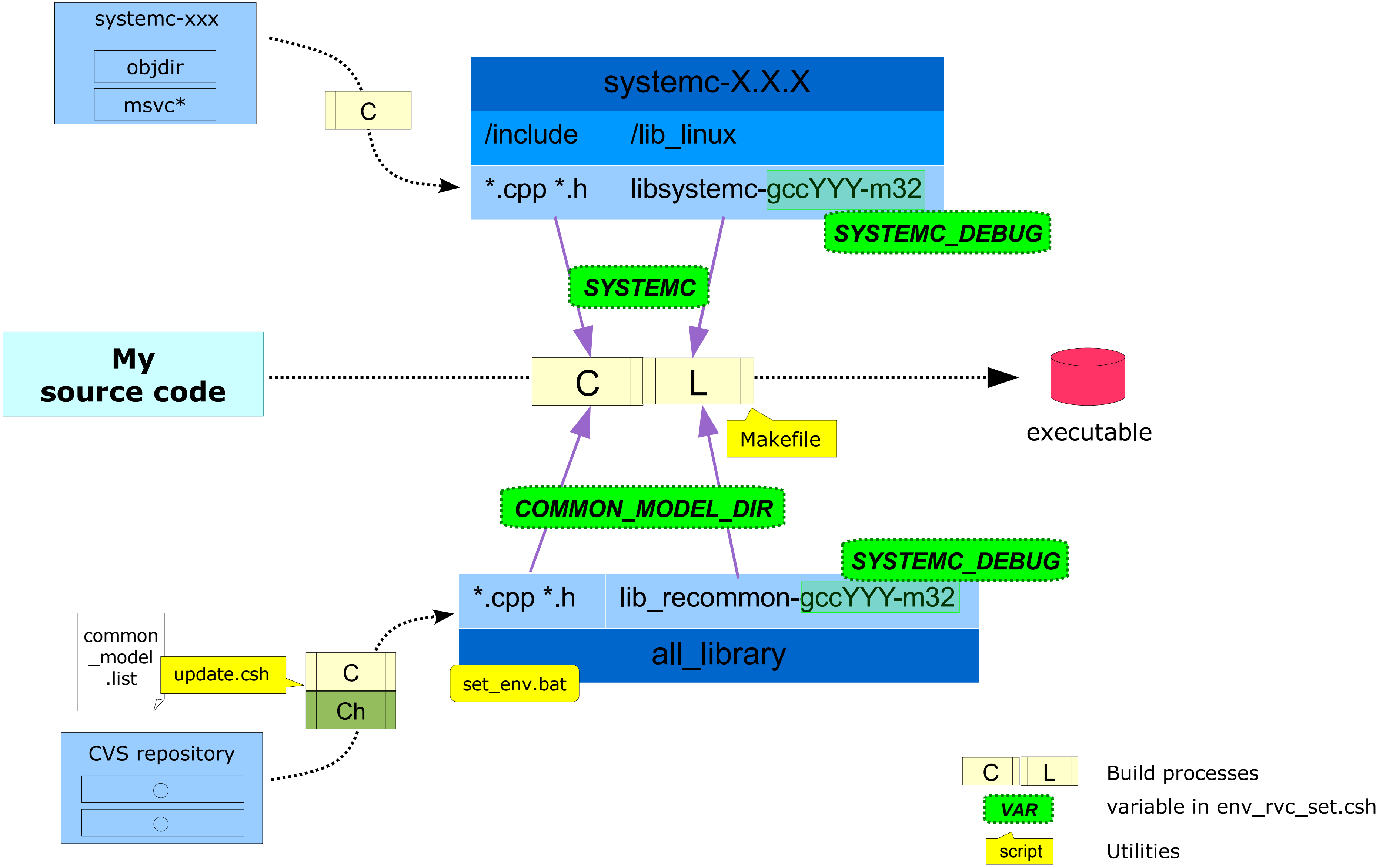
Confidential

Outline

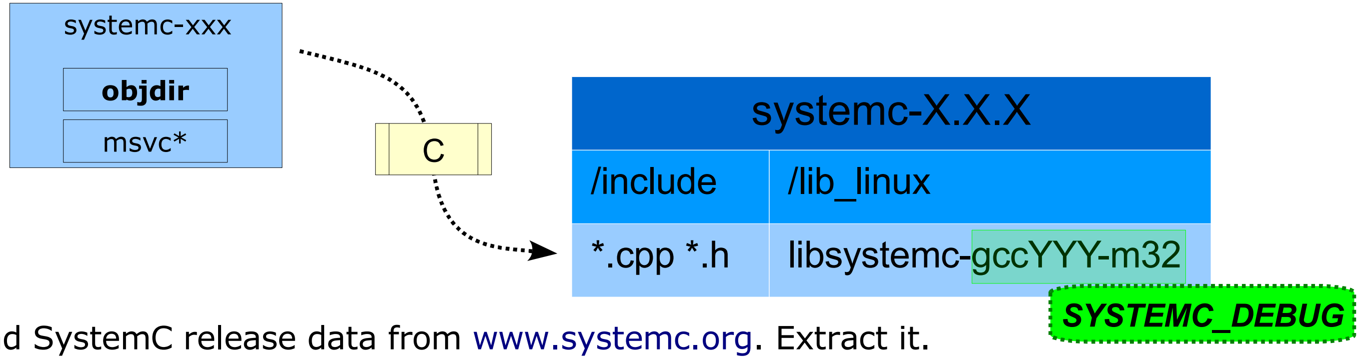
- Linux
  - Global image
  - Build SystemC library
  - Build 'all_library'
- Windows
  - Global image
  - Build SystemC library
  - Build 'all_library'

```
[Dir.SC] = /shsv/sld/ipp/Work/DUCDUONG/SystemC_build
[Dir.AL] = /shsv/sld/ipp/common/04_lib/99_latest_common_src/Script/all_library
```

Global image (Linux)



Build SystemC library (Linux)



systemc-X.X.X	
/include	/lib_linux
*.cpp *.h	libsystemc-gccYYY-m32

1. Download SystemC release data from www.systemc.org. Extract it.
2. Installation guide is stored in <ROOT>/INSTALL file. Step-by-step:
    1. Set up environmental variables: setenv CXX g++, set PATH for g++.
       (refer to *setup_32.csh* or *setup_64.csh* in [Dir.SC])
