[1. 微软产品中的R2是什么意思 2](#_Toc465781454)

[2. Windows SDK 4](#_Toc465781455)

[2-1. Visual Studio 2015 update1 的安装盘里有Windows SDK 的目录 4](#_Toc465781456)

[2-1-1. VS2015 安装时会提示安装Windows SDK 4](#_Toc465781457)

[2-2. SDK Installation Location 5](#_Toc465781458)

[2-2-1. Tools in NETFX 4.X Tools 6](#_Toc465781459)

[3. Little-Endian Big-Endian 6](#_Toc465781460)

[3-1. 指的是CPU结构的事情，不是操作系统或者文件 6](#_Toc465781461)

[3-2. 具体内容 6](#_Toc465781462)

[3-3. 数字的高位地位表示和在文件中的位置 7](#_Toc465781463)

[3-4. 和一般数字不同的是 数字和字节 8](#_Toc465781464)

[4. Windows 注册表 8](#_Toc465781465)

[4-1. 注册表其实是个数据库 8](#_Toc465781466)

[4-2. 注册表的位置 8](#_Toc465781467)

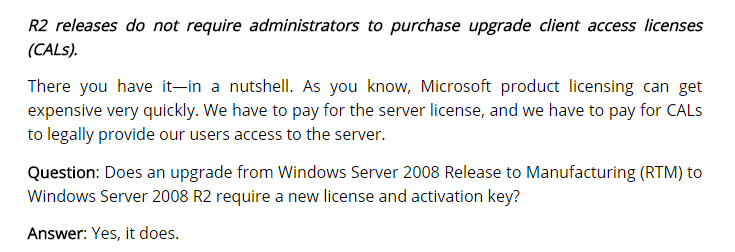
# 微软产品中的R2是什么意思

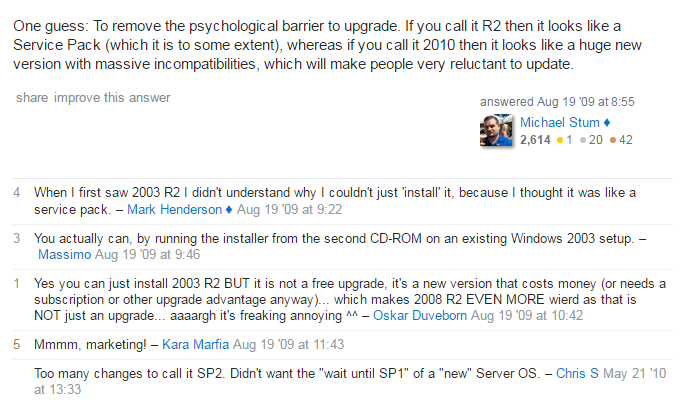
Release 2， 也就是第二次release 的意思。 并不仅仅是一个SP， service pack， 更接近于是一个New Version。 只所以叫R2 而不是SP2， 可能的改动内容比一个SP要多，但是又没有一个通常意义上的New Version 那么多，所以就叫R2 而不是SP2了。比如SQL Server 2008 R2 其实也叫SQL Server 2010

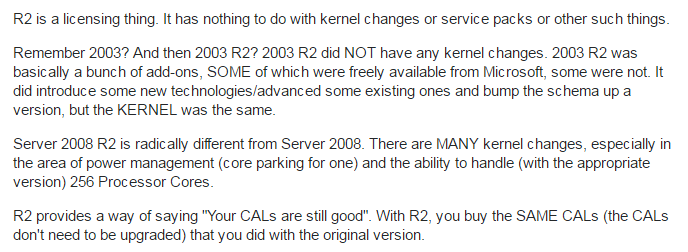
从文件大小上来说，SP2 是一个升级包，所以最多也就几十，上百M吧。而R2从形式上一个完整的新的Release，所以一般也得有几个G

