

GUIDE DE DEMARRAGE RAPIDE - DIDACTICIEL



The most powerful
way to develop
Web Applications!

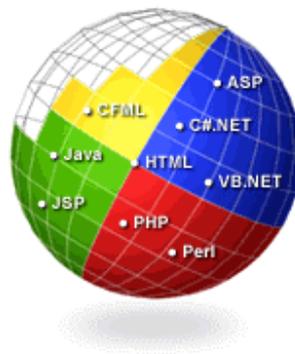




TABLE DES MATIERES

A PROPOS DE LA DOCUMENTATION DE CODECHARGE STUDIO	4
PREAMBULE	5
INTRODUCTION	6
CREATION D'UN GESTIONNAIRE DE TACHES AVEC LE G GENERATEUR D'APPLICATION	7
Etape 1	7
Créer un Nouveau Projet	7
Lancer le Générateur d'Application	7
Spécifier les Propriétés du Projet	8
Sélectionner la Connexion Base de Données	9
Configurer le Générateur d'Application	10
Mise en place de la Sécurité du Site et de l'Authentification	10
Sélectionner les Tables de Bases de Données	11
Configurer les Pages du Site	11
Spécifier la Mise en Page du Site et les Menus	12
Sélectionner le Thème du Site	13
Passer en Revue les Pages et Créer le Site	14
Etape 2	14
Personnalisation de la Page Liste de Tâches	14
Ouvrir la Page Liste de Tâches	15
Tester la Page	15
Relations Implicites	16
Supprimer des Colonnes Inutiles	18
Changer la Légende d'un Champ	18
Synchroniser le code HTML et le Code de Programmation	19
Visualiser et Tester la Live Page	19
Etape 3	20
Ajouter une ListBox Recherche pour les Noms de Projet	20
Ajouter une ListBox Recherche - Insérer une ListBox Contrôle	21
Ajouter une ListBox Recherche - Paramétrer les Propriétés de la ListBox	21
Ajouter une ListBox Recherche - Déplacer une Ligne de la Table	23
Filtrer les Enregistrements de la Grille - Sélectionner la Propriété "Where"	23
Filtrer les Enregistrements de la Grille - Ajouter un Paramètre Recherche	24
Filtrer les Enregistrements de la Grille - Grouper les Paramètres "Where"	25
Filtrer les Enregistrements de la Grille - Paramétrer l'Opérateur AND	25
Visualiser la Page Fonctionnelle	26
Connexion au Système	27
Accéder à la Page Maintenance d'Enregistrement	27
Etape 4	28
Changer des Etiquettes de Champ	28
Créer des Champs Etiquette	29
Réorganiser les Champs Etiquette	30
Prévisualiser la Page Maintenance de Tâches	31
Extension des Fonctionnalités de l'Application avec la Programmation d'Evénements	33
ASP et VBScript	33
Extension des Fonctionnalités de l'Application avec la Programmation d'Evénements (ASP et VBScript)	33
Etape 1	33
Utiliser l'Événement Before Show Row pour Changer la Couleur du Texte	33
Contrôler par Programmation la Valeur Champ	34
Prévisualiser la Page Liste de Tâches	35
Etape 2	36
Modifier un Champ Etiquette sur la Page Maintenance de Tâches	36
Utiliser l'Événement Before Show pour Changer la Valeur d'une Etiquette	37
Etape 3	39
Ajouter un Champ Caché "Assigned By" pour la Mise à Jour Automatique de Nouvelles Tâches	39
Ajouter un Champ Caché "Date Created" au Formulaire Enregistrement	40
Tester les Champs Etiquette et Caché	40
Etape 4	41



Programmation du Formulaire Enregistrement.....	41
Ajouter du Code dans l'Événement After Insert pour Envoyer des Emails	41
Utiliser l'Événement After Update pour Envoyer des Emails	44
Test de l'Envoi d'Email	45
Etape 5	45
Implémenter la Sécurité d'Enregistrement dans l'Événement After Initialize.....	45
C# et VB.Net	46
Extension des Fonctionnalités de l'Application avec la Programmation d'Événements (C# et VB.Net)	46
Etape 1	48
Utiliser l'Événement Before Show Row pour Changer la couleur du Texte	48
Contrôler par Programmation la Valeur Champ	48
Prévisualiser la Page Liste de Tâches	49
Etape 2	50
Modifier un Champ Etiquette sur la Page Maintenance de Tâches	50
Utiliser l'Événement Before Show pour Changer la Valeur d'une Etiquette	51
Etape 3	54
Ajouter un Champ Caché "Assigned By" pour la Mise à Jour Automatique de Nouvelles Tâches	54
Ajouter un Champ Caché "Date Created" au Formulaire Enregistrement	54
Tester les Champs Etiquette et Caché.....	55
Etape 4	56
Programmation du Formulaire Enregistrement.....	56
Ajouter du Code dans l'Événement After Insert pour Envoyer des Emails	56
Utiliser l'Événement After Update pour Envoyer des Emails	59
Test de l'Envoi d'Email	61
Etape 5	62
Implémenter la Sécurité d'Enregistrement dans l'Événement After Initialize.....	62
JSP	65
Extension des Fonctionnalités de l'Application avec la Programmation d'Événements (JSP)	65
Etape 1	65
Utiliser l'Événement Before Show Row pour Changer la Couleur du Texte	65
Contrôler par Programmation la Valeur Champ	66
Prévisualiser la Page Liste de Tâches	67
Etape 2	68
Modifier un Champ Etiquette sur la Page Maintenance de Tâches	68
Utiliser l'Événement Before Show pour Changer la Valeur d'une Etiquette	68
Etape 3	71
Ajouter un Champ Caché "Assigned By" pour la Mise à Jour Automatique de Nouvelles Tâches	71
Ajouter un Champ Caché "Date Created" au Formulaire Enregistrement	72
Tester les Champs Etiquette et Caché.....	72
Etape 4	73
Programmation du Formulaire Enregistrement.....	73
Ajouter du Code dans l'Événement After Insert pour Envoyer des Emails	73
Utiliser l'Événement After Update pour Envoyer des Emails	75
Test de l'Envoi d'Email	76
Etape 5	76
Implémenter la Sécurité d'Enregistrement dans l'Événement After Initialize.....	76
PHP	78
Extension des Fonctionnalités de l'Application avec la Programmation d'Événements (PHP)	78
Etape 1	79
Utiliser l'Événement Before Show Row pour Changer la Couleur du Texte	79
Contrôler par Programmation la Valeur Champ	79
Prévisualiser la Page Liste de Tâches	80
Etape 2	81
Modifier un Champ Etiquette sur la Page Maintenance de Tâches	81
Utiliser l'Événement Before Show pour Changer la Valeur d'une Etiquette	82
Etape 3	84
Ajouter un Champ Caché "Assigned By" pour la Mise à Jour Automatique de Nouvelles Tâches	84
Ajouter un Champ Caché "Date Created" au Formulaire Enregistrement	85
Tester les Champs Etiquette et Caché.....	86
Etape 4	87
Programmation du Formulaire Enregistrement.....	87
Ajouter du Code dans l'Événement After Insert pour Envoyer des Emails	88
Utiliser l'Événement After Update pour Envoyer des Emails	90



Test de l'Envoi d'Email	91
Etape 5	92
Implémenter la Sécurité d'Enregistrement dans l'Événement After Initialize	92
ColdFusion	93
Extension des Fonctionnalités de l'Application avec la Programmation d'Événements (ColdFusion)	93
Etape 1	94
Utiliser l'Événement Before Show Row pour Changer la Couleur du Texte	94
Contrôler par Programmation la Valeur Champ	94
Prévisualiser la Page Liste de Tâches	95
Etape 2	96
Modifier un Champ Etiquette sur la Page Maintenance de Tâches	96
Utiliser l'Événement Before Show pour Changer la Valeur d'une Etiquette	97
Etape 3	99
Ajouter un Champ Caché "Assigned By" pour la Mise à Jour Automatique de Nouvelles Tâches	99
Ajouter un Champ Caché "Date Created" au Formulaire Enregistrement	100
Tester les Champs Etiquette et Caché	100
Etape 4	101
Programmation du Formulaire Enregistrement	101
Ajouter du Code dans l'Événement After Insert pour Envoyer des Emails	101
Utiliser l'Événement After Update pour Envoyer des Emails	103
Test de l'Envoi d'Email	104
Etape 5	104
Implémenter la Sécurité d'Enregistrement dans l'Événement After Initialize	104
Perl	106
Extension des Fonctionnalités de l'Application avec la Programmation d'Événements (Perl)	106
Etape 1	106
Utiliser l'Événement Before Show Row pour Changer la Couleur du Texte	106
Contrôler par Programmation la Valeur Champ	107
Prévisualiser la Page Liste de Tâches	108
Etape 2	110
Modifier un Champ Etiquette sur la Page Maintenance de Tâches	110
Utiliser l'Événement Before Show pour Changer la Valeur d'une Etiquette	110
Etape 3	113
Ajouter un Champ Caché "Assigned By" pour la Mise à Jour Automatique de Nouvelles Tâches	113
Ajouter un Champ Caché "Date Created" au Formulaire Enregistrement	113
Tester les Champs Etiquette et Caché	114
Etape 4	115
Programmation du Formulaire Enregistrement	115
Ajouter du Code dans l'Événement After Insert pour Envoyer des Emails	115
Utiliser l'Événement After Update pour Envoyer des Emails	117
Test de l'Envoi d'Email	118
Etape 5	118
Implémenter la Sécurité d'Enregistrement dans l'Événement After Initialize	118
Conclusion	120



A PROPOS DE LA DOCUMENTATION DE CODECHARGE STUDIO

La documentation du logiciel CodeCharge Studio est décomposée en quatre volumes :

Volume n°1 : Le **Guide Utilisateur** est une introduction à CodeCharge Studio et il explique comment utiliser les nombreuses fonctionnalités disponibles dans l'IDE (Environnement de Développement Intégré) de CodeCharge Studio. Si vous découvrez CodeCharge Studio, nous vous recommandons avant toute chose de lire le Guide Utilisateur afin d'acquérir une connaissance fonctionnelle de l'environnement de CodeCharge Studio.

Volume n°2 : Le **Guide de Démarrage Rapide (Didacticiel)** contient des instructions détaillées, étape par étape, destinées à guider les nouveaux utilisateurs pour réaliser différents types de fonctions au sein de CodeCharge Studio. Ce guide propose une approche pratique pour se former au logiciel par la construction d'exemples d'applications Web de A à Z.

Volume n°3 : Le **Guide de Référence des Composants** contient des informations sur la programmation des différents composants qui sont utilisés pour créer des applications Web utilisant CodeCharge Studio. Cela inclut les différents Contrôles, Méthodes, Propriétés, Événements et Actions. Une fois que vous aurez acquis une bonne connaissance de l'IDE de CodeCharge Studio (Cf. Guide Utilisateur), nous vous recommandons de lire le Guide de Référence. Ce guide sert également de référence lors de l'écriture de code événementiel ou de code personnalisé.

Volume n°4 : La **Section Exemples et Techniques** contient des astuces et des exemples de programmes pour réaliser des tâches communes et variées utilisant du code événementiel ou du code personnalisé. Cette section montre des cas concrets pour appliquer certaines des constructions qui sont détaillées dans le Guide de Référence des Composants. Les techniques démontrées peuvent également être utilisées dans des projets en cours pour atteindre différents objectifs.



PREAMBULE

CodeCharge et CodeCharge Studio sont des logiciels développés par la société américaine YesSoftware, Inc.

CodeCharge et CodeCharge Studio sont des marques déposées de YesSoftware, Inc.

Toutes les autres marques citées sont des marques ou des marques déposées de leurs propriétaires respectifs.

Le présent **Guide de Démarrage Rapide (Didacticiel)** a été traduit en français par la société française KAPITEC SOFTWARE S.A.S. (mars 2004).

KAPITEC SOFTWARE S.A.S. est le distributeur exclusif des logiciels CodeCharge et CodeCharge Studio pour la France.

Coordonnées de l'éditeur :



YesSoftware, Inc.

6330 S. Eastern Ave. Suite 5 - Las Vegas, NV 89119 - Etats-Unis

Tél.: +1 (888) 241-7338 - Fax : +1 (866) 312-8049 - www.codecharge.com

Coordonnées du distributeur français :



KAPITEC SOFTWARE S.A.S.

57, boulevard de l'Embouchure - 31075 Toulouse Cedex 2 - France

Tél. : 05 34 27 90 03 - Fax : 05 34 27 90 04 - www.kapitec.com

NOTE SUR L'INTERFACE EN FRANÇAIS :

L'interface en français de CodeCharge Studio est disponible depuis la version 2.

1 - Pour changer la langue de l'interface de CodeCharge Studio, sélectionnez le menu **Tools**, puis **Language** et après **Français**.

2 - Pour que le changement de langue soit effectif, relancez CodeCharge Studio.



INTRODUCTION

Le **Guide de Démarrage Rapide (Didacticiel)** détaille les différentes étapes pour créer des applications Web, et il est destiné à vous former de façon pratique à l'utilisation du logiciel CodeCharge Studio. Les applications Web détaillées vont de la simple application avec une page à des applications entièrement fonctionnelles. La plupart des étapes nécessaires pour construire les applications sont décrites dans ce guide. Toutefois, si vous aviez besoin de plus de détails concernant une caractéristique spécifique, merci de vous référer au Guide Utilisateur. Il est à noter que nous partons du principe que vous disposez d'un serveur Web complètement fonctionnel (sur lequel les pages Web seront publiées) et d'une base de données fonctionnelle. Là où une connexion base de données est requise, l'exemple de base de données Intranet (se trouvant dans le dossier exemples de l'installation de CodeCharge Studio) est utilisé.

Le Guide de Démarrage Rapide traduit en français propose à ce jour uniquement le didacticiel sur la Création d'un Gestionnaire de Tâches avec le Générateur d'Application de CodeCharge Studio.

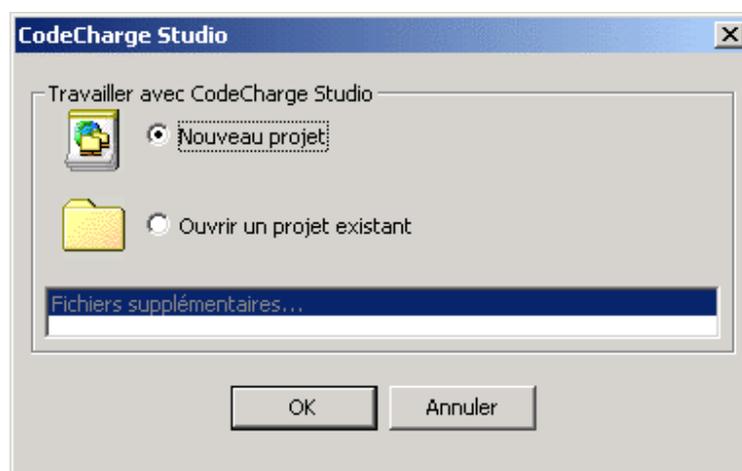


CREATION D'UN GESTIONNAIRE DE TACHES AVEC LE GENERATEUR D'APPLICATION

Le Générateur d'Application est une puissante fonctionnalité de CodeCharge Studio qui peut automatiquement convertir une base de données en une application Web opérationnelle. En suivant ces quelques étapes courtes, vous pouvez créer des pages pour ajouter, éditer et passer en revue le contenu d'une base de données, ou créer un squelette d'application Web que vous pourrez étendre en une application Web complète. Ce didacticiel présente étape par étape les instructions nécessaires pour créer un Gestionnaire de Tâches prêt à l'emploi : la création se fait principalement avec le Générateur d'Application de CodeCharge Studio, et l'application est ensuite finalisée en faisant appel à d'autres caractéristiques du logiciel.

ETAPE 1

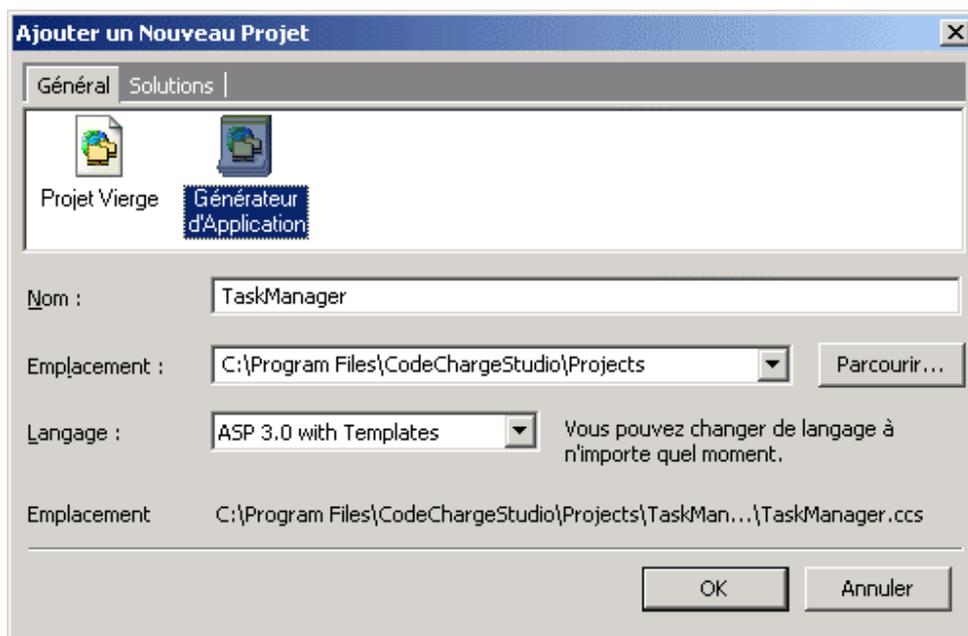
Créer un Nouveau Projet



- 1 - Démarrez CodeCharge Studio.
- 2 - Sélectionnez **Nouveau projet** dans l'écran initial.

Lancer le Générateur d'Application

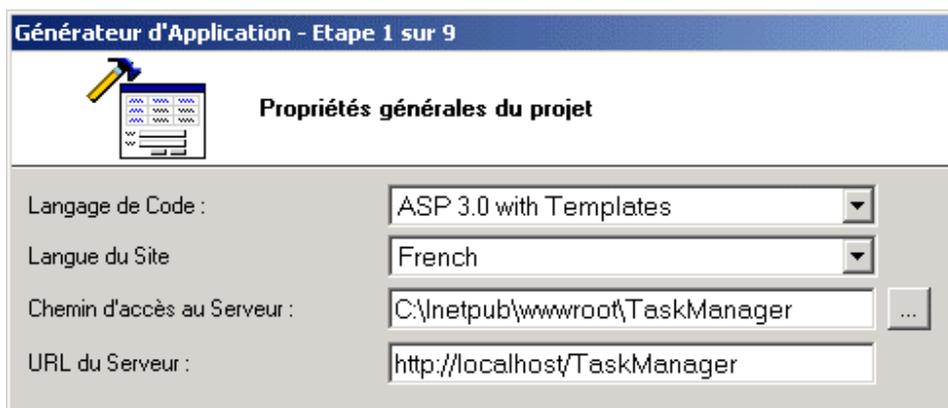
Pour démarrer le Générateur d'Application :



- 1 - Spécifiez le nom du nouveau projet : *TaskManager* (Gestionnaire de Tâches), l'emplacement sur le disque sur lequel le projet sera publié, et le langage de programmation.
- 2 - Double-cliquez sur l'icône **Générateur d'Application**.
- 3 - Sélectionnez l'icône **Générateur d'Application**, et ensuite cliquez sur le bouton **OK** pour le lancer.

Spécifier les Propriétés du Projet

Spécifiez un nombre de paramètres requis par le Générateur d'Application pour générer le site.



1 - Langage du Code :

Langage de programmation ou technologie devant être généré. Les technologies actuellement supportées par CodeCharge Studio sont :

ASP 3.0 avec Templates : génère du code ASP 3.0 et des templates .html séparées.

ASP.Net C# : génère des fichiers .aspx avec du code C#.

ASP.Net VB : génère des fichiers .aspx avec du code VB.Net.

CFML 4.0.1 : génère du code ColdFusion 4.0.1.

CFML 4.0.1 avec Templates : génère du code ColdFusion 4.0.1 (.cfm) et des fichiers template .html séparés.



JSP 1.1 JDK 1.3 : génère du code JSP 1.1.

PHP 4.0 avec Templates : génère du code PHP 4.0 (.php) et des fichiers template .html séparés.

Servlets 2.2 JDK 1.3 avec Templates : génère du code Java qui utilisent des templates .html.

2 - Langue du Site :

Spécifiez la langue parlée devant être utilisée lors de la génération de messages texte pour le site. Par exemple, le texte *No records*, qui apparaît quand il n'y a plus d'autre enregistrement à afficher dans une grille, peut être généré dans n'importe laquelle des langues supportées.

3 - Chemin d'Accès du Serveur :

Le chemin d'accès complet où les fichiers générés doivent être publiés (localement). Ce chemin est habituellement prédéfini par le Générateur d'Application et peut être laissé tel quel.

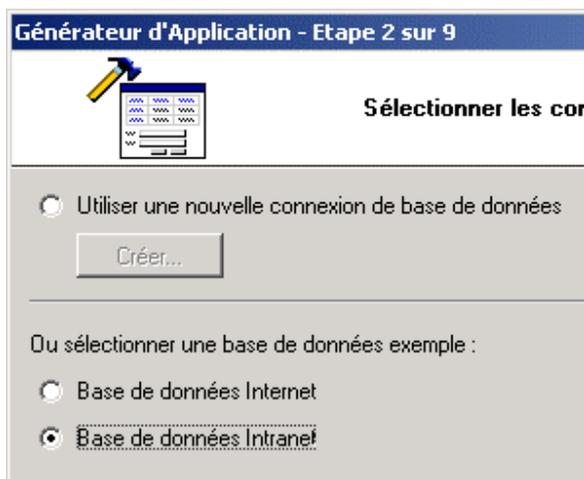
4 - URL du Serveur :

L'adresse Web correspondant au Chemin d'Accès du Serveur. Cette URL sera utilisée pour visualiser les pages en mode **Live Page**. Le Générateur d'Application détecte automatiquement l'URL appropriée qui correspond au chemin d'accès du serveur.

5 - Cliquez sur **Suivant** pour continuer.

Sélectionner la Connexion Base de Données

Cette application utilise l'exemple de base de données *Intranet*, aussi vous n'avez pas besoin de créer une nouvelle connexion base de données. L'exemple de base de données *Intranet* contient les tables *tasks* (tâches), *priorities* (priorités) et *employees* (employés) qui seront utilisées dans ce projet.



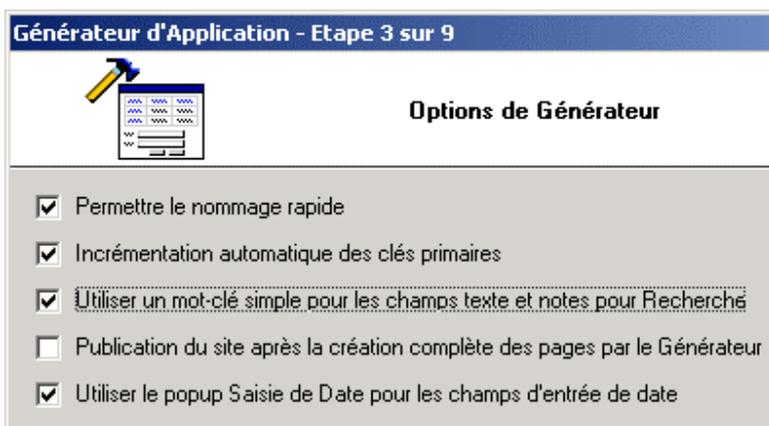
1 - Cochez **Base de données Intranet**.

2 - Cliquez sur **Suivant** pour continuer.

Il est à noter que vous pouvez utiliser cet exemple de base de données sans tenir compte du langage que vous avez sélectionné dans l'étape précédente, car la connexion en mode conception est faite par CodeCharge Studio et est indépendante du langage utilisé. Si vous utilisez un langage tel que PHP ou ColdFusion, vous devrez ultérieurement configurer la connexion côté serveur pour vous connecter à MySQL ou à une autre base de données.



Configurer le Générateur d'Application



1 - Configurez le Générateur d'Application en spécifiant les options de configuration comme indiqué ci-dessous :

Permettre le nommage rapide : Sélectionnez cette option aussi le Générateur d'Application convertira automatiquement les noms des tables en légendes en Anglais. Par exemple, la table *employees* (employés) sera montrée comme une grille avec le titre "*List of Employees*". Des champs comme *task_name* seront convertis en en-têtes de colonne, tels que "*Task Name*".

Incrémentation automatique des clés primaires : Sélectionnez cette option pour indiquer que les tables de base de données contiennent des champs clé qui sont automatiquement incrémentés. Le Générateur d'Application cachera alors les champs clé des formulaires Maintenance d'Enregistrement puisque les utilisateurs n'ont pas besoin de saisir des valeurs clé.

Utiliser un mot-clé simple pour les champs texte et notes pour Recherche : Sélectionnez cette option pour générer un seul champ Recherche qui recherche tous les champs texte et notes dans les tables/grilles. Si cette option est décochée, le Générateur d'Application crée une section Recherche avec plusieurs champs Recherche (un champ Recherche pour chaque champ texte ou notes dans la table de la base de données).

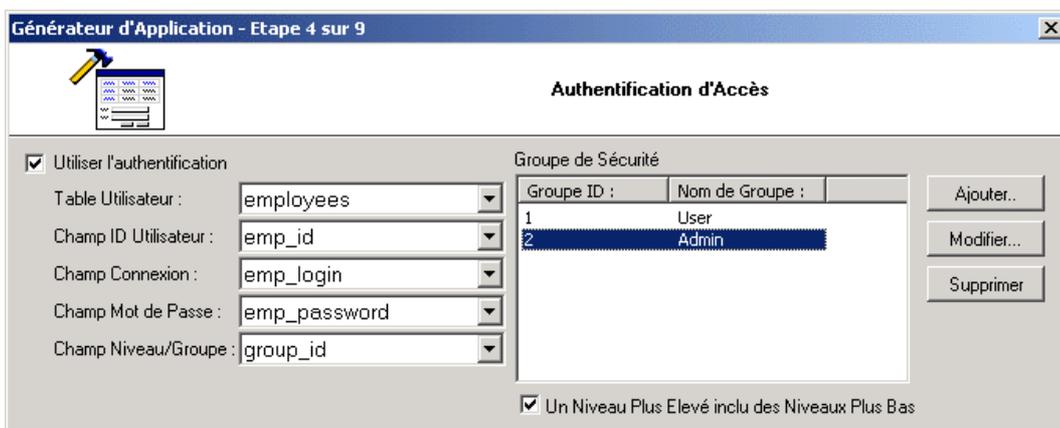
Publication du Site après la création complète des pages par le Générateur : Indiquez que vous voulez publier le site dès que le Générateur d'Application aura créé toutes les pages nécessaires. Ainsi, vous n'aurez pas à vous inquiéter de savoir si vous avez oublié de générer/publier des fichiers requis pour le bon fonctionnement de l'application.

Utiliser le popup Saisie de Date pour les champs d'Entrée de date : Sélectionnez cette option aussi les champs qui sont détectés comme étant des champs date sont créés avec un composant Saisie de Date (*Date Picker*) correspondant. Le composant Saisie de Date permet à l'utilisateur de spécifier facilement les valeurs de date.

2 - Cliquez sur **Suivant** pour continuer.

Mise en place de la Sécurité du Site et de l'Authentification

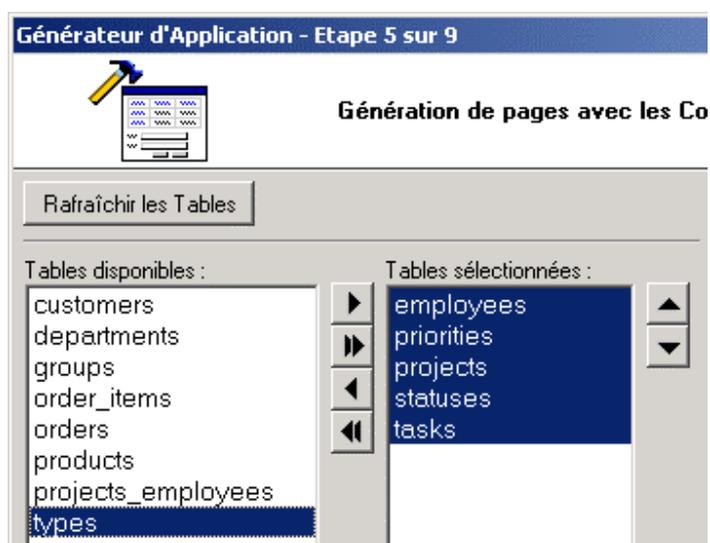
Dans cette étape, vous pouvez indiquer si vous voulez utiliser l'authentification et vérifier les droits d'accès des utilisateurs avant de leur permettre d'accéder à certaines pages.



1 - Sélectionnez **Utiliser l'authentification** et laissez toutes les options par défaut. Le Générateur d'Application créera alors la page Login et vous permettra de définir un niveau de sécurité pour chacune des pages créées.

2 - Cliquez sur **Suivant** pour continuer.

Sélectionner les Tables de Bases de Données



1 - Sélectionnez les tables de bases de données suivantes sur lesquelles l'application sera construite :

employees (employés)

priorities (priorités)

projects (projets)

statuses (statuts)

tasks (tâches)

2 - Cliquez sur **Suivant** pour continuer.

Configurer les Pages du Site

Maintenant vous devez configurer votre site en spécifiant les options et les paramètres de sécurité pour chacune des pages. Le Générateur d'Application crée deux pages pour chacune des tables :



Page Recherche (Chercher) et Grille

Page Maintenance d'Enregistrement

En cliquant sur le nom de n'importe laquelle des tables, vous pouvez indiquer si les formulaires Recherche, Grille et Enregistrement doivent tous être sur la même page pour chacune des tables.

1 - Pour ce didacticiel, configurez les pages comme indiqué ci-après. Le Générateur d'Application convertira alors les tables en pages Web :

Table	Pages de Grilles			Maintenance d'enregistrement/page de visualisation		
	Grille	Chercher	Niveau de Sécurité	Enregistrement	Pouvant être mis à jour	Niveau de Sécurité
(Cliquez pour plus de détails) Tous	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
employees	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	User	<input checked="" type="checkbox"/>	<input type="checkbox"/>	User
priorities	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Admin	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Admin
projects	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Admin	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Admin
statutes	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Admin	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Admin
tasks	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	None	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	User

Table *employees* : page liste de choix d'employés et page informations sur les employés, seulement accessibles par les utilisateurs autorisés (*User*).

Table *priorities* : page liste des priorités et page maintenance des priorités, seulement accessibles par les administrateurs (*Admin*).

Table *projects* : page liste de projets et page maintenance des projets, seulement accessibles par les administrateurs (*Admin*).

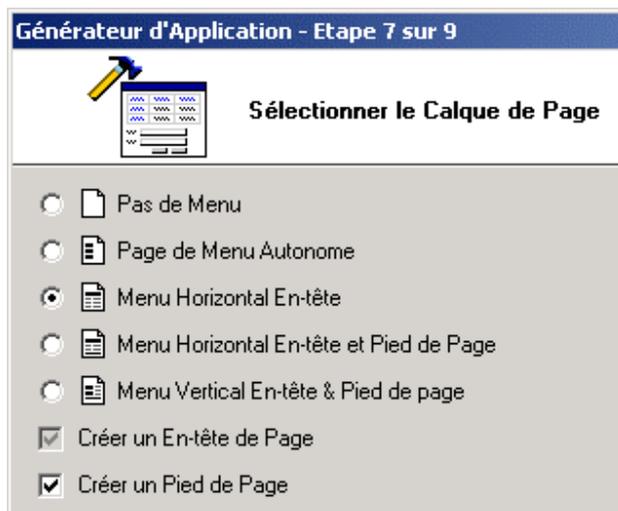
Table *statutes* : liste des statuts et page maintenance des statuts, seulement accessibles par les administrateurs (*Admin*).

Table *tasks* : liste de choix des tâches accessible par n'importe qui (*None*), et page maintenance des tâches accessible par les utilisateurs autorisés (*User*).

2 - Cliquez sur **Suivant** pour continuer.

Spécifier la Mise en Page du Site et les Menus

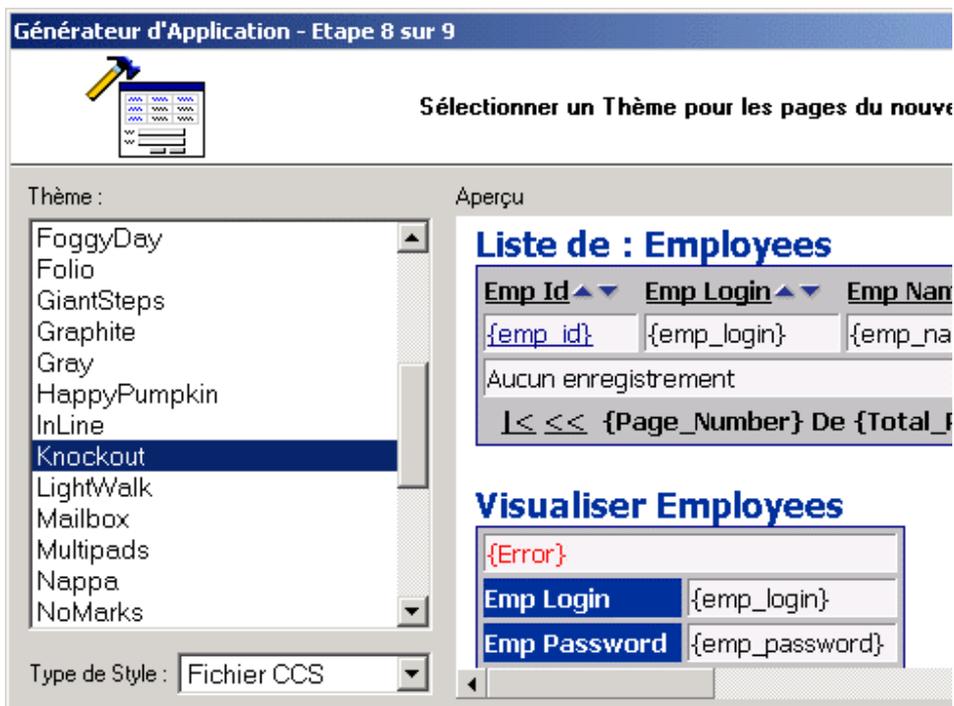
Le Générateur d'Application crée un en-tête de page avec un menu qui est alors inclu dans toutes les autres pages pour faciliter la navigation.



Cliquez sur **Suivant** pour laisser la position horizontale par défaut des menus pour toutes les pages.

Sélectionner le Thème du Site

Pour terminer...



1 - Sélectionnez le thème *Knockout* à appliquer au site.

Le Générateur d'Application utilisera le thème pour appliquer des polices et des couleurs définies, à chaque page. Le champ **Type de Style** doit être paramétré sur le fichier CSS¹.

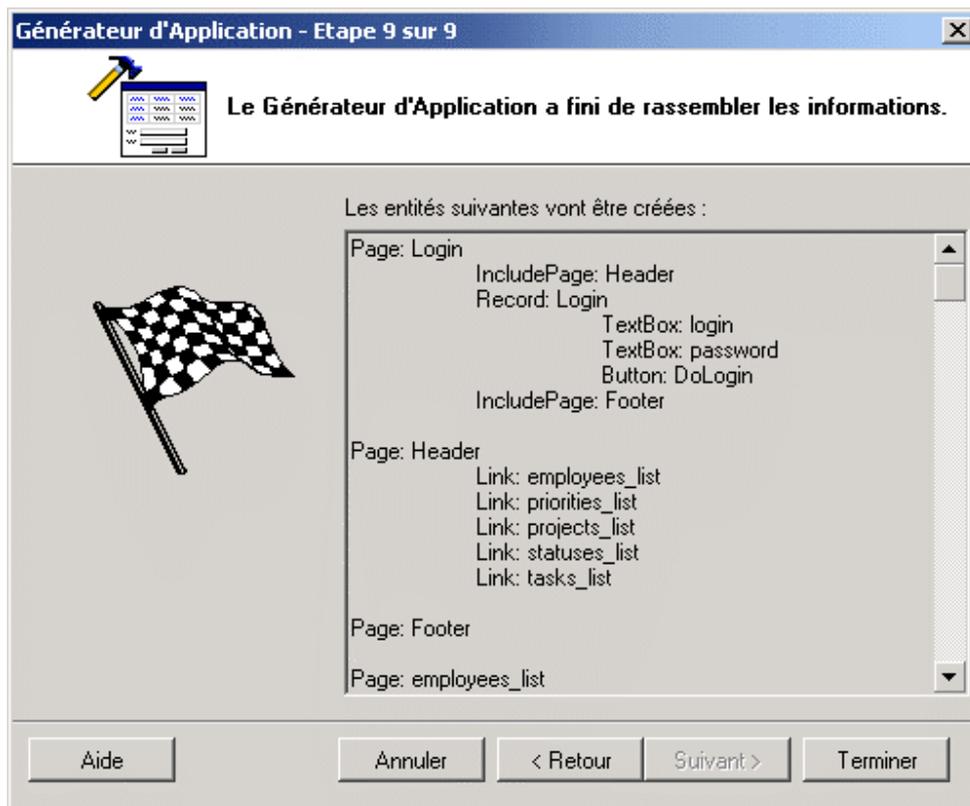
2 - Cliquez sur **Suivant** pour continuer.

