



Insight into

DBMS: Design and Implementation

Project 5: Debug PostgreSQL



mlinking@126.com
+86 15010255486

Outline of the chapters covered

- **Introduction**
- **Overview of the projects**
- **Demonstration of Development environment**
 - Watch and practice by yourself
- **My understanding about (R)DBMS**
 - History and D&I
- **SQL translation with 2 conversions**
 - SQL → RA (Relational Algebra)
 - RA → Sequence of File operations
- **Transaction control**
- **Deeper**
 - File, (R)DBMS, ERP, DW, Big Data (No SQL, SQL again)
 - SQL on MPP and Hadoop (Greenplum, **HAWQ**)

You can start some projects ASAP 100 pts based on

■ Basic (70)

- 1.
- 2.
- 3.
- 4.
5. Debug PostgreSQL/Greenplum/HAWQ

■ 1 Web-based Application project (20)



■ Advanced (10)



Project requirements

□ Project 5 [10 pts]: Debug PostgreSQL

■ <https://www.postgresql.org/>

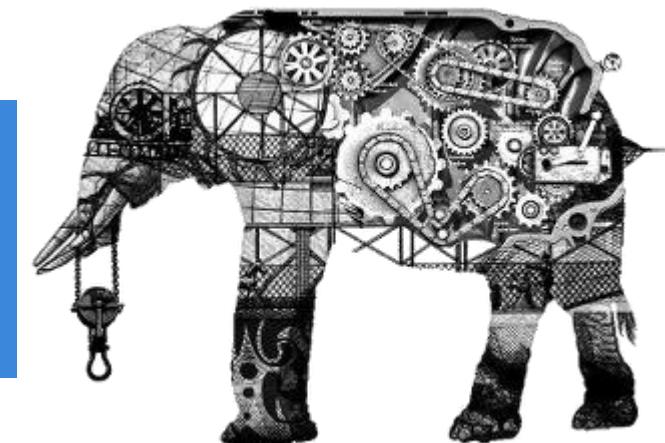
- Very helpful to follow source code to understand the D&I of many softwares
 - Minix/Linux for OS
 - PostgreSQL for DBMS



The Internals of PostgreSQL

for database administrators and system developers

<http://www.interdb.jp/pg/>



-
- I prepared some slides to demonstrate how to use Perl to compile PostgreSQL from source code

- Tasks of Project 5

1. Use PostgreSQL

- With the given dataset – “Online Retail.xlsx” – 541909 records
 - ✓ <https://archive.ics.uci.edu/ml/datasets/Online+Retail>
 - ✓ Try to load it into PostgreSQL
 - » Define the table same as xlsx? Redundant or not?
 - ✓ Try SQL sentences
 - » I.D.U.S

□ Tasks of Project 5

2. Debug PostgreSQL
 - a) Change server prompt to your name
 - ✓ lbkong>>
 - b) Cut the SQL processing in PostgreSQL (Client and Server), and just echo the inputted SQL
 - ✓ lbkong>> your input is: SELECT * From students
 - c) Try to print out the parse tree of a simple SQL supported by PostgreSQL, like “SELECT * from students”
 - ✓ Of course you should create a “students” table first, and input some students records
 - ✓ Name, studentID, Memo (for 40 words)

Chapter 2: Build PostgreSQL on Windows 10

❑ History of PostgreSQL

- Ingres
- PostgreSQL
- more details later For Data Warehouse, Big Data
 - ✓ Greenplum, HAWQ

❑ Build PostgreSQL on Windows 10

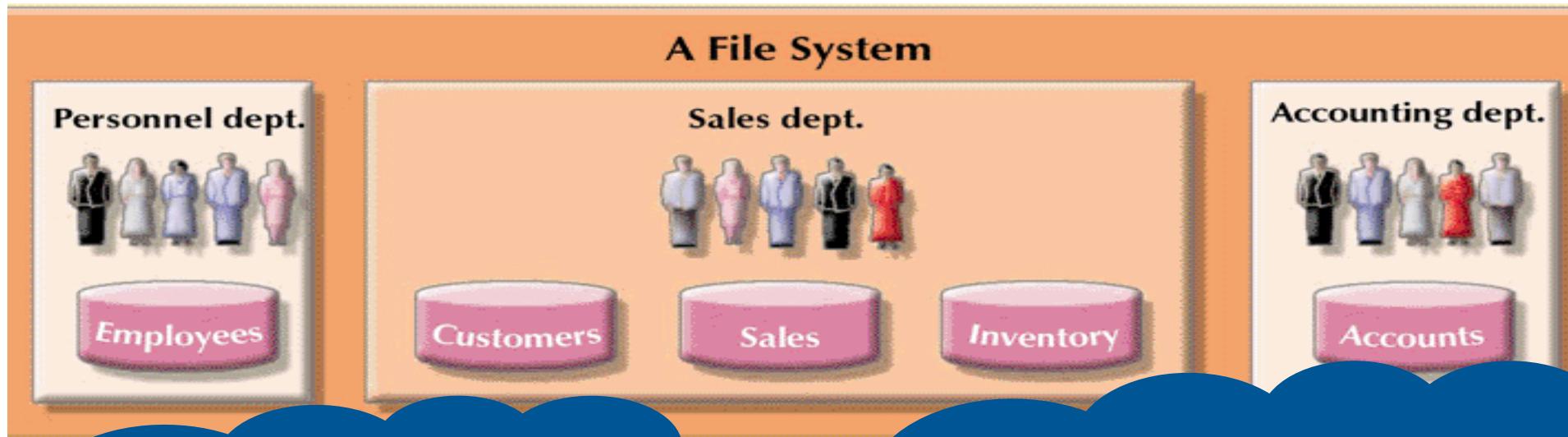
- Perl + VS

❑ Interact with PostgreSQL

❑ Debug PostgreSQL with VS

DBMS? – Data Management!

- By data management, file operations (open, close, fseek, ...) are the basis
 - When we use computers to carry out “data management”

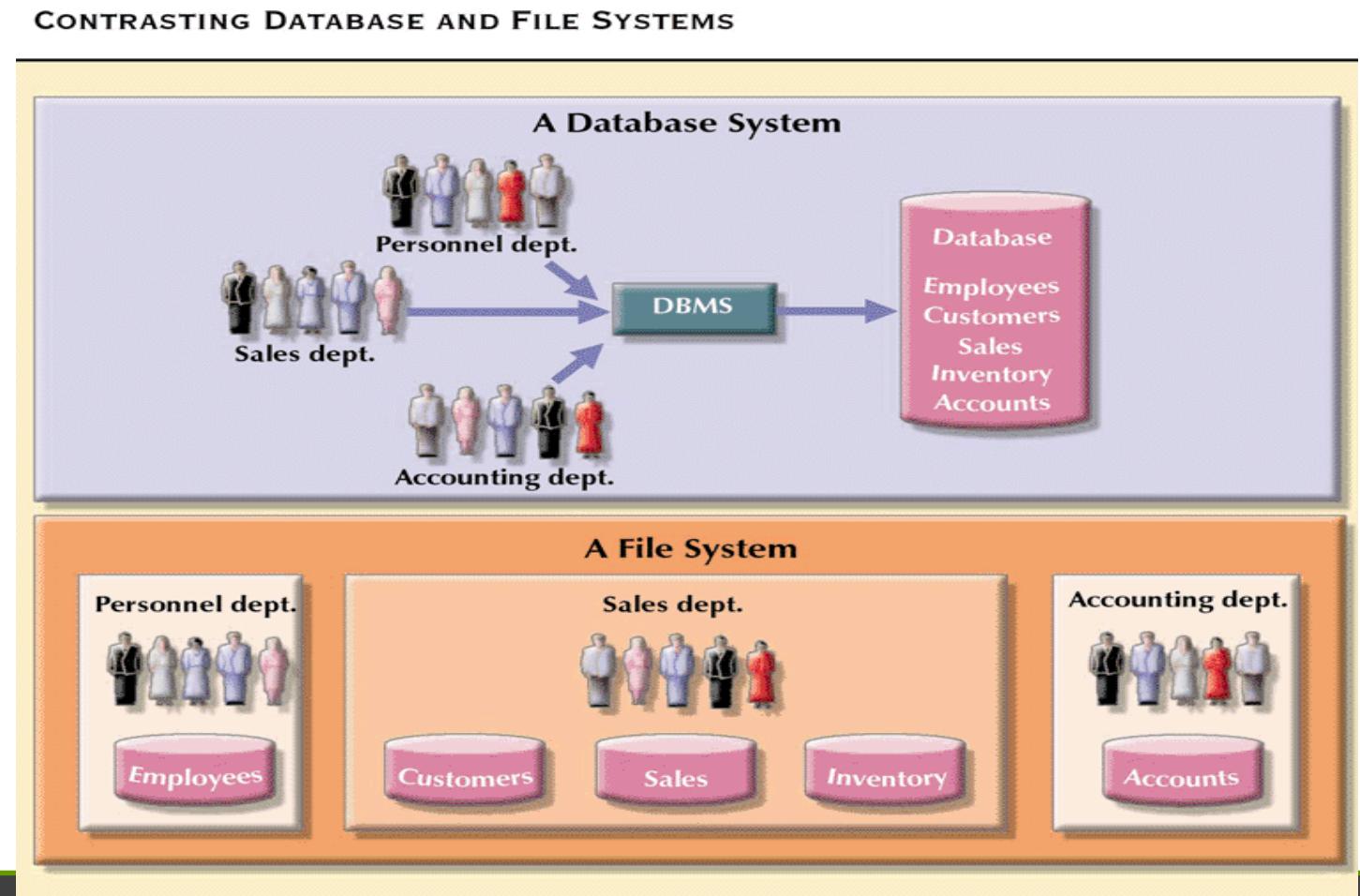


However, if there are
too many records –
millions, billions...?

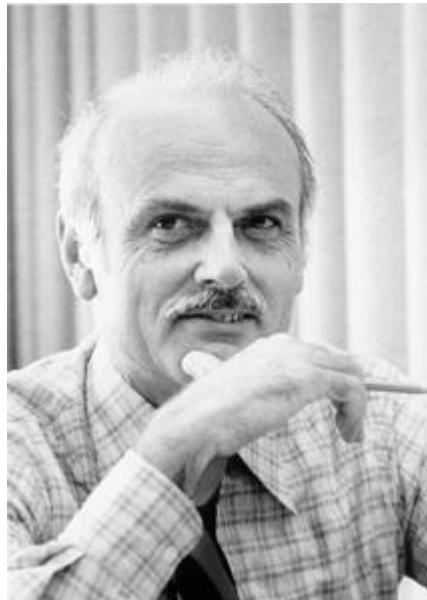
Btw, this is not effective
for sharing – Information
Sharing is important –
leading to DBMS later

□ Here comes DBMS – a wrapper/encapsulation (封装) of file management for more friendly and effective data management

- Friendly Interface for users/clients
- Effective (or safe) management of the data among concurrent users/clients



- Relational algebra proposed by E. F. Codd in 1970s is a powerful model to summary the data and operations



Edgar Frank Codd

Enrolled

cid	grade	sid
Carnatic101	C	53666
Reggae203	B	53666
Topology112	A	53650
History105	B	53666

Students

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

- Pre-Relational:

write a program

- Relational SQL

Select name, cid from students s, enrolled e where s.sid = e.sid

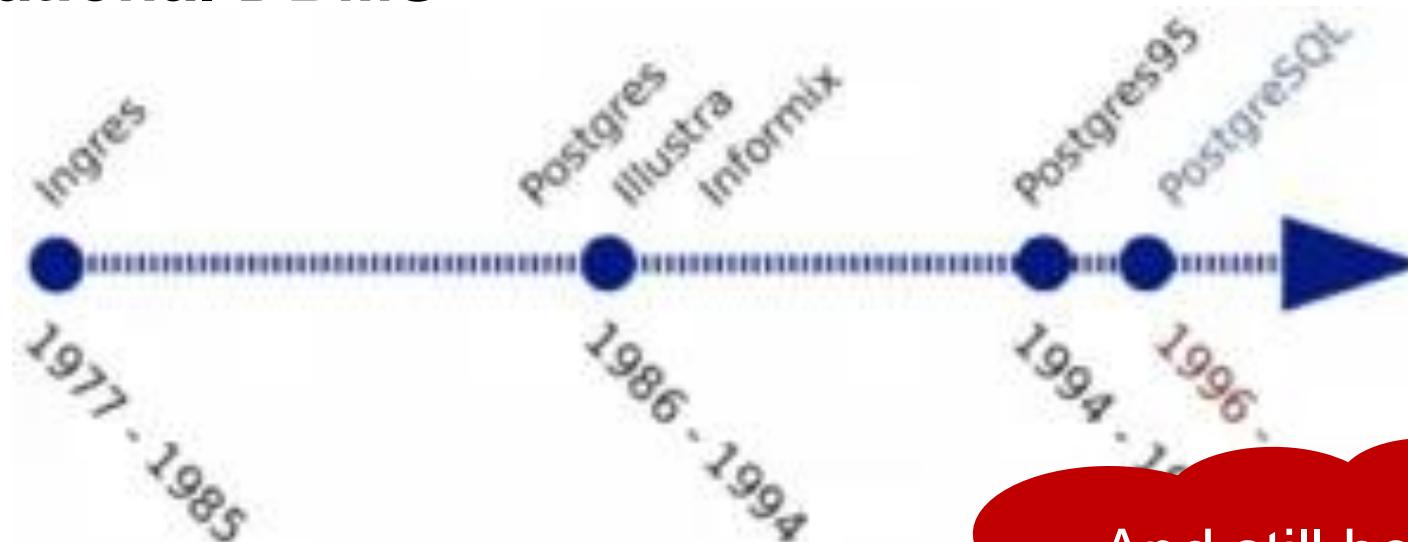
- Relational Algebra

$$\pi_{name, cid} (Students \bowtie_{sid} Enrolled)$$

- Relational Calculus

$$\{R \mid R.name = S.name \wedge R.cid = S.cid \wedge \exists S(S \in Students \wedge \exists E(E \in Enrolled \wedge E.sid = S.sid))\}$$

- Here came Prof. Michael Stonebraker who proposed Ingres – the 1st relational DBMS



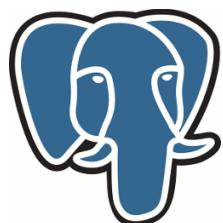
M. Stonebraker



E. Wong

And still hot in
now Big Data era!

the world's most advanced
open source database



PostgreSQL
the world's most advanced open source database

Chapter 2: Build PostgreSQL on Windows 10

❑ History of PostgreSQL

- Ingres
- PostgreSQL
- more details later For Data Warehouse, Big Data
 - ✓ Greenplum, HAWQ

❑ Build PostgreSQL on Windows 10

- Perl + VS

❑ Interact with PostgreSQL

❑ Debug PostgreSQL with VS

□ Prepare necessary tools

■ Visual Studio Community

➤ <https://visualstudio.microsoft.com/zh-hans/downloads/>

■ Perl

■ Bison 2.4.1

■ Diffutils 2.8.7

■ Flex 2.5.4a



正在安装 — Visual Studio Community 2019 — 16.7.2

工作负载 单个组件 语言包 安装位置

Web 和云 (4)



ASP.NET 和 Web 开发

使用 ASP.NET Core、ASP.NET、HTML/JavaScript 和包括 Docker 支持的容器生成 Web 应用程序。



Azure 开发

用于使用 .NET Core 和 .NET Framework 开发云应用和创建资源的 Azure SDK、工具和项目。还包含用于实现应用程...



Python 开发

对 Python 进行编辑、调试、交互式开发和源代码管理。



Node.js 开发

使用 Node.js (一个由异步事件驱动的 JavaScript 运行时)生成可缩放的网络应用程序。

桌面应用和移动应用 (5)



.NET 桌面开发

将 C#、Visual Basic 和 F# 用于 .NET Core 和 .NET Framework，生成 WPF、Windows 窗体和控制台应用程...



使用 C++ 的桌面开发

使用所选工具(包括 MSVC、Clang、CMake 或 MSBuild)生成适用于 Windows 的现代 C++ 应用。



通用 Windows 平台开发

使用 C#、VB、或 C++ (可选)为通用 Windows 平台创建应



使用 .NET 的移动开发

使用 Xamarin 对 iOS、Android 或 Windows 生成跨平台应

位置

C:\Program Files (x86)\Microsoft Visual Studio\2019\Community [更改...](#)

继续操作即表示你同意所选 Visual Studio 版本的[许可证](#)。我们还允许使用 Visual Studio 下载其他软件。该软件需要进行单独许可，如[第三方通告](#)或其随附的许可证中所述。继续即表示你同意这些许可证。

安装详细信息

- Windows 10 SDK (10.0.18362.0)
- 实时调试器
- C++ 分析工具
- 用于 Windows 的 C++ CMake 工具
- 适用于最新 v142 生成工具的 C++ ATL (x86 和 x...
- Boost.Test 测试适配器
- Google Test 测试适配器
- Live Share
- C++ AddressSanitizer (实验性)
- 适用于最新 v142 生成工具的 C++ MFC (x86 和 x...
- 对 v142 生成工具(14.27)的 C++/CLI 支持
- 用于 v142 生成工具的 C++ 模块(x64/x86 - 实验...
- 适用于 Windows 的 C++ Clang 工具(10.0.0 - x64...
- JavaScript 诊断
- Incredibuild - 生成加速
- Windows 10 SDK (10.0.17763.0)
- Windows 10 SDK (10.0.17134.0)
- Windows 10 SDK (10.0.16299.0)
- MSVC v141 – VS 2017 C++ x64/x86 生成工具(v1...
- MSVC v140 - VS 2015 C++ 生成工具(v14.00)



所需总空间 8.41 GB

下载时安装

安装

正在安装 — Visual Studio Community 2019 — 16.7.2

工作负载 单个组件 语言包 安装位置

Visual Studio IDE i

D:\Programs\Microsoft Visual Studio\2019\Community



3.85 GB

⚠ 驱动器将影响 Visual Studio 的性能并导致其运行速度降低。下载缓存 i

D:\ProgramData\Microsoft\VisualStudio\Packages



2.16 GB

 安装完成后保留下载缓存共享组件、工具和 SDK i

C:\Program Files (x86)\Microsoft Visual Studio\Shared

11 MB

系统缓存、工具和位置固定的 SDK

4.56 GB

位置

D:\Programs\Microsoft Visual Studio\2019\Community

系统驱动器 (C) 4.57 GB

其他驱动器 6 GB

所需总空间 10.56 GB

继续操作即表示你同意所选 Visual Studio 版本的[许可证](#)。我们还允许使用 Visual Studio 下载其他软件。该软件需要进行单独许可，如[第三方通告](#)或其随附的许可证中所述。继续即表示你同意这些许可证。

下载时安装

安装

安装详细信息

- Windows 10 SDK (10.0.18362.0)
- 实时调试器
- C++ 分析工具
- 用于 Windows 的 C++ CMake 工具
- 适用于最新 v142 生成工具的 C++ ATL (x86 和 x...
- Boost.Test 测试适配器
- Google Test 测试适配器
- Live Share
- C++ AddressSanitizer (实验性)
- 适用于最新 v142 生成工具的 C++ MFC (x86 和 x...
- 对 v142 生成工具(14.27)的 C++/CLI 支持
- 用于 v142 生成工具的 C++ 模块(x64/x86 - 实验...
- 适用于 Windows 的 C++ Clang 工具(10.0.0 - x64...
- JavaScript 诊断
- Incredibuild - 生成加速
- Windows 10 SDK (10.0.17763.0)
- Windows 10 SDK (10.0.17134.0)
- Windows 10 SDK (10.0.16299.0)
- MSVC v141 – VS 2017 C++ x64/x86 生成工具(v1...
- MSVC v140 - VS 2015 C++ 生成工具(v14.00)

工作负载 单个组件 语言包 安装位置

Visual Studio IDE

D:\Programs\Microsoft Visual Studio\2019\Community

驱动器将影响 Visual Studio 的性能并导致其运行速度降低。

下载缓存

D:\ProgramData\Microsoft\VisualStudio\Packages

 安装完成后保留下载缓存

共享组件、工具和 SDK

C:\Program Files (x86)\Microsoft Visual Studio\Shared

是否更改 Visual Studio IDE 默认位置?

你有固态系统驱动器。可在非 SSD 驱动器上安装 Visual Studio IDE，但会影响性能并导致其运行速度降低。

Visual Studio IDE 需要 3.85 GB。

你希望做什么?

 安装在系统驱动器上(推荐)

安装路径: C:\Program Files (x86)\Microsoft Visual Studio\2019\Community

 继续使用我的选择

安装路径: D:\Programs\Microsoft Visual Studio\2019\Community

确定

安装详细信息

 Windows 10 SDK (10.0.18362.0)

实时调试器

分析工具

用于 Windows 的 C++ CMake 工具

适用于最新 v142 生成工具的 C++ ATL (x86 和 x64)

cost.Test 测试适配器

Google Test 测试适配器

File Share

AddressSanitizer (实验性)

适用于最新 v142 生成工具的 C++ MFC (x86 和 x64)

适用于 v142 生成工具(14.27)的 C++/CLI 支持

适用于 v142 生成工具的 C++ 模块(x64/x86 - 实验性)

适用于 Windows 的 C++ Clang 工具(10.0.0 - x64)

JavaScript 诊断

CrediBuild - 生成加速

Windows 10 SDK (10.0.17763.0)

Windows 10 SDK (10.0.17134.0)

Windows 10 SDK (10.0.16299.0)

MSVC v141 – VS 2017 C++ x64/x86 生成工具(v1...

