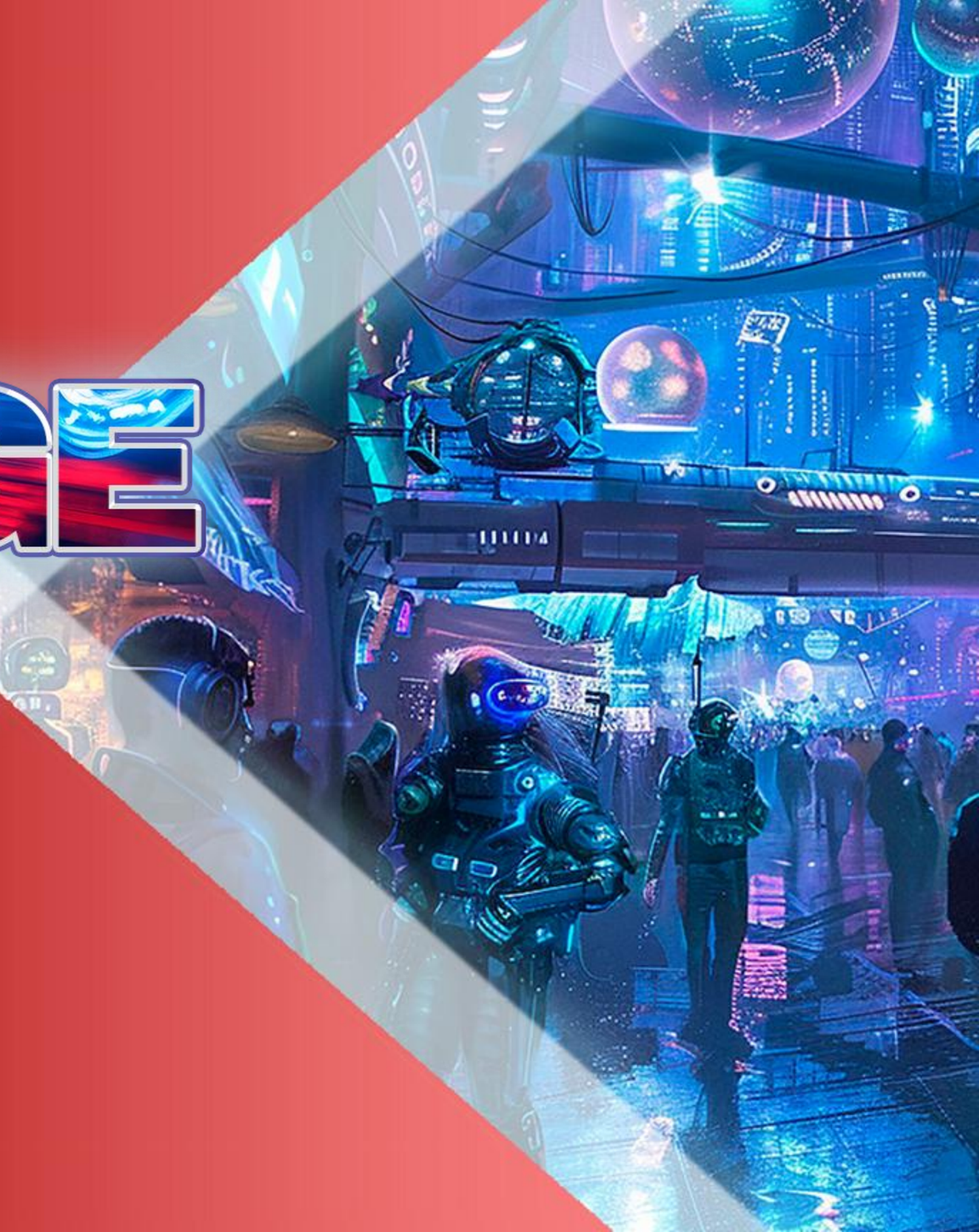
CODERAGE 2025



December 1-5 / 8-10

10 am -4 pm (CST)

embarcadero®

Working with Job Objects

Steffen Nyeland
Embarcadero MVP

FixedByCode

Friday, December 5 – 10:00 CST (17:00 CET)