¹ CCS : Cascading Style Sheet. Les définitions de style sont stockées dans un fichier .css externe.



Passer en Revue les Pages et Créer le Site

La dernière fenêtre contient un résumé de l'application qui sera construite.



Pour construire l'application, cliquez sur le bouton **Terminer**.

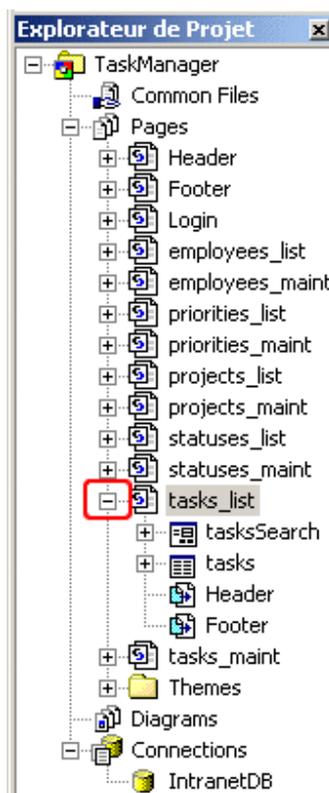
ETAPE 2

Personnalisation de la Page Liste de Tâches

Après avoir créé les pages avec le Générateur d'Application, vous voudrez probablement personnaliser les pages pour répondre à vos besoins spécifiques. Par exemple, vous pourrez vouloir changer du texte et des légendes, enlever certaines des colonnes de la grille, ajouter des champs supplémentaires de recherche et des ListBoxes, etc.



Ouvrir la Page Liste de Tâches



Dans la fenêtre **Explorateur de Projet** cliquez sur le signe '+' à côté de la page *tasks_list* pour l'ouvrir, ceci afin d'effectuer des modifications.

Tester la Page

- 1 - Une fois la page ouverte, cliquez sur **Live Page** pour visualiser et tester la page fonctionnelle.
- 2 - Utilisez **george/george** comme Login.
- 3 - Après la page Login, vous devriez voir une page similaire à celle montrée ci-dessous :

[Employees](#) [Priorities](#) [Projects](#) [Statuses](#) [Tasks](#)

Rechercher Tasks

Mot clé	<input type="text"/>
<input type="button" value="Rechercher"/>	

Liste de : Tasks

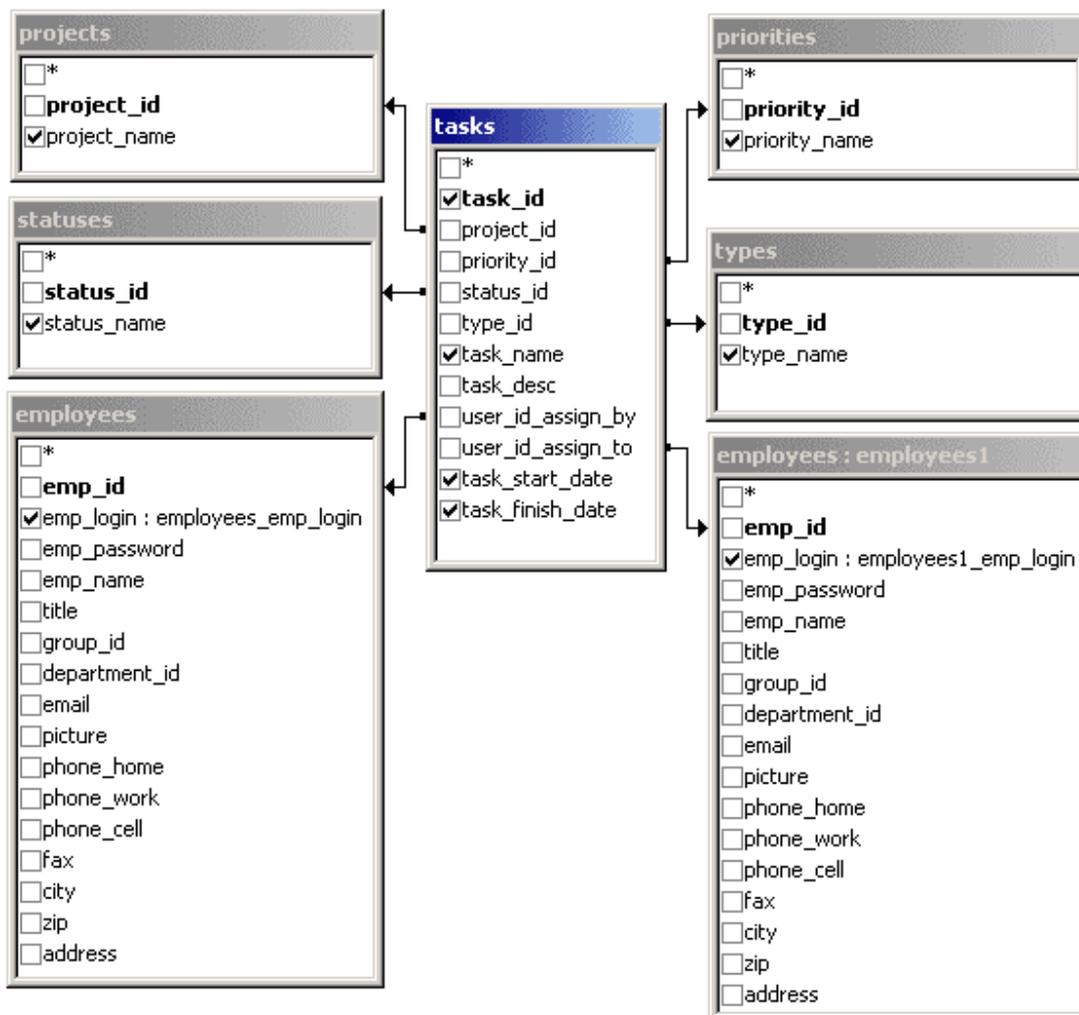
<u>Id</u>	<u>Project Name</u>	<u>Priority Name</u>	<u>Status Name</u>	<u>Type Name</u>	<u>Name</u>	<u>Emp Login</u>	<u>Emp Login</u>	<u>Start Date</u>	<u>Finish Date</u>
1	CodeCharge	High	On hold	Issue	Great Project needs	ken	helen	01.02.2003	02.02.2003
2	CodeCharge	Highest	Closed	Question	Fix ALL bugs	ken	george	02.02.2003	03.02.2003
3	CodeCharge	High	Closed	Task	Get ready to click	ken	peter	03.02.2003	04.02.2003
4	My Project	Highest	Open	Task	Finish My Project	ken	ignace	04.02.2003	05.02.2003
5	Test Project	High	In progress	Task	Test this project.	ken	ken	05.02.2003	06.02.2003
6	CodeCharge	Highest	Open	Task	Code with one hand	ken	alexander	06.02.2003	07.02.2003
7	Test Project	Highest	On hold	Task	Get armed	ken	helen	07.02.2003	08.02.2003
8	Test Project	Highest	Open	Question	Write more code	ken	ignace	08.02.2003	09.02.2003
9	Super Project	Highest	In progress	Task	Code, code, code...	ken	george	09.02.2003	10.02.2003
10	Test Project	Lowest	On hold	Task	Sleep	ken	ken	10.02.2003	11.02.2003
11	Super Project	Highest	Open	Task	Have fun	ken	alexander	11.02.2003	12.02.2003

[Nouveau](#) 1 of 1

Le formulaire Grille dans cette page est basé sur la table *tasks* de la base de données. Toutefois, quand le Générateur d'Application a créé le formulaire, il a également pris en compte toutes les relations existantes entre la table *tasks* de la base de données et d'autres tables avec des noms de champs similaires. Aussi par exemple, le champ *Project Name* ci-dessous est basé sur une relation entre la table *tasks* et la table *projects*, qui contient les noms de tous les projets.

Relations Implicites

Concrètement le formulaire Grille dans le *tasks_list* n'est pas basé seulement sur la table *tasks*, mais plutôt sur une requête qui réunit tous les champs dans les différentes tables liées à la table *tasks*. Quand le Générateur d'Application crée un formulaire, il tente de détecter toute relation existante entre la table de la base (dans ce cas précis la table *tasks*), et d'autres tables au sein de la base de données. Si une relation est trouvée, le Générateur d'Application crée une requête pour faire une jointure des champs qui constituent la relation. Le diagramme ci-dessous montre la jointure utilisée pour le formulaire Grille dans la page *tasks_list* :



Il est à noter que la capacité à détecter des relations varie en fonction du type de base de données. Alors qu'il est possible de détecter des relations dans des bases de données, telles que Microsoft Access, d'autres bases de données, comme MySQL, ne supportent pas cette caractéristique. Si le Générateur d'Application n'est pas en mesure de détecter les relations, elles devront alors être implémentées manuellement en utilisant la propriété **Data Source** (Source de Données) du formulaire concerné.

A moins que cela ne soit spécifié différemment pendant le processus de création, le Générateur d'Application construit des formulaires Grille qui incluent tous les champs dans une table donnée. Cependant, vous pouvez très bien vouloir ne pas montrer certains champs.



Liste de : Tasks

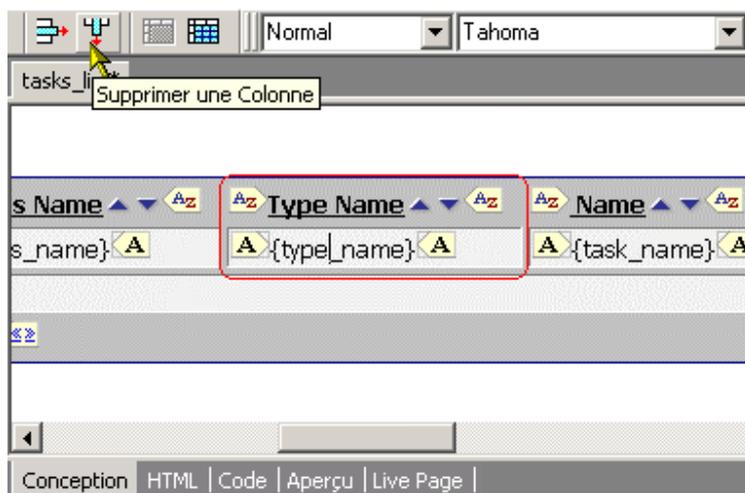
<u>Id</u>	<u>Project Name</u>	<u>Priority Name</u>	<u>Status Name</u>	<u>Type Name</u>	<u>Name</u>	<u>Emp Login</u>	<u>Emp Login</u>	<u>Start Date</u>	<u>Finish Date</u>
1	CodeCharge	High	On hold	Issue	Great Project needs to be greater	ken	helen	01.02.2003	02.02.2003
2	CodeCharge	Highest	Closed	Question	Fix ALL bugs	ken	george	02.02.2003	03.02.2003
3	CodeCharge	High	Closed	Task	Get ready to click	ken	peter	03.02.2003	04.02.2003
4	My Project	Highest	Open	Task	Finish My Project	ken	ignace	04.02.2003	05.02.2003
5	Test Project	High	In progress	Task	Test this project.	ken	ken	05.02.2003	06.02.2003
6	CodeCharge	Highest	Open	Task	Code with one hand.	ken	alexander	06.02.2003	07.02.2003
7	Test Project	Highest	On hold	Task	Get armed	ken	helen	07.02.2003	08.02.2003
8	Test Project	Highest	Open	Question	Write more code	ken	ignace	08.02.2003	09.02.2003
9	Super Project	Highest	In progress	Task	Code, code, code...	ken	george	09.02.2003	10.02.2003
10	Test Project	Lowest	On hold	Task	Sleep	ken	ken	10.02.2003	11.02.2003
11	Super Project	Highest	Open	Task	Have fun	ken	alexander	11.02.2003	12.02.2003

Nouveau 1 of 1

Dans notre cas, nous ne voulons pas montrer les champs *Type Name*, *Emp Login*, *Start Date* et *Finish Date*. Vous noterez aussi qu'il y a deux champs *Emp Login* : le premier champ correspond au champ *user_id_assign_by* dans la table *tasks*, et le second correspond à *user_id_assign_to*. Nous enlèverons le premier champ qui correspond à *user_id_assign_by* et renommerons alors le second avec un nom plus significatif.

Supprimer des Colonnes Inutiles

Pour supprimer des colonnes :



1 - Repassez en mode **Conception** et sélectionnez la colonne inutile en cliquant et en positionnant le curseur à l'intérieur de la colonne.

2 - Cliquez sur l'icône **Supprimer une Colonne** dans la barre d'outils pour enlever la colonne.

Utilisez cette méthode pour supprimer les colonnes suivantes :

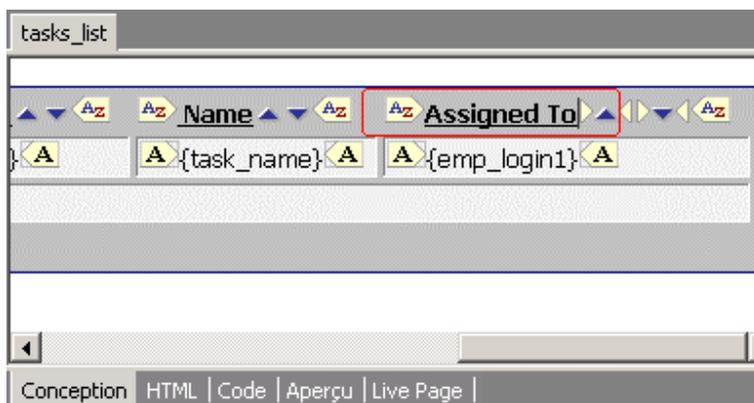
Type Name

Emp Login (le premier champ)

Start Date

Finish Date

Changer la Légende d'un Champ



Toujours en mode **Conception**, changez le texte de légende pour les champs suivants :

Emp Login \rightarrow Assigned To

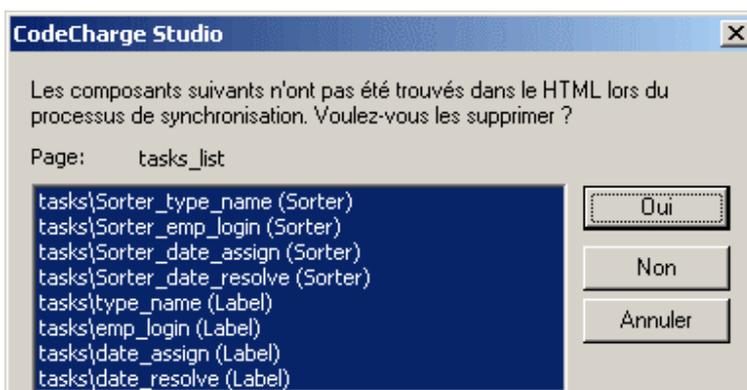
Project Name \rightarrow Project

Priority Name \rightarrow Priority

Status Name \rightarrow Status

Synchroniser le code HTML et le Code de Programmation

Des champs ayant été précédemment supprimés, nous avons besoin de synchroniser le code de programmation pour qu'il corresponde au code HTML. Pour synchroniser :



1 - Cliquez sur l'onglet **Live Page**.

Une fenêtre avec un message s'affiche pour vous avertir que certains des composants n'ont pas été trouvés dans le code HTML.

2 - Cliquez sur **Oui** pour confirmer la suppression des colonnes et continuez.

Visualiser et Tester la Live Page

Enfin, vous pouvez visualiser votre page avec une grille contenant la liste des tâches qui peuvent être triées en cliquant sur les en-têtes de colonnes, ou recherchées en saisissant un mot clé.



[Employees](#) [Priorities](#) [Projects](#) [Statutes](#) [Tasks](#)

Rechercher Tasks

Mot clé	<input type="text"/>
<input type="button" value="Rechercher"/>	

Liste de : Tasks

<u>Id</u>	<u>Project</u>	<u>Priority</u>	<u>Status</u>	<u>Name</u>	<u>Assigned To</u>
1	CodeCharge	High	On hold	Great Project needs to be greater	helen
2	CodeCharge	Highest	Closed	Fix ALL bugs	george
3	CodeCharge	High	Closed	Get ready to click	peter
4	My Project	Highest	Open	Finish My Project	ignace
5	Test Project	High	In progress	Test this project.	ken
6	CodeCharge	Highest	Open	Code with one hand.	alexander
7	Test Project	Highest	On hold	Get armed	helen
8	Test Project	Highest	Open	Write more code	ignace
9	Super Project	Highest	In progress	Code, code, code...	george
10	Test Project	Lowest	On hold	Sleep	ken
11	Super Project	Highest	Open	Have fun	alexander

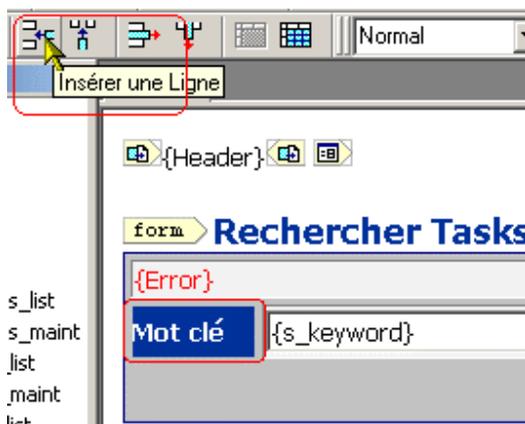
Nouveau 1 of 1

ETAPE 3

Ajouter une ListBox Recherche pour les Noms de Projet

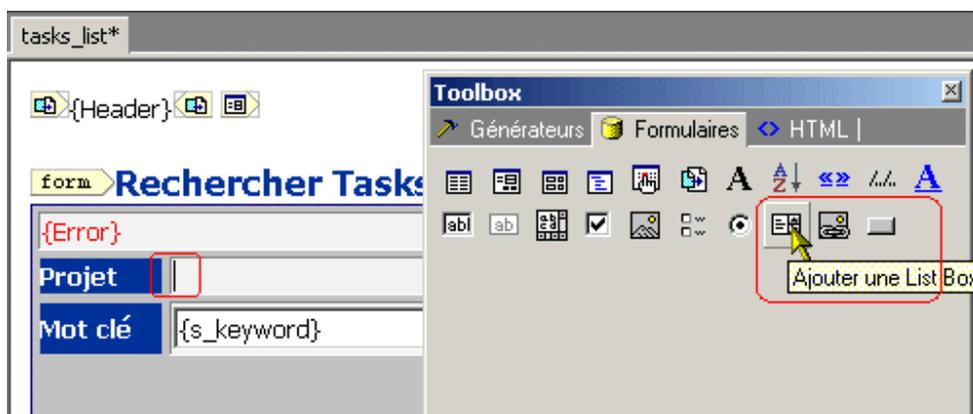
Maintenant, nous allons ajouter un champ supplémentaire de recherche au formulaire Recherche. Nous allons ajouter une ListBox avec des noms de projet, aussi les utilisateurs pourront filtrer la grille en sélectionnant un projet depuis la ListBox.

En premier lieu, nous avons besoin d'ajouter une ligne dans le formulaire Recherche dans lequel la ListBox sera placée.



- 1 - Positionnez le curseur quelque part au sein du texte **Mot clé** en cliquant dessus.
- 2 - Sélectionnez l'icône **Insérer une ligne** pour ajouter une nouvelle ligne de table en haut de la section Recherche.

Ajouter une ListBox Recherche - Insérer une ListBox Contrôle



- 1 - Saisissez le texte *Project* au sein de la cellule nouvellement créée (à gauche de la table).
- 2 - Positionnez le curseur dans la cellule de droite, comme montré ci-dessus.
- 3 - Cliquez sur l'icône **Ajouter une ListBox** dans la **Boîte à Outils** (*Toolbox*) pour l'ajouter à la page.

Ajouter une ListBox Recherche - Paramétrer les Propriétés de la ListBox

Configurez les propriétés de la ListBox en cliquant dessus et en indiquant les valeurs requises dans l'**Editeur de Propriétés**, comme indiqué ci-dessous :



The screenshot displays the CodeCharge Studio interface. The main window shows a web form titled "Rechercher Tasks" with fields for "Projet" (a dropdown menu) and "Mot clé" (a text input). Below the form is a table titled "Liste de : Tasks" with columns for "Id", "Project Name", and "Priority". The table is currently empty, showing "Aucun enregistrement". The "Messages" pane at the bottom left shows the generation process for the "tasks_list" form. The "Boîte à outils" (Toolbox) on the right is open to the "HTML" section, and the "Propriétés" (Properties) pane shows the configuration for the "List Box: s_project".

Name	Value
Name	s_project
Caption	
Source Type	Database Column
Control Source	
Connection	IntranetDB
List Source Type	Table/View
List Data Source	projects
Bound Column	project_id
Text Column	project_name
Data Type	Integer
Identifier name used in generated code	

Name (Nom) : `s_project` - ce nom sera utilisé plus tard comme un nom de variable d'entrée (*input*) pour la valeur sélectionnée.

Connection (Connexion) : `IntranetDB` - connexion à la base de données à utiliser pour retrouver les valeurs de la ListBox.

Data Source (Source de Données) : `projects` - table contenant les valeurs de la ListBox.

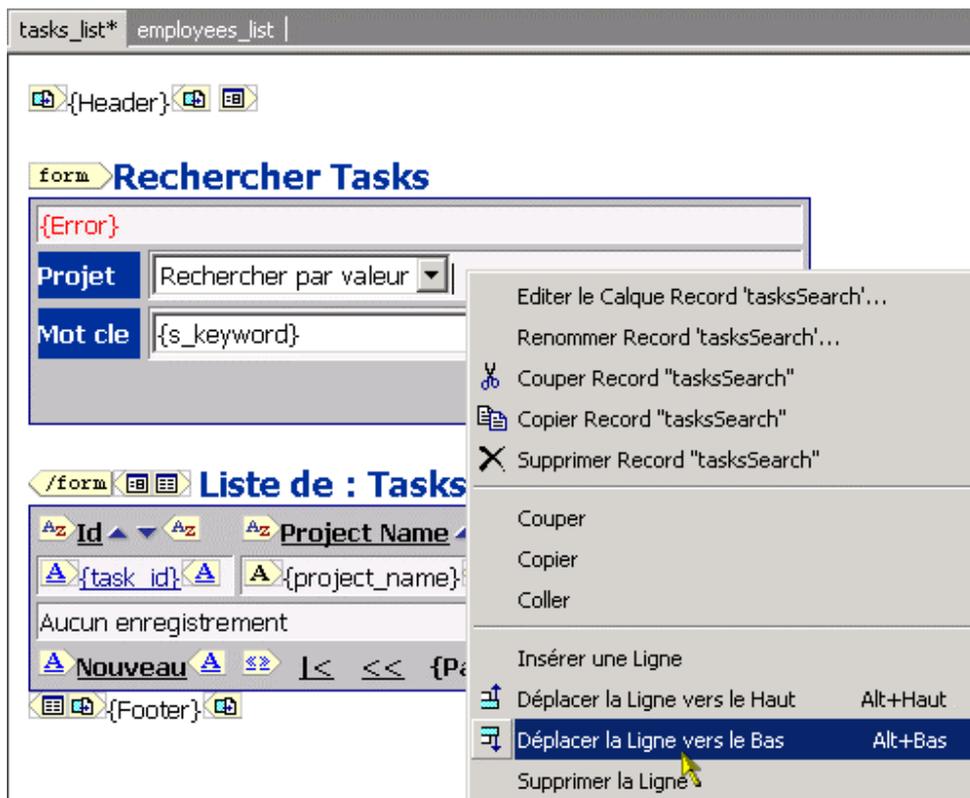
Bound Column : `project_id` - champ de la table dont la valeur sera utilisée comme paramètre de recherche.

Text Column : `project_name` - champ de la table dont la valeur sera affichée dans la ListBox.

Data Type (Type de Données) : `Integer` - type de la valeur qui sera utilisée comme paramètre de recherche (`project_id` est numérique).



Ajouter une ListBox Recherche - Déplacer une Ligne de la Table



Pour terminer, déplacez la ligne de la table contenant la ListBox vers le bas en :

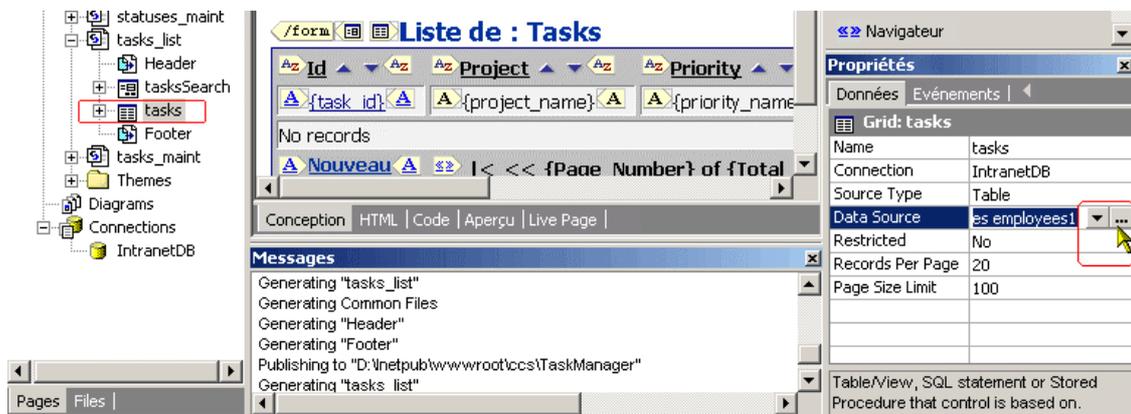
Double-cliquant à côté de la ListBox,

En sélectionnant Déplacer la ligne vers le Bas.

Vous pouvez également effectuer cette opération en positionnant le curseur à côté de la ListBox et en utilisant le raccourci clavier *Alt + Flèche bas*.

Filtrer les Enregistrements de la Grille - Sélectionner la Propriété "Where"

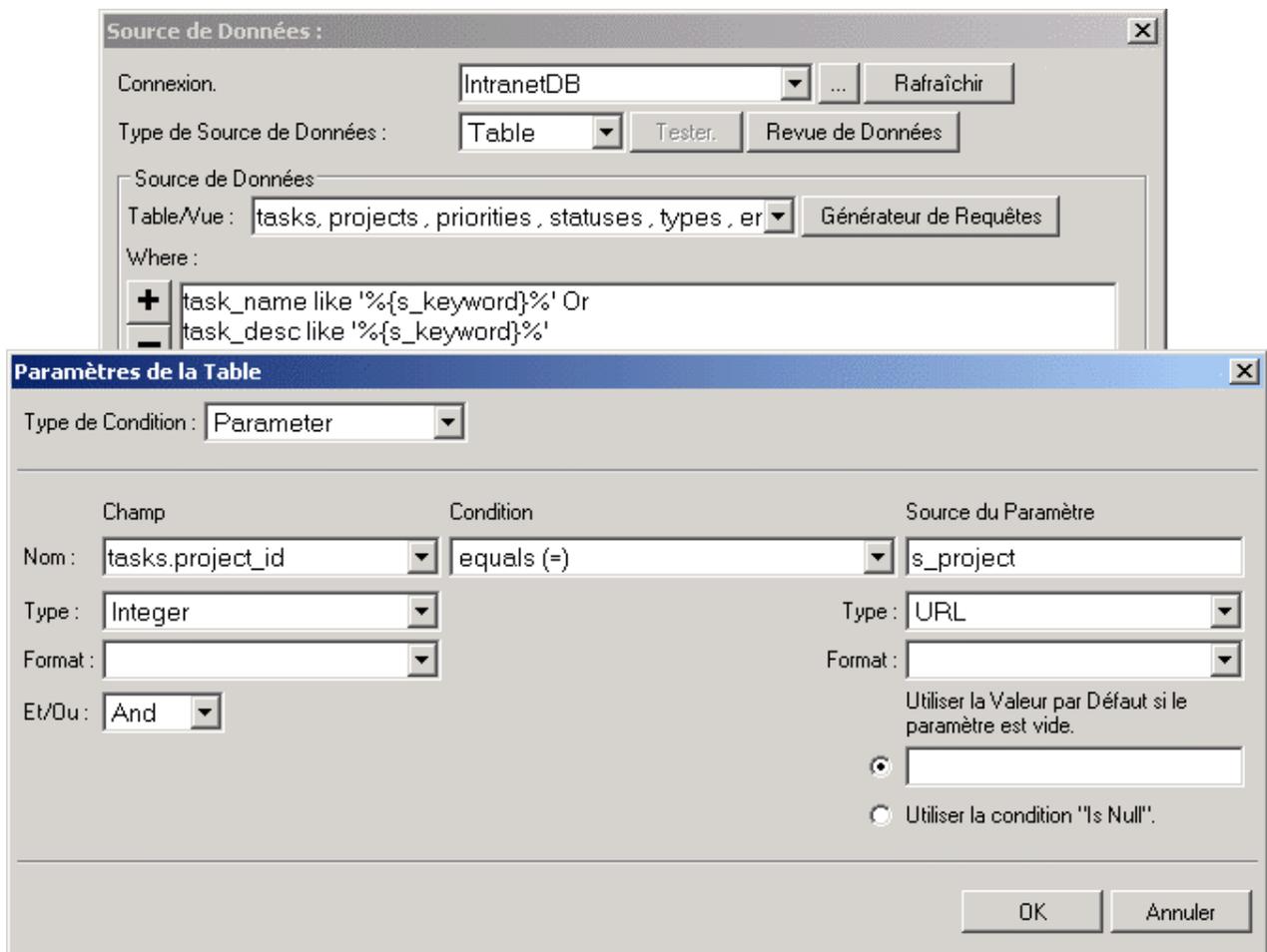
Une ListBox fonctionnelle est maintenant créée sur la page mais elle ne peut pas être utilisée pour filtrer les enregistrements de la grille tant que la grille elle-même n'utilise pas le paramètre passé à la ListBox. Vous devez alors paramétrer le critère *WHERE* au sein de la source de données de la grille.



1 - Sélectionnez la grille en cliquant n'importe où dans la légende de la grille sur la page ou en la sélectionnant dans l'Explorateur de Projet.

2 - Cliquez sur le bouton [...] de la propriété **Data Source** (Source de Données).

Filter the Grid Registrations - Add a Search Parameter



1 - Ajoutez un nouveau paramètre de recherche à la grille en :



Cliquant sur le bouton [+],

Et ensuite en spécifiant le champ `tasks.project_id` pour qu'il corresponde au paramètre `s_project`, qui est le nom de la ListBox précédemment créée.

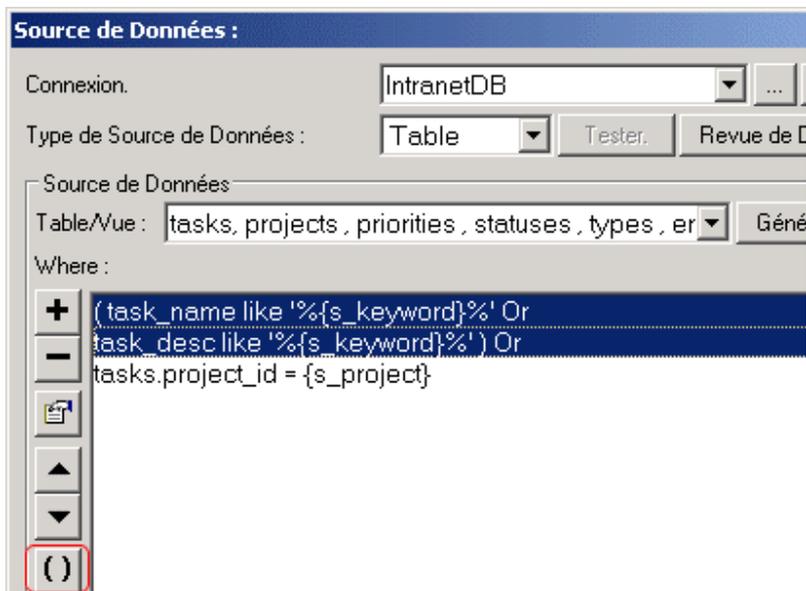
2 - Le champ **Type** pour le paramètre doit également être paramétré à *Integer* (Entier).

Le formulaire Grille recevra alors ce nouveau paramètre par l'intermédiaire de l'URL et montrera seulement les résultats correspondants.

3 - Cliquez sur **OK** une fois les informations saisies.

Filterer les Enregistrements de la Grille - Grouper les Paramètres "Where"

Dans la fenêtre Source de Données, vous verrez trois paramètres, dont deux précédemment configurés par le Générateur d'Application pour le mot clé Recherche, et un que vous venez juste d'ajouter pour la ListBox Projet. Maintenant la grille renverra des résultats si n'importe lequel des critères est rencontré. Toutefois nous voulons que les résultats soient retournés seulement si les paramètres *mot clé* et *listbox* correspondent. Par exemple, si un utilisateur saisit le mot clé "bug" et sélectionne le projet "CodeCharge", la grille ne renverra pas d'enregistrements qui sont dans le projet sélectionné ou qui contiennent le mot "bug". A la place, nous voudrions que la grille nous affiche les résultats qui sont dans le projet "CodeCharge" et qui contiennent le mot "bug".



Pour configurer la grille ainsi :

- 1 - Groupez ensemble tous les paramètres qui correspondent à "`s_keyword`".
- 2 - Sélectionnez les deux premiers paramètres par un glisser de la souris ou en maintenant la touche du clavier *Ctrl* enfoncée et en cliquant sur chacun des paramètres.
- 3 - Ensuite cliquez sur le bouton [()] pour ajouter des parenthèses autour des paramètres sélectionnés, ce qui permettra de les grouper ensemble.

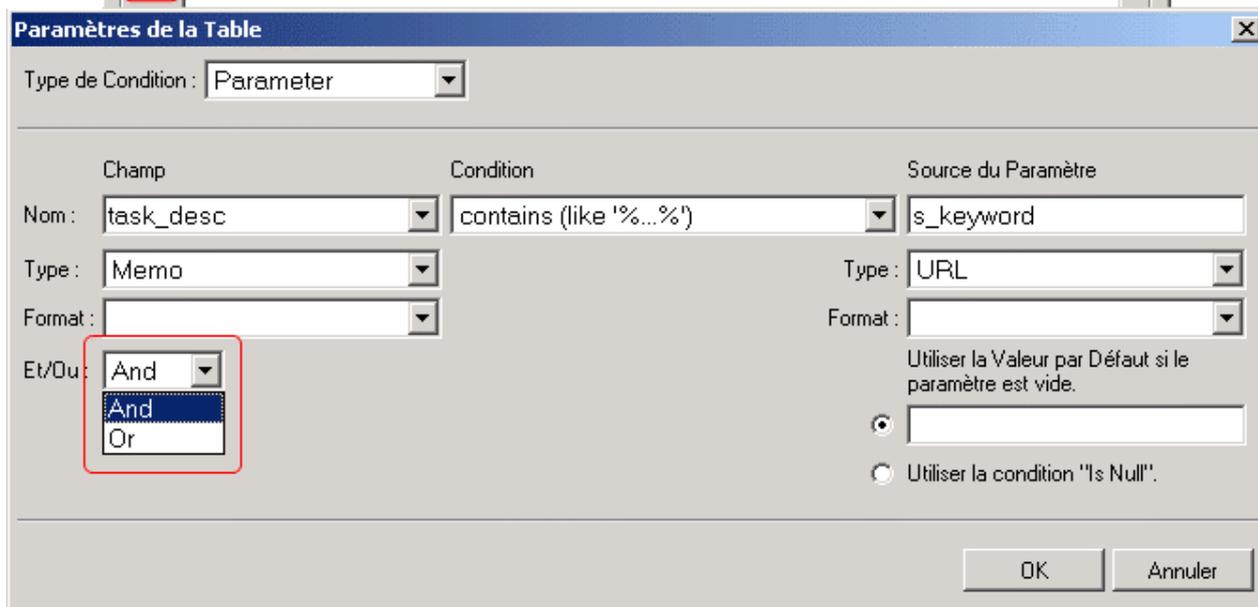
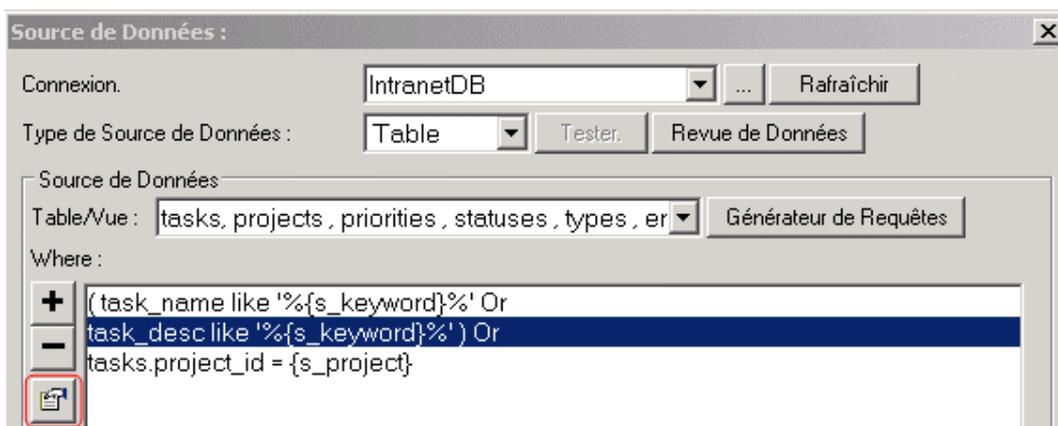
Filterer les Enregistrements de la Grille - Paramétrer l'Opérateur AND

Maintenant que tous les paramètres de recherche sont en place, il ne reste plus qu'à indiquer que le dernier paramètre mot clé doit ajouter l'opérateur "AND" de manière à ce que les paramètres de recherche se lisent comme suit :

```
(task_name like '%{s_keyword}%' or
```



```
task_desc like '%{s_keyword}%' ) and  
tasks.project_id equals (=) '{s_project}'
```



Pour configurer l'opérateur :

- 1 - Double-cliquez sur le paramètre *task_resolution* pour ouvrir la fenêtre **Paramètres de la Table**.
- 2 - Changez l'opérateur **Or** (Ou) par **And** (Et).

