



# System Management Fundamentals

使用 **Windows** 中的子系统 - **Linux**

Get started with the Windows  
Subsystem for Linux

May 2021

Microsoft Reactor | Ryan Chung

```
led by player to  
s.load_image("kg.png")  
  
[self]:  
    initialize Dog object and create Text of  
g, self).__init__(image = Dog.image,  
                    x = games.mouse.x,  
                    bottom = games.screen  
                    re = games.Text(value = 0, size = 24,  
                                     top = 5, right = game  
                                     screen.add(self.score)  
1 = games.Text(value = 0, size = 24,  
                 top = 5, left = game
```



# Ryan Chung

Instructor / DevelopIntelligence  
Founder / MobileDev.TW

@ryanchung403 on WeChat  
Ryan@MobileDev.TW







# Reactor



[developer.microsoft.com/reactor/](https://developer.microsoft.com/reactor/)  
@MSFTReactor on Twitter



# 适用于 Linux 的 Windows 子系统入门

10 分钟 剩余 · 模块 · 已完成 7 个单元, 共 8 个

★★★★★ 4.7 (431)

对其进行评级

初级

开发人员

Windows

Visual Studio Code

了解如何启用适用于 Linux 的 Windows 子系统 (WSL)、安装最常用 Linux 发行版本、使用 Visual Studio Code 创建集成开发环境等。

## 学习目标

在本模块中, 你将:

- 在 Windows 设备上启用适用于 Linux 的 Windows 子系统。
- 安装 Linux 发行版本。
- 使用 Linux 命令, 并跨 Windows 和 Linux 文件系统工作。
- 使用在 WSL 上运行的 Node.js 创建网站。
- 使用 Visual Studio Code 设置开发环境。
- 调试 Node.js Express 应用。
- 管理多个 Linux 发行版本。

## 先决条件

- 运行最新版 Windows 10 的计算机。若要验证运行的版本是否为最新版本, 请依次转到“设置”>“Windows 更新”。

# 学习目标

- 启用Windows子系统-Linux (WSL)
- 安装Linux发行版
- 使用Linux指令
- Node.js on WSL
- VS Code开发环境设定
- Node网站侦错
- 管理多重Linux发行版

# 比较习惯用Linux架站?

- WSL – **W**indows **S**ubsystem for **L**inux
  - 可以自行选择喜好的发行版
  - 透过Bash直接存取Windows中的资料
  - Bash与PowerShell并存
- 常见应用
  - Node.JS
  - Ruby on Rails
  - Django & Python

# 启用WSL

- 以系统管理员身分启用命令提示字元
- 输入pwsh进入Power Shell
- 输入以下指令，并准备重新开机
  - `Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux`







# 安装任一个Linux发行版

- 依据你的需求，下载合适的Linux发行版
- 打开Microsoft Store，搜寻Linux

下列項目的結果： Linux

部門 所有部門 適用平台 PC

應用程式 (142) [顯示全部](#)

					
Ubuntu ★★★★★ 27 🖥️	Kali Linux ★★★★★ 3 🖥️	Ubuntu 20.04 LTS 🖥️	Ubuntu 18.04 LTS ★★★★★ 10 🖥️	SUSE Linux Enterprise Server 1... 🖥️	Linux Cheatsheet 🖥️ 📱
免費	免費	免費	免費	免費	免費+



# 安装任一个Linux发行版

- 本教学范例以Ubuntu为例，按下取得



## Ubuntu

Canonical Group Limited • 開發人員工具 > 公用程式

★★★★★ 27 [分享](#)

Ubuntu on Windows allows you to use Ubuntu Terminal and run Ubuntu command line utilities including bash, ssh, git, apt and many more.

