

Base Class Library

A CLR always ships with a set of assemblies called a Base Class Library (BCL). A BCL provides core functionality to programmers, such as collections, input/output, text processing, XML/JSON handling, networking, encryption, interop, concurrency, and parallel programming. A BCL also implements types that the C# language itself requires (for features such as enumeration, querying, and asynchrony) and lets you explicitly access features of the CLR, such as Reflection and memory management.

ترجمه پاراگراف

کتابخانه‌ی کلاس پایه (Base Class Library - BCL) یک CLR همیشه همراه با مجموعه‌ای از اسمبلی‌ها ارائه می‌شود که کتابخانه‌ی کلاس پایه (BCL) نام دارد.

BCL عملکردهای اصلی را برای برنامه‌نویسان فراهم می‌کند، از جمله:

- کالکشن‌ها (collections)
- ورودی/خروجی (input/output)
- پردازش متن
- کار با XML/JSON
- شبکه (networking)
- رمزنگاری (encryption)
- تعامل با کدهای دیگر (interop)
- برنامه‌نویسی همزمان (concurrency) و موازی (parallel programming)

BCL همچنین انواعی (types) را پیاده‌سازی می‌کند که خود زبان C# به آن‌ها نیاز دارد (برای قابلیت‌هایی مثل enumeration، query و asynchrony) و به شما امکان می‌دهد به‌طور صریح به ویژگی‌های CLR مثل Reflection و مدیریت حافظه دسترسی داشته باشید.

❖ توضیحات تکمیلی و نکات ارائه

1. BCL = هسته‌ی اصلی .NET.
 - یک جعبه‌ابزار بزرگ از قابلیت‌های پایه.
 - تقریباً هر پروژه‌ی C# به‌طور مستقیم یا غیرمستقیم از BCL استفاده می‌کند.
2. عملکردهای کلیدی BCL:
 - Collections: مثل List<T>, Dictionary<K,V>
 - IO: خواندن/نوشتن فایل، کار با استریم‌ها
 - Text processing: مثل string.Replace, Regex
 - XML/JSON: کلاس‌های XmlDocument, System.Text.Json
 - Networking: HttpClient, Socket
 - Encryption: SHA256, AES
 - Interop: فراخوانی DLL ها یا COM
 - Concurrency/Parallelism: Task, Thread, Parallel.For
3. نیازهای داخلی زبان: C#

- خیلی از قابلیت‌های زبان روی BCL سوار هستند:
 - Enumeration: IEnumerable<T>
 - LINQ querying
 - Async/Await: Task, Task<T>
 - 4. دسترسی به CLR از طریق: BCL
 - منادیتا و Reflection بررسی ساختار کد در زمان اجرا.
 - Memory Management دسترسی به GC مثل GC.Collect()).
-

جمع‌بندی ارائه برای این بخش:
"هر CLR همراه خودش یک Base Class Library میاره که ابزار اصلی برنامه‌نویس‌هاست. همه‌چی از کار با فایل و شبکه گرفته تا JSON، امنیت و پردازش موازی داخلش هست. حتی قابلیت‌های زبانی C# مثل async و enumeration هم روی BCL ساخته شدن. در واقع BCL قلب NET. ه که هم نیازهای برنامه‌نویس رو پوشش میده و هم به CLR وصل میشه".

Runtimes

A runtime (also called a framework) is a deployable unit that you download and install. A runtime consists of a CLR (with its BCL), plus an optional application layer specific to the kind of application that you're writing—web, mobile, rich client, etc. (If you're writing a command-line console application or a non-UI library, you don't need an application layer.) When writing an application, you target a particular runtime, which means that your application uses and depends on the functionality that the runtime provides. Your choice of runtime also determines which platforms your application will support.

ترجمه پاراگراف

Runtime‌ها

یک Runtime (که به آن Framework هم گفته می‌شود) یک واحد قابل نصب است که شما آن را دانلود و نصب می‌کنید.

یک Runtime شامل یک CLR همراه با BCL آن) و یک لایه اختیاری اپلیکیشن است که مخصوص نوع برنامه‌ای است که می‌نویسید — مثل وب، موبایل یا rich client. اگر برنامه‌ی شما یک اپلیکیشن خط فرمان یا یک کتابخانه‌ی بدون رابط کاربری باشد، به لایه‌ی اپلیکیشن نیازی ندارید).

وقتی یک اپلیکیشن می‌نویسید، آن را برای یک Runtime مشخص هدف‌گذاری (target) می‌کنید. این یعنی برنامه‌ی شما از قابلیت‌هایی استفاده می‌کند که آن Runtime ارائه می‌دهد و به آن‌ها وابسته است.

انتخاب Runtime همچنین تعیین می‌کند که برنامه‌ی شما از کدام پلتفرم‌ها پشتیبانی خواهد کرد.

توضیحات تکمیلی و نکات ارائه

1. Runtime = چارچوب اجرایی:
 - چیزی که نصب می‌کنیم تا برنامه‌ها مون روی اون اجرا بشن.
 - مثال. NET 8 Runtime, .NET Core, .NET Framework :
2. اجزای Runtime:
 - CLR + BCL هسته‌ی اصلی
 - Application Layer (اختیاری):
 - ASP.NET Core برای وب
 - MAUI/Xamarin برای موبایل
 - WPF/WinForms برای rich client دسکتاپ
 - بدون لایه → برای console apps یا کتابخانه‌ها
3. Targeting a Runtime:
 - وقتی اپ می‌نویسی، باید تعیین کنی کد روی کدوم Runtime اجرا بشه.
 - مثلاً:
 - اگر Target رو NET 8 بزنی → برنامه روی همه‌ی پلتفرم‌ها (cross-platform) می‌تونه اجرا بشه.
 - اگر Target رو NET Framework 4.8 بزنی → فقط روی ویندوز کار می‌کنه.
4. Platform Support وابسته به Runtime:
 - انتخاب Runtime = تعیین پلتفرم.
 - مثلاً:
 - ASP.NET Core روی ویندوز، لینوکس، مک کار می‌کنه.
 - UWP فقط روی ویندوز ۱۰. devices (Xbox, Surface Hub, HoloLens)

جمع‌بندی ارائه برای این بخش:

Runtime مثل چارچوب اجراییه که باید نصب بشه تا برنامه کار کنه. داخلش CLR و BCL هست و بسته به نوع اپ، ممکنه لایه‌های اضافی مثل ASP.NET یا MAUI داشته باشه. وقتی اپ می‌نویسی باید مشخص کنی Target Runtime چی باشه، چون این انتخاب تعیین می‌کنه برنامه روی چه پلتفرم‌هایی قابل اجراست."

The following table lists the major runtime options:

Application layer	CLR/BCL	Program type	Runs on...
ASP.NET	.NET 8	Web	Windows, Linux, macOS
Windows Desktop	.NET 8	Windows	Windows 10+
WinUI 3	.NET 8	Windows	Windows 10+
MAUI	.NET 8	Mobile, desktop	iOS, Android, macOS, Windows 10+
.NET Framework	.NET Framework	Web, Windows	Windows 7+

Figure 1-2 shows this information graphically and also serves as a guide to what's covered in the book.