

Cygwin详解

版本：v1.0

Crifan Li

摘要

本文主要介绍了什么是Cygwin，如何安装Cygwin，使用Cygwin时的一些常见问题总结，总结Cygwin的使用技巧和心得，其他和Cygwin相关的东西



本文提供多种格式供：

在线阅读	HTML ¹	HTMLs ²	PDF ³	CHM ⁴	TXT ⁵	RTF ⁶	WEBHELP ⁷
下载（7zip压缩包）	HTML ⁸	HTMLs ⁹	PDF ¹⁰	CHM ¹¹	TXT ¹²	RTF ¹³	WEBHELP ¹⁴

HTML版本的在线地址为：

http://www.crifan.com/files/doc/docbook/cygwin_intro/release/html/cygwin_intro.html

有任何意见，建议，提交bug等，都欢迎去讨论组发帖讨论：

http://www.crifan.com/bbs/categories/cygwin_intro/

修订历史

修订 1.0	2013-09-04	crl
1. 将原先部分内容从软件开发基础知识(soft_dev_basic)中整理过来		
2. 添加后来对于Cygwin的一些折腾，主要是如何重新安装Cygwin的某个软件包，如何选择合适的安装源		
3. 整理一些和Cygwin相关的东西，比如crosstool-ng等		

¹ http://www.crifan.com/files/doc/docbook/cygwin_intro/release/html/cygwin_intro.html

² http://www.crifan.com/files/doc/docbook/cygwin_intro/release/htmls/index.html

³ http://www.crifan.com/files/doc/docbook/cygwin_intro/release/pdf/cygwin_intro.pdf

⁴ http://www.crifan.com/files/doc/docbook/cygwin_intro/release/chm/cygwin_intro.chm

⁵ http://www.crifan.com/files/doc/docbook/cygwin_intro/release/txt/cygwin_intro.txt

⁶ http://www.crifan.com/files/doc/docbook/cygwin_intro/release/rtf/cygwin_intro.rtf

⁷ http://www.crifan.com/files/doc/docbook/cygwin_intro/release/webhelp/index.html

⁸ http://www.crifan.com/files/doc/docbook/cygwin_intro/release/html/cygwin_intro.html.7z

⁹ http://www.crifan.com/files/doc/docbook/cygwin_intro/release/htmls/index.html.7z

¹⁰ http://www.crifan.com/files/doc/docbook/cygwin_intro/release/pdf/cygwin_intro.pdf.7z

¹¹ http://www.crifan.com/files/doc/docbook/cygwin_intro/release/chm/cygwin_intro.chm.7z

¹² http://www.crifan.com/files/doc/docbook/cygwin_intro/release/txt/cygwin_intro.txt.7z

¹³ http://www.crifan.com/files/doc/docbook/cygwin_intro/release/rtf/cygwin_intro.rtf.7z

¹⁴ http://www.crifan.com/files/doc/docbook/cygwin_intro/release/webhelp/cygwin_intro.webhelp.7z

Cygwin详解:

Crifan Li

版本 : v1.0

出版日期 2013-09-04

版权 © 2013 Crifan, <http://crifan.com>

本文章遵从 : [署名-非商业性使用 2.5 中国大陆\(CC BY-NC 2.5\)](http://creativecommons.org/licenses/by-nc/2.5/)¹⁵

¹⁵ http://www.crifan.com/files/doc/docbook/soft_dev_basic/release/html/soft_dev_basic.html#cc_by_nc

目录

前言	v
1. 本文目的	v
2. 待完成	v
1. Cygwin简介	1
1.1. 什么是Cygwin	1
1.2. 什么会出现Cygwin	1
1.3. 什么人需要用到Cygwin	2
1.3.1. 举例说明Cygwin的各种用途	2
1.3.1.1. 在Cygwin下建立Docbook开发环境	2
1.4. Cygwin的版本	3
2. 下载和安装Cygwin	4
2.1. Cygwin的安装方式	4
2.2. Cygwin在线安装指南	5
2.2.1. Cygwin的setup.exe的安装首界面	5
2.2.2. 选择Cygwin的安装方式	6
2.2.3. 选择Cygwin的安装根目录	7
2.2.4. 选择Cygwin的本地安装包下载路径	8
2.2.5. 选择网络连接方式	9
2.2.6. 选择(所要连接的)服务器(去用于后续下载文件)	10
2.2.7. 选择需要安装的模块(安装包)	12
2.2.7.1. Cygwin中模块的各种分类	13
2.2.7.2. Cygwin中如何查看某分类下,默认已经安装了哪些模块	13
2.2.7.3. cygwin中如何取消/选择安装单个模块	13
2.2.7.4. Cygwin中如何取消/选择/重新/卸载安装某分类下的所有模块	14
2.2.7.5. Cygwin中如何知道某个模块是否已经选择安装了	15
2.2.7.6. 关于安装Cygwin模块方面的最简单,最核心的提示	16
2.2.8. 安装所选模块直至完成	18
3. Cygwin中各个模块详解	19
3.1. iconv	19
3.2. libxml2	19
3.3. libxslt	20
3.4. zlib	20
3.5. openSSH	21
4. Cygwin使用心得和技巧总结	22
4.1. 和Cygwin自身相关的一些工具	22
4.1.1. cygcheck	22
4.1.2. cygpath	22
4.2. Cygwin中使用注意事项	22
4.2.1. Cygwin的环境变量等路径中可能包含空格,括号等特殊字符而导致出错	22
4.3. Cygwin使用技巧总结	23
4.3.1. Windows下,添加右键打开Cygwin	23
4.3.2. 修改Cygwin的默认启动路径	25
4.3.3. 切换当前路径到别的磁盘,比如C盘, D盘等	26
5. Cygwin中常见错误及解决办法	27
5.1. Cygwin中常见错误及解决办法	27
5.1.1. Cygwin下的文件可以编辑但是无法保存	27
6. 与Cygwin相关的东西	29
6.1. 在Cygwin下搭建Docbook开发环境	29
6.2. 在Cygwin下编译crosstool-ng	29
6.3. 在Cygwin下编译Buildroot	29
参考书目	30

插图清单

2.1. Cygwin Net Release Setup Program	6
2.2. Choose A Download Site	11

前言

1. 本文目的

本文目的在于，详细解释什么是Cygwin，如何使用Cygwin

以实现，对于Cygwin不了解的人，看了后，可以达到：

知道什么是Cygwin

在具体如何安装和使用时，都有合适的参考资料

2. 待完成

- x86和x64的Cygwin
对于Cygwin的新划分除了x86和x64，抽空去研究搞懂，除了本身x86和x64的区别之外，细节上
有何其他区别。

- 163的源，现在也分x86和x64了
所以之前介绍的，添加：<http://mirrors.163.com/cygwin/>的方法，也是错了的
需要同步去更新。
对应章节：[为何手动添加URL时，输入的是而不是](#)

- 举例说明Cygwin的各种用途中更多的例子

- 把之前cygwin安装单个模块的内容整理过来
之前折腾cygwin中把make从3.82换成3.81的过程，整理过来
[【记录】Cygwin下把make从v3.82换成v3.81¹](#)
[【整理】Cygwin使用心得和使用技巧²](#)

¹ http://www.crifan.com/cygwin_change_make_from_3_82_to_old_3_81/

² http://www.crifan.com/summary_cygwin_usage_tricks/

第 1 章 Cygwin简介

相关旧帖：

1.1. 什么是Cygwin

Cygwin是一个集合，这个集合包含了一堆的工具，目的在于，在windows平台上，提供一个Linux的环境。

作为在Windows下面的软件，Cygwin中最核心的，是个dll库，叫做cygwin1.dll，

其提供了底层的Linux API的所有功能，供Cygwin下面的各种工具，模块等调用，以此来实现在Windows下模拟Linux环境。

从cygwin的具体表现和文件组成上来说，就是：核心的Cygwin DLL文件 + 各种Linux下面的模块（工具/软件/库）

1.2. 什么会出现Cygwin

计算机的世界，在操作系统方面，目前用的最广泛的，有三类：开源免费的Linux，微软的Windows，苹果的Mac。

随着计算机的发展，很多时候，尤其是做软件开发方面，常常会遇到在某种平台上，希望利用到另外一种平台上的工具，或环境，其中就包括在Windows下面，利用Linux中的环境和工具。

对于此种需求，最终才出现了各种方案，Cygwin就是其中的一种，提供了一个Windows下面使用Linux环境的平台。

有人会问了，不是很多功能，windows下面也有类似的软件可以实现吗？为何非要（在Windows环境下）去用Linux的环境（中的软件）呢？

那是因为，等你对Linux有更多了解的时候，就会发现，Linux下面的很多很多的更强大的功能，是windows下面没有对应的功能，或者即使有，也不好。

比如Linux世界中，最常用的项目编译过程管理的工具makefile，Windows下面的就没有足够好用的类似的工具

而且一般与Makefile搭配使用的其他很多别的工具，使得组合起来使用，可以实现非常强大且复杂的一些功能。

随着软件开发的深入，至少我个人感觉，才能越加发现Linux的各种强大和好处。

所以才会遇到，不少时候，需要在Windows下去模拟Linux环境，以便去使用Linux中的各种强大功能的软件，工具的



关于常见的系统Windows，Linux，Mac

对于常见的系统，Windows，Linux，Mac不熟悉的人，

可以去参考：

[【教程】软件开发基础知识之操作系统类型：Windows/Linux/Mac，32位/64位](#)¹

¹ http://www.crifan.com/software_develop_basic_knowledge_os_type_windows_linux_mac_32bit_64bit/

1.3. 什么人需要用到Cygwin

不论是做偏底层的嵌入式开发，还是做上层软件的开发，都可能会用到Cygwin。

简单说就是：

想要在Windows平台上，折腾Linux系统中的东西

然后才用到Cygwin去模拟Linux环境，去在Linux环境下实现各种复杂的功能和目标

此处，随便举一些例子：

1.3.1. 举例说明Cygwin的各种用途

1.3.1.1. 在Cygwin下建立Docbook开发环境

就像我之前写的教程中：

[纯Windows环境下的Docbook开发环境的搭建](#)²

详细介绍的一样，对于Docbook环境的搭建，其实是可以使用Windows下的二进制的xsltproc+fop来实现的

但是，其总体上能够实现的功能相对较弱。

而我此处，有个相对比较复杂的需求：

需要管理20多个Docbook的book

如果编译每一个book，都需要手动去输入对应的编译命令的话，相关的工作量极大且极其繁琐，且容易出错，直接就把人累死了

而通过Cygwin中的makefile来实现，就相对轻松多了。

所以后来就去花点时间，去写了对应的makefile

以实现，对于每个book下的src文件夹下，去：

```
make html
make pdf
make htmls
make rtf
make chm
make txt
```

就可以对于每个book，生成其对应的不同格式的输出文件。

而在windows下面，类似的功能，我最多只能用batch批处理文件去实现，

但是却缺少了Makefile中的依赖特性，其可以自动检测所依赖的文件是否已更新，决定是否重新编译。

即，对于多个文件，尤其是中型大型等项目的管理方面，windows的batch，比linux中的makefile差的太远了。

另外，由于要生成chm等格式的文件，我此处还需要一些其他的辅助工具

² http://www.crifan.com/files/doc/docbook/docbook_dev_note/release/htmls/pure_win_docbook_dev_env.html

比如用于转换编码的iconv，而这类好用易用的各种工具，在Linux下面，都是必备的，无需额外安装和太操心

但是如果想要在windows下使用，还要去一个个找到对应的windows下面的版本，或者找到相应的源码，自己去编译一个版本。

如此，经常需要利用到Linux下面的各种强大好用的工具，但是windows下面常常没有或者有类似的但不好用或功能很弱，

因此，才需要用到Cygwin去模拟Linux，

然后在Linux下面，利用已有的各种工具，实现各种复杂的需求的。

关于实现细节，可参考：

[Windows环境下的，基于Cygwin的Docbook开发环境的搭建](#)³

1.4. Cygwin的版本

cygwin的版本，涉及到两个东西。

一个是cygwin的核心==Cygwin DLL文件==cygwin1.dll，其最新版本，可以从[cygwin主页](#)⁴上看到，此刻，2012-06-20，是1.7.15-1。

另外一个就是在线安装文件setup.exe的版本，其在安装过程中可以看到，比如[图 2.1 “Cygwin Net Release Setup Program”](#)中的是2.774。

所以，如果你看到有人写的cygwin的版本是2.x的，那都是指的是setup.exe安装包的版本。如果写的是1.x.x的，那都是指的是核心的dll文件的版本。

³ http://www.crifan.com/files/doc/docbook/docbook_dev_note/release/htmls/cygwin_docbook_dev_env.html