 MSVC v140 - VS 2015 C++ 生成工具(v14.00)

位置

D:\Programs\Microsoft Visual Studio\2019\Community

系统驱动器 (C) 4.57 GB

其他驱动器 6 GB

所需总空间 10.56 GB

继续操作即表示你同意所选 Visual Studio 版本的[许可证](#)。我们还允许使用 Visual Studio 下载其他软件。该软件需要进行单独许可，如[第三方通告](#)或其随附的许可证中所述。继续即表示你同意这些许可证。

下载时安装

安装

□ Prepare necessary tools

- Visual Studio Community

- Perl

- <https://www.perl.org/>

- Bison 2.4.1

- Diffutils 2.8.7

- Flex 2.5.4a





Get ActivePerl Community Edition

Windows

Linux

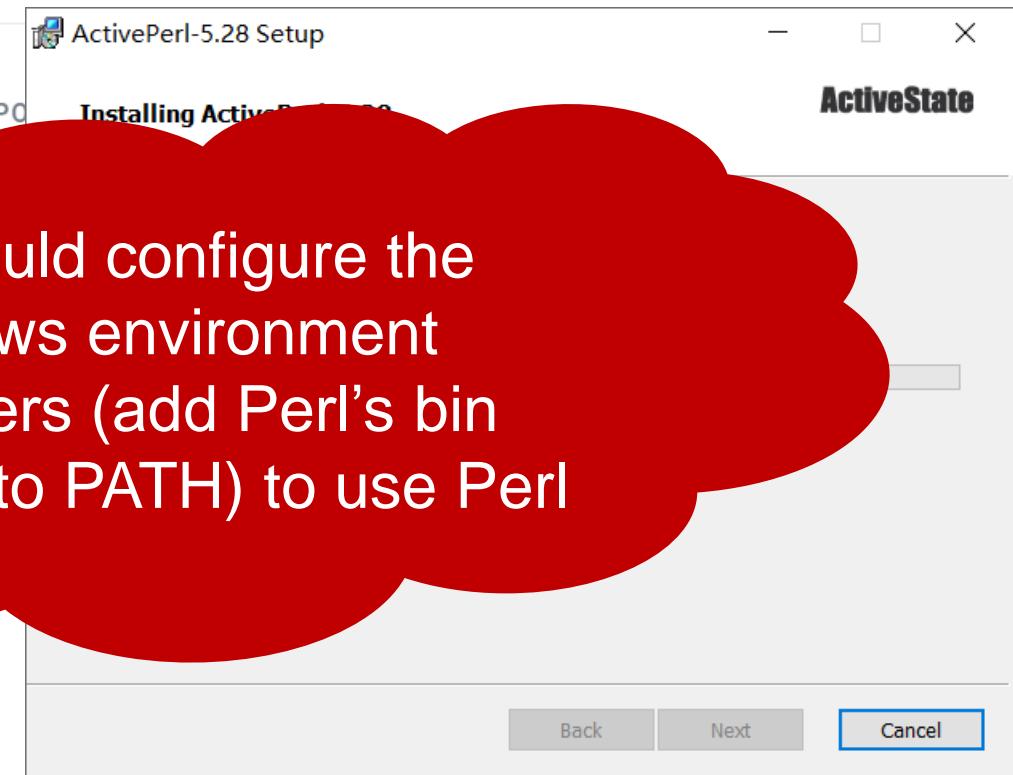
Mac

ActivePerl 5.28

ActivePerl 5.26

If you need **32-bit** or other older/legacy versions, they are available through our new ActiveState Platform by subscribing to at least Team Tier. [See pricing here.](#)

You should configure the Windows environment parameters (add Perl's bin directory into PATH) to use Perl



管理员: x64 Native Tools Command Prompt for VS 2019

```
*****  
** Visual Studio 2019 Developer Command Prompt v16.7.2  
** Copyright (c) 2020 Microsoft Corporation  
*****  
[vcvarsall.bat] Environment initialized for: 'x64'  
  
C:\Windows\System32>perl -h  
  
Usage: perl [switches] [--] [programfile] [arguments]  
  -0[octal]          specify record separator (\0, if no argument)  
  -a                autosplit mode with -n or -p (splits $_ into @F)  
  -C[number/list]    enables the listed Unicode features  
  -c                check syntax only (runs BEGIN and CHECK blocks)  
  -d[:debugger]      run program under debugger  
  -D[number/list]    set debugging flags (argument is a bit mask or alphabets)  
  -e program         one line of program (several -e's allowed, omit programfile)  
  -E program         like -e, but enables all optional features  
  -f                don't do $sitelib/sitecustomize.pl at startup  
  -F/pattern/        split() pattern for -a switch (//'s are optional)  
  -i[extension]      edit <> files in place (makes backup if extension supplied)  
  -Idirectory       specify @INC/#include directory (several -I's allowed)  
  -l[octal]          enable line ending processing, specifies line terminator  
  -[mM] [-]module    execute "use/no module..." before executing program  
  -n                assume "while (<>) { ... }" loop around program  
  -p                assume loop like -n but print line also, like sed  
  -s                enable rudimentary parsing for switches after programfile  
  -S                look for programfile using PATH environment variable  
  -t                enable tainting warnings  
  -T                enable tainting checks  
  -u                dump core after parsing program  
  -U                allow unsafe operations  
  -v                print version, patchlevel and license  
  -V[:variable]     print configuration summary (or a single Config.pm variable)  
  -w                enable many useful warnings  
  -W                enable all warnings  
  -x[directory]     ignore text before #!perl line (optionally cd to directory)  
  -X                disable all warnings
```

Run 'perldoc perl' for more help with Perl.





□ Prepare necessary tools

■ Visual Studio Community

<https://mirrors.aliyun.com/gn>

■ Perl

- Bison 2.4.1
- Diffutils 2.8.7
- Flex 2.5.4a

Msys2



□ MSYS2

- is a collection of tools and libraries providing you with an easy-to-use environment for building, installing and running native Windows software

The screenshot shows two windows side-by-side. On the left is the 'MSYS2 64bit 安装程序' (Installation Wizard) window, which displays the title '安装程序 - MSYS2 64bit' and the message '欢迎使用 MSYS2 64bit 安装向导。'. At the bottom are two buttons: '下一步(N)' (Next) and '退出(E)' (Exit). On the right is a file explorer window titled '选择安装文件夹' (Select Installation Directory). The path shown is '此电脑 > 新加卷 (D:) > Programs >'. The file list shows various installed programs like Ant Download Manager, BaiduNetdisk, csf, EdrawMax, EVCapture, GIMP 2, Microsoft Visual Studio, and Solid Converter v10. A folder named 'msys64' is selected and highlighted with a blue selection bar. The status bar at the bottom of the file explorer shows '文件夹: msys64'. At the bottom right of the file explorer are three buttons: '选择文件夹' (Select Directory), '取消' (Cancel), and another '选择文件夹' button with a mouse cursor icon pointing to it.

□ Change the updating source

- MSYS2 中 pacman 的源文件位于 C:\msys64\etc\pacman.d 目录
- 分别编辑里面的三个文件添加地址即可
 - 编辑 /etc/pacman.d/mirrorlist.mingw32 , 在文件开头添加:
 - ✓ Server = http://mirrors.ustc.edu.cn/msys2/mingw/i686
 - 编辑 /etc/pacman.d/mirrorlist.mingw64 , 在文件开头添加
 - ✓ Server = http://mirrors.ustc.edu.cn/msys2/mingw/x86_64
 - 编辑 /etc/pacman.d/mirrorlist.msys , 在文件开头添加
 - ✓ Server = http://mirrors.ustc.edu.cn/msys2/msys/\$arch



Install bison, flex and diffutils

□ \$ Pacman –S bison

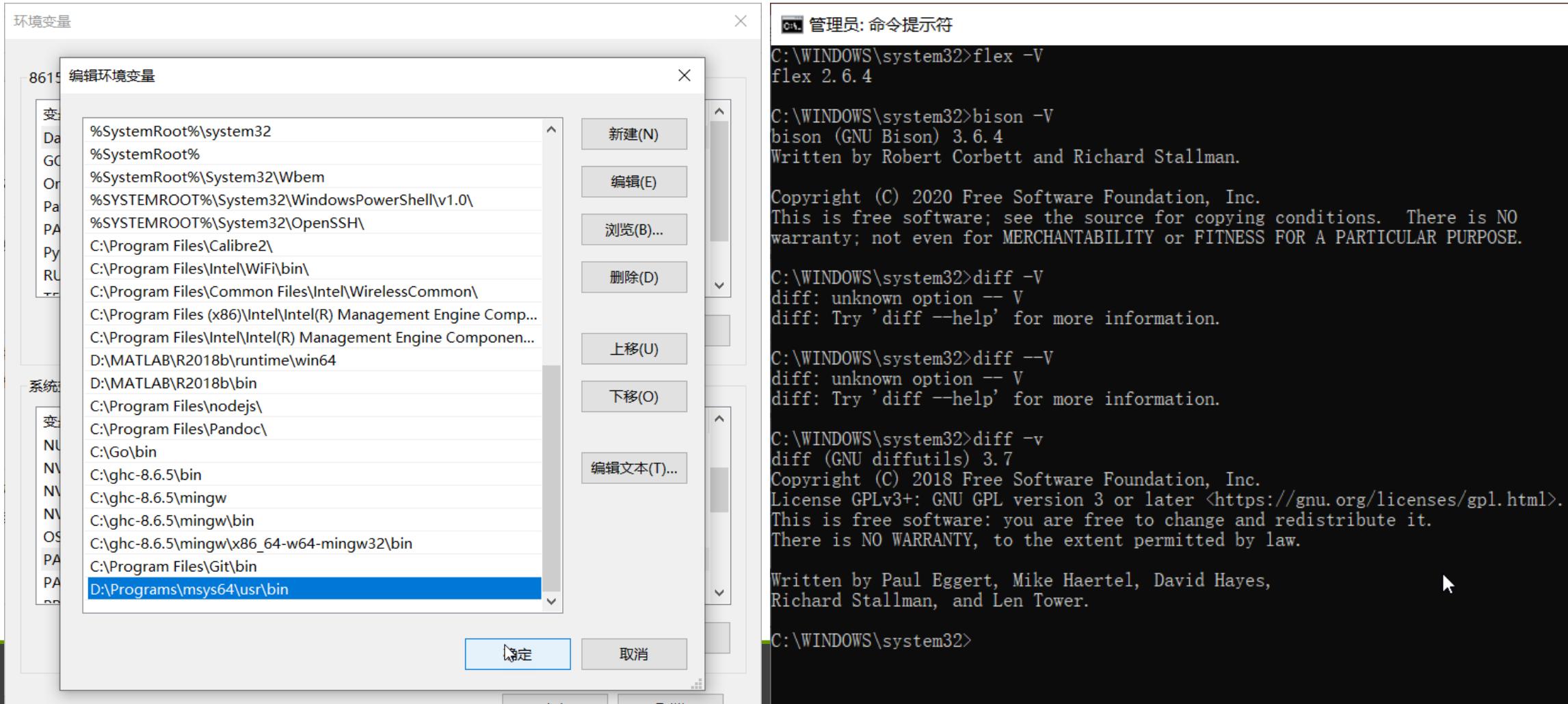
The image shows two terminal windows side-by-side, both titled 'MSYS ~'. The left window displays the command '\$ Pacman -S bison' followed by the package download and installation process for m4 and bison. The right window shows the command '\$ Pacman -S diffutils' followed by the package download and installation process for flex and diffutils. Both windows show progress bars at 100% completion.

```
$ Pacman -S bison
正在解析依赖关系...
正在查找软件包冲突...
软件包 (2) m4-1.4.18-2 bison-3.6.4-1
下载大小: 0.91 MiB
全部安装大小: 2.49 MiB
:: 进行安装吗? [Y/n] y
:: 正在获取软件包.....
m4-1.4.18-2-x86_64 218.4 KiB 3.68 MiB
bison-3.6.4-1-x86_64 708.6 KiB 10.5 MiB
(2/2) 正在检查密钥环里的密钥
(2/2) 正在检查软件包完整性
(2/2) 正在加载软件包文件
(2/2) 正在检查文件冲突
(2/2) 正在检查可用存储空间
:: 正在处理软件包的变化...
(1/2) 正在安装 m4
(2/2) 正在安装 bison
:: 正在运行事务后钩子函数...
(1/1) Updating the info directory file...
86150@mLinkingHW MSYS ~
$ Pacman -S flex
正在解析依赖关系...
正在查找软件包冲突...
软件包 (1) flex-2.6.4-1
下载大小: 0.29 MiB
全部安装大小: 1.46 MiB
:: 进行安装吗? [Y/n]
```

```
(1/1) 正在检查可用存储空间
:: 正在处理软件包的变化...
(1/1) 正在安装 flex
:: 正在运行事务后钩子函数...
(1/1) Updating the info directory file...
86150@mLinkingHW MSYS ~
$ flex -v
86150@mLinkingHW MSYS ~
$ Pacman -S diffutils
正在解析依赖关系...
正在查找软件包冲突...
软件包 (1) diffutils-3.7-1
下载大小: 0.33 MiB
全部安装大小: 1.37 MiB
:: 进行安装吗? [Y/n] y
:: 正在获取软件包.....
diffutils-3.7-1-x86_64 342.0 KiB 4.23 MiB/s 00:00
(1/1) 正在检查密钥环里的密钥
(1/1) 正在检查软件包完整性
(1/1) 正在加载软件包文件
(1/1) 正在检查文件冲突
(1/1) 正在检查可用存储空间
:: 正在处理软件包的变化...
(1/1) 正在安装 diffutils
:: 正在运行事务后钩子函数...
(1/1) Updating the info directory file...
86150@mLinkingHW MSYS ~
$ |
```

core perl	bzcmp	clear.exe	db replicate.exe	expr.exe	qdbm dump.exe	gpqv.
site perl	bzdiff	cmd	db stat.exe	factor.exe	qdbm load.exe	gpg-v
vendor perl	bzegrep	cmp.exe	db tuner.exe	false.exe	qdbmtool.exe	gprof
[.exe	bzgrep	col.exe	db upgrade.exe	fdisk.exe	gdbus.exe	grep.
addgnupghome	bzgrep	colcrt.exe	db verify.exe	fgrep	gencat.exe	group
addr2line.exe	bzip2.exe	colrm.exe	dd.exe	file.exe	getconf.exe	gsetti
agetty.exe	bzip2recover.exe	column.exe	dep-search	find.exe	getent.exe	gunzi
applygnupgdefaults	bzless	comm.exe	df.exe	findfs.exe	getfacl.exe	gzexe
ar.exe	bzmore	cp.exe	diff.exe	flex.exe	getopt.exe	gzip.e
arch.exe	c rehash	cpp.exe	diff3.exe	flex++.exe	gettext.exe	hardli
as.exe	c++.exe	csplit.exe	dir.exe	flock.exe	gettext.sh	head.
ash.exe	c++filt.exe	curl.exe	dircolors.exe	fmt.exe	gettextize	hexdu
autopoint	c89	cut.exe	dirmngr.exe	fold.exe	gio-querymodules.exe	hmac
awk.exe	c99	cygcheck.exe	dirmngr-client.exe	fsck.cramfs.exe	gkill.exe	hostic
b2sum.exe	cal.exe	cygpath.exe	dirname.exe	fsck.minix.exe	glib-compile-schemas.exe	hostn
base32.exe	captoinfo.exe	cygwin-console-helper.exe	dlltool.exe	ftp.exe	gobject-query.exe	iconv.
base64.exe	cat.exe	dash.exe	dllwrap.exe	g++.exe	gpg.exe	id.exe
basename.exe	cc.exe	date.exe	dnsdomainname.exe	gapplication.exe	gpg-agent.exe	info.e
basenc.exe	cfdisk.exe	db archive.exe	du.exe	gawk.exe	gpgconf.exe	infocr
bash.exe	chattr.exe	db checkpoint.exe	dumper.exe	gawk-5.1.0.exe	gpg-connect-agent.exe	infotc
bashbug	chcon.exe	db deadlock.exe	dumpsexp.exe	gcc.exe	gpg-error.exe	install
bison.exe	checkupdates	db dump.exe	echo.exe	gcc-ar.exe	gpgme-json.exe	install
blkid.exe	chgrp.exe	db hotbackup.exe	egrep	gcc-nm.exe	gpgme-tool.exe	ipcmk
brotli.exe	chmod.exe	db load.exe	elfedit.exe	gcc-ranlib.exe	gpgparsemail.exe	isosizo
bsdtar.exe	chown.exe	db log verify.exe	env.exe	gcov.exe	gpgscm.exe	join.e
bunzip2.exe	chroot.exe	db printlog.exe	envsubst.exe	gcov-dump.exe	gpgsm.exe	kbxut
bzcat.exe	cksum.exe	db recover.exe	expand.exe	gcov-tool.exe	gpqtar.exe	kill.ex

□ Test bison/flex/diff after configuration of Windows's Environment path



Chapter 2: Build PostgreSQL on Windows 10

❑ History of PostgreSQL

- Ingres
- PostgreSQL
- more details later For Data Warehouse, Big Data
 - ✓ Greenplum, HAWQ

❑ Build PostgreSQL on Windows 10

- Perl + VS

❑ Interact with PostgreSQL

❑ Debug PostgreSQL with VS

<https://www.postgresql.org/>

<https://www.postgresql.org/ftp/source/>

□ Get the source code of PostgreSQL

postgresql.org/ftp/source/

Logout Download Documentation Community Developers Support Donate Your account

13th August 2020: PostgreSQL 12.4, 11.9, 10.14, 9.6.19, 9.5.23, and 13 Beta 3 Released!

File Browser

Top → source

Directories



[Parent Directory]



v13beta3



v13beta2



v13beta1



v12.4



v12.3



v12.2



v12.1



v12.0



v11.9



v11.8



v11.7

File Browser

Top → source → v12.4

Directories

[Parent Directory]

Files

postgresql-12.4.tar.bz2

2020-08-10 21:29:24

postgresql-12.4.tar.bz2.md5

2020-08-10 21:29:24

postgresql-12.4.tar.bz2.sha256

2020-08-10 21:29:24

postgresql-12.4.tar.gz

2020-08-10 21:29:25

postgresql-12.4.tar.gz.md5

2020-08-10 21:29:25

postgresql-12.4.tar.gz.sha256

2020-08-10 21:29:25



□ Modify the code

■ postgresql-12.4\src\tools\msvc\VSObjectFactory.pm

- DetermineVisualStudioVersion
- \$output =~ /(\d+)\.(.\d+)\.\d+(\.\d+)?\$/m
- Change ? To .*

```
112 sub DetermineVisualStudioVersion
113 {
114
115     # To determine version of Visual Studio we use nmake as it has
116     # existed for a long time and still exists in current Visual
117     # Studio versions.
118     my $output = `nmake /? 2>&1`;
119     $? >> 8 == 0
120     or croak
121     "Unable to determine Visual Studio version: The nmake command wasn't found.";
122     if ($output =~ /(\d+)\.(.\d+)\.\d+(\.\d+).*/)
123     {
124         return _GetVisualStudioVersion($1, $2);
125     }
126
127     croak
128     "Unable to determine Visual Studio version: The nmake version could not be determined.";
129 }
```

□ Build PostgreSQL with Perl

■ postgresql-12.4\src\tools\msvc

■ DEBUG 模式编译

➤ perl build.pl DEBUG

■ 回归测试

➤ perl vcregress.pl check

■ 安装

➤ perl install.pl D:\SQLTEMP

管理员: x64 Native Tools Command Prompt for VS 2019

```

plancache          ... ok    2349 ms
limit              ... ok    2195 ms
plpgsql            ... ok    5779 ms
copy2              ... ok    3968 ms
temp               ... ok    1223 ms
domain             ... ok    5705 ms
rangefuncs         ... ok    4001 ms
prepare             ... ok    2537 ms
conversion         ... ok    2203 ms
truncate            ... ok    5509 ms
alter_table        ... ok    8091 ms
sequence            ... ok    3924 ms
polymorphism        ... ok    3888 ms
rowtypes            ... ok    3877 ms
returning           ... ok    2437 ms
largeobject         ... ok    2560 ms
with                ... ok    3778 ms
xml                 ... ok    3739 ms
parallel group (7 tests): reloptions hash_part partition_info indexing partition_join partition_ne
                           partition_join      ... ok    3816 ms
                           partition_prune     ... ok    6477 ms
                           reloptions          ... ok    314 ms
                           hash_part           ... ok    1875 ms
                           indexing            ... ok    3353 ms
                           partition_aggregate ... ok    4142 ms
                           partition_info       ... ok    1878 ms
test event_trigger   ... ok    324 ms
test fast_default    ... ok    350 ms
test stats           ... ok    760 ms
===== shutting down postmaster
===== removing temporary instance
=====
All 193 tests passed.
=====

D:\postgresql-12.4\src\tools\msvc>

```

Chapter 2: Build PostgreSQL on Windows 10

❑ History of PostgreSQL

- Ingres
- PostgreSQL
- more details later For Data Warehouse, Big Data
 - ✓ Greenplum, HAWQ

❑ Build PostgreSQL on Windows 10

- Perl + VS

❑ Interact with PostgreSQL

❑ Debug PostgreSQL with VS

Install PostgreSQL

□ 2 ways

- Download the PostgreSQL **EXE** from the website, and install it into your Windows 10 system

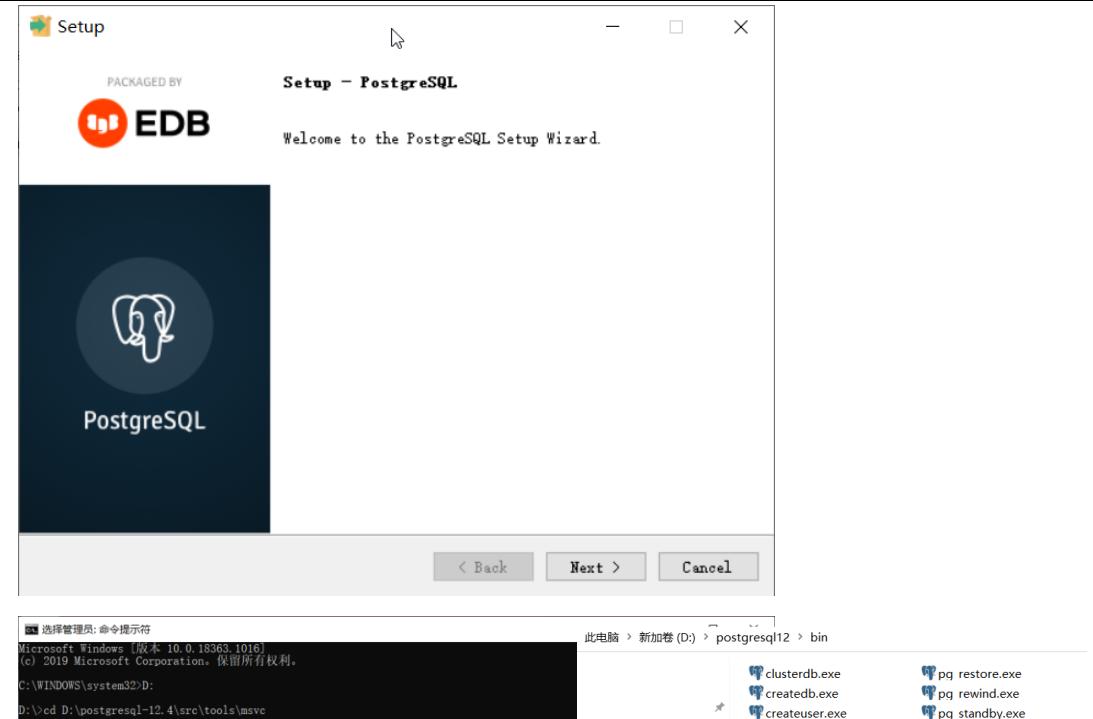
➤ You can run it

✓ <https://www.enterprisedb.com/downloads/postgres-postgresql-downloads>

- “**perl install.pl D:\SQLTEMP**” after building in the above part

➤ You have to use commands
➤ PostgreSQL also packed a compressed binary file

✓ <https://www.enterprisedb.com/download-postgresql-binaries>



Download PostgreSQL Binaries

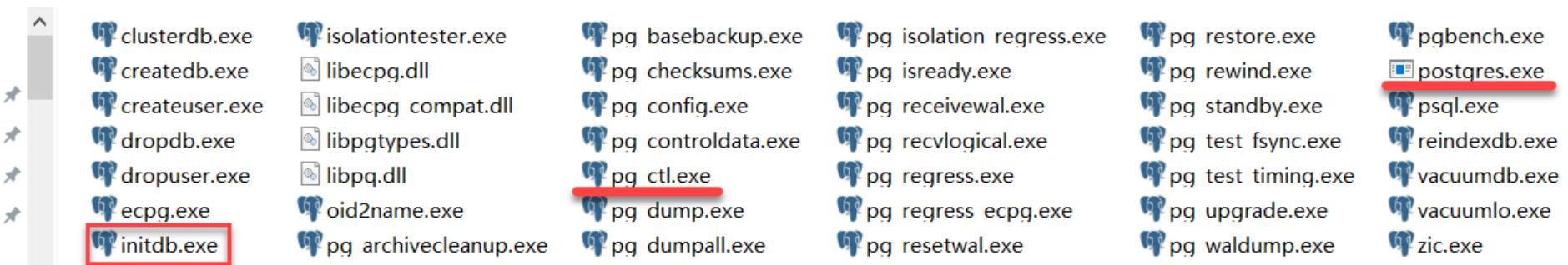
The links below allow you to download archives of the files installed by our PostgreSQL installers. These archives are provided unless you specifically need these files, you should download an installer instead.