也有一个说法是系统管理员一般安装SP 时比较害怕，因为担心会损坏现有的系统。但是R2既然是一个全新的Release，那么管理员在安装时一般会踏实一些。

从Licence的角度，虽然从第一版升级到R2需要购买新的Key，但是访问访问server的客户端的license CAL 并不需要升级和重新购买，这样也会比直接Release 一个New Version 对客户来说更经济一些。



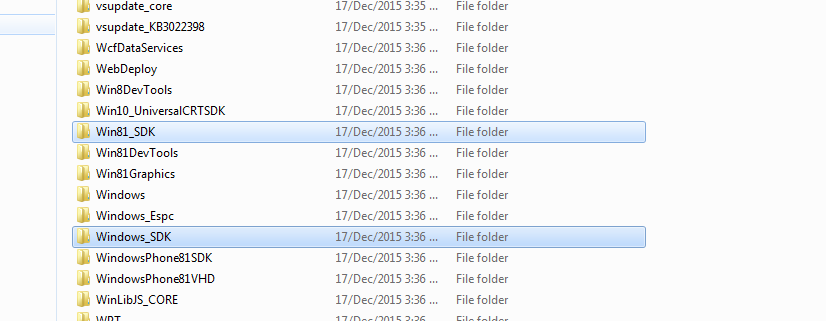




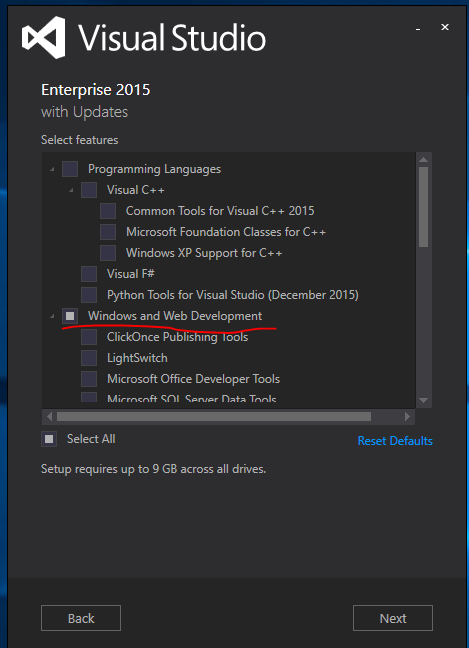
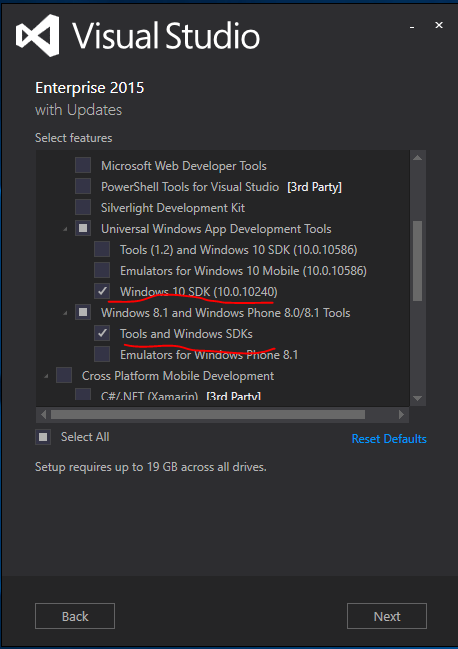
可以参考[What does the ‘R2’ mean in Windows Server 2008 R2](docs/What%20does%20the%20‘R2’%20mean%20in%20Windows%20Server%202008%20R2.pdf)