Visualiser la Page Fonctionnelle

Voilà, votre première page est complète. Vous pouvez rechercher et visualiser la liste des tâches, les trier, ou encore cliquer sur une tâche donnée pour obtenir plus de détails.



[Employees](#) [Priorities](#) [Projects](#) [Statutes](#) [Tasks](#)

Rechercher Tasks

Mot clé	<input type="text" value="bug"/>
Projet	<input type="text" value="CodeCharge"/>
<input type="button" value="Rechercher"/>	

Liste de : Tasks

<u>Id</u>	<u>Project</u>	<u>Priority</u>	<u>Status</u>	<u>Name</u>	<u>Assigned To</u>
2	CodeCharge	Highest	Closed	Fix ALL bugs	george
Nouveau 1 of 1					

Une fois que vous avez terminé de visualiser la page, cliquez sur un ID de tâche pour n'importe laquelle des tâches afin de tester la page Maintenance de Tâches.

Connexion au Système

Quand vous cliquez sur n'importe lequel des ID de tâche sur la page Liste des Tâches (Liste de : Tasks), vous êtes redirigé (e) vers la page Login sur laquelle vous devez saisir vos login et mot de passe. En effet, la page Enregistrement pour la table *tasks* (*tasks_maint*) est configurée de manière à ce que seuls les utilisateurs connectés puissent y accéder.

[Employees](#) [Priorities](#) [Projects](#) [Statutes](#) [Tasks](#)

Login

Login	<input type="text" value="george"/>
Password	<input type="password" value="*****"/>
<input type="button" value="Login"/>	

Saisissez **george / george** pour vous connecter en tant que George Pennington.

Cette entrée sera stockée dans une variable de session sur le serveur, rendant inutile la reconnexion au système jusqu'à l'expiration de la session utilisateur.

Accéder à la Page Maintenance d'Enregistrement

Après avoir sélectionné un ID de tâche dans la page Liste de Tâches et avoir dépassé la page Login, vous arriverez à la page Maintenance d'Enregistrement sur laquelle vous pouvez visualiser les détails de la tâche, et les mettre à jour. L'étape suivante consiste alors à personnaliser cette page.



[Employees](#) [Priorities](#) [Projects](#) [Statuses](#) [Tasks](#)

Ajouter/Editer Tasks

Project Id	CodeCharge
Priority Id	Highest
Status Id	Closed
Type Id	Question
Name	Fix ALL bugs
Desc	Staying up at night coding? Get CodeCharge, go home, get rest.
User Id Assign By	ken
User Id Assign To	george
Start Date	02.02.2003
Finish Date	03.02.2003
<input type="button" value="Confirmer"/> <input type="button" value="Supprimer"/>	

ETAPE 4

Changer des Etiquettes de Champ

Après avoir utilisé le Générateur d'Application pour générer une ébauche de l'application et avoir ajusté la page Liste de Tâches, vous devez maintenant effectuer quelques personnalisations sur la page Maintenance de Tâches (*tasks_maint*).

La première chose à faire est de changer les étiquettes de certains des champs pour qu'elles soient plus significatives.



form > **Ajouter/Editer Tasks**

{Error}

Project Id	Rechercher par valeur ▼
Priority Id	Rechercher par valeur ▼
Status Id	Rechercher par valeur ▼
Type Id	Rechercher par valeur ▼
Name	{task_name}
Desc	{task_desc}
Assign By	Rechercher par valeur ▼
Assign To	Rechercher par valeur ▼
Start Date	{task_start_da} [calendar icon] [time icon]
Finish Date	{task_finish_d} [calendar icon] [time icon]

1 - Dans la fenêtre **Explorateur de Projet**, double-cliquez sur la page `tasks_maint` pour l'ouvrir.

2 - Pour changer les étiquettes des champs, remplacez :

Project Id → Project

Priority Id → Priority

Status Id → Status

Type Id → Type

Desc → Description

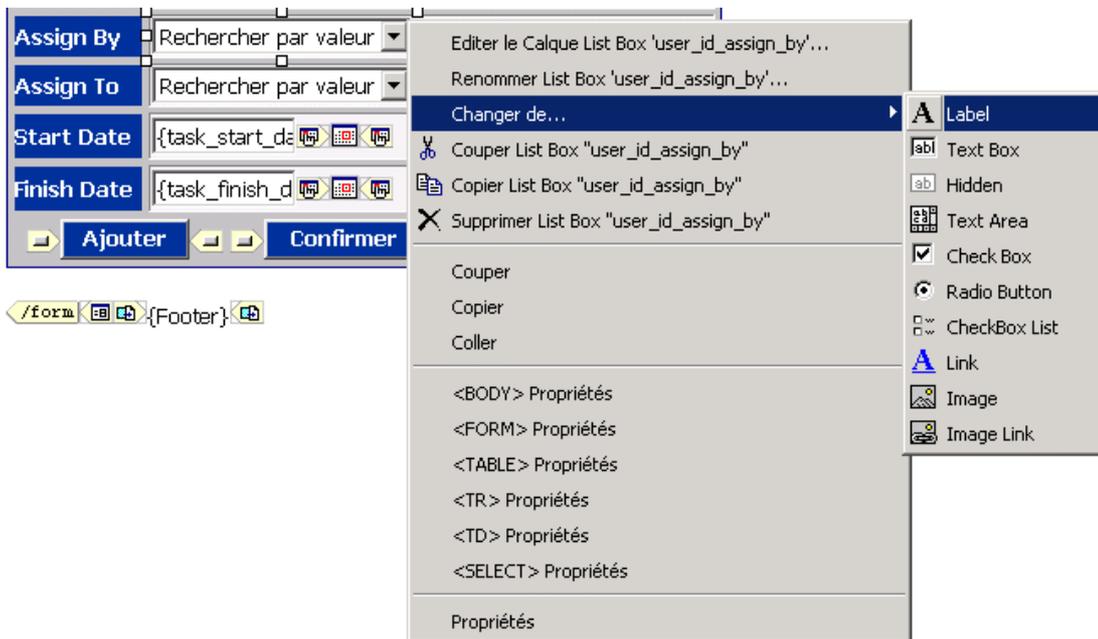
User Id Assign By → Assigned By

User Id Assign To → Assigned To

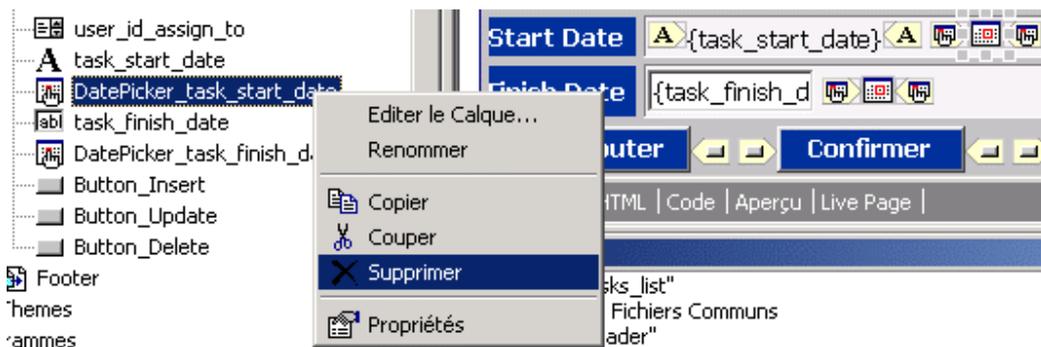
Créer des Champs Etiquette

Certains des champs sur la page Maintenance de Tâches ne nécessitent pas d'être mis à jour manuellement, mais peuvent être mis à jour automatiquement. Par exemple, si une personne crée une nouvelle tâche, son nom sera automatiquement entré en tant que créateur de la tâche, ainsi que la date et l'heure de la création.

L'étape qui suit consiste à désactiver ces champs en les convertissant en champs Etiquette.



- 1 - Faites un clic droit sur le champ *user_id_assign_by*.
- 2 - Sélectionnez : **Changer De... à Etiquette** et faites de même pour le champ *task_start_date*. Le champ *task_start_date* ayant été changé en champ Etiquette, nous n'avons plus besoin du composant Saisie de Date (*Date Picker*) qui apparaît après le champ. Le champ Etiquette ne permettant pas la saisie de donnée par l'utilisateur, le composant Saisie de Date devient inutile.
- 3 - Dans l'**Explorateur de Projet**, faites un clic droit sur le composant *DatePicker_task_start_date*.
- 4 - Sélectionnez l'option **Supprimer**.



Réorganiser les Champs Etiquette

Afin de faciliter la saisie, nous devons maintenant déplacer tous les champs Etiquette en bas du formulaire, ainsi les champs qui requièrent une saisie de données apparaîtront ensemble en haut.



form **Ajouter/Editer Tasks**

{Error}

Project	Rechercher par valeur ▾
Priority	Rechercher par valeur ▾
Status	Rechercher par valeur ▾
Type	Rechercher par valeur ▾
Name	{task_name}
Desc	{task_desc}
Assigned To	Rechercher par valeur ▾
Finish Date	{task_finish_d} 📅 🕒 🕒
Assigned By	▲ {user_id_assign_by} ▲
Start Date	▲ {task_start_date} ▲

➤ Ajouter ➤ Confirmer ➤ Supprimer

- 1 - Placez votre curseur à côté du champ Etiquette `task_start_date`.
- 2 - Appuyez en même temps sur les touches du clavier `Alt` et `Flèche du bas` pour déplacer la ligne en bas.
- 3 - Recommencez pour le champ `user_id_assign_by` pour qu'il apparaisse juste au-dessus du champ `task_start_date`.

Prévisualiser la Page Maintenance de Tâches

Félicitations ! La version de base de votre Gestionnaire de Tâches est maintenant prête.



[Employees](#) [Priorities](#) [Projects](#) [Statutes](#) [Tasks](#)

Ajouter/Editer Tasks

Project	CodeCharge
Priority	Highest
Status	Closed
Type	Question
Name	Fix ALL bugs
Desc	Staying up at night coding? Get CodeCharge, go home, get rest.
Assigned To	george
Finish Date	03.02.2003
Assigned By	3
Start Date	02.02.2003
<input type="button" value="Confirmer"/> <input type="button" value="Supprimer"/>	

1 - Prévisualisez la page fonctionnelle Maintenance de Tâches en :

Sélectionnant la page *tasks_list* dans l'**Explorateur de Projet**,

Et en sélectionnant ensuite une des tâches existantes.

2 - Vous pouvez également ajouter une nouvelle tâche en cliquant sur **Ajouter Nouveau** sous la grille *tasks* sur la page principale *tasks_list*.

Vous remarquerez que la valeur du champ *Assigned By* est montrée comme une valeur numérique. Vous devrez plus tard implémenter une relation pour afficher des noms. La section suivante explique comment le faire, ainsi que comment implémenter d'autres améliorations.

REMARQUE :

La section qui suit se réfère spécifiquement à l'implémentation en VBScript/ASP de l'exemple du Gestionnaire de Tâches. Les autres implémentations sont décrites dans les sections suivantes et selon cet ordre :

ASP et VBScript

ASP.NET (C#)

JSP

PHP

ColdFusion

Perl



EXTENSION DES FONCTIONNALITES DE L'APPLICATION AVEC LA PROGRAMMATION D'EVENEMENTS

ASP et VBScript

Extension des Fonctionnalités de l'Application avec la Programmation d'Événements (ASP et VBScript)

Vous avez certainement remarqué que jusqu'à présent vous avez construit votre Gestionnaire de Tâches sans avoir à vous préoccuper du code. En effet, CodeCharge Studio peut vous aider à construire des systèmes fonctionnels sans programmation ; toutefois la création de systèmes plus sophistiqués requiert un minimum de programmation. Heureusement, CodeCharge Studio facilite la programmation en proposant un éditeur de code de haut niveau, en plus des Événements et des Actions qui vous aident à insérer des morceaux de code pré-programmés dans le programme.

Voici les définitions d'une Action et d'un Événement :

Action

Composant de génération de code définissable par l'utilisateur, qui insère un bloc de code dans une procédure événementielle. CodeCharge Studio est fourni avec de nombreuses Actions pré-définies, qui se trouvent dans le dossier suivant : (*CCS folder*)\Components\Actions. Intérieurement, les actions consistent en du code XML et XSL pouvant être facilement personnalisé. Par exemple, une action peut être paramétrée sur un Textbox pour valider une adresse e-mail.

Procédure Événementielle

Une procédure automatiquement exécutée en réponse à un événement lancé par un programme à son exécution. Les événements sont les meilleurs emplacements pour mettre du Code Personnalisé (*Custom Code*).

Etape 1

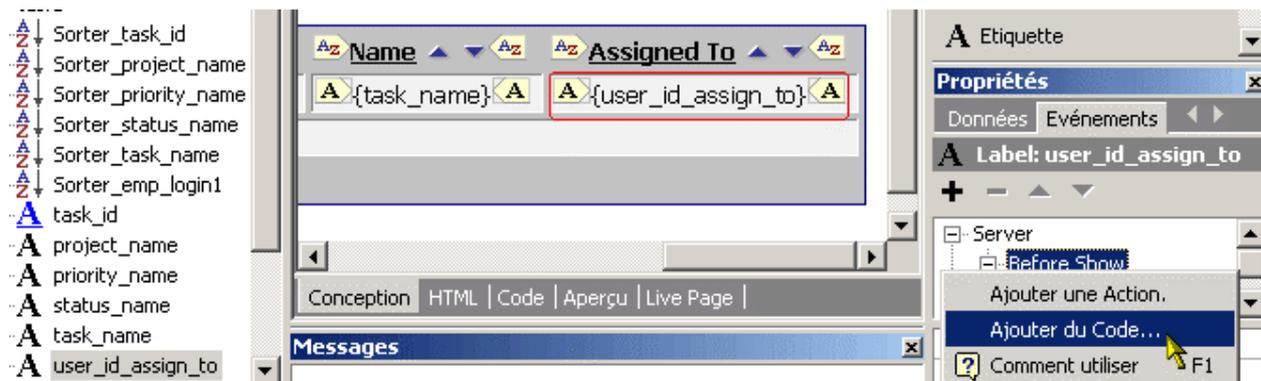
Utiliser l'Événement Before Show Row pour Changer la Couleur du Texte

Démarrons la programmation basique avec une tâche simple de changement de la couleur d'un champ grille sur notre page Liste de Tâches. Concrètement, nous allons marquer les tâches qui vous sont attribuées en affichant votre nom en bleu dans la grille.

- 1 - Ouvrez la page *tasks_list* dans l'**Explorateur de Projet**.
- 2 - Ouvrez l'arborescence de la grille *tasks*.
- 3 - Faites un clic droit sur le champ *emp_login1* et sélectionnez **Propriétés**.
- 4 - Dans l'onglet **Données**, paramétrez la valeur de la propriété **Name** (Nom) à *user_id_assign_to*.
- 5 - Dans l'**Explorateur Projet**, faites un clic droit sur le champ *task_name* et sélectionnez **Propriétés**.
- 6 - Dans l'onglet **Données**, paramétrez la valeur de la propriété **Content** (Contenu) à *HTML*.
- 7 - Sélectionnez la grille *tasks* dans l'**Explorateur de Projet**, ou cliquez n'importe où dans la légende du formulaire.
- 8 - Sélectionnez l'onglet **Événements** dans la fenêtre **Propriétés**.
- 9 - Faites un clic droit sur l'événement **Before Show Row** et sélectionnez **Ajouter du Code...** .



L'événement *Before Show Row* se déclenche dans le programme après que les valeurs champ soient attribuées, mais avant d'être générées en HTML. En ajoutant du code dans cet événement, vous pouvez modifier la valeur champ avant qu'elle ne soit montrée.



Contrôler par Programmation la Valeur Champ

Une fois que vous aurez ajouté le Code Personnalisé (*Custom Code*) à l'événement, vous verrez la fenêtre d'édition du code avec l'emplacement approprié pour entrer le nouveau code.

1 - Remplacez cette ligne de code :

```
' Write your own code here.
```

Par les lignes suivantes (ASP/VBScript) :

```
If tasks.user_id_assign_to.Value = Session("UserLogin") Then  
    tasks.task_name.Value = "<b><font color=""blue"">" & tasks.task_name.Value & "</font></b>"  
End If
```

Ce qui suit explique le fonctionnement du code ajouté ci-dessus :

```
If tasks.user_id_assign_to.Value = Session("UserLogin") then
```

C'est une condition *if* (si) qui est vraie seulement si la valeur de l'étiquette *user_id_assign_to* est égale au nom de Login de l'employé connecté. Une fois connecté au système, le programme reconnaîtra vos tâches en comparant votre nom de Login à la valeur du champ de la personne à qui est attribuée une tâche.

UserLogin est une des variables de session utilisées par les programmes générés par CodeCharge, et la variable conserve le nom de Login de la session utilisateur en cours jusqu'à son expiration.

Note :

Il s'agit des variables de session créées par défaut par CodeCharge Studio :

UserID : la valeur champ clé primaire de la session utilisateur

UserLogin : le nom de Login de l'utilisateur connecté au système

GroupID : le niveau/groupe de sécurité de l'utilisateur connecté au système

```
tasks.task_name.Value = "<b><font color=""blue"">" & tasks.task_name.Value & "</font></b>"
```

Le code du dessus est exécuté si la condition précédente *if* est rencontrée. Elle modifie la valeur du champ *task_name*. La valeur champ est remplacée par sa valeur de bases de données encapsulée dans du code HTML, qui spécifie la couleur de la police (le bleu), et ajoute un tag HTML ** pour afficher la police en gras.



Vous remarquerez que le mot *blue* est entre des double guillemets, qui seront remplacés par un guillemet simple. Dans la mesure où les guillemets marquent le début et la fin d'une chaîne de caractères, l'utilisation de double guillemets vous permet d'insérer un guillemet dans une chaîne de caractères.

De plus, vous remarquez que le code est orienté objet et vous indiquez que vous voulez attribuer une valeur au champ *task_name* dans la grille *tasks*. *Value* (Valeur) est une propriété de ce champ, qui peut à la fois être lue et modifiée.

End if

Cette ligne marque la fin de la condition *if*, aussi l'exécution du code restant n'est pas affectée par cette condition.

Prévisualiser la Page Liste de Tâches

- 1 - Sauvegardez votre projet.
- 2 - Passez en mode **Live Page** pour visualiser votre page fonctionnelle. Si aucun nom de tâche n'est mis en évidence dans la colonne "*Name*", c'est que vous n'êtes probablement pas connecté(e) au système.
- 3 - Le menu ne contenant pas encore de lien à la page Login, vous pouvez y accéder en essayant d'ouvrir une des pages restreintes, par exemple la page Maintenance de Tâches.
- 4 - Cliquez sur n'importe lequel des ID de projet, et vous devriez voir la page Login.
- 5 - Connectez-vous en tant que **george/george**, puis cliquez sur le lien **Tasks** sur le menu pour retourner à la page Liste de Tâches.

Maintenant, vous devriez voir que certaines tâches sont mises en évidence ; ces tâches correspondent aux tâches de l'utilisateur pour lequel vous avez ouvert une session.



[Employees](#) [Priorities](#) [Projects](#) [Statuses](#) [Tasks](#)

Rechercher Tasks

Mot clé	<input type="text"/>
Projet	Rechercher par valeur ▼
<input type="button" value="Rechercher"/>	

Liste de : Tasks

<u>Id</u>	<u>Project</u>	<u>Priority</u>	<u>Status</u>	<u>Name</u>	<u>Assigned To</u>
1	CodeCharge	High	On hold	Great Project needs to be greater	helen
2	CodeCharge	Highest	Closed	Fix ALL bugs	george
3	CodeCharge	High	Closed	Get ready to click	peter
4	My Project	Highest	Open	Finish My Project	ignace
5	Test Project	High	In progress	Test this project.	ken
6	CodeCharge	Highest	Open	Code with one hand.	alexander
7	Test Project	Highest	On hold	Get armed	helen
8	Test Project	Highest	Open	Write more code	ignace
9	Super Project	Highest	In progress	Code, code, code...	george
10	Test Project	Lowest	On hold	Sleep	ken
11	Super Project	Highest	Open	Have fun	alexander

Nouveau 1 of 1

Etape 2

Modifier un Champ Etiquette sur la Page Maintenance de Tâches

Maintenant procédons à une modification nécessaire sur la page Maintenance de Tâches sur laquelle vous avez dû remarqué que le champ Etiquette *Assigned By* n'affiche pas le nom de l'employé, mais l'ID, comme montré ci-dessous ; c'est parce que la table *tasks* contient seulement l'ID utilisateur, alors que la table *employees* contient les noms des utilisateurs réels.



[Employees](#) [Priorities](#) [Projects](#) [Statutes](#) [Tasks](#)

Ajouter/Editer Tasks

Project	CodeCharge
Priority	Highest
Status	Closed
Type	Question
Name	Fix ALL bugs
Desc	Staying up at night coding? Get CodeCharge, go home, get rest.
Assigned To	george
Finish Date	03.02.2003
Assigned By	3
Start Date	02.02.2003
<input type="button" value="Confirmer"/> <input type="button" value="Supprimer"/>	

Il y a plusieurs méthodes possibles pour traiter de la question. Merci de vous référer aux explications qui suivent :

1 - Créez une Requête qui contient plusieurs tables et qui peut être utilisée en tant que source de données par le formulaire Enregistrement, comme vous l'avez déjà fait avec la grille sur la page Liste de Tâches. Malheureusement, les requêtes qui contiennent plusieurs tables ne peuvent pas être mises à jour de par leur nature, et donc le formulaire Enregistrement entier peut s'arrêter de fonctionner. En d'autres termes, si vous avez indiqué que vous vouliez utiliser une requête contenant une table *tasks* et *employees* dans votre formulaire Enregistrement, alors si vous attribuez une tâche à quelqu'un d'autre, le programme ne pourra pas savoir si vous vouliez mettre à jour la table *tasks* avec le nouveau *employee_id*, ou si vous vouliez mettre à jour la table *employees* et changer le nom de l'employé.

Ainsi, si vous utilisez plusieurs tables comme source de données pour le formulaire Enregistrement, vous devrez également définir les opérations *Custom Insert*, *Custom Update* et *Custom Delete* dans les propriétés du formulaire Enregistrement pour indiquer quels sont les champs de la base de données qui devront être mis à jour avec les valeurs correspondantes saisies sur la page. Cette méthode semble effectivement représenter beaucoup d'effort juste pour afficher une valeur supplémentaire sur la page.

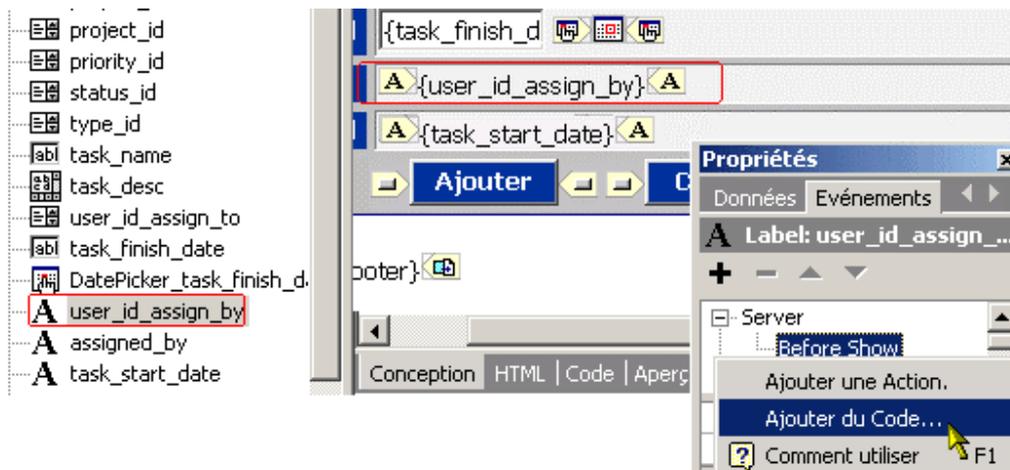
2 - Utilisez une Procédure Événementielle pour insérer du Code Personnalisé (*Custom Code*) où vous pourrez produire la valeur désirée par programmation. Cette méthode est très flexible, dans la mesure où elle vous permet d'étendre le code généré par l'ajout de votre propre code. L'étape suivante décrit en détail cette méthode.

Utiliser l'Événement Before Show pour Changer la Valeur d'une Etiquette

- 1 - En mode **Conception**, sélectionnez l'Etiquette *user_id_assign_by*,
- 2 - Dans la fenêtre **Propriétés**, cliquez sur l'onglet **Données**.
- 3 - Sélectionnez *Text* pour la propriété **Data Type** (Type de Données).
- 4 - Dans la fenêtre **Propriétés**, cliquez sur l'onglet **Événements**.
- 5 - Faites un clic droit sur l'événement **Before Show** et sélectionnez **Ajouter du Code...**



CodeCharge Studio va alors automatiquement passer en mode **Visualisation du Code**.



Une fois en mode **Visualisation du Code**, remplacez le texte suivant :

'Write your own code here.'

Par :

```
If tasks.EditMode Then
    tasks.user_id_assign_by.Value = CCDLookup("emp_name", "employees", "emp_id=" &
    DBIntranetDB.ToSQL(tasks.user_id_assign_by.Value, ccsInteger) , DBIntranetDB)
Else
    tasks.user_id_assign_by.Value = CCDLookup("emp_name", "employees", "emp_id=" &
    DBIntranetDB.ToSQL(CCGetUserID, ccsInteger), DBIntranetDB)
End If
```

Le code ci-dessus est composé des éléments suivants :

tasks : le nom du formulaire Enregistrement sur la page.

EditMode : propriété du formulaire, qui indique si l'enregistrement est édité. En fonction de la valeur de cette propriété, nous affichons soit le nom de la personne qui a initialement soumis la tâche (mode Edition), ou celui de la personne qui soumet la tâche (mode Insertion).

user_id_assign_by : le nom de l'Etiquette au sein de la Grille, et en même temps le nom du champ de la base de données qui a été utilisé pour créer cette Etiquette et qui est maintenant sa source de données.

Value : la propriété d'un objet (dans ce cas l'Etiquette), qui peut être lue et/ou modifiée.

tasks.user_id_assign_by.Value : propriété *Value* (Valeur) entièrement qualifiée, qui dit au programme quel objet y appartient. En d'autres termes, c'est la propriété *Value* qui appartient au champ *user_id_assign_by*, qui en retour appartient à la Grille *tasks*.

CCDLookup : fonction de CodeCharge qui supporte la récupération de valeur de base de données basée sur un nom de champ, un nom de table, ou une condition. Ici, cette fonction récupère le Nom de l'Employé (*emp_name*) dans la table *employees* en utilisant la condition suivante : la clé (*emp_id*) est égale à la valeur courante de l'Etiquette.

DBIntranetDB : le nom de l'objet qui définit la connexion base de données que vous voulez utiliser dans la fonction *CCDLookup*.

ToSQL : propriété de connexion qui convertit une valeur dans le format supporté par la base de données. Cette propriété requiert un paramètre qui dit si une valeur doit être convertie en un nombre (*ccsInteger*, *ccsFloat*), en du texte (*ccsText*, *ccsMemo*), en une date (*ccsDate*), ou en booléen (*ccsBoolean*). Dans ce cas, cette propriété convertit la valeur courante de l'Etiquette en un nombre qui peut être utilisé avec la fonction *CCDLookup*. Il est conseillé de toujours utiliser cette propriété avec *CCDLookup*.



`CCGetUserID` : fonction de CodeCharge qui retourne l'ID de l'utilisateur connecté au système.

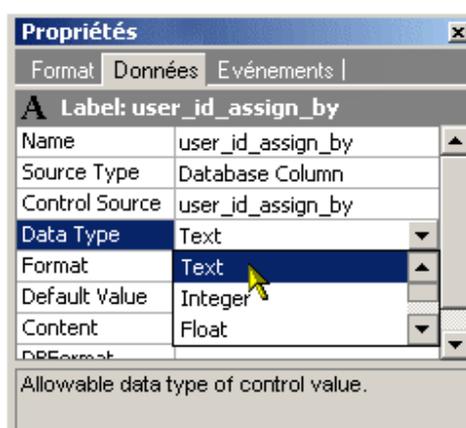
Le code se lit approximativement comme suit :

If a record is being edited : Attribue le nom de la personne qui a initialement soumis la question à l'Etiquette `user_id_assign_by`, par la recherche du nom de l'employé dans la table `employees`, en utilisant la fonction `CCDLookup`, qui utilise la connexion `IntranetDB` et la valeur de l'Etiquette `user_id_assign_by`.

If a new record is being created : Attribue un utilisateur connecté à l'Etiquette `user_id_assign_by` en récupérant son nom dans la table `employees` et en utilisant la fonction `CCDLookup`, qui utilise la connexion `IntranetDB` et la fonction `CCGetUserID` (qui permet d'obtenir l'ID de l'utilisateur connecté).

Maintenant que vous avez modifié par programmation la valeur de l'Etiquette `user_id_assign_by` pour sortir le Nom de l'Employé à la place de l'ID, vous allez devoir spécifier que ce champ est désormais un champ Texte, au lieu d'un champ Numérique.

- 1 - Cliquez sur l'onglet **Données** dans la fenêtre **Propriétés**.
- 2 - Sélectionnez `Text` en tant que **Data Type** (Type de Données).



Etape 3

Ajouter un Champ Caché "Assigned By" pour la Mise à Jour Automatique de Nouvelles Tâches

Vous avez précédemment utilisé l'événement `Before Show` pour afficher le nom de la personne à qui est attribuée une tâche. Cependant, les champs Etiquette ne peuvent pas être mis à jour de par leur nature, alors bien que le nom de l'employé soit affiché sur la page, il n'est pas écrit dans la base de données. Dans la mesure où nous voulons que la base de données enregistre le nom ou l'ID de la personne qui soumet une tâche, nous allons devoir ajouter la logique de programmation pour effectuer cela.

- 1 - Ajoutez un champ Caché à votre page depuis l'onglet **Formulaires** de la **Boîte à Outils** (*Toolbox*).

Ce type de champ n'est pas visible dans le navigateur, mais il sera utilisé pour stocker une valeur et mettre à jour la base de données.

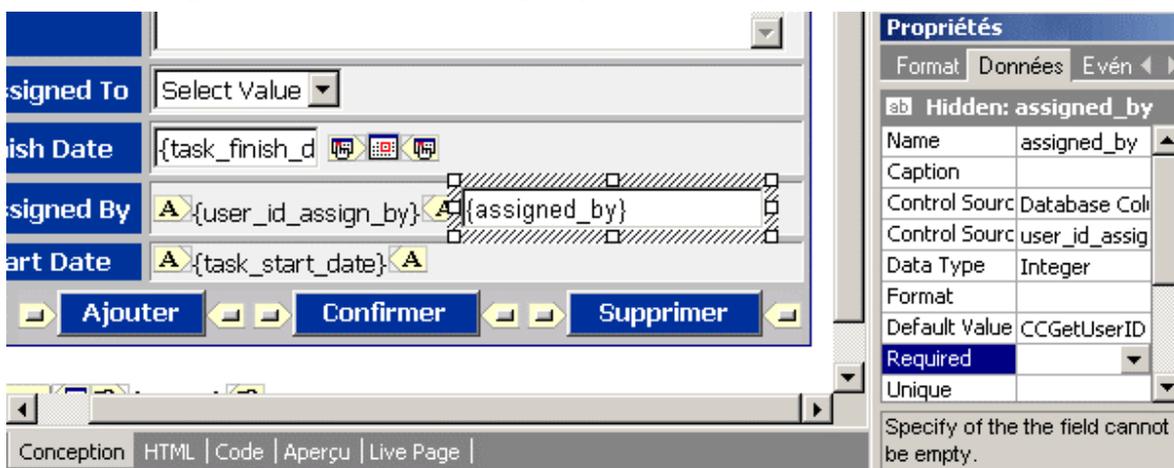
- 2 - Configurez le nouveau champ en paramétrant ses propriétés comme indiqué ci-dessous :

Name (Nom) : `assigned_by` - le nom du champ Caché qui vient d'être ajouté. Cela peut être n'importe quel nom.

Control Source : `user_id_assign_by` - le champ/colonne de la base de données qui sera utilisé(e) pour récupérer la valeur du champ et qui sera mis(e) à jour avec la nouvelle valeur en cas de changement.

Data Type (Type de Données) : `Integer` - le type de la valeur liée au Control Source. Nos ID utilisateur/employé sont numériques.

Default (Par Défaut) : `CCGetUserID - CCGetUserID` est une fonction de CodeCharge qui retrouve l'ID de l'utilisateur connecté au système. Ainsi vous pouvez simplement spécifier que vous voulez enregistrer l'ID de l'utilisateur connecté dans le champ `user_id_assign_by` pour chaque nouvelle tâche qui sera soumise.



Ajouter un Champ Caché "Date Created" au Formulaire Enregistrement

Maintenant, nous allons ajouter un autre champ Caché à votre page, qui sera utilisé pour soumettre la date et l'heure courantes dans le champ `task_start_date` dans la base de données.

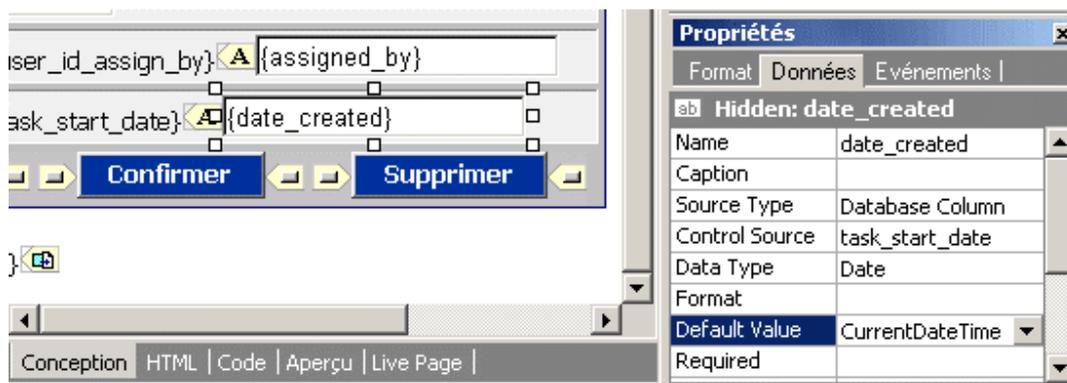
1 - Configurez le nouveau champ comme indiqué ci-dessous :

Name (Nom) : `date_created`

Control Source : `task_start_date`

Data Type (Type de Données) : Date

Default (Par Défaut) : `CurrentDateTime` - La valeur `CurrentDateTime` vous permet d'attribuer automatiquement la date et l'heure courantes aux nouvelles tâches. La propriété `Default` n'affecte pas les enregistrements existants, donc la date et l'heure des tâches existantes ne seront pas modifiées lors des mises à jour.



