

Version bêta - 12/2015





### Table des matières

Objectif	
Présentation du contexte	2
Etape 1 : Création du dossier API et génération des interfaces.	3
Etape 2 : Ouverture du projet d'extension	9
Etape 3 : Ajouter l'extension à un dossier de l'application EBP Gestion Commerciale Open Line	

## Objectif

L'objectif de ce tutoriel est d'expliquer comment personnaliser l'application EBP Gestion Commerciale Open Line en fonction d'une demande client.

Pour cela, on utilisera les tables personnalisées, une fonctionnalité déjà présente dans l'application.

Après le paramétrage de ces tables, on générera des DLL qui pourront être intégrées dans Visual Studio, pour permettre de développer les fonctionnalités souhaitées.

Pour ce faire, nous allons utiliser un exemple concret et suivre les différentes étapes nécessaires à la génération de l'extension.



### Présentation du contexte

L'objectif est d'ajouter des fonctionnalités supplémentaires autour des contacts de vente. Pour cela, nous allons ajouter 3 tables :

- « Fonction Contact » qui permet de définir le poste du contact (Cadre, Employé, Agent de Maitrise, ...),
- « Sous-fonction Contact » qui permet de définir la fonction du contact (Commercial, Développeur, ....),
- « Service Contact » qui correspond au service auquel le contact est rattaché (Développement, Commercial, Marketing, ...)



## Etape 1 : Création du dossier API et génération des interfaces.

- 1. Lancer l'application EBP Gestion Commerciale Open Line, sans ouvrir de dossier.
- 2. Afficher le menu développeur : dans *Outils / Options / Apparence*, cocher la case **Afficher le menu développeur**.

Options		_ 🗆 X
<rechercher></rechercher>	Vous pouvez modifier l'organisation de l'interface utilisateur.	
Chemins           Apparence           Import/Export paramétrable           Services Internet           Proxy HTTP           Proxy FTP           Sauvegarde en ligne	Apparence         Afficher la barre d'état         Afficher la barre d'outils         Afficher la barre de fenêtres         Ouvrir les fenêtres en plein écran         Afficher le bouton calculatrice sur les éditeurs numériques         Activer le correcteur orthographique         Afficher le texte enrichi en utilisant GDI         Imprimer le texte enrichi en utilisant GDI         Imprimer le texte enrichi en mode WYSIWYG         Convertir les formats de texte (RTF, HTML, Texte) en utilisant les composants Microsoft         Afficher la barre de menu dans les favoris         Acccélérer le chargement des fiches (utilise plus de mémoire)         Afficher le bouton de déclenchement de la recherche dans les listes         Ne pas sélectionner automatiquement la 1ère ligne des listes         Nombre de champs personnalisés par onglet       30         Revenir au thème Windows XP         Afficher le nom du dossier dans le titre des fenêtres       Ne pas l'afficher         Afficher le nom du dossier dans le titre des grilles en bande (Stocks article, Fournisseurs d'un article, Adresses, Coordonne         Afficher le menu développeur	ées bancair
Aide	ОК	Annuler

3. Passer en mode développeur : dans *Développeur*, sélectionner **Passer en mode développeur**.





4. Créer un dossier nommé *API* avec comme « Préfixe pour les tables/colonnes » *MYEXT*. Il est important de donner un préfixe non générique afin d'éviter d'éventuels conflits avec d'autres extensions (Exemple : les 3 premières lettres pouvant correspondre à votre société).

🛎 Nouveau				_ 🗆 ×
Informations relatives à la base de données				
Renseignez sur cette page les informations de base de données.				5
Nom du dossier				1
Préfixe pour les tables/colonnes MYEXT				
Afficher les options de connexion avancées				
Aide	< <u>P</u> récédent	<u>S</u> uivant >	Lancer	<u>A</u> nnuler



