**Adapted Developer Workstation Setup Instructions for Brocc**

*Disclaimer: This version has been adapted for Brocc's use. Please note that certain internal references and modules have been removed. As a result, some sections may require additional adaptation to suit your specific environment.*

**Current Objective**

The aim of this document is to provide a streamlined process for setting up a local development environment to build and debug the solution. This adapted version is tailored for Brocc; however, further adjustments may be necessary to fully meet your requirements.

**Prerequisites**

* Visual Studio 2019 or later
* SQL Server Developer Edition
* Internet Information Services (IIS)
* .NET Framework (Full Framework)
* .NET Core along with the Hosting Module
* Source code repositories for:
  + Core modules
  + Embedded client sites (if applicable)
  + Additional services (as needed)
* A folder containing client-specific configuration

**Overview**

Please adhere to the specified folder structure as it greatly facilitates sharing configuration files. Consistency in directory paths is essential.

**Setup Process**

**Clone the Repository**  
Bind Trunk to C:\Projects\Näktergal\Trunk.

**Client Configuration Files**  
Obtain the configuration files and adapt them.

**Module Pre-Build Steps**

Each of the modules has a pre build step that looks like this:

if $(ConfigurationName) == Debug (

xcopy /Q /Y "C:\Naktergal\MagellanCurrentConfiguration\$(ProjectName).appsettings.debug.config" "$(ProjectDir)"

xcopy /Q /Y "C:\Naktergal\MagellanCurrentConfiguration\$(ProjectName).connectionstrings.debug.config" "$(ProjectDir)"

)

So rebuilding the solution after running the script will switch the appsettings and connection strings.

This works because web.config has this section by default:

<appSettings configSource="ncredit.appsettings.debug.config" />

<connectionStrings configSource="ncredit.connectionstrings.debug.config" />

These debug-files are not checked in. This is to prevent checked in appsettings or connectionstring to make it into production by accident.

There is then a web.config-transform that changes this for release builds causing production to require proper appsettings/connectionstrings which are copied in during deployment.

<configuration xmlns:xdt="http://schemas.microsoft.com/XML-Document-Transform">

<appSettings configSource="ncredit.appsettings.config" xdt:Transform="Replace" />

<connectionStrings configSource="ncredit.connectionstrings.config" xdt:Transform="Replace" />

<system.web>

<compilation xdt:Transform="RemoveAttributes(debug)" />

</system.web>

</configuration>

After this is done you should be able to open the everything solution and build it.

Note, open visual studio in admin mode or you won’t be able to attach a debugger to IIS.

**Open and Build the Solution**Open the complete solution in Visual Studio.  
It is advisable to run Visual Studio with administrative privileges to allow attaching the debugger to IIS.  
Build the solution. The pre-build steps should automatically switch the configuration files to the appropriate debug versions.

**Installation of IIS and Required Features**

Execute the following PowerShell commands with administrative privileges:

powershell

Kopiera

Set-ExecutionPolicy Bypass -Scope Process

Enable-WindowsOptionalFeature -Online -FeatureName IIS-WebServerRole

Enable-WindowsOptionalFeature -Online -FeatureName IIS-WebServer

Enable-WindowsOptionalFeature -Online -FeatureName IIS-CommonHttpFeatures

Enable-WindowsOptionalFeature -Online -FeatureName IIS-HttpErrors

Enable-WindowsOptionalFeature -Online -FeatureName IIS-HttpRedirect

Enable-WindowsOptionalFeature -Online -FeatureName IIS-ApplicationDevelopment

Enable-WindowsOptionalFeature -Online -FeatureName IIS-NetFxExtensibility45

Enable-WindowsOptionalFeature -Online -FeatureName IIS-HealthAndDiagnostics

Enable-WindowsOptionalFeature -Online -FeatureName IIS-HttpLogging

Enable-WindowsOptionalFeature -Online -FeatureName IIS-LoggingLibraries

Enable-WindowsOptionalFeature -Online -FeatureName IIS-RequestMonitor

Enable-WindowsOptionalFeature -Online -FeatureName IIS-HttpTracing

Enable-WindowsOptionalFeature -Online -FeatureName IIS-Security

Enable-WindowsOptionalFeature -Online -FeatureName IIS-RequestFiltering

Enable-WindowsOptionalFeature -Online -FeatureName IIS-Performance

Enable-WindowsOptionalFeature -Online -FeatureName IIS-WebServerManagementTools

Enable-WindowsOptionalFeature -Online -FeatureName IIS-IIS6ManagementCompatibility