    2. Build:
        a. Create temporary directory:
        ```
        >> mkdir objdir
        >> cd objdir
        ```
        b. Configure:
        ```
        >> ../configure [--prefix=<installed directory>] [<flags>=<value>]
        ```
        Common flags: CFLAGS, CXXFLAGS, CPPFLAGS and LDFLAGS

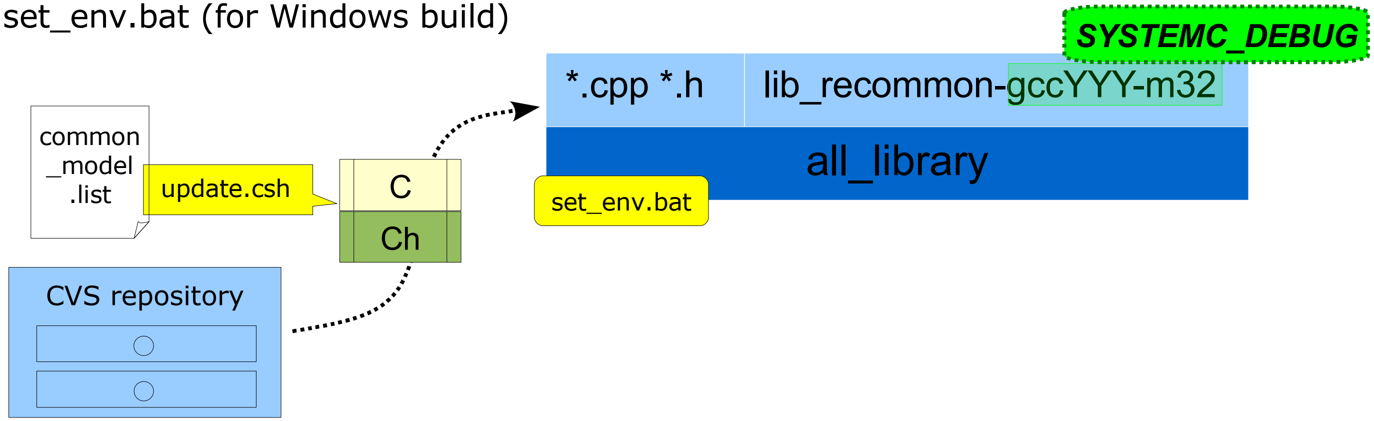
        c. Compile:
        ```
        >> gmake
        ```
        d. Install:
        ```
        >> gmake install
        ```
        e. Check (run examples with installed SystemC library for confirmation)
        ```
        >> gmake check
        ```
    (refer to *build.csh* in [Dir.SC])
3. Rename the output: lib_linux[64]/***libsystemc.a*** to lib_linux/***libsystemc-*****$(SYSTEMC_DEBUG)**

Build all_library (Linux)