What is a Job Object

It is a Windows kernel object that manages a set of one or more processes

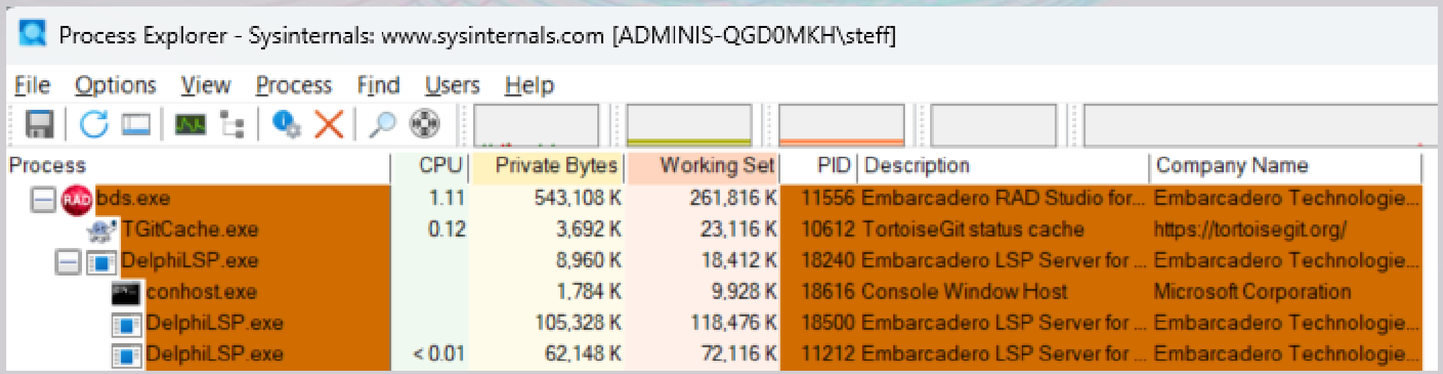
Exists since XP SP3, but only since Windows 8/Windows Server 2012 can jobs be nested.

For Delphi it is wrapped within Winapi.Windows unit. Prior to Delphi 12.x the CreateProcess flag CREATE_BREAKAWAY_FROM_JOB was missing, so you had to define it yourself like:

```
const
  CREATE_BREAKAWAY_FROM_JOB          = $01000000;
  {$EXTERNALSYM CREATE_BREAKAWAY_FROM_JOB}
```

How to find Job Objects

- Download Process Explorer (v17.08 at this point in time) from Microsofts Sysinternals site: https://learn.microsoft.com/en-us/sysinternals/downloads/process-explorer
- By default, Jobs are not shown/highlighted within the process tree – so enable that under Options|Color Selection|Jobs