Enable-WindowsOptionalFeature -Online -FeatureName IIS-Metabase

Enable-WindowsOptionalFeature -Online -FeatureName IIS-ManagementConsole

Enable-WindowsOptionalFeature -Online -FeatureName IIS-BasicAuthentication

Enable-WindowsOptionalFeature -Online -FeatureName IIS-WindowsAuthentication

Enable-WindowsOptionalFeature -Online -FeatureName IIS-StaticContent

Enable-WindowsOptionalFeature -Online -FeatureName IIS-DefaultDocument

Enable-WindowsOptionalFeature -Online -FeatureName IIS-WebSockets

Enable-WindowsOptionalFeature -Online -FeatureName IIS-ApplicationInit

Enable-WindowsOptionalFeature -Online -FeatureName IIS-ASPNET45 -All

Enable-WindowsOptionalFeature -Online -FeatureName IIS-ISAPIExtensions

Enable-WindowsOptionalFeature -Online -FeatureName IIS-ISAPIFilter

Enable-WindowsOptionalFeature -Online -FeatureName IIS-HttpCompressionStatic

Ensure that the source code for all modules is set up before configuring IIS.

**IIS Module Configuration**  
Configure each module as an IIS site, binding them to unique localhost ports.

The site content should reside directly within the source folder, ensuring that the sites are continuously running.

All sites are configured to run within the same application pool. This facilitates debugging across multiple services.

Note that due to IIS caching mechanisms, some changes may not be immediately reflected. In such cases, an IIS reset (executed twice) may be necessary.

**.NET Core Hosting for Specific Modules**

Install the .NET Core Hosting Bundle as required.

For .NET Core sites, please note that local debugging may require deploying the site rather than running directly from the source.

When debugging, stop the IIS site and launch the service from Visual Studio (using F5) on the corresponding port.

**Embedded Client Sites**

In cases where client-specific public interfaces are developed as single-page applications (SPA) embedded within a module, clone the client repository (e.g., to C:\Projects\Client) and install the necessary npm packages using:

npm install

For debugging, run .\serve-dev-embedded.ps1

Create symbolic links to integrate the SPA into the module’s folder structure. For example:

New-Junction -LiteralPath "C:\Projects\Näktergal\Trunk\Naktergal\nCustomerPages\a" -TargetPath "C:\Projects\Git\bluestep-fi-customerpages\dist\bluestep-fi-customerpages"

Without using New-Junction:

New-Item -Path "C:\Projects\Naktergal\Naktergal\nCustomerPages\a" -ItemType SymbolicLink -Value "C:\Projects\client-bluestep\CustomerPages\dist\bluestep-fi-customerpages" -Force

(you need to update above url to where you have both your nCustomerPages-folder and where you have cloned ex. client-bluestep)

More modern example:

New-Item -Path "C:\Projects\core-naktergal\CoreModules\nCustomerPages\n" -ItemType SymbolicLink -Value C:\Projects\core-naktergal\SelfContainedModules\ntech-backoffice-ui\dist\customerpages" -Force

**Database Configuration**  
Set up databases for each module and client as required.

Ensure that the database names match those specified in the connection strings within the configuration files.

It is advisable to implement backup and restore procedures for each client’s databases. Additional scheduled tasks (SQL Agent jobs) for backing up and restoring databases are recommended for test scenarios.

**Additional Considerations**

**Special Characters in Project Names:**  
Avoid using special characters in project names as they may cause issues with certain tools.

**Local Environment Constraints:**  
The local IIS setup and database configuration must remain on the same machine or within the same network segment to prevent performance issues.

**Administrative Privileges:**  
Running Visual Studio and IIS configuration tasks with administrative rights is required to ensure proper debugging and deployment.