1 - Cliquez sur le champ `task_start_date` dans l'**Explorateur de Projet**.

2 - Dans la fenêtre **Propriétés**, paramétrez **Default Value** (Valeur Par Défaut) à `CurrentDateTime` comme vous l'avez fait avec le champ `date_created`.

C'est ainsi que le champ Etiquette peut afficher la date bien que le champ Caché ne soit pas visible pour l'utilisateur.

Tester les Champs Etiquette et Caché

Pour terminer :

1 - Passez en mode **Live Page**.



- 2 - Sélectionnez ou ajoutez une Tâche, et vous voyez alors votre Etiquette afficher le nom de la personne qui a attribué la tâche.
- 3 - La version de base de votre Gestionnaire de Tâches est désormais finalisée. Ne pas oublier de sauvegarder !

[Employees](#) [Priorities](#) [Projects](#) [Statutes](#) [Tasks](#)

Ajouter/Editer Tasks

Project	My Project
Priority	Highest
Status	Closed
Type	Task
Name	Finish my project
Desc	Implement "Assigned By" Label to show values from another table
Assigned To	alexander
Finish Date	<input type="text"/>
Assigned By	George Pennington
Start Date	08.04.2003 16:48:07
<input type="button" value="Ajouter"/>	

Etape 4

Programmation du Formulaire Enregistrement

Maintenant que vous avez créé une application simple de Gestion de Tâches, vous souhaitez certainement l'étendre pour qu'elle soit à la fois plus pratique et plus utile. Dans cette section, vous allez avoir un aperçu sur la manière d'implémenter des applications pratiques et sophistiquées par l'ajout de code et d'actions, ceci afin d'offrir des fonctionnalités supplémentaires à votre Gestionnaire de Tâches.

Vous allez apprendre à :

Envoyer des notifications d'email à la personne à qui la tâche est attribuée,

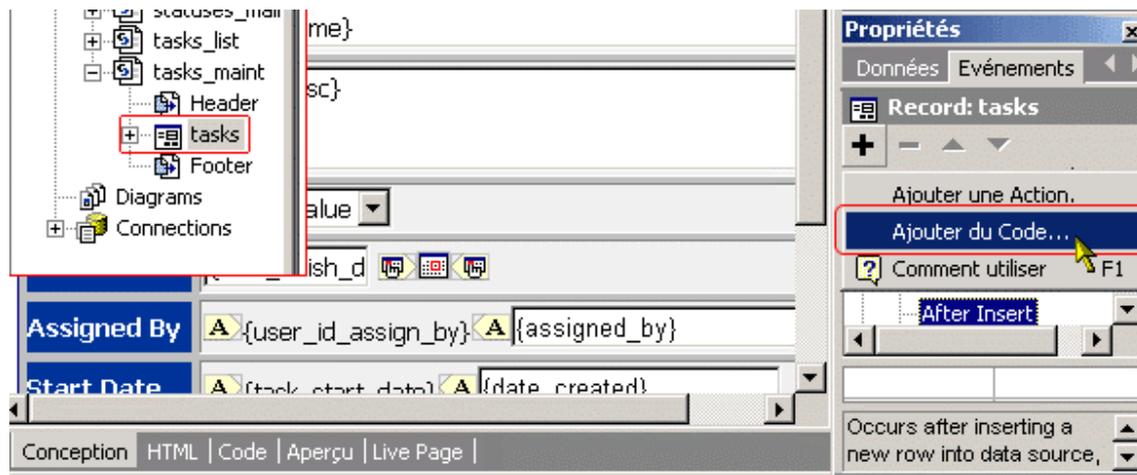
Autoriser seulement la personne à qui la tâche est attribuée à la modifier.

Ajouter du Code dans l'Événement After Insert pour Envoyer des Emails

- 1 - Sélectionnez le formulaire *tasks* dans la page *tasks_maint* en le sélectionnant depuis l'**Explorateur de Projet**, ou en cliquant n'importe où dans la légende du formulaire.
- 2 - Dans la fenêtre **Propriétés**, cliquez sur l'onglet **Événements**.
- 3 - Sélectionnez l'événement **After Insert**.
- 4 - Cliquez sur le bouton **[+]**.



5 - Sélectionnez **Ajouter du Code...**



Une fois en mode **Visualisation du Code**, remplacez le commentaire généré :

' Write your own code here.

Par le code ci-dessous :

```
Dim Mailer
Set Mailer = Server.CreateObject("Persits.MailSender")

Mailer.From = CCDLookUp("email", "employees", "emp_id=" & _
    DBIntranetDB.ToSQL(CCGetUserID, ccsInteger), DBIntranetDB)
Mailer.FromName = CCDLookUp("emp_name", "employees", "emp_id=" & _
    DBIntranetDB.ToSQL(CCGetUserID, ccsInteger), DBIntranetDB)
Mailer.AddAddress CCDLookUp("email", "employees", "emp_id=" & _
    DBIntranetDB.ToSQL(tasks.user_id_assign_to.Value, ccsInteger), DBIntranetDB)
Mailer.Host = "mysmtphost.com"
Mailer.IsHTML = True
Mailer.Subject = "New task for you"
Mailer.Body = "The following task was submitted:<br><br>" & _
    "Task ID: " & CCDLookUp("max(task_id)", "tasks", "user_id_assign_by=" & _
    DBIntranetDB.ToSQL(CCGetUserID, ccsInteger), DBIntranetDB) & _
    "<br><br>" & tasks.task_desc.Text

Mailer.Send

set Mailer = Nothing
```

Comme vous avez dû vous en rendre compte, le code ci-dessus envoie des emails aux utilisateurs à qui de nouvelles tâches sont attribuées. Voici des informations supplémentaires que vous devez connaître :

1 - Le code du dessus nécessite que vous installiez sur votre serveur le composant Email "ASPEmail", que vous pouvez obtenir gratuitement sur <http://www.aspemail.com/> Il y a beaucoup d'autres composants email et vous pouvez modifier le programme du dessus en vous référant à la documentation concernant le composant que vous avez choisi d'utiliser.



2 - Vous devez remplacer le paramètre "mysmtphost.com" par un serveur SMTP que vous êtes autorisé à utiliser. C'est généralement le même serveur que celui configuré comme "Outgoing Mail Server (SMTP)" dans votre messagerie email client (Microsoft Outlook ou Outlook Express).

Ce qui suit est une explication du code du dessus :

`Dim Mailer :`

Définit l'objet Mailer, qui plus tard initialisera le composant ASPEmail.

`Set Mailer = Server.CreateObject("Persits.MailSender")`

Crée l'objet Mailer et initialise le composant ASPEmail.

`Mailer.From = CCDLookUp("email", "employees", "emp_id=" & DBIntranetDB.ToSQL(CCGetUserID, ccsInteger), DBIntranetDB)`

Paramètre l'adresse email *From* (*Expéditeur*) à la valeur du champ *email* dans la table *employees* où *emp_id* correspond à l'utilisateur connecté. La fonction *CCDLookUp* est utilisée pour récupérer une valeur de base de données, tandis que *CCGetUserID* retrouve l'ID de la connexion utilisateur en cours.

`Mailer.FromName = CCDLookUp("emp_name", "employees", "emp_id=" & DBIntranetDB.ToSQL(CCGetUserID, ccsInteger), DBIntranetDB)`

Paramètre le nom *From* (*Expéditeur*) à la valeur du champ *emp_name* pour l'utilisateur en cours.

`Mailer.AddAddress CCDLookUp("email", "employees", "emp_id=" & DBIntranetDB.ToSQL(tasks.user_id_assign_to.Value, ccsInteger), DBIntranetDB)`

Paramètre l'adresse email *To* (*Destinataire*) à l'email de la personne à qui est attribuée la tâche. La fonction *CCDLookUp* est utilisée ici pour récupérer l'adresse email appropriée.

`Mailer.Host = "mysmtphost.com"`

Spécifie le serveur SMTP par lequel l'email sera envoyé (remplacez cette valeur par un hôte SMTP que vous êtes autorisé à utiliser).

`Mailer.IsHTML = True`

Spécifie que l'email sera envoyé au format HTML (et non pas au format texte brut).

`Mailer.Subject = "New task for you"`

L'objet de l'email à envoyer.

`Mailer.Body = "The following task was submitted:

" & "Task ID: " & CCDLookUp("max(task_id)", "tasks", "user_id_assign_by=" & DBIntranetDB.ToSQL(CCGetUserID, ccsInteger), DBIntranetDB) & "

" & tasks.task_desc.Text`



Le corps de l'email qui comprend la description de la tâche et de l'ID de tâche. Le dernier ID de tâche insérée peut être obtenu en utilisant différentes méthodes avec plusieurs bases de données. Malheureusement, Microsoft Access ne supporte pas la récupération du dernier enregistrement inséré, aussi vous devrez utiliser la fonction *CCDLookUp* pour récupérer le plus grand ID de tâche soumis par l'utilisateur connecté (en supposant que les ID de tâches soient créés par incrémentation).

`Mailer.Send`

Envoie l'email.

`set Mailer = Nothing`

Se débarrasse de l'objet Mailer pour libérer des ressources matérielles.

Utiliser l'Événement After Update pour Envoyer des Emails

Vous avez précédemment ajouté le code nécessaire qui envoie une notification d'email à la personne à qui la tâche est attribuée, une fois l'enregistrement d'une nouvelle tâche effectuée dans le système. Maintenant, il nous faut implémenter une fonctionnalité similaire dans l'événement *After Update* pour informer la personne, à qui la tâche est attribuée, lorsqu'une tâche existante est mise à jour et réattribuée à quelqu'un d'autre.

- 1 - Cliquez sur le formulaire *tasks* dans l'**Explorateur de Projet**, ou cliquez n'importe où dans la légende du formulaire.
- 2 - Dans la fenêtre **Propriétés**, sélectionnez l'onglet **Événements**.
- 3 - Ajoutez le **Custom Code** (Code Personnalisé) dans l'événement **After Update** :

```
Dim Mailer
```

```
If CCGetUserID <> tasks.user_id_assign_to.Value then
    Set Mailer = Server.CreateObject("Persits.MailSender")
    Mailer.From = CCDLookUp("email", "employees", "emp_id=" & _
        DBIntranetDB.ToSQL(CCGetUserID, ccsInteger), DBIntranetDB)
    Mailer.FromName = CCDLookUp("emp_name", "employees", "emp_id=" & _
        DBIntranetDB.ToSQL(CCGetUserID, ccsInteger), DBIntranetDB)
    Mailer.AddAddress CCDLookUp("email", "employees", "emp_id=" & _
        DBIntranetDB.ToSQL(tasks.user_id_assign_to.Value, ccsInteger), DBIntranetDB)
    Mailer.Host = "mysmtphost.com"
    Mailer.IsHTML = True
    Mailer.Subject = "A task was assigned to you"
    Mailer.Body = "The following task was assigned to you:<br><br>" & _
        "Task ID: " & Request.QueryString("task_id")& _
        "<br><br>" & tasks.task_desc.Text
```

```
Mailer.Send
```

```
set Mailer = Nothing
```

```
End if
```

Les principales différences entre le code ci-dessus et celui utilisé dans l'événement *After Insert* sont les suivantes :

- 1 - Une condition *if* a été ajoutée pour envoyer un email seulement si un utilisateur attribue une tâche à une autre personne.



2 - `task_id` est récupéré depuis l'URL utilisant la fonction `Request.QueryString`. Nous pouvons utiliser cette méthode parce que des tâches peuvent être mises à jour seulement si l'utilisateur est arrivé à la page en cours par une URL qui contient un ID de tâche à mettre à jour. Une telle URL ressemble à : http://localhost/TaskManager/tasks_maint.jsp?task_id=9

Test de l'Envoi d'Email

Avant de tester l'envoi d'email :

1 - Vous devez ajouter de nouveaux utilisateurs à votre base de données avec des adresses email correctes, ou modifier les utilisateurs existants en changeant leur adresse email :

A - Vous pouvez le faire en ouvrant la base de données *Intranet.mdb* qui se trouve dans votre répertoire *Projet*.

B - Sinon, vous pouvez utiliser le Gestionnaire de Tâches. Allez à la page Employés pour y visualiser les emails utilisateur et les modifier.

2 - Une fois vos utilisateurs configurés avec le test d'emails, sauvegardez votre projet et passez en mode **Live Page** pour tester votre système.

Remarque : Vous aurez besoin de Microsoft Access 2000 ou une version supérieure pour éditer manuellement le fichier base de données. Si votre code email fonctionne correctement, vous devez alors vous retrouver sur la page Liste de Tâches après l'ajout ou la modification d'une tâche, et la personne à qui la tâche a été attribuée doit recevoir un email.

Etape 5

Implémenter la Sécurité d'Enregistrement dans l'Événement After Initialize

Votre Gestionnaire de Tâches est désormais pratiquement complet, seule une fonctionnalité pouvant s'avérer importante manque, à savoir la Sécurité. Dans l'état actuel du gestionnaire, n'importe qui peut modifier et supprimer n'importe quelle tâche. Vous voudrez certainement limiter l'accès pour que seuls les employés autorisés puissent mettre à jour leurs tâches. Il y a plusieurs façons de procéder, et nous allons examiner plusieurs d'entre elles.

1 - Cliquez sur la page *tasks_maint* dans l'**Explorateur de Projet**.

2 - Dans la fenêtre **Propriétés**, sélectionnez l'onglet **Événements**.

3 - Ajoutez **Custom Code** (Code Personnalisé) à l'événement **After Initialize** de la page. Procédez comme suit :

Une fois en mode **Visualisation de Code**, remplacez le commentaire généré :

```
' Write your own code here.
```

Par ce code :

```
Dim current_task
current_task = CInt(Request.QueryString("task_id"))
If current_task <> 0 and CCGetUserID <> CCDLookup("user_id_assign_to", "tasks", "task_id=" &
    DBIntranetDB.ToSQL(current_task, ccsInteger), DBIntranetDB) Then
    tasks.Visible = False
    ' Redirect = "tasks_list.asp"
    ' tasks.UpdateAllowed = False
    ' tasks.DeleteAllowed = False
End if
```

Le code ci-dessus vous permet de tester les méthodes suivantes pour implémenter la sécurité d'enregistrement :

Ne pas montrer la Tâche (formulaire Enregistrement) sur la page si la tâche sélectionnée n'appartient pas à l'utilisateur connecté. Un utilisateur non autorisé verra une page blanche.



Vous pouvez cacher n'importe quel formulaire sur une page en lui attribuant une valeur *False* (Faux) pour la propriété *Visible* (visible) du formulaire. Le code *current_task <> 0* dans la condition *if* précise que le code doit seulement être exécuté si un utilisateur essaie de modifier une tâche existante et qui ne lui est pas attribuée. La condition *if* s'assure aussi que tous les utilisateurs peuvent créer de nouvelles tâches. Vous pouvez tester cette fonctionnalité en insérant le code du dessus dans l'événement, puis en passant en mode **Live Page** et en essayant de modifier une tâche qui ne vous est pas attribuée... dans ce cas vous verrez une page vide. Bien qu'une telle fonctionnalité ne soit pas très utile, elle illustre comment cacher des formulaires sur une page. Ajoutez un autre formulaire Enregistrement à votre page qui ne puisse pas être mis à jour et qui ait seulement des champs Etiquette qui montrent les informations. Une fois que vous avez deux formulaires sur la page, vous pouvez cacher chaque formulaire par programmation en utilisant des critères d'exclusion.

Rediriger des utilisateurs non autorisés sur une autre page. Seuls les utilisateurs à qui sont attribuées des tâches peuvent visualiser la page.

Vous pouvez mettre en application et tester cette fonctionnalité en modifiant légèrement le code du dessus comme montré ci-dessous :

```
Dim current_task

current_task = CInt(Request.QueryString("task_id"))

If current_task <> 0 and CCGetUserID <> CCDLookUp("user_id_assign_to", "tasks", "task_id=" &
    DBIntranetDB.ToSQL(current_task, ccsInteger), DBIntranetDB) Then
    ' tasks.Visible = False

    Redirect = "tasks_list.asp"

    ' tasks.UpdateAllowed = False

    ' tasks.DeleteAllowed = False

End if
```

Le code ci-dessus montre que vous devez commenter la ligne active précédemment, et activer la ligne qui commence avec *Redirect*. *Redirect* est une variable utilisée par CodeCharge Studio pour déterminer si la page en cours doit être redirigée vers une autre page, par exemple dans le cas où un utilisateur n'est pas connecté. Cette variable peut être utilisée seulement sur les pages qui ont un accès restreint et qui requièrent l'ouverture d'une session. Vous pouvez simplement attribuer la page de destination à la variable *Redirect* et la page sera alors automatiquement redirigée. Vous pouvez tester cette fonctionnalité en modifiant le code comme montré, puis en passant en mode **Live Page**, et en essayant de modifier une tâche qui ne vous est pas attribuée.

C# et VB.Net

Extension des Fonctionnalités de l'Application avec la Programmation d'Événements (C# et VB.Net)

Vous avez certainement remarqué que jusqu'à présent vous avez construit votre Gestionnaire de Tâches sans avoir à vous préoccuper du code. En effet, CodeCharge Studio peut vous aider à construire des systèmes fonctionnels sans programmation ; toutefois la création de systèmes plus sophistiqués requiert un minimum de programmation. Heureusement, CodeCharge Studio facilite la programmation en proposant un éditeur de code de haut niveau, en plus des Événements et des Actions qui vous aident à insérer des morceaux de code pré-programmés dans le programme.

Voici les définitions d'une Action et d'un Événement :

Action

Composant de génération de code définissable par l'utilisateur, qui insère un bloc de code dans une procédure événementielle. CodeCharge Studio est fourni avec de nombreuses Actions pré-définies, qui se trouvent dans le dossier suivant : (*CCS folder*)\Components\Actions. Intérieurement, les actions consistent en du code XML et XSL pouvant être facilement personnalisé. Par exemple, une action peut être paramétrée sur un Textbox pour valider une adresse e-mail.

Procédure Événementielle



CodeCharge Studio - Guide de Démarrage Rapide (Didacticiel)

47

Une procédure automatiquement exécutée en réponse à un événement lancé par un programme à son exécution. Les événements sont les meilleurs emplacements pour mettre du Code Personnalisé (*Custom Code*).



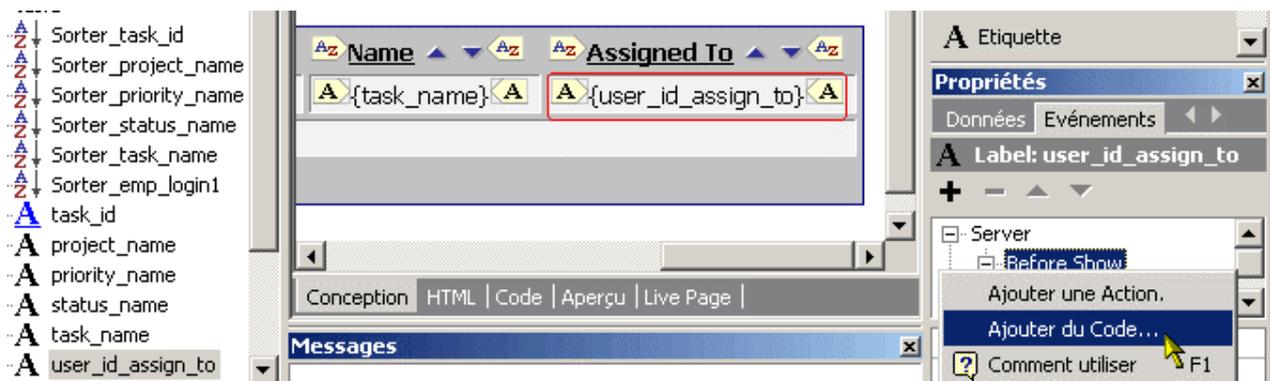
Etape 1

Utiliser l'Événement Before Show Row pour Changer la couleur du Texte

La première amélioration va donc consister à changer la couleur d'un champ grille sur la page *Task List*. Concrètement, nous allons faire ressortir les tâches attribuées à l'utilisateur ayant une session ouverte, en les affichant en bleu.

- 1 - Ouvrez la page *tasks_list* dans l'**Explorateur de Projet**.
- 2 - Ouvrez l'arborescence de la grille *tasks*.
- 3 - Faites un clic droit sur le champ *emp_login1* et sélectionnez **Propriétés**.
- 4 - Dans l'onglet **Données**, paramétrez la valeur de la propriété **Name** (Nom) à *user_id_assign_to*.
- 5 - Dans l'**Explorateur Projet**, faites un clic droit sur le champ *task_name* et sélectionnez **Propriétés**.
- 6 - Dans l'onglet **Données**, paramétrez la valeur de la propriété **Contenu** (Contenu) à *HTML*.
- 7 - Sélectionnez la grille *tasks* dans l'**Explorateur de Projet**, ou cliquez n'importe où dans la légende du formulaire.
- 8 - Sélectionnez l'onglet **Événements** dans la fenêtre **Propriétés**.
- 9 - Faites un clic droit sur l'événement **Before Show Row** et sélectionnez **Ajouter du Code...**

L'événement *Before Show Row* se déclenche dans le programme après que les valeurs champs soient attribuées, mais avant d'être générées en HTML. En ajoutant du code dans cet événement, vous pouvez modifier la valeur champ avant qu'elle ne soit montrée.



Contrôler par Programmation la Valeur Champ

Une fois votre Code Personnalisé (*Custom Code*) ajouté à l'événement, vous verrez alors la fenêtre d'édition du code avec l'emplacement approprié pour entrer le nouveau code.

Remplacez cette ligne de code :

C#

```
// Write your own code here.
```

VB.Net

```
' Write your own code here.
```

Par les lignes suivantes :

C#



```
if((DataItem.user_id_assign_to.Value).ToString() == DBUtility.UserLogin )
{
    taskstask_name.Text = "<b><font color='blue'>" + (DataItem.task_name.Value).ToString() + "</b></font>";
}
```

VB.Net

```
If tasksuser_id_assign_to.Text = DBUtility.UserLogin Then
    taskstask_name.Text = "<b><font color='blue'>" & taskstask_name.Text & "</b></font>"
End if
```

Ce qui suit explique comment le code ajouté ci-dessus fonctionne :

La condition *if* (si) est *true* (vraie) seulement si le champ grille *task_name* (contenant le nom de login de l'utilisateur à qui la tâche est attribuée) est égale au nom de login de l'employé connecté au gestionnaire. Donc, une fois connecté au gestionnaire, le programme reconnaîtra vos tâches en comparant votre nom à celui de la personne à qui est attribuée chaque tâche.

C#

```
taskstask_name.Text = "<b><font color='blue'>" + (DataItem.task_name.Value).ToString() + "</b></font>";
```

VB.Net

```
taskstask_name.Text = "<b><font color='blue'>" & taskstask_name.Text & "</b></font>"
```

Ce code est exécuté si la condition précédente *if* est rencontrée. Il modifie la valeur de l'Etiquette *taskstask_name*. La valeur champ est remplacée par la valeur nom de login encapsulée dans du code HTML, qui spécifie la couleur de la police comme le bleu, et ajoute un tag HTML ** pour obtenir une police en gras.

Prévisualiser la Page Liste de Tâches

- 1 - Sauvegardez votre projet.
- 2 - Passez en mode **Live Page** pour visualiser votre page fonctionnelle. Si aucun nom de tâche n'est mis en évidence dans la colonne "*Name*", c'est que vous n'êtes probablement pas connecté(e) au système.
- 3 - Le menu ne contenant pas encore de lien à la page Login, vous pouvez y accéder en essayant d'ouvrir une des pages restreintes, par exemple la page Maintenance de Tâches.
- 4 - Cliquez sur n'importe lequel des ID projet, et vous devriez voir la page Login.
- 5 - Connectez-vous en tant que **george/george**, puis cliquez sur le lien **Tasks** sur le menu pour retourner à la page Liste de Tâches.

Maintenant, vous devriez voir que certaines tâches sont mises en évidence ; ces tâches correspondent aux tâches de l'utilisateur pour lequel vous avez ouvert une session.



[Employees](#) [Priorities](#) [Projects](#) [Statutes](#) [Tasks](#)

Rechercher Tasks

Mot clé	<input type="text"/>
Projet	Rechercher par valeur ▼
<input type="button" value="Rechercher"/>	

Liste de : Tasks

<u>Id</u>	<u>Project</u>	<u>Priority</u>	<u>Status</u>	<u>Name</u>	<u>Assigned To</u>
1	CodeCharge	High	On hold	Great Project needs to be greater	helen
2	CodeCharge	Highest	Closed	Fix ALL bugs	george
3	CodeCharge	High	Closed	Get ready to click	peter
4	My Project	Highest	Open	Finish My Project	ignace
5	Test Project	High	In progress	Test this project.	ken
6	CodeCharge	Highest	Open	Code with one hand.	alexander
7	Test Project	Highest	On hold	Get armed	helen
8	Test Project	Highest	Open	Write more code	ignace
9	Super Project	Highest	In progress	Code, code, code...	george
10	Test Project	Lowest	On hold	Sleep	ken
11	Super Project	Highest	Open	Have fun	alexander

Nouveau 1 of 1

Etape 2

Modifier un Champ Etiquette sur la Page Maintenance de Tâches

Maintenant procédons à une modification nécessaire sur la page Maintenance de Tâches sur laquelle vous avez dû remarqué que le champ Etiquette *Assigned By* n'affiche pas le nom de l'employé, mais l'ID, comme montré ci-dessous ; c'est parce que la table *tasks* contient seulement l'ID utilisateur, alors que la table *employees* contient les noms des utilisateurs réels.



[Employees](#) [Priorities](#) [Projects](#) [Statutes](#) [Tasks](#)

Ajouter/Editer Tasks

Project	CodeCharge
Priority	Highest
Status	Closed
Type	Question
Name	Fix ALL bugs
Desc	Staying up at night coding? Get CodeCharge, go home, get rest.
Assigned To	george
Finish Date	03.02.2003
Assigned By	3
Start Date	02.02.2003
<input type="button" value="Confirmer"/> <input type="button" value="Supprimer"/>	

Il y a plusieurs méthodes possibles pour traiter de la question. Merci de vous référer aux explications qui suivent :

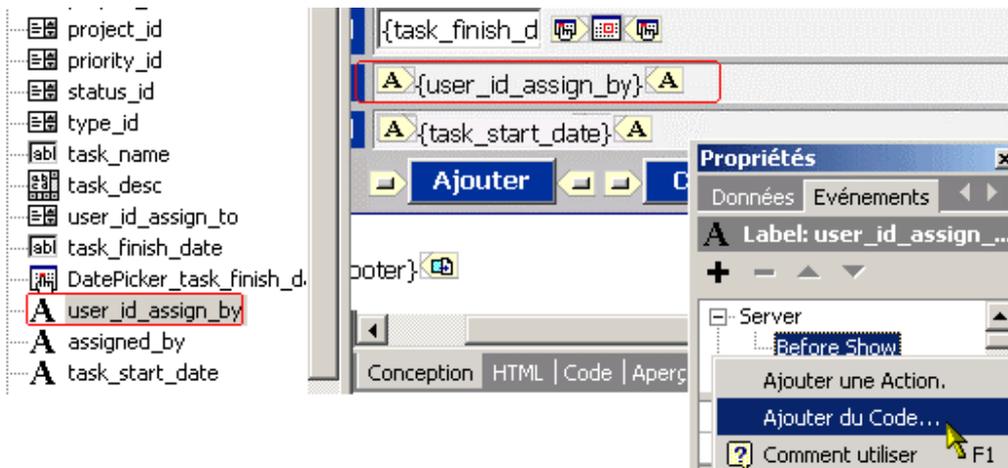
1 - Créez une Requête qui contient plusieurs tables et qui peut être utilisée en tant que source de données par le formulaire enregistrement, comme vous l'avez déjà fait avec la grille sur la page Liste de Tâches. Malheureusement, les requêtes qui contiennent plusieurs tables ne peuvent pas être mises à jour de par leur nature, et donc le formulaire Enregistrement entier peut s'arrêter de fonctionner. En d'autres termes, si vous avez indiqué que vous vouliez utiliser une requête contenant une table *tasks* et *employees* dans votre formulaire Enregistrement, alors si vous attribuez une tâche à quelqu'un d'autre, le programme ne pourra pas savoir si vous vouliez mettre à jour la table *tasks* avec le nouveau *employee_id*, ou si vous vouliez mettre à jour la table *employees* et changer le nom de l'employé.

Ainsi, si vous utilisez plusieurs tables comme source de données pour le formulaire Enregistrement, vous devrez également définir les opérations *Custom Insert*, *Custom Update* et *Custom Delete* dans les propriétés du formulaire Enregistrement pour indiquer quels champs de la base de données devront être mis à jour avec les valeurs correspondantes saisies sur la page. Cette méthode semble effectivement représenter beaucoup d'effort juste pour afficher une valeur supplémentaire sur la page.

2 - Utilisez une Procédure Événementielle pour insérer du Code Personnalisé (*Custom Code*) où vous pourrez produire la valeur désirée par programmation. Cette méthode est très flexible, dans la mesure où elle vous permet d'étendre le code généré par l'ajout de votre propre code. L'étape suivante décrit en détail cette méthode.

Utiliser l'Événement Before Show pour Changer la Valeur d'une Etiquette

- 1 - En mode **Conception**, sélectionnez l'Etiquette *user_id_assign_by*,
 - 2 - Dans la fenêtre **Propriétés**, cliquez sur l'onglet **Données**.
 - 3 - Sélectionnez *Text* pour la propriété **Data Type** (Type de Données).
 - 4 - Dans la fenêtre **Propriétés**, cliquez sur l'onglet **Événements**.
 - 5 - Faites un clic droit sur l'événement **Before Show** et sélectionnez **Ajouter du Code...** .
- CodeCharge Studio va alors automatiquement passer en mode **Visualisation du Code**.



Une fois en mode **Visualisation du Code**, remplacez le texte suivant :

C#

```
// Write your own code here.
```

VB.Net

```
` Write your own code here.
```

Par :

C#

```
if (IsInsertMode)
{
    SqlCommand userNameCmd = new SqlCommand("SELECT emp_name FROM" +
        "employees WHERE emp_id=" + DBUtility.UserId.ToString(),
        Settings.IntranetDBDataAccessObject);
    tasksuser_id_assign_by.Text = userNameCmd.ExecuteScalar().ToString();
}
else
{
    SqlCommand userNameCmd = new SqlCommand( "SELECT emp_name FROM" +
        "employees WHERE emp_id=" + item.user_id_assign_by.Value.ToString(),
        Settings.IntranetDBDataAccessObject);
    tasksuser_id_assign_by.Text = userNameCmd.ExecuteScalar().ToString();
}
```



VB.Net

```
If IsInsertMode = True Then
    tasksuser_id_assign_by.Text = Convert.ToString(Settings.IntranetDBDataAccessObject.ExecuteScalar(
        "SELECT emp_name FROM employees WHERE emp_id=" & DBUtility.UserId ))
Else
    tasksuser_id_assign_by.Text = Convert.ToString(Settings.IntranetDBDataAccessObject.ExecuteScalar(
        "SELECT emp_name FROM employees WHERE emp_id=" &
        Convert.ToString( tasksuser_id_assign_by.Text)))
End If
```

Dans le morceau de code ci-dessus, nous avons un bloc *if* qui vérifie la valeur de la variable Booléenne *IsInsertMode*. Cette variable est générée par CodeCharge Studio dans l'événement *Before Show* des formulaires Enregistrement. Puisque le même formulaire Enregistrement CodeCharge Studio peut être utilisé pour mettre à jour les enregistrements, pour supprimer les enregistrements comme pour insérer de nouveaux enregistrements, cette variable est utilisée pour déterminer l'état du formulaire Enregistrement au *runtime* (à l'exécution). Si cette variable est *true* (vraie), alors le formulaire Enregistrement insère un nouvel enregistrement, sinon elle est soit mise à jour, soit supprimée.

Les blocs *if* et *else* définissent un objet *SqlCommand*, qui comme indiqué précédemment vous aide à exécuter des commandes de la base de données. La requête SQL passée aux deux méthodes retourne le nom de l'utilisateur stocké dans le champ *emp_name* de la table *employees*. Une fois que le nom de l'utilisateur est récupéré, il est paramétré sur l'Etiquette *tasksuser_id_assign_by*. La seule différence entre les deux requêtes SQL est que si l'utilisateur insère un nouvel enregistrement i.e. la variable *IsInsertMode* est *true* (vraie), vous passez l'ID de l'utilisateur connecté depuis la variable *DBUtility.UserId*. Si l'utilisateur met à jour ou supprime l'enregistrement, alors vous passez l'ID utilisateur de l'utilisateur à qui la tâche était précédemment attribuée. La propriété *user_id_assign_by* de l'objet *item* contient cette valeur récupérée dans la base de données.

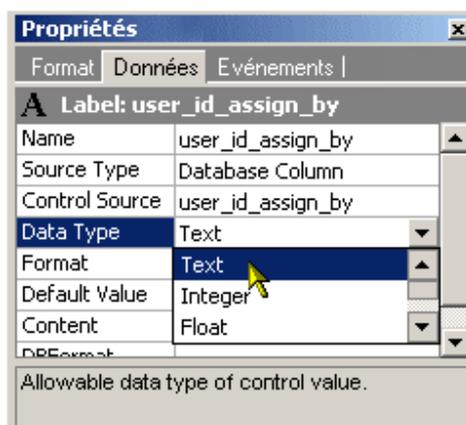
Le code entier se lit approximativement comme suit :

If a new record is being created : Utilise la valeur de *DBUtility.UserId* pour obtenir le nom de l'utilisateur connecté et pour attribuer son nom à la valeur de l'Etiquette *tasksuser_id_assign_by*.

If a record is being edited : Utilise la valeur de *tasksuser_id_assign_by* pour obtenir le nom de l'utilisateur à qui la tâche a été attribuée et pour attribuer son nom à la valeur de l'Etiquette *tasksuser_id_assign_by*.

Maintenant que vous avez modifié par programmation la valeur de l'Etiquette *user_id_assign_by* pour produire le Nom de l'Employé au lieu de son ID, vous allez avoir besoin de spécifier que ce champ est désormais un champ Texte, et non un champ Numérique.

- 1 - Cliquez sur l'onglet **Données** dans la fenêtre **Propriétés**.
- 2 - Sélectionnez *Text* en tant que **Data Type** (Type de Données).





Etape 3

Ajouter un Champ Caché "Assigned By" pour la Mise à Jour Automatique de Nouvelles Tâches

Vous avez précédemment utilisé l'événement *Before Show* pour afficher le nom de la personne à qui est attribuée une tâche. Cependant, les champs Etiquette ne peuvent pas être mis à jour de par leur nature, alors bien que le nom de l'employé soit affiché sur la page, il n'est pas écrit dans la base de données. Dans la mesure où nous voulons que la base de données enregistre le nom ou l'ID de la personne qui soumet une tâche, nous allons devoir ajouter la logique de programmation pour effectuer cela.

1 - Ajoutez un champ **Caché** à votre page depuis l'onglet **Formulaires** de la **Boîte à Outils (Toolbox)**.

Ce type de champ n'est pas visible dans le navigateur, mais il sera utilisé pour stocker une valeur et mettre à jour la base de données.

2 - Configurez le nouveau champ en paramétrant ses propriétés comme indiqué ci-dessous :

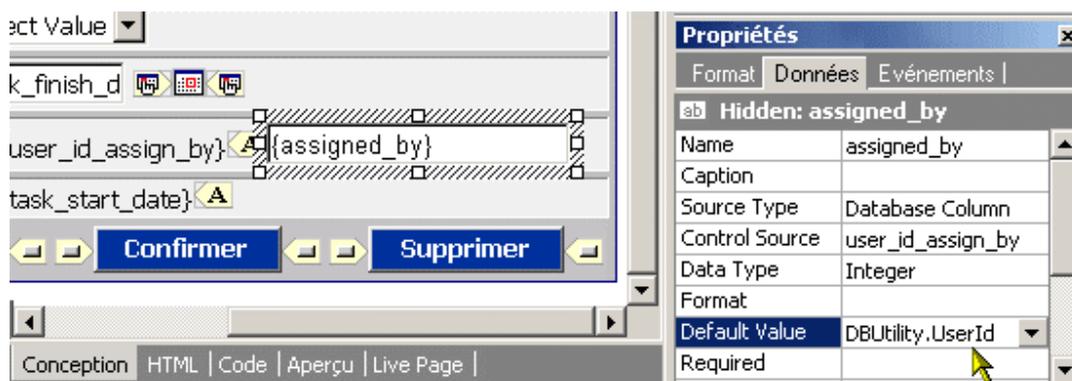
Name (Nom) : *assigned_by* - le nom du champ Caché qui vient d'être ajouté. Cela peut être n'importe quel nom.