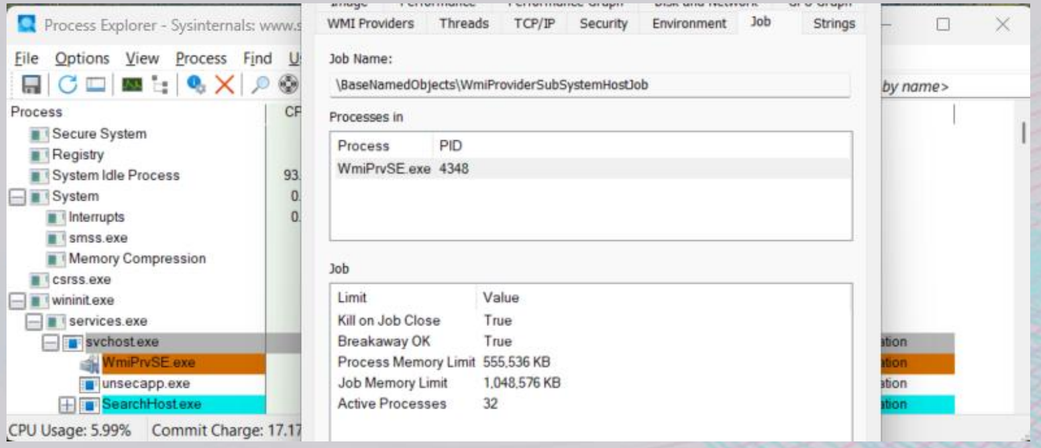
Why are Job Objects useful

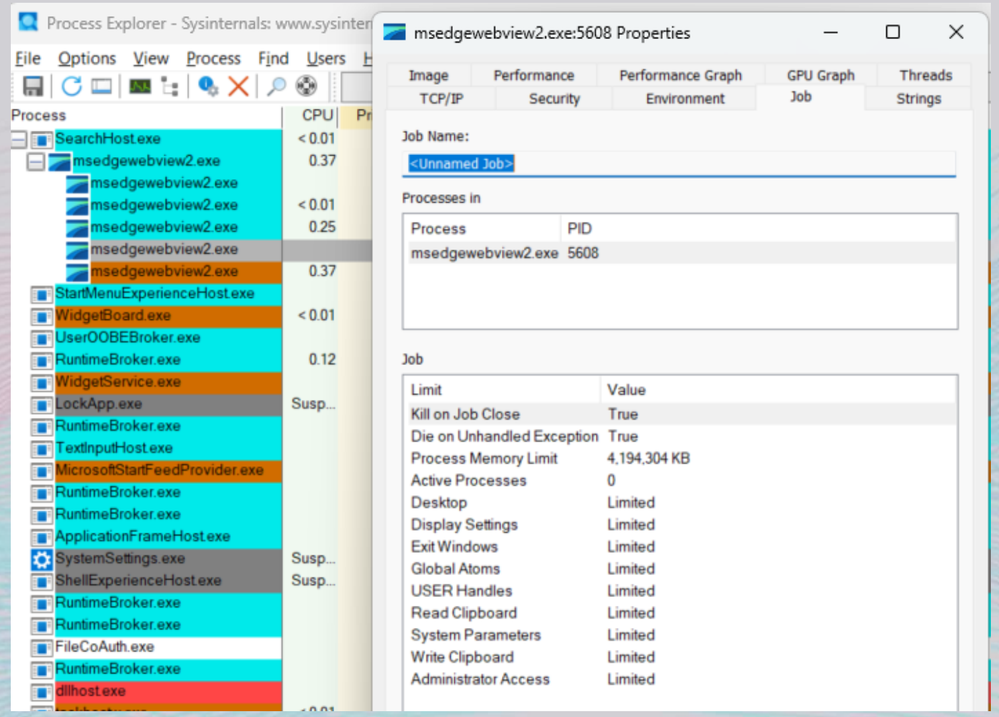
- Allow a group of processes to be managed as one.
- Starting and terminating related external processes.
- They can be named.
- If main process dies, Windows will terminate the child processes.
- Can be nested.
- Accounting info – I/O, CPU usage
- Increasingly used – since more application consists of many processes
- Can associate an I/O completion port (optional)

Limits and Notifications on Job Objects

- Limits can be set as
  - CPU: Maximum processes active, time, Affinity, rate control
  - Memory: minimum and maximum working set, commit maximum
  - Network: Maximum bandwidth
  - I/O: Maximum rate, read and write bytes
  - UI: User and GDI handles, clipboard access, exiting windows, desktop switching/creation
- If a process assigned to the job tries to change and exceed any of the limits, the function call succeeds but is ignored.
- If an optional I/O completion port is associated to the job; it can get notifications when limits are exceeded.

Examples of jobs with limits





- Fill one of the following record structures and call the SetInformationJobObject function with the job object and type of structure.

TJobObjectBasicLimitInformation (JOBOBJECT_BASIC_LIMIT_INFORMATION),
TJobObjectBasicUIRestrictions(JOBOBJECT_BASIC_UI_RESTRICTIONS),
TJobObjectCPURateControlInformation(JOBOBJECT_CPU_RATE_CONTROL_INFORMATION),
TJobObjectExtendedLimitInformation(JOBOBJECT_EXTENDED_LIMIT_INFORMATION),
TJobObjectNotificationLimitInformation(JOBOBJECT_NOTIFICATION_LIMIT_INFORMATION)