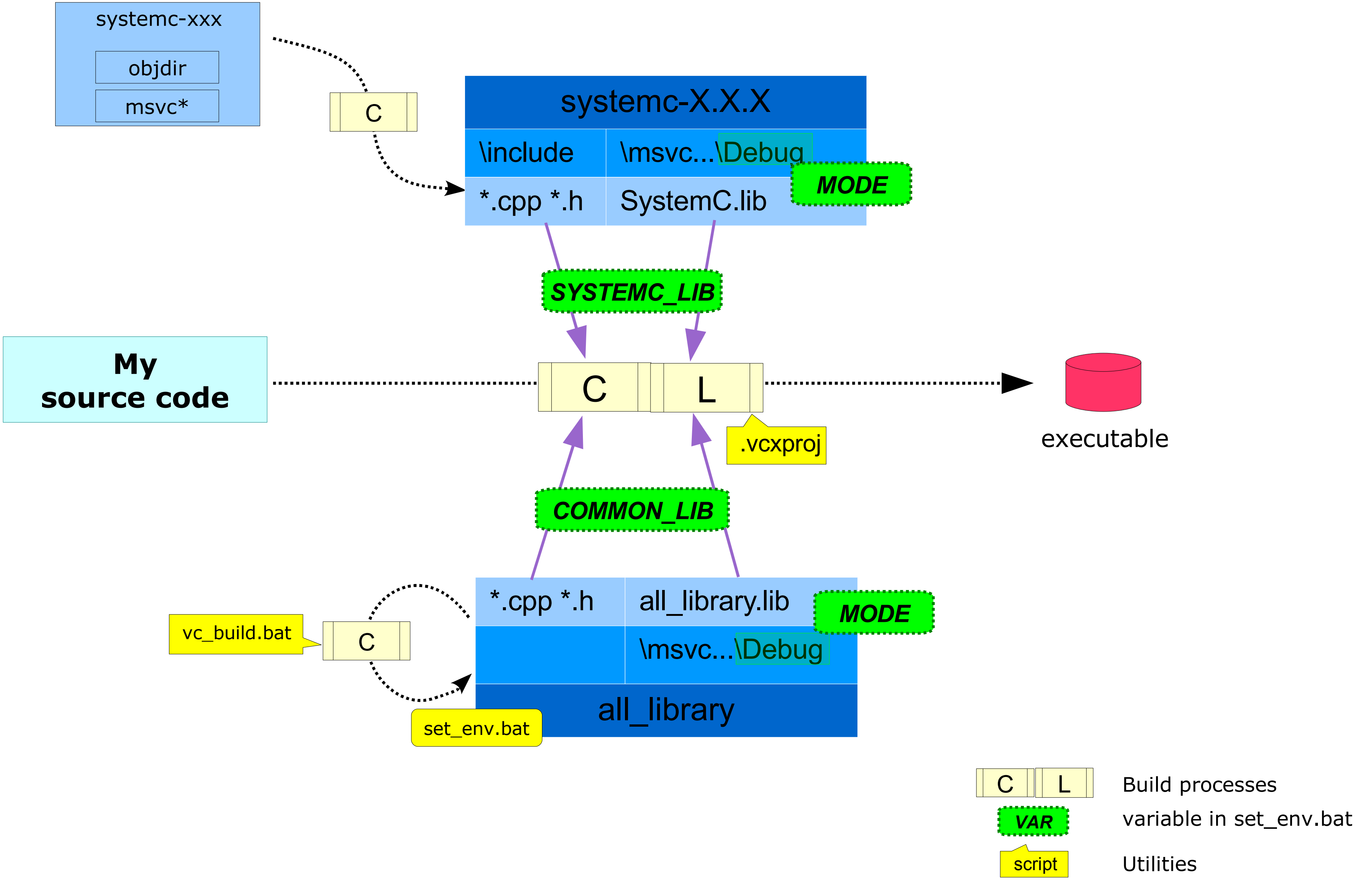
1. Get the all_library build package with correct version (specified in Common REQ). Latest version at [Dir.AL].
2. Prepare common_model.list file.
    1. List of common models with CVS version/CVS tag (e.g. “re_register v2013_05_28”).
    2. List of additional include directories (AdditionalIncludeDirectories).
    3. List of preprocessor definitions (PreprocessorDefinitions, e.g. REGIF_SC_REPORT).
3. Set up environmental variables:
    - Set PATH for g++, paths to SystemC, TLM.
    - Set SYSTEMC_DEBUG.

    (refer to env_rvc_set_lite.csh in [Dir.AL]/env_set)
4. Build:
    1. Checkout from CVS repository:
        ```
        >> update.csh common_model.list checkout
        ```
    2. Build:
        ```
        >> update.csh common_model.list build
        ```

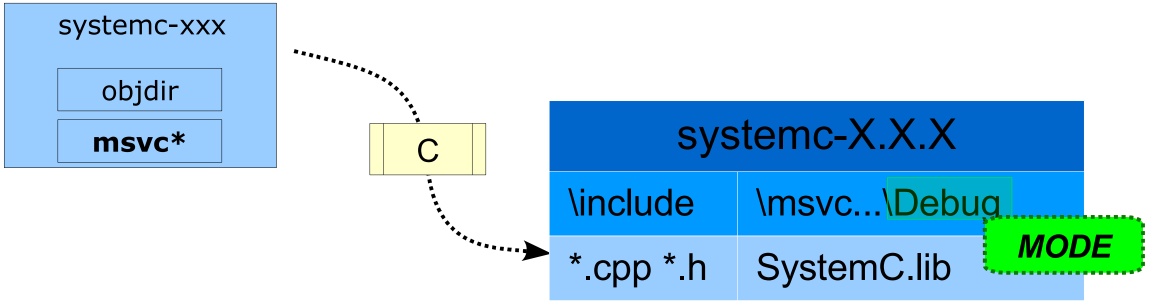
    For both checkout and build:
    ```
    >> update.csh common_model.list
    ```
5. Output:
    1. Checked-out source files and ***lib_recommon_*****$(SYSTEMC_DEBUG)*****.a***
    2. set_env.bat (for Windows build)



Global image (Windows)



Build SystemC library (Windows)



systemc-xxx

objdir

msvc*

C

systemc-X.X.X

\include

\msvc...\Debug

*.cpp *.h

SystemC.lib

MODE

1. Download SystemC release data from www.systemc.org. Extract it on Windows PC.
2. Clone the already existed msvc* (e.g. msvc80) directory and rename to your Visual C++ version.
    - msvc90: Visual C++ 2008
    - msvc100: Visual C++ 2010
3. Open “msvc*\SystemC\***SystemC.sln***” solution file in Visual C++ GUI.
4. Set up 4 configurations: Project > Properties > C/C++ > Code Generation > Runtime Library
    1. Debug: Multi-threaded Debug DLL (/MDd)
    2. Debug_MT: Multi-threaded Debug (/MTd)
    3. Release: Multi-threaded DLL (/MD)
    4. Release_MT: Multi-threaded (/MT)
5. Build the solution for all configurations one by one.
6. Output: ***SystemC.lib*** in Debug, Debug_MT, Release, Release_MT directories.
7. Put built SystemC directory at C:\Users\%USERNAME%\Documents\Visual Studio ...\Projects

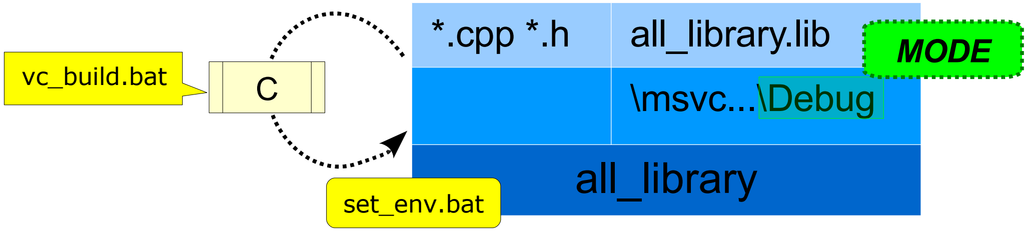
Build all_library (Windows)

1. Copy the output of “all_library” built on Linux (with generated set_env.bat) to Windows PC.
2. Add environmental variable setting to set_env.bat: set paths to SystemC, TLM.
(refer to set_env_lite.bat in [Dir.AL]/env_set)
3. Build all_library VC++ project:

```
> vc_build.bat <msvc90|msvc100>
```

(one execution for all 4 configurations)

4. Output: ***all_library.lib*** in Debug, Debug_MT, Release, Release_MT directories.
5. Put built all_library directory at C:\Users\%USERNAME%\Documents\Visual Studio ...\Projects\R-IP\Common



Appendix: Linux vs. Windows

Linux	Windows	Meaning	Example
g++	Visual C++	Compiler	-
Makefile	VC++ project file (*.vcproj, *.vcxproj)	Build properties file	-
-I	<AdditionalIncludeDirectories>	Directories of included source files (for compiler)	SYSTEMC, TLM
-L	<AdditionalLibraryDirectories>	Directories of object/lib files (for linker)	obj/linux, msvc\Debug
-l	<AdditionalDependencies>	Object/lif filename	re_common-gcc412-m32 all_library.lib
-D	<ProcessorDefinitions>	Preprocessor definition	IS_MODELED_ENDIAN_BIG
${var_name}	%VAR_NAME%	Environment variable in script	${SYSTEMC_DEBUG} %USERNAME%

Renesas Design Vietnam Co., Ltd.