Control Source : *user_id_assign_by* - le champ/colonne de la base de données qui sera utilisé(e) pour récupérer la valeur du champ et qui sera mis(e) à jour avec la nouvelle valeur en cas de changement.

Data Type (Type de Données) : *Integer* - le type de la valeur liée au *Control Source*. Nos ID utilisateur/employé sont numériques.

Default (Par Défaut) : *DBUtility.UserId* - valeur *Default* pour ce champ s'il est vide.

DBUtility.UserId : propriété de CodeCharge qui récupère l'ID utilisateur de l'utilisateur connecté au système. Ainsi vous pouvez simplement spécifier que vous voulez enregistrer l'ID de l'utilisateur courant dans le champ *user_id_assign_by* pour chaque nouvelle tâche qui sera soumise.



Ajouter un Champ Caché "Date Created" au Formulaire Enregistrement

Maintenant, nous allons ajouter un autre champ Caché à votre page, qui sera utilisé pour soumettre la date et l'heure courantes dans le champ *task_start_date* dans la base de données.

1 - Configurez le nouveau champ comme indiqué ci-dessous :

Name (Nom) : *date_created*

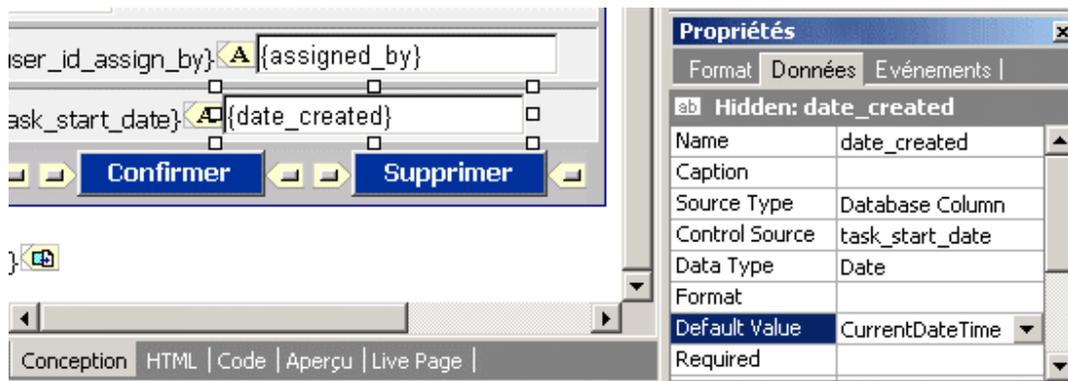
Control Source : *task_start_date*

Data Type (Type de Données) : *Date*

Default (Par Défaut) : *CurrentDateTime* - La valeur *CurrentDateTime* vous permet d'attribuer automatiquement la date et l'heure courantes aux nouvelles tâches. La propriété Par Défaut n'affecte pas les enregistrements existants, donc la date et l'heure des



tâches existantes ne seront pas modifiées lors de mises à jour ultérieures.



- 1 - Cliquez sur le champ `task_start_date` dans l'Explorateur de Projet.
 - 2 - Dans la fenêtre **Propriétés**, paramétrez **Default Value** (Valeur Par Défaut) à `CurrentDateTime`.
- C'est ainsi que le champ Etiquette peut afficher la date bien que le champ Caché ne soit pas visible pour l'utilisateur.

Tester les Champs Etiquette et Caché

Pour finir :

- 1 - Passez en mode **Live Page**.
- 2 - Sélectionnez ou ajoutez une Tâche, et vous voyez alors votre Etiquette afficher le nom de la personne qui a attribué la tâche.
- 3 - La version de base de votre Gestionnaire de Tâches est désormais finalisée. Ne pas oublier de sauvegarder !

[Employees](#) [Priorities](#) [Projects](#) [Statuses](#) [Tasks](#)

Ajouter/Editer Tasks

Project	My Project
Priority	Highest
Status	Closed
Type	Task
Name	Finish my project
Desc	Implement "Assigned By" Label to show values from another table
Assigned To	alexander
Finish Date	
Assigned By	George Pennington
Start Date	08.04.2003 16:48:07
Ajouter	



Etape 4

Programmation du Formulaire Enregistrement

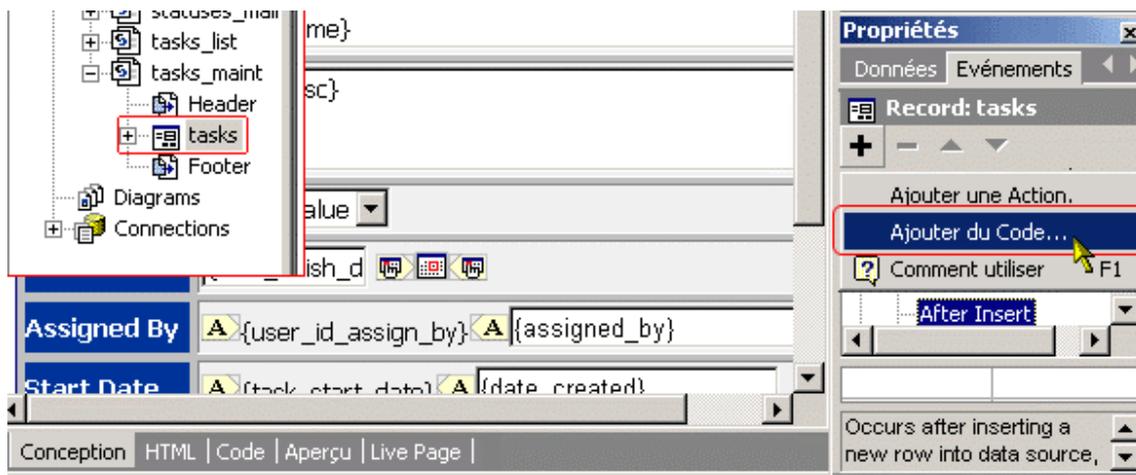
Maintenant que vous avez créé une application simple de Gestion de Tâches, vous souhaitez certainement l'étendre pour qu'elle soit à la fois plus pratique et plus utile. Dans cette section, vous allez avoir un aperçu sur la manière d'implémenter des applications pratiques et sophistiquées par l'ajout de code et d'actions, ceci afin d'offrir des fonctionnalités supplémentaires à votre Gestionnaire de Tâches.

Vous allez apprendre à :

- Envoyer des notifications d'email à la personne à qui la tâche est attribuée,
- Autoriser seulement la personne à qui la tâche est attribuée à la modifier.

Ajouter du Code dans l'Événement After Insert pour Envoyer des Emails

- 1 - Sélectionnez le formulaire *tasks* dans la page *tasks_maint* en le sélectionnant depuis l'**Explorateur de Projet**, ou en cliquant n'importe où dans la légende du formulaire.
- 2 - Dans la fenêtre **Propriétés**, cliquez sur l'onglet **Événements**.
- 3 - Sélectionnez l'événement **After Insert**.
- 4 - Cliquez sur le bouton **[+]**.
- 5 - Sélectionnez **Ajouter du Code...**



Une fois en mode **Visualisation du Code**, remplacez le commentaire généré :

C#

```
' Write your own code here.
```

VB.Net

```
' Write your own code here.
```



Par le code ci-dessous :

C#

```
SqlCommand userEmail = new SqlCommand( "SELECT email FROM" +
    "employees WHERE emp_id=" + DBUtility.UserId,
    Settings.IntranetDBDataAccessObject );

SqlCommand assignedUserEmail = new SqlCommand( "SELECT email FROM" +
    "employees WHERE emp_id=" + item.user_id_assign_to.Value.ToString(),
    Settings.IntranetDBDataAccessObject );

SqlCommand taskId = new SqlCommand( "SELECT max(task_id) FROM" +
    "tasks WHERE user_id_assign_by=" + DBUtility.UserId,
    Settings.IntranetDBDataAccessObject );

System.Web.Mail.MailMessage newMessage = new System.Web.Mail.MailMessage();

newMessage.From = userEmail.ExecuteScalar().ToString();
newMessage.To = assignedUserEmail.ExecuteScalar().ToString();
newMessage.Subject = "New task for you!";
newMessage.BodyFormat = System.Web.Mail.MailFormat.Html;
newMessage.Body = "The following task was submitted:<br><br>" +
    "Task ID :"+ taskId.ExecuteScalar().ToString() +
    "<br><br>" + item.task_desc.Value;

System.Web.Mail.SmtpMail.SmtpServer = "localhost";
System.Web.Mail.SmtpMail.Send(newMessage);
```

**VB.Net**

```
Dim userEmail, assignedUserEmail, taskId As String

userEmail = Convert.ToString(Settings.IntranetDBDataAccessObject.ExecuteScalar("SELECT email
    FROM employees WHERE emp_id=" & DBUtility.UserId))

assignedUserEmail = Convert.ToString(Settings.IntranetDBDataAccessObject.ExecuteScalar("SELECT email
    FROM employees WHERE emp_id=" & Convert.ToString(item.user_id_assign_to.Value)))

taskId = Convert.ToString(Settings.IntranetDBDataAccessObject.ExecuteScalar("SELECT max(task_id)
    FROM tasks WHERE user_id_assign_by=" & DBUtility.UserId))

Dim newMessage As New System.Web.Mail.MailMessage()

newMessage.From = userEmail
newMessage.To = assignedUserEmail
newMessage.Subject = "New task for you!"
newMessage.BodyFormat = System.Web.Mail.MailFormat.Html
newMessage.Body = "The following task was submitted:<br><br> Task ID : " & taskId &
    "<br><br>" & item.task_desc.Value

System.Web.Mail.SmtpMail.SmtpServer = "localhost"
System.Web.Mail.SmtpMail.Send(newMessage)
```

Comme vous pouvez le constater, le code ci-dessus envoie des emails aux utilisateurs à qui de nouvelles tâches sont attribuées. Voici des informations que vous devez connaître :

1 - Le code ci-dessus utilise les classes fournies par le framework .NET pour envoyer des e-mails, aussi vous n'avez pas besoin d'installer des composants supplémentaires.

2 - Les classes du framework .NET reposent sur le composant CDO pour envoyer des e-mails ; par conséquent vous avez besoin d'un service SMTP installé sur le serveur hébergeant cette application.

Ce qui suit est une explication du code du dessus :

```
SqlCommand userEmail = new SqlCommand("SELECT email FROM " +
    "employees WHERE emp_id=" + DBUtility.UserId,
    Settings.IntranetDBDataAccessObject );
```

```
SqlCommand assignedUserEmail = new SqlCommand( "SELECT email FROM " +
    "employees WHERE emp_id=" + item.user_id_assign_to.Value.ToString(),
    Settings.IntranetDBDataAccessObject );
```



```
SqlCommand taskId = new SqlCommand("SELECT max(task_id) FROM" +  
"tasks WHERE user_id_assign_by=" + DBUtility.UserId ,  
Settings.IntranetDBDataAccessObject );
```

Dans le code du dessus trois objets *SqlCommand* sont définis. Comme vous devez vous en souvenir, ces objets sont utilisés pour exécuter des requêtes de la base de données. La première requête SQL est utilisée pour récupérer l'adresse email de l'utilisateur connecté. La seconde requête SQL récupère l'adresse email de l'utilisateur à qui la tâche est attribuée. La propriété *user_id_assign_to* de l'objet *item* est utilisée pour obtenir l'ID de l'utilisateur à qui la tâche est attribuée. Enfin, la dernière requête SQL est utilisée pour obtenir l'ID de la tâche. Le dernier ID inséré de tâche peut être obtenu en utilisant différentes méthodes avec différentes bases de données. Malheureusement, Microsoft Access ne supporte pas la récupération du dernier enregistrement inséré ; donc vous allez avoir besoin d'utiliser l'objet *SqlCommand* pour consulter le plus grand ID de tâche soumis par l'utilisateur connecté (en supposant que les ID de tâches sont créés par incrémentation).

```
System.Web.Mail.MailMessage newMessage = new System.Web.Mail.MailMessage();
```

Crée un objet *MailMessage*, qui est sous l'espace de nommage *System.Web.Mail*.

```
newMessage.From = userEmail.ExecuteScalar().ToString();  
newMessage.To = assignedUserEmail.ExecuteScalar().ToString();  
newMessage.Subject = "New task for you!";  
newMessage.BodyFormat = System.Web.Mail.MailFormat.Html;  
newMessage.Body = "The following task was submitted:<br><br>" +  
"Task ID :"+ taskId.ExecuteScalar().ToString() +  
"<br><br>" + item.task_desc.Value;
```

Le code du dessus exécute la requête de la base de données et positionne les diverses propriétés de l'objet *MailMessage*. Positionnez la propriété "*BodyFormat*" de l'objet *MailMessage* à *Html*, ce qui permettra d'envoyer un contenu au format HTML (et non au format texte brut). Le corps de email comprend la description de la tâche et de l'ID de tâche.

```
System.Web.Mail.SmtpMail.SmtpServer = "localhost"
```

```
System.Web.Mail.SmtpMail.Send(newMessage);
```

Positionnez l'adresse du serveur SMTP qui sera utilisé pour envoyer l'email, puis utilisez la méthode statique *Send* de la classe *SmtpMail* pour envoyer l'email. Si vous n'avez pas de serveur SMTP sur votre machine en local, vous pouvez entrer l'adresse d'un serveur SMTP externe auquel vous avez accès (mail.yourdomain.com).

Utiliser l'Événement After Update pour Envoyer des Emails

Vous avez précédemment ajouté le code nécessaire qui envoie une notification d'email à la personne à qui la tâche est attribuée, une fois l'enregistrement d'une nouvelle tâche effectuée dans le système. Maintenant, il nous faut implémenter une fonctionnalité similaire dans l'événement *After Update* pour informer la personne à qui la tâche est attribuée, lorsqu'une tâche existante est mise à jour et réattribuée à quelqu'un d'autre.

- 1 - Cliquez sur le formulaire *tasks* dans l'**Explorateur de Projet**.
- 2 - Dans la fenêtre **Propriétés**, sélectionnez l'onglet **Événements**.
- 3 - Ajoutez le **Custom Code** (Code Personnalisé) dans l'événement **After Update** :



C#

```
if(DBUtility.UserId.ToString() != item.user_id_assign_to.Value.ToString())
{
    SqlCommand userEmail = new SqlCommand("SELECT email FROM" +
        "employees WHERE emp_id=" + DBUtility.UserId,
        Settings.IntranetDBDataAccessObject);
    SqlCommand assignedUserEmail = new SqlCommand("SELECT email FROM" +
        "employees WHERE emp_id=" + item.user_id_assign_to.Value.ToString(),
        Settings.IntranetDBDataAccessObject);
    System.Web.Mail.MailMessage newMessage = new System.Web.Mail.MailMessage();
    newMessage.From = userEmail.ExecuteScalar().ToString();
    newMessage.To = assignedUserEmail.ExecuteScalar().ToString();
    newMessage.Subject = " A task was assigned to you";
    newMessage.BodyFormat = System.Web.Mail.MailFormat.Html;
    newMessage.Body = "The following task was assigned to you:<br><br>" +
        "Task ID : " + Request.QueryString["task_id"].ToString() +
        "<br><br>" + item.task_desc.Value;
    System.Web.Mail.SmtpMail.SmtpServer = "localhost";
    System.Web.Mail.SmtpMail.Send(newMessage );
}
```

**VB.Net**

```
If Convert.ToString (DBUtility.UserId) <> Convert.ToString(item.user_id_assign_to.Value) Then

    Dim userEmail, assignedUserEmail As String
    userEmail = Convert.ToString(Settings.IntranetDBDataAccessObject.ExecuteScalar("SELECT email
        FROM employees WHERE emp_id=" & DBUtility.UserId))
    assignedUserEmail = Convert.ToString(Settings.IntranetDBDataAccessObject.ExecuteScalar("SELECT email
        FROM employees WHERE emp_id=" & Convert.ToString(item.user_id_assign_to.Value)))

    Dim newMessage As New System.Web.Mail.MailMessage()

    newMessage.From = userEmail
    newMessage.To = assignedUserEmail
    newMessage.Subject = "A task was assigned to you"
    newMessage.BodyFormat = System.Web.Mail.MailFormat.Html
    NwMessage.Body = "The following task was assigned to you :<br><br> Task ID : " &
        Convert.ToString(Request.QueryString("task_id")) & "<br><br>" & item.task_desc.Value

    System.Web.Mail.SmtpMail.SmtpServer = "localhost"
    System.Web.Mail.SmtpMail.Send(newMessage)

End if
```

Les principales différences entre le code ci-dessus et celui utilisé dans l'événement *After Insert* sont les suivantes :

- 1 - Une condition *if* a été ajoutée pour envoyer un email seulement si un utilisateur attribue une tâche à une autre personne.
- 2 - *task_id* est récupéré depuis l'URL utilisant l'indexeur *Request.QueryString*. Nous pouvons utiliser cette méthode parce que des tâches peuvent être mises à jour seulement si l'utilisateur est arrivé à la page en cours par une URL qui contient un ID de tâche à mettre à jour. Une telle URL ressemble à : http://localhost/TaskManager/tasks_maint.aspx?task_id=9

Test de l'Envoi d'Email

Avant de tester l'envoi d'email :

- 1 - Vous devez ajouter de nouveaux utilisateurs à votre base de données avec des adresses email correctes, ou modifier les utilisateurs existants en changeant leur adresse email :
 - A - Vous pouvez le faire en ouvrant la base de données *Intranet.mdb* qui se trouve dans votre répertoire projet.
 - B - Sinon, vous pouvez utiliser le Gestionnaire de Tâches. Allez à la page Employés pour y visualiser les emails utilisateur et les modifier.
- 2 - Une fois vos utilisateurs configurés avec le test d'emails, sauvegardez votre projet et passez en mode **Live Page** pour tester votre système.



Remarque : Vous aurez besoin de Microsoft Access 2000 ou une version supérieure pour éditer manuellement le fichier base de données. Si votre code email fonctionne correctement, vous devez alors vous retrouver sur la page Liste de Tâches après l'ajout ou la modification d'une tâche, et la personne à qui la tâche a été attribuée doit recevoir un email.

Etape 5

Implémenter la Sécurité d'Enregistrement dans l'Événement After Initialize

Votre Gestionnaire de Tâches est désormais pratiquement complet, seule une fonctionnalité pouvant s'avérer importante manque, à savoir la Sécurité. En l'état actuel n'importe qui peut modifier et supprimer n'importe quelle tâche. Vous voudrez certainement limiter l'accès pour que seuls les employés autorisés puissent mettre à jour leurs tâches. Il y a plusieurs façons de procéder, et nous allons examiner plusieurs d'entre elles.

- 1 - Cliquez sur la page `tasks_maint` dans l'**Explorateur de Projet**.
- 2 - Dans la fenêtre **Propriétés**, sélectionnez l'onglet **Événements**.
- 3 - Ajoutez **Custom Code** (Code Personnalisé) à l'événement **After Initialize** de la page. Procédez comme suit :

C#

```
// Write your own code here.
```

VB.Net

```
' Write your own code here.
```

Avec le code ci-dessous :

C#

```
if(Request.QueryString["task_id"] != null)
{
    SqlCommand taskCmd = new SqlCommand("SELECT user_id_assign_to FROM " +
        "tasks WHERE task_id=" + Request.QueryString["task_id"],
    Settings.IntranetDBDataAccessObject);
    int assignedUserId = (int)taskCmd.ExecuteScalar();

    if(int.Parse(DBUtility.UserId.ToString()) != assignedUserId)
    {
        tasksHolder.Visible = false;
        //Response.Redirect("tasks_list.aspx");
    }
}
```

**VB.Net**

```
if Request.QueryString("task_id") <> " " Then
    Dim assignedUserId As Integer

    assignedUserId = Settings.IntranetDBDataAccessObject.ExecuteScalar("SELECT user_id_assign_to
        FROM tasks WHERE task_id=" & Request.QueryString("task_id"))

    if Convert.ToInt32(DBUtility.UserId) <> assignedUserId Then
        tasksHolder.Visible = false;
        ` Response.Redirect( "tasks_list.aspx" )
    End if
End if
```

Le code du dessus permet de tester les méthodes suivantes pour implémenter la sécurité d'enregistrement :

Ne pas montrer la tâche (formulaire Enregistrement) sur la page si la tâche sélectionnée n'appartient pas à l'utilisateur connecté. Un utilisateur non autorisé verra une page blanche.

Vous pouvez cacher n'importe quel formulaire sur une page en attribuant une valeur *false* (faux) à la propriété *Visible* (visible) de la page dont le formulaire dépend. Tout d'abord, vérifiez la présence de la variable *query string task_id*, qui précise le mode du formulaire Enregistrement (mode Mise à jour ou Suppression), puisque vous voulez limiter les droits des utilisateurs : visualisation et modification des tâches qui leur sont attribuées. Le bloc *if* s'assure que tous les utilisateurs peuvent créer de nouvelles tâches. Vous pouvez tester cette fonctionnalité en insérant le code du dessus dans l'événement, puis en passant en mode **Live Page** et en essayant de modifier une tâche qui ne vous est pas attribuée... et dans ce cas, vous verrez une page vide (seulement avec un en-tête). Bien qu'une telle fonctionnalité ne soit pas très utile, elle illustre comment vous pouvez cacher les formulaires sur une page. Ajoutez un autre formulaire enregistrement à votre page, qui ne puisse pas être mis à jour et qui ait seulement des champs Etiquette qui montre les informations relatives aux tâches. Une fois que vous avez deux formulaires sur la même page, vous pouvez cacher chaque formulaire par programmation en utilisant des critères d'exclusion.

Rediriger les utilisateurs non autorisés sur une autre page. Seuls les utilisateurs à qui sont attribuées des tâches peuvent visualiser la page.

Vous pouvez mettre en application et tester cette fonctionnalité en modifiant légèrement le code du dessus comme montré ci-dessous :

**C#**

```
if(Request.QueryString["task_id"] != null)
{
    SqlCommand taskCmd = new SqlCommand( "SELECT user_id_assign_to FROM" +
        "tasks WHERE task_id=" + Request.QueryString["task_id"],
        Settings.IntranetDBDataAccessObject );

    int assignedUserId = (int)taskCmd.ExecuteScalar();

    if(int.Parse(DBUtility.UserId.ToString()) != assignedUserId)
    {
        //tasksHolder.Visible = false;
        Response.Redirect("tasks_list.aspx");
    }
}
```

VB.Net

```
if Request.QueryString("task_id") <> " " Then
    Dim assignedUserId As Integer

    assignedUserId = Settings.IntranetDBDataAccessObject.ExecuteScalar("SELECT user_id_assign_to
    FROM tasks WHERE task_id=" & Request.QueryString("task_id"))

    if Convert.ToInt32(DBUtility.UserId) <> assignedUserId Then;
        `taskHolder.Visible = false
        Response.Redirect("tasks_list.aspx")
    End if
End if
```

Le code ci-dessus indique que vous devez commenter la ligne précédemment active et activer la ligne qui commence avec *Response.Redirect*. La méthode *Redirect* de l'objet *Response* est utilisée pour rediriger l'utilisateur sur une nouvelle page. Vous pouvez simplement attribuer la page de destination à la méthode *Redirect* et la page sera alors automatiquement redirigée. Pour tester cette fonctionnalité : modifiez le code tel que montré, passez en mode **Live Page**, puis essayez de modifier une tâche qui ne vous est pas attribuée.



JSP

Extension des Fonctionnalités de l'Application avec la Programmation d'Événements (JSP)

Vous avez certainement remarqué que jusqu'à présent vous avez construit votre Gestionnaire de Tâches sans avoir à vous préoccuper du code. En effet, CodeCharge Studio peut vous aider à construire des systèmes fonctionnels sans programmation ; toutefois la création de systèmes plus sophistiqués requiert un minimum de programmation. Heureusement, CodeCharge Studio facilite la programmation en proposant un éditeur de code de haut niveau, en plus des Événements et des Actions qui vous aident à insérer des morceaux de code pré-programmés dans le programme.

Voici les définitions d'une Action et d'un Événement :

Action

Composant de génération de code définissable par l'utilisateur, qui insère un bloc de code dans une procédure événementielle. CodeCharge Studio est fourni avec de nombreuses Actions pré-définies, qui se trouvent dans le dossier suivant : (*CCS folder*)\Components\Actions. Intérieurement, les actions consistent en du code XML et XSL pouvant être facilement personnalisé. Par exemple, une action peut être paramétrée sur un Textbox pour valider une adresse e-mail.

Procédure Événementielle

Une procédure automatiquement exécutée en réponse à un événement lancé par un programme à son exécution. Les événements sont les meilleurs emplacements pour mettre du Code Personnalisé (*Custom Code*).

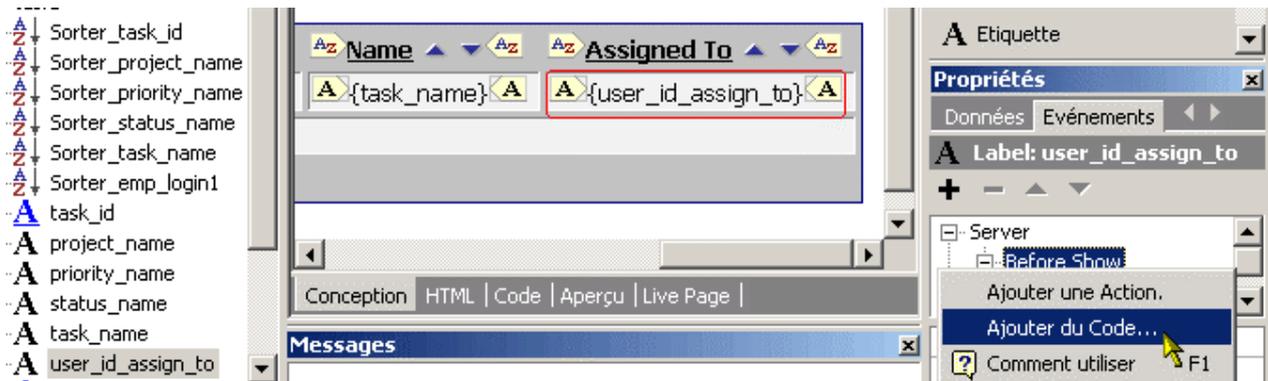
Étape 1

Utiliser l'Événement Before Show Row pour Changer la Couleur du Texte

Démarrons la programmation basique avec une tâche simple de changement de la couleur d'un champ grille sur notre page Liste de Tâches. Concrètement, nous allons mettre en évidence les tâches qui vous sont attribuées en affichant votre nom en bleu dans la grille.

- 1 - Ouvrez la page *tasks_list* dans l'**Explorateur de Projet**.
- 2 - Ouvrez l'arborescence de la grille *tasks*.
- 3 - Faites un clic droit sur le champ *emp_login1* et sélectionnez **Propriétés**.
- 4 - Dans l'onglet **Données**, paramétrez la valeur de la propriété **Name** (Nom) à *user_id_assign_to*.
- 5 - Dans l'**Explorateur Projet**, faites un clic droit sur le champ *task_name* et sélectionnez **Propriétés**.
- 6 - Dans l'onglet **Données**, paramétrez la valeur de la propriété **Content** (Contenu) à *HTML*.
- 7 - Sélectionnez la grille *tasks* dans l'**Explorateur de Projet**, ou cliquez n'importe où dans la légende du formulaire.
- 8 - Sélectionnez l'onglet **Événements** dans la fenêtre **Propriétés**.
- 9 - Faites un clic droit sur l'événement **Before Show Row** et sélectionnez **Ajouter du Code...**

L'événement *Before Show Row* se déclenche dans le programme après que les valeurs champ soient attribuées, mais avant d'être générées en HTML. En ajoutant du code dans cet événement, vous pouvez modifier la valeur champ avant qu'elle ne soit montrée.



Contrôler par Programmation la Valeur Champ

Une fois que vous avez ajouté du Code Personnalisé (*Custom Code*) à l'Événement, vous verrez la fenêtre d'édition du code avec l'emplacement approprié pour entrer le nouveau code.

Remplacez la ligne de code suivante :

```
/* Write your own code here. */
```

par :

```
String value = e.getControl().getFormattedValue();
String login = Utils.getUserLogin(e.getPage());
if ( value != null && value.equals(login))
{
    e.getControl().setFormattedValue("<b><font color=\"blue\">" + value + "</font></b>");
}
```

Voyons comment le code JSP du dessus fonctionne :

```
String value = e.getControl().getFormattedValue();
String login = Utils.getUserLogin(e.getPage());
if ( value != null && value.equals(login)) {
```

Les lignes du dessus prépare et exécute la condition "if" (si) qui vérifie si la valeur du contrôle en cours (*user_id_assign_to*) est égale au nom du login utilisateur. Une fois connecté au système, le programme reconnaîtra vos tâches en comparant votre nom de login à la valeur *emp_login* de la personne à qui la tâche est attribuée. La méthode *Utils.getUserLogin(Page)* est une des méthode statique auxiliaire dans la classe *Utils*, et elle conserve le nom de Login de la session utilisateur en cours jusqu'à son expiration.

Note :

Ce qui suit sont des différentes fonctions très utiles de la classe *Utils* :

getUserId(Page) : la valeur champ clé primaire de la session utilisateur

getUserLogin(Page) : le nom de Login de l'utilisateur connecté au système

getUserGroup(Page) : le niveau/groupe de sécurité de l'utilisateur connecté au système

Le paramètre Page est requis pour accéder à l'objet Session dans lequel ces valeurs sont en fait stockées.



```
e.getControl().setFormattedValue("<b><font color=\"blue\">" + value + "</font></b>");
```

Ce code est exécuté si la condition précédente "if" est rencontrée. Elle modifie la valeur du contrôle (*user_id_assign_to*). La valeur du champ est remplacée avec la valeur du nom du login utilisateur encapsulé dans du code HTML qui spécifie la couleur de police (le bleu), et ajoute un tag HTML `` pour obtenir une police en gras. Vous remarquerez que le mot *blue* a des double guillemets, qui seront remplacés par un guillemet simple à l'exécution. Dans la mesure où les guillemets marquent le début et la fin de la chaîne de caractères, l'utilisation de double guillemet permet d'insérer un guillemet dans une chaîne de caractères.

Prévisualiser la Page Liste de Tâches

- 1 - Sauvegardez votre projet.
- 2 - Passez en mode **Live Page** pour visualiser votre page fonctionnelle. Si aucun nom de tâche n'est mis en évidence dans la colonne ("Assigned To"), c'est que vous n'êtes probablement pas connecté(e) au système.
- 3 - Le menu ne contenant pas encore de lien à la page Login, vous pouvez y accéder en essayant d'ouvrir une des pages restreintes, par exemple la page Maintenance de Tâches.
- 4 - Cliquez sur n'importe lequel des ID projet, et vous devriez voir la page Login.
- 5 - Connectez-vous en tant que **george/george**,
- 6 - Cliquez sur le lien **Tasks** sur le menu pour retourner à la page Liste de Tâches.

Maintenant, vous devriez voir que certaines tâches sont mises en évidence ; ces tâches correspondent aux tâches de l'utilisateur pour lequel vous avez ouvert une session.

[Employees](#) [Priorities](#) [Projects](#) [Statutes](#) [Tasks](#)

Rechercher Tasks

Mot clé	<input type="text"/>
Projet	Rechercher par valeur ▼
<input type="button" value="Rechercher"/>	

Liste de : Tasks

<u>Id</u>	<u>Project</u>	<u>Priority</u>	<u>Status</u>	<u>Name</u>	<u>Assigned To</u>
1	CodeCharge	High	On hold	Great Project needs to be greater	helen
2	CodeCharge	Highest	Closed	Fix ALL bugs	george
3	CodeCharge	High	Closed	Get ready to click	peter
4	My Project	Highest	Open	Finish My Project	ignace
5	Test Project	High	In progress	Test this project.	ken
6	CodeCharge	Highest	Open	Code with one hand.	alexander
7	Test Project	Highest	On hold	Get armed	helen
8	Test Project	Highest	Open	Write more code	ignace
9	Super Project	Highest	In progress	Code, code, code...	george
10	Test Project	Lowest	On hold	Sleep	ken
11	Super Project	Highest	Open	Have fun	alexander

Nouveau 1 of 1



Etape 2

Modifier un Champ Etiquette sur la Page Maintenance de Tâches

Maintenant procédons à une modification nécessaire sur la page Maintenance de Tâches sur laquelle vous avez dû remarqué que le champ Etiquette *Assigned By* n'affiche pas le nom de l'employé, mais l'ID utilisateur, comme montré ci-dessous ; c'est parce que la table *employees* contient le nom de l'utilisateur, juste comme la ListBox *Assigned To* affiche les noms des employés d'une autre table.

[Employees](#) [Priorities](#) [Projects](#) [Statuses](#) [Tasks](#)

Ajouter/Editer Tasks

Project	CodeCharge
Priority	Highest
Status	Closed
Type	Question
Name	Fix ALL bugs
Desc	Staying up at night coding? Get CodeCharge, go home, get rest.
Assigned To	george
Finish Date	03.02.2003
Assigned By	3
Start Date	02.02.2003
<input type="button" value="Confirmer"/> <input type="button" value="Supprimer"/>	

Il y a plusieurs méthodes possibles pour traiter de la question. Merci de vous référer aux explications qui suivent :

1 - Créez une Requête qui contient plusieurs tables et qui peut être utilisée en tant que source de données par le formulaire Enregistrement. Malheureusement, des requêtes qui contiennent plusieurs tables ne peuvent pas être mises à jour de par leur nature, et donc votre formulaire Enregistrement entier peut s'arrêter de fonctionner. En d'autres termes, si vous avez indiqué que vous vouliez utiliser une requête contenant une table *tasks* et *employees* dans votre formulaire Enregistrement, alors si vous attribuez une tâche à quelqu'un d'autre, le programme ne pourra pas savoir si vous vouliez mettre à jour la table *tasks* avec le nouveau *employee_id*, ou si vous vouliez mettre à jour la table *employees* et changer le nom de l'employé.

Ainsi, si vous utilisez plusieurs tables comme source de données pour le formulaire Enregistrement, vous devrez également définir les opérations *Custom Insert*, *Custom Update* et *Custom Delete* dans les propriétés du formulaire Enregistrement pour indiquer quels champs de la base de données devront être mis à jour avec les valeurs correspondantes saisies sur la page. Cette méthode semble en effet représenter beaucoup d'effort juste pour afficher une valeur supplémentaire sur la page.

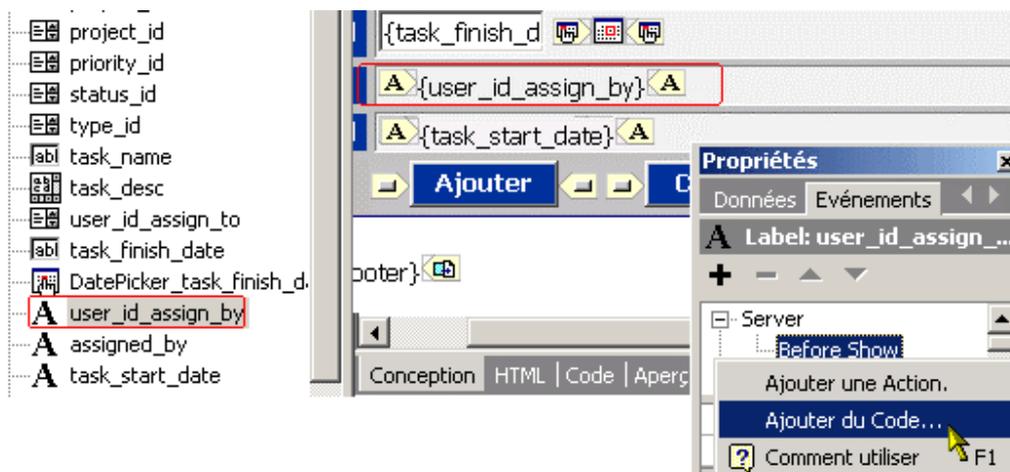
2 - Utilisez une Procédure Événementielle pour insérer du Code Personnalisé (*Custom Code*) où vous pourrez produire la valeur désirée par programmation. Cette méthode est très flexible, dans la mesure où elle vous permet d'étendre le code généré par l'ajout de votre propre code. L'étape qui suit décrit en détail cette méthode.