[更多](#)



一般

普遍級

免費

[取得](#)

[♥ 願望清單](#)

# 安装任一个Linux发行版

- 下载完成后，按下启动
- 设定使用者名称与密码

 已安裝此產品。



## Ubuntu

Canonical Group Limited • 開發人員工具 > 公用程式

★★★★★ 27  分享

Ubuntu on Windows allows you to use Ubuntu Terminal and run Ubuntu command line utilities including bash, ssh, git, apt and many more.

[更多](#)

 一般  
普遍級

免費

取得

 願望清單

啟動 ...

# 开始使用并更新

- 更新作业系统
  - `sudo apt update && sudo apt upgrade`
- 启动套件安装功能
  - `sudo apt-get update`

# Linux指令测试

- 显示目前工作路径

`pwd`

- 列出目前路径档案

`ls`

`ls -a` (显示隐藏档案)

- 确认在使用者工作路径下，建立LearnWSL资料夹

`cd /home/user-name`

`mkdir LearnWSL`

- 进入该资料夹

`cd LearnWSL`

# 建立网页

- 新增网页档案

`touch wsl-rocks.html`

- 编辑网页档案

`nano wsl-rocks.html`

- 编辑内容如下

```
<!DOCTYPE html>
<html>
  <body style="background-color: tan; font-family: calibri; text-align: center">
    <h1 style="color: blue">My First Heading</h1>
    <p>My first paragraph.</p>
    
  </body>
</html>
```



# 建立网页

- 确认网页档案

cat wsl-rocks.html

```
~/LearnWSL$ cat wsl-rocks.html
<!DOCTYPE html>
<html>
  <body style="background-color:tan;font-family:calibri;text-align:center">
    <h1 style="color:blue">My First Heading</h1>
    <p>My first paragraph.</p>
    
  </body>
</html>
```

# Windows & Linux

- 在Windows中使用Linux指令
  - cmd启动windows命令提示字元
  - 直接打ls无法使用
  - 输入wsl ls可以使用
  - 只打wsl 可以进入到Linux的命令列介面
- 在Windows PowerShell中进入Ubuntu
  - Windows命令提示字元输入pwsh进入PowerShell
  - 在PowerShell中输入bash.exe可进入Ubuntu
- 在Linux中存取Windows的资料
  - 在Ubuntu的命令指令列中
  - 输入 `cd mnt/c`
  - 输入ls 可看见在Windows中C磁碟机中的档案

# 新增档案至网页资料夹

- 在Linux中，安装locate套件

```
sudo apt install mlocate
```

- 更新搜寻资料库

```
sudo updatedb
```

- 使用locate指令找寻网页资料夹

```
locate LearnWSL
```

- 回到该资料夹

```
cd /home/User-name/LearnWSL
```

- 建立资料夹

```
mkdir temp
```

# Windows & Linux 指令互通

- 在Linux中启动Windows的应用程式
  - 在Ubuntu指令列中，输入 `notepad.exe` 启用记事本
  - 在Ubuntu指令列中，输入 `ipconfig.exe` 查看网路设定
  - 在Ubuntu指令列中，输入 `cmd.exe` 进入Windows指令列
- 接着在Windows指令列下
  - `cd ../..` 回到根目录
  - `dir` 列出当前目录中档案与资料夹
  - `dir | findstr Users` 只列出Users资料夹
  - `exit` 离开Windows指令列

# Windows & Linux 指令混用

- 在Linux介面中  
执行ipconfig，但只看有IPv4的内容  
ipconfig.exe | grep IPv4

```
~/LearnWSL$ ipconfig.exe | grep IPv4  
Autoconfiguration IPv4 Address. . : 169.254.97.6  
IPv4 Address. . . . . : 192.168.0.99
```



# 在Ubuntu中使用Python

- 启动Ubuntu
- 确认Python版本  
`python3 --version`
- 测试使用Python  
`python3`  
`print("Hello World")`  
`exit()`