⁴ <http://www.cygwin.com/>

第 2 章 下载和安装Cygwin

2.1. Cygwin的安装方式

Cygwin的安装方式，总的来说，你一般所能见到的，有两种：

1. 利用官网的setup.exe去下载安装
Cygwin的官网中，有个安装文件setup.exe，其固定的不变的下载的地址是[Cygwin的setup.exe](#)。

此setup.exe本身很小，比如此刻，2012-06-20，下载下来的setup.exe，就只有689KB：

名称	大小	类型	修改日期
setup.exe	689 KB	应用程序	2012-6-20 23:02

而这些在线下载或本地已有的文件，则是非常大，一般来说，都是多达1,2G，甚至3,4G，而安装完毕之后的cygwin则往往要占用5,6G甚至更多的硬盘空间。

当然，如果你安装时候所选择要安装的模块很少的话，占用硬盘空间则会相应变小一些，但估计也还是以GB为单位。

关于具体的安装过程，请参考[第 2.2 节 “Cygwin在线安装指南”](#)

2. 利用本地已有的cygwin的ISO去本地安装
你可能也许听说，有个Cygwin的ISO，也是用于安装Cygwin的。

而ISO中所包含的文件，就是一个setup.exe和下载下来的各个模块的安装包。

这些文件，其实就是上述通过setup.exe（在安装过程中）下载下来（并安装）的而已。

所以，并没有比setup.exe有啥特殊的地方。

不过，很明显，其有个最大的好处，那就是不用非要去通过setup.exe去在线下载和安装了。

因为，通过setup.exe去链接某个服务器，其速度往往不是那么快。

即便是目前对国内来说最快的163的服务器，速度已经很快了，但是还是没有（比如用迅雷）单独去下载ISO速度来得快。

所以，如果可以的话，还是ISO下载，来得快，然后本地安装，就避免了在线连接服务器去下载太慢的问题。

关于目前可用的ISO镜像的下载地址，请参考[目前已知的可用的Cygwin的ISO下载地址](#)



目前已知的可用的Cygwin的ISO下载地址

关于可用的Cygwin的ISO，之前就整理过一些帖子：

[【网络资源下载】Cygwin ISO Download](#)²

[【整理】cygwin 本地 离线 精简版 下载](#)³

¹ <http://cygwin.com/setup.exe>

² http://www.crifan.com/web_download_cygwin_iso_download/

³ http://www.crifan.com/order_cygwin_local_offline_starter_edition_download/

此处简要整理如下：

1. 之前的伯克利的那个地址：[700MB Cygwin ISO image](#)⁴，已经不可用了。因为国内的人去下载的太多了，导致人家限制了ip范围，只允许其校内的ip，至少不允许国内的ip去下载了。
2. 我整理的那个iso：[《Cygwin完整版v2.738 20111105》\(Cygwin_fullVersion_v2.738_20111105\)2.738\[光盘镜像\]](#)⁵，目前发现有点问题，暂时也不推荐下载了。等有空再去弄个可用的ISO。
3. 目前发现的，关于精简版本的cygwin iso，倒有几个，推荐一下：
 - Cygwin 2.738 本地 离线 安装包 精简版 88M
[cygwin离线安装包5-1](#)⁶
[cygwin离线安装包5-2](#)⁷
[cygwin离线安装包5-3](#)⁸
[cygwin离线安装包5-4](#)⁹
[cygwin离线安装包5-5](#)¹⁰
 - Cygwin 本地 离线 安装包 精简版 20M+
[Cygwin2.738本地安装包-Part1](#)¹¹
[Cygwin2.738本地安装包-Part2](#)¹²
[Cygwin2.738本地安装包-Part3](#)¹³
[Cygwin2.738本地安装包-Part4](#)¹⁴

2.2. Cygwin在线安装指南

下面详细解释Cygwin的setup.exe的安装过程：

2.2.1. Cygwin的setup.exe的安装首界面

双击setup.exe，则会出现安装界面：

⁴ <http://inst.eecs.berkeley.edu/~instcd/iso/cygwin-cd-release-20050311.iso>

⁵ <http://www.verycd.com/topics/2908689/>

⁶ <http://download.csdn.net/detail/vivianke/3291998>

⁷ <http://download.csdn.net/detail/vivianke/3292027>

⁸ <http://download.csdn.net/detail/vivianke/3292059>

⁹ <http://download.csdn.net/detail/vivianke/3292086>

¹⁰ <http://download.csdn.net/detail/vivianke/3292114>

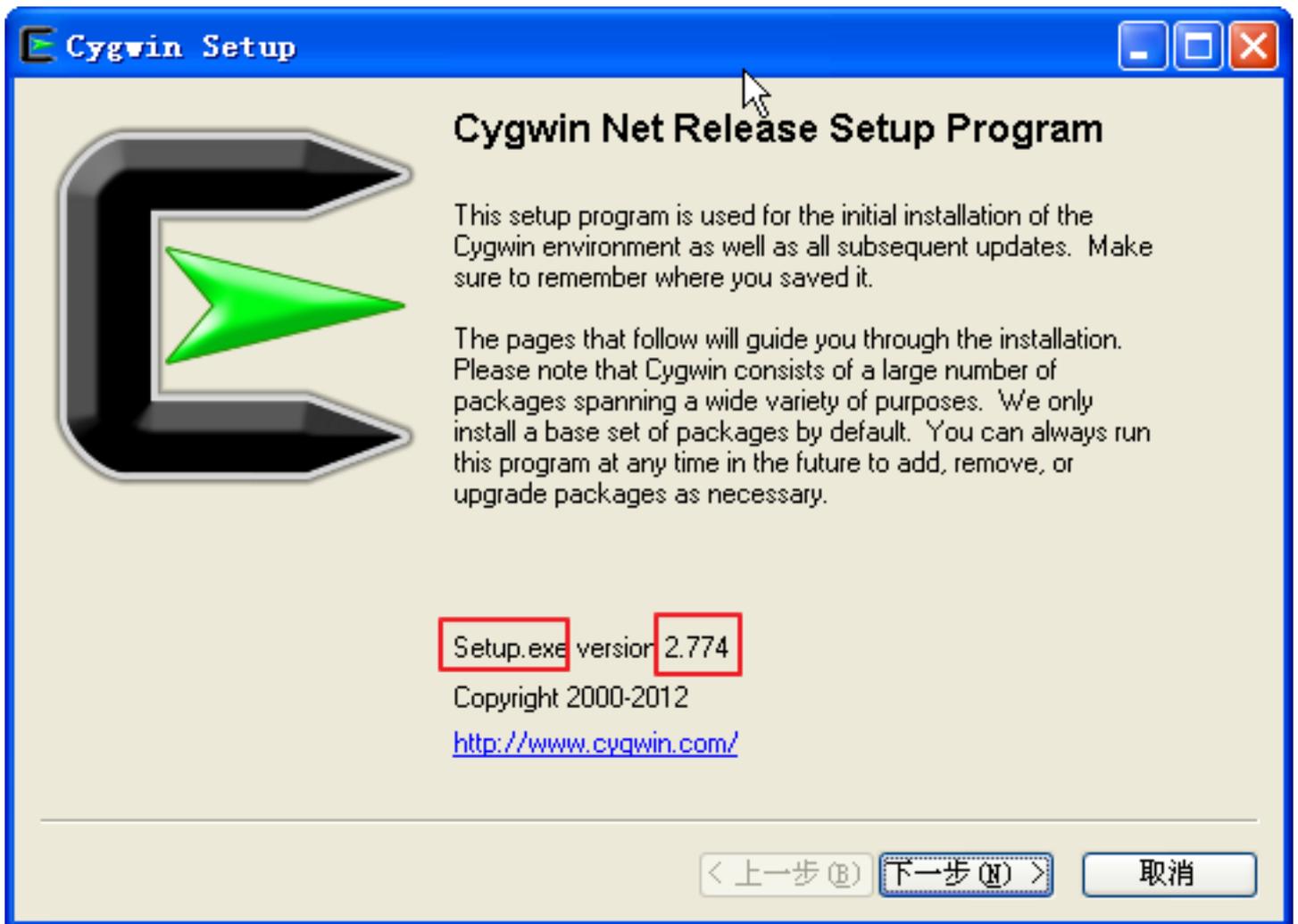
¹¹ <http://download.csdn.net/detail/qq280948982/3265007>

¹² <http://download.csdn.net/detail/qq280948982/3265174>

¹³ <http://download.csdn.net/detail/qq280948982/3265188>

¹⁴ <http://download.csdn.net/detail/qq280948982/3264982>

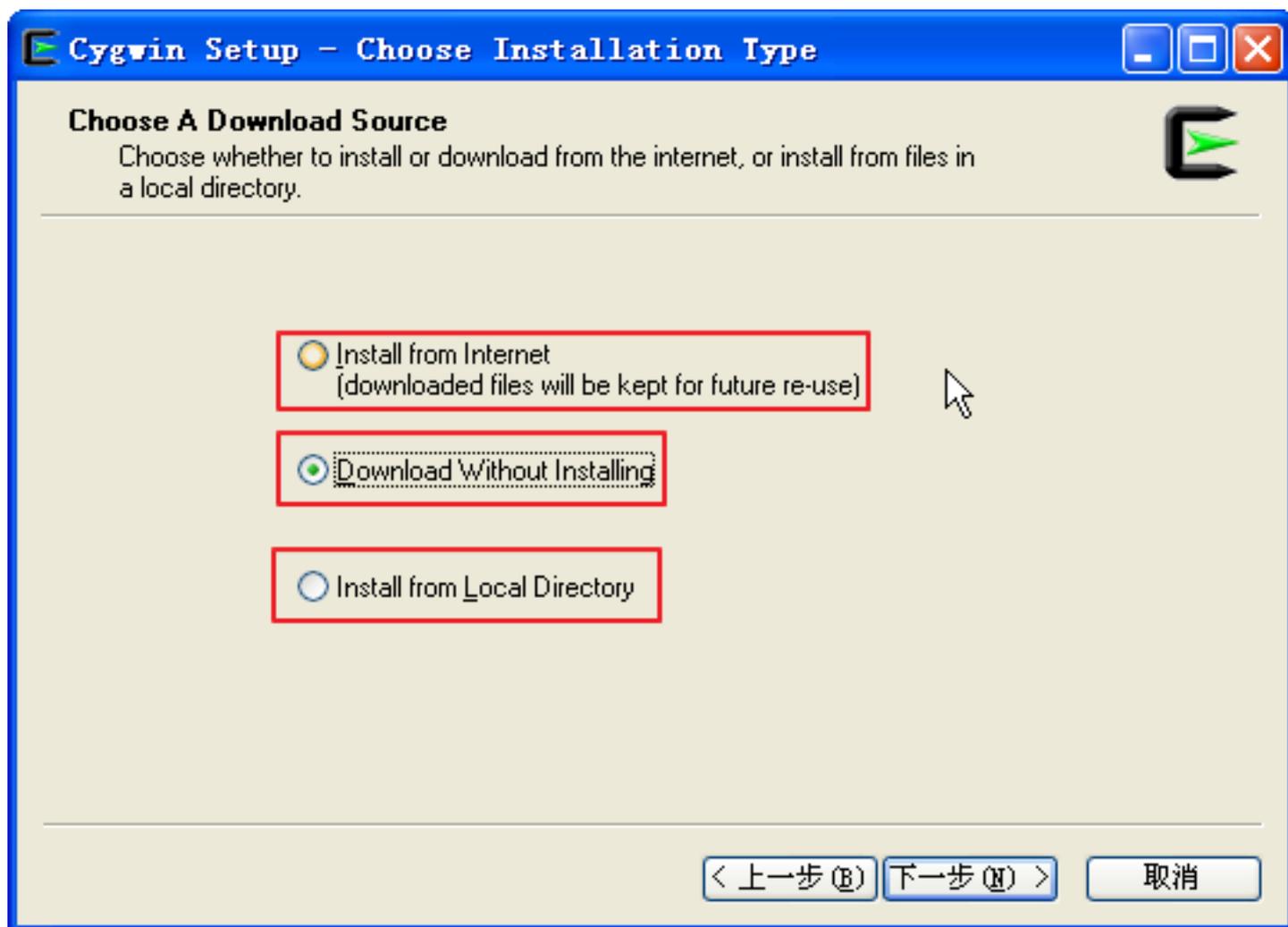
图 2.1. Cygwin Net Release Setup Program



可以看到，其中对于setup.exe，即此安装包，也是有对应的版本的，此处的是2.774。

2.2.2. 选择Cygwin的安装方式

点击“下一步”，出现“Choose installation type”界面：



其中有三个选项：

- Install from internet(downloaded files will be kept for feature reuse)
从网上下载（下载下来的文件，也可以将来再用）