# Windows SDK

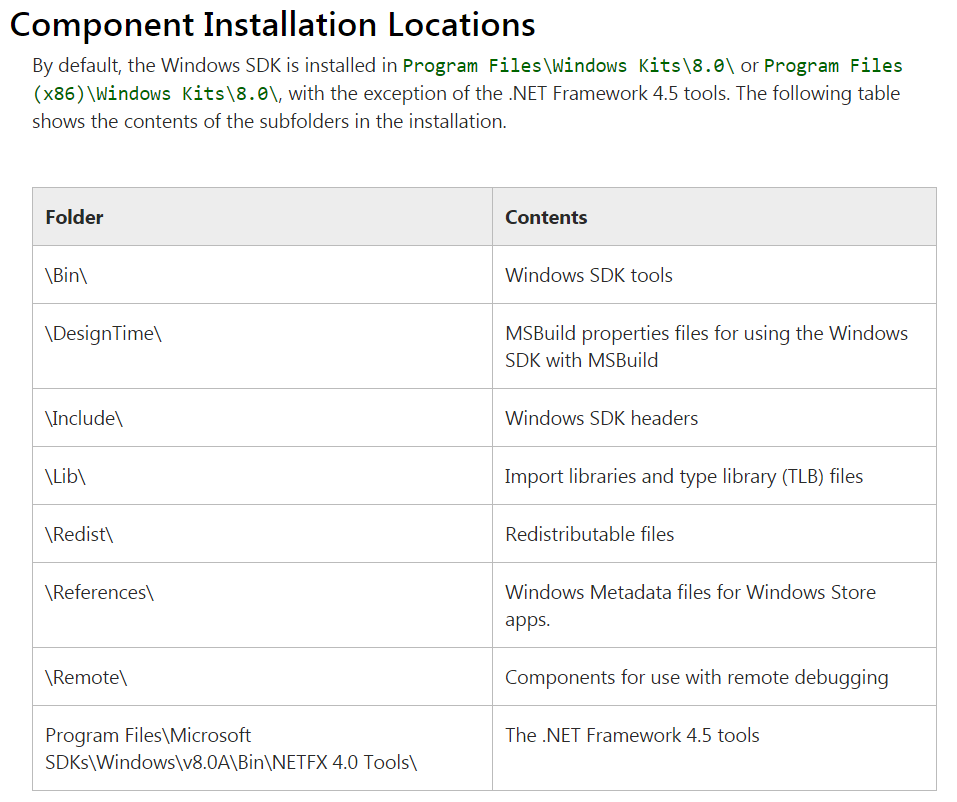
## Visual Studio 2015 update1 的安装盘里有Windows SDK 的目录



