

RhapsodyGuru

Cygwin for IBM Rhapsody Installation Guide

Contents

Rhapsody Cygwin support and Installation.....	3
Download and Install Cygwin.....	3
Testing the Cygwin installation.....	8
Compiling Rhapsody Framework with Cygwin.....	8
Compiling Rhapsody Webservices with Cygwin.....	12

Rhapsody Cygwin support and Installation

Rhapsody supports Cygwin out of the box and expects the compiler to be installed in `C:\Cygwin`. If you want to install Cygwin in a different folder then its recommended to install the Cygwin compiler before you install IBM® Rational® Rhapsody®. Then the installation wizard can detect and integrate with the compiler.


If you installed the Cygwin compiler after you installed Rational Rhapsody, open `Cygwinmake.bat` in `<Rational Rhapsody installation path>\Share\etc` and change the **Path** command to point to your Cygwin installation. For example: `PATH=C:\cygwin\bin;%PATH%`


Download and Install Cygwin

This document guides you through all steps to install a minimalistic cygwin environment for Rhapsody.

1. To install or update Cygwin from <https://www.cygwin.com> select one of the two options

- Open <https://cygwin.com/setup-x86.exe> in your browser any time you want to update or install a Cygwin 32bit package.
- Open https://www.cygwin.com/setup-x86_64.exe in your browser any time you want to update or install a Cygwin 64bit package .

 **Tip:** Which bit version shall I use ? As a rule we could say, also to simplify the upcoming installation steps: Download Cygwin32 bit if you have Rhapsody 32bit installed, download Cygwin 64bit, if you have Rhapsody 64 bit version. Otherwise, the benefit of **Cygwin64** is, it allows for maximal use of RAM on 64-bit machines, which probably doesn't matter much unless you're doing some very memory-intensive work.

 **Warning:** Once you commit to 32-bit or 64-bit (and assuming you don't install both), it's recommended to stick to this choice when installing anything else that will interact with Cygwin

The **Cygwin Setup** Dialog Window will appear:

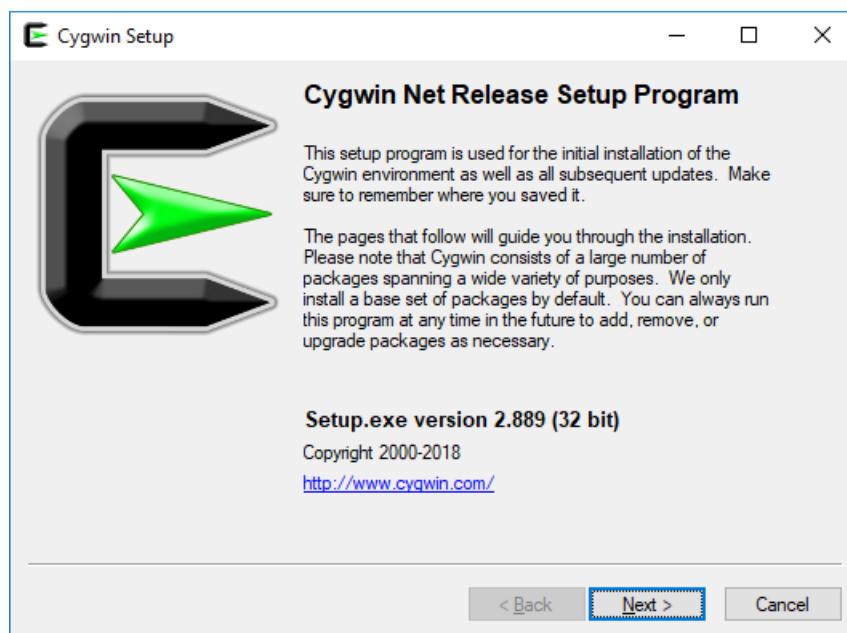
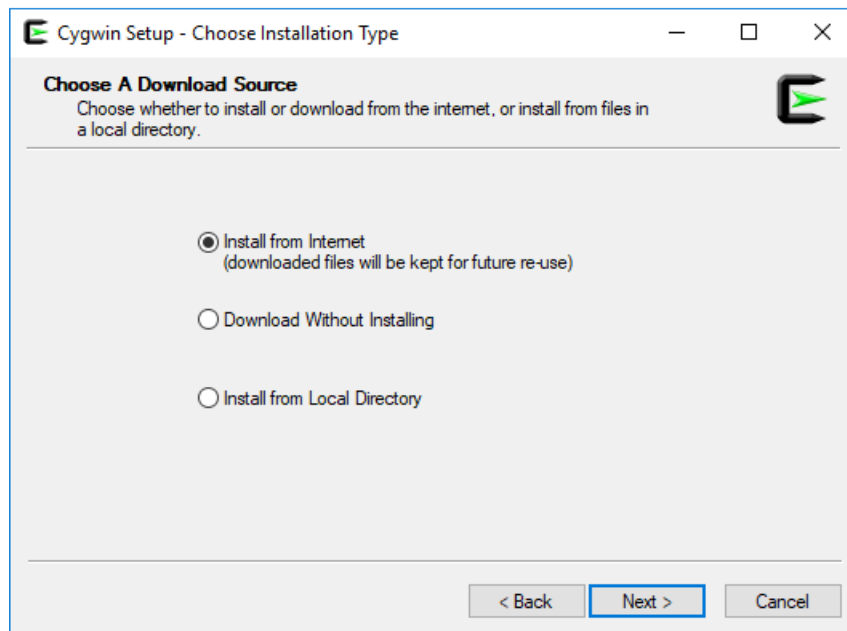