其意思是，从网上下载的文件，存在本地硬盘后，以后万一遇到诸如某个模块被破坏了，不能用了，则可以再通过此setup.exe去重新安装一下，但是选择的是下面要说的第三项，即选择从本地某个文件夹安装，即此处之前下载好了的，效果相当于，windows中安装程序的修复功能。

关于如何从网上下载安装，接下来会详细解释。

- Download Without Installing
只下载不安装

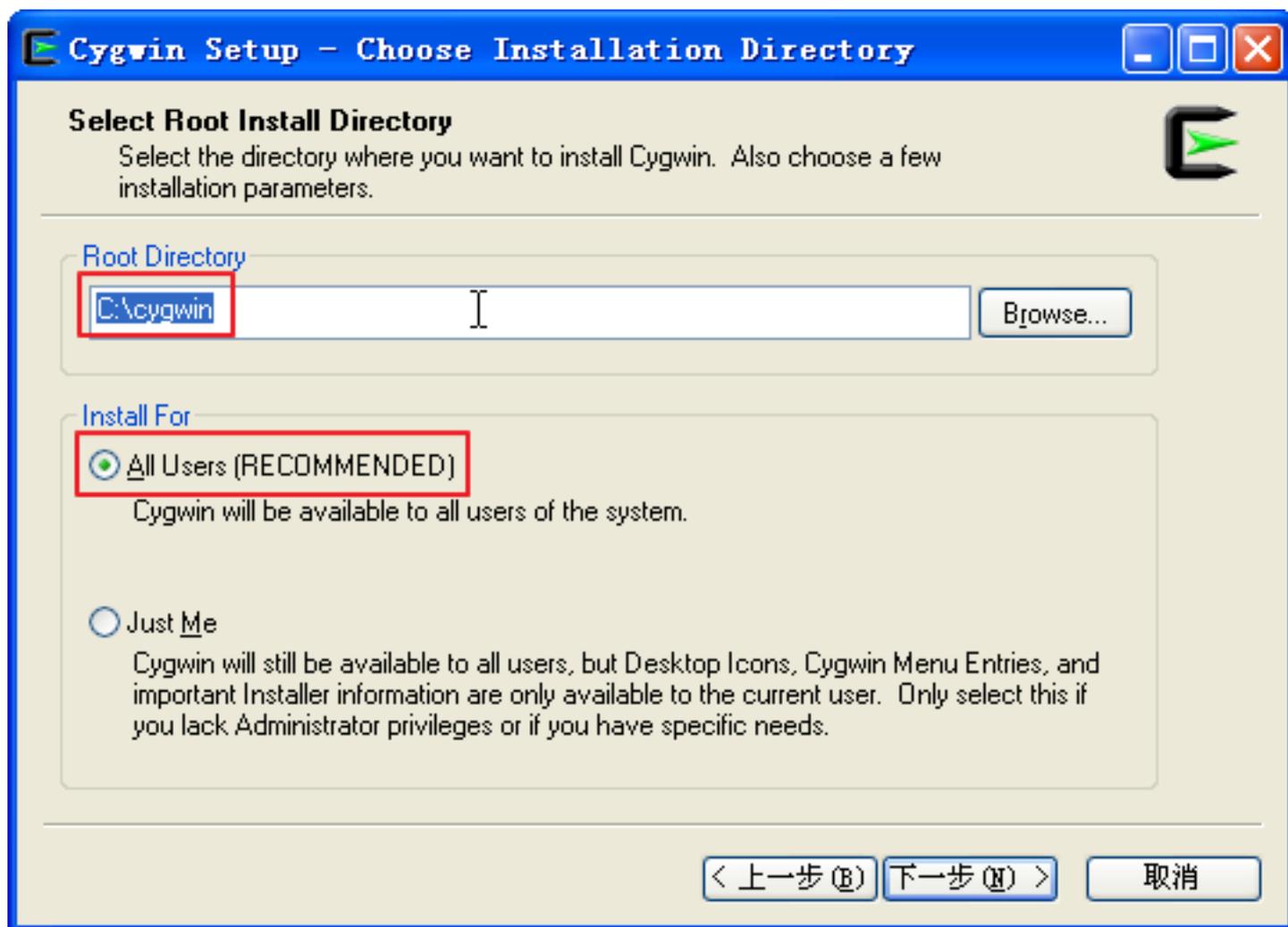
其用意，暂时没有完全搞懂。大概是为了先完全下载下来，然后方便选择性的安装自己需要的模块吧。

- Install from Local Directory
为选择从本地安装

上面已经提到了，其前提是，之前已经下载过了对应的所需的各个安装模块了，此时可以选择此项，去全新安装或者修复式安装某个模块。

2.2.3. 选择Cygwin的安装根目录

点击“下一步”后，出现“Select Root Install Directory”的界面：



其中Root Directory，指的是你所要将Cygwin安装到哪个目录。

默认为C:\cygwin，此处可以改为自己所要的路径，也可以用默认值，都可以。

我此处改为我所要的路径：E:\DevTools\Cygwin\cygwin_install

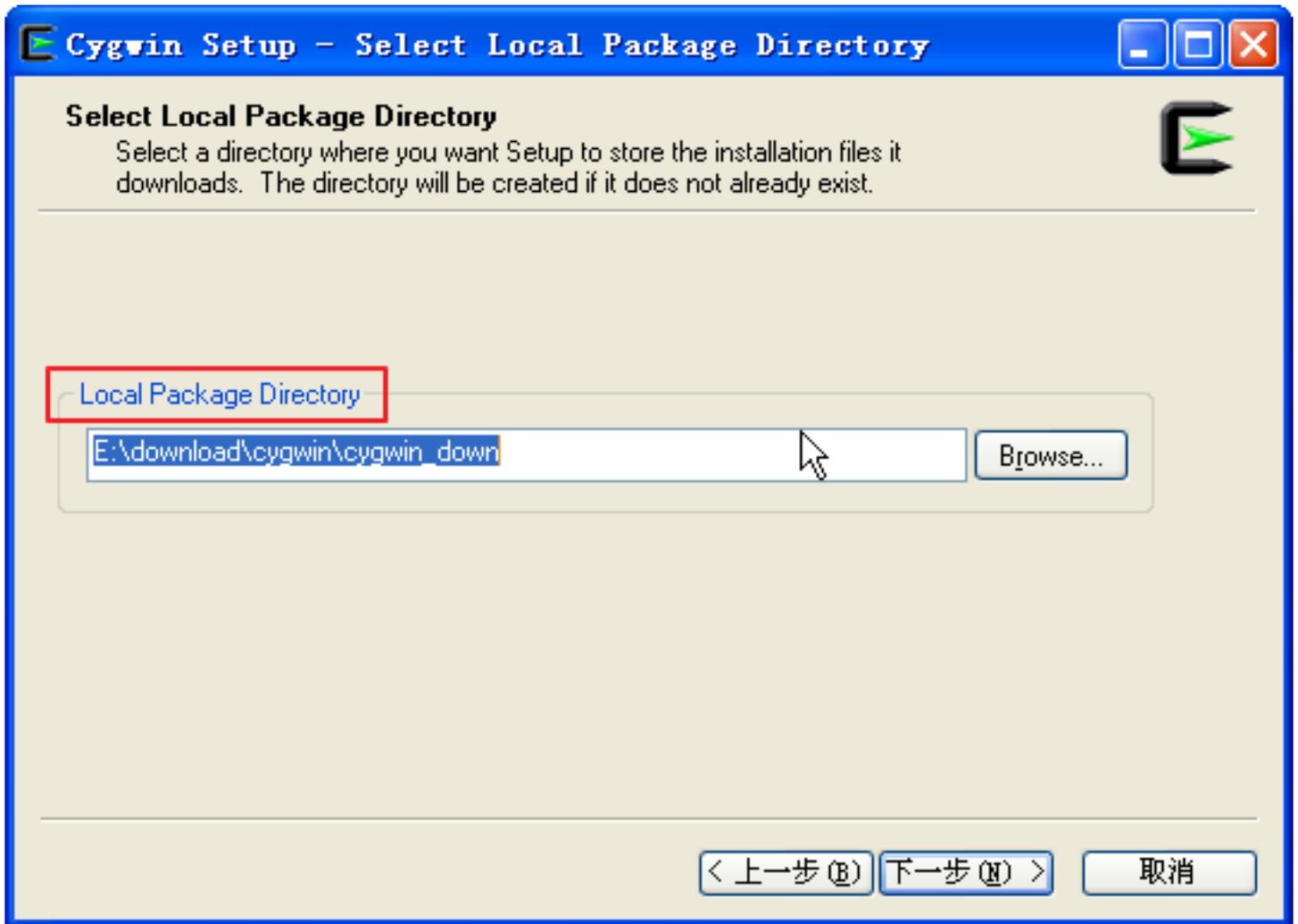
然后对于Install for，有两种选择：

- All User(RECOMMENDED)
为windows当前所有用户都有效。
- Just Me
只对当前的windows用户有效。

此处选择默认的，对所有用户都有效，即可。

2.2.4. 选择Cygwin的本地安装包下载路径

点击“下一步”后，进入“Select Local Package Directory”的界面：



其中Local Package Directory部分，是由于前面选择的是从网上下载并安装，所以此处让你选择对于下载下来的各个模块的安装文件，都放在哪里，此处，根据自己需要，设置一个路径即可。

我此处设置为E:\download\cygwin\cygwin_down



设置的路径（文件夹）最好是已经创建好的

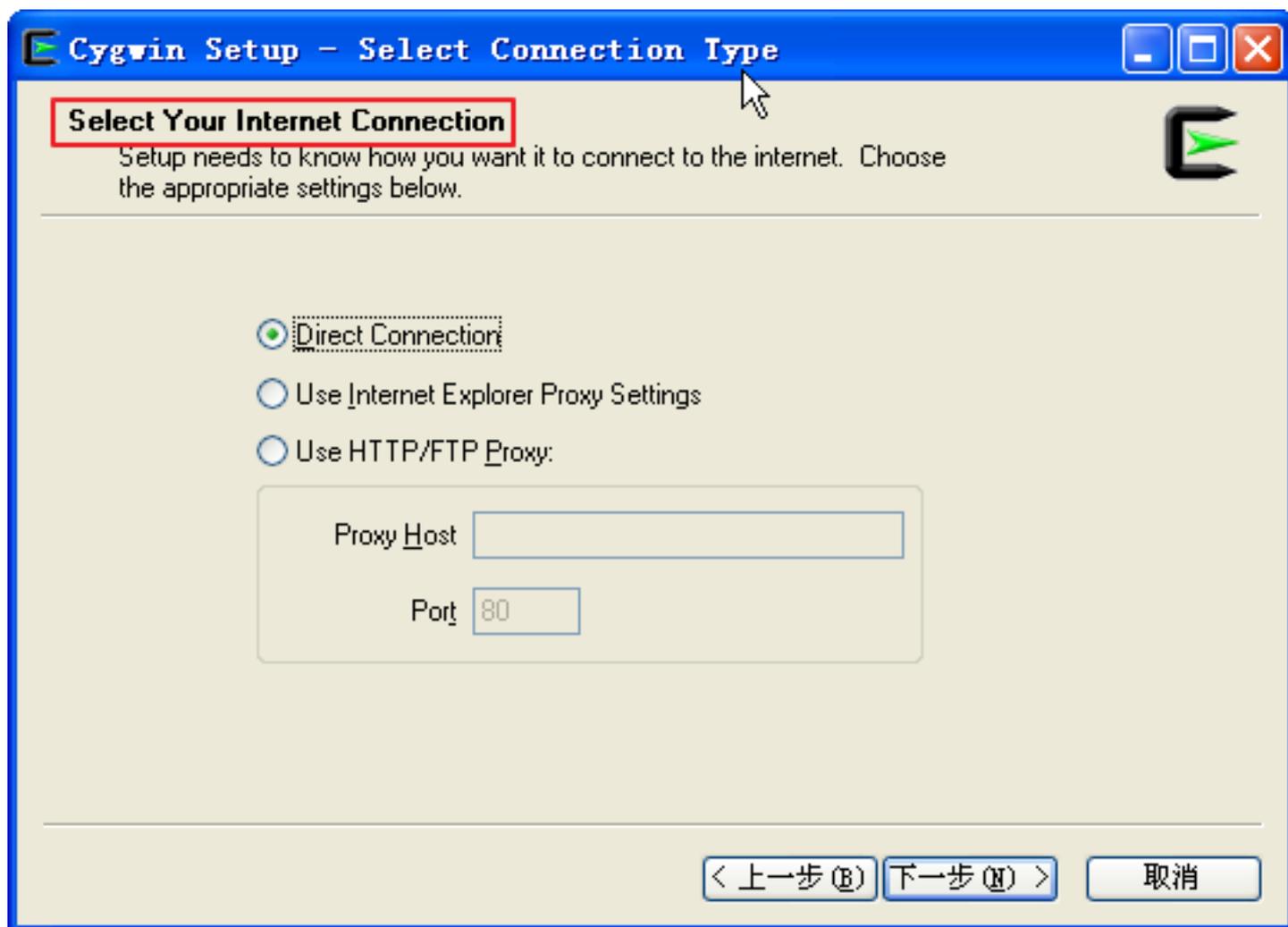
不过要注意一点的是，确保你的设置的路径，是真实存在的，否则就会出现这样的错误提示了：