Utiliser l'Événement Before Show pour Changer la Valeur d'une



Etiquette

- 1 - En mode **Conception**, sélectionnez l'Etiquette `user_id_assign_by`,
 - 2 - Dans la fenêtre **Propriétés**, cliquez sur l'onglet **Evénements**.
 - 3 - Faites un clic droit sur l'événement **Before Show** et sélectionnez **Ajouter du Code...** .
- CodeCharge Studio va alors automatiquement passer en mode **Visualisation du Code**.



Une fois en mode **Visualisation du Code**, vous devriez voir les lignes de code suivantes :

```
//user_id_assign_by Label Handler Head @13-91DFA71C-->
public class tasksuser_id_assign_byLabelHandler implements LabelListener
{
    public void beforeShow(CCSEvent e)
    {
        //End user_id_assign_by Label Handler Head-->
        //Event BeforeShow Action Custom Code @13-4EFC2132-->
        /* Write your own code here. */
        //End Event BeforeShow Action Custom Code-->
        //user_id_assign_by Label Handler Tail @13-F5FC18C5-->
    }
}
//End user_id_assign_by Label Handler Tail-->
```

- 2 - Remplacez le texte :

```
/* Write your own code here. */
```



par :

```
if (e.getRecord().isEditMode())
{
    e.getControl().setValue(DBTools.dLookUp("emp_name", "employees", "emp_id=" +
    DBTools.toSql(e.getControl().getFormattedValue(),JDBCConnection.INTEGER,"IntranetDB"), "IntranetDB"));
} else
{
    e.getControl().setValue(DBTools.dLookUp("emp_name","employees", "emp_id="+
    DBTools.toSql(Utils.getUserId(e.getPage()),JDBCConnection.INTEGER,"IntranetDB") , "IntranetDB"));
}
```

Dans le code du dessus, le premier bloc "if" vérifie si le formulaire Enregistrement en cours est en mode mise à jour, et si tel est le cas, il récupère le nom de l'employé approprié, et lui attribue la valeur de contrôle en cours. Si le formulaire n'est pas en mode mise à jour, le nom de la session utilisateur en cours est attribué comme valeur de contrôle.

Les éléments suivants constituent le code :

e : le pointeur de l'Etiquette *user_id_assign_by*, pour laquelle l'événement est appelé.

getFormattedValue and *setFormattedValue* : les méthodes qui lisent et modifient la valeur du contrôle (dans ce cas l'Etiquette). Ces méthodes utilisent en option un paramètre qui spécifie si une valeur doit être convertie en un nombre (Entier) ou en du texte (Chaîne de caractères).

isEditMode : propriété du formulaire (parent du contrôle), qui spécifie si l'enregistrement peut être édité/mis à jour. En fonction de la valeur de cette propriété, nous affichons soit le nom de la personne qui a initialement soumis la tâche (mode Mise à jour), ou celui de la personne qui soumet la tâche (mode Insertion).

getRow : une méthode qui récupère une valeur de base de données basée sur une requête SQL. Ici, cette méthode récupère le nom de l'Employée (*emp_name*) dans la table *employees* en utilisant la condition suivante : la clé (*emp_id*) est égale à la valeur courante de l'Etiquette.

IntranetDB : le nom de la connexion objet *pooled* qui définit la connexion base de données utilisées pour récupérer le nom de l'employé.

Utils.getUserId(e.getPage()) : une méthode qui récupère la valeur de la session "UserID", qui contient l'ID de l'utilisateur connecté au système.

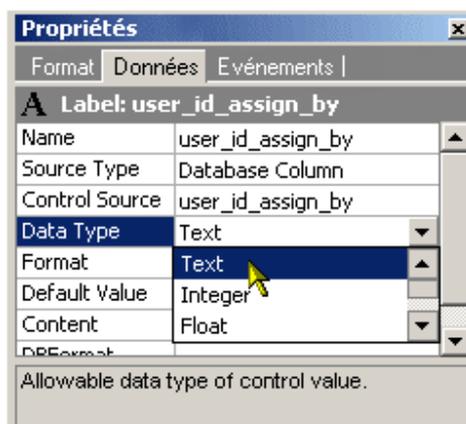
Le morceau de code se lit approximativement comme suit :

If record is being edited : Attribue le nom de la personne qui a initialement soumis la question à l'Etiquette *user_id_assign_by*, en cherchant le nom de l'employé dans la table *employees* utilisant la connexion *IntranetDB* et la valeur de l'Etiquette *user_id_assign_by*.

If new record is being created : Attribue un utilisateur connecté à l'Etiquette *user_id_assign_by* en récupérant son nom dans la table *employees* utilisant la connexion *IntranetDB* et la session "UserID" qui contient l'ID utilisateur courant.

Maintenant que vous avez modifié par programmation la valeur de l'Etiquette *tasksuser_id_assign_by* pour sortir le nom de l'employé à la place de l'ID, vous allez devoir spécifier que ce champ est un champ Texte, et non plus un champ Numérique.

- 1 - Cliquez sur l'onglet **Données** dans la fenêtre **Propriétés**.
- 2 - Sélectionnez *Text* en tant que **Data Type** (Type de Données).



Etape 3

Ajouter un Champ Caché "Assigned By" pour la Mise à Jour Automatique de Nouvelles Tâches

Vous avez précédemment utilisé l'événement *Before Show* pour afficher le nom de la personne à qui est attribuée une tâche. Cependant, les champs Etiquette ne peuvent pas être mis à jour de par leur nature, alors bien que le nom de l'employé soit affiché sur la page, il n'est pas écrit dans la base de données. Dans la mesure où nous voulons que la base de données enregistre le nom de l'ID de la personne qui soumet une tâche, nous allons donc devoir ajouter la logique de programmation.

1 - Ajoutez un champ **Caché** à votre page depuis l'onglet **Formulaires** de la **Boîte à Outils** (*Toolbox*).

Ce type de champ n'est pas visible dans le navigateur, mais il sera utilisé pour stocker une valeur et mettre à jour la base de données.

2 - Configurez le nouveau champ en paramétrant ses propriétés comme indiqué ci-après :

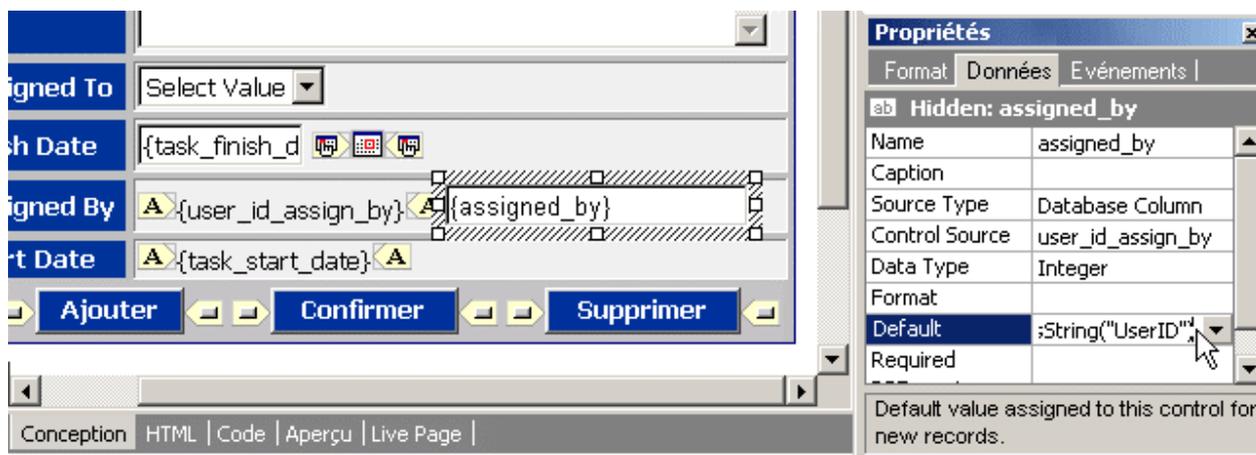
Name (*Nom*) : *assigned_by* - le nom du champ Caché qui vient d'être ajouté. Cela peut être n'importe quel nom.

Control Source : *user_id_assign_by* - le champ/colonne de la base de données qui sera utilisé(e) pour retrouver la valeur du champ et qui sera mis(e) à jour avec la nouvelle valeur en cas de changement.

Data Type (*Type de Données*) : *Integer* - le type de la valeur liée au *Control Source*. Nos ID utilisateur/employé sont numériques.

Default (Par Défaut) : *Utils.getUserId(e.getPage())* - Valeur Default pour ce champ s'il est vide.

Ce code récupère l'ID utilisateur de la session en cours. Ainsi vous pouvez simplement indiquer que vous voulez enregistrer l'ID de l'utilisateur connecté dans le champ *user_id_assign_by* pour chaque nouvelle tâche qui sera soumise.



Ajouter un Champ Caché "Date Created" au Formulaire Enregistrement

Maintenant, ajoutez un autre champ Caché à votre page, qui sera utilisé pour soumettre la date et l'heure courantes dans le champ `task_start_date` dans la base de données.

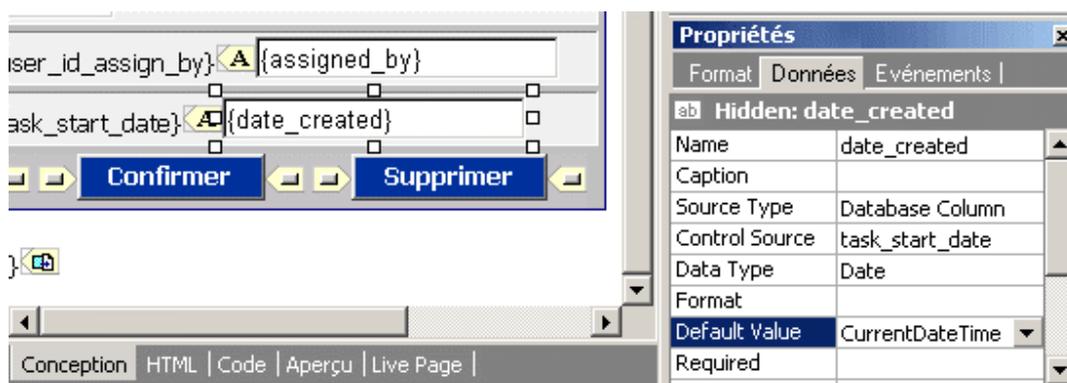
Configurez le nouveau champ comme indiqué ci-dessous :

Name (Nom) : `date_created`

Control Source : `task_start_date`

Data Type (Type de Données) : `Date`

Default (Par Défaut) : `CurrentDateTime` - La valeur `CurrentDateTime` vous permet d'attribuer automatiquement la date et l'heure courantes aux nouvelles tâches. La propriété `Default` n'affecte pas les enregistrements existants, donc la date et l'heure des tâches existantes ne seront pas modifiées lors des mises à jour.



Tester les Champs Etiquette et Caché

Pour terminer :

- 1 - Passez en mode **Live Page**.
- 2 - Sélectionnez ou ajoutez une Tâche, et vous voyez alors votre Etiquette afficher le nom de la personne qui a attribué la tâche.
- 3 - La version de base de votre Gestionnaire de Tâches est désormais finalisée. Ne pas oublier de sauvegarder !



[Employees](#) [Priorities](#) [Projects](#) [Statutes](#) [Tasks](#)

Ajouter/Editer Tasks

Project	My Project
Priority	Highest
Status	Closed
Type	Task
Name	Finish my project
Desc	Implement "Assigned By" Label to show values from another table
Assigned To	alexander
Finish Date	<input type="text"/>
Assigned By	George Pennington
Start Date	08.04.2003 16:48:07
<input type="button" value="Ajouter"/>	

Etape 4

Programmation du Formulaire Enregistrement

Maintenant que vous avez créé une application simple de Gestion de Tâches, vous souhaitez certainement l'étendre pour qu'elle soit à la fois plus pratique et plus utile... Dans cette section, vous allez avoir un aperçu sur la manière d'implémenter des applications pratiques et sophistiquées par l'ajout de code et d'actions, ceci afin d'offrir des fonctionnalités supplémentaires à votre Gestionnaire de Tâches.

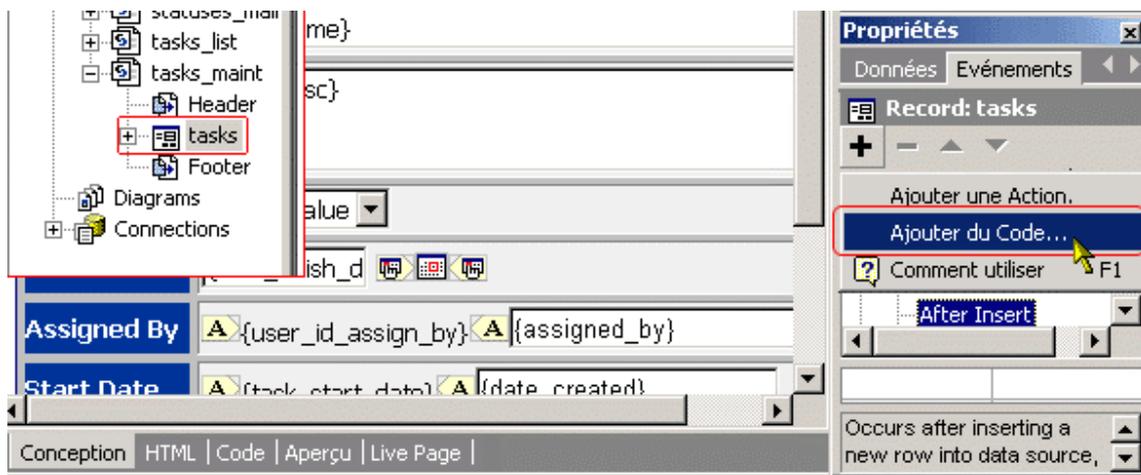
Vous allez apprendre à :

Envoyer des notifications d'email à la personne à qui la tâche est attribuée,

Autoriser seulement la personne à qui la tâche est attribuée à la modifier.

Ajouter du Code dans l'Événement After Insert pour Envoyer des Emails

- 1 - Sélectionnez le formulaire *tasks* en le sélectionnant depuis l'**Explorateur de Projet**, ou en cliquant n'importe où dans la légende du formulaire.
- 2 - Dans la fenêtre **Propriétés**, cliquez sur l'onglet **Événements**.
- 3 - Sélectionnez l'événement **After Insert**.
- 4 - Cliquez sur le bouton **[+]**.
- 5 - Sélectionnez **Ajouter du Code...**



Une fois en mode **Visualisation du Code**, remplacez le commentaire généré :

```
' Write your own code here.'
```

Par le code ci-dessous :

```
String uid = Utils.getUserId(e.getPage());
String from = (String)DBTools.dLookup("emp_name", "employees", "emp_id=" + uid, "IntranetDB");
String to = (String)DBTools.dLookup("email", "employees", "emp_id=" +
    e.getRecord().getControl("user_id_assign_to").getFormattedValue(), "IntranetDB");
String host = "mysmtpost.com";
String subject = "New task for you";
String body = "The following task was submitted:<br><br>" +
    "Task ID:" + DBTools.dLookup("max(task_id)", "tasks", "user_id_assign_by=" + uid, "IntranetDB") +
    "<br><br>" + e.getRecord().getControl("task_desc").getFormattedValue();
SimpleMailer.sendEmail(from, to, subject, body, true, host);
```

Comme vous avez dû vous en rendre compte, le code du dessus permet d'envoyer des emails aux utilisateurs à qui de nouvelles tâches sont attribuées. Voici une explication du code :

```
String uid = Utils.getUserId(e.getPage());
```

Positionne la variable *uid* à la valeur de l'ID utilisateur de l'utilisateur connecté. La fonction statique *Utils.getUserId* est utilisée pour récupérer l'ID utilisateur en cours.

```
String from = (String)DBTools.dLookup("emp_name", "employees", "emp_id=" + uid);
```

Positionne la variable *from* (expéditeur) à la valeur du champ *emp_name* pour l'utilisateur en cours. La fonction statique *DBTools.dLookup* est utilisée pour récupérer la valeur de base de données.

```
String to = (String)DBTools.dLookup("email", "employees", "emp_id=" +
e.getRecord().getControl("user_id_assign_to").getFormattedValue());
```

Positionne la variable *to* (destinataire) à l'email de la personne à qui est attribuée la tâche. La fonction *dLookup* est utilisée ici pour récupérer l'adresse email appropriée.



```
String host = "mysmtphost.com";
```

Le serveur SMTP par lequel l'email sera envoyé.

```
String subject = "New task for you";
```

L'objet de l'email à envoyer.

```
String body = "The following task was submitted:<br><br>" + "Task ID:" + DBTools.dLookUp("max(task_id)",  
"tasks", "user_id_assign_by=" + uid + "<br><br>" +  
e.getRecord().getControl("task_desc").getFormattedValue());
```

La variable *body* (corps) contient le corps de l'email à envoyer. La fonction *dLookUp* est utilisée pour récupérer le plus grand ID de tâche soumis par l'utilisateur connecté (en supposant que les ID de tâche soient créés par incrémentation).

```
SimpleMailer.sendEmail(from, to, subject, body, true, host);
```

Envoie l'email en utilisant les variables créées dans le code du dessus. Le cinquième argument indique que l'email sera envoyé au format HTML (et non pas au format texte brut).

Utiliser l'Événement After Update pour Envoyer des Emails

Vous avez précédemment ajouté le code nécessaire qui envoie une notification d'email à la personne à qui la tâche est attribuée, une fois l'enregistrement d'une nouvelle tâche effectuée dans le système. Maintenant, il nous faut implémenter une fonctionnalité similaire dans l'événement **After Update** pour informer la personne à qui la tâche est attribuée, lorsqu'une tâche existante est mise à jour et réattribuée à quelqu'un d'autre.

- 1 - Cliquez sur le formulaire *tasks* dans l'**Explorateur de Projet**.
- 2 - Dans la fenêtre **Propriétés**, sélectionnez l'onglet **Événements**.
- 3 - Ajoutez le **Custom Code** (Code Personnalisé) dans l'événement **After Update** :

```
String uid = Utils.getUserId(e.getPage());  
if (uid.equals(e.getRecord().getControl("user_id_assign_to").getFormattedValue()))  
{  
    String from = (String)DBTools.dLookUp("email", "employees", "emp_id=" + uid, "IntranetDB");  
    String to = (String)DBTools.dLookUp("email", "employees", "emp_id=" +  
        e.getRecord().getControl("user_id_assign_to").getFormattedValue(), "IntranetDB");  
    String host = "mysmtphost.com";  
    String subject = "A task was assigned to you";  
    String body = "The following task was assigned to you:<br><br>" +  
        "Task ID:" + e.getPage().getParameter("task_id") +  
        "<br><br>" + e.getRecord().getControl("task_desc").getFormattedValue();  
    SimpleMailer.sendEmail(from, to, subject, body, true, host);  
}
```

Les principales différences entre le code ci-dessus et celui utilisé dans l'événement *After Insert* sont les suivantes :



- 1 - Une condition *if* a été ajoutée pour envoyer un email seulement si un utilisateur attribue une tâche à une autre personne.
- 2 - *task_id* est récupéré depuis l'URL en utilisant la fonction *getParameter*. Nous pouvons utiliser cette méthode parce que des tâches peuvent être mises à jour seulement si l'utilisateur est arrivé à la page en cours par une URL qui contient un ID de tâche à mettre à jour. Une telle URL ressemble à : http://localhost/TaskManager/tasks_maint.asp?task_id=9

Test de l'Envoi d'Email

Avant de tester l'envoi d'email :

- 1 - Vous devez ajouter de nouveaux utilisateurs à votre base de données avec des adresses email correctes, ou modifier les utilisateurs existants en changeant leur adresse email :
 - A - Vous pouvez le faire en ouvrant la base de données *Intranet.mdb* qui se trouve dans votre répertoire projet.
 - B - Autrement, vous pouvez utiliser le Gestionnaire de Tâches. Allez à la page Employés pour y visualiser les emails utilisateur et les modifier.
- 2 - Une fois vos utilisateurs configurés avec le test d'emails, sauvegardez votre projet et passez en mode **Live Page** pour tester votre système.

Remarque : Vous aurez besoin de Microsoft Access 2000 ou une version supérieure pour éditer manuellement le fichier base de données. Si votre code email fonctionne correctement, vous devez alors vous retrouver sur la page Liste de Tâches après l'ajout ou la modification d'une tâche, et la personne à qui la tâche a été attribuée doit recevoir un email.

Etape 5

Implémenter la Sécurité d'Enregistrement dans l'Événement After Initialize

Votre Gestionnaire de Tâches est désormais pratiquement complet, seule une fonctionnalité pouvant s'avérer importante manque, à savoir la Sécurité. Dans l'état actuel du gestionnaire, n'importe qui peut modifier et supprimer n'importe quelle tâche. Vous voudrez certainement limiter l'accès pour que seuls les employés autorisés puissent mettre à jour leurs tâches. Il y a plusieurs façons de faire, et nous allons examiner plusieurs d'entre elles.

- 1 - Cliquez sur la page *tasks_maint* dans l'**Explorateur de Projet**.
- 2 - Dans la fenêtre **Propriétés**, sélectionnez l'onglet **Événements**.
- 3 - Ajoutez **Custom Code** (Code Personnalisé) à l'événement **After Initialize** de la page.

Une fois en mode **Visualisation du Code**, remplacez le commentaire généré par :

```
/* Write your own code here. */
```



par le code ci-dessous :

```
String currentTask = e.getPage().getHttpGetParams().getParameter("task_id");
if (!StringUtil.isEmpty(currentTask))
{
    String uid = Utils.getUserId(e.getPage());
    Object task_id = DBTools.dLookup("user_id_assign_to","tasks","task_id="+DBTools.toSql(currentTask,
        JDBCConnection.INTEGER, "IntranetDB"));
    if (!uid.equals(String.valueOf(task_id)))
    {
        e.getPage().getRecord("tasks").setVisible(false);
        // e.getPage().setRedirect("tasks_list.jsp");
        // e.getPage().getRecord("tasks").setAllowUpdate(false);
        // e.getPage().getRecord("tasks").setAllowDelete(false);
    }
}
```

Le code du dessus permet de tester les méthodes suivantes pour implémenter la sécurité d'enregistrement :

Ne pas montrer la tâche (formulaire Enregistrement) sur la page si la tâche sélectionnée n'appartient pas à l'utilisateur connecté. Un utilisateur non autorisé verra une page blanche.

Vous pouvez cacher n'importe quel formulaire sur une page en lui attribuant une valeur *false* (faux) pour la propriété *Visible* (Visible) du formulaire avec la méthode *setVisible*. Le code "*if (!StringUtil.isEmpty(currentTask))*" dans la condition "*if*" indique que le code doit seulement être exécuté si un utilisateur essaie de modifier une tâche existante, mais qui ne lui est pas attribuée. La condition "*if*" s'assure ainsi que tous les utilisateurs peuvent créer de nouvelles tâches. Vous pouvez tester cette fonctionnalité en insérant le code du dessus dans l'événement, en passant en mode **Live Page** et en essayant de modifier une tâche qui ne vous est pas attribuée... et dans ce cas, vous verrez une page vide. Bien qu'une telle fonctionnalité ne soit pas très utile, elle montre comment cacher des formulaires sur une page. Ajoutez un autre formulaire Enregistrement sur la page qui ne puisse pas être mis à jour et qui ait seulement des champs Etiquette qui affichent les informations. Une fois que vous avez deux formulaires sur la page, vous pouvez cacher chaque formulaire par programmation en utilisant des critères d'exclusion.

Rediriger des utilisateurs non autorisés sur une autre page. Seuls les utilisateurs à qui une tâche est attribuée peuvent visualiser la page.



Vous pouvez mettre en application et tester cette fonctionnalité en modifiant légèrement le code du dessus comme montré ci-dessous :

```
String currentTask = e.getPage().getHttpGetParams().getParameter("task_id");
if (!StringUtils.isEmpty(currentTask))
{
    String uid = Utils.getUserId(e.getPage());
    Object task_id = DBTools.dLookUp("user_id_assign_to","tasks","task_id="+DBTools.toSql(currentTask,
        JDBCConnection.INTEGER, "IntranetDB"));
    if (!uid.equals(String.valueOf(task_id)))
    {
        // e.getRecord().setVisible(false);
        e.getPage().setRedirect("tasks_list.jsp");
        // e.getPage().getRecord("tasks").setAllowUpdate(false);
        // e.getPage().getRecord("tasks").setAllowDelete(false);
    }
}
```

Le code du dessus montre que vous devez commenter la ligne précédemment active et activer la ligne qui contient "*setRedirect*".

setRedirect est une méthode utilisée par CodeCharge Studio pour déterminer si la page en cours doit être redirigée sur une autre page, par exemple si un utilisateur n'a pas de session ouverte. Cette variable peut être utilisée seulement sur les pages ayant un accès restreint et qui requièrent une authentification de l'utilisateur. Vous pouvez simplement attribuer la page de destination en utilisant la méthode *setRedirect*, et la page sera alors automatiquement redirigée. Vous pouvez tester cette fonctionnalité en modifiant le code comme montré, en passant en mode **Live Page**, et en essayant alors de modifier une tâche qui ne vous est pas attribuée.

PHP

Extension des Fonctionnalités de l'Application avec la Programmation d'Événements (PHP)

Vous avez certainement remarqué que jusqu'à présent vous avez construit votre application de Gestion de Tâches sans avoir à vous préoccuper du code. En effet, CodeCharge Studio peut vous aider à construire des systèmes fonctionnels sans programmation ; toutefois la création de systèmes plus sophistiqués requiert un minimum de programmation. Heureusement, CodeCharge Studio facilite la programmation en proposant un éditeur de code de haut niveau, en plus des Événements et des Actions qui vous aident à insérer des morceaux de code pré-programmés dans le programme.

Voici les définitions d'une Action et d'un Événement :

Action

Composant de génération de code définissable par l'utilisateur, qui insère un bloc de code dans une procédure événementielle. CodeCharge Studio est fourni avec de nombreuses Actions pré-définies, qui se trouvent dans le dossier suivant : (*CCS folder*)\Components\Actions. Intérieurement, les actions consistent en du code XML et XSL pouvant être facilement personnalisé. Par exemple, une action peut être paramétrée sur un Textbox pour valider une adresse e-mail.

Procédure Événementielle

Une procédure automatiquement exécutée en réponse à un événement lancé par un programme à son exécution. Les événements sont les meilleurs emplacements pour mettre du Code Personnalisé (*Custom Code*).



Connexion à la Base de Données côté Serveur en PHP

Bien que la connexion en mode Conception (via l'utilisateur de l'assistant), quand vous utilisez le PHP puisse être faite à une base de données Microsoft Access, la connexion côté Serveur doit être configurée différemment. La base de donnée la plus utilisée avec le PHP est MySQL ; la connexion côté Serveur doit alors être configurée pour se connecter à MySQL. Avant de connecter la base, vous devez tout d'abord créer la base. Le dossier *exemples* de l'installation de CodeCharge Studio propose un fichier script MySQL, qui peut être utilisé pour recréer les tables de la base de données Intranet.

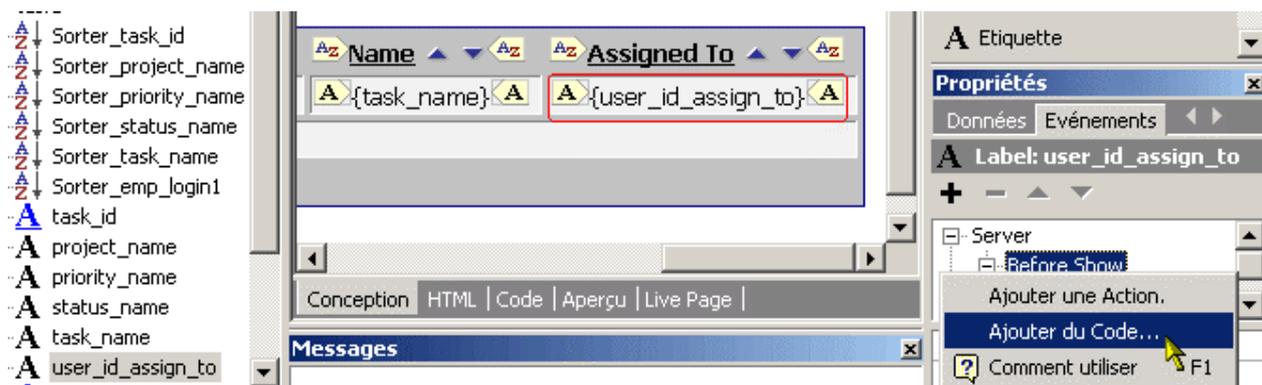
Etape 1

Utiliser l'Événement Before Show Row pour Changer la Couleur du Texte

Démarrons la programmation basique avec une tâche simple de changement de la couleur d'un champ grille sur notre page Liste de Tâches. Concrètement, nous allons mettre en évidence les tâches qui vous sont attribuées en affichant votre nom en bleu dans la grille.

- 1 - Ouvrez la page *tasks_list* dans l'**Explorateur de Projet**.
- 2 - Ouvrez l'arborescence de la grille *tasks*.
- 3 - Faites un clic droit sur le champ *emp_login1* et sélectionnez **Propriétés**.
- 4 - Dans l'onglet **Données**, paramétrez la valeur de la propriété **Name** (Nom) à *user_id_assign_to*.
- 5 - Dans l'**Explorateur Projet**, faites un clic droit sur le champ *task_name* et sélectionnez **Propriétés**.
- 6 - Dans l'onglet **Données**, paramétrez la valeur de la propriété **Contenu** (Contenu) à *HTML*.
- 7 - Sélectionnez la grille *tasks* dans l'**Explorateur de Projet**, ou cliquez n'importe où dans la légende du formulaire.
- 8 - Sélectionnez l'onglet **Événements** dans la fenêtre **Propriétés**.
- 9 - Faites un clic droit sur l'événement **Before Show Row** et sélectionnez **Ajouter du Code...**

L'événement *Before Show Row* se déclenche dans le programme après que les valeurs champs soient attribuées, mais avant d'être générées en HTML. En ajoutant du code dans cet événement, vous pouvez modifier la valeur champ avant qu'elle ne soit montrée.



Contrôler par Programmation la Valeur Champ

Une fois que vous avez ajouté votre Code Personnalisé (Custom Code) à l'événement, vous verrez la fenêtre d'édition du code avec l'emplacement approprié pour entrer le nouveau code.

Remplacez cette ligne de code :

```
//Write your own code here.
```



Par ces lignes (PHP) :

```
global $tasks;
if ($tasks->user_id_assign_to->GetValue() == CCGetSession("UserLogin") && strlen(CCGetSession("UserLogin")))
{
    $tasks->task_name->SetValue("<b><font color= \"blue\">".
    $tasks->task_name->GetValue()."</font></b>");
}
}
```

Ce qui suit est une explication sur le fonctionnement du code PHP du dessus :

```
if ($tasks->user_id_assign_to->GetValue() == CCGetSession("UserLogin") &&
strlen(CCGetSession("UserLogin")))
```

C'est une condition *if* (si) qui est vraie seulement si la valeur du champ *user_id_assign_to* est égale au nom du login de l'employé connecté. Le champ de base de données *employees1_emp_login* fournit la valeur pour le champ *user_id_assign_to* dans le formulaire Grille. Une fois connecté au système, le programme reconnaît vos tâches en comparant votre nom de login à la valeur *emp_login* de la personne à qui une tâche est attribuée. *UserLogin* est une variable de session utilisée par les programmes générés par CodeCharge, et cette variable conserve le nom de Login de la session utilisateur en cours jusqu'à son expiration.

Note :

Il s'agit de variable de session créées par défaut par CodeCharge Studio :

UserID : la valeur champ clé primaire de la session utilisateur

UserLogin : le nom de Login de l'utilisateur connecté au système

GroupID : le niveau/groupe de sécurité de l'utilisateur connecté au système

```
$tasks->task_name->SetValue("<b><font color= \"blue\">". $tasks->task_name->
GetValue()."</font></b>");
```

Le code est exécuté si la condition précédente *if* est rencontrée. Elle modifie la valeur du champ *task_name*. La valeur du champ est remplacée par sa valeur de base de données encapsulée dans du code HTML qui spécifie la couleur de la police, comme le bleu, et ajoute un tag HTML ** pour afficher une police en gras. De plus, vous remarquerez que le code est orienté objet et vous indiquez que vous voulez attribuer une valeur au champ *task_name* dans la grille *tasks*. *SetValue* est une méthode d'un objet, pouvant être utilisée pour modifier la valeur d'un objet.

Prévisualiser la Page Liste de Tâches

- 1 - Sauvegardez votre projet.
- 2 - Passez en mode **Live Page** pour visualiser votre page fonctionnelle. Si aucun nom de tâche n'est mis en évidence dans la colonne "Name", c'est que vous n'êtes probablement pas connecté(e) au système.
- 3 - Le menu ne contenant pas encore de lien à la page Login, vous pouvez y accéder en essayant d'ouvrir une des pages restreintes, par exemple la page Maintenance de Tâches.
- 4 - Cliquez sur n'importe lequel des ID projet, et vous devriez voir la page Login.
- 5 - Connectez-vous en tant que **george/george**, puis cliquez sur le lien **Tasks** sur le menu pour retourner à la page Liste de Tâches.

Maintenant, vous devriez voir que certaines tâches sont mises en évidence ; ces tâches correspondent aux tâches de l'utilisateur pour lequel vous avez ouvert une session.



[Employees](#) [Priorities](#) [Projects](#) [Statuses](#) [Tasks](#)

Rechercher Tasks

Mot clé	<input type="text"/>
Projet	Rechercher par valeur ▼
<input type="button" value="Rechercher"/>	

Liste de : Tasks

<u>Id</u>	<u>Project</u>	<u>Priority</u>	<u>Status</u>	<u>Name</u>	<u>Assigned To</u>
1	CodeCharge	High	On hold	Great Project needs to be greater	helen
2	CodeCharge	Highest	Closed	Fix ALL bugs	george
3	CodeCharge	High	Closed	Get ready to click	peter
4	My Project	Highest	Open	Finish My Project	ignace
5	Test Project	High	In progress	Test this project.	ken
6	CodeCharge	Highest	Open	Code with one hand.	alexander
7	Test Project	Highest	On hold	Get armed	helen
8	Test Project	Highest	Open	Write more code	ignace
9	Super Project	Highest	In progress	Code, code, code...	george
10	Test Project	Lowest	On hold	Sleep	ken
11	Super Project	Highest	Open	Have fun	alexander

Nouveau 1 of 1

Etape 2

Modifier un Champ Etiquette sur la Page Maintenance de Tâches

Maintenant procédons à une modification nécessaire sur la page Maintenance de Tâches sur laquelle vous avez dû remarqué que le champ Etiquette *Assigned By* n'affiche pas le nom de l'employé, mais l'ID, comme montré ci-dessous ; c'est parce que la table *tasks* contient seulement l'ID utilisateur, alors que la table *employees* contient les noms des utilisateurs réels.



[Employees](#) [Priorities](#) [Projects](#) [Statutes](#) [Tasks](#)

Ajouter/Editer Tasks

Project	CodeCharge
Priority	Highest
Status	Closed
Type	Question
Name	Fix ALL bugs
Desc	Staying up at night coding? Get CodeCharge, go home, get rest.
Assigned To	george
Finish Date	03.02.2003
Assigned By	3
Start Date	02.02.2003
<input type="button" value="Confirmer"/> <input type="button" value="Supprimer"/>	

Il y a plusieurs méthodes possibles pour traiter de la question. Merci de vous référer aux explications qui suivent :

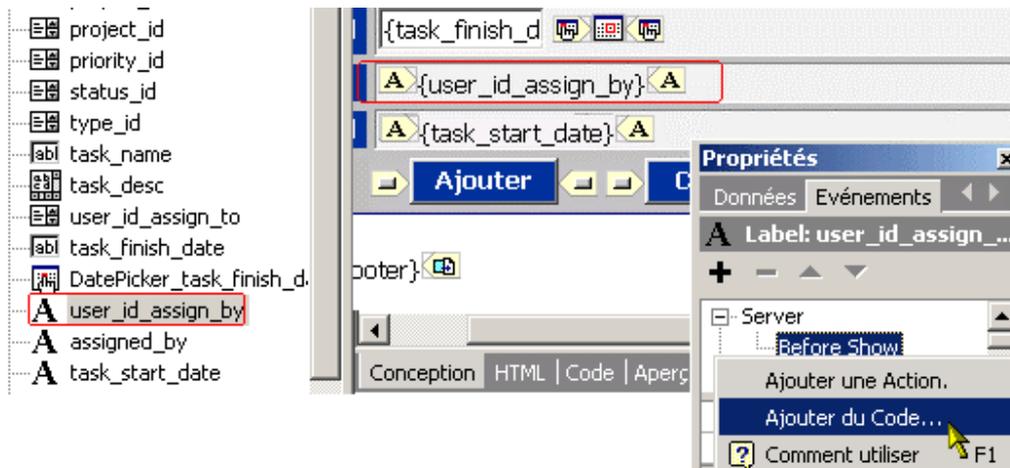
1 - Créez une Requête qui contient plusieurs tables et qui peut être utilisée en tant que source de données par le formulaire Enregistrement, comme vous l'avez déjà fait avec la grille sur la page Liste de Tâches. Malheureusement, les requêtes qui contiennent plusieurs tables ne peuvent pas être mises à jour de par leur nature, et donc le formulaire Enregistrement entier peut s'arrêter de fonctionner. En d'autres termes, si vous avez indiqué que vous vouliez utiliser une requête contenant une table *tasks* et *employees* dans votre formulaire Enregistrement, alors si vous attribuez une tâche à quelqu'un d'autre, le programme ne pourra pas savoir si vous vouliez mettre à jour la table *tasks* avec le nouveau *employee_id*, ou si vous vouliez mettre à jour la table *employees* et changer le nom de l'employé.