Binaries from installer Version 13 Beta



Binaries from installer Version 12.4





- In CMD, dive into the BIN directory of PostgreSQL
- D:\postgresql12\bin>**initdb -D D:\postgresql12\data**

```
D:\postgresql12\bin>initdb -D D:\postgresql12\data
The files belonging to this database system will be owned by user "86150".
This user must also own the server process.

The database cluster will be initialized with locale "Chinese (Simplified)_China.936".
Encoding "GBK" implied by locale is not allowed as a server-side encoding.
The default database encoding will be set to "UTF8" instead.
initdb: could not find suitable text search configuration for locale "Chinese (Simplified)_China.936"
The default text search configuration will be set to "simple".

Data page checksums are disabled.

fixing permissions on existing directory D:/postgresql12/data ... ok
creating subdirectories ... ok
selecting dynamic shared memory implementation ... windows
selecting default max_connections ... 100
selecting default shared_buffers ... 128MB
selecting default time zone ... Asia/Hong_Kong
creating configuration files ... ok
running bootstrap script ... ok
performing post-bootstrap initialization ... ok
syncing data to disk ... ok

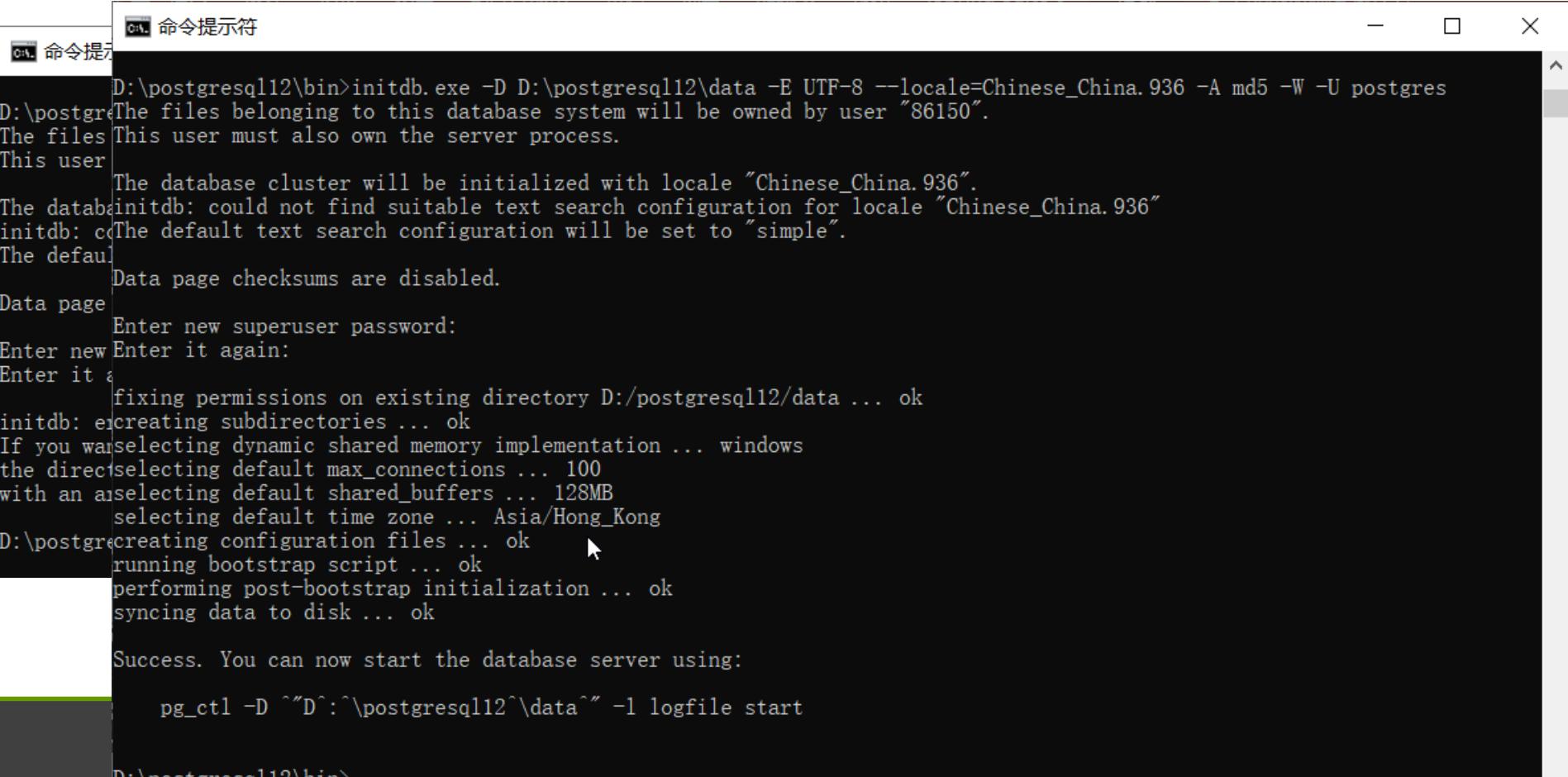
initdb: warning: enabling "trust" authentication for local connections
You can change this by editing pg_hba.conf or using the option -A, or
--auth-local and --auth-host, the next time you run initdb.

Success. You can now start the database server using:

    pg_ctl -D ""D:\\postgresql12\\data"" -l logfile start
```

□ Or more professional

- initdb.exe -D D:\postgresql12\data -E UTF-8 --locale=Chinese_China.936 -A md5 -W -U postgres



```
D:\postgresql12\bin>initdb.exe -D D:\postgresql12\data -E UTF-8 --locale=Chinese_China.936 -A md5 -W -U postgres
D:\postgrThe files belonging to this database system will be owned by user "86150".
The filesThis user must also own the server process.
This userThe database cluster will be initialized with locale "Chinese_China.936".
The databainitdb: could not find suitable text search configuration for locale "Chinese_China.936"
initdb: cThe default text search configuration will be set to "simple".
The defauData page checksums are disabled.
Data pageEnter new superuser password:
Enter newEnter it again:
Enter it &fixing permissions on existing directory D:/postgresql12/data ... ok
initdb: ecreating subdirectories ... ok
If you wanselecting dynamic shared memory implementation ... windows
the direcselecting default max_connections ... 100
with an aiselecting default shared_buffers ... 128MB
selecting default time zone ... Asia/Hong_Kong
D:\postgrcreating configuration files ... ok
running bootstrap script ... ok
performing post-bootstrap initialization ... ok
syncing data to disk ... ok

Success. You can now start the database server using:

    pg_ctl -D ^"D:^"\postgresql12^"\data^" -l logfile start

D:\postgresql12\bin>
```

□ pg_ctl -D "D:\postgresql12\data" -l logfile start

```
命令提示符
selecting dynamic shared memory implementation ... windows
selecting default max_connections ... 100
selecting default shared_buffers ... 128MB
syncing data to disk ... ok

initdb: warning: enabling "trust" authentication for local connections
You can change this by editing pg_hba.conf or using the option -A, or
--auth-local and --auth-host, the next time you run initdb.

Success. You can now start the database server using:

    pg_ctl -D ^"D^:^postgresql12^\\data^" -l logfile start

D:\postgresql12\bin>pg_ctl -D ^"D^:^postgresql12^\\data^" -l logfile start
waiting for server to start..... done
server started
D:\postgresql12\bin>
```

pg_ctl stop
pg_ctl restart

□ pg_ctl -D "D:\postgresql12\data" status

```
D:\postgresql12\bin>pg_ctl -D "D:\postgresql12\data" status
pg_ctl: server is running (PID: 26628)
D:/postgresql12/bin/postgres.exe "-D" "D:/postgresql12/data"

D:\postgresql12\bin>
```

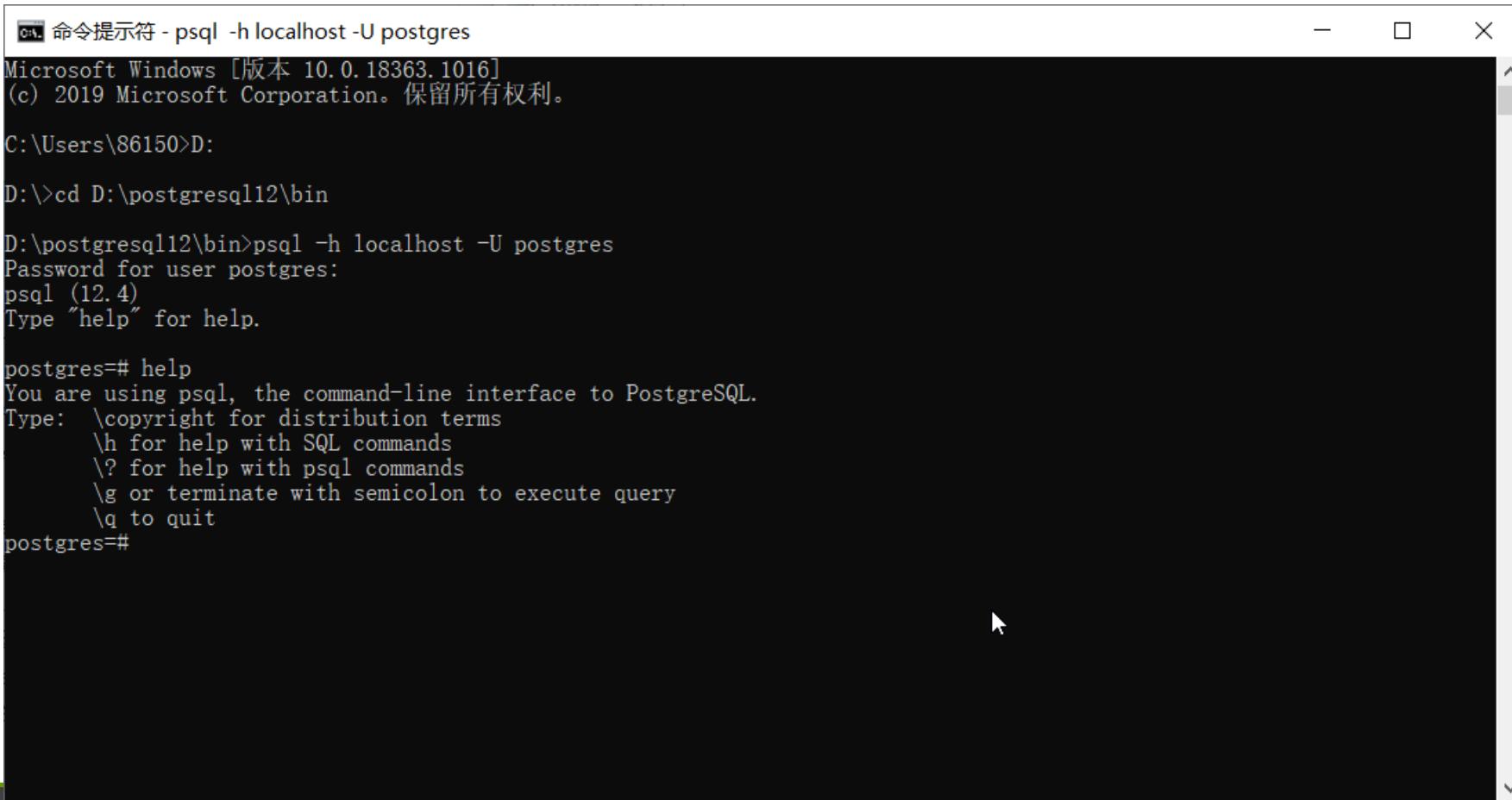
- You can also use “postgres” directly to start a server

- `postgres -D D:\postgresql12\data`

```
D:\postgresql12\bin>postgres -D D:\postgresql12\data
2020-08-26 18:00:43.024 HKT [9868] LOG:  starting PostgreSQL 12.4, compiled by Visual C++ build 1927, 64-bit
2020-08-26 18:00:43.037 HKT [9868] LOG:  listening on IPv6 address "::1", port 5432
2020-08-26 18:00:43.042 HKT [9868] LOG:  listening on IPv4 address "127.0.0.1", port 5432
2020-08-26 18:00:43.242 HKT [12332] LOG:  database system was shut down at 2020-08-26 17:59:09 HKT
2020-08-26 18:00:43.362 HKT [9868] LOG:  database system is ready to accept connections
```

□ Now you can use “psql” to connect the server

■ D:\postgresql12\bin>psql -h localhost -U postgres



命令提示符 - psql -h localhost -U postgres

Microsoft Windows [版本 10.0.18363.1016]
(c) 2019 Microsoft Corporation。保留所有权利。

C:\Users\86150>D:

D:\>cd D:\postgresql12\bin

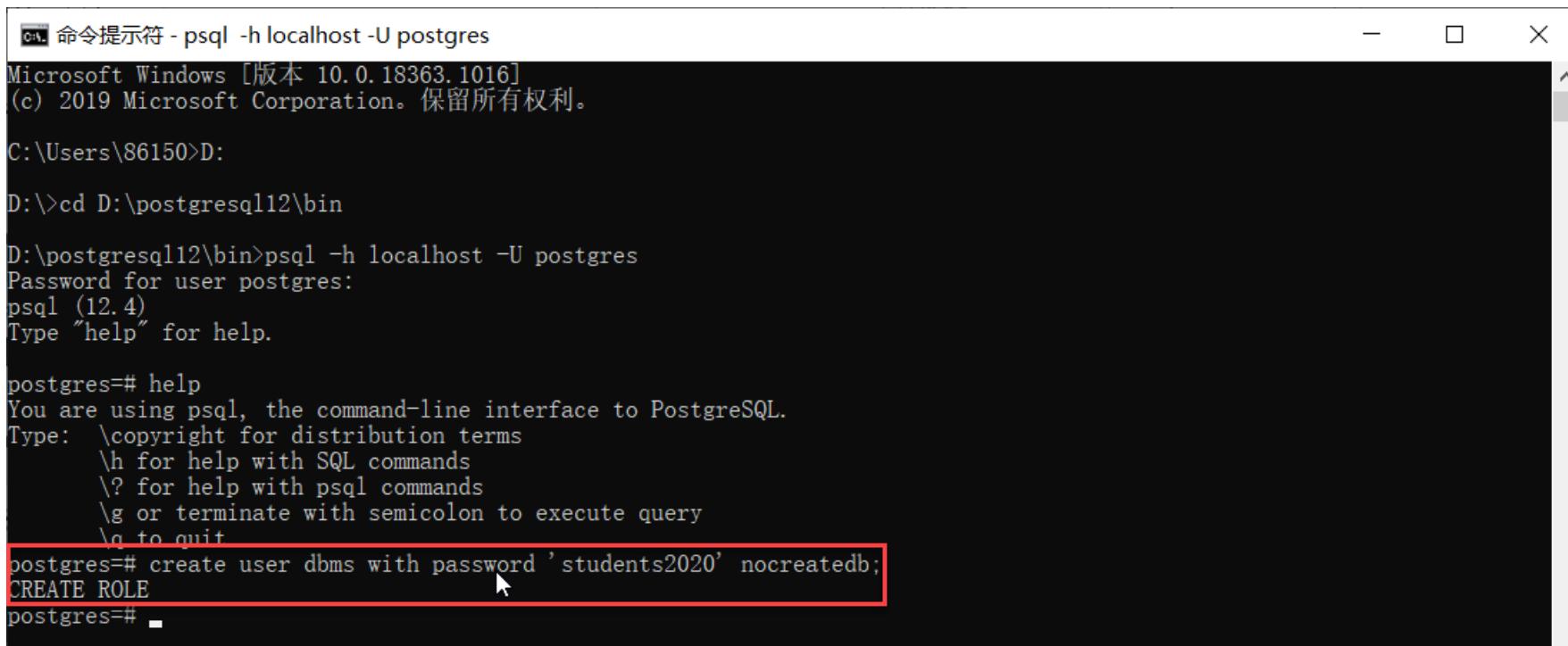
D:\postgresql12\bin>psql -h localhost -U postgres

Password for user postgres:
postgres (12.4)
Type "help" for help.

postgres=# help
You are using psql, the command-line interface to PostgreSQL.
Type: \copyright for distribution terms
 \h for help with SQL commands
 \? for help with psql commands
 \g or terminate with semicolon to execute query
 \q to quit
postgres=#

□ You'd better to create a unique user (name and psd) for every new database

- create user dbms with password 'students2020' nocreatedb;



The screenshot shows a Windows Command Prompt window titled "命令提示符 - psql -h localhost -U postgres". The window displays the following text:

```
Microsoft Windows [版本 10.0.18363.1016]
(c) 2019 Microsoft Corporation。保留所有权利。

C:\Users\86150>D:

D:\>cd D:\postgresql12\bin

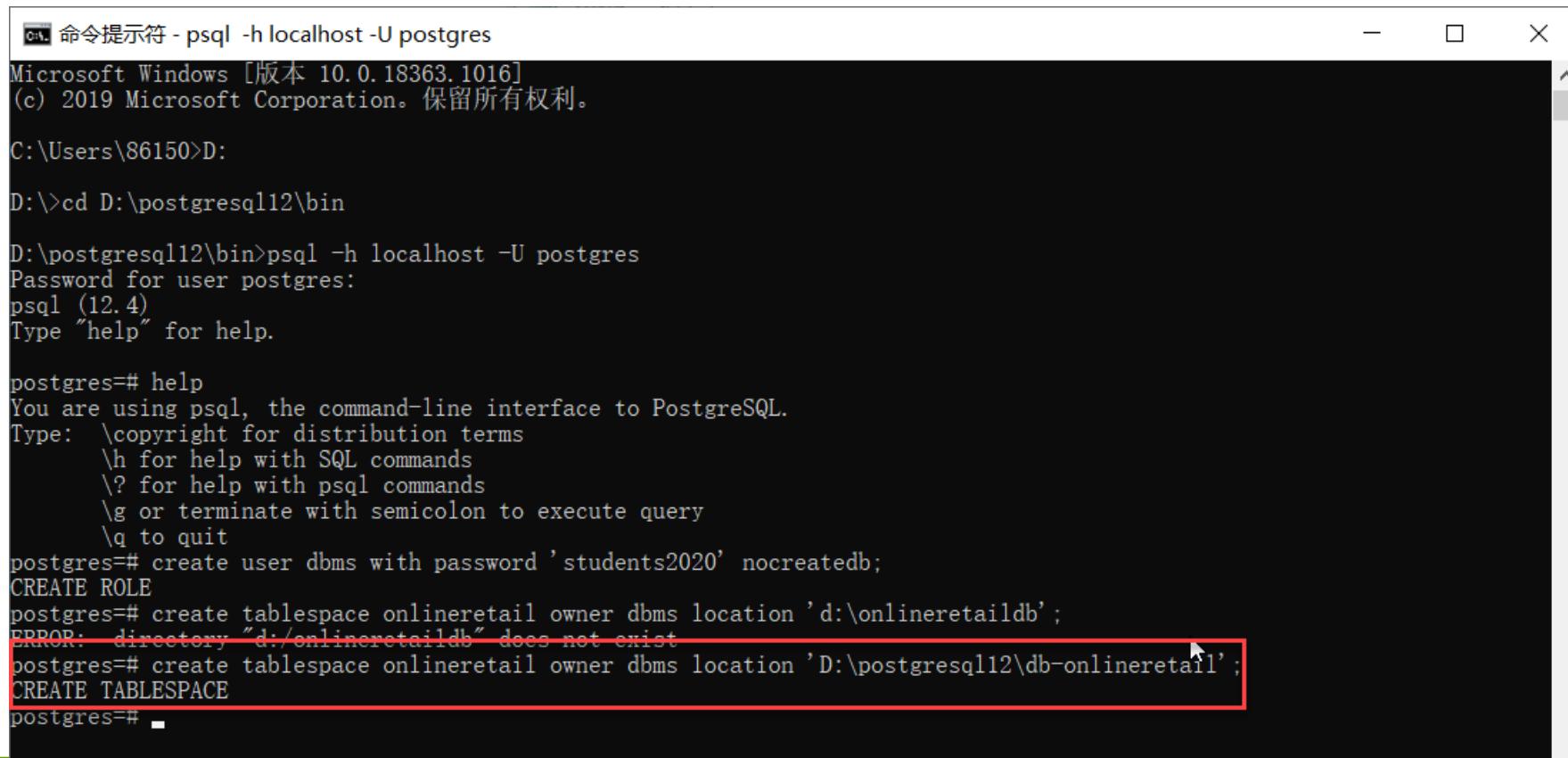
D:\postgresql12\bin>psql -h localhost -U postgres
Password for user postgres:
psql (12.4)
Type "help" for help.

postgres=# help
You are using psql, the command-line interface to PostgreSQL.
Type: \copyright for distribution terms
      \h for help with SQL commands
      \? for help with psql commands
      \g or terminate with semicolon to execute query
      \q to quit
postgres=# create user dbms with password 'students2020' nocreatedb;
CREATE ROLE
postgres=#
```

The command `create user dbms with password 'students2020' nocreatedb;` is highlighted with a red rectangle.