# 在Ubuntu中启动Web Server

- 进入网页资料夹

`cd /home/User-name/LearnWSL`

- 使用Python启动Web Server

`python3 -m http.server`

- 打开浏览器进行测试

<http://localhost:8000/wsl-rocks.html>

- 关闭伺服器

`Ctrl + C`



# 在Ubuntu中使用Node.js

- 更新套件清单、升级套件版本

`sudo apt update`

`sudo apt upgrade`

- 安装curl

`sudo apt install curl`

- 安装node

- `sudo apt-get install -y node.js`

- `-y` 代表安装过程中所有问题都回答Yes

- 安装Express

- `sudo npm install express-generator -g`

# 确认安装

- 确认node安装  
`node --version`
- 确认Express安装  
`express --version`

```
~/LearnWSL$ node --version  
v12.22.1  
~/LearnWSL$ express --version  
4.16.1
```

# 测试Node

- 启动node

在Ubuntu指令列中，输入node

- 查看作业系统种类

os.platform()

- 查看CPU架构

os.arch()

- 查看各CPU资讯

os.cpus()

- 离开

.exit



# 在Linux上设置一个Express网站架构

- 使用Express指令产生网站

`express HelloWorld --view=pug`

- 确认产生的内容

`ls HelloWorld`

- 进入网站资料夹

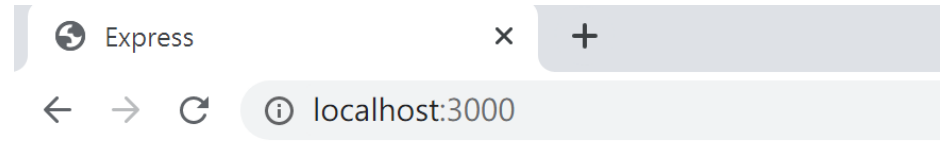
`cd HelloWorld`

- 安装套件

`sudo npm install`

- 启动网站

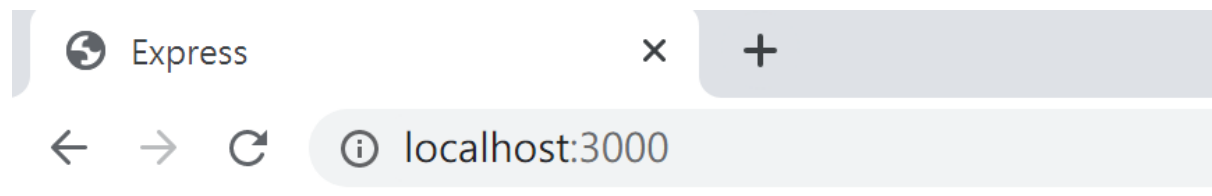
`DEBUG=HelloWorld:* npm start`



## Express

Welcome to Express

# 在浏览器中确认网站



# Express

Welcome to Express

<http://localhost:3000/>

# VS Code & WSL

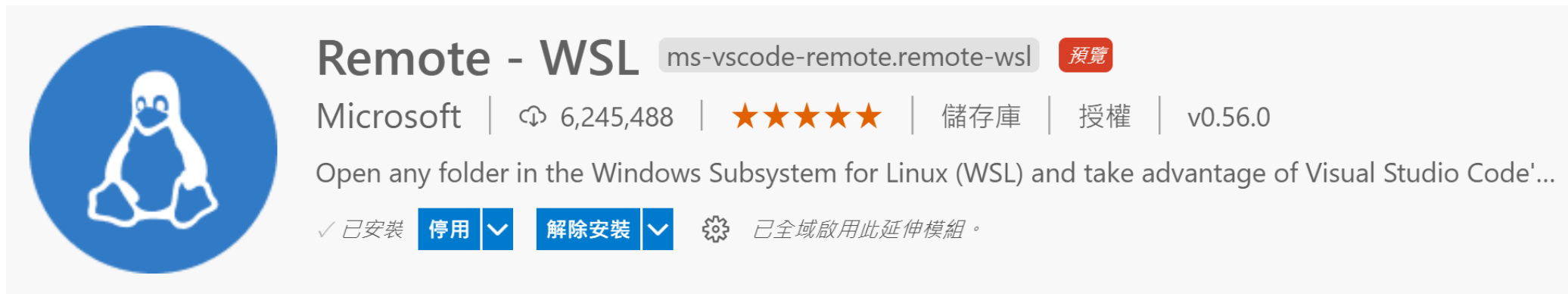
- 安裝 VS Code

<https://code.visualstudio.com/>

- 安裝 Git

<https://git-scm.com/>

- 安裝 VS Code Extension : Remote - WSL



The image shows the Visual Studio Code extension marketplace interface for the 'Remote - WSL' extension. On the left is a blue circular icon with a white penguin. To the right, the extension name 'Remote - WSL' is displayed in large bold text, followed by the publisher 'ms-vscode-remote.remote-wsl' in a grey box and a red '預覽' (Preview) badge. Below this, the publisher 'Microsoft' is listed, followed by a download count of '6,245,488', a five-star rating, and the words '儲存庫' (Repository), '授權' (License), and 'v0.56.0'. A description reads: 'Open any folder in the Windows Subsystem for Linux (WSL) and take advantage of Visual Studio Code...'