Figure 1: Cygwin Setup Window

2. Click **Next>**
3. In the **Choose Installation Type** Window select *Install from Internet* then press **Next>**



4. In **Choose Installation Directory** select *Root Directory* type `c:\cygwin` then press **Next|>**

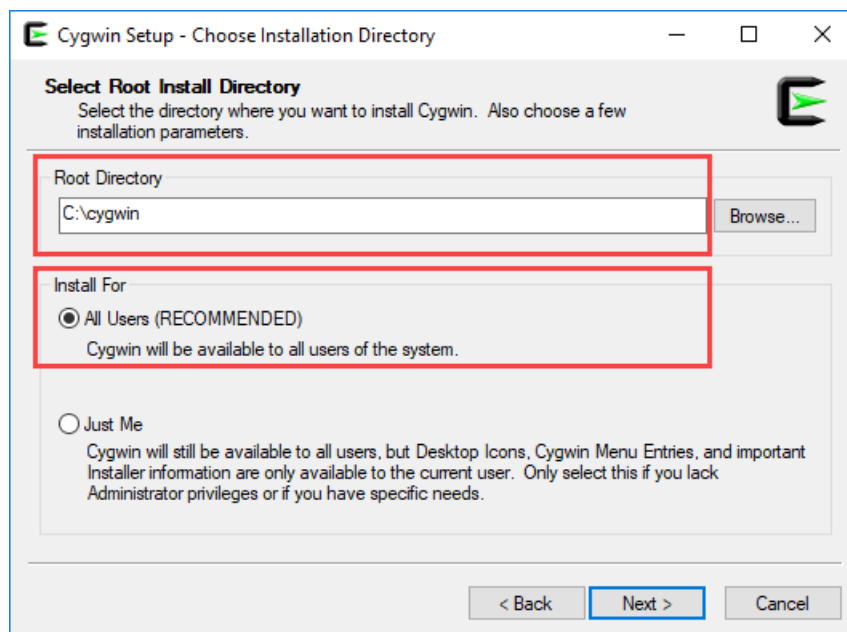


Figure 2: Choose Installation Directory



Note: Per default Rhapsody expects *Cygwin* to be located at this location. If you chose a different one, you can point in Rhapsody to a different one during Rhapsody Installation.

5. In **Select Local Package Directory** select the location where you want to store the Installations Files for later reuse

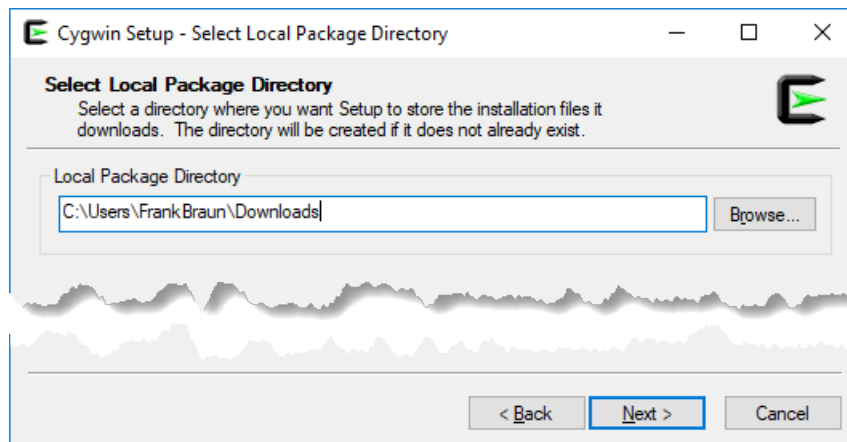


Figure 3: Select Local Package Directory

6. In **Select Internet Connection** window select how you want to connect to the internet (in general *Direct Connection*) then press **Next**>