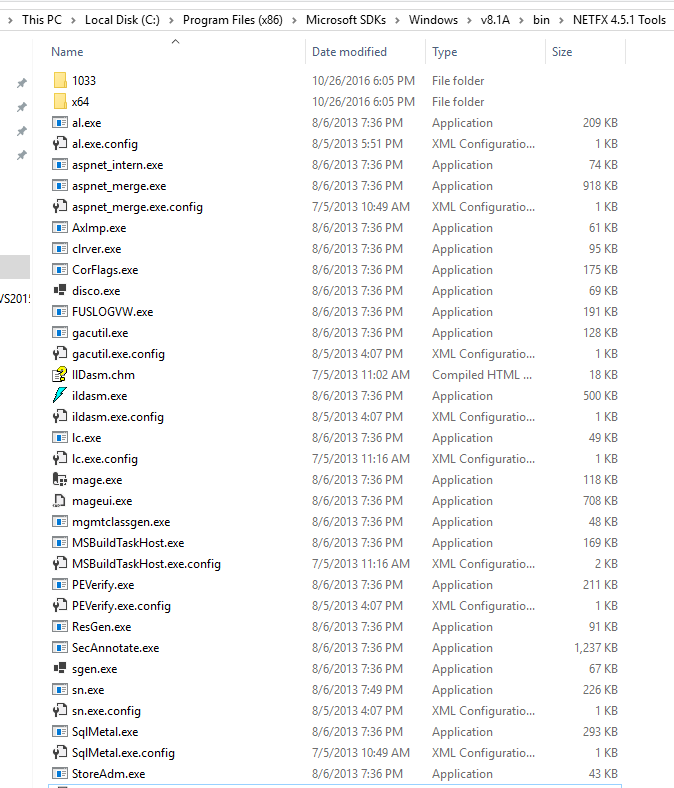
### VS2015 安装时会提示安装Windows SDK

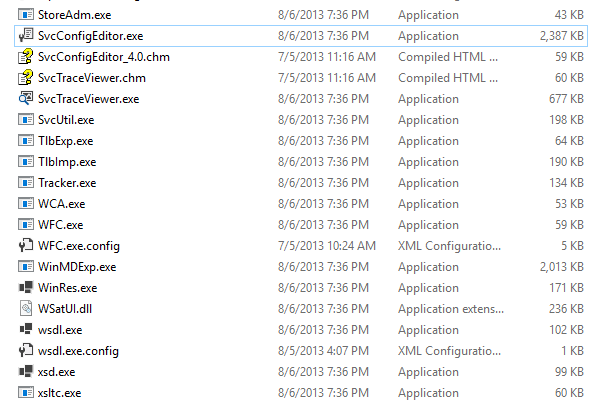
 

## SDK Installation Location



### Tools in NETFX 4.X Tools





# Little-Endian Big-Endian

## 指的是CPU结构的事情，不是操作系统或者文件

所以Little Endian还是Big Endian 的全称应该是CPU是Little Endian还是Big Endian。

## 具体内容

http://blog.163.com/huang\_bp/blog/static/123119837201081611131341

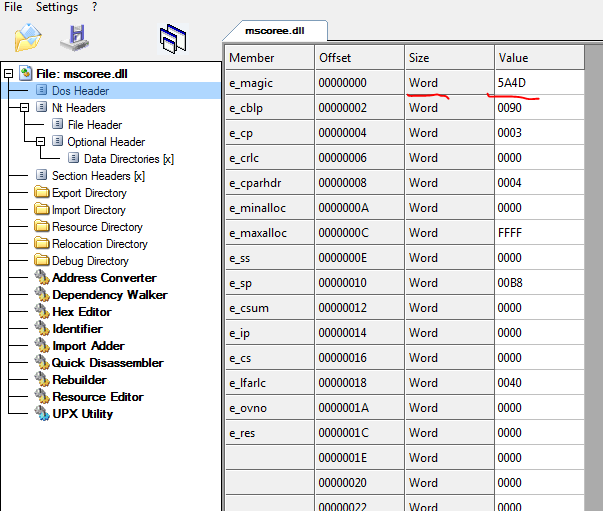
计算机的所有内存以字节数组的方式进行编址。

当一个逻辑上长于一个字节的整形数据放置在内存中时（比如16位，32位，和64位的整数），需要考虑这些字节的存储顺序。Little Endian: 将字节的逻辑顺序与物理顺序一致，即将逻辑上较低的字节放置在物理上较低的字节上，比如Intel x86系列；

Big Endian字节的逻辑顺序与物理顺序相反，即将逻辑上较低的字节放置在物理上较高的字节上，比如Motorola的PowerPC以及Sun Sparc。还有一些平台同时支持两种方案，由开发者决定使用哪一种。

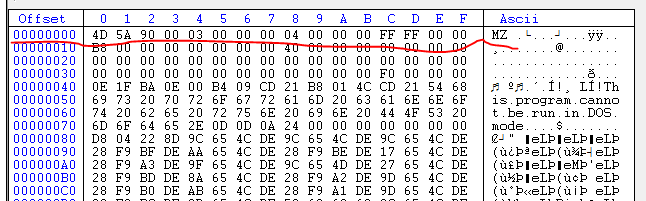
## 数字的高位地位表示和在文件中的位置

一般我们读一个数的时候，如一个16位的Word 值5A4D（代表字符MZ），都是左边的是逻辑高位，右边的是逻辑地位，5A4D的完整表示应该是0x5A4D



当我们看以HEX的形式看一个文件时，最左边是地位地址，右边是高位地址

从00000000 到0000000F如下：



所以0x5A4D 在文件中就显示成了4D 5A。

所以在文件中要读一个数字，不能简单把几个byte连接起来。首先要知道这个数字的长度，byte，word ，int， int64， 然后找到最地位和最高位的byte，然后然后逻辑高位到逻辑地位拼接出来。