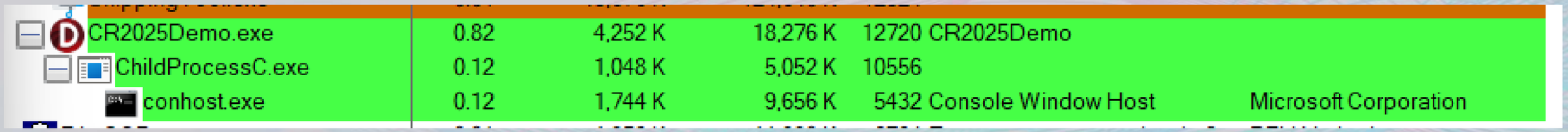
Code example 1

- We will use a custom thread class to create process and pipe its STDERR back via a callback
- The thread will execute a small C++ program that just writes 10 times to STDERR and the terminates

```
int _tmain(int argc, _TCHAR* argv[])
{
    for (int i = 0; i < 10; i++) {
        std::cerr << "Another mystic error from the C++ child process..." << std::endl;
        Sleep(1500);
    }
}
```

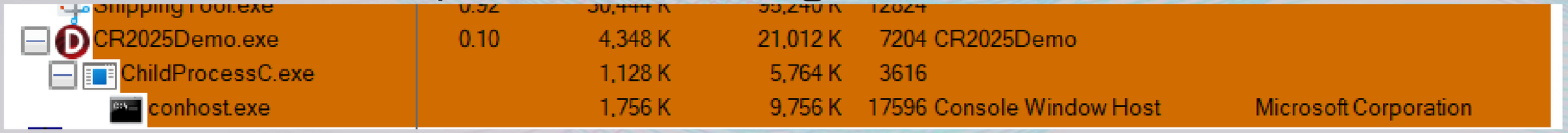
Code example 1

- Viewed in Process Explorer on startup



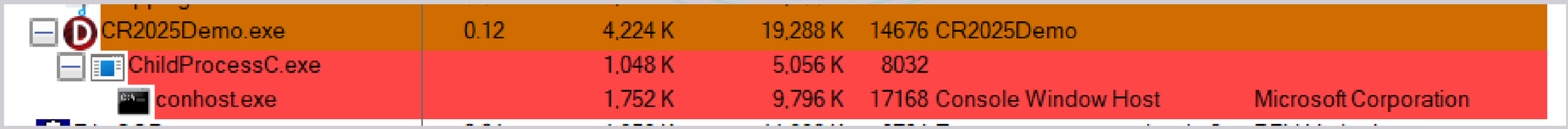
CR2025Demo.exe	0.82	4,252 K	18,276 K	12720	CR2025Demo	
ChildProcessC.exe	0.12	1,048 K	5,052 K	10556		
conhost.exe	0.12	1,744 K	9,656 K	5432	Console Window Host	Microsoft Corporation

- Viewed in Process Explorer while running



CR2025Demo.exe	0.10	4,348 K	21,012 K	7204	CR2025Demo	
ChildProcessC.exe		1,128 K	5,764 K	3616		
conhost.exe		1,756 K	9,756 K	17596	Console Window Host	Microsoft Corporation

- Viewed in Process Explorer when child process terminates



CR2025Demo.exe	0.12	4,224 K	19,288 K	14676	CR2025Demo	
ChildProcessC.exe		1,048 K	5,056 K	8032		
conhost.exe		1,752 K	9,796 K	17168	Console Window Host	Microsoft Corporation

Code example 2

- A Delphi port of the example from Pavel Yosifovich’s Job Objects YouTube video
- Limit the Job Object to only have a max of 2 active processes
- Limit the Job Object to maximum utilize 10% of the CPU.
- As in the original example, we will use CPU Stress 3.2 (or later) to illustrate the limits.
- We will with the example code attach an open instance of CPU Stress and put that under the control of the job objects limits

..where does this lead to.

- Well everything seems connected – so Windows 10 introduced Silos – Windows Containers
- Silos are upgraded Job Objects – filesystem, registry and name space.
- Docker can be a Windows Container/Silo `--isolation=process`
- Limits would normally be set by container runtime like: `docker --memory --cpus`
- On Linux cgroups would equivalent to Job Objects

Links

- Job Objects documentation: https://learn.microsoft.com/en-us/windows/win32/procthread/job-objects
- Process Explorer: https://learn.microsoft.com/en-us/sysinternals/downloads/process-explorer
- Blog post: https://fixedbycode.blogspot.com/2025/06/job-objects.html
- Pavel Yosifovich on Job Objects: https://www.youtube.com/watch?v=2QRkNCrBrjI
- CPU Stress 3.0: https://github.com/zodiacon/CPUStress

Thank You!