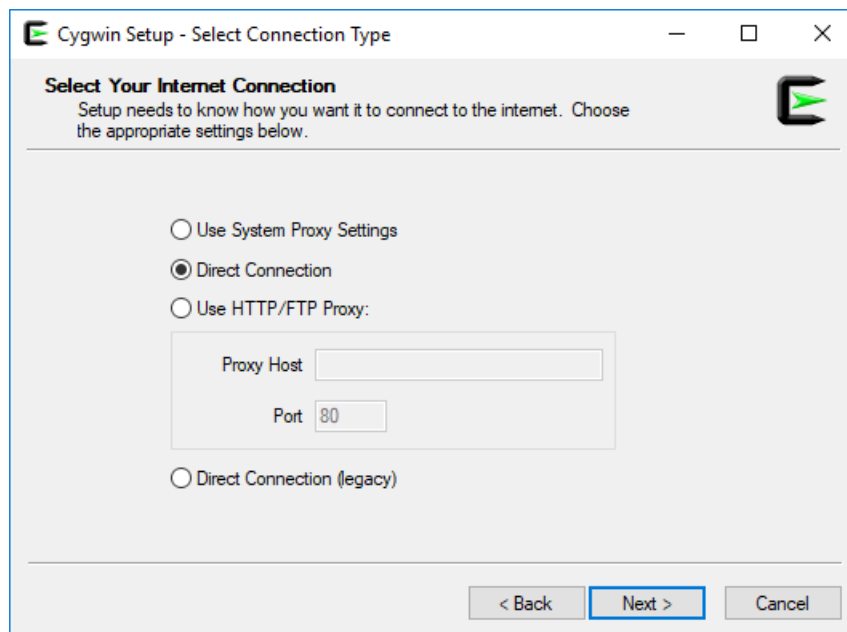


Figure 4: Select Internet Connection Type

7. In **Choose Downloadsite** select a site close to you geography then click **Next**>

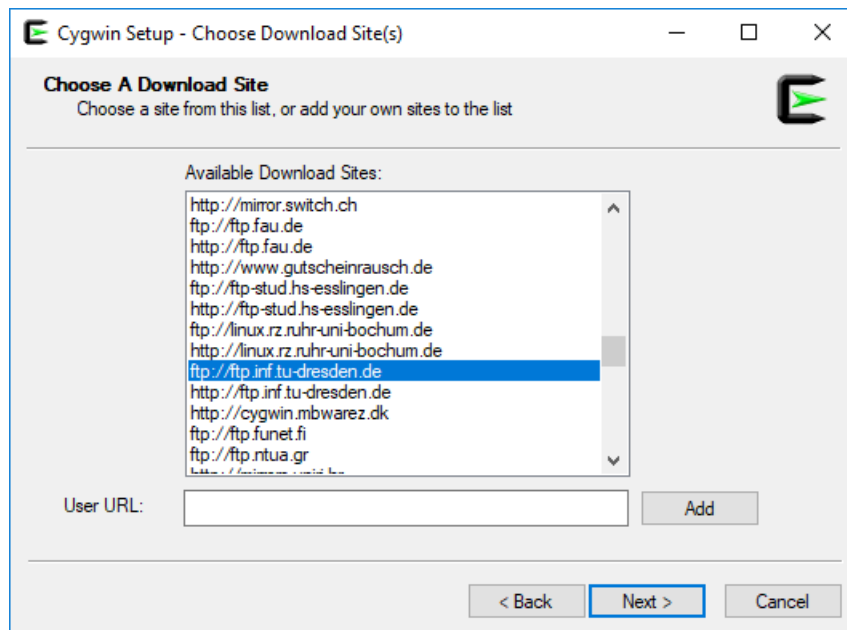
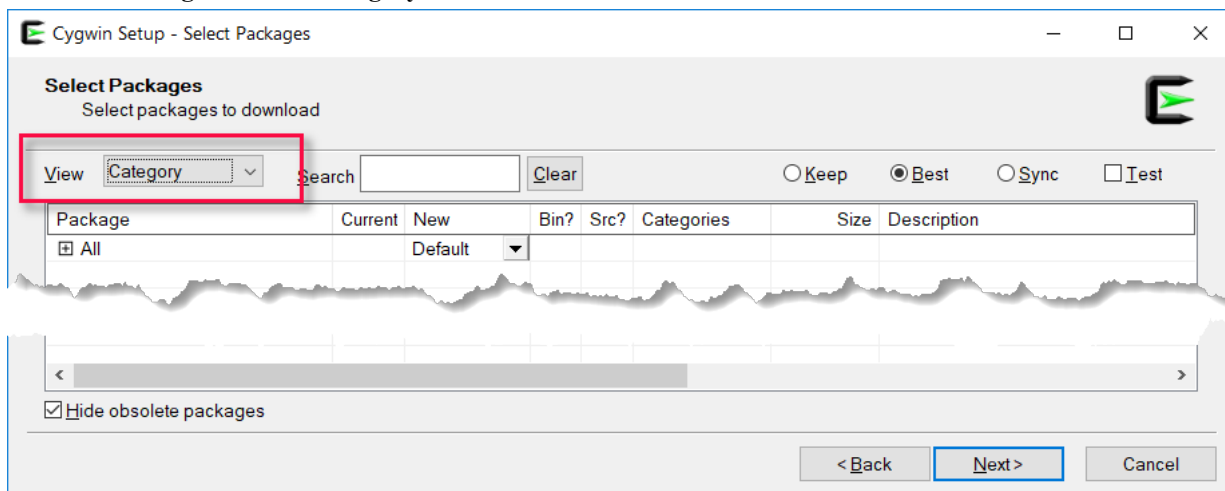
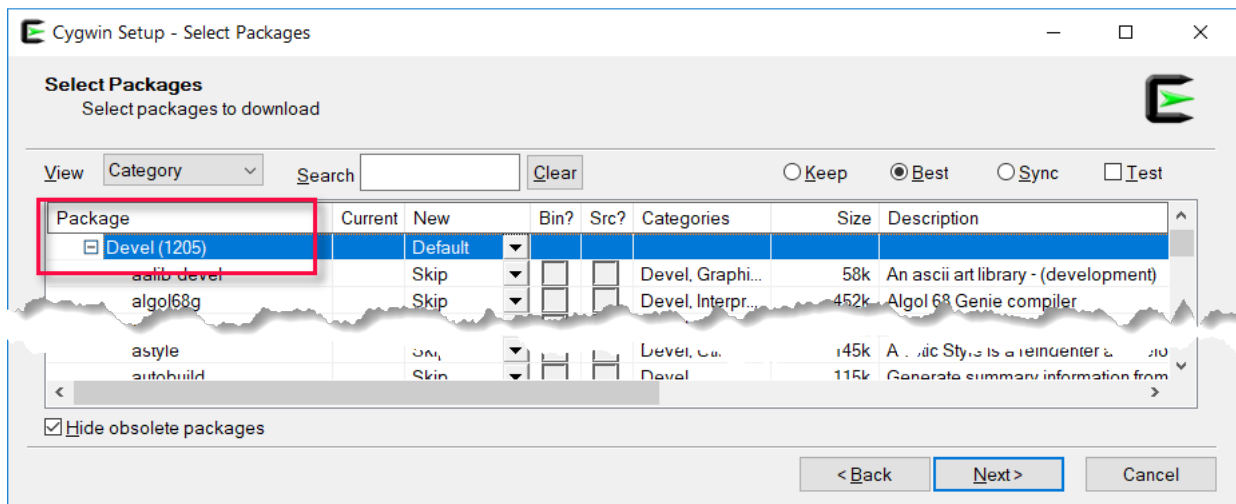