□ Create tablespace better in a new directory

- create **tablespace** onlineretail owner dbms location 'D:\postgresql12\db-onlineretail';



```
命令提示符 - psql -h localhost -U postgres
Microsoft Windows [版本 10.0.18363.1016]
(c) 2019 Microsoft Corporation。保留所有权利。

C:\Users\86150>D:
D:\>cd D:\postgresql12\bin

D:\postgresql12\bin>psql -h localhost -U postgres
Password for user postgres:
psql (12.4)
Type "help" for help.

postgres=# help
You are using psql, the command-line interface to PostgreSQL.
Type: \copyright for distribution terms
      \h for help with SQL commands
      \? for help with psql commands
      \g or terminate with semicolon to execute query
      \q to quit
postgres=# create user dbms with password 'students2020' nocreatedb;
CREATE ROLE
postgres=# create tablespace onlineretail owner dbms location 'd:\onlineretaildb';
ERROR:  directory "d:\onlineretaildb" does not exist
postgres=# create tablespace onlineretail owner dbms location 'D:\postgresql12\db-onlineretail';
CREATE TABLESPACE
postgres=#
```

□ Create database **onlineretaildb**

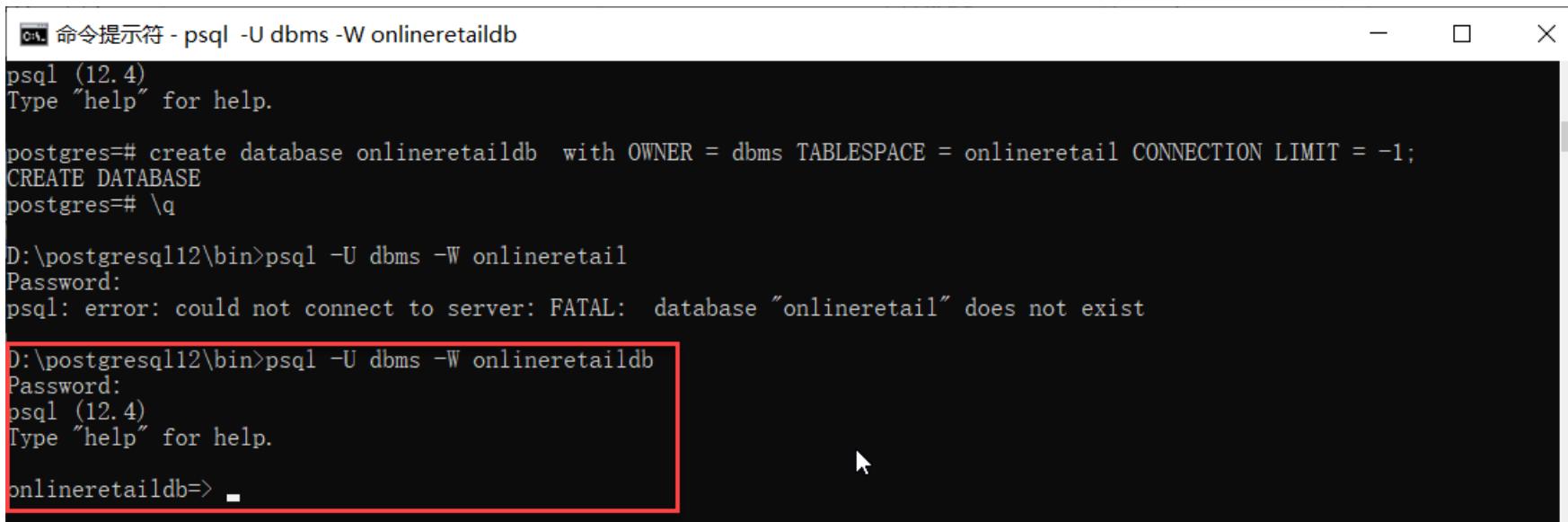
- create **database onlineretaildb** with OWNER = dbms TABLESPACE = **onlineretail** CONNECTION LIMIT = -1;
 - “CONNECTION LIMIT = -1” means no limit for connections

```
命令提示符 - psql -U postgres
psql (12.4)
Type "help" for help.

postgres=# create database onlineretaildb  with OWNER = dbms TABLESPACE = onlineretail CONNECTION LIMIT = -1;
CREATE DATABASE
postgres=#
```

□ Create table test

- You should now connect “**onlineretaildb**” with new created user “**dbms**”
 - psql -U dbms -W onlineretaildb



The screenshot shows a terminal window titled "命令提示符 - psql -U dbms -W onlineretaildb". The window contains the following text:

```
psql (12.4)
Type "help" for help.

postgres=# create database onlineretaildb with OWNER = dbms TABLESPACE = onlineretail CONNECTION LIMIT = -1;
CREATE DATABASE
postgres=# \q

D:\postgresql12\bin>psql -U dbms -W onlineretail
Password:
psql: error: could not connect to server: FATAL:  database "onlineretail" does not exist

D:\postgresql12\bin>psql -U dbms -W onlineretaildb
Password:
psql (12.4)
Type "help" for help.

onlineretaildb=>
```

A red box highlights the final command "onlineretaildb=>" at the bottom of the terminal window.

□ Now you can use SQL commands you learn from DBMS theoretic course to try PostgreSQL

```
CREATE TABLE DEPARTMENT(  
    ID INT PRIMARY KEY    NOT NULL,  
    DEPT      CHAR(50) NOT NULL,  
    EMP_ID    INT    NOT NULL  
)
```

```
CREATE TABLE COMPANY(  
    ID INT PRIMARY KEY    NOT NULL,  
    NAME      TEXT    NOT NULL,  
    AGE       INT     NOT NULL,  
    ADDRESS   CHAR(50),  
    SALARY    REAL  
)
```

```
命令提示符 - psql -U dbms -W onlineretaildb  
D:\postgresql12\bin>psql -U dbms -W onlineretaildb  
Password:  
psql (12.4)  
Type "help" for help.  
  
onlineretaildb=> CREATE TABLE COMPANY(  
onlineretaildb(>     ID INT PRIMARY KEY    NOT NULL,  
onlineretaildb(>     NAME      TEXT    NOT NULL,  
onlineretaildb(>     AGE       INT     NOT NULL,  
onlineretaildb(>     ADDRESS   CHAR(50),  
onlineretaildb(>     SALARY    REAL  
onlineretaildb(> )  
onlineretaildb-> ;  
CREATE TABLE  
onlineretaildb=> CREATE TABLE DEPARTMENT(  
onlineretaildb(>     ID INT PRIMARY KEY    NOT NULL,  
onlineretaildb(>     DEPT      CHAR(50) NOT NULL,  
onlineretaildb(>     EMP_ID    INT    NOT NULL  
onlineretaildb(> );  
CREATE TABLE  
onlineretaildb=> \d  
          List of relations  
 Schema |   Name   | Type | Owner  
-----+-----+-----+-----  
 public | company | table | dbms  
 public | department | table | dbms  
(2 rows)  
  
onlineretaildb=>
```

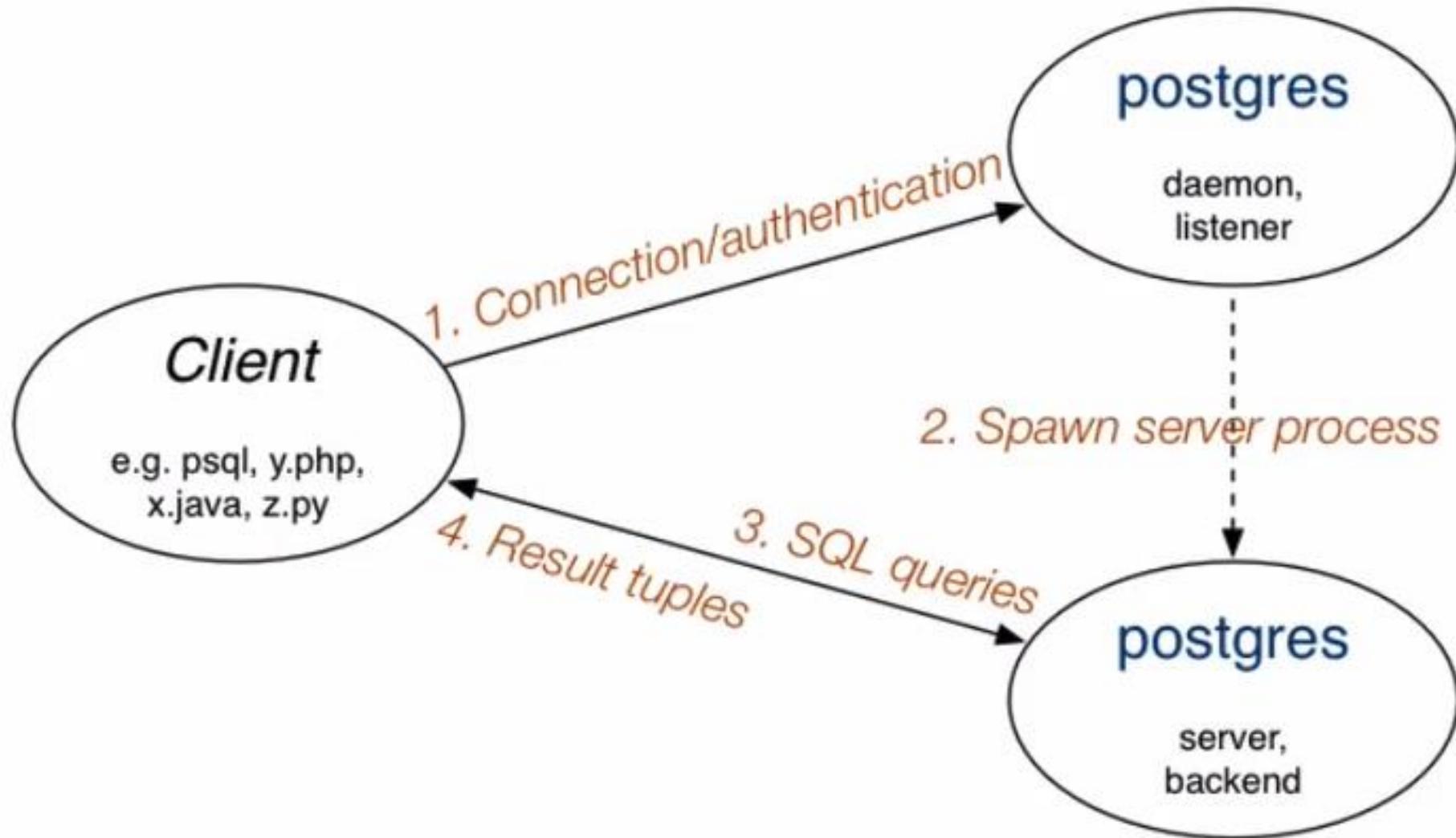
□ Don't forget to stop the postgres server

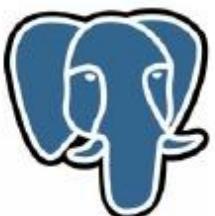
- pg_ctl -D "D:\postgresql12\data" stop

```
D:\postgresql12\bin>pg_ctl -D "D:\postgresql12\data" status
pg_ctl: server is running (PID: 30340)
D:/postgresql12/bin/postgres.exe "-D" "D:/postgresql12/data"

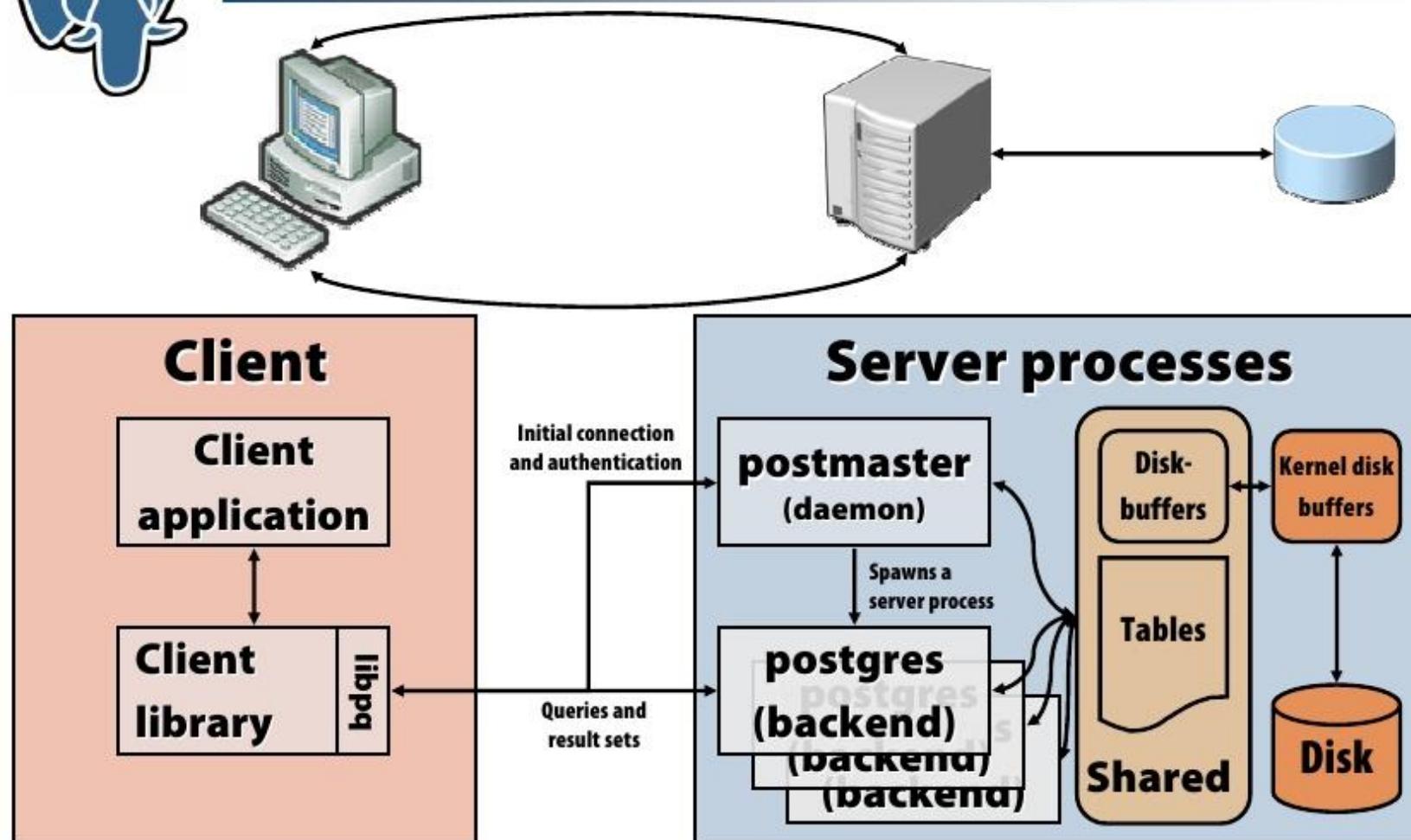
D:\postgresql12\bin>pg_ctl -D "D:\postgresql12\data" stop
waiting for server to shut down.... done
server stopped

D:\postgresql12\bin>
```

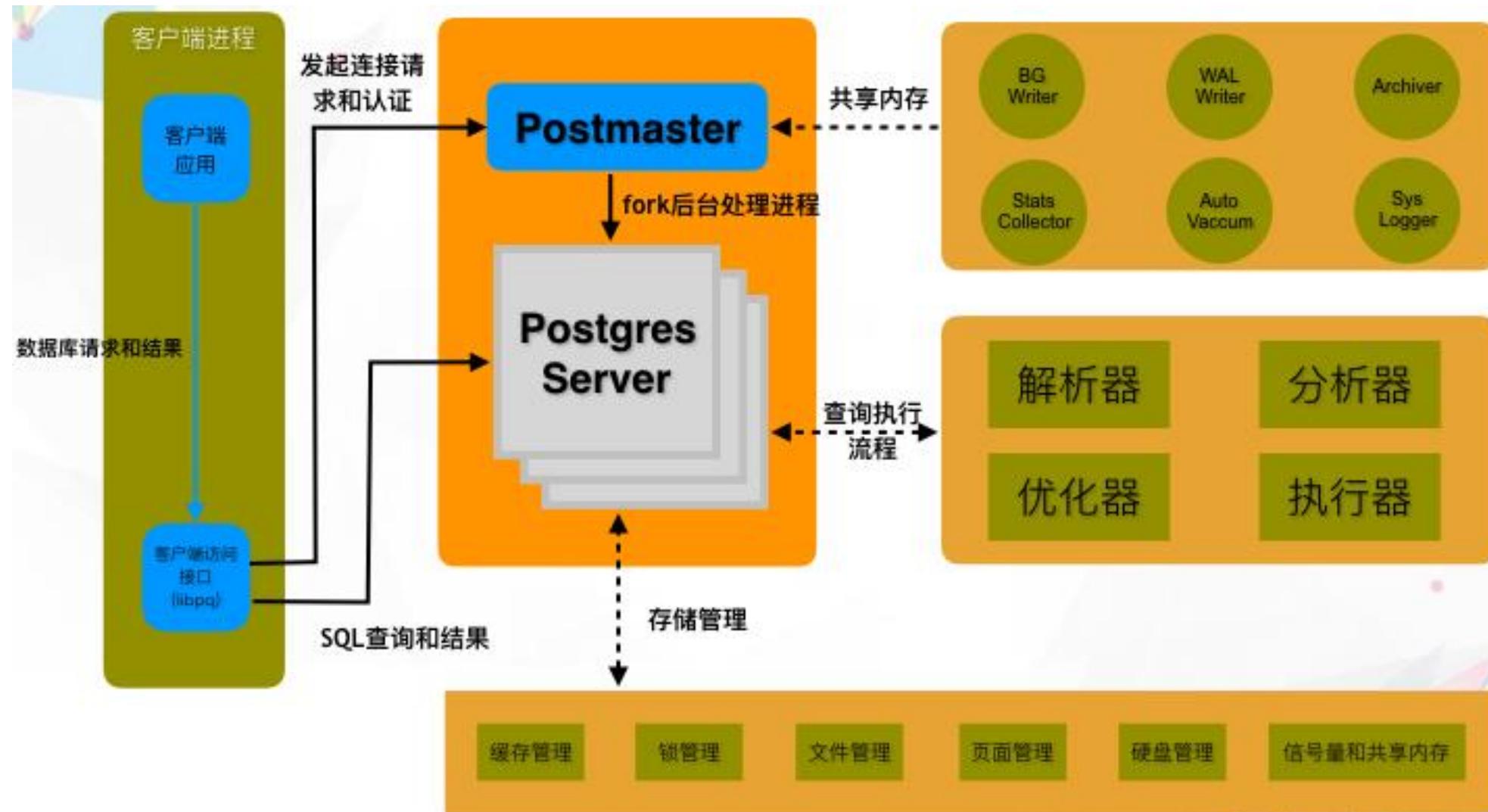




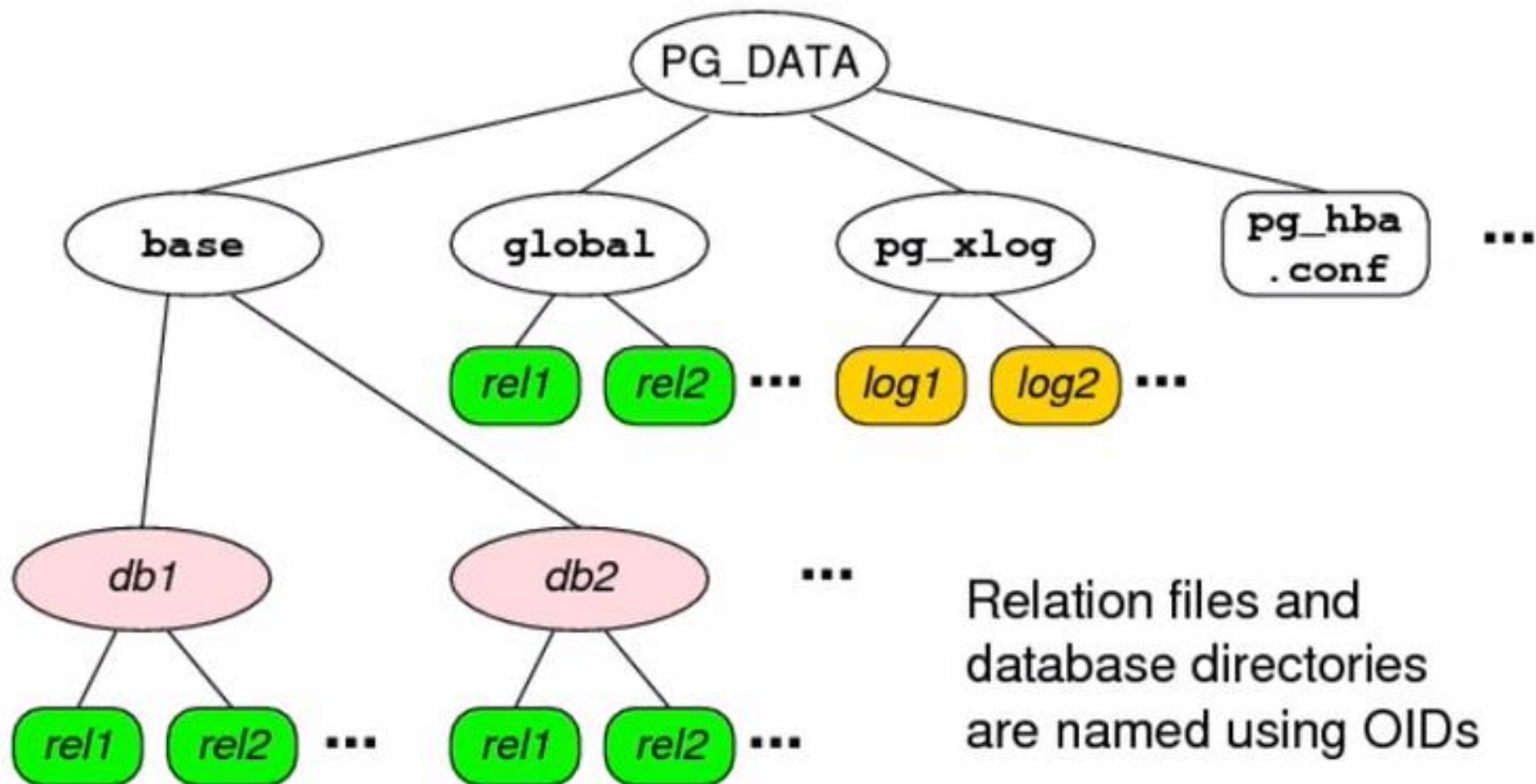
Overall architecture



```
23961 Ss 0:05.64 /usr/local/bin/postmaster (postgres)
23963 S 0:01.13 postmaster: stats buffer process (postgres)
23966 S 0:03.24 postmaster: stats collector process (postgres)
36324 I 0:00.43 postmaster: oddbjorn testdb [local] idle (postgres)
36428 I 0:00.23 postmaster: oddbjorn testdb [local] idle (postgres)
```



File-system architecture:



Chapter 2: Build PostgreSQL on Windows 10

❑ History of PostgreSQL

- Ingres
- PostgreSQL
- more details later For Data Warehouse, Big Data
 - ✓ Greenplum, HAWQ