5. Aller dans le menu Paramètres / Personnalisation des champs/tables





6. Pour ajouter une table personnalisée nommée *ContactFunction*, cliquer sur le bouton *Ajouter une table*.

📑 Pe	rsonnalisation des champs/ta	ables						- 🗆 >
🛖 Aj	outer une table 🗙 Supprimer une tal	ble 🔚 Enre	gistrer 🐁 A	nnuler				
	Table	Alertes	Champs obligatoires	Champs personnalisés		Alertes Champs obligatoires Ch	amps personnalisés	
- m	Activités				^	🧲 🛉 Ajouter 🖉 Modifier	🗙 🛅 🔤 Rechercher >	🛛 📄 🕶 营 👻 📄 Déplier tout 📄 Plier tout
	Adresse				l l P	Code 🔺 Libellé		
	Affaire				t i F			
400	Article							
	Bien							
	Service							
	Fabrication				-			
	Commerciale/Composé							
	Article en stock							
	Article lié			Aioutor une	tabl	le personnalisée X		
	Banque			Ajouter une	lavi			
	Barème			Nom	Conta	actFunction		
4-111	Catégorie tarifaire			Table maître	-	Sans table maître		
	Client		-				1	
	Fournisseur		Ty	pe de cie primaire	A	exte j *		
-	Civilités			Ok		Annuler	Inser ou Cliquez a	sur Ajouter pour créer un
	Client						Inser ou ciiquez s	sui Ajoutei pour creer un
1	Client						nouvel enregist	trement)
	Prospect							
	Clients attachés à la facturation péri							
	Code postal							
	Commercial/collaborateur							
	Contact							
	Contrat de maintenance	<b>v</b>						
	Coordonnées bancaires (tiers)							
	Coûts d'un contrat de maintenance							
	Coûts supplémentaires d'affaire							
	Coûts supplémentaires de modèle de							
	Coûts supplémentaires de suivis inci							
	DEB							
	Déraissement				- 1			

Avertissement : Les libellés des tables doivent être distincts. Il est donc fortement recommandé de les rendre suffisamment explicites pour éviter que deux tables d'extensions différentes possèdent le même libellé.

7. Renseigner les Propriétés générales de la table.

Alertes Champs obligatoires Champs pe	ersonnalisés									
💠 Ajouter - 🗶 Supprimer 🔶 Monter 🖶 Descendre										
A 📑 Fonction Contact	tact Propriétés générales									
🔒 Code [xy_sysId]	Nom interne	xyx_MYEXT_ContactFunction								
	Nom affiché	Fonction Contact								
	Table maître									
	Infos de création									
	Autoriser les champs calculés									
	Trier la table									
	Colonne à afficher dans les listes déroulantes	🔒 Code								
	Icône	8								
	Compteur auto-incrémenté									
	Compteur d'initialisation	FC0001								



8. Pour créer les champs que vous souhaitez renseigner dans votre table, cliquer sur le bouton *Ajouter* puis indiquer les différentes propriétés qui lui sont propres.

📫 Personnalisation des champs/ta	ables							_ = ×
💠 Ajouter une table 🗙 Supprimer une tal	ble 🔚 Enre	egistrer 🐁 A	nnuler					
Table	Alertes	Champs obligatoires	Champs personnalisés	,	Alertes Champs oblig	gatoires Champs pers	onnalisés	
Paramétrage des modèles par fourni				-	💠 Ajouter 🛛 🗙 Su	ıpprimer   Montei	🗣 Descendre	
Pays					A T Enction Cont	act	Propriétés générales	
					Code [xy	sysId]	Nom interne	xyx MYEXT ContactEunction
							Nom affiché	Fonction Contact
							Table maître	
	<b>V</b>						Infos de création	
Produits dient d'un incident	<b>V</b>						Autoriser les champs calculés	
							Trier la table	
							Colonne à afficher	O cada
Règlement de vente	<b>v</b>						dans les listes déroulantes	e Code
- Remise complémentaire			🕹 /	Ajoι	ter un champ pe	rsonnal – ×		
Remise en banque			-		<b>T</b> 1		Icône	<u> </u>
			Тур		Texte	*		
			Nor	n Lib	ellé		Compteur auto-incrémenté	
Sous-famille article					Ok	Annuler	Compteur d'initialisation	FC0001
							1	
Sous-famille fournisseur							-	
Client								
Fournisseur								
Taxe								
Texte standard								
-TVA								
Type de garantie								
Type d'évènement								
				-[]]				
Dossier				-[]]				
ContactFunction		·						

9. Procéder de la même manière pour les différentes tables nécessaires au projet.

U L'aide en ligne du logiciel contient de nombreuses informations. Elle vous aidera à déterminer les choix à faire lors de la création de vos tables personnalisées.