当然，此时也是可以点击“是(Y)”去让其帮你创建对应的路径的。

2.2.5. 选择网络连接方式

然后就进入“Select Your Internet Connection”的界面了：



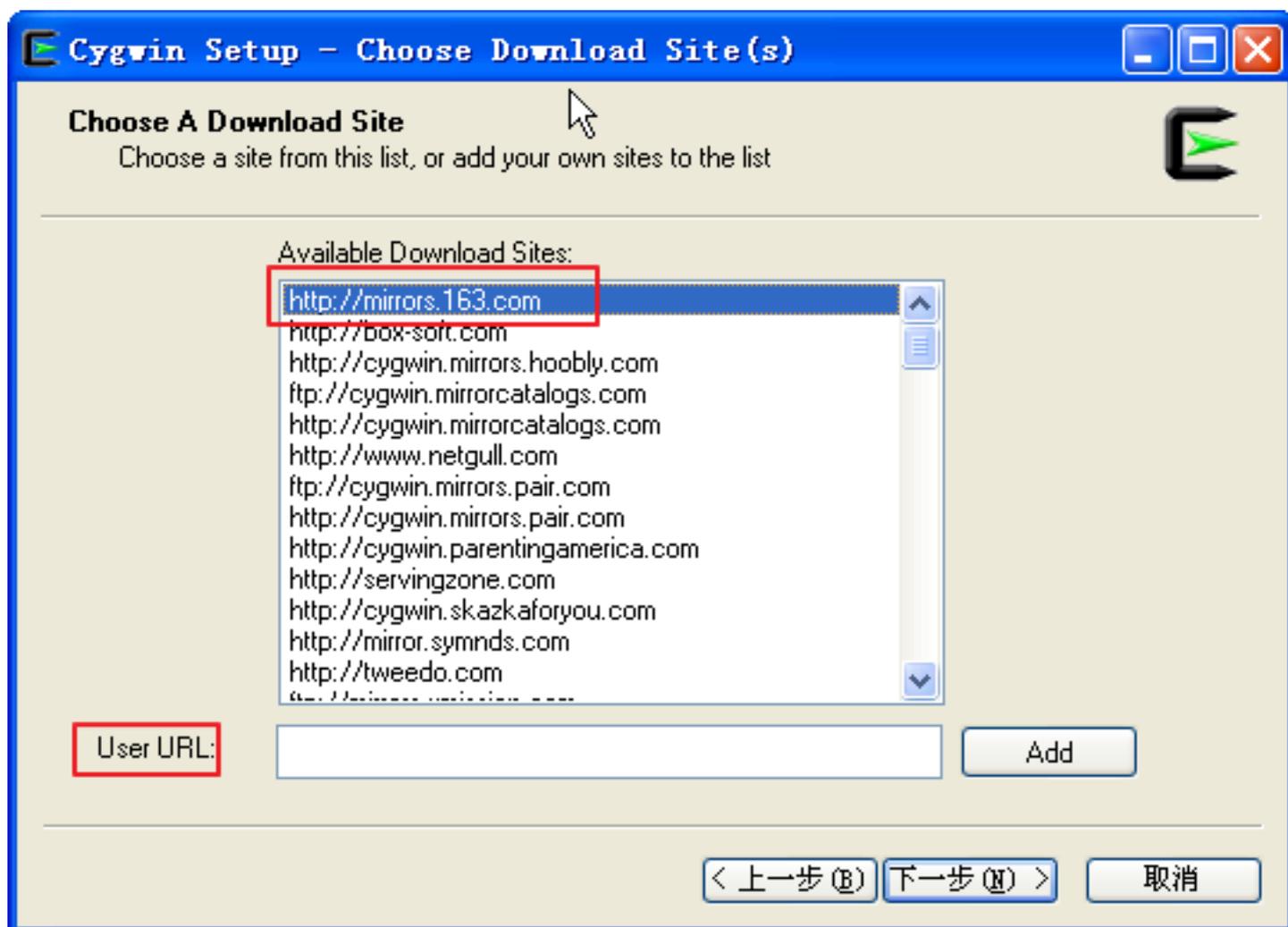
此时对于网络连接方式，有几种选择：

- Direct Connection
一般多数用户都是这种直接连接的网络，所以都是直接使用默认设置即可。
- Use Internet Explorer Proxy Settings
使用IE代理设置，如果你本身上网是通过代理上的，且IE中已经设置好了代理，那么就可以用此种设置了。
- Use HTTP/FTP Proxy
使用HTTP或FTP类型的代理。同理，如果有需要，自己选择此项后，设置对应的代理地址和端口，即可。

2.2.6. 选择（所要连接的）服务器（去用于后续下载文件）

点击“下一步”后，其会自动去下载一个服务器的列表，然后跳转到“Choose A Download Site”的界面：

图 2.2. Choose A Download Site



此处需要选择一个服务器，之后就是从该服务器下载对应的安装所需的模块文件了。

因此，此处如果选择的服务器不合适的话，尽管你的网络本身速度很快，但是此处下载速度很慢。

所以，此处需要选择一个速度比较快的服务器。

对于国内来说，连接很多国外的服务器，速度一般都很慢。

不过，目前已经的，口碑较好的，速度较快的服务器，是近几年刚出现的163的服务器。

所以，此处对于多数人，最好的选择就是选择163的服务器。

此处，我这里，由图可见，已经默认选择好了163的服务器，所以不需要额外设置了。

如果你的选择其他的服务器，而且服务器列表中也找不到163的话，那么可以自己手动添加：

在Use URL处，输入：

`http://mirrors.163.com/cygwin/`



关于163镜像服务器

163是个好公司，因为刚看到<http://mirrors.163.com/help/>中的介绍，得知其前身就是cn99镜像。

之前玩过Ubuntu的人，估计多数都知道cn99这个镜像，因为对于国内来说，apt的源的地址中，属其速度最快。

而且，现又发现，163的镜像，除了提供cygwin，其还提供了其他很多资源的镜像，都可以在<http://mirrors.163.com/>中找到。

包括很多常见的，比如mandriva，openSUSE,ubuntu等等。



为何手动添加URL时，输入的是<http://mirrors.163.com/cygwin/>而不是<http://mirrors.163.com/>

首先要说的是，直接输入在图 2.2 “Choose A Download Site” 中所看的地址<http://mirrors.163.com/>，是无法正常连接163服务器去下载cygwin的。

这也是我之前在【记录】[Cygwin下载过程记录](#)¹⁵中所遇到并解决了的问题。

而具体的原因是为何，我也是刚刚才注意到的：

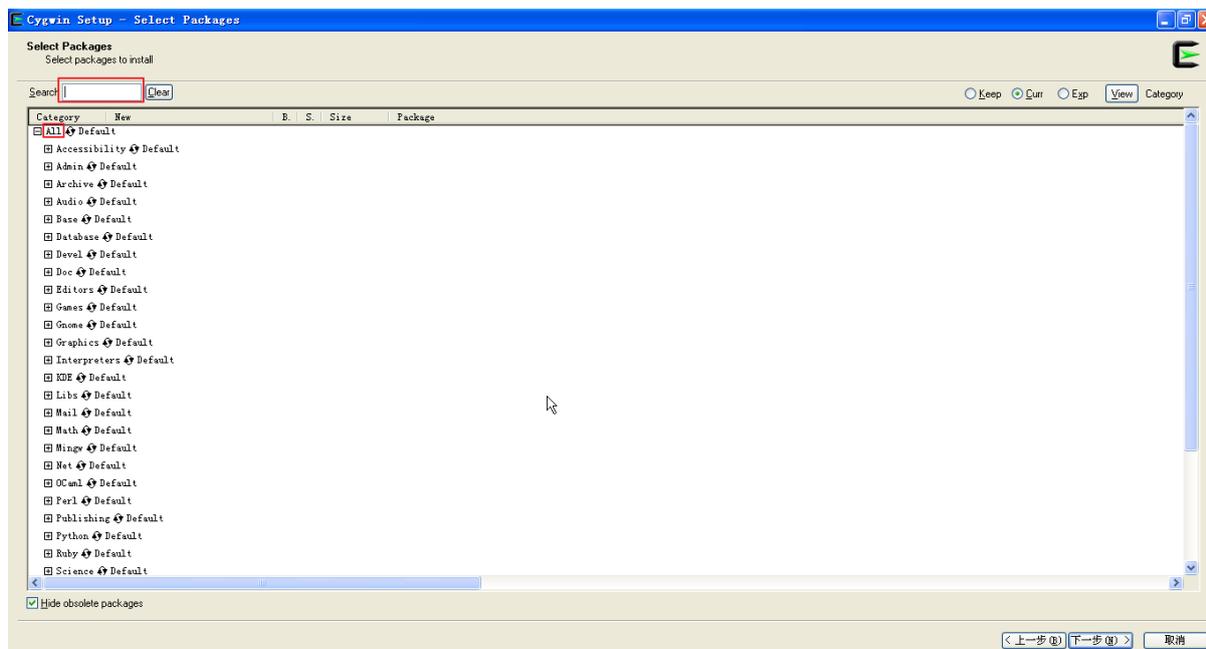
因为cygwin去安装的话，不论是在线安装还是本地安装，都会需要一个对应的配置文件：setup.ini，而你打开<http://mirrors.163.com/>中，是找不到对应的setup.ini的，但是在<http://mirrors.163.com/cygwin/>中是可以找到对应的setup.ini的。

因此，你给cygwin的setup.exe指定了<http://mirrors.163.com/>后，其从此URL中，找不到对应的setup.ini，所以无法继续安装cygwin。

所以URL中需要输入<http://mirrors.163.com/cygwin/>。

2.2.7. 选择需要安装的模块（安装包）

点击“下一步”后，其会解析一下，然后进入“Select Package”的界面：



此界面，才是整个Cygwin安装过程中的最重要，最需要详细讲解的部分。

¹⁵ http://www.crifan.com/records_record_cygwin_download_process/

2.2.7.1. Cygwin中模块的各种分类

先来说说，那一堆的列表。

可以看到，其有Accessibility，Base，Devel，Editors，Math等很多的部分。

这些，是总体的分类，对于Cygwin中所包含的N个模块的分门别类。

此处，对于安装Cygwin来说，就是安装各种各样的模块而已。

具体安装什么模块，则是根据你自己的需要，去选择不同的模块。

而对于大多数人，尤其是不熟悉的人，其实，最核心的，要记住的一点，那就是，记住一定要安装Devel这个部分的模块，其中包含了各种开发所用到的工具或模块。

而对于其他部分的设置，如果不熟悉，那么可以直接使用默认配置即可。

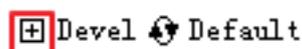
2.2.7.2. Cygwin中如何查看某分类下，默认已经安装了哪些模块

下面再来解释一下，如何选择或者取消对应的模块。

可以看到，其默认的各个分类后面的是Default字样，其意思很明显，就是cygwin帮我们设置好的，默认的选择安装了一些模块，其具体包含哪些模块，可以通过点击对应的分类前面的加号以展开。