Figure 5: Site location

8. In **Select Packages** click on **Category View**.



9. Underneath **Category** expand **Devel** Category.



10. Click on the  icon to select the most recent of following packages:

```
gdb: The GNU Debugger
gcc-g++: GNU Compiler Collection
make: The GNU version of 'make' Utility
```

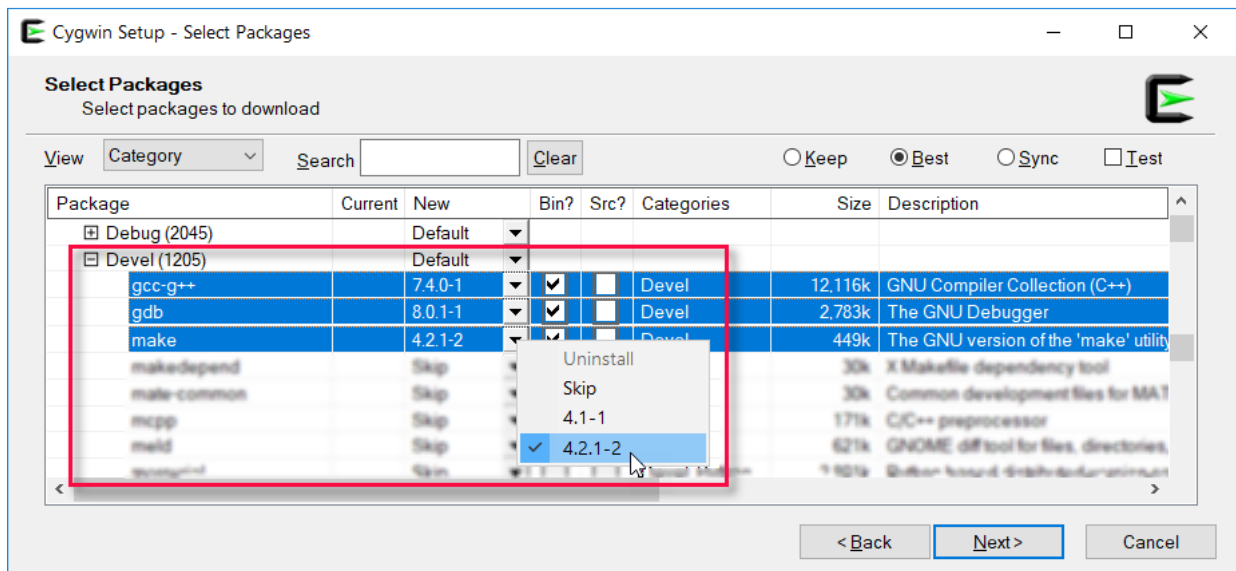


Figure 6: Select Packages

11. Press Next>

12. Cygwin resolves missing dependencies to required packages.

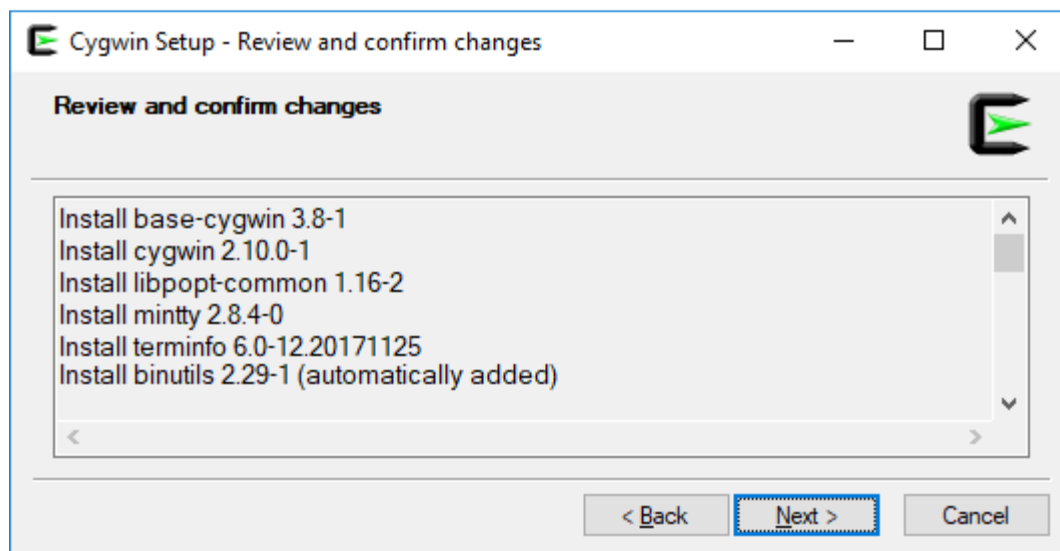


Figure 7: Resolved Dependencies

13. Click **Next>** to start the download and installation process.
14. When the installation is complete click **Finish>**

Testing the Cygwin installation

1. Open the cygwin Terminal



2. Type `gcc --version` and `g++ --version`
3. Type `make --version`

 A screenshot of a Cygwin Terminal window. The window title is '~'. The terminal output shows the following:


```
AzureAD+FrankBraun@RhapsodyDemoComputer ~
$ gcc --version
gcc (GCC) 6.4.0

AzureAD+FrankBraun@RhapsodyDemoComputer ~
$ g++ --version
g++ (GCC) 6.4.0

AzureAD+FrankBraun@RhapsodyDemoComputer ~
$ make --version