## 和一般数字不同的是 数字和字节

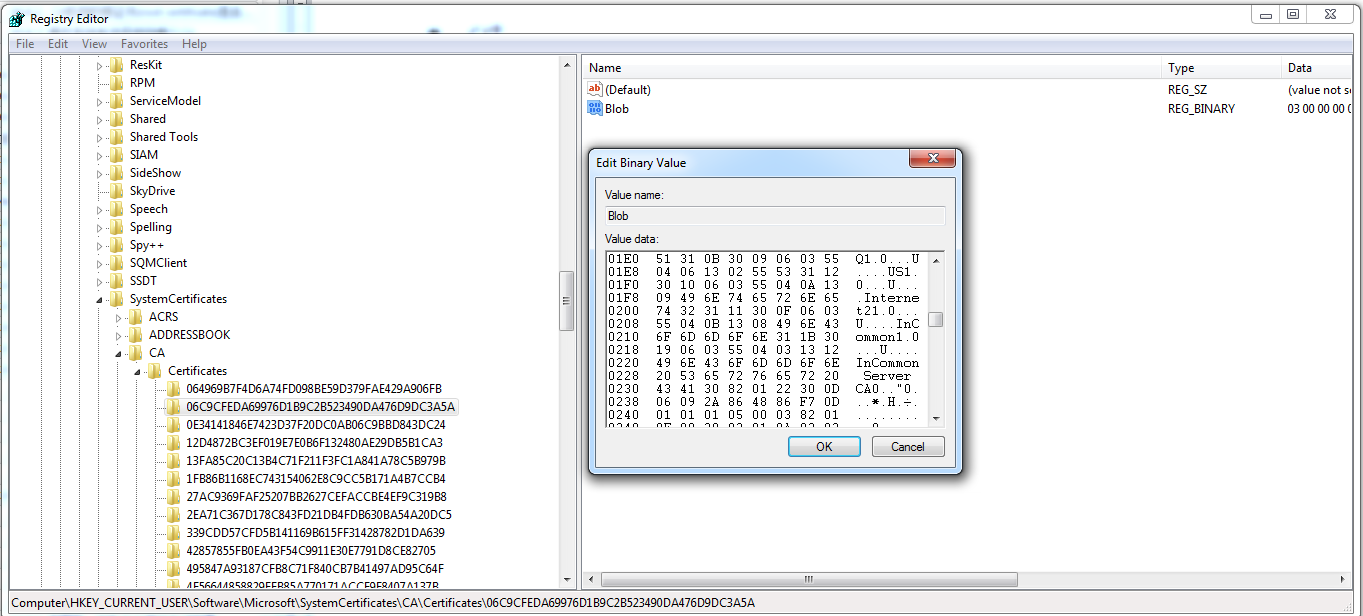
对一个十进制的数字来说，它的每一个位置就一个数字

对于little-endian 和 big-endian来说，安排的是一个字节的顺序。在十六进制表示下，一个字节由两个十六进制字符组成，这两个字符是不存在little-endian 和big-endian的问题的，永远都是高位在前，低位在后

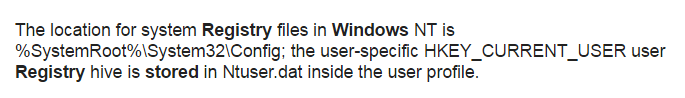
# Windows 注册表

## 注册表其实是个数据库

比如客户安装的数字证书就是直接存在注册表里面。对应的数据类型是Blob



## 注册表的位置



<http://www.intowindows.com/location-of-registry-files-in-windows-7810/>