以Devel为例：

点击加号：



然后就可以看到，cygwin已经默认帮忙选择了哪些安装哪些模块：

	Skip	n/a	n/a	5,600k	gdb: The GNU Debugger
	Skip	n/a	n/a	5k	gdk-pixbuf2-wmf: Windows Metafile library - (GdkPixbuf loader)
	Skip	n/a	n/a	22k	gendef: Generates exports definitions by analyzing DLLs.
	0.18.1.1-2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	237k	gettext: GNU Internationalization library and core utilities (PLUS LINK LIBS)
	Skip	n/a	n/a	2,665k	gettext-devel: GNU Internationalization development utilities (but NOT LINK LIBS)
	Skip	n/a	n/a	3,884k	git: Fast Version Control System - core files
	Skip	n/a	n/a	16k	git-completion: Fast Version Control System - git bash completion
	Skip	n/a	n/a	187k	git-gui: Fast Version Control System - git-gui viewer
	Skip	n/a	n/a	64k	git-svn: Fast Version Control System - git-svn conversions
	Skip	n/a	n/a	106k	gitk: Fast Version Control System - gitk viewer
	Skip	n/a	n/a	99k	gnome-common: Common development files for GNOME packages
	Skip	n/a	n/a	1,802k	gnutls-devel: Library implementing TLS 1.0 and SSL 3.0 protocols - (development)
	Skip	n/a	n/a	79k	gperf: Perfect hash function generator

可以看到，有很多n/a的标示，其表示Not Available不可用，此处即为不安装的意思，对应的模块的前面也写的是Skip，同样是跳过安装此模块的意思。

而对应方括号中间打叉，表示的是选择安装该模块，上图中即为选择安装gettext这个模块。

2.2.7.3. cygwin中如何取消/选择安装单个模块

而想要安装某单个模块，比如gdb模块的话，那么就是对着那个skip前面的图标，点击一下，然后就可以看到gdb模块被选中了，表示安装此模块：

⚙️ Skip	n/a	n/a	5k	gccmakedep: Xorg preprocessor dependency maker
👉 7.3.50-3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	5,600k	gdb: The GNU Debugger
⚙️ Skip	n/a	n/a	5k	gdk-pixbuf2-wmf: Windows Metafile library - (GdkPixbuf loader)
⚙️ Skip	n/a	n/a	22k	gendeF: Generates exports definitions by analyzing DLLs.
👉 0.18.1.1-2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	237k	gettext: GNU Internationalization library and core utilities (PLUS LINK LIBS)
⚙️ Skip	n/a	n/a	2,665k	gettext-devel: GNU Internationalization development utilities (but NOT LINK LIBS)
⚙️ Skip	n/a	n/a	3,884k	git: Fast Version Control System - core files

2.2.7.4. Cygwin中如何取消/选择/重新/卸载安装某分类下的所有模块

很明显，如果想要对几十个，几百个模块都想要安装的话，如果都要一个个点击，那岂不是累死了。

对此，cygwin早就想好了，给我们提供了更方便的设置，那就是，对于此Devel分类下面的所有的模块，如果你想要都安装，那么可以点击Devel后面的那个Default：

☐ Devel	👉 Default			
⚙️ Skip	n/a	n/a	157k	ELFIO: ELF file reader and producer implemented as a C++ library
⚙️ Skip	n/a	n/a	2,132k	SWI-Prolog: Prolog Interpreter
⚙️ Skip	n/a	n/a	849k	XmHTML-devel: A widget capable of displaying HTML 3.2 conforming text - (development)
⚙️ Skip	n/a	n/a	104k	aalib-devel: An ascii art library - (development)
⚙️ Skip	n/a	n/a	1,061k	algot68g: Algol 68 Genie compiler.
⚙️ Skip	n/a	n/a	14k	ascii: ASCII character tables

那么Default就会变成Install，同时，其下所有的模块，也都变成选择安装了：

☐ Devel	👉 Install			
👉 1.9.2-1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	157k	ELFIO: ELF file reader and producer implemented as a C++ library
👉 5.6.36-1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2,132k	SWI-Prolog: Prolog Interpreter
👉 1.1.7-10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	849k	XmHTML-devel: A widget capable of displaying HTML 3.2 conforming text - (development)
👉 1.4rc5-10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	104k	aalib-devel: An ascii art library - (development)
👉 2.3.7.4-0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1,061k	algot68g: Algol 68 Genie compiler.
👉 3.11-1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	14k	ascii: ASCII character tables
👉 8.6.3-1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	170k	asciidoc: Text based document generation

对应的，再次点击Install，则会变成Reinstall，对应的Devel下面的各个模块，也都变成相应的skip了：

☐ Devel	👉 Reinstall			
⚙️ Skip	n/a	n/a	157k	ELFIO: ELF file reader and producer implemented as a C++ library
⚙️ Skip	n/a	n/a	2,132k	SWI-Prolog: Prolog Interpreter
⚙️ Skip	n/a	n/a	849k	XmHTML-devel: A widget capable of displaying HTML 3.2 conforming text - (development)
⚙️ Skip	n/a	n/a	104k	aalib-devel: An ascii art library - (development)
⚙️ Skip	n/a	n/a	1,061k	algot68g: Algol 68 Genie compiler.
⚙️ Skip	n/a	n/a	14k	ascii: ASCII character tables
⚙️ Skip	n/a	n/a	170k	asciidoc: Text based document generation
⚙️ Skip	n/a	n/a	142k	astyle: Artistic Style is a reindenter and reformatter of C, C++, C# and Java source code.
⚙️ Skip	n/a	n/a	115k	autobuild: Generate summary information from build logs
⚙️ Skip	n/a	n/a	4k	autoconf: Wrapper scripts for autoconf commands

此处对于reinstall之所以变成skip，那是因为之前就没有安装过，所以此处对于之前没有安装过的模块，就是变成skip，即也不安装了。而如果你是第二次运行此cygwin安装程序，那么对于那些之前安装过的，则会变成对应的reinstall的。

同理，点击Reinstall，则又会变成Uninstall：

☐ Devel	👉 Uninstall			
⚙️ Skip	n/a	n/a	157k	ELFIO: ELF file reader and producer implemented as a C++ library
⚙️ Skip	n/a	n/a	2,132k	SWI-Prolog: Prolog Interpreter
⚙️ Skip	n/a	n/a	849k	XmHTML-devel: A widget capable of displaying HTML 3.2 conforming text - (development)
⚙️ Skip	n/a	n/a	104k	aalib-devel: An ascii art library - (development)
⚙️ Skip	n/a	n/a	1,061k	algot68g: Algol 68 Genie compiler.
⚙️ Skip	n/a	n/a	14k	ascii: ASCII character tables
⚙️ Skip	n/a	n/a	170k	asciidoc: Text based document generation
⚙️ Skip	n/a	n/a	142k	astyle: Artistic Style is a reindenter and reformatter of C, C++, C# and Java source code.

含义很明显，卸载对应的模块的意思。

如此，对于某个分类，点击对应的后面的安装选项的话，会在：Default，Install，Reinstall，Uninstall这几个选项之间来回切换。

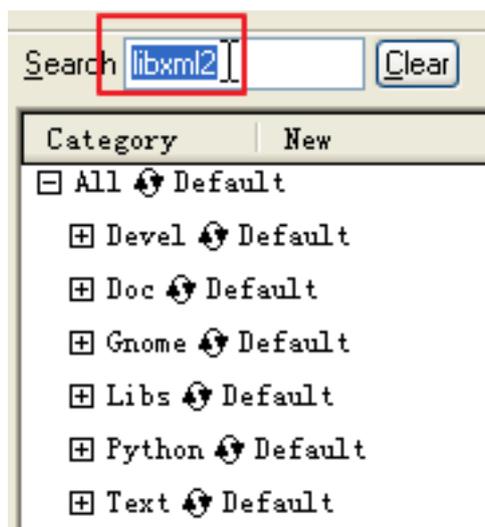
2.2.7.5. Cygwin中如何知道某个模块是否已经选择安装了

而对于想要知道某个库，或者说模块，程序等，是否已经选择安装了，

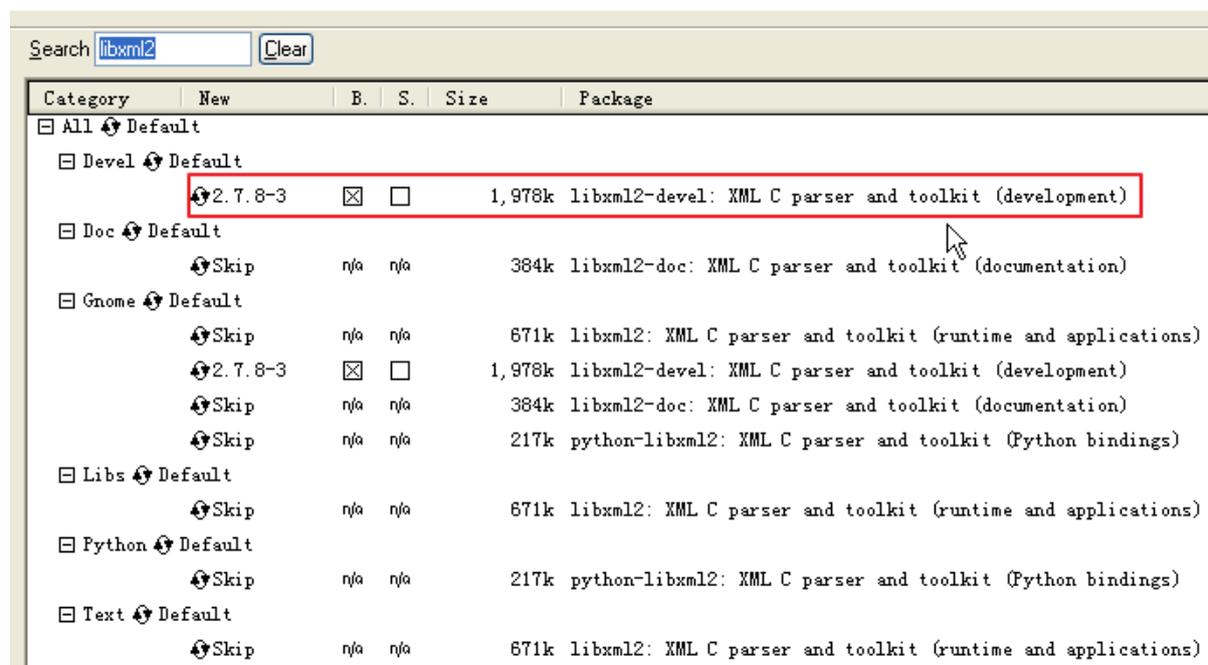
多数人都会被cygwin里面太多的模块而搞晕掉，所以其提供了一个Search的功能。

比如，我想要去确定（用于搭建Docbook环境所必须的其中）一个叫做libxml2的模块，

那么就可以去search框中输入libxml2，则其会自动把与此libxml2相关的模块，都列出来：

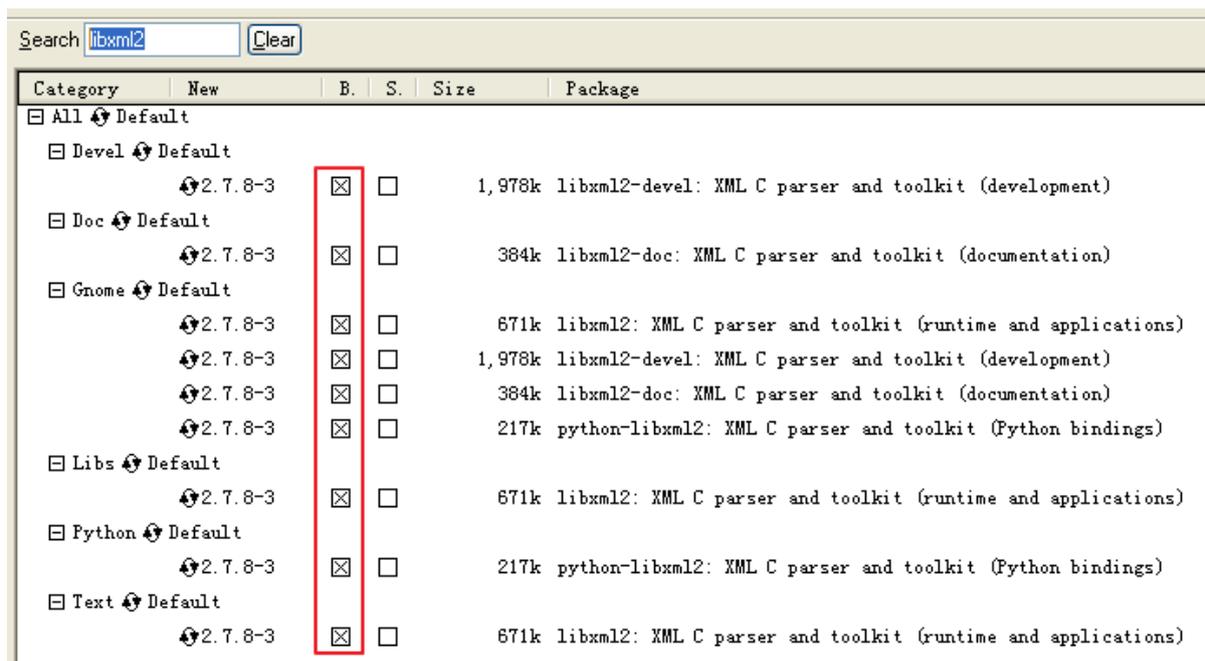


不过由于各个子分类都是加号，没有展开的，所以点击加号，全部展开：



可以看到此处已经安装了Devel中的libxml2-devel这个模块了，其已经基本满足开发需求了。

但是很多时候，我们可能对于剩下的其他的与此libxml2相关的模块，不是很清楚其功能，不知道是否需要安装，那么此时，其实最简单省事的办法就是，把相关的模块，都点击Skip，直到都变成Install：