GNU Make 4.2.1
Built for i686-pc-cygwin
There is NO WARRANTY, to the extent permitted by law.
```

Figure 8: Check for no errors

Compiling Rhapsody Framework with Cygwin

Rhapsody Code generation is framework-based: it includes a fixed, predefined framework called the *Object Execution Framework* (OXF). The Rhapsody generated code reuses that framework. The compiled framework libraries which

are located in `Share/Lang<Language>/lib` are linked to the application generated from the Rational Rhapsody model.

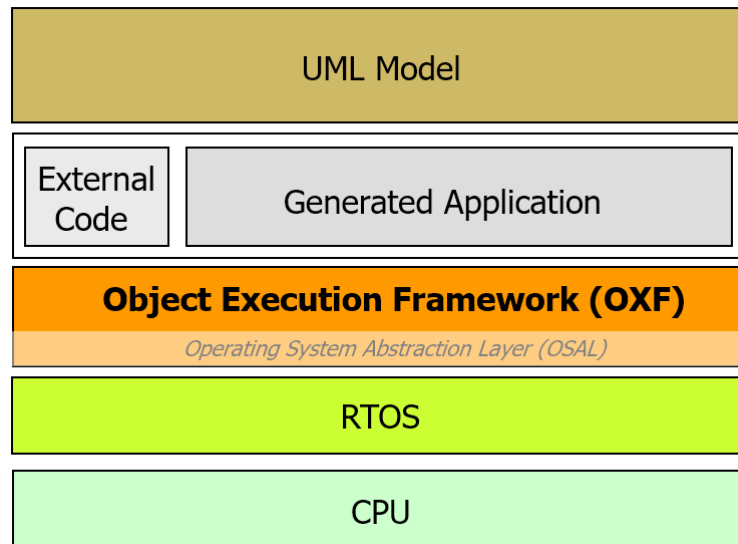


Figure 9: OXF Framework

Before we start, its a best practise to rebuild the OXF with the current *Cygwin* compiler

1. Launch Rhapsody and create a project
 - a) Start Rhapsody in C++ and create an initial project or open an existing one
 - b) Select the active configuration of the project

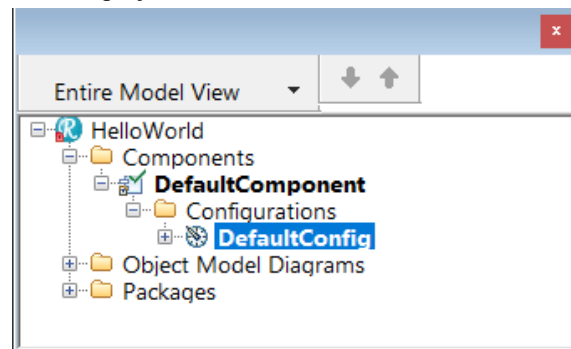


Figure 10: Initial Project

2. Select and configure Build-Environment
 - a) Open the Features Dialog of the active *Configuration*
 - b) In the **Settings** Tabs under **Environment Settings** select *Cygwin* as environment