❑ Build PostgreSQL on Windows 10

- Perl + VS

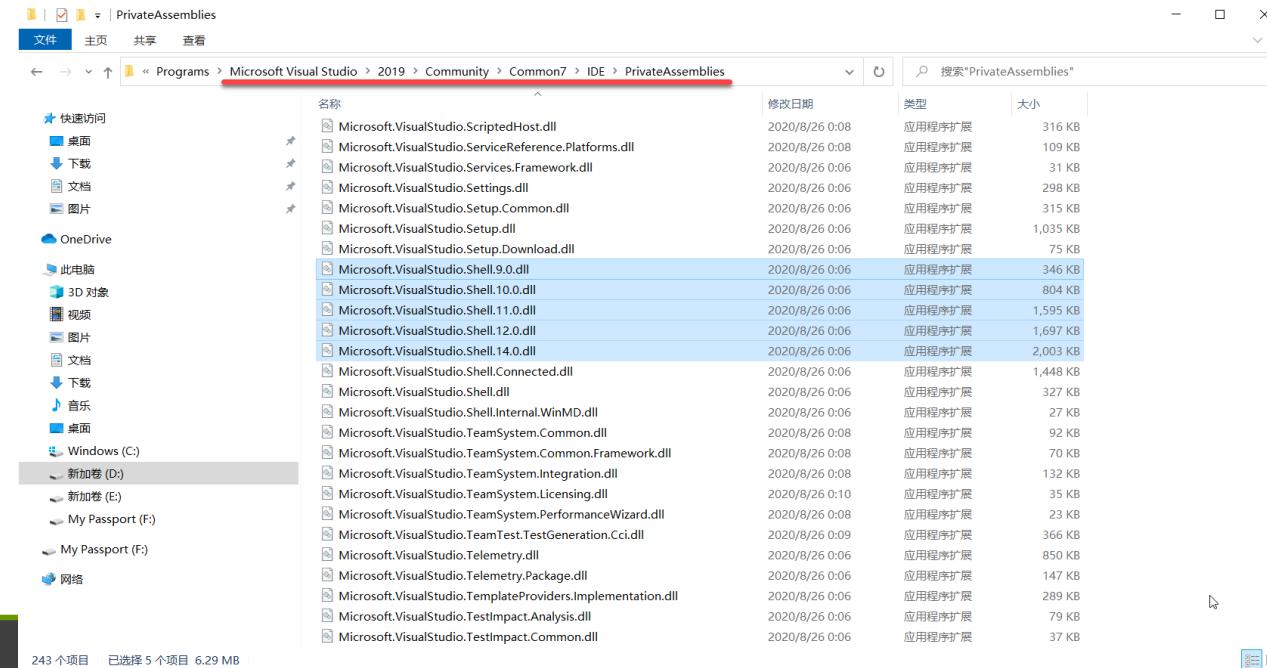
❑ Interact with PostgreSQL

❑ Debug/Reading PostgreSQL with VS

Debug PG

□ Add necessary Shell.dlls

1. Developer Command Prompt for VS 2019 (vs2019开发人员命令提示符) with admin role
2. cd into vs2019的安装目录
➤ D:\Programs\Microsoft Visual Studio\2019\Community\Common7\IDE\PrivateAssemblies



□ Add necessary Shell.dlls

3. gacutil -i Microsoft.VisualStudio.Shell.14.0.dll

```
管理员: Developer Command Prompt for VS 2019
D:\Programs\Microsoft Visual Studio\2019\Community\Common7\IDE\PrivateAssemblies>gacutil -i Microsoft.VisualStudio.Shell
.14.0.dll
Microsoft (R) .NET Global Assembly Cache Utility. Version 4.0.30319.0
版权所有(C) Microsoft Corporation. 保留所有权利。
将程序集添加到缓存失败： 系统找不到指定的文件。

D:\Programs\Microsoft Visual Studio\2019\Community\Common7\IDE\PrivateAssemblies>gacutil -i Microsoft.VisualStudio.Shell
.14.0.dll
Microsoft (R) .NET Global Assembly Cache Utility. Version 4.0.30319.0
版权所有(C) Microsoft Corporation. 保留所有权利。
程序集已成功添加到缓存中

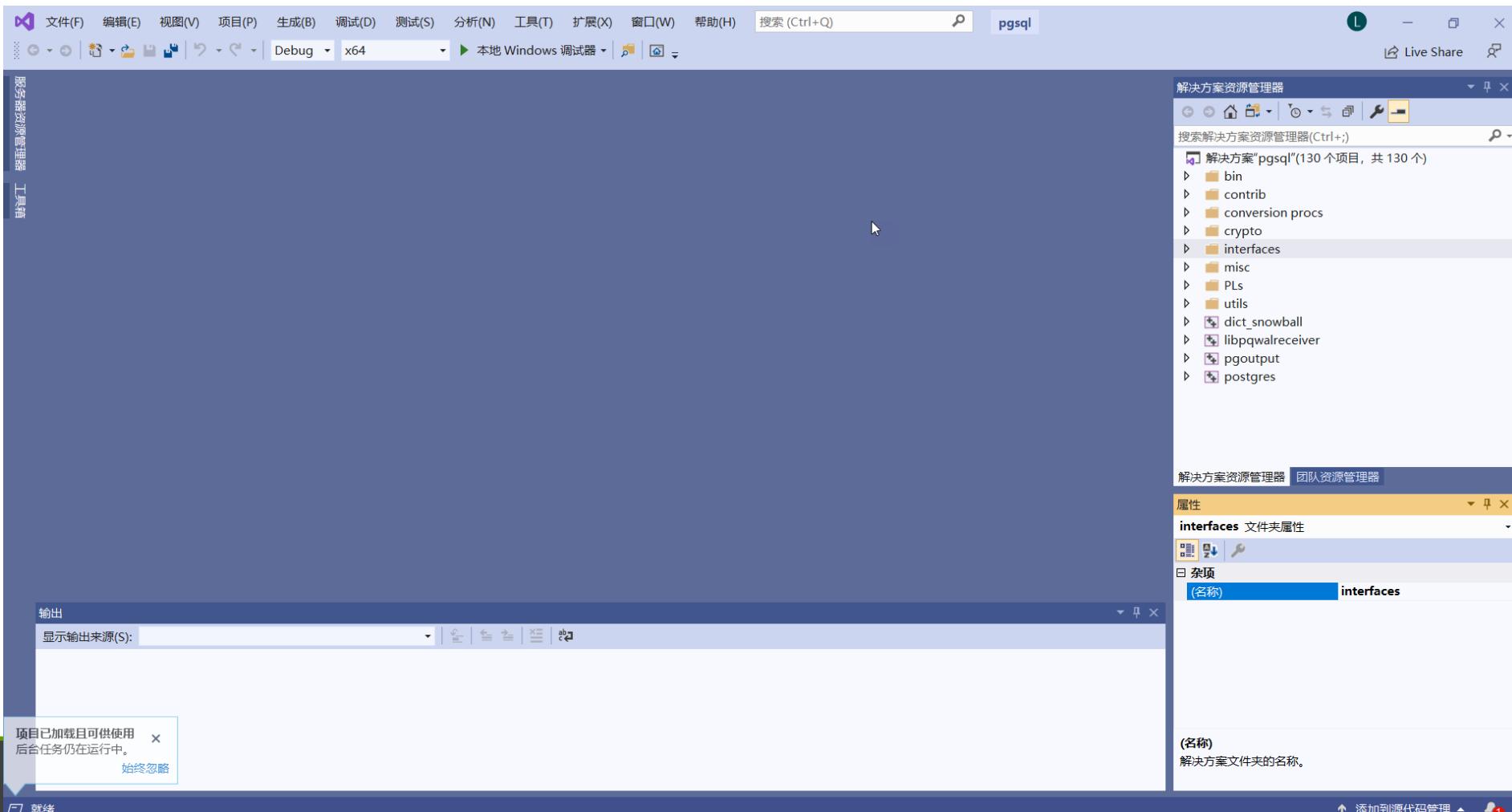
D:\Programs\Microsoft Visual Studio\2019\Community\Common7\IDE\PrivateAssemblies>gacutil -i Microsoft.VisualStudio.Shell
.12.0.dll
Microsoft (R) .NET Global Assembly Cache Utility. Version 4.0.30319.0
版权所有(C) Microsoft Corporation. 保留所有权利。
程序集已成功添加到缓存中

D:\Programs\Microsoft Visual Studio\2019\Community\Common7\IDE\PrivateAssemblies>gacutil -i Microsoft.VisualStudio.Shell
.11.0.dll
Microsoft (R) .NET Global Assembly Cache Utility. Version 4.0.30319.0
版权所有(C) Microsoft Corporation. 保留所有权利。
程序集已成功添加到缓存中

D:\Programs\Microsoft Visual Studio\2019\Community\Common7\IDE\PrivateAssemblies>
```

□ Open pgsql.sln with VS Community 2019

■ D:\postgresql-12.4\pgsql.sln

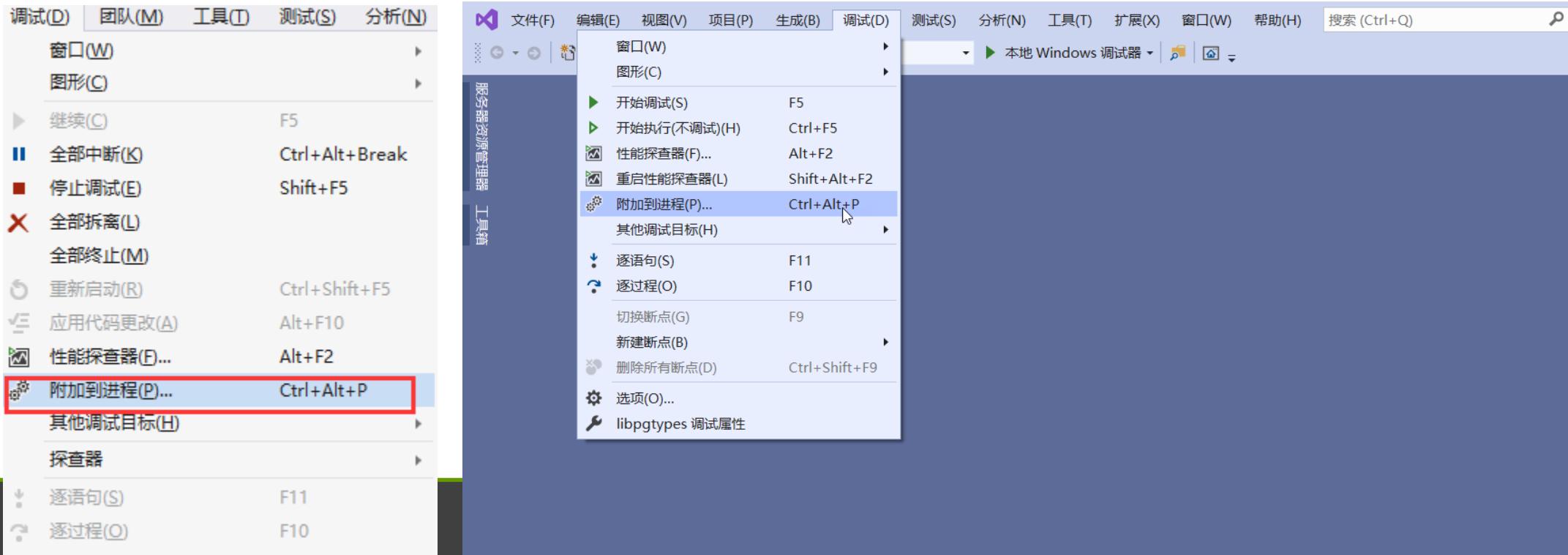


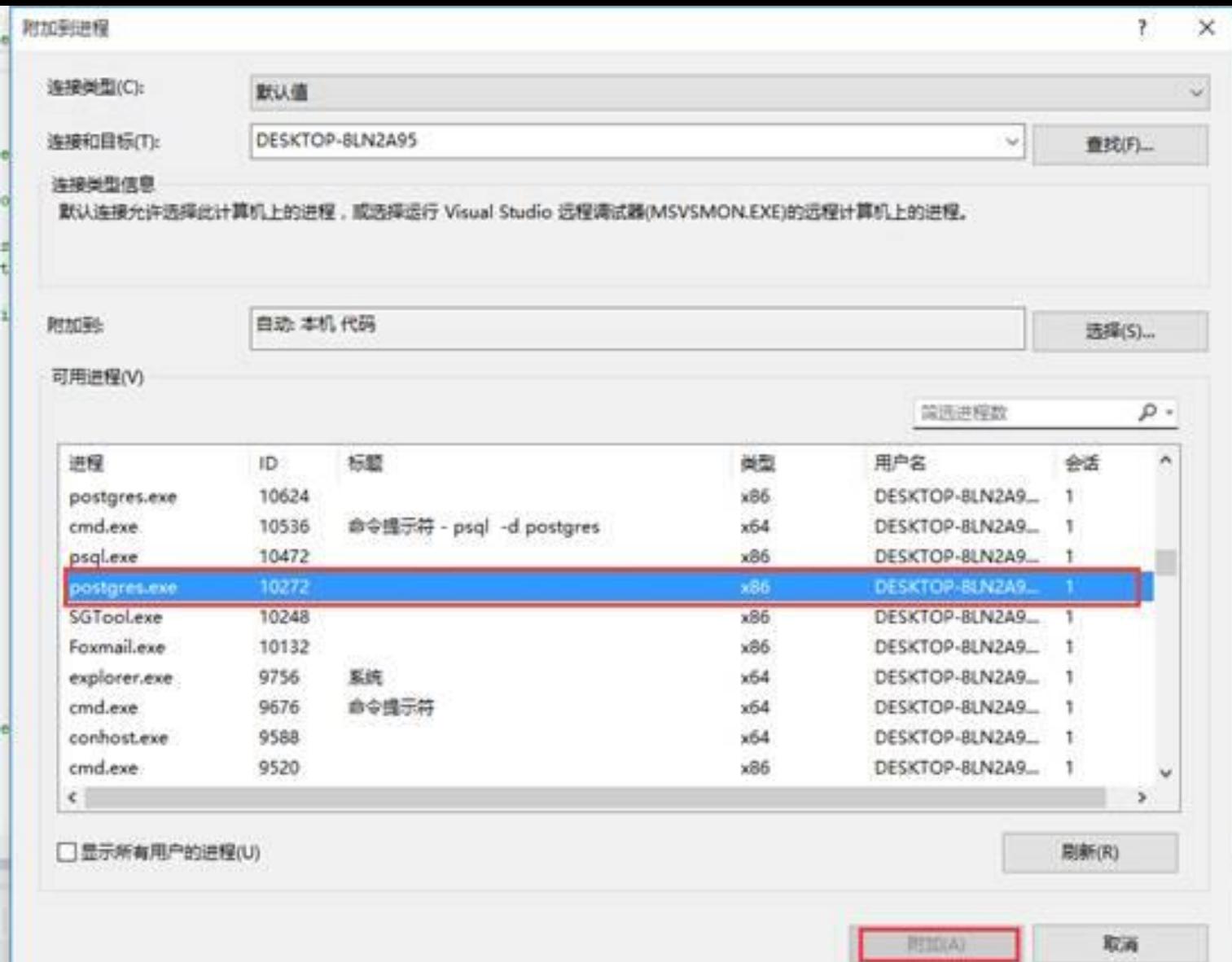
□ Add postgres server into VS

1. Get ID number of postgres process

```
postgres=# select pg_backend_pid()
postgres-# ;
pg_backend_pid
-----
10272
(1 row)
```

2. Add postgres process into VS





□ Stop all interrupts first, then add your own interrupts, RUN ...

The screenshot shows a debugger interface with two main windows. On the left is a menu bar with options like 调试(D), 团队(M), 工具(T), 测试(S), and 分析(N). A red box highlights the "全部中断(K)" option under the "继续(C)" section, which is associated with the keyboard shortcut Ctrl+Alt+Break. On the right is a code editor window titled "postgres" showing assembly code. A red box highlights a specific instruction at line 1499: "rc = WaitForMultipleObjects(set->nevents + 1, set->handles, FALSE, cur_timeout);". Below the code editor is a call stack window with a red border, listing external code frames. The bottom right corner of the slide contains the number 55.

```
r = WSASend(cur_event->fd, &buf, 1, &sent, 0, NULL, NULL);
if (r == 0 || WSAGetLastError() != WSAEWOULDBLOCK)
{
    occurred_events->pos = cur_event->pos;
    occurred_events->user_data = cur_event->user_data;
    occurred_events->events = WL_SOCKET_WRITEABLE;
    occurred_events->fd = cur_event->fd;
    return 1;
}

/*
 * Sleep.
 *
 * Need to wait for ->nevents + 1, because signal handle is in [0].
 */
rc = WaitForMultipleObjects(set->nevents + 1, set->handles, FALSE,
                           cur_timeout);

/* Check return code */
if (rc == WAIT_FAILED)
    elog(ERROR, "WaitForMultipleObjects() failed: error code %lu",
         GetLastError());
```

名称
[外部代码]
(下面的框架可能不正确和/或缺失，没有为 ntdll.dll 加载符号)

```
postgres.exe!WaitEventSetWaitBlock(WaitEventSet * set, int cur_timeout, WaitEvent * occurred_events, int nevents) 行 1499
postgres.exe!WaitEventSetWait(WaitEventSet * set, long timeout, WaitEvent * occurred_events, int nevents) 行 1007
postgres.exe!secure_read(Port * port, void * ptr, unsigned int len) 行 149
postgres.exe!pq_recvbuf() 行 940
postgres.exe!pq_getbyte() 行 983
postgres.exe!socketBackend(StringInfoData * inBuf) 行 326
postgres.exe!readCommand(StringInfoData * inBuf) 行 499
postgres.exe!PostgresMain(int argc, char ** argv, const char * dbname, const char * username) 行 4019
postgres.exe!BackendRun(Port * port) 行 4294
postgres.exe!SubPostmasterMain(int argc, char ** argv) 行 4792
postgres.exe!main(int argc, char ** argv) 行 205
```

[外部代码]

□ Your input will be traced by your interrupts

```
postgres=# select * from aaaa;
```

The screenshot shows a debugger interface with assembly code and a call stack.

Assembly Code:

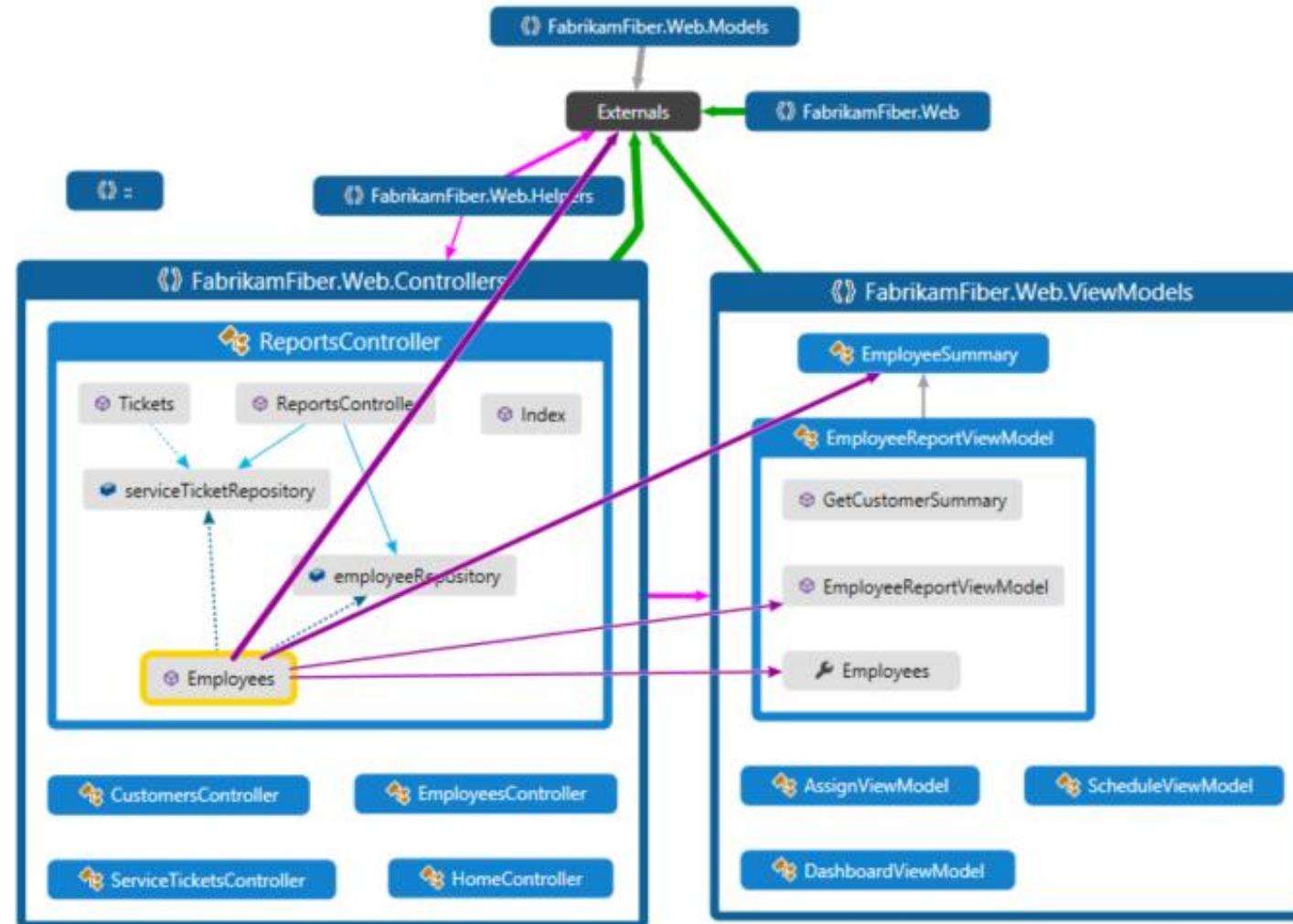
```
796 int
797     errmsg(const char *fmt,...)
798     { 已用时间=12,634ms
799         ErrorData *edata = &errordata[errordata_stack_depth];
800         MemoryContext oldcontext;
801
802         recursion_depth++;
803         CHECK_STACK_DEPTH();
804         oldcontext = MemoryContextSwitchTo(edata->assoc_context);
805
806         edata->message_id = .fmt;
807         EVALUATE_MESSAGE(edata->domain, message, false, true);
808
809         MemoryContextSwitchTo(oldcontext);
```

Call Stack:

- postgres.exe!errmsg(const char * fmt, ...) 行 798
- postgres.exe!parserOpenTable(ParseState * pstate, const RangeVar * relation, int lockmode) 行 1157
- postgres.exe!addRangeTableEntry(ParseState * pstate, RangeVar * relation, Alias * alias, char inh, char inFromC) 行 1197
- postgres.exe!transformTableEntry(ParseState * pstate, RangeVar * r) 行 439
- postgres.exe!transformFromClauseItem(ParseState * pstate, Node * n, RangeTblEntry * top_rte, int * top_rti, List * namespace) 行 872
- postgres.exe!transformFromClause(ParseState * pstate, List * frmList) 行 130
- postgres.exe!transformSelectStmt(ParseState * pstate, SelectStmt * stmt) 行 1153
- postgres.exe!transformStmt(ParseState * pstate, Node * parseTree) 行 270
- postgres.exe!transformTopLevelStmt(ParseState * pstate, Node * parseTree) 行 215
- postgres.exe!parse_analyze(Node * parseTree, const char * sourceText, unsigned int * paramTypes, int numParams) 行 108
- postgres.exe!pg_analyze_and_rewrite(Node * parseTree, const char * query_string, unsigned int * paramTypes, int numParams) 行 650
- postgres.exe!exec_simple_query(const char * query_string) 行 1011
- postgres.exe!PostgresMain(int argc, char ** argv, const char * dbname, const char * username) 行 4074
- postgres.exe!BackendRun(Post * port) 行 4294
- postgres.exe!SubPostmasterMain(int argc, char ** argv) 行 4792
- postgres.exe!main(int argc, char ** argv) 行 205
- (外部代码)

[下面的框架可能不正确和/或缺失，没有为 kernel32.dll 加载符号]

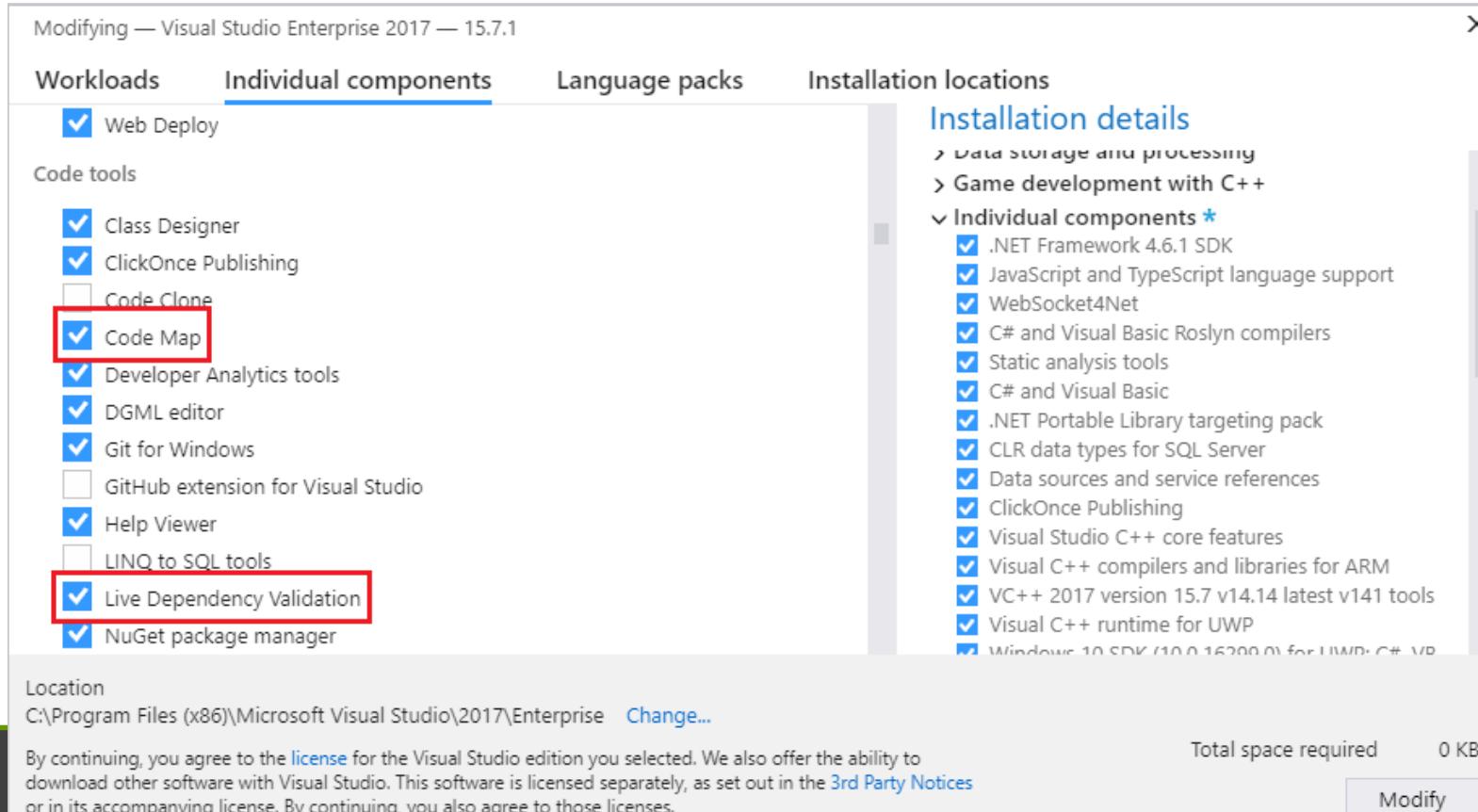
Tools in VS for code reading



When installing VS

□ 若要在 Visual Studio 中创建代码图, 请首先安装 **代码图** 和 **实时依赖项验证** 组件:

- 打开 Visual Studio 安装程序。可以通过 Windows "开始" 菜单或在 Visual Studio 中选择 "工具" " > 获取工具和功能" 来打开它。
- 选择“各个组件”选项卡。
- 向下滚动到 " 代码工具 " 部分, 并选择 " 代码图 " 和 " 实时依赖项验证 "。

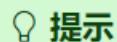


添加代码图

您可以创建空代码图，并将项拖动到其上，包括程序集引用、文件和文件夹，也可以为解决方案的全部或部分生成代码图。

添加空代码图：

1. 在“解决方案资源管理器”中，打开顶级解决方案节点的快捷菜单 ***。选择“添加 > 新项”。
2. 在“添加新项”对话框中的“已安装”下，选择“常规”类别。
3. 选择**定向关系图文档 (.dgml)** 模板，然后选择“添加”。



提示

此模板可能不会按字母顺序显示，因此，如果看不到，请向下滚动到模板列表的底部。

解决方案的“**解决方案项**”文件夹中将显示一个空白地图。

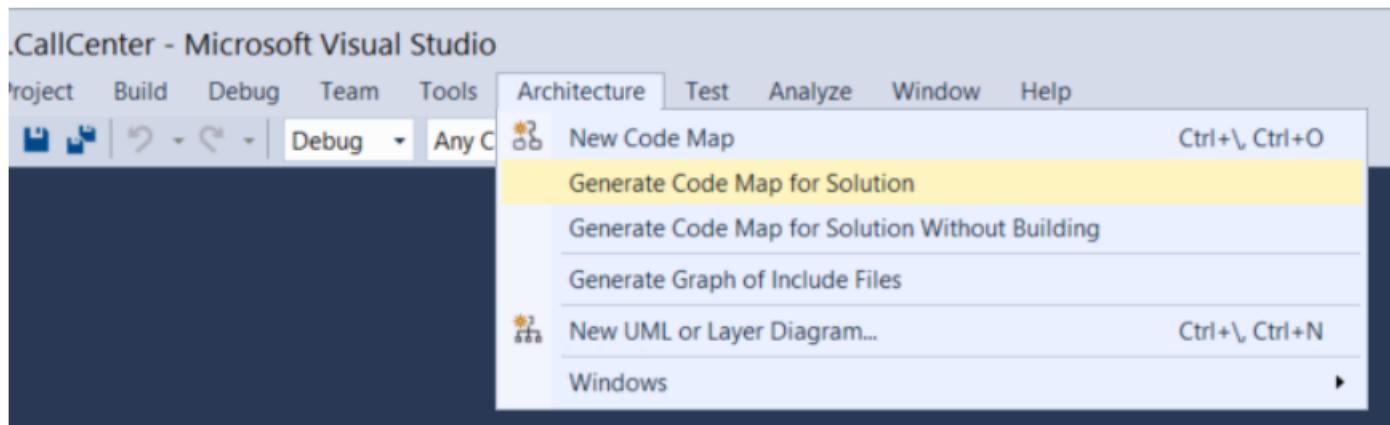
同样，可以通过选择“**体系结构**” “> **新建代码图**” 或“**文件**” “> **新建 > 文件**” 来创建新的代码映射文件，而无需将其添加到解决方案。



为解决方案生成代码图

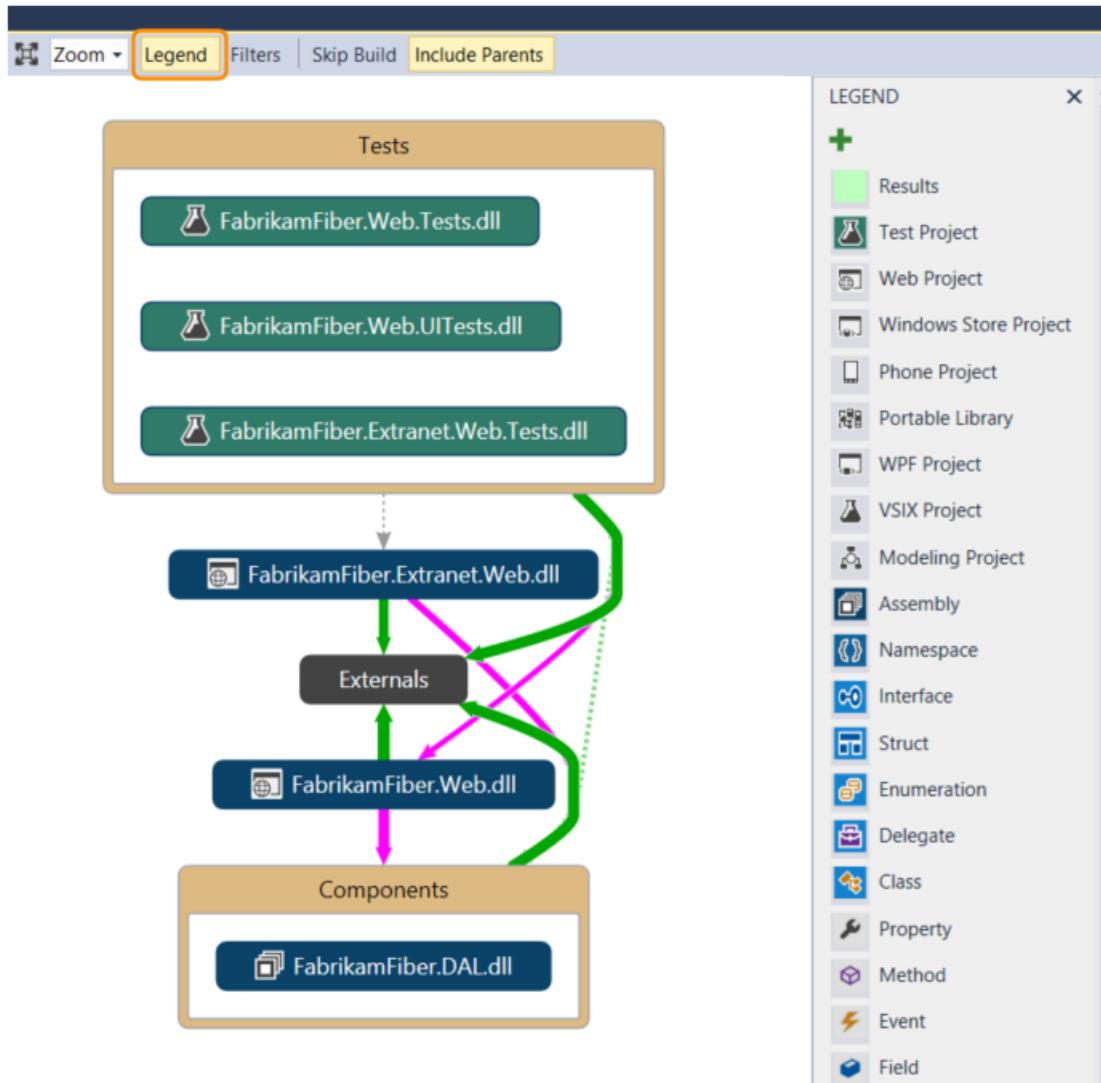
查看解决方案中的所有依赖项：

1. 在菜单栏上，选择 “**体系结构**” > **生成解决方案的代码图**”。 如果代码自上次生成后未发生更改，则可以选择 “**在不生成 >** 的情况下生成解决方案的代码图”。



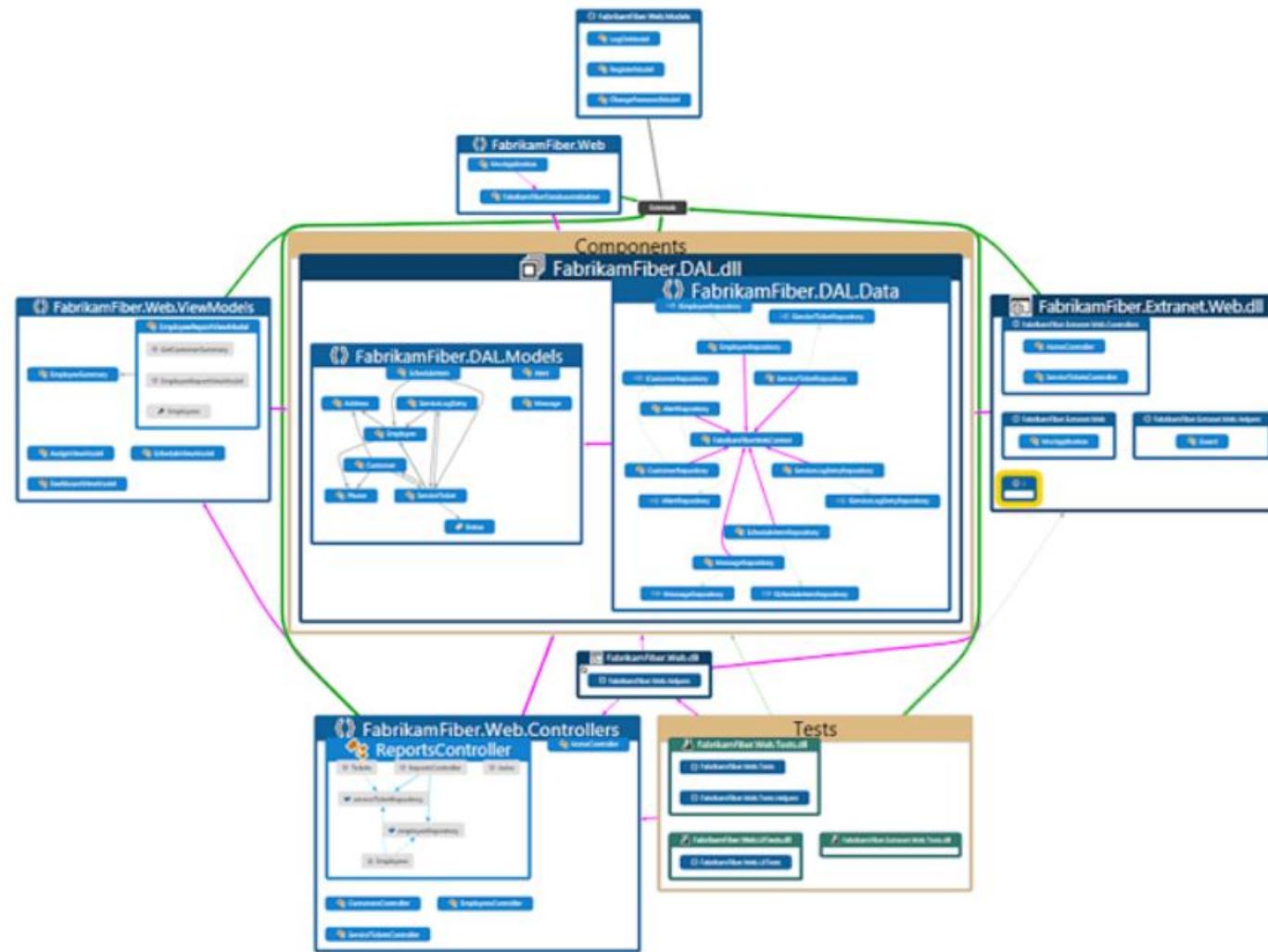
将生成一个映射，其中显示顶级程序集和它们之间的聚合链接。聚合链路越广，它所代表的依赖关系就越多。

2. 使用代码图工具栏上的“图例” **** 按钮以显示或隐藏项目类型图标（如测试项目、Web 项目和 Phone 项目）、代码项（如类、方法和属性）以及关系类型（如继承自、实现和调用）的列表。



此示例解决方案包含解决方案文件夹（“测试”**** 和“组件”****），测试项目、Web 项目和程序集。默认情况下，所有包含关系均以“组”的形式显示，你可以对其进行展开和折叠。**** “外部”组包含你的解决方案之外的任何内容，包括平台依赖项。外部程序集仅显示已使用的项。默认情况下，系统基类型隐藏在代码图中以减少混乱。

3. 若要深入了解代码图中，请展开表示项目和程序集的组。可以通过按“CTRL+A” **** 展开全部内容以选择所有节点，然后选择“组” ****，再选择快捷菜单中的“展开” ****。



4. 但是，这对大型解决方案可能并无作用。事实上，对于复杂的解决方案，内存限制可能会阻止你展开所有组。相反，若要在单个节点内进行查看，则将其展开。将鼠标指针移动到节点的顶部，然后单击出现的 V 形。

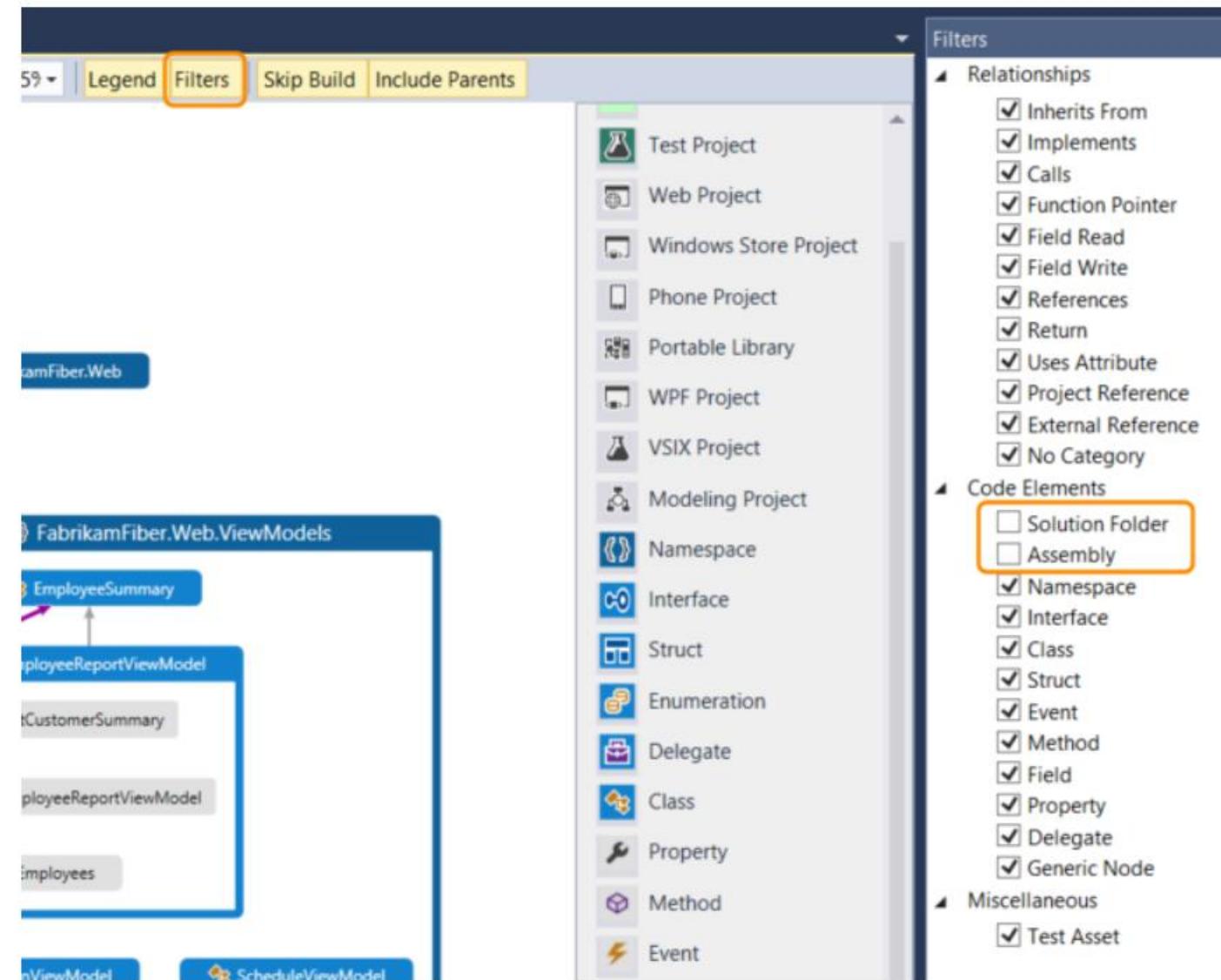


或通过选择项，然后按加号键 () 来使用键盘 + 。 若要查看更深层次的代码，请对命名空间、类型和成员执行相同操作。

💡 提示

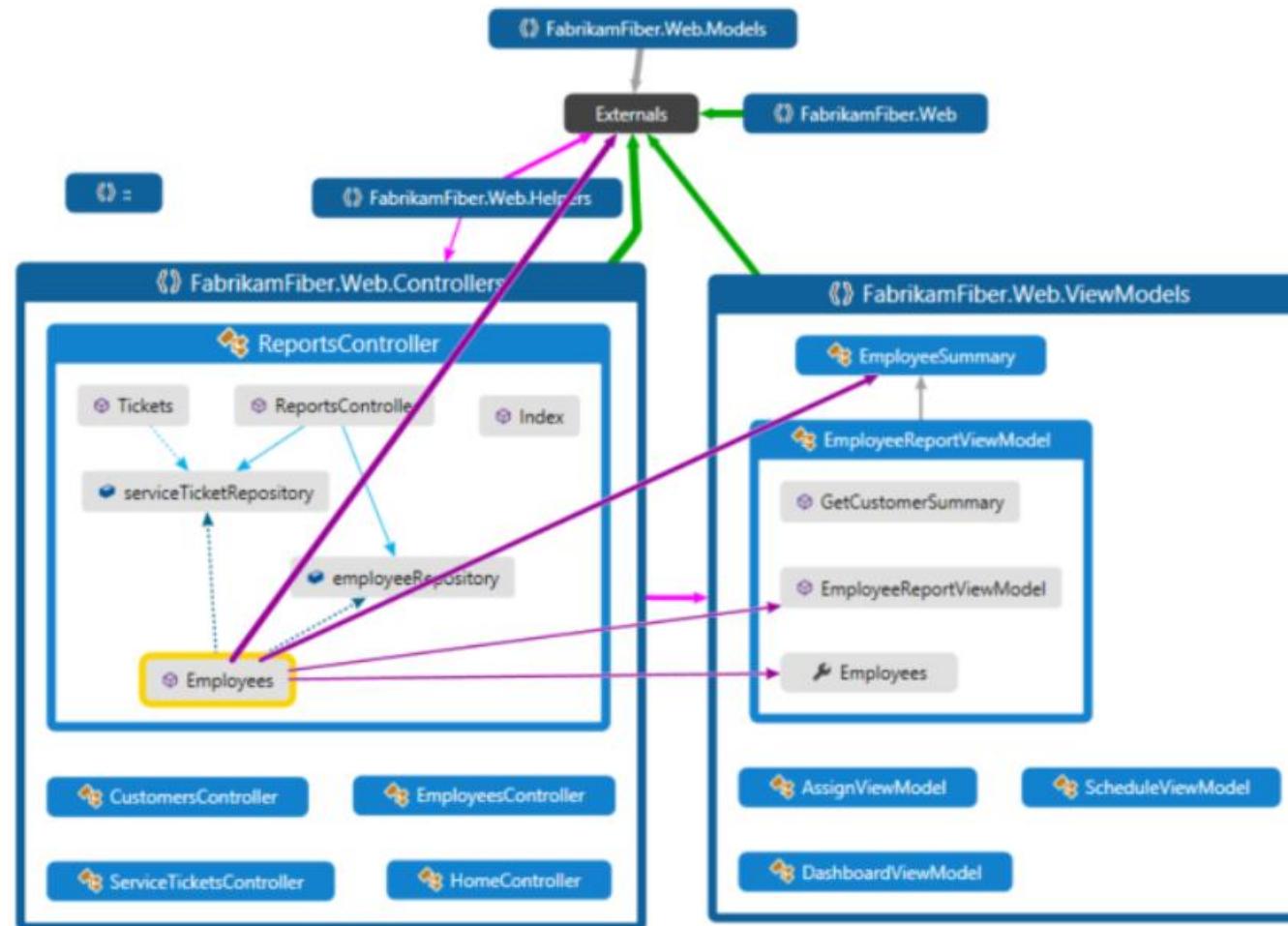
有关使用鼠标、键盘和触控处理代码图的详细信息，请参阅 [浏览和重新排列代码图](#)。