. At the bottom, there are buttons for '已安裝' (Installed), '停用' (Disable), and '解除安裝' (Uninstall), along with a gear icon and the text '已全域啟用此延伸模組。' (This extension is enabled globally).

**Remote - WSL** ms-vscode-remote.remote-wsl 預覽

Microsoft | 6,245,488 | ★★★★★ | 儲存庫 | 授權 | v0.56.0

Open any folder in the Windows Subsystem for Linux (WSL) and take advantage of Visual Studio Code'...

✓ 已安裝 停用 解除安裝 已全域啟用此延伸模組。

# 使用VS Code编辑网站资料夹

- 打开VS Code
- 检视 -> 命令选择区

Remote-WSL: Reopen Folder in WSL

- 选择Ubuntu
- 档案 -> 开启资料夹
- 找到 /home/**User-name**/LearnWSL/HelloWorld

# 在VS Code 中使用Ubuntu Bash

- 检视 -> 终端  
确认下方打开的是Ubuntu的Bash
- 编辑 `views/index.pug`



# views / index.pug

```
extends layout
```

```
block content
```

```
  h1 #{title}
```

```
  p I'm over the moon!
```

```
  img(src="https://docs.microsoft.com/learn/media/crescent-timeline.png", alt="Moon Photo", srcset="")
```

# 再次启动伺服器，打开浏览器检视

- 回到指令列，按下Ctrl+C停止
- 再次输入

DEBUG=HelloWorld:\* npm start



<http://localhost:3000/>

# 使用VS Code 即时侦错

- 修改package.json

```
"scripts": {  
  "start": "node ./bin/www",  
  "debug": "node --inspect ./bin/www"  
},
```

- 点击左方侦错图案  建立 launch.json
- 选择Node.js
- 点击下拉选单新增组态，Node.js – Launch via NPM
- 选择Launch via NPM，按下启动 
- 确认网站依旧可顺利开启

# 修改程式并重新启动侦错

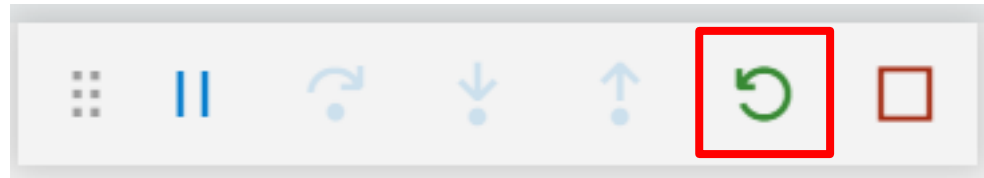
- 修改 routes/index.js

```
var express = require('express');
var router = express.Router();

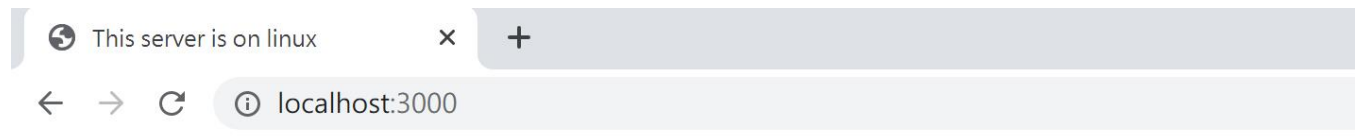
/* GET home page. */
router.get('/', function(req, res, next) {
  res.render('index', { title: `This server is on ${process.platform}` });
});

module.exports = router;
```