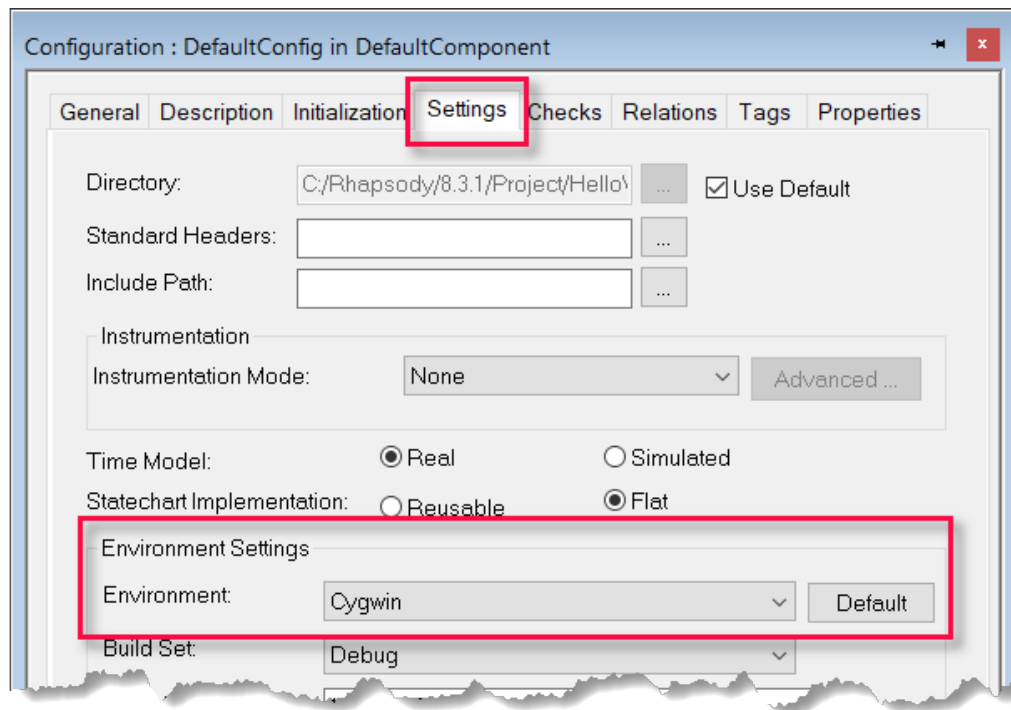


Figure 11: Environment Settings

3. Checking the CPU property

- a) In the Features dialog of the active configuration select the **Properties** Tab

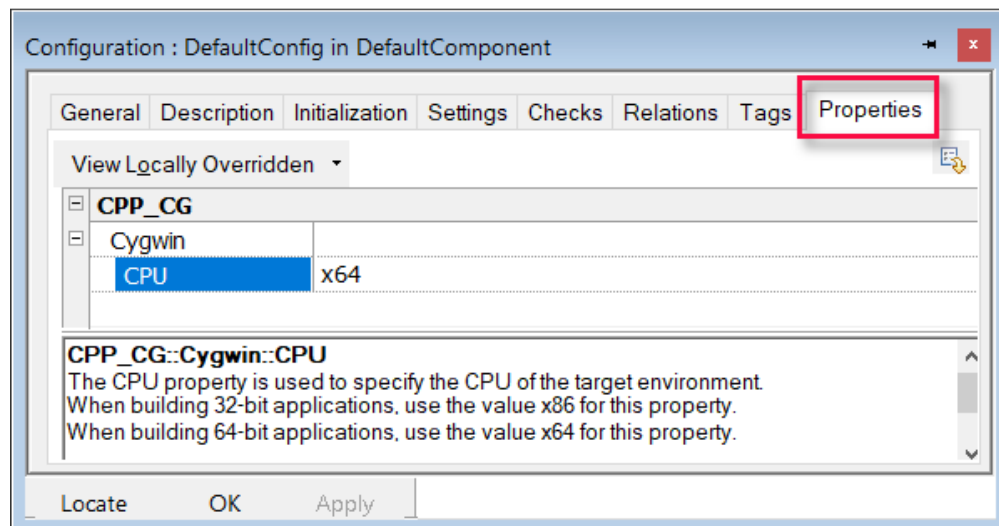


Figure 12: CPU Property using Cygwin 64bit

When you install Rational Rhapsody, the value that is assigned to the property [lang]_CG::Cygwin::CPU is determined by the version of Rational Rhapsody that you installed.

- If you installed the Rhapsody 64-bit version, the value of the property is set to x64.
- If you installed the Rhapsody 32-bit version, the value of the property is left blank.