Ainsi, si vous utilisez plusieurs tables comme source de données pour le formulaire Enregistrement, vous devrez également définir les opérations *Custom Insert*, *Custom Update* et *Custom Delete* dans les propriétés du formulaire Enregistrement pour indiquer quels champs de la base de données devront être mis à jour avec les valeurs correspondantes saisies sur la page. Cette méthode semble représenter beaucoup d'effort juste pour afficher une valeur supplémentaire sur la page.

2 - Utilisez une Procédure Événementielle pour insérer du Code Personnalisé (*Custom Code*) où vous pourrez produire la valeur désirée par programmation. Cette méthode est très flexible, dans la mesure où elle vous permet d'étendre le code généré par l'ajout de votre propre code. L'étape qui suit décrit en détail cette méthode.

Utiliser l'Événement Before Show pour Changer la Valeur d'une Etiquette

- 1 - En mode **Conception**, sélectionnez l'Etiquette *user_id_assign_by*,
 - 2 - Dans la fenêtre **Propriétés**, cliquez sur l'onglet **Données**.
 - 3 - Sélectionnez *Text* pour la propriété **Data Type** (Type de Données).
 - 4 - Dans la fenêtre **Propriétés**, cliquez sur l'onglet **Événements**.
 - 5 - Faites un clic droit sur l'événement **Before Show** et sélectionnez **Ajouter du Code....**
- CodeCharge Studio va alors automatiquement passer en mode **Visualisation du Code**.



Une fois en mode **Visualisation du Code**, si vous générez du PHP, vous devriez voir le fichier `tasks_maint_events.php`, avec les lignes de code suivantes :

```
//Custom Code @28-73254650
//-----
//Write your own code here.
//-----
//End Custom Code
```

Remplacez le texte :

```
// Write your own code here.
```

Par :

```
global $tasks;
global $DBIntranetDB;
if ($tasks->EditMode)
{
    $tasks->user_id_assign_by->SetValue(CCDLookUp("emp_name", "employees", " emp_id=" .
    $DBIntranetDB->ToSQL($tasks->user_id_assign_by->GetValue(), ccsInteger), $DBIntranetDB));
}
else
{
    $tasks->user_id_assign_by->SetValue(CCDLookUp("emp_name", "employees", " emp_id=" .
    $DBIntranetDB->ToSQL(CCGetUserID(), ccsInteger), $DBIntranetDB));
}
```

Le code du dessus comprend les éléments suivants :

`tasks` : le nom du formulaire Enregistrement sur la page

`EditMode` : propriété du formulaire, qui indique si l'enregistrement peut être édité. En fonction de la valeur de cette propriété, nous affichons soit le nom de la personne qui a initialement soumis la tâche (mode Edition), ou de la personne qui soumet la tâche (mode Insertion).



`user_id_assign_by` : le nom de l'Etiquette dans la grille et en même temps le nom du champ de base de données qui a été utilisé pour créer cette Etiquette, et qui est maintenant sa source de données.

`SetValue` : une méthode d'un objet (dans ce cas l'Etiquette), qui peut être utilisée pour modifier la valeur de l'objet.

`$tasks->user_id_assign_by->SetValue` : méthode `SetValue` entièrement qualifiée, qui dit au programme à quel objet il se réfère. En d'autres termes, c'est la méthode `SetValue` qui affecte le champ `user_id_assign_by`, qui en retour appartient à la grille `tasks`.

`CCDLookup` : fonction de CodeCharge qui supporte la récupération d'une valeur de base de données sur un nom de champ, un nom de table, et une condition. Ici, cette fonction récupère le nom de l'Employé (`emp_name`) dans la table `employees`, en utilisant la condition suivante : la clé (`emp_id`) est égale à la valeur courante de l'Etiquette.

`ToSQL` : fonction de CodeCharge qui convertit une valeur en un format supporté par la base de données. Cette fonction requiert un paramètre qui dit si une valeur doit être convertie en un nombre (*Integer*) ou en du texte. Dans ce cas, cette fonction convertit la valeur courante de l'Etiquette en un nombre pouvant être utilisé avec la fonction `CCDLookup`. Il est conseillé de toujours utiliser cette fonction avec `CCDLookup`.

`$DBIntranetDB` : le nom de l'objet qui définit la connexion base de données que vous voulez utiliser avec la fonction `CCDLookup`.

`CCGetUserID()` : fonction de CodeCharge qui retourne l'ID de l'utilisateur connecté au système.

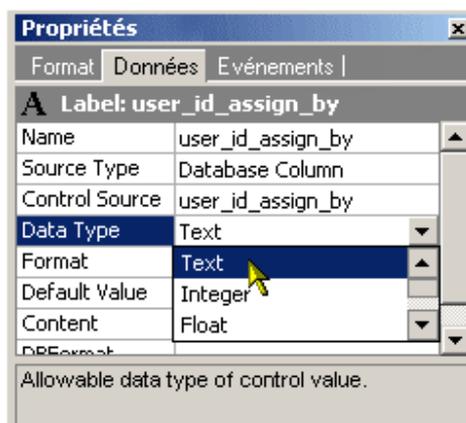
Le code se lit approximativement comme suit :

If a record is being edited : Attribue le nom de la personne qui a initialement soumis la question à l'Etiquette `user_id_assign_by`, par la recherche du nom de l'employé dans la table `employees`, en utilisant la fonction `CCDLookup` qui utilise la connexion `IntranetDB` et la valeur de l'Etiquette `user_id_assign_by`.

If a new record is being created : Attribue un utilisateur connecté à l'Etiquette `user_id_assign_by` en récupérant son nom dans la table `employees` et en utilisant la fonction `CCDLookup` qui utilise la connexion `IntranetDB` et la fonction `CCGetUserID` (qui permet d'obtenir l'ID de l'utilisateur connecté).

Maintenant que vous avez modifié par programmation la valeur de l'Etiquette `user_id_assign_by` pour sortir le Nom de l'Employé au lieu de l'ID, vous allez devoir indiquer que ce champ est désormais un champ Texte, et non un champ Numérique.

- 1 - Cliquez sur l'onglet **Données** dans la fenêtre **Propriétés**.
- 2 - Sélectionnez `Text` en tant que **Data Type** (Type de Données).



Etape 3

Ajouter un Champ Caché "Assigned By" pour la Mise à Jour Automatique de Nouvelles Tâches

Vous avez précédemment utilisé l'événement `Before Show` pour afficher le nom de la personne à qui est assignée une tâche. Cependant, les champs Etiquette ne peuvent pas être mis à jour de par leur nature, alors bien que le nom de l'employé soit affiché sur la page, il n'est pas écrit dans la base de données. Comme nous voulons que la base de données enregistre le nom ou l'ID de la personne qui soumet une tâche, nous allons devoir ajouter la logique de programmation pour effectuer cela.



1 - Ajoutez un champ **Caché** à votre page depuis l'onglet **Formulaires** de la **Boîte à Outils** (*Toolbox*).

Ce type de champ n'est pas visible dans le navigateur, mais il sera utilisé pour stocker une valeur et mettre à jour la base de données.

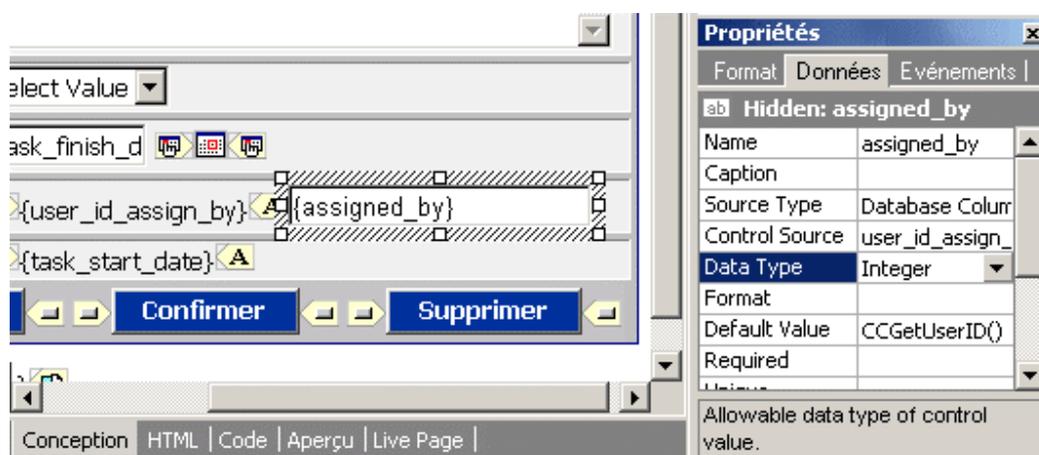
2 - Configurez le nouveau champ en paramétrant ses propriétés comme indiqué ci-dessous :

Name (Nom) : *assigned_by* - le nom du champ Caché qui vient d'être ajouté. Cela peut être n'importe quel nom.

Control Source : *user_id_assign_by* - le champ/colonne de la base de données qui sera utilisé(e) pour récupérer la valeur du champ et qui sera mis(e) à jour avec la nouvelle valeur en cas de changement.

Data Type (Type de Données) : *Integer* - le type de la valeur liée au *Control Source*. Nos ID utilisateur/employé sont numériques.

Default (par Défaut) : *CCGetUserID()* - *CCGetUserID()* est une fonction de CodeCharge qui récupère l'ID de l'utilisateur connecté au système. Ainsi vous pouvez facilement indiquer que vous voulez enregistrer l'ID de l'utilisateur dans le champ *user_id_assign_by* pour chaque nouvelle tâche qui sera soumise.



Ajouter un Champ Caché "Date Created" au Formulaire Enregistrement

Maintenant, nous allons ajouter un autre champ Caché à votre page, qui sera utilisé pour soumettre la date et l'heure courantes dans le champ *task_start_date* dans la base de données.

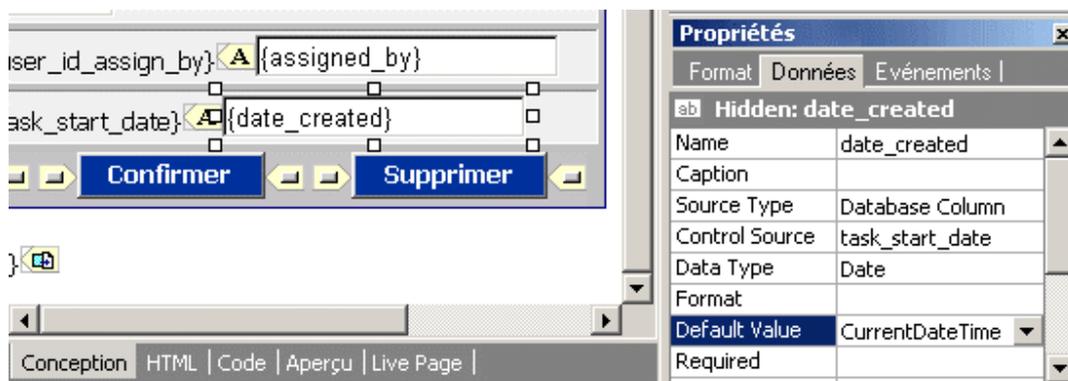
1 - Configurez le nouveau champ comme indiqué ci-dessous :

Name (Nom) : *date_created*

Control Source : *task_start_date*

Data Type (Type de Données) : *Date*

Default (Par Défaut) : *CurrentDateTime* - La valeur *CurrentDateTime* vous permet d'attribuer automatiquement la date et l'heure courantes aux nouvelles tâches. La propriété Par Défaut n'affecte pas les enregistrements existants, donc la date et l'heure des tâches existantes ne seront pas modifiées lors de mises à jour ultérieures.



- 1 - Cliquez sur le champ `task_start_date` dans l'Explorateur de Projet.
 - 2 - Dans la fenêtre **Propriétés**, paramétrez **Default Value** (Valeur Par Défaut) à `CurrentDateTime`.
- C'est ainsi que le champ Etiquette peut afficher la date bien que le champ Caché ne soit pas visible pour l'utilisateur.

Tester les Champs Etiquette et Caché

Pour terminer :

- 1 - Passez en mode **Live Page**.
- 2 - Sélectionnez ou ajoutez une Tâche, et vous voyez alors votre Etiquette afficher le nom de la personne qui a attribué la tâche.
- 3 - La version de base de votre Gestionnaire de Tâches est désormais finalisée. Ne pas oublier de sauvegarder !

[Employees](#) [Priorities](#) [Projects](#) [Statuses](#) [Tasks](#)

Ajouter/Editer Tasks

Project	My Project
Priority	Highest
Status	Closed
Type	Task
Name	Finish my project
Desc	Implement "Assigned By" Label to show values from another table
Assigned To	alexander
Finish Date	
Assigned By	George Pennington
Start Date	08.04.2003 16:48:07
Ajouter	



Etape 4

Programmation du Formulaire Enregistrement

Maintenant que vous avez créé une application simple de Gestion de Tâches, vous souhaitez certainement l'étendre pour qu'elle soit à la fois plus pratique et plus utile... Dans cette section, vous allez avoir un aperçu sur la manière d'implémenter des applications pratiques et sophistiquées par l'ajout de code et d'actions, ceci afin d'offrir des fonctionnalités supplémentaires à votre Gestionnaire de Tâches.

Vous allez apprendre à :

Envoyer des notifications d'email à la personne à qui la tâche est attribuée,

Autoriser seulement la personne à qui la tâche est attribuée à la modifier.



Ajouter du Code dans l'Événement After Insert pour Envoyer des Emails

- 1 - Sélectionnez le formulaire *tasks* dans la page *tasks_maint* en le sélectionnant depuis l'**Explorateur de Projet**, ou en cliquant n'importe où dans la légende du formulaire.
- 2 - Dans la fenêtre **Propriétés**, cliquez sur l'onglet **Événements**.
- 3 - Sélectionnez l'événement **After Insert**.
- 4 - Cliquez sur le bouton **[+]**.
- 5 - Sélectionnez **Ajouter du Code...**



Une fois en mode **Visualisation du Code**, remplacez le commentaire généré :

```
// Write your own code here.
```



Par le code du dessus :

```
global $DBIntranetDB;
global $tasks;
$from_name = CCDLookUp("emp_name", "employees", "emp_id=".
$DBIntranetDB->ToSQL(CCGetUserID(), ccsInteger), $DBIntranetDB);
$from_email=CCDLookUp("email", "employees", "emp_id=".
$DBIntranetDB->ToSQL(CCGetUserID(), ccsInteger), $DBIntranetDB);
$to_email=CCDLookUp("email", "employees", "emp_id=".
$DBIntranetDB->ToSQL($tasks->user_id_assign_to->GetValue(), ccsInteger), $DBIntranetDB);

$headers = "From: ".$from_name."<".$from_email.">";
$headers .= "Content-Type: text/html";
$subject = "New task for you";
$message = "The following task was submitted:<br><br>".
    "Task ID: ".CCDLookUp("max(task_id)", "tasks", "user_id_assign_by=".
    $DBIntranetDB->ToSQL(CCGetUserID(), ccsInteger), $DBIntranetDB)."<br><br>".
    $tasks->task_desc->GetText();

mail$to_email,
$subject, $message, $headers);
```

Comme vous pouvez le constater, le code ci-dessus envoie des emails aux utilisateurs à qui de nouvelles tâches sont attribuées.

Ce qui suit est une explication du code :

```
$from_name = CCDLookUp("emp_name", "employees", "emp_id=".$DBIntranetDB->ToSQL (CCGetUserID(),
ccsInteger),$DBIntranetDB);
```

Paramètre *from_name* à la valeur du champ *emp_name* pour l'utilisateur connecté.

```
$from_email = CCDLookUp("email", "employees", "emp_id=".$DBIntranetDB->ToSQL(CCGetUserID(),
ccsInteger), $DBIntranetDB);
```

Paramètre *from_email* à la valeur du champ *email* dans la table *employees* où *emp_id* correspond à l'utilisateur connecté. La fonction *CCDLookUp* est utilisée pour récupérer une valeur de base de données, alors que *CCGetUserID* retrouve l'ID de la connexion utilisateur en cours.

```
$to_email = CCDLookUp("email", "employees", "emp_id=".$DBIntranetDB->ToSQL($tasks->user_id_assign_to-
>GetValue(), ccsInteger), $DBIntranetDB);
```



Paramètre *to_email* à l'email de la personne à qui la tâche est attribuée. La fonction *CCDLookup* est utilisée ici pour récupérer l'adresse email appropriée.

```
$headers = "From: ".$from_name."<".$from_email.">";
```

Ajoute des détails sur l'expéditeur dans l'en-tête d'email

```
$headers .= "Content-Type: text/html;";
```

Indique que l'email sera envoyé au format HTML, et non pas au format texte brut.

```
$subject = "New task for you";
```

Le sujet de l'email à envoyer.

```
$message = "The following task was submitted:<br><br>". CCDLookup("max(task_id)", "tasks",  
"user_id_assign_by=".$DBIntranetDB->ToSQL(CCGetUserID, ccsInteger), $DBIntranetDB). "<br><br>".  
$tasks->task_desc->GetText();
```

La variable *message* contient le corps de l'email qui sera envoyé. La fonction *CCDLookup* est utilisée pour récupérer le plus grand ID de tâche soumis par l'utilisateur connecté (en supposant que les ID de tâche soient créés par incrémentation).

```
mail ($to_email, $subject, $message, $headers);
```

Envoie l'email en utilisant les variables créées dans le code du dessus.

Utiliser l'Événement After Update pour Envoyer des Emails

Vous avez précédemment ajouté le code nécessaire qui envoie une notification d'email à la personne à qui la tâche est attribuée, une fois l'enregistrement d'une nouvelle tâche effectuée dans le système. Maintenant, il nous faut implémenter une fonctionnalité similaire dans l'événement *After Update* pour informer la personne à qui la tâche est attribuée, lorsqu'une tâche existante est mise à jour et réattribuée à quelqu'un d'autre.

- 1 - Cliquez sur le formulaire *tasks* dans l'**Explorateur de Projet**.
- 2 - Dans la fenêtre **Propriétés**, sélectionnez l'onglet **Événements**.



3 - Ajoutez le **Custom Code** (Code Personnalisé) dans l'événement **After Update** :

```
global $DBIntranetDB;
global $tasks;
if (CCGetUserID() != $tasks->user_id_assign_to->GetValue())
{
    $from_name = CCDLookUp("emp_name", "employees", "emp_id=" .
    $DBIntranetDB->ToSQL(CCGetUserID(), ccsInteger), $DBIntranetDB);
    $from_email = CCDLookUp("email", "employees", "emp_id=" .
    $DBIntranetDB->ToSQL(CCGetUserID(), ccsInteger), $DBIntranetDB);
    $to_email = CCDLookUp("email", "employees", "emp_id=" .
    $DBIntranetDB->ToSQL($tasks->user_id_assign_to->GetValue(), ccsInteger), $DBIntranetDB);
    $headers = "From: ".$from_name."<".$from_email.">";
    $headers .= "Content-Type: text/html;";
    $subject = "A task was assigned to you";
    $message = "The following task was assigned to you:<br><br>".
        "Task ID: ".CCGetFromGet ("task_id", " ")."<br><br>".
    $tasks->task_desc->GetText();
    mail($to_email, $subject, $message, $headers);
}
```

Les principales différences entre le code ci-dessus et celui utilisé dans l'événement *After Insert* sont les suivantes :

- 1 - Une condition *if* a été ajoutée pour envoyer un email seulement si un utilisateur attribue une tâche à une autre personne.
- 2 - *task_id* est récupéré depuis l'URL utilisant la fonction *CCGetFromGet*. Nous pouvons utiliser cette méthode parce que des tâches peuvent être mises à jour seulement si l'utilisateur est arrivé à la page en cours par une URL qui contient un ID de tâche à mettre à jour. Une telle URL ressemble à : http://localhost/TaskManager/tasks_maint.php?task_id=9

Test de l'Envoi d'Email

Avant de tester l'envoi d'email :

1 - Vous devez ajouter de nouveaux utilisateurs à votre base de données avec des adresses email correctes, ou modifier les utilisateurs existants en changeant leur adresse email :

A - Vous pouvez le faire en ouvrant la base de données *Intranet.mdb* qui se trouve dans votre répertoire *Projet*.

B - Autrement, vous pouvez utiliser le Gestionnaire de Tâches. Allez à la page Employés pour y visualiser les emails utilisateur et les modifier.

2 - Une fois vos utilisateurs configurés avec le test d'emails, sauvegardez votre projet et passez en mode **Live Page** pour tester votre système.

Remarque : Vous aurez besoin de Microsoft Access 2000 ou une version supérieure pour éditer manuellement le fichier base de données. Si votre code email fonctionne correctement, vous devez alors vous retrouver sur la page Liste de Tâches après l'ajout ou la modification d'une tâche, et la personne à qui la tâche a été attribuée doit recevoir un email.



Etape 5

Implémenter la Sécurité d'Enregistrement dans l'Événement After Initialize

Votre Gestionnaire de Tâches est désormais pratiquement complet, seule une fonctionnalité pouvant s'avérer importante manque, à savoir la Sécurité. Dans l'état actuel n'importe qui peut modifier et supprimer n'importe quelle tâche. Vous voudrez certainement limiter l'accès pour que seuls les employés autorisés puissent mettre à jour leurs tâches. Il y a plusieurs façons de procéder, et nous allons examiner plusieurs d'entre elles.

- 1 - Cliquez sur la page `tasks_maint` dans l'**Explorateur de Projet**.
- 2 - Dans la fenêtre **Propriétés**, sélectionnez l'onglet **Événements**.
- 3 - Ajoutez **Custom Code** (Code Personnalisé) à l'événement **After Initialize** de la page. Procédez comme suit :

Une fois en mode **Visualisation de Code**, remplacez le commentaire généré :

```
// Write your own code here.
```

Par ce code :

```
global $tasks;
global $Redirect;
global $DBIntranetDB;
$current_task = CCGetParam("task_id", "");
if ($current_task != 0 && CCGetUserID() != CCDLookUp("user_id_assign_to", "tasks", "task_id=" .
$DBIntranetDB->ToSQL($current_task, ccsInteger), $DBIntranetDB))
{
    $tasks->Visible = false;
    // $Redirect = "tasks_list.php";
    // $tasks->UpdateAllowed = false;
    // $tasks->DeleteAllowed = false;
}
```

Le code du dessus permet de tester les méthodes suivantes pour implémenter la sécurité d'enregistrement :

Ne pas montrer la Tâche (formulaire Enregistrement) sur la page si la tâche sélectionnée n'appartient pas à l'utilisateur connecté. Un utilisateur non autorisé verra une page blanche.

Vous pouvez cacher n'importe quel formulaire sur une page en lui attribuant une valeur *False* (Faux) pour la propriété *Visible* (visible) du formulaire. Le code `$current_task != 0` dans la condition *if* indique que le code doit seulement être exécuté si l'utilisateur essaie de modifier une tâche existante et qui ne lui est pas attribuée. La condition *if* s'assure aussi que tous les utilisateurs peuvent créer de nouvelles tâches. Vous pouvez tester cette fonctionnalité en insérant le code du dessus dans l'événement, puis en passant en mode **Live Page**, et en essayant de modifier une tâche qui ne vous est pas attribuée... et alors une page vide devrait s'afficher. Une telle fonctionnalité n'est pas très utile, mais elle montre comment cacher des formulaires sur une page. Ajoutez un autre formulaire Enregistrement sur votre page qui ne puisse pas être mis à jour et qui ait seulement des champs Etiquette qui montrent les informations. Une fois que vous avez deux formulaires sur la page, vous pouvez cacher chaque formulaire par programmation en utilisant des critères d'exclusion.



Rediriger des utilisateurs non autorisés sur une autre page. Seuls les utilisateurs à qui sont attribuées des tâches peuvent visualiser la page.

Vous pouvez mettre en application et tester cette fonctionnalité en modifiant légèrement le code du dessus, comme montré ci-dessous :

```
global $tasks;
global $Redirect;
global $DBIntranetDB;
$current_task = CCGetParam("task_id", "");
if($current_task != 0 && CCGetUserID() != CCDLookUp("user_id_assign_to", "tasks", "task_id=" .
$DBIntranetDB->ToSQL($current_task, ccsInteger), $DBIntranetDB))
{
    // $tasks->Visible = false;
    $Redirect = "tasks_list.php";
    // $tasks->UpdateAllowed = false;
    // $tasks->DeleteAllowed = false;
}
```

Le code du dessus montre que vous devez commenter la ligne précédemment active, et activer la ligne qui commence avec *\$Redirect*. *\$Redirect* est une variable utilisée par CodeCharge Studio pour déterminer si la page en cours doit être redirigée sur une autre page, par exemple dans le cas où un utilisateur n'a pas de session ouverte. Cette variable peut être utilisée seulement sur les pages ayant un accès restreint et nécessitant une authentification. Vous pouvez simplement attribuer la page de destination à la variable *\$Redirect*, et alors la page sera automatiquement redirigée. Testez cette fonctionnalité en modifiant le code comme montré, passez en mode **Live Page**, et essayez de modifier une tâche qui ne vous est pas attribuée.

ColdFusion

Extension des Fonctionnalités de l'Application avec la Programmation d'Événements (ColdFusion)

Vous avez certainement remarqué que jusqu'à présent vous avez construit votre Gestionnaire de Tâches sans avoir à vous préoccuper du code. En effet, CodeCharge Studio peut vous aider à construire des systèmes fonctionnels sans programmation ; toutefois la création de systèmes plus sophistiqués requiert un minimum de programmation. Heureusement, CodeCharge Studio facilite la programmation en proposant un éditeur de code de haut niveau, en plus des Événements et des Actions qui vous aident à insérer des morceaux de code pré-programmés dans le programme.

Voici les définitions d'une Action et d'un Événement :

Action

Composant de génération de code définissable par l'utilisateur, qui insère un bloc de code dans une procédure événementielle. CodeCharge Studio est fourni avec de nombreuses Actions pré-définies, qui se trouvent dans le dossier suivant : (*CCS folder*)\Components\Actions. Intérieurement, les actions consistent en du code XML et XSL pouvant être facilement personnalisé. Par exemple, une action peut être paramétrée sur un Textbox pour valider une adresse e-mail.

Procédure Événementielle

Une procédure automatiquement exécutée en réponse à un événement lancé par un programme à son exécution. Les événements sont les meilleurs emplacements pour mettre du Code Personnalisé (*Custom Code*).

Connexion à la Base de Données Côté Serveur en ColdFusion



Bien que la connexion en mode Conception (via l'utilisation de l'assistant), quand vous utilisez ColdFusion, puisse être faite directement à une base de données Access, la connexion côté Serveur doit être configurée différemment. Dans ce cas de ColdFusion, la connexion côté Serveur doit être faite à une source de données enregistrée. Si vous utilisez ColdFusion 4.01, vous aurez besoin de faire la connexion en utilisant ODBC DSN. Pour des versions supérieures de ColdFusion (version 5.0 ou MX), vous pouvez utiliser n'importe quelle source de données qui a été enregistrée dans l'administrateur ColdFusion. Pour plus d'informations sur les connexions base de données en ColdFusion, merci de vous référer à la section sur les connexions côté Serveur en ColdFusion.

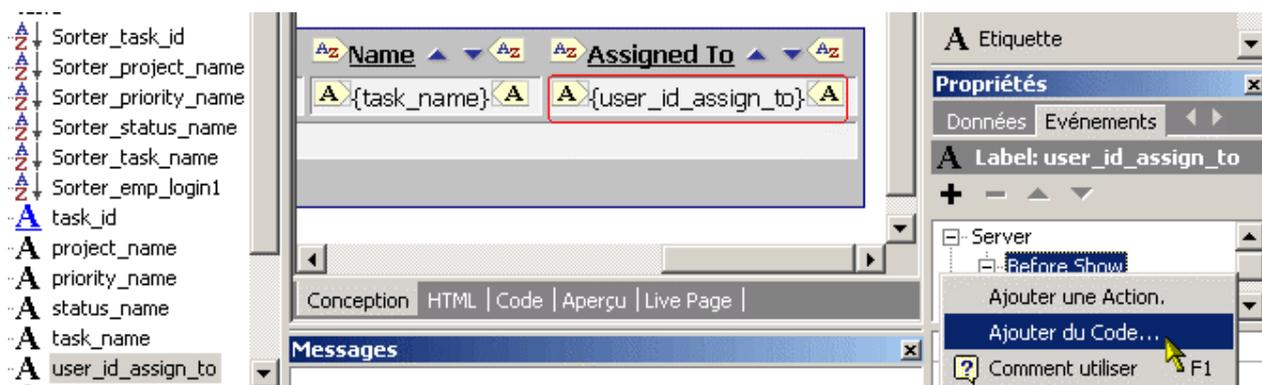
Etape 1

Utiliser l'Événement Before Show Row pour Changer la Couleur du Texte

Démarrons la programmation basique avec une tâche simple de changement de la couleur d'un champ grille sur notre page Liste de Tâches. Concrètement, nous allons mettre en évidence les tâches qui vous sont attribuées en affichant votre nom en bleu dans la grille.

- 1 - Ouvrez la page `tasks_list` dans l'**Explorateur de Projet**.
- 2 - Ouvrez l'arborescence de la grille `tasks`.
- 3 - Faites un clic droit sur le champ `emp_login1` et sélectionnez **Propriétés**.
- 4 - Dans l'onglet **Données**, paramétrez la valeur de la propriété **Name** (Nom) à `user_id_assign_to`.
- 5 - Dans l'**Explorateur Projet**, faites un clic droit sur le champ `task_name` et sélectionnez **Propriétés**.
- 6 - Dans l'onglet **Données**, paramétrez la valeur de la propriété **Contenu** (Contenu) à `HTML`.
- 7 - Sélectionnez la grille `tasks` dans l'**Explorateur de Projet**, ou cliquez n'importe où dans la légende du formulaire.
- 8 - Sélectionnez l'onglet **Événements** dans la fenêtre **Propriétés**.
- 9 - Faites un clic droit sur l'événement **Before Show Row** et sélectionnez **Ajouter du Code...**

L'événement *Before Show Row* se déclenche dans le programme après que les valeurs champs soient attribuées, mais avant d'être générées en HTML. En ajoutant du code dans cet événement, vous pouvez modifier la valeur champ avant qu'elle ne soit montrée.



Contrôler par Programmation la Valeur Champ

Une fois que vous aurez ajouté le Code Personnalisé (*Custom Code*) à l'événement, vous verrez la fenêtre d'édition du code avec l'emplacement approprié pour entrer le nouveau code.

Remplacez cette ligne de code :

```
<!-- write your own code here -->
```



par les lignes suivantes (ColdFusion) :

```
<CFPARAM Name="Session.UserLogin" Default="">
<CFIF flduser_id_assign_to EQ Session.UserLogin AND Session.UserLogin NEQ "">
    <CFSET fldtask_name= '<b><font color="blue">' & fldtask_name & '</font></b>'>
</CFIF>
```

Ce qui suit est une explication sur le fonctionnement du code du dessus :

```
<CFPARAM Name="Session.UserLogin" Default="">
```

Ce code vérifie l'existence de la variable de session *UserLogin*. Si la variable n'existe pas, elle est créée et ses valeurs paramétrées sur une chaîne vide.

```
<CFIF fldtask_name EQ Session.UserLogin AND Session.UserLogin NEQ "">
```

Cette instruction conditionnelle est vraie seulement si la valeur du champ *fldtask_name* est égale au nom de Login de l'employé connecté au système. Le champ de base de données *employees1_emp_login* fournit la valeur pour le champ *fldtask_name* dans le formulaire Grille. Une fois connecté au système, le programme reconnaîtra vos tâches en comparant votre nom de Login à la valeur *emp_login* de la personne à qui la tâche est attribuée. *UserLogin* est une des variables de session utilisées par les programmes générés par CodeCharge, et elle conserve le nom de Login de la session utilisateur en cours jusqu'à son expiration.

Note :

Il s'agit des variables de session créées par défaut par CodeCharge Studio :

UserID : la valeur champ clé primaire de la session utilisateur

UserLogin : le nom de Login de l'utilisateur connecté au système

GroupID : le niveau/groupe de sécurité de l'utilisateur connecté au système

```
<CFSET fldtask_name= '<b><font color="blue">' & fldtask_name & '</font></b>'>
```

Ce code est exécuté si la condition précédente *if* (si) est rencontrée. Elle modifie la valeur du champ *task_name*. La valeur champ est remplacée par sa valeur de base de données encapsulée dans du code HTML, qui spécifie la couleur de police, comme le bleu, et ajoute un tag HTML ** pour afficher la police en gras.

```
</CFIF>
```

Cette ligne manque la fin de la condition *if*, aussi l'exécution du code restant n'est pas affectée par cette condition.

Prévisualiser la Page Liste de Tâches

- 1 - Sauvegardez votre projet.
- 2 - Passez en mode **Live Page** pour visualiser votre page fonctionnelle. Si aucun nom de tâche n'est mis en évidence dans la colonne "*Name*", c'est que vous n'êtes probablement pas connecté(e) au système.
- 3 - Le menu ne contenant pas encore de lien à la page Login, vous pouvez y accéder en essayant d'ouvrir une des pages restreintes, par exemple la page Maintenance de Tâches.
- 4 - Cliquez sur n'importe lequel des ID projet, et vous devriez voir la page Login.
- 5 - Connectez-vous en tant que **george/george**, puis cliquez sur le lien **Tasks** sur le menu pour retourner à la page Liste de Tâches.

Maintenant, vous devriez voir que certaines tâches sont mises en évidence ; ces tâches correspondent aux tâches de l'utilisateur pour lequel vous avez ouvert une session.



[Employees](#) [Priorities](#) [Projects](#) [Statuses](#) [Tasks](#)

Rechercher Tasks

Mot clé	<input type="text"/>
Projet	Rechercher par valeur ▼
<input type="button" value="Rechercher"/>	

Liste de : Tasks

<u>Id</u>	<u>Project</u>	<u>Priority</u>	<u>Status</u>	<u>Name</u>	<u>Assigned To</u>
1	CodeCharge	High	On hold	Great Project needs to be greater	helen
2	CodeCharge	Highest	Closed	Fix ALL bugs	george
3	CodeCharge	High	Closed	Get ready to click	peter
4	My Project	Highest	Open	Finish My Project	ignace
5	Test Project	High	In progress	Test this project.	ken
6	CodeCharge	Highest	Open	Code with one hand.	alexander
7	Test Project	Highest	On hold	Get armed	helen
8	Test Project	Highest	Open	Write more code	ignace
9	Super Project	Highest	In progress	Code, code, code...	george
10	Test Project	Lowest	On hold	Sleep	ken
11	Super Project	Highest	Open	Have fun	alexander

Nouveau 1 of 1

Etape 2

Modifier un Champ Etiquette sur la Page Maintenance de Tâches

Maintenant vous allez procéder à une modification nécessaire sur la page Maintenance de Tâches sur laquelle vous avez dû remarqué que le champ Etiquette *Assigned By* n'affiche pas le nom de l'employé, mais l'ID, comme montré ci-dessous ; c'est parce que la table *tasks* contient seulement l'ID utilisateur, alors que la table *employees* contient les noms des utilisateurs réels.