如此，就不怕漏装某个libxml2的库，而导致以后需要回来重新安装了。

如此，安装自己需求，一点点去选择自己需要的模块，即可。

2.2.7.6. 关于安装Cygwin模块方面的最简单，最核心的提示

还是那句话，对于新手，很多不清楚各个模块的作用：

那么最简单的做法是，全部都选上。

当然觉得全部都选上，又太浪费下载时间和安装后的空间的话，那么最为开发用途的cygwin，则至少可以把

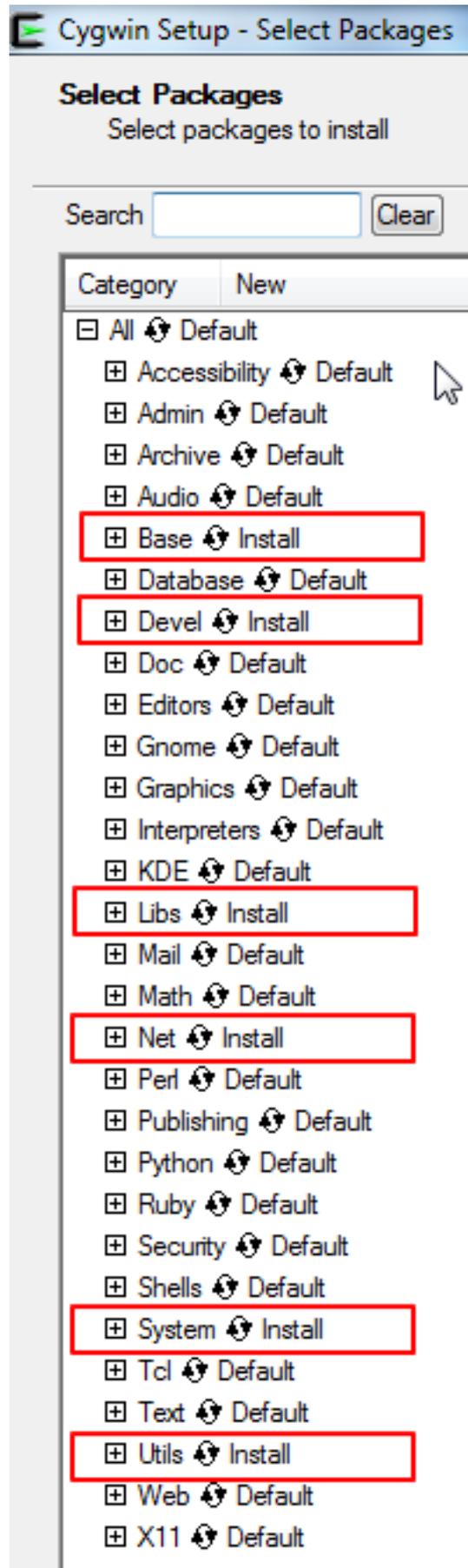
Base , Devel , Libs , Net , System , Utils

等这几个最基本的分类下面的模块都选上。

即点击

Base , Devel , Libs , Net , System , Utils

的Default，使其都变成Install即可：



当然，如果对于这些，你还嫌弃还是太多，那么就只选择Devel，也基本够你日常开发所用了。

2.2.8. 安装所选模块直至完成

选择好需要安装的模块后，就可以点击“下一步”去安装了。

然后后面就很简单了，基本上都是下一步，然后cygwin就会自动去下载并安装对应的模块了。

这个过程所需时间，取决于你所选模块的多少，和连接（前面所设置的163）服务器的速度快慢，决定了下载所用时间，以及最后加上安装所有模块的时间，总体上来说，应该都是以小时为单位，比如2,3个小时之类的时间。

最后安装完毕，会在桌面出现一个cygwin的图标，双击后，第一次运行，其会初始化一下，之后就是正常的使用Linux的环境了。



关于安装到最后有时候会死掉的解决办法

如果遇到了安装到最后，比如95%的时候死掉的问题，可以参考：

[【记录】Cygwin下载过程记录](#)¹⁶



离线安装Cygwin

离线安装Cygwin，即从本地安装，其安装方法，和上面的从setup.exe安装相比，主要的不同点就是选择从本地安装，剩下的过程都类似，就不多解释了。

¹⁶ http://www.crifan.com/records_record_cygwin_download_process/

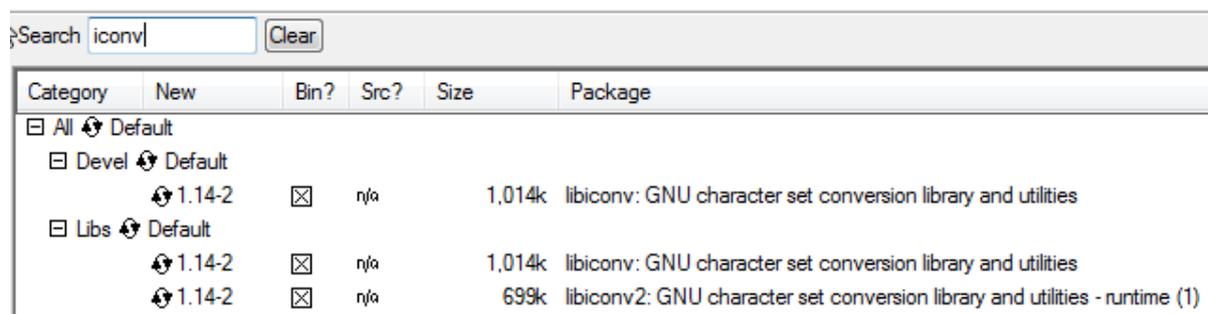
第 3 章 Cygwin中各个模块详解

此处，针对Cygwin中的一些模块，工具，库等，做个简单的介绍。

当然，实际上，这些东西，其实都是Linux中的常见工具，库等，

此处，只是针对我所了解的一些，做个解释，仅供参考：

3.1. iconv



The screenshot shows a search interface with 'iconv' entered in the search box. Below the search bar is a table with columns: Category, New, Bin?, Src?, Size, and Package. The results are grouped by category: All, Devel, and Libs. Under 'Devel', there is one entry for version 1.14-2 with a size of 1,014k. Under 'Libs', there are two entries for version 1.14-2, one with a size of 1,014k and another with a size of 699k.

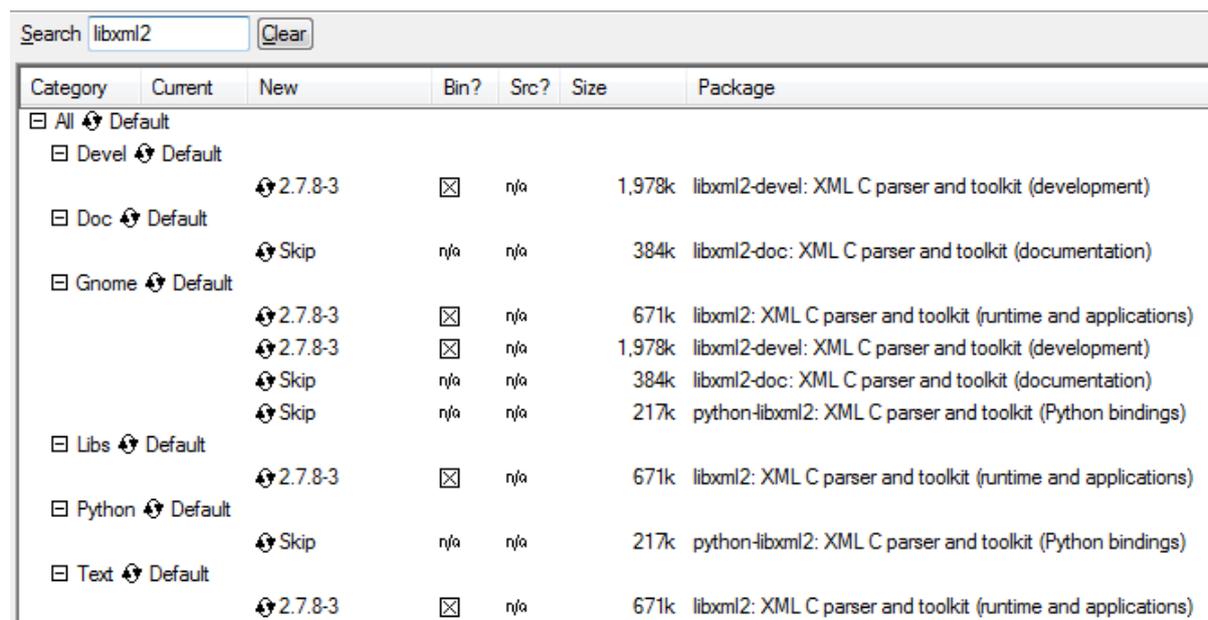
Category	New	Bin?	Src?	Size	Package
[-] All	[-] Default				
[-] Devel	[-] Default				
	1.14-2	<input checked="" type="checkbox"/>	n/a	1,014k	libiconv: GNU character set conversion library and utilities
[-] Libs	[-] Default				
	1.14-2	<input checked="" type="checkbox"/>	n/a	1,014k	libiconv: GNU character set conversion library and utilities
	1.14-2	<input checked="" type="checkbox"/>	n/a	699k	libiconv2: GNU character set conversion library and utilities - runtime (1)

是个针对编码转换的工具，其支持很多常见的编码，包括UTF-8，GB2312等等。

比如我在[生成Docbook的chm格式文件](#)¹的时候，就用到将UTF-8编码的文件，转换为本地编码（GB18030编码）的格式，就是用的iconv这个工具。

详情可参考[iconv](#)²

3.2. libxml2



The screenshot shows a search interface with 'libxml2' entered in the search box. Below the search bar is a table with columns: Category, Current, New, Bin?, Src?, Size, and Package. The results are grouped by category: All, Devel, Doc, Gnome, Libs, Python, and Text. Under 'Devel', there is one entry for version 2.7.8-3 with a size of 1,978k. Under 'Doc', there is one entry for 'Skip' with a size of 384k. Under 'Gnome', there are three entries for version 2.7.8-3 with sizes of 671k, 1,978k, and 384k. Under 'Libs', there is one entry for version 2.7.8-3 with a size of 671k. Under 'Python', there is one entry for 'Skip' with a size of 217k. Under 'Text', there is one entry for version 2.7.8-3 with a size of 671k.