! **Important: If you are using the 32 Bit Cygwin compiler, you have to make sure that value of the CPU property is set to blank.**

4. Build Framework

- a) You can build the Framework inside Rhapsody simply select in **Code->Build Framework**

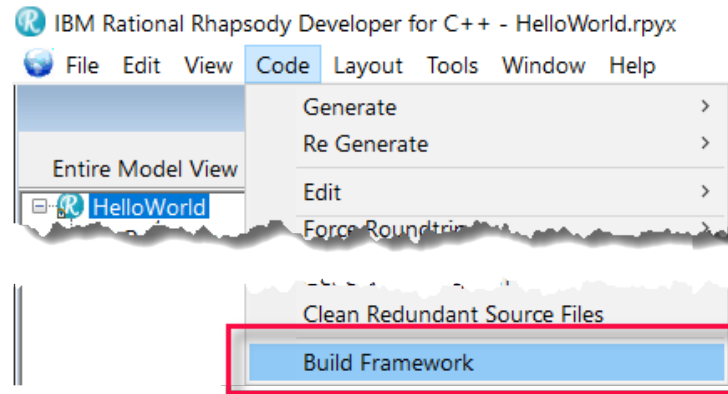


Figure 13: Building the Framework inside Rhapsody

b) Observe the compiling progress in the log **Output Window** and wait until the build is done.

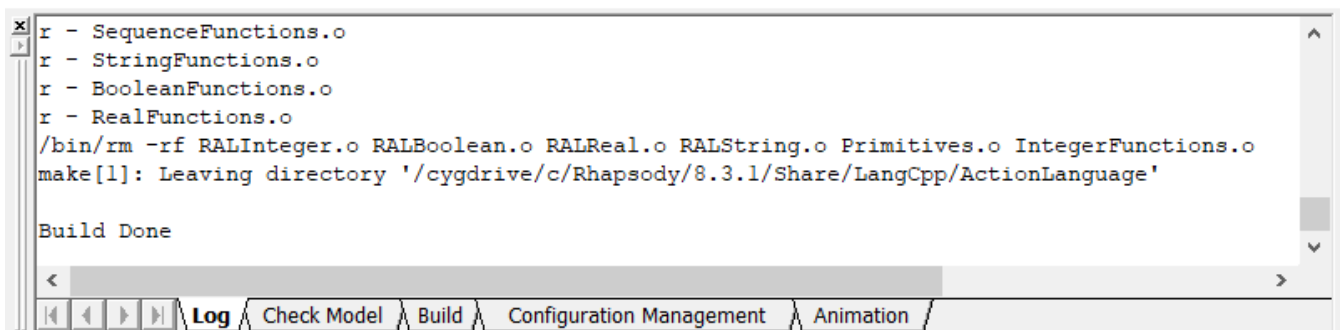


Figure 14: Output Window

c) Observe the build result in the <RhpInstall>/Share/LangCpp/lib path

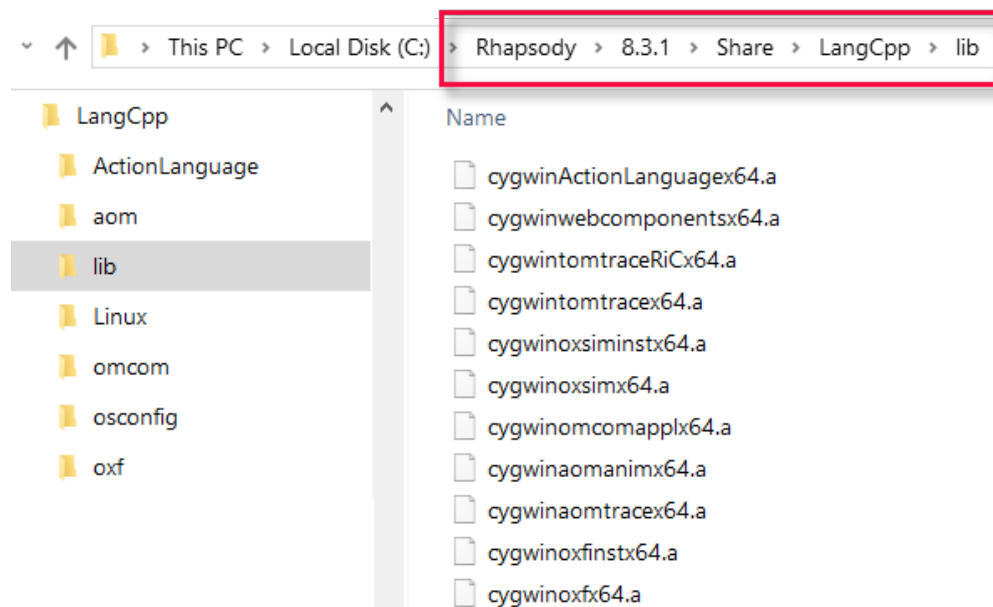


Figure 15: 64 bit Cygwin Framework libraries

Compiling Rhapsody Webservices with Cygwin

IBM® Rational® Rhapsody® webify provides a Web-Interface to control and monitor your Rhapsody created application. Its purpose is to setup a quick easy User-interface for on the fly testing and visual verification.