- 点击右上方重新整理



# 在浏览器中重新整理，查看网页



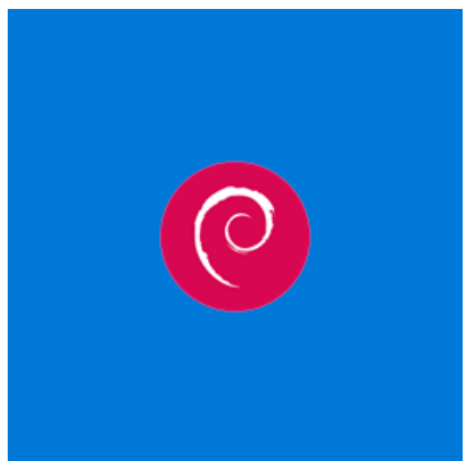
**This server is on linux**

I'm over the moon!



# 子系统可以不只有一个!

- 打开Microsoft Store，搜寻Debian，取得



## Debian

The Debian Project • 開發人員工具

🔗 分享

With this app you get Debian for the Windows Subsystem for Linux (WSL).  
You will be able to use a complete Debian command line environment containing a full current stable release environment.

[更多](#)



一般

免費

取得

♥ 願望清單

## 设定第二个子系统

- 下载完成后，启动Debian
- 建立使用者名称与密码
- 打开Windows命令提示字元，输入pwsh进入PowerShell，确认是否已有两个子系统

`wslconfig.exe /list`

```
PS C:\Program Files\> wslconfig.exe /list
Windows 子系统 Linux 版发布:
Ubuntu (预设值)
Debian
```

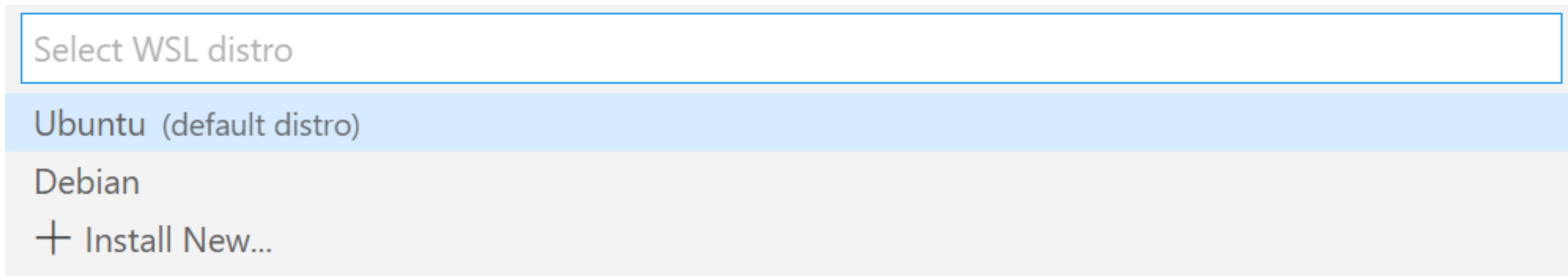
# 子系统相关指令

- 设定Debian为预设子系统
  - `wslconfig.exe /setdefault Debian`
- 删除子系统
  - `wslconfig.exe /unregister Debian`