Category	Current	New	Bin?	Src?	Size	Package
[-] All	[-] Default					
[-] Devel	[-] Default					
		2.7.8-3	<input checked="" type="checkbox"/>	n/a	1,978k	libxml2-devel: XML C parser and toolkit (development)
[-] Doc	[-] Default					
		Skip	n/a	n/a	384k	libxml2-doc: XML C parser and toolkit (documentation)
[-] Gnome	[-] Default					
		2.7.8-3	<input checked="" type="checkbox"/>	n/a	671k	libxml2: XML C parser and toolkit (runtime and applications)
		2.7.8-3	<input checked="" type="checkbox"/>	n/a	1,978k	libxml2-devel: XML C parser and toolkit (development)
		Skip	n/a	n/a	384k	libxml2-doc: XML C parser and toolkit (documentation)
		Skip	n/a	n/a	217k	python-libxml2: XML C parser and toolkit (Python bindings)
[-] Libs	[-] Default					
		2.7.8-3	<input checked="" type="checkbox"/>	n/a	671k	libxml2: XML C parser and toolkit (runtime and applications)
[-] Python	[-] Default					
		Skip	n/a	n/a	217k	python-libxml2: XML C parser and toolkit (Python bindings)
[-] Text	[-] Default					
		2.7.8-3	<input checked="" type="checkbox"/>	n/a	671k	libxml2: XML C parser and toolkit (runtime and applications)

¹ http://www.crifan.com/use_docbook_output_htmlhelp_fix_messy_cn_char_of_chm_title_and_toc_index/

² <http://zh.wikipedia.org/wiki/Iconv>

和xml解析相关的核心库，可被很多其他程序，工具用到。比如[docbook所用工具xsltproc](#)³。

详情可参考[libxml2](#)⁴

3.3. libxslt

Category	New	B...	S...	Size	Package
Search <input type="text" value="libxslt"/> <input type="button" value="Clear"/>					
[-] All	[-] Default				
[-] Doc	[-] Default				
	<input type="checkbox"/> Skip	n/a	n/a	189k	libxslt: The GNOME XSLT C library (runtime)
	<input type="checkbox"/> Skip	n/a	n/a	194k	libxslt-doc: The GNOME XSLT C library (documentation)
[-] Gnome	[-] Default				
	<input type="checkbox"/> Skip	n/a	n/a	189k	libxslt: The GNOME XSLT C library (runtime)
	<input type="checkbox"/> Skip	n/a	n/a	35k	libxslt-devel: The GNOME XSLT C library (development)
	<input type="checkbox"/> Skip	n/a	n/a	194k	libxslt-doc: The GNOME XSLT C library (documentation)
	<input type="checkbox"/> Skip	n/a	n/a	40k	python-libxslt: The GNOME XSLT C library (Python bindings)
[-] Interpreters	[-] Default				
	<input type="checkbox"/> Skip	n/a	n/a	189k	libxslt: The GNOME XSLT C library (runtime)
[-] Libs	[-] Default				
	<input type="checkbox"/> Skip	n/a	n/a	189k	libxslt: The GNOME XSLT C library (runtime)
	<input type="checkbox"/> Skip	n/a	n/a	35k	libxslt-devel: The GNOME XSLT C library (development)
	<input type="checkbox"/> Skip	n/a	n/a	40k	python-libxslt: The GNOME XSLT C library (Python bindings)
[-] Python	[-] Default				
	<input type="checkbox"/> Skip	n/a	n/a	40k	python-libxslt: The GNOME XSLT C library (Python bindings)
[-] Text	[-] Default				
	<input type="checkbox"/> Skip	n/a	n/a	189k	libxslt: The GNOME XSLT C library (runtime)

xslt是一种语言，用来描述如何将xml转换为别的格式的文件。此库常被别的程序所用到，比如：[docbook所用工具xsltproc](#)⁵。

3.4. zlib

Category	Current	New	B...	S...	Size	Package
Search <input type="text" value="zlib"/> <input type="button" value="Clear"/>						
[-] All	[-] Default					
[-] Base	[-] Default					
	<input checked="" type="checkbox"/> 1.2.5-1	<input checked="" type="checkbox"/>	n/a		44k	zlib0: The zlib compression/decompression library (runtime)
[-] Devel	[-] Default					
	<input type="checkbox"/> Skip	n/a	n/a		69k	zlib-devel: The zlib compression/decompression library (development)
[-] Libs	[-] Default					
	<input type="checkbox"/> Skip	n/a	n/a		31k	zlib: The zlib compression/decompression library (doc)
	<input type="checkbox"/> Skip	n/a	n/a		69k	zlib-devel: The zlib compression/decompression library (development)
	<input checked="" type="checkbox"/> 1.2.5-1	<input checked="" type="checkbox"/>	n/a		44k	zlib0: The zlib compression/decompression library (runtime)

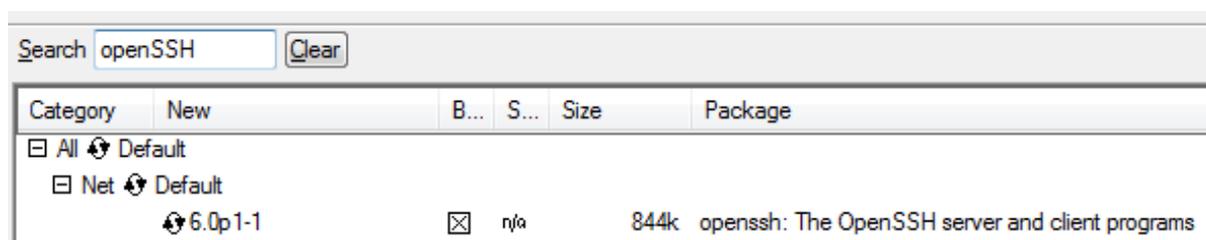
³ http://www.crifan.com/files/doc/docbook/docbook_dev_note/release/html/docbook_dev_note.html#download_xsltproc_win

⁴ <http://zh.wikipedia.org/wiki/Libxml2>

⁵ http://www.crifan.com/files/doc/docbook/docbook_dev_note/release/html/docbook_dev_note.html#download_xsltproc_win

最常用的压缩工具之一。

3.5. openSSH



SSH是用来建立加密的安全连接的。相对于普通的非加密链接，提高了安全性。而此openssh是个开源免费的，用的最广泛的一个ssh的实现。

第 4 章 Cygwin使用心得和技巧总结

4.1. 和Cygwin自身相关的一些工具

此处主要指的是和Cygwin本身有关的一些工具。

关于Cygwin的工具，其实cygwin官网中已有详细介绍了：[Cygwin Utilities](#)¹，此处只是简单解释一下，以及提醒一些注意事项：

4.1.1. cygcheck

作用：用于检测，校验cygwin。

如果cygwin本身的运行有问题了，比如缺少了某库，导致程序无法运行，则可以使用此工具，去检测到底问题出在哪里。

比如：[【基本解决】Cygwin中运行程序出现错误：error while loading shared libraries: ?: cannot open shared object file: No such file or directory](#)²

4.1.2. cygpath

作用：实现Linux/Unix和Windows之间的路径转换。

其中，主要包含了Windows路径中反斜杠和Linux路径中的斜杠之间的转换，已经Windows下面的环境变量和Linux下面的环境变量的处理。

而环境变量中，又包含，多个值所用分隔符的不同，Windows是分号，Linux是冒号。

不过好像当Linux和Windows的环境变量混用，有时候，会出现一些bug，比如

[【终极解决】fop错误：Exception in thread "main" java.lang.NoClassDefFoundError:org/apache/xmlgraphics/image/loader/ImageContext 的终极解决办法，即cygpath有bug，转换路径出错，导致部分路径被截断](#)³

4.2. Cygwin中使用注意事项

Cygwin的使用过程中，也有很多注意事项，现总结如下；

4.2.1. Cygwin的环境变量等路径中可能包含空格，括号等特殊字符而导致出错

Linux的环境中，路径中，是不允许包含空格的

Windows的环境中是运行的

所以，在使用Cygwin期间，常常会遇到：

PATH等环境变量，由于即包含了Windows中的很多路径

又包含了Cygwin（即Linux）中的很多路径

¹ <http://cygwin.com/cygwin-ug-net/using-utils.html>

² http://www.crifan.com/cygwin_run_app_error_while_loading_shared_libraries_cannot_open_shared_object_file_no_such_file_or_directory/

³ http://www.crifan.com/fop_java_lang_noclassdeffounderror_xmlgraphics_cygpath_bug_path_truncated/

但是程序，脚本等处理这些路径时，有些没考虑到这种带路径的情况而导致出现各种问题。

比如，我之前就遇到一个：

[【已解决】Cygwin下为xscale编译Buildroot时make出错：/bin/sh: -c: line 0: syntax error near unexpected token `\('](#)⁴

所以，以后使用时，在这方面，要小心。

以及出现类似问题，要想到，或许可能是Cygwin路径中包含空格，括号等特殊字符而导致的

4.3. Cygwin使用技巧总结

下面介绍一些Cygwin中相关的小技巧：

4.3.1. Windows下，添加右键打开Cygwin

打开注册表（regedit）并定位到

HKEY_CLASSES_ROOT\Directory\shell



关于注册表regedit

不熟悉windows注册表regedit的，请去看

[Windows的注册表编辑器: regedit](#)⁵

然后右击shell->新建->项，然后名字随便起，比如RunCygwin

在其下再新建一项，叫做command，表示要执行的命令，command下会有一个图标是ab字样的键值，名称是(默认)，类型是REG_SZ，

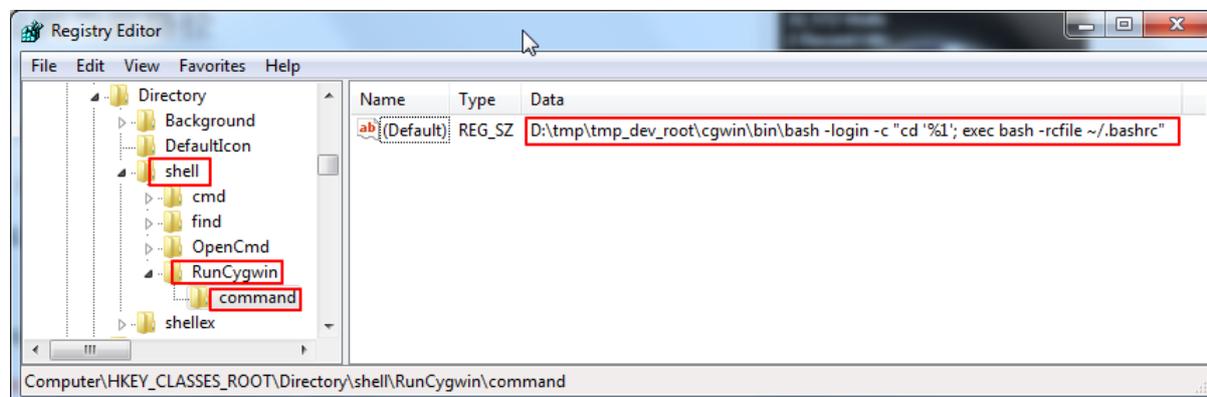
双击其会弹出一个“修改字符串”的窗口，修改数据数值为：

```
D:\tmp\tmp_dev_root\cygwin\bin\bash -login -c "cd '%1'; exec bash -rcfile ~/.bashrc"
```

其中

D:\tmp\tmp_dev_root\cygwin\

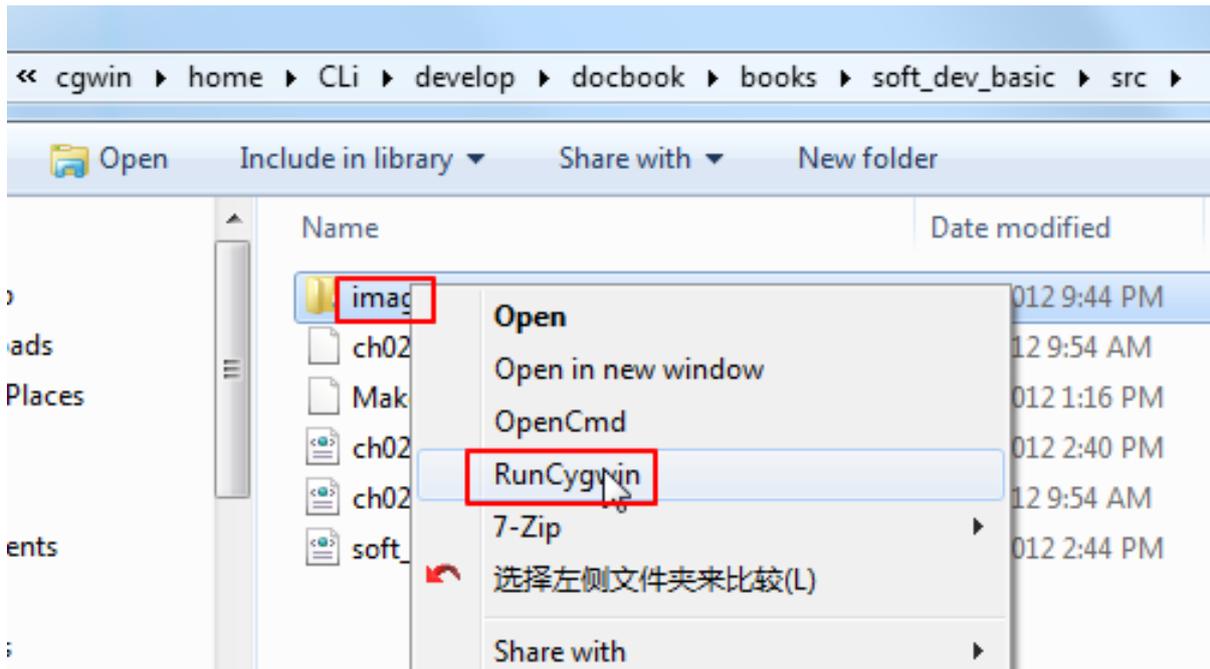
是我此处Cygwin安装路径下的bash的位置，其下会存在bash工具：bin\bash