5. 若要简化代码图并将重点关注各个部分，选择代码图工具栏上的“筛选器”****，然后仅选择你感兴趣的节点和链接类型。例如，可以隐藏所有的解决方案文件夹和程序集容器。



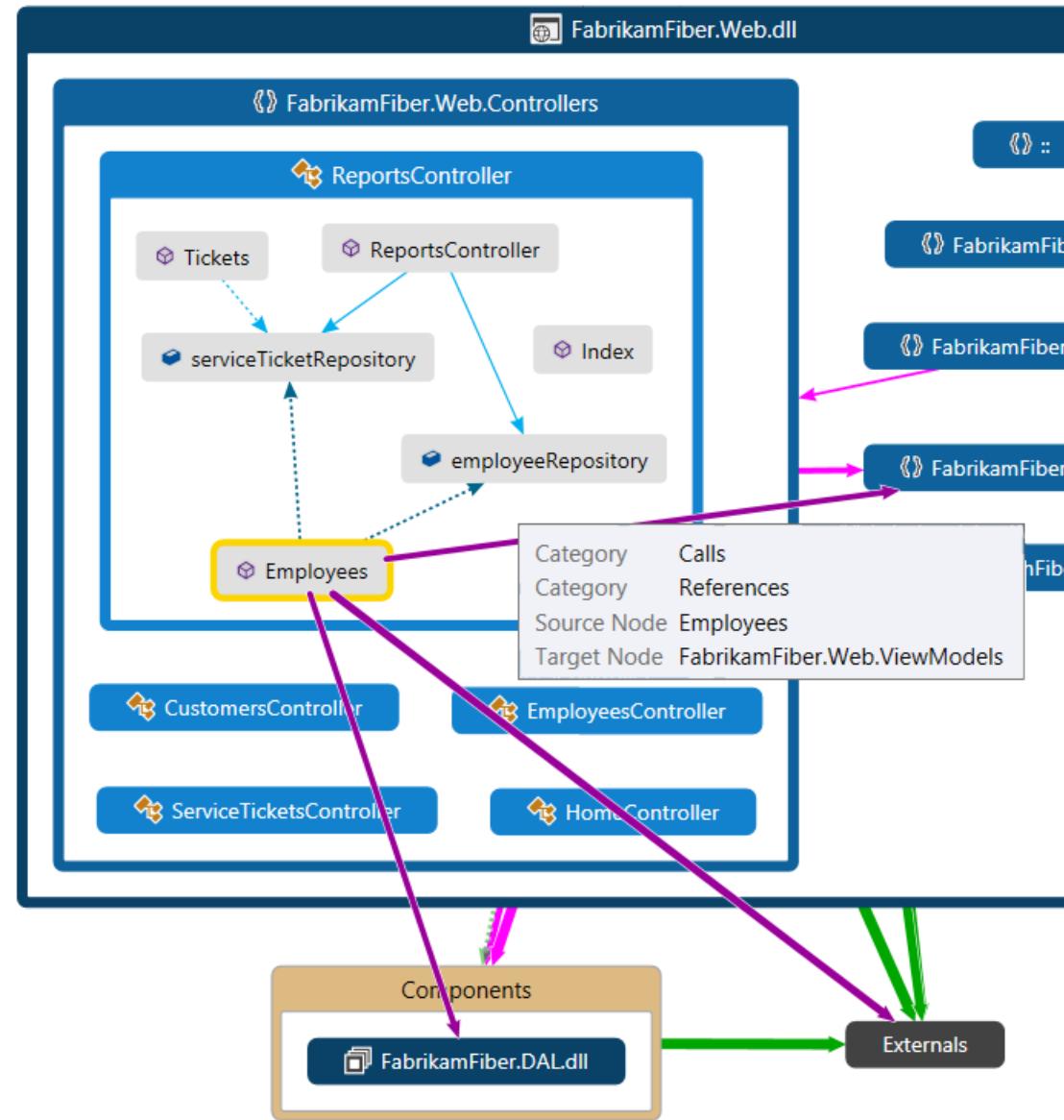
还可以通过从代码图中隐藏或删除单个组和项来简化代码图，而不影响基础解决方案代码。

6. 若要查看各项间的关系，请在代码图中选择各项。链接的颜色指示关系的类型，如“图例”****窗格中所示。

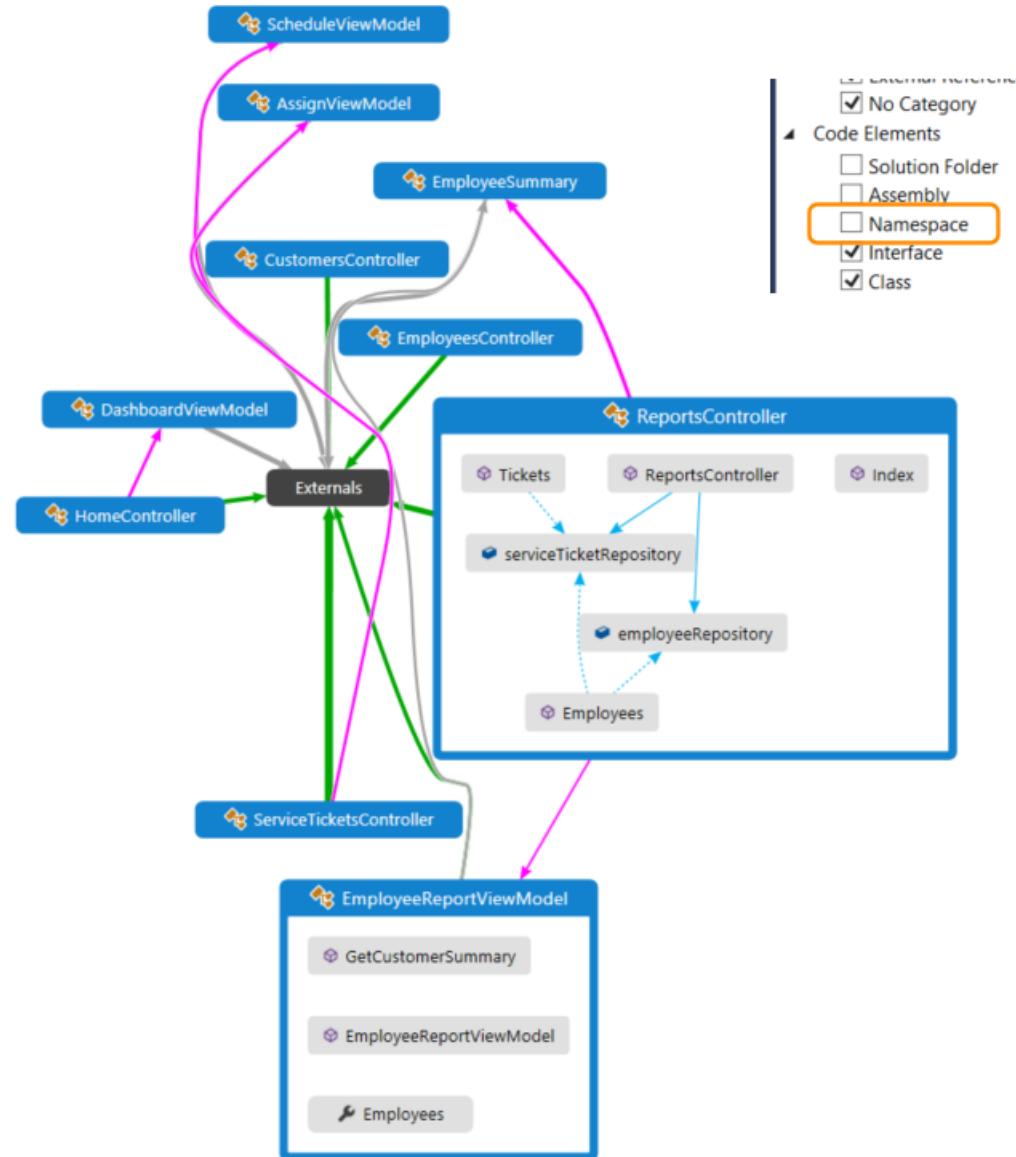


在此示例中，紫色链接为调用，点式链接为引用，浅蓝色链接为字段访问。绿色链接可以是继承，也可能是指示一个以上关系（或“类别”）类型的“聚合链接”。

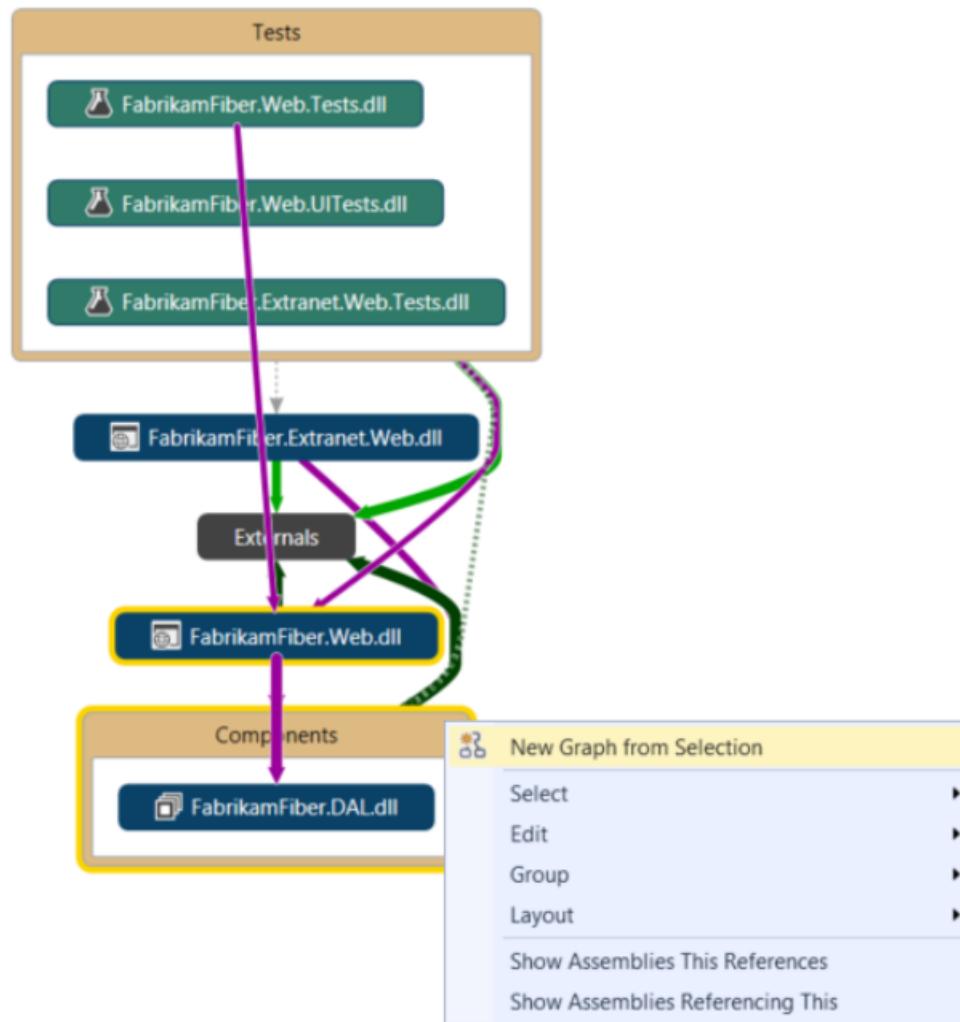
7. 若要获取有关项或链接的详细信息，请将指针移至其顶部，直至出现工具提示。这将显示代码元素或链接所表示的类别的详细信息。



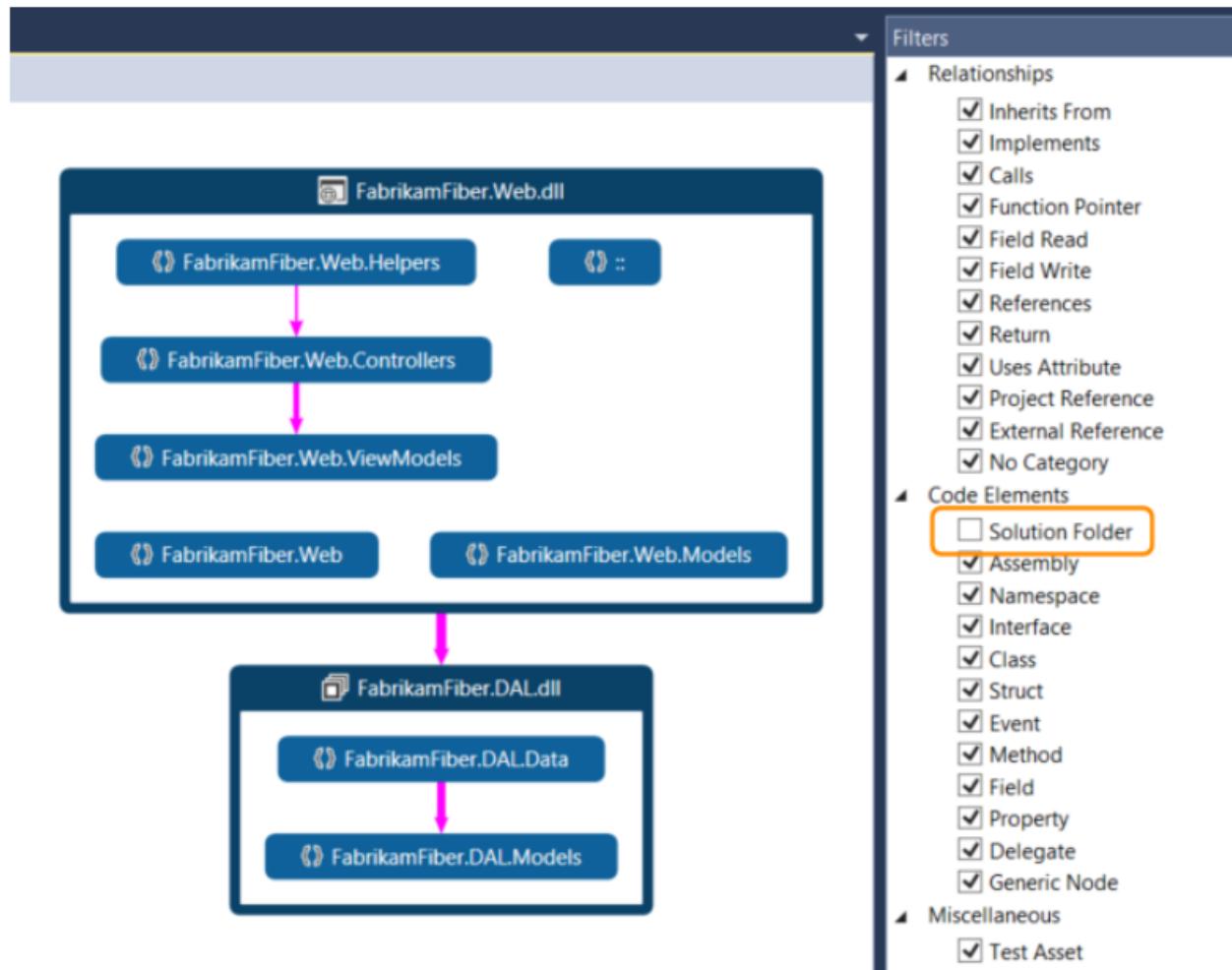
9. 若要重点关注地图的特定部分，可继续删除不感兴趣的项。例如，若要深入了解类和成员视图，只需筛选“筛选器”****窗格中的所有命名空间节点。



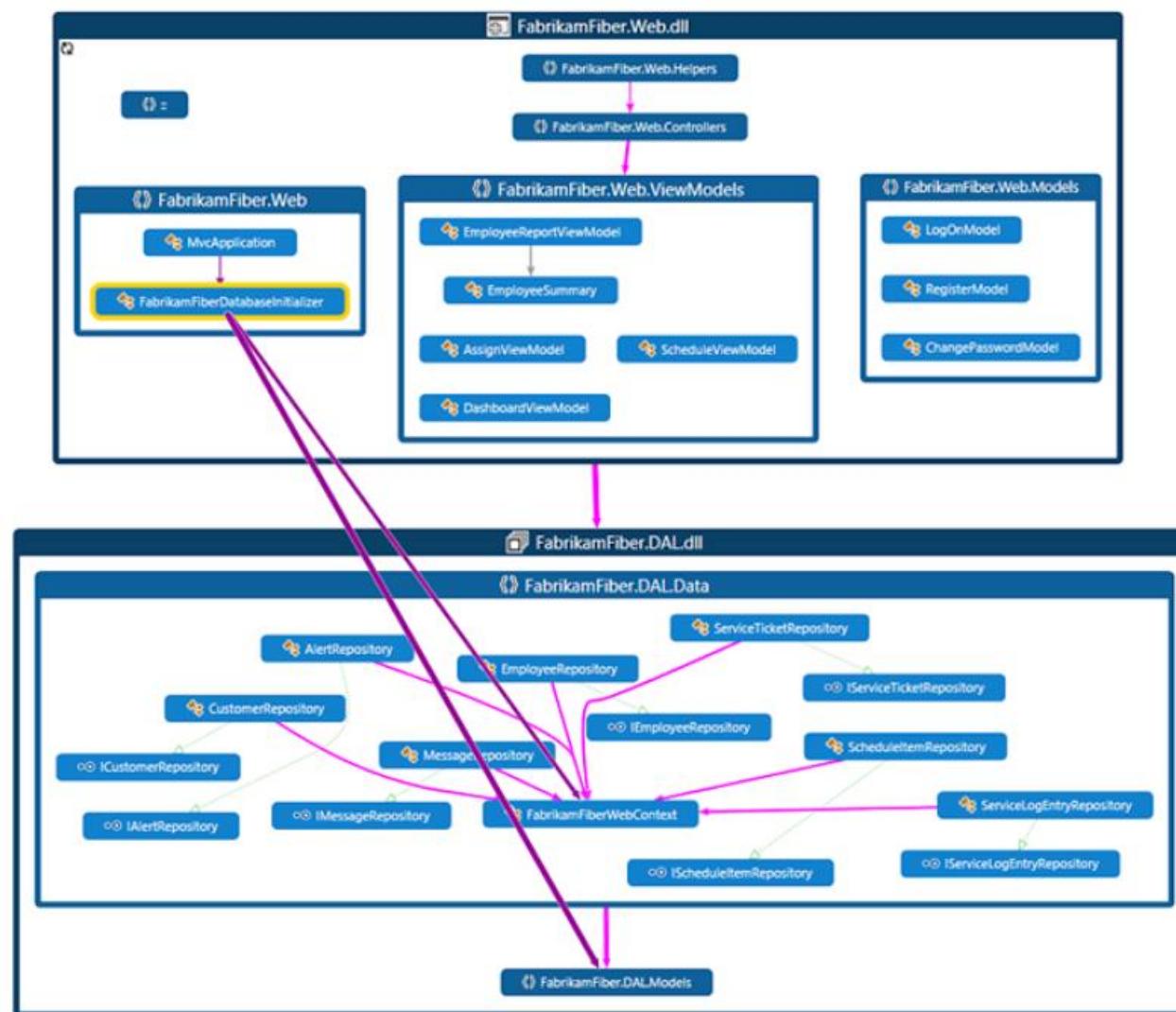
10. 集中精力处理复杂的解决方案代码图的另一种方式是从现有代码图生成包含选定项的新代码图。按住 Ctrl 的同时选择要关注的项，打开快捷菜单，然后从“选择”中选择“新建关系图”。