# 在VS Code中选择不同的子系统

- 检视 -> 命令选择区 -> Remote-WSL: New Window using Distro
- 可看见多个提供选择



# 知识检查

1. 若要在计算机上结合使用 Windows 和 Linux（即完全集成的文件系统），应执行以下操作：

- ☐ 在“Windows 功能”设置中启用 Hyper-V；2) 打开“Hyper-V 快速创建”，并选择 Linux 发行版本 OS；3) 打开 Hyper-V 管理器，以命名 VM 并连接。
- ☐ 在“Windows 功能”设置中（或使用 PowerShell 脚本）启用 WSL；2) 从 Microsoft Store 安装 Linux 发行版本；3) 打开 Linux 发行版本，并设置帐户名和密码。
- ☐ 下载 Linux 发行版本，并将它保存到 U 盘；2) 重新引导计算机，并在启动过程中进入 BIOS；3) 将引导顺序更改为从 USB 上的 Linux 发行版本引导；4) 在分区上安装发行版本。

2. 通过在 WSL 终端中使用以下命令可以显示文件夹/目录中的所有隐藏文件：

- ☐ `dir`
- ☐ `pwd`
- ☐ `ls -a`

3. 以下哪个不是用于快速编辑文件中的文本，而无需离开 WSL 命令行的内置工具：

- ☐ Vim
- ☐ Nano
- ☐ Zune

# 知识检查

4. 使用以下哪种方法无法在 PowerShell 中使用 Linux 命令：

- ☐ 在命令前输入 `wsl`。例如，`wsl touch file_name.txt`。
- ☐ 在命令前输入 `linux`。例如，`linux touch file_name.txt`。
- ☐ 输入 `wsl.exe` 或 `bash.exe`，然后输入 Linux 命令。

5. 混合使用 Windows 和 Linux 工具和命令时，以下哪个命令不起作用：

- ☐ 在 Ubuntu (Linux) 终端中，输入 `powershell.exe`，然后使用 PowerShell cmdlet `Get-Service` 检索计算机上安装的服务的相关信息。
- ☐ 可以在 PowerShell 中安装一版 Python，版本不同于 Ubuntu（使用 WSL）上的 Python 和 Debian（使用 WSL）上的 Python。
- ☐ 可以复制 Internet 浏览器中的代码片段，并按 Ctrl+V 将代码片段粘贴到 Ubuntu 命令行中。

6. 以下哪个不是用于安装软件的包管理器：

- ☐ npm
- ☐ Git
- ☐ cURL

# 知识检查

7. 可使用以下哪个命令来检查 Node.js 的当前版本：

- ☐ `node -ver`
- ☐ `node --version`
- ☐ `node *v`

8. 若要将应用/项目配置为可使用 Visual Studio Code 和 Windows Subsystem for Linux 进行调试，需要：

- ☐ 将带有 `runWSL` 的 `config.js` 文件设置为 `true`。
- ☐ `launch.json` 文件。
- ☐ 将带有 `openWSL` 的 `app.config` 文件设置为 `VS Code`。

# Summary

- 启用WSL在Windows中安装Linux子系统
- 可在Linux中使用Windows的指令
- 可在Linux中存取Windows的数据
- 子系统如有需求，可以安装多个不同的Linux发行版
- VS Code支援使用子系统架设伺服器，并编辑档案



# Reactor



[developer.microsoft.com/reactor/](https://developer.microsoft.com/reactor/)  
[@MSFTReactor](https://twitter.com/MSFTReactor) on Twitter

# 议程结束 感谢聆听



请记得填写课程回馈问卷 (Event ID : **XXXXXX**)  
<https://aka.ms/Reactor/Survey>

© 2019 Microsoft Corporation. All rights reserved. The text in this document is available under the Creative Commons Attribution 3.0 License, additional terms may apply. All other content contained in this document (including, without limitation, trademarks, logos, images, etc.) are not included within the Creative Commons license grant. This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.