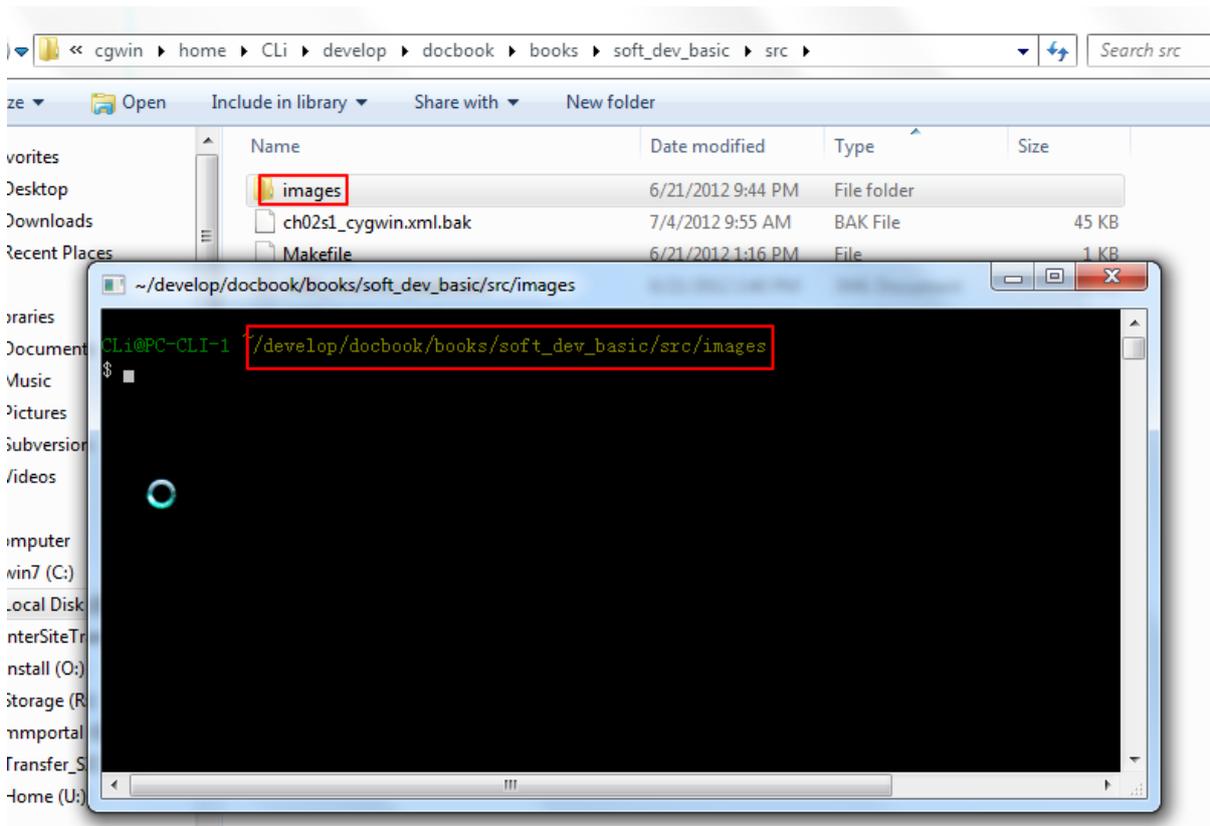
⁴

http://www.crifan.com/cygwin_buildroot_xscales_make_bin_sh_c_line_0_syntax_error_near_unexpected_token_left_parenthesis/
⁵http://www.crifan.com/files/doc/docbook/soft_dev_basic/release/htmls/windows.html#windows_regedit

然后就可以进入任何目录,右键点击,就会看到对应的可选项,此处是RunCygwin :



点击即可进入cygwin了 :



关于更多的相关的解释, 请参考: [【整理】在windows下添加右键使得可以运行cygwin且直接定位到当前文件夹](#) + [【整理】如何在Windows XP中, 右键添加命令行cmd command, 打开当前文件夹, 定位到当前路径](#)⁶

⁶

http://www.crifan.com/finishing_in_the_windows_makes_it_possible_to_add_the_right_to_run_cygwin_and_navigate_directly_to_the_current_folder_finishing_in_w...
[add_command_line_cmd_command_open_the_current_folde/](#)

4.3.2. 修改Cygwin的默认启动路径

原先启动Cygwin，其默认路径是当前用户的Home文件夹：

```
CLi@PC-CLI-1 ~  
$ pwd  
/home/CLi
```

而Cygwin默认启动路径，就是由环境变量HOME的值决定的。

因此，此处就是要去改变对应的环境变量HOME的值。

此处，先去看看原先HOME的值是多少，有两种方法：

- 可以在cygwin下输入env查看当前所有的环境变量：

```
CLi@PC-CLI-1 ~  
$ env  
HOMEPATH=\Users\CLi  
...  
SHELL=/bin/bash  
TERM=xterm  
...  
CommonProgramFiles(x86)=C:\Program Files (x86)\Common Files  
OS=Windows_NT  
...  
HOME=/home/CLi  
...
```

其中就可以看到，此处的HOME值为/home/CLi

- 直接通过echo一个环境变量，得知该环境变量的值：

```
CLi@PC-CLI-1 ~  
$ echo $HOME  
/home/CLi
```

而想要改变HOME的值，也有两种方法：

- 去windows中建立一个HOME的环境变量，将值设置为自己想要的值

而关于如何添加Windows的环境变量，不熟悉的人，请参考

[windows中新建一个环境变量](#)⁷

- 通过export去修改环境变量：

```
CLi@PC-CLI-1 ~  
$ echo $HOME  
/home/CLi
```

⁷ http://www.crifan.com/files/doc/docbook/soft_dev_basic/release/htmls/env_var.html#add_new_win_env_var

```
CLi@PC-CLI-1 ~  
$ export HOME=/home/CLi/develop  
  
CLi@PC-CLI-1 /home/CLi  
$ echo $HOME  
/home/CLi/develop
```

即可把HOME从/home/CLi改为/home/CLi/develop



注意

此处通过export的方式改变的值，只是当前（会话/窗口）有效，关闭了窗口后，就失效了，所以下次重启后，还是无法达到修改的效果

4.3.3. 切换当前路径到别的磁盘，比如C盘，D盘等

Cygwin下，从当前路径切换到Windows中别的磁盘，比如C盘，之前的做法，是通过cygdrive：

```
CLi@PC-CLI-1 /cygdrive/c  
$ pwd  
/cygdrive/c  
  
CLi@PC-CLI-1 /cygdrive/c  
$ cd /cygdrive/d  
  
CLi@PC-CLI-1 /cygdrive/d  
$ pwd  
/cygdrive/d
```

现在刚看到[\[1\]](#)，才知道也可以直接通过C:的形式去切换到C盘了：

```
CLi@PC-CLI-1 /home/CLi  
$ pwd  
/home/CLi  
  
CLi@PC-CLI-1 /home/CLi  
$ cd C:  
  
CLi@PC-CLI-1 /cygdrive/c  
$ pwd  
/cygdrive/c
```

第 5 章 Cygwin中常见错误及解决办法

5.1. Cygwin中常见错误及解决办法

5.1.1. Cygwin下的文件可以编辑但是无法保存

之前在折腾：

[【记录】Cygwin下用crosstool-ng为arm制作交叉编译器](#)¹

期间，就遇到这种问题：

[【已解决】Win7中用Notepad++编辑cygwin下面的文件结果保存出错：请检查文件是否在别的程序中打开](#)²

具体现象是：

Cygwin下面的文件，是可以其他编辑工具，比如[Notepad++](#)³，[Sublime](#)⁴等去编辑的

但是，编辑之后，去保存的时候，都会出错，都无法正常保存。

经过一番折腾，结果找到了根本原因：

Cygwin下面的有些文件，其文件的拥有者（owner），不是Win7中当前的用户

导致有些Cygwin下面的文件，在Win7下面，会没有修改删除等权限。

所以，对应的解决办法就是：

给Win7的当前用户，增加操作这些文件的权限。

其实现起来也有各种办法，相对最省事，最常用的办法是：

Win7的资源管理器中，右键，cygwin中那些没有操作权限的文件所在的文件夹

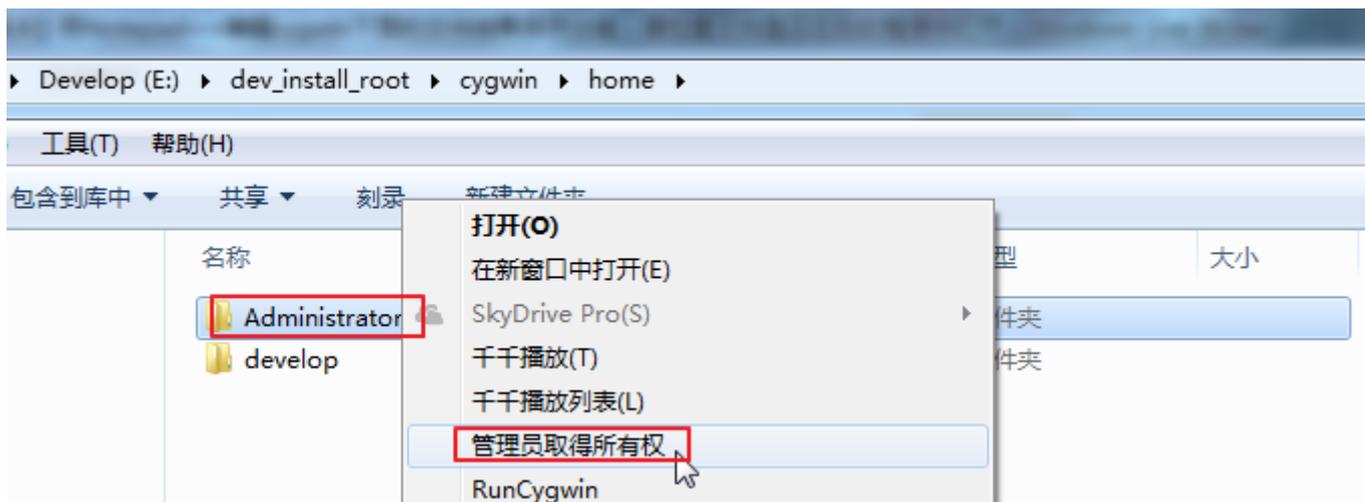
然后选择：“管理员取得所有权”

¹ http://www.crifan.com/cygwin_use_crosstool_ng_build_cross_compiler_for_arm/

² http://www.crifan.com/notepad_plus_plus_edit_file_under_cygwin_error_please_check_file_opened_by_another_application/

³ http://www.crifan.com/files/doc/docbook/rec_soft_npp/release/html/rec_soft_npp.html

⁴ http://www.crifan.com/recommend_a_relative_good_text_editor_sublime_2/



就会调出命令行，给该文件夹下所有的文件，都加上当前用户，即管理员，的所有权限。

然后就可以在Win7下编辑文件，保存文件了。

第 6 章 与Cygwin相关的东西

6.1. 在Cygwin下搭建Docbook开发环境

如之前教程：

[Windows环境下的，基于Cygwin的Docbook开发环境的搭建](#)¹

介绍的一样，其实是在Windows下的Cygwin中，去搭建Docbook的环境，效果会比纯Windows环境下Docbook环境，更好，功能更强大，但是当然，也更加复杂。

6.2. 在Cygwin下编译crosstool-ng

crosstool-ng支持Cygwin环境，也可以在Cygwin下编译crosstool-ng：

[支持多种主机编译环境：各种Linux发行版，Cygwin等](#)²

6.3. 在Cygwin下编译Buildroot

Buildroot，截止目前（2013-09-03），官网是不支持Cygwin环境的

但是，经过一番研究后发现，现在最新版本的Buildroot，好像已经有人成功在Cygwin上编译了。

所以，后来自己也就去折腾了，感兴趣的可以去看看：

[【记录】在Cygwin上用Buildroot为xscale建立交叉工具链](#)³

¹ http://www.crifan.com/files/doc/docbook/docbook_dev_note/release/htmls/cygwin_docbook_dev_env.html

² http://www.crifan.com/files/doc/docbook/crosstool_ng/release/html/crosstool_ng.html#ctng_feature_linux_and_cygwin

³ http://www.crifan.com/try_under_cygwin_compile_buildroot_for_xscale_arm_cross_toolchain_compiler/

参考书目

[1] [Cygwin中环境变量的配置](#)¹

[2] [【记录】Cygwin下用crosstool-ng为arm制作交叉编译器](#)²

¹ <http://hi.baidu.com/%CE%DE%CA%AB%CE%C4/blog/item/099c1766c27fd822ab184cb5.html>

² http://www.crifan.com/cygwin_use_crosstool_ng_build_cross_compiler_for_arm/