10. Pour générer l'interface, cliquer sur le menu *Développeur / Générer les interfaces*. Dans la fenêtre *Options de l'interface d'extension*, sélectionner le dossier de destination (de préférence un répertoire dédié) puis cliquer sur le bouton *Générer*.



11. Une fois la génération terminée, fermer le dossier. Dans le menu Développeur, *choisir Sortir du mode Développeur* puis quitter l'application.



## Etape 2 : Ouverture du projet d'extension

1. Dans **Visual Studio**, créer un projet de type *Class Library.NET Framework* version 4.0 ou supérieure nommé Extension Gestion.

			New Project		? ×
Recent		.NET Fr	amework 4 - Sort by: Default	- # E	Search Installed Templates (Ctrl+E)
▲ Installed		C <sup>#</sup>	Windows Forms Application	Visual C#	Type: Visual C#
▲ Templates ▲ Visual C# ▷ Store Apps			WPF Application	Visual C#	A project for creating a C# class library (.dll)
Windows D	esktop	<u>,</u> C:/	Console Application	Visual C#	
Office/Shar Cloud	ePoint		Class Library	Visual C#	
LightSwitch Reporting	1		Class Library (Portable)	Visual C#	
Silverlight Test		∰	WPF Browser Application	Visual C#	
WCF Workflow		<b>S</b>	Empty Project	Visual C#	
<ul> <li>Other Languag</li> <li>Other Project T</li> </ul>	es ypes	3	Windows Service	Visual C#	
Samples			WPF Custom Control Library	Visual C#	
		ů	WPF User Control Library	Visual C#	•
			Click here to go online and find templates.		
Name:	Extension Gestion	n			]
Location:	C:\Sources\			Browse	
Solution:	Create new solut	ion		*	
Solution name:	Extension Gestion	n			<ul> <li>Create directory for solution</li> <li>Add to source control</li> </ul>
					OK Cancel



2. Référencer les dll générées précédemment, disponibles dans le dossier de destination indiqué lors de la génération de l'interface.

Reference Manager - Extension Gestion								
Assemblies				Search Browse (Ctrl+E)	ρ-			
▷ Solution		Name	Path	Name:				
▶ COM	<b>V</b>	EBP.Api.Main.Interfaces	C:\Sources\Extension Gestion\ApiFiles\EBP.Api.Main.Interface	EBP.Api.Main.Interfaces.dll				
▲ Browse	<b>v</b>	EBP.Api.dll	C:\Sources\Extension Gestion\ApiFiles\EBP.Api.dll	created by.				
Recent	~	EBP.Api.Common.dll	C:\Sources\Extension Gestion\ApiFiles\EBP.Api.Common.dll	File Version: 1.0.0.0				
	4							
			Brows	e OK Can	cel			



3. Modifier le répertoire de sortie du projet en « C:\ProgramData\EBP\Invoicing8.0FR40\Extensions\Extension Gestion »

Extension Gestion* $+$ ×	
Application Build*	Configuration: Active (Debug) v Platform: Active (Any CPU) v
Build Events	General
Debug Resources Services Settings Reference Paths Signing	Conditional compilation symbols: Define DEBUG constant Define TRACE constant Platform target: Prefer 32-bit
Code Analysis	Allow unsafe code Optimize code
	Errors and warnings Warning level: 4 Suppress warnings:
	Treat warnings as errors   None  All  Specific warnings:
	Output       Output path:       ta\EBP\Invoicing8.0FR40\Extensions\Extension Gestion\]       Browse

- 4. Créer une classe pour l'extension
  - Créer une classe dérivant de la classe Extension : les propriétés et les méthodes évoquées ensuite sont indispensables.

```
using EBP.Api.Extension;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace Extension_Gestion
{
    public class ExtensionGestion : Extension
    {
    }
}
```



