Dynamics 365 Plugin Development Cheatsheet

What is a Plugin?

A plugin is a custom piece of business logic written in .NET (C# or VB.NET) that is executed within the Dynamics 365 event execution pipeline. It allows developers to extend the platform by reacting to specific events such as create, update, delete, or retrieve.

Common Use Cases

- Validate data before saving (e.g., ensure Opportunity value is greater than 0).
- Automate processes such as setting default values or updating related records.
- Enforce complex business rules that cannot be achieved with workflows or Power Automate.
- Integrate with external systems by making service calls.

Steps to Create a Plugin

1. Create a new Class Library project in Visual Studio.
2. Add references to Microsoft.CrmSdk assemblies (via NuGet).
3. Implement the IPlugin interface.
4. Write your business logic inside the Execute method.
5. Build the DLL.

Example (Simplified)

A plugin that validates the estimated revenue of an Opportunity before saving:

```
public void Execute(IServiceProvider serviceProvider) {
// Obtain context
IPluginExecutionContext context =
(IPluginExecutionContext)serviceProvider.GetService(typeof(IPluginExecutionContext));
Entity entity = (Entity)context.InputParameters["Target"];
if (entity.Contains("estimatedvalue") && ((Money)entity["estimatedvalue"]).Value <= 0)
throw new InvalidPluginExecutionException("Estimated revenue must be greater than 0.");
}
```

Registering the Plugin

1. Open Plugin Registration Tool (part of XrmTooling).
2. Connect to your Dynamics 365 environment.
3. Register a new Assembly (upload the DLL).
4. Register a new Step:
   - Message: Create, Update, Delete, etc.
   - Primary Entity: e.g., Opportunity.
   - Pipeline Stage: Pre-Validation, Pre-Operation, or Post-Operation.
5. Test the plugin in Dynamics 365.

Best Practices

- Keep plugins lightweight to avoid performance issues.
- Use Pre-Validation for validations and Post-Operation for actions after save.
- Always handle exceptions with clear messages.
- Use tracing service (ITracingService) for debugging.
- Avoid hardcoding values, use configuration where possible.