11. 将包含上下文转入新代码图。 使用“筛选器”窗格隐藏解决方案文件夹以及不想看到的任何其他容器。



12. 展开组，选择代码图中的项，以便查看关系。

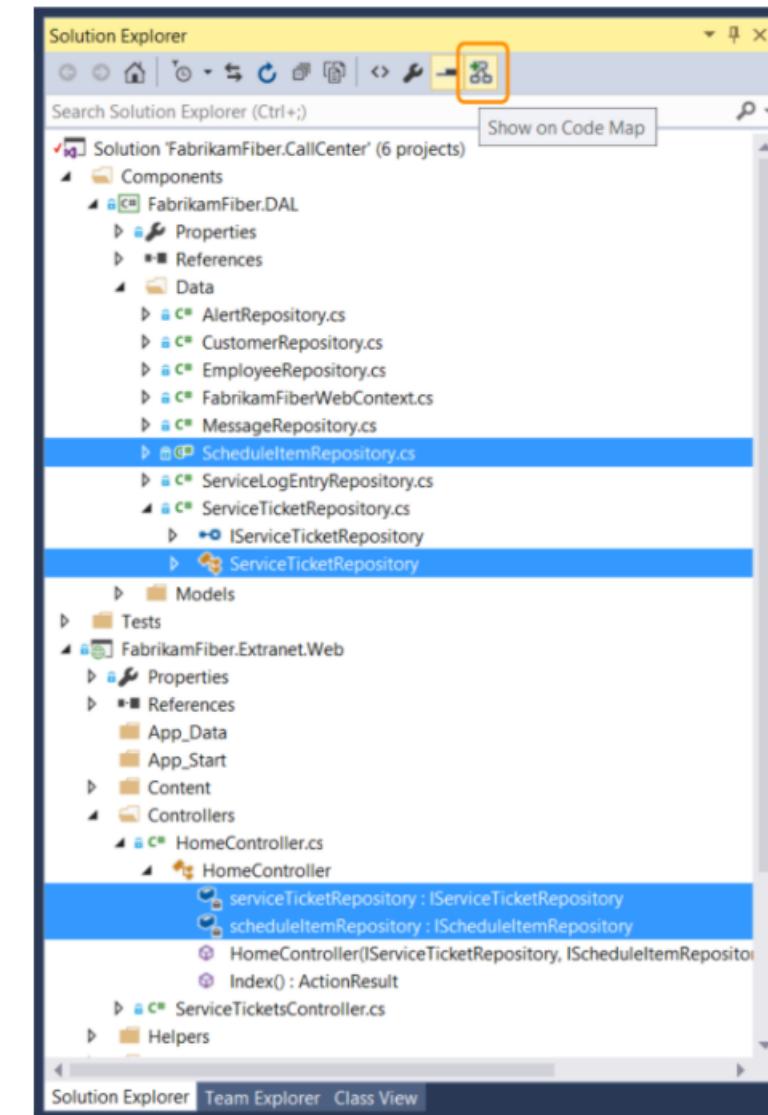
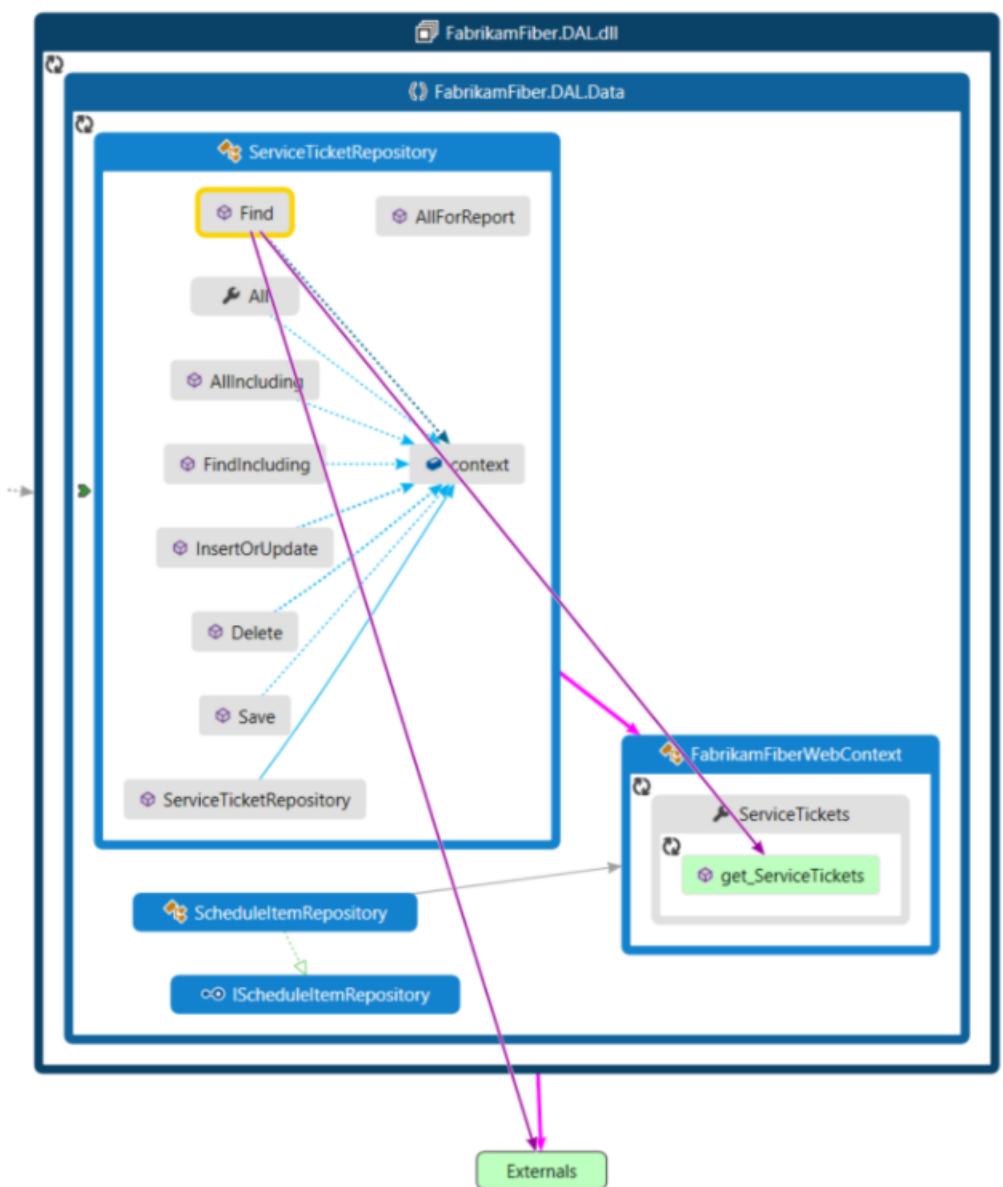


查看代码图中的特定依赖关系

假设你有一个

代码评审，可在某些具有挂起更改的文件中执行。若要查看这些更改中的依赖关系，请从这些文件中创建一个代码图。

1. 在解决方案资源管理器中，选择要映射的项目、程序集引用、文件夹、文件、类型或成员。



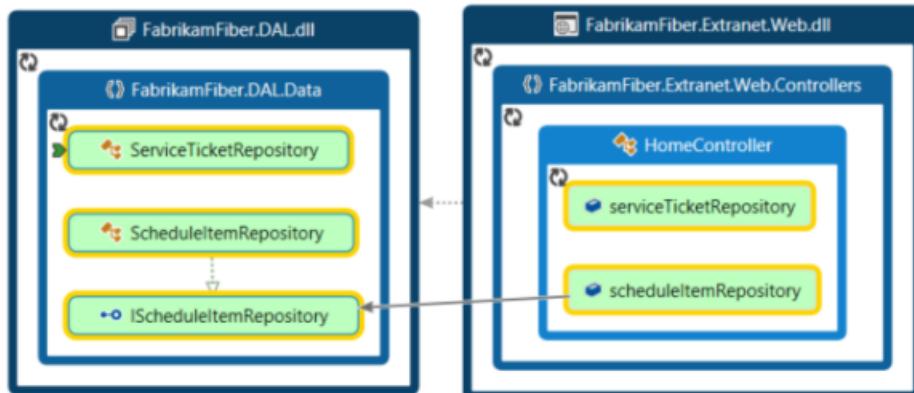
2. 在解决方案资源管理器 工具栏上，选择“在代码图上显示”。或者，打开一个或一组项目的快捷菜单，然后选择“在代码图上显示”。

还可以将项从 解决方案资源管理器、类视图 或 对象浏览器 拖动到 新 的或现有的代码图中。若要包含项的父层次结构，请在拖动项时按住 Ctrl 键，或者使用代码图工具栏上的“包括父级”按钮来指定默认操作。你还可以从 Visual Studio 外（例如，从 Windows 资源管理器）中拖动程序集文件。

① 备注

当你从在多个应用之间共享的项目（如 Windows Phone 或 Microsoft Store）添加项时，这些项将与当前活动的应用程序项目一起显示在映射上。如果你将上下文更改为另一个应用程序项目并从共享的项目添加更多项，则这些项现在将与最近活动的应用程序项目一起显示。你对代码图上的项执行的操作仅适用于共享相同上下文的项。

3. 代码图将显示在选定项的包含程序集内的选定项。

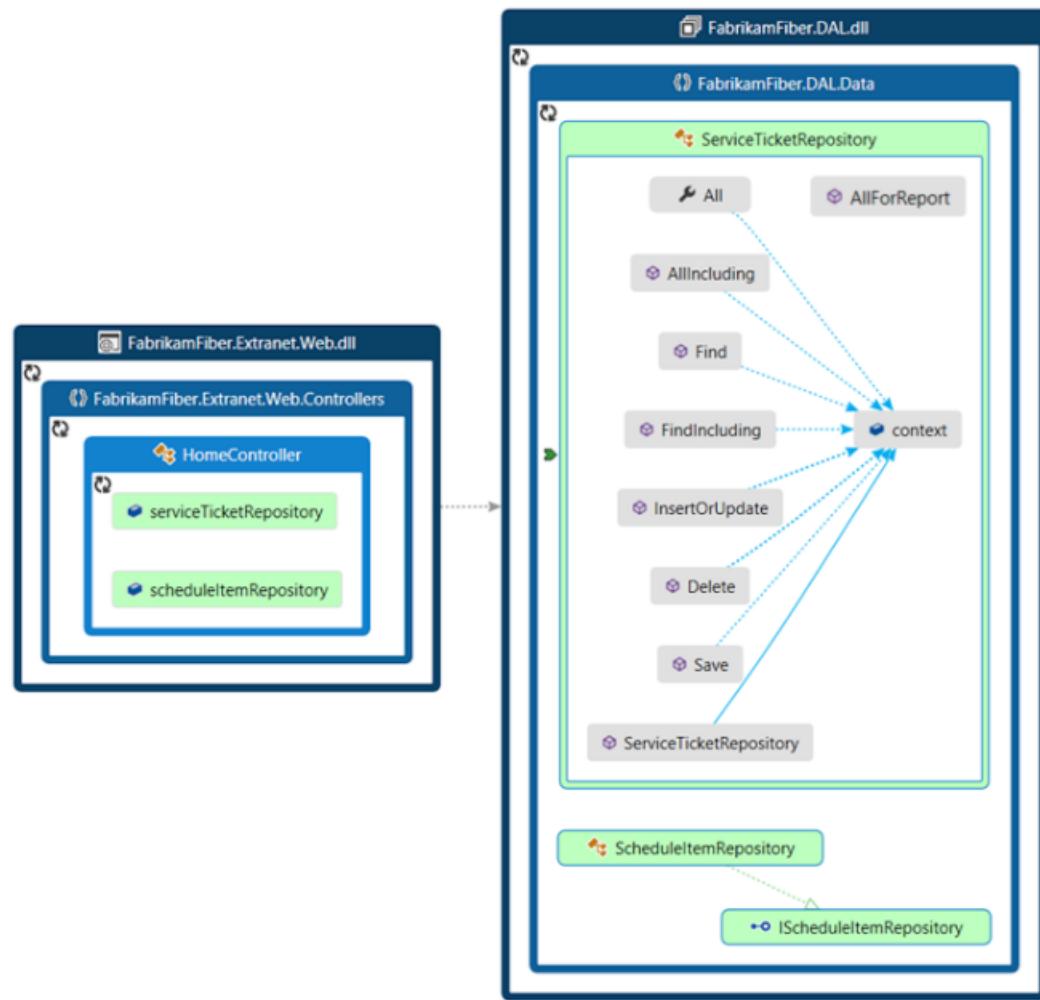


4. 如要查看项，请把项展开。将鼠标指针移动到项的顶部，然后单击出现的 V 形（向下键）。



若要展开所有项，请使用Ctrl + A选择它们，然后打开映射的快捷菜单并选择“组 > 展开”。然而，如果展开所有组会生成不可用的代码图或产生内存问题，则该选项不可用。

5. 如果需要，请继续展开感兴趣的项，直到类和成员级别。



若要查看代码中的成员，但不显示在地图上，请单击组左上角的“重新提取子级”图标。

6. 若要查看更多与代码图上的项相关的项,请选择其中一个,再选择代码图工具栏上的“显示相关内容”****,然后选择要添加到图中的相关项的类型。或者,选择一个或多个项,打开快捷菜单,然后为要添加到映射的相关项的类型选择“显示”选项。例如:

对于 程序集,请选择:

选项	说明
显示此项引用的程序集	添加此程序集引用的程序集。外部程序集将显示在“外部”****组中。
显示引用此项的程序集	在解决方案中添加引用此程序集的程序集。

对于 命名空间,请选择“显示包含程序集”****(如果它不可见)。

对于 类 或 接口,请选择:

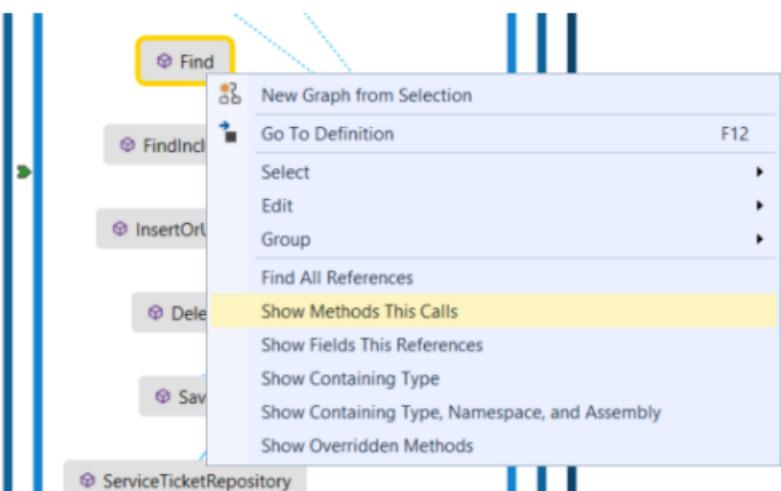
选项	描述
显示基类型	对于类,添加基类和实现的接口。 对于接口,请添加基接口。
显示派生类型	对于类,添加派生类。 对于接口,请添加派生接口和实现类或结构。
显示此项引用的类型	添加此类使用的所有类及其成员。
显示引用此项的类型	添加使用此类的所有类及其成员。
显示包含命名空间	添加父命名空间。
显示包含命名空间和包含程序集	添加父容器的层次结构。
显示所有基类型	以递归方式添加基类或接口层次结构。
显示所有派生类型	对于类,以递归方式添加所有派生类。 对于接口,请以递归方式添加所有派生接口和实现类或结构。

对于 方法,请选择:

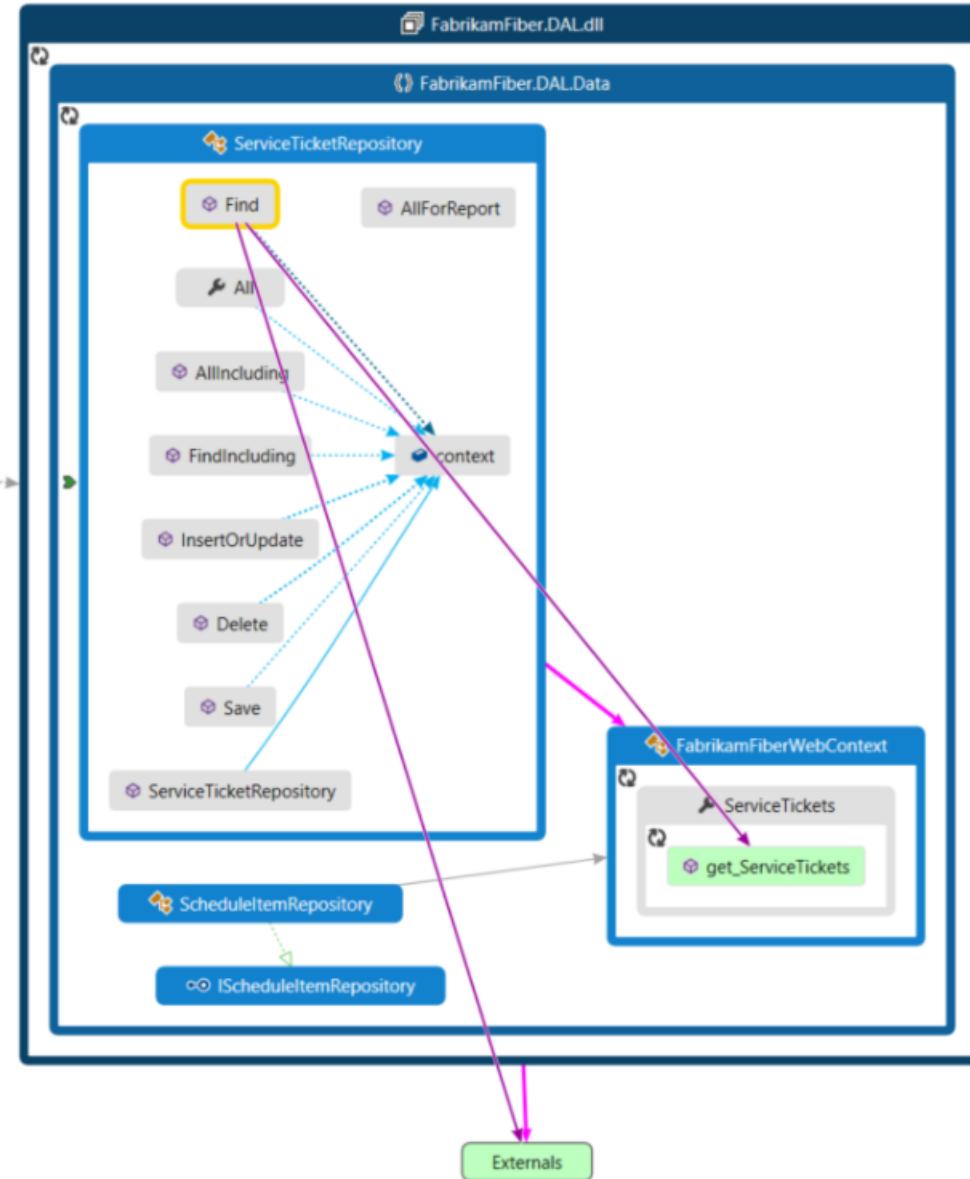
选项	说明
显示此项调用的方法	添加此方法调用的方法。
显示此项引用的字段	添加此方法引用的字段。
显示包含类型	添加父类型。
显示包含类型、包含命名空间和包含程序集	添加父容器的层次结构。
显示重写的方法	对于重写其他方法或者实现接口方法的方法,在已重写的基类和已实现的接口方法(如果有)中添加所有抽象方法或虚方法。

对于 字段 或 属性,请选择:

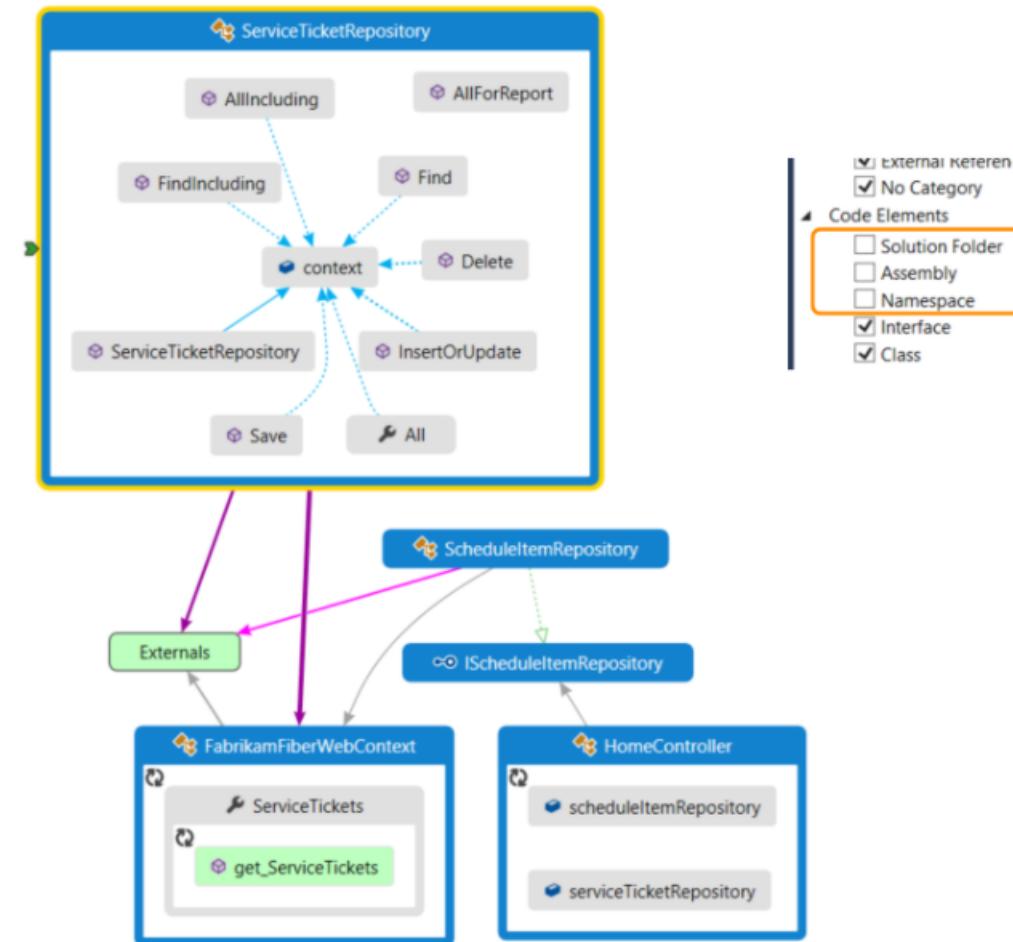
选项	说明
显示包含类型	添加父类型。
显示包含类型、包含命名空间和包含程序集	添加父容器的层次结构。



7. 代码图显示关系。在此示例中，映射显示方法调用的方法 `Find` 及其在解决方案中或在外部的位置。



■ 8. 若要简化代码图并将重点关注各个部分，选择代码图工具栏上的“筛选器”****，然后仅选择你感兴趣的节点和链接类型。例如，关闭解决方案文件夹、程序集和命名空间的显示。



Reading codes – Understand

scitools.com

<https://scitools.com/>

126网易免费邮--你的 北京交通大学·校园 Ebook-dl | Free Dow 西林街 :: 全能型网盘 难以忽视的真相 2,资 VGG Practical https://libgen.pw

Jnderstand™

[Free Trial](#) [Update](#) [Buy](#)

Legacy Code Maintenance

Visual Static Analysis

C++ Editor

GitAhead™

[Free Trial](#) [Buy](#)

Revision Control