#### • Surcharger les propriétés abstraites :

#### ° ApiInterfaceLinkType

```
using EBP.Api.Extension;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace Extension Gestion
{
    public class ExtensionGestion : Extension
    {
        /// <summary>
        /// Gets the type of the API interface ApiInterfaceLink generated class.
        /// </summary>
        /// <remarks>Return the type EBP.Api.Interfaces.ApiInterfaceLink generated in the
interface dll EBP.Api.XXXXXX.Interfaces.dll</remarks>
        protected override Type ApiInterfaceLinkType
        {
            get { return typeof(ApiInterfaceLink); }
        }
    }
}
```

#### ° Description : description de l'extension

```
using EBP.Api.Extension;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace Extension_Gestion
{
    public class ExtensionGestion : Extension
    {
        /// <summary>
        /// Gets the extension description.
        /// </summary>
        /// <value>The extension description</value>
        protected override string Description
        {
            get { return "Extension pour la Gestion Commerciale"; }
        }
    }
}
```



 ExtensionId : Identifiant de l'extension (cette valeur ne doit pas être modifiée une fois votre extension utilisée !)

```
using EBP.Api.Extension;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace Extension_Gestion
{
    public class ExtensionGestion : Extension
    {
        /// <summary>
        ///\ensuremath{\mathsf{Gets}} the extension Id. Must be unique for each extension.
        /// </summary>
        /// <value> The extension Id </value>
        /// <remarks> Do not change this value otherwise it may cause unpredictable behavior
in the application. </remarks>
        protected override Guid ExtensionId
        {
            get { return new Guid("43FF83AB-9F86-495E-9D13-356181639333"); }
        }
    }
}
```

° Name : Nom de l'extension

```
using EBP.Api.Extension;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace Extension_Gestion
{
    public class ExtensionGestion : Extension
    {
        /// <summary>
        /// Gets the extension name.
        /// </summary>
        /// <value>The extension name</value>
        protected override string Name
        {
            get { return "Extension Gestion"; }
        }
   }
}
```



- Surcharger les méthodes
  - ConfirmUserDefinedSchemaObjectDelete : cette méthode doit être surchargée mais ne sera pas détaillée ici.

```
using EBP.Api.Extension;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace Extension Gestion
{
    public class ExtensionGestion : Extension
    {
        /// <summary>
        /// Gets whether a schema object can be safely deleted
        /// </summary>
        /// <param name="tableName">Table name</param>
        /// <param name="columnName">Column name. Null if it's a delete of a table</param>
        /// <returns>
        /// True for confirm this delete, otherwise false
        /// </returns>
       protected override bool ConfirmUserDefinedSchemaObjectDelete(string tableName,
       string columnName)
       {
            return true ;
       }
   }
```



