

# Managed Extensibility Framework

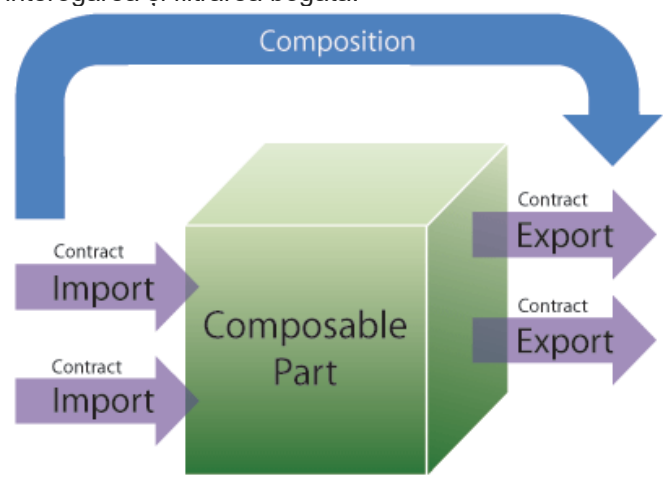
Managed Extensibility Framework (MEF) este o componentă a .NET Framework 4.0 care vizează crearea de aplicații ușor și extensibile. Scopul său este de a permite dezvoltatorilor de aplicații .NET să descopere și să utilizeze extensii fără configurare necesară. De asemenea, intenționează să permită dezvoltatorilor de extensii să încapsuleze codul cu ușurință și să evite dependențele dure fragile. În plus, urmărește să permită reutilizarea extensiilor în cadrul aplicațiilor. MEF a fost introdus ca parte a .NET 4.0 și Silverlight 4.

MEF urmărește să rezolve problema extensibilității în timpul rulării. Fără MEF, orice aplicație care dorește să suporte un model de plugin trebuie să își creeze propria infrastructură de la zero. Aceste pluginuri vor fi adesea specifice aplicațiilor și nu pot fi refolosite în mai multe implementări.

- MEF își propune să ofere un mod standard aplicației gazdă de a se expune și de a consuma extensii externe.

Extensiile, prin natura lor, ar putea fi refolosite între diferite aplicații. Cu toate acestea, o extensie poate fi încă implementată într-un mod care este specific aplicației. Extensiile în sine pot depinde una de cealaltă și MEF își propune să se asigure că sunt conectate împreună în ordinea corectă, economisind dezvoltatorul de a face manual.

- MEF oferă un set de abordări de descoperire pentru aplicație pentru localizarea și încărcarea extensiilor disponibile.
- MEF permite extensiilor de etichetare cu metadate suplimentare, care are scopul de a facilita interogarea și filtrarea bogată.



În loc de înregistrare explicită a componentelor disponibile, MEF oferă o modalitate de a le descoperi implicit, prin compoziție. O componentă MEF, numită parte, specifică declarativ atât dependențele sale (cunoscute sub numele de importuri), cât și capacitățile (cunoscute sub numele de exporturi) pe care le pune la dispoziție. Când se creează o parte, motorul compoziției MEF satisface importurile sale cu ceea ce este disponibil din alte părți.

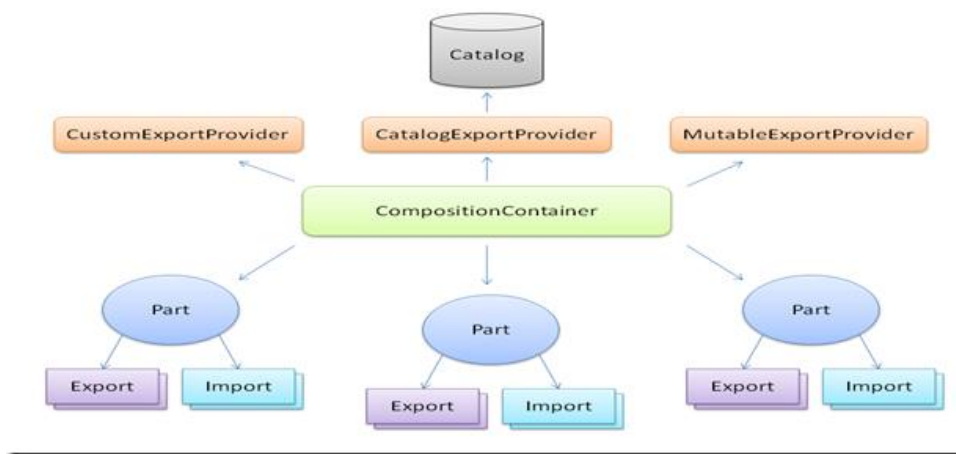
Această abordare rezolvă problemele discutate în secțiunea anterioară. Deoarece componentele MEF specifică în mod declarativ capacitățile lor, ele pot fi descoperite în timpul execuției, ceea ce înseamnă că o aplicație poate folosi componentele fără referințe codate greu sau fișiere de configurare fragile. MEF permite aplicațiilor să descopere și să examineze părți prin metadatele lor, fără a

le instanțiza sau chiar încărca ansamblurile. Ca urmare, nu este nevoie să specificați cu atenție momentul și modul în care trebuie încărcate extensiile.

Pe lângă exporturile furnizate, o parte poate specifica importurile sale, care vor fi completate de alte părți. Acest lucru face comunicarea între părți nu numai posibilă, dar ușor, și permite factoring bun de cod. De exemplu, serviciile comune mai multor componente pot fi incluse într-o parte separată și ușor modificate sau înlocuite.

Deoarece modelul MEF nu necesită dependență puternică de un anumit ansamblu de aplicații, permite extensiile să fie refolosite din aplicație în aplicație. Acest lucru facilitează, de asemenea, dezvoltarea unui sistem de testare, independent de aplicație, pentru a testa componentele extensiei.

O aplicație extensibilă scrisă utilizând MEF declară un import care poate fi completat de componentele extensiei și poate declara exporturi pentru a expune serviciile de aplicații la extensii. Fiecare componentă extensie declară un export și poate declara și importuri. În acest fel, componentele extensiei sunt în mod automat extensibile.



Versiunile anterioare ale .NET Framework au introdus Managed Add-in Framework(MAF), conceput pentru a permite aplicațiilor să izoleze și să gestioneze extensiile. Focusul MAF este ușor mai ridicat decât MEF, concentrându-se pe izolarea extensiei și încărcarea și descărcarea ansamblului, în timp ce MEF se concentrează pe descoperire, extensibilitate și portabilitate. Cele două cadre interoperează fără probleme, iar o singură aplicație poate profita de ambele.

Bibliografie:

<https://docs.microsoft.com/en-us/dotnet/framework/mef/>

[https://en.wikipedia.org/wiki/Managed\\_Extensibility\\_Framework](https://en.wikipedia.org/wiki/Managed_Extensibility_Framework)

<http://buxbaum.us/2011/08/20/gentle-introduction-to-mefpart-one/>