[Employees](#) [Priorities](#) [Projects](#) [Statutes](#) [Tasks](#)

Ajouter/Editer Tasks

Project	CodeCharge
Priority	Highest
Status	Closed
Type	Question
Name	Fix ALL bugs
Desc	Staying up at night coding? Get CodeCharge, go home, get rest.
Assigned To	george
Finish Date	03.02.2003
Assigned By	3
Start Date	02.02.2003
<input type="button" value="Confirmer"/> <input type="button" value="Supprimer"/>	

Il y a plusieurs méthodes possibles pour traiter de la question. Merci de vous référer aux explications qui suivent :

1 - Créez une Requête qui contient plusieurs tables et qui peut être utilisée en tant que source de données par le formulaire Enregistrement, comme vous l'avez déjà fait avec la grille sur la page Liste de Tâches. Malheureusement, les requêtes qui contiennent plusieurs tables ne peuvent pas être mises à jour de par leur nature, et donc le formulaire Enregistrement entier peut s'arrêter de fonctionner. En d'autres termes, si vous avez indiqué que vous vouliez utiliser une requête contenant une table *tasks* et *employees* dans votre formulaire Enregistrement, alors si vous attribuez une tâche à quelqu'un d'autre, le programme ne pourra pas savoir si vous vouliez mettre à jour la table *tasks* avec le nouveau *employee_id*, ou si vous vouliez mettre à jour la table *employees* et changer le nom de l'employé.

En conséquence, si vous utilisez plusieurs tables comme source de données pour le formulaire Enregistrement, vous devrez également définir les opérations *Custom Insert*, *Custom Update* et *Custom Delete* dans les propriétés du formulaire Enregistrement pour préciser quels champs de la base de données devront être mis à jour avec les valeurs correspondantes saisies sur la page. Cette méthode semble représenter beaucoup d'effort juste pour afficher une valeur supplémentaire sur la page.

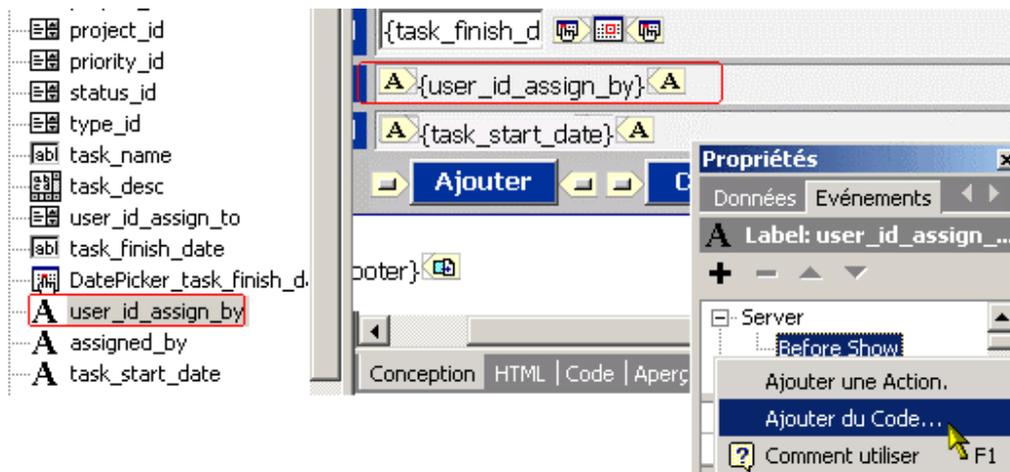
2 - Utilisez une Procédure Événementielle pour insérer du Code Personnalisé (*Custom Code*) où vous pourrez produire la valeur désirée par programmation. Cette méthode est très flexible, dans la mesure où elle vous permet d'étendre le code généré par l'ajout de votre propre code. L'étape suivante décrit en détail cette méthode.

Utiliser l'Événement Before Show pour Changer la Valeur d'une Etiquette

- 1 - En mode **Conception**, sélectionnez l'Etiquette *user_id_assign_by*,
- 2 - Dans la fenêtre **Propriétés**, cliquez sur l'onglet **Données**.
- 3 - Sélectionnez *Text* pour la propriété **Data Type** (Type de Données).
- 4 - Dans la fenêtre **Propriétés**, cliquez sur l'onglet **Événements**.
- 5 - Faites un clic droit sur l'événement **Before Show** et sélectionnez **Ajouter du Code...**



CodeCharge Studio va alors automatiquement passer en mode **Visualisation du Code**.



Une fois en mode **Visualisation du Code**, remplacez le texte suivant :

```
<!-- write your own code here -->
```

par :

```
<CFIF bInEditModeTasks>
  <CF_CCToSQL Value= "#flduser_id_assign_by#" Type= "#ccsInteger#" >
  <CF_CCDLookUp Field="emp_name" Table="employees" Where="emp_id=#CCToSQL#"
Connection="IntranetDB">
<CFSET flduser_id_assign_by=CCDLookUp>
<CFELSE>
  <CF_CCToSQL Value= "#Session.UserID#" Type= "#ccsInteger#" >
  <CF_CCDLookUp Field="emp_name" Table="employees" Where="emp_id=#CCToSQL#"
Connection="IntranetDB">
<CFSET flduser_id_assign_by=CCDLookUp>
</CFIF>
```

Le code du dessus est composé des éléments suivants :

bInEditModeTasks : variable de formulaire, qui indique si l'enregistrement est éditable. En fonction de la valeur de cette propriété, le programme affichera soit le nom de la personne qui a initialement soumis la tâche (mode Edition), soit le nom de la personne qui soumet la tâche (mode Insertion).

user_id_assign_by : le nom de l'Étiquette dans la Grille, et en même temps le nom du champ de la base de données qui a été utilisé pour créer cette étiquette, et qui est maintenant sa source de données.

flduser_id_assign_by : le nom de la variable utilisée par le programme généré pour se référer à l'étiquette et au champ de la base de données à changer. CodeCharge Studio construit des noms de variables par ajout du préfixe "fld" au nom du champ.

CCDLookUp : tag personnalisé de CodeCharge qui supporte la récupération d'une valeur de base de données basée sur un nom de champ, un nom de table, et une condition. Ici, cette fonction récupère le Nom de l'Employé *emp_name* dans la table *employees* en utilisant la condition suivante : la clé *emp_id* est égale à la valeur courante de l'Étiquette.

CCToSQL : tag personnalisé de CodeCharge qui convertit une valeur dans le format supporté par la base de données. Cette fonction requiert un paramètre qui dit si une valeur doit être convertie en un nombre (Entier) ou en du texte. Dans ce cas, cette fonction convertit la valeur courante de l'Étiquette en un nombre pouvant être utilisé avec la fonction *CCDLookUp*. Il est conseillé de toujours utiliser cette fonction avec *CCDLookUp*.



IntranetDB : le nom de la connexion base de données que vous voulez utiliser dans la fonction *CCDLookup*.

Session.UserID : la variable de session qui contient l'ID de l'utilisateur connecté au système.

Le code se lit approximativement comme suit :

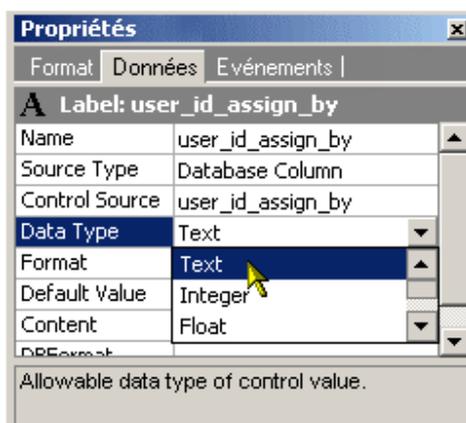
If a record is being edited : Attribue le nom de la personne qui a initialement soumis la question à l'Etiquette *user_id_assign_by*, par la recherche du nom de l'employé dans la table *employees* en utilisant le tag personnalisé *CCDLookup* qui utilise la connexion *IntranetDB* et la valeur de l'Etiquette *user_id_assign_by*.

If a new record is being created : Attribue l'utilisateur connecté à l'Etiquette *user_id_assign_by* en récupérant son nom dans la table *employees* et en utilisant le tag personnalisé *CCDLookup*, qui utilise la connexion *IntranetDB* et le tag personnalisé *CCGetUserID* (qui permet d'obtenir l'ID de l'utilisateur connecté).

Maintenant que vous avez modifié par programmation la valeur de l'Etiquette *user_id_assign_by* pour sortir le Nom de l'Employé à la place de l'ID, vous allez devoir indiquer que ce champ est désormais un champ Texte, et non plus un champ Numérique.

1 - Cliquez sur l'onglet **Données** dans la fenêtre **Propriétés**.

2 - Sélectionnez *Text* en tant que **Data Type** (Type de Données).



Etape 3

Ajouter un Champ Caché "Assigned By" pour la Mise à Jour Automatique de Nouvelles Tâches

Vous avez précédemment utilisé l'événement *Before Show* pour afficher le nom de la personne à qui une tâche est attribuée. Cependant, les champs Etiquette ne peuvent pas être mis à jour de par leur nature, alors bien que le nom de l'employé soit affiché sur la page, il n'est pas écrit dans la base de données. Dans la mesure où nous voulons que la base de données enregistre le nom ou l'ID de la personne qui soumet une tâche, nous allons devoir ajouter la logique de programmation pour effectuer cela.

1 - Ajoutez un champ **Caché** à votre page depuis l'onglet **Formulaires** de la **Boîte à Outils** (*Toolbox*).

Ce type de champ n'est pas visible dans le navigateur, mais il sera utilisé pour stocker une valeur et mettre à jour la base de données.

2 - Configurez le nouveau champ en paramétrant ses propriétés comme indiqué ci-dessous :

Name (*Nom*) : *assigned_by* - le nom du champ Caché qui vient d'être ajouté. Cela peut être n'importe quel nom.

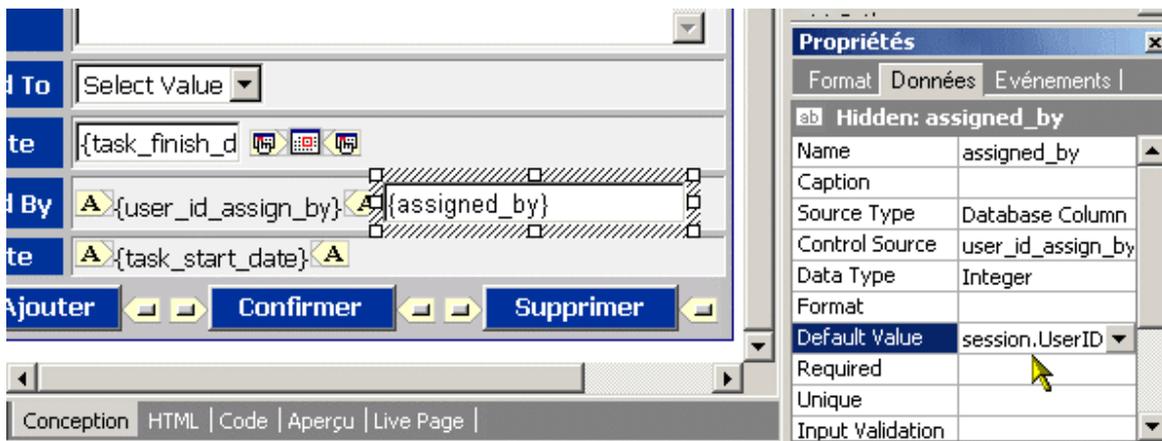
Control Source : *user_id_assign_by* - le champ/colonne de la base de données qui sera utilisé(e) pour récupérer la valeur du champ et qui sera mis(e) à jour avec la nouvelle valeur en cas de changement.

Data Type (*Type de Données*) : *Integer* - le type de la valeur liée au *Control Source*. Nos ID utilisateur/employé sont numériques.

Default (*Par Défaut*) : *Session.UserID* - *Session.UserID* est une variable de session créée par CodeCharge qui conserve l'ID de l'utilisateur connecté au système. Ainsi, vous pouvez simplement spécifier que vous voulez enregistrer l'ID de l'utilisateur



connecté dans le champ `user_id_assign_by` pour chaque nouvelle tâche qui est soumise.



Ajouter un Champ Caché "Date Created" au Formulaire Enregistrement

Maintenant, nous allons ajouter un autre champ Caché à votre page, qui sera utilisé pour soumettre la date et l'heure courantes dans le champ `date_assign` dans la base de données.

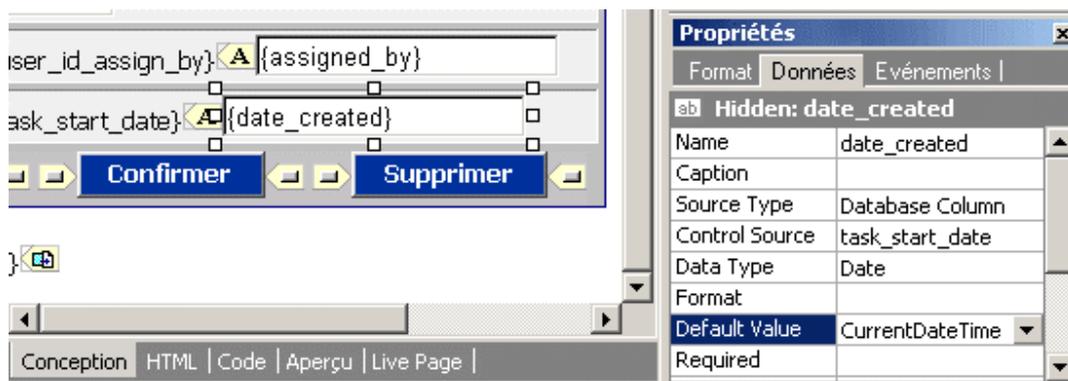
1 - Configurez le nouveau champ comme indiqué ci-dessous :

Name (Nom) : `date_created`

Control Source : `task_start_date`

Data Type (Type de Données) : Date

Default (Par Défaut) : `CurrentDateTime` - La valeur `CurrentDateTime` vous permet d'attribuer automatiquement la date et l'heure courantes aux nouvelles tâches. La propriété Par Défaut n'affecte pas les enregistrements existants, donc la date et l'heure des tâches existantes ne seront pas modifiées lors de mises à jour ultérieures.



1 - Cliquez sur le champ `task_start_date` dans l'Explorateur de Projet.

2 - Dans la fenêtre **Propriétés**, paramétrez **Default Value** (Valeur Par Défaut) à `CurrentDateTime`.

C'est ainsi que le champ Etiquette peut afficher la date bien que le champ Caché ne soit pas visible pour l'utilisateur.

Tester les Champs Etiquette et Caché

1 - Passez en mode **Live Page**.

2 - Sélectionnez ou ajoutez une Tâche, et vous voyez alors votre Etiquette afficher le nom de la personne qui a attribué la tâche.



3 - La version de base de votre Gestionnaire de Tâches est désormais finalisée. Ne pas oublier de sauvegarder !

[Employees](#) [Priorities](#) [Projects](#) [Statuses](#) [Tasks](#)

Ajouter/Editer Tasks

Project	My Project
Priority	Highest
Status	Closed
Type	Task
Name	Finish my project
Desc	Implement "Assigned By" Label to show values from another table
Assigned To	alexander
Finish Date	<input type="text"/>
Assigned By	George Pennington
Start Date	08.04.2003 16:48:07
<input type="button" value="Ajouter"/>	

Etape 4

Programmation du Formulaire Enregistrement

Maintenant que vous avez créé une application simple de Gestion de Tâches, vous souhaitez certainement l'étendre pour qu'elle soit à la fois plus pratique et plus utile... Dans cette section, vous allez avoir un aperçu sur la manière d'implémenter des applications pratiques et sophistiquées par l'ajout de code et d'actions, ceci afin d'offrir des fonctionnalités supplémentaires à votre Gestionnaire de Tâches.

Vous allez apprendre à :

Envoyer des notifications d'email à la personne à qui la tâche est attribuée,

Autoriser seulement la personne à qui la tâche est attribuée à la modifier.

Ajouter du Code dans l'Événement After Insert pour Envoyer des Emails

- 1 - Sélectionnez le formulaire *tasks* dans la page *tasks_maint* en le sélectionnant depuis l'**Explorateur de Projet**, ou en cliquant n'importe où dans la légende du formulaire.
- 2 - Dans la fenêtre **Propriétés**, cliquez sur l'onglet **Événements**.
- 3 - Sélectionnez l'événement **After Insert**.
- 4 - Cliquez sur le bouton **[+]**.
- 5 - Sélectionnez **Ajouter du Code...**



Une fois en mode **Visualisation du Code**, remplacez le commentaire généré :

```
<!-- write your own code here -->
```

par le code du dessous :

```
<CF_CCDLookUp Field="email" Table="employees" Where="emp_id=#flduser_id_assign_to#"
Connection="IntranetDB">
<CFSET mai1_To=CCDLookUp>
    <CF_CCDLookUp Field="email" Table="employees" Where="emp_id=#Session.UserID#"
Connection="IntranetDB">
<CFSET mai1_From=CCDLookUp>
<CF_CCDLookUp Field="max(task_id)" Table="tasks" Where="user_id_assign_by=#Session.UserID#"
Connection="IntranetDB">
<CFMAIL TO="#mai1_To#" FROM="#mai1_From#" SUBJECT="New task for you" TYPE="HTML" >
    The following task was submitted:<br><br>
    Task ID:#CCDLookUp#
    <br><br>#fldtask_desc#
</CFMAIL>
```

Ce qui suit est une explication du code du dessus :

```
<CF_CCDLookUp Field="email" Table="employees" Where="emp_id=#flduser_id_assign_to#"
Connection="IntranetDB">
<CFSET mai1_To= CCDLookUp>
```

Paramètre l'adresse email *To* (Destinataire) à l'adresse email de la personne à qui est attribuée la tâche. Le tag personnalisé *CCDLookUp* est utilisé ici pour récupérer l'adresse email appropriée.

```
<CF_CCDLookUp Field="email" Table="employees" Where="emp_id=#Session.UserID#"
Connection="IntranetDB">
<CFSET mai1_From= CCDLookUp>
```



Paramètre l'adresse email *From* (Expéditeur) à la valeur du champ *email* dans la table *employees* où *emp_id* correspond à l'utilisateur connecté. La fonction *CCDLookup* est utilisée pour récupérer une valeur de base de données, alors que *Session.UserID* retrouve l'ID de la connexion utilisateur en cours.

```
<CF_CCDLookup Field="max(task_id)" Table="tasks" Where="user_id_assign_by=#Session.UserID#"
Connection="IntranetDB">
```

Récupère l'ID de tâche courante. Le dernier ID de tâche inséré peut être obtenu en utilisant différentes méthodes avec différentes bases. Malheureusement, Microsoft Access ne supporte pas la récupération du dernier enregistrement inséré, alors vous devrez utiliser la fonction *CCDLookup* pour récupérer le plus grand ID de tâche soumis par l'utilisateur connecté (en supposant que les ID de tâche soient créés par incrémentation).

```
<CFMAIL TO="#mai1_To#" FROM="#mai1_From#" SUBJECT="New task for you" TYPE="HTML" >
  The following task was submitted:<br><br>
  Task ID:#CCDLookup#
  <br><br>#fldtask_desc#
</CFMAIL>
```

Le tag *CFMail* envoie une adresse email en utilisant les variables paramétrées dans le tag. Le corps du tag comprend le corps de l'adresse email à envoyer.

Utiliser l'Événement After Update pour Envoyer des Emails

Vous avez précédemment ajouté le code nécessaire qui envoie une notification d'email à la personne à qui la tâche est attribuée, une fois l'enregistrement d'une nouvelle tâche effectuée dans le système. Maintenant, il nous faut implémenter une fonctionnalité similaire dans l'événement *After Update* pour informer la personne à qui la tâche est attribuée, lorsqu'une tâche existante est mise à jour et réattribuée à quelqu'un d'autre.

- 1 - Cliquez sur le formulaire *tasks* dans l'**Explorateur de Projet**.
- 2 - Dans la fenêtre **Propriétés**, sélectionnez l'onglet **Événements**.
- 3 - Ajoutez le **Custom Code** (Code Personnalisé) dans l'événement **After Update** :

```
<CFIF #flduser_id_assign_to# NEQ #Session.UserID#>
  <CF_CCDLookup Field="email" Table="employees" Where="emp_id=#flduser_id_assign_to#"
Connection="IntranetDB">
  <CFSET mai1_To=CCDLookup>
  <CF_CCDLookup Field="email" Table="employees" Where="emp_id=#Session.UserID#"
Connection="IntranetDB">
  <CFSET mai1_From= CCDLookup>
  <CF_CCGetParam strName="task_id" def="">
  <CFMAIL TO="#mai1_To#" FROM="#mai1_From#" SUBJECT="New task for you" TYPE="HTML">
    The following task was assigned to you:<br><br>
    Task ID:#CCGetParam#
    <br><br>#fldtask_desc#
  </CFMAIL>
</CFIF>
```



Les principales différences entre le code ci-dessus et celui utilisé dans l'événement *After Insert* sont les suivantes :

- 1 - Une condition *if* (si) a été ajoutée pour envoyer un email seulement si un utilisateur attribue une tâche à une autre personne.
- 2 - *task_id* est récupéré depuis l'URL utilisant le tag personnalisé *CCGetParam*. Nous pouvons utiliser cette méthode parce que des tâches peuvent être mises à jour seulement si l'utilisateur est arrivé à la page en cours par une URL qui contient un ID de tâche à mettre à jour. Une telle URL ressemble à : http://localhost/TaskManager/tasks_maint.cfm?task_id=9

Test de l'Envoi d'Email

Avant de tester l'envoi d'email :

1 - Vous devez ajouter de nouveaux utilisateurs à votre base de données avec des adresses email correctes, ou modifier les utilisateurs existants en changeant leur adresse email :

A - Vous pouvez le faire en ouvrant la base de données *Intranet.mdb* qui se trouve dans votre répertoire *Projet*.

B - Sinon, vous pouvez utiliser le Gestionnaire de Tâches. Allez à la page *Employés* pour y visualiser les emails utilisateur et les modifier.

2 - Une fois vos utilisateurs configurés avec le test d'emails, sauvegardez votre projet et passez en mode **Live Page** pour tester votre système.

Remarque : Vous aurez besoin de Microsoft Access 2000 ou une version supérieure pour éditer manuellement le fichier base de données. Si votre code email fonctionne correctement, vous devez alors vous retrouver sur la page *Liste de Tâches* après l'ajout ou la modification d'une tâche, et la personne à qui la tâche a été attribuée doit recevoir un email.

Etape 5

Implémenter la Sécurité d'Enregistrement dans l'Événement After Initialize

Votre Gestionnaire de Tâches est désormais pratiquement complet, seule une fonctionnalité pouvant s'avérer importante manque, à savoir la Sécurité. Dans l'état actuel n'importe qui peut modifier et supprimer n'importe quelle tâche. Vous voudrez certainement limiter l'accès pour que seuls les employés autorisés puissent mettre à jour leurs tâches. Il y a plusieurs façons de procéder, et nous allons examiner plusieurs d'entre elles.

- 1 - Cliquez sur la page *tasks_maint* dans l'**Explorateur de Projet**.
- 2 - Dans la fenêtre **Propriétés**, sélectionnez l'onglet **Événements**.
- 3 - Ajoutez **Custom Code** (Code Personnalisé) à l'événement **After Initialize** de la page. Procédez comme suit :

Une fois en mode **Visualisation de Code**, remplacez le commentaire généré :

```
<!------->
<!-- Write your own code here --->
<!------->
```



par le code du dessous :

```
<CF_CCGetParam strName="task_id" outputvar="current_task">
<CFPARAM Name="Session.UserID" Default="">
<CF_CCToSQL Value="#current_task#" Type="#ccsInteger#">
<CF_CCDLookUp Field="user_id_assign_to" Table="tasks" Where="task_id=#CCToSQL#">
<CFIF current_task NEQ 0 AND Session.UserID NEQ CCDLookUp>
  <CFSET hideTasks = True>
  <!--<CFSET strRedirect = "tasks_list.cfm"> -->
  <!--<CFSET binUpdateAllowedTasks = False> -->
  <!--<CFSET binDeleteAllowedTasks = False> -->
  <!--<CFSET hideUpdate = True>-->
  <!--<CFSET hideDelete= True>-->
</CFIF>
```

Le code du dessus vous permet de tester les méthodes pour implémenter la sécurité d'enregistrement :

Ne pas montrer la Tâche (formulaire Enregistrement) sur la page si la tâche sélectionnée n'appartient pas à l'utilisateur connecté. Un utilisateur non autorisé verra une page blanche.

Vous pouvez cacher n'importe quel formulaire sur une page en attribuant une valeur *True* (vraie) à la variable *hideTasks* (cacher Tâches). Le code "*current_task NEQ 0*" dans la condition *CFIF* indique que le code sera seulement exécuté si un utilisateur essaie de modifier une tâche existante et qui ne lui est pas attribuée. Vous pouvez tester cette fonctionnalité en insérant le code du dessus dans l'événement, passer en mode **Live Page** et en essayant de modifier une tâche qui ne vous est pas attribuée... et dans ce cas, vous verrez une page vide. Bien que cette fonctionnalité ne soit pas très utile, elle illustre comment cacher des formulaires sur une page. Maintenant ajoutez un autre formulaire Enregistrement à votre page qui ne puisse pas être mis à jour et qui ait seulement des champs Etiquette qui montrent les informations. Une fois que vous avez deux formulaires sur la page, vous pouvez cacher chaque formulaire par programmation en utilisant des critères d'exclusion.

Rediriger des utilisateurs non autorisés sur une autre page. Seuls les utilisateurs à qui les tâches sont attribuées peuvent visualiser la page.

Vous pouvez mettre en application cette fonctionnalité en modifiant légèrement le code du dessus comme montré ci-dessous :

```
<CF_CCGetParam strName="task_id" outputvar="current_task">
<CFPARAM Name="Session.UserID" Default="">
<CF_CCToSQL Value="#current_task#" Type="#ccsInteger#">
<CF_CCDLookUp Field="user_id_assign_to" Table="tasks" Where="task_id=#CCToSQL#">
<CFIF current_task NEQ 0 AND Session.UserID NEQ CCDLookUp>
  <!--<CFSET hideTasks = True> -->
  <CFSET strRedirect = "tasks_list.cfm">
  <!--<CFSET binUpdateAllowedTasks = False> -->
  <!--<CFSET binDeleteAllowedTasks = False> -->
  <!--<CFSET hideUpdate = True>-->
  <!--<CFSET hideDelete= True>-->
</CFIF>
```



Le code du dessus montre que vous devez commenter la ligne précédemment active, et activer la ligne qui commence avec `<CFET stRedirect`. `StrRedirect` est une variable utilisée par CodeCharge Studio pour déterminer si la page en cours doit être redirigée sur une autre page, par exemple si un utilisateur n'a pas de session ouverte. Cette variable peut être utilisée seulement sur les pages ayant un accès restreint et nécessitant une authentification. Vous pouvez simplement attribuer la page de destination à la variable `strRedirect` et la page sera alors automatiquement redirigée. Testez cette fonctionnalité en modifiant le code comme montré, ensuite passez en mode **Live Page**, et essayez alors de modifier une tâche qui ne vous est pas attribuée.

Perl

Extension des Fonctionnalités de l'Application avec la Programmation d'Événements (Perl)

Vous avez certainement remarqué que jusqu'à présent vous avez construit votre Gestionnaire de Tâches sans avoir à vous préoccuper du code. En effet, CodeCharge Studio peut vous aider à construire des systèmes fonctionnels sans programmation ; toutefois la création de systèmes plus sophistiqués requiert un minimum de programmation. Heureusement, CodeCharge Studio facilite la programmation en proposant un éditeur de code de haut niveau, en plus des Événements et des Actions qui vous aident à insérer des morceaux de code pré-programmés dans le programme.

Voici les définitions d'une Action et d'un Événement :

Action

Composant de génération de code définissable par l'utilisateur, qui insère un bloc de code dans une procédure événementielle. CodeCharge Studio est fourni avec de nombreuses Actions pré-définies, qui se trouvent dans le dossier suivant : (*CCS folder*)\Components\Actions. Intérieurement, les actions consistent en du code XML et XSL pouvant être facilement personnalisé. Par exemple, une action peut être paramétrée sur un Textbox pour valider une adresse e-mail.

Procédure Événementielle

Une procédure automatiquement exécutée en réponse à un événement lancé par un programme à son exécution. Les événements sont les meilleurs emplacements pour mettre du Code Personnalisé (*Custom Code*).

Connexion à la Base de Données côté Serveur en Perl

Bien que la connexion en mode conception (via l'utilisation de l'assistant), quand vous utilisez Perl, puisse être faite à une base de données Microsoft Access, la connexion côté Serveur doit être configurée différemment. En particulier, l'interface DBI doit être utilisée pour que la connexion utilise un driver DBD approprié, disponible sur le serveur sur lequel les pages Web seront publiées. Pour plus d'informations sur la configuration des connexions côté Serveur, merci de vous référer à la section sur les connexions côté Serveur en Perl.

Etape 1

Utiliser l'Événement Before Show Row pour Changer la Couleur du Texte

Démarrons la programmation basique avec une tâche simple de changement de la couleur d'un champ grille sur notre page Liste de Tâches. Concrètement, nous allons mettre en évidence les tâches qui vous sont attribuées en affichant votre nom en bleu dans la grille.

- 1 - Ouvrez la page `tasks_list` dans l'**Explorateur de Projet**.
- 2 - Ouvrez l'arborescence de la grille `tasks`.
- 3 - Faites un clic droit sur le champ `emp_login1` et sélectionnez **Propriétés**.
- 4 - Dans l'onglet **Données**, paramétrez la valeur de la propriété **Name** (Nom) à `user_id_assign_to`.
- 5 - Dans l'**Explorateur Projet**, faites un clic droit sur le champ `task_name` et sélectionnez **Propriétés**.
- 6 - Dans l'onglet **Données**, paramétrez la valeur de la propriété **Content** (Contenu) à `HTML`.

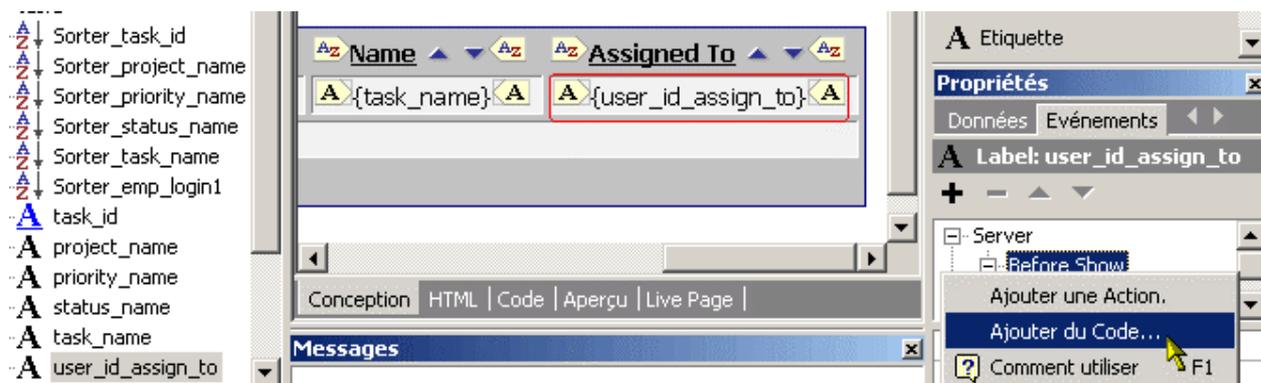


7 - Sélectionnez la grille *tasks* dans l'**Explorateur de Projet**, ou cliquez n'importe où dans la légende du formulaire.

8 - Sélectionnez l'onglet **Evénements** dans la fenêtre **Propriétés**.

9 - Faites un clic droit sur l'événement **Before Show Row** et sélectionnez **Ajouter du Code...** .

L'événement *Before Show Row* se déclenche dans le programme après que les valeurs champs soient attribuées, mais avant d'être générées en HTML. En ajoutant du code dans cet événement, vous pouvez modifier la valeur champ avant qu'elle ne soit montrée.



Contrôler par Programmation la Valeur Champ

Une fois votre Code Personnalisé (*Custom Code*) ajouté, vous verrez la fenêtre d'édition du code avec l'emplacement approprié pour entrer le nouveau code.

Remplacez cette ligne de code :

```
# Write your own code here.
```

Par les lignes suivantes (Perl) :

```
if ($tasks->{user_id_assign_to}->GetValue() eq CCGetSession("UserLogin"))
{
    $tasks->{task_name}->SetValue("<b><font color='blue'>" .
        $tasks->{task_name}->GetValue().
        "</font></b>");
}
```

Ce qui suit est une explication sur le fonctionnement du code Perl du dessus :

```
if ($tasks->{user_id_assign_to}->GetValue() eq CCGetSession("UserLogin"))
```

C'est une condition *if* (si) qui est vraie seulement si la valeur du champ *task_name* est égale au nom de Login de l'employé ayant une session ouverte. Le champ base de données fournit la valeur pour le champ *task_name* dans le formulaire Grille. Une fois connecté au système, le programme reconnaîtra vos tâches en comparant votre nom de Login à la valeur *emp_login* de la personne à qui cette tâche est attribuée. *UserLogin* est une des variables de session utilisées par les programmes générés par CodeCharge, et elle conserve le nom de Login de la session utilisateur jusqu'à son expiration.

Note :

Il s'agit des variables de session créées par défaut par CodeCharge Studio:

UserID : la valeur du champ clé primaire de la session utilisateur



UserLogin : le nom de Login de l'utilisateur connecté au système

GroupID : le niveau/groupe de sécurité de l'utilisateur connecté au système

```
$tasks->{task_name}->SetValue("<b><font color='blue'>" . $tasks->{task_name}->GetValue() .  
"</font></b>");
```

Ce code est exécuté si la condition *if* est rencontrée. Elle modifie la valeur du champ *task_name*. La valeur du champ est remplacée par sa valeur de base de données encapsulée dans du code HTML, qui spécifie la couleur de police, telle que le bleu, et ajoute un tag HTML ** pour afficher la police en gras. De plus, notez que le code est orienté objet et que vous indiquez que vous voulez attribuer une valeur au champ *task_name* dans la grille *tasks*. *SetValue* est une méthode d'un objet, pouvant être utilisée pour modifier la valeur de l'objet.

Prévisualiser la Page Liste de Tâches

1 - Sauvegardez votre projet.

2 - Passez en mode **Live Page** pour visualiser votre page fonctionnelle. Si aucun nom de tâche n'est mis en évidence dans la colonne "*Name*", c'est que vous n'êtes probablement pas connecté(e) au système.

3 - Le menu ne contenant pas encore de lien à la page Login, vous pouvez y accéder en essayant d'ouvrir une des pages restreintes, par exemple la page Maintenance de Tâches.

4 - Cliquez sur n'importe lequel des ID projet, et vous devriez voir la page Login.

5 - Connectez-vous en tant que **george/george**, puis cliquez sur le lien **Tasks** sur le menu pour retourner à la page Liste de Tâches.

Maintenant, vous devriez voir que certaines tâches sont mises en évidence ; ces tâches correspondent aux tâches de l'utilisateur pour lequel vous avez ouvert une session.



[Employees](#) [Priorities](#) [Projects](#) [Statuses](#) [Tasks](#)

Rechercher Tasks

Mot clé	<input type="text"/>
Projet	Rechercher par valeur ▼
Rechercher	

Liste de : Tasks

<u>Id</u>	<u>Project</u>	<u>Priority</u>	<u>Status</u>	<u>Name</u>	<u>Assigned To</u>
<u>1</u>	CodeCharge	High	On hold	Great Project needs to be greater	helen
<u>2</u>	CodeCharge	Highest	Closed	Fix ALL bugs	george
<u>3</u>	CodeCharge	High	Closed	Get ready to click	peter
<u>4</u>	My Project	Highest	Open	Finish My Project	ignace
<u>5</u>	Test Project	High	In progress	Test this project.	ken
<u>6</u>	CodeCharge	Highest	Open	Code with one hand.	alexander
<u>7</u>	Test Project	Highest	On hold	Get armed	helen
<u>8</u>	Test Project	Highest	Open	Write more code	ignace
<u>9</u>	Super Project	Highest	In progress	Code, code, code...	george
<u>10</u>	Test Project	Lowest	On hold	Sleep	ken
<u>11</u>	Super Project	Highest	Open	Have fun	alexander

Nouveau 1 of 1



Etape 2

Modifier un Champ Etiquette sur la Page Maintenance de Tâches

Maintenant procédons à une modification nécessaire sur la page Maintenance de Tâches sur laquelle vous avez dû remarqué que le champ Etiquette *Assigned By* n'affiche pas le nom de l'employé, mais l'ID, comme montré ci-dessous ; c'est parce que la table *tasks* contient seulement l'ID utilisateur, alors que la table *employees* contient les noms des utilisateurs réels.

[Employees](#) [Priorities](#) [Projects](#) [Statutes](#) [Tasks](#)

Ajouter/Editer Tasks

Project	CodeCharge
Priority	Highest
Status	Closed
Type	Question
Name	Fix ALL bugs
Desc	Staying up