° OnInitialized : permet d'initialiser toutes les entités et les fenêtres de l'extension

```
namespace Extension Gestion
{
    public class ExtensionGestion : Extension
    {
        /// <summary>
        /// Initializes the extension. Override this method to register all extensions
            entities, forms, etc.
        /// </summary>
        /// <param name="errors">Errors to fill in if initialization fails</param>
        /// <returns><c>true</c> if extension initialization succeed, otherwise
            <c>false</c></returns>
        protected override bool OnInitialized(IErrors errors)
        {
            if (!base.OnInitialized(errors))
                    return false;
            RegisterExtensionUserDefinedMenu(typeof(IContactFunctionEntryForm),
            "Fonction Contact",
            new Guid(MenuGuids.SaleCustomer), new Guid(NavBarGuids.SalesCustomer));
            RegisterExtensionUserDefinedMenu(typeof(IContactServiceEntryForm), "Service
           Contact",
            new Guid(MenuGuids.SaleCustomer), new Guid(NavBarGuids.SalesCustomer));
            RegisterExtensionUserDefinedMenu(typeof(ISubFonctionContactEntryForm),
            "Sous-Fonction Contact",
            new Guid(MenuGuids.SaleCustomer), new Guid(NavBarGuids.SalesCustomer));
            return true;
       }
  }
}
```



```
• Voici au final le code obtenu :
```

```
using EBP.Api.Extension;
using EBP.Api.Interfaces;
using EBP.Api.Interfaces.Invoicing.Module.Windows;
using EBP.Api.Interfaces.Misc;
using EBP.Api.Interfaces.UserDefinedForm;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace Extension_Gestion
{
    public class ExtensionGestion : Extension
    {
        /// <summary>
        /// Initializes the extension. Override this method to register all extensions
            entities, forms, etc.
        /// </summary>
        /// <param name="errors">Errors to fill in if initialization fails</param>
        /// <returns><c>true</c> if extension initialization succeed, otherwise
            <c>false</c></returns>
        protected override bool OnInitialized(IErrors errors)
        {
            if (!base.OnInitialized(errors))
                    return false:
            RegisterExtensionUserDefinedMenu(typeof(IContactFunctionEntryForm),
            "Fonction Contact",
            new Guid(MenuGuids.SaleCustomer), new Guid(NavBarGuids.SalesCustomer));
            RegisterExtensionUserDefinedMenu(typeof(IContactServiceEntryForm), "Service
            Contact",
            new Guid(MenuGuids.SaleCustomer), new Guid(NavBarGuids.SalesCustomer));
            RegisterExtensionUserDefinedMenu(typeof(ISubFonctionContactEntryForm),
            "Sous-Fonction Contact",
            new Guid(MenuGuids.SaleCustomer), new Guid(NavBarGuids.SalesCustomer));
            return true;
        }
        /// <summary>
        /// Gets whether a schema object can be safely deleted
        /// </summary>
        /// <param name="tableName">Table name</param>
        /// <param name="columnName">Column name. Null if it's a delete of a table</param>
        /// <returns>
        /// True for confirm this delete, otherwise false
        /// </returns>
        protected override bool ConfirmUserDefinedSchemaObjectDelete(string tableName,
string columnName)
        {
            return true;
        }
        /// <summary>
        /// Gets the extension name.
        /// </summary>
        /// <value>The extension name</value>
        protected override string Name
        {
            get { return "Extension Gestion"; }
        }
```



```
/// <summary>
        /// Gets the extension Id. Must be unique for each extension.
        /// </summary>
        /// <value>The extension Id</value>
        /// \ \mbox{second} not change this value otherwise it may cause unpredictable behavior
in the application.</remarks>
        protected override Guid ExtensionId
        {
            get { return new Guid("43FF83AB-9F86-495E-9D13-356181639333"); }
        }
        /// <summary>
        /// Gets the extension description.
        /// </summary>
        /// <value>The extension description</value>
        protected override string Description
        {
            get { return "Extension pour la Gestion Commerciale"; }
        }
        /// <summary>
        /// Gets the type of the API interface ApiInterfaceLink generated class.
        /// </summary>
        /// <remarks>Return the type EBP.Api.Interfaces.ApiInterfaceLink generated in the
interface dll EBP.Api.XXXXXX.Interfaces.dll</remarks>
        protected override Type ApiInterfaceLinkType
        {
            get { return typeof(ApiInterfaceLink); }
        }
    }
}
```

5. Compiler le projet. En cas de soucis à la compilation, vérifier si l'application EBP Gestion Commerciale Open Line est bien fermée. Dans le cas contraire, fermer l'application puis relancer la compilation du projet.



# Etape 3 : Ajouter l'extension à un dossier de l'application EBP Gestion Commerciale Open Line

1. Vérifier que vous n'êtes pas en mode développeur



- 2. Créer un nouveau dossier ou ouvrir un dossier existant
- 3. Aller dans le menu Paramètres / Société / Extensions liées
- 4. Cliquer sur le bouton Ajouter pour ajouter votre extension dans votre dossier





5. Sélectionner la dll générée par le projet Visual Studio, puis cliquer sur Ouvrir.

Ouvrir										
	🖒 Reche	rcher dans : Extension G 🔎								
Organiser 🔻 Nouveau dossier					= - 🔟 🔞					
🖉 🔆 Favoris	^ Nom	Modifié le	Туре	Taille						
🛄 Bureau	EBP.Api.Common.dll	02/12/2015 11:30	Extension de l'app	89 Ko						
📳 Emplacements récents	🚳 EBP.Api.dll	02/12/2015 11:30	Extension de l'app	86 Ko						
🍌 Téléchargements	EBP.Api.Main.Interfaces.dll	02/12/2015 11:30	Extension de l'app	19 Ko						
	BP.Api.MYEXT.Interfaces.dll	07/12/2015 16:02	Extension de l'app	219 042 Ko						
🖉 🖳 Ce PC	S Extension Gestion.dll	08/12/2015 11:49	Extension de l'app	6 Ко						
🖻 🌺 Bureau										
Documents										
🗅 📔 Images										
🛛 🚺 Musique										
🛛 🐌 Téléchargements										
Vidéos										
Disque local (C:)	<b>v</b>									
Nom du fichier : Exten	ision Gestion.dll			✓ Exten	sion de l'application (*.dll) 🗸					
				C	Ouvrir Annuler					



#### La fenêtre suivante apparait :



Avertissement : Le bouton Supprimer sert à supprimer l'extension Vous pourrez la réimporter après l'avoir supprimée mais les données en rapport avec celle-ci seront définitivement perdues. Un message d'avertissement est affiché lors de l'utilisation de ce bouton.



🂫 EBI	P Gestion Commerciale Ligne PME Open Line ×							
?	Suppression d'extension							
	Vous êtes sur le point de supprimer l'extension et ses données associées (données saisies, fiches/grilles/listes personnalisées, etc) ? CETTE SUPPRESSION PRENDRA EFFET IMMEDIATEMENT ET SERA DEFINITIVE							
	Vous devez effectuer une sauvegarde du dossier avant cette suppression. Voulez-vous continuer ?							
	Ok Annuler							

- 6. Fermer les options en cliquant sur OK et rouvrir le dossier.
- 7. Cliquer sur le menu Ventes / Clients : les tables créées sont disponibles.

EBP Gestion Commerci	ale Ligne	e PME Open	Line									
Fichier Édition Affichage	Ventes	Règlements	Achats	Sto	ocks /	Articles	Opérations	Paramètres	Outils	Développeur	Fenêtres	?
O- O-	📲 Ope	n Guide Ventes										
Ventes «	Doc	uments de vent	e	•								
Navigation	Clier	nts		•	8	Prospect	s					
Open Guide Ventes	Règi	lements		•	-	Clients						
✓ Documents de vente	Maii	ntenance / SAV		•	-	Tous						
Devis	Plan	ining		•	<u>^</u>	Contacts	;					
Bons de livraison	Con	nmerciaux/colla	borateurs	•		Familles	clients/prospe	cts				
Bons de retour	Imp	ressions		•	5	Sous-fan	nilles clients/p	rospects				
Factures						Historia	ue clients/artic	les				
Avoirs						Chatiatian	ie cherics/ artic					
Avoirs d'acompte						Statistiqu	ies					
Tous						Facturati	on périodique					
✓ Clients						Affaires						
Prospects					Ē	Activités						
Clients					2	Fonction	Contact					
Contacts						Service C	ontact					
M Familles clients/pro						Sous-For	nction Contact					
🌇 Sous-familles client					_							
Historique clients/a												
Statistiques								Clients				Fou
Affaires												
Activités												



8. Voici la fiche Fonction Contact :

Fonction Conta	ct (Nouveau)						×
Fichier Édition \	/ues Fenêtres	?					
릚 Enregistrer  🚹 Enre	egistrer et Fermer	🔒 Enregistro	er et Nouveau	🚖 🖻 🖣	• 🔶 👘 🛛	XIO	×
Général Champs person	inalisés						
Code							
Libellé							
Statut		-					
Nombre de jours de RTT	L						
I							

La personnalisation des fiches, pour les mettre en forme selon vos souhaits, sera à effectuer en mode Développeur, dans le dossier permettant de générer votre extension, afin que toutes les informations soient intégrées dans celle-ci.