This build in Webserver is provided as compiled webservice library for various targets. The corresponding `cygwinWebServices.a` are already available for *Cygwin* 32bit and 64bit in `<RhpInstall>/Share/Lib` folder. It's not mandatory but good practise to rebuild them with your current compiler version.

Create Batch file

- a) In the folder `<RhpInstall>Share/WebServices` create a new batch file called `Rebuild_Cygwin_WebServices.bat`

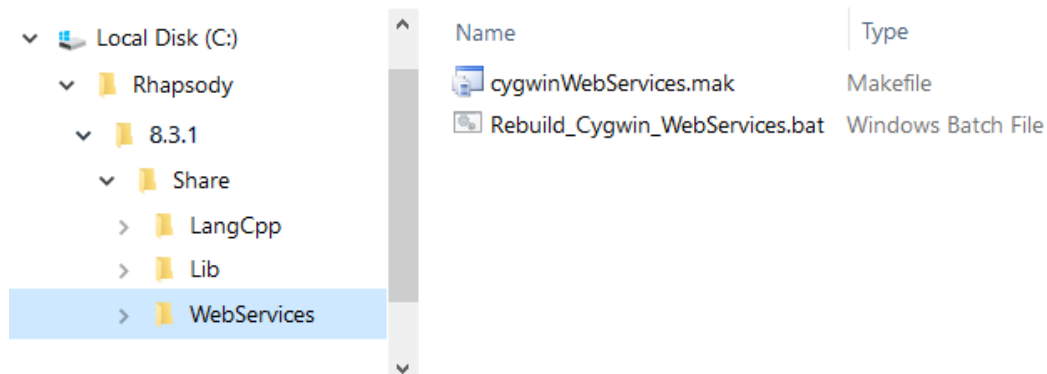


Figure 16: Batch file

- b) Fill the batch file with the following content. In case your using Cygwin 32 bit keep CPU empty

```
@echo off
set CYGWIN=nodosfilewarning
set LANG=en_US
set PATH=C:\Cygwin\bin;%PATH%
echo Setting environment for Cygwin
make -s -f cygwinWebServices.mak CPU=X64
pause
```

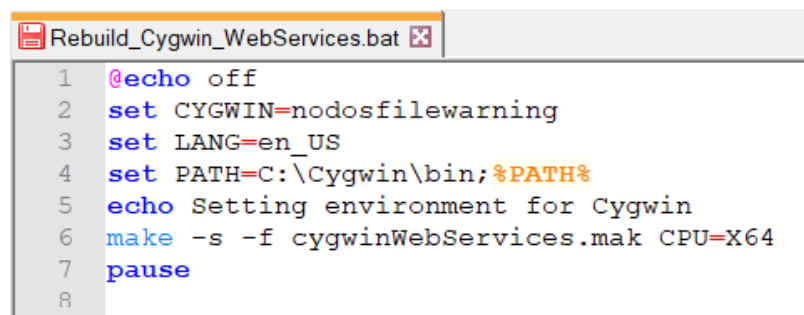
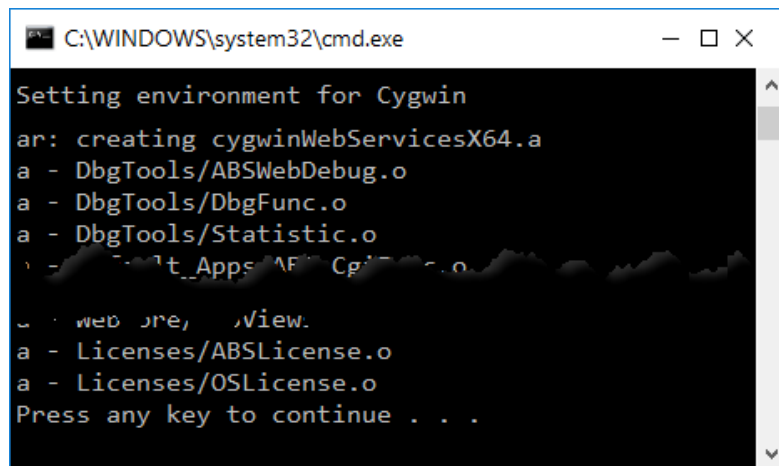


Figure 17: Code

- c) Double click on the batch file to execute it



```
C:\WINDOWS\system32\cmd.exe
Setting environment for Cygwin
ar: creating cygwinWebServicesX64.a
a - DbgTools/ABSWebDebug.o
a - DbgTools/DbgFunc.o
a - DbgTools/Statistic.o
a - Default_Apps_ABS_CygWeb.o
a - web_core_view.o
a - Licenses/ABSLicense.o
a - Licenses/OSLicense.o
Press any key to continue . . .
```

Figure 18: Executed Batchfile to compile the Webservices

d) In the folder <RhpInstall>/Share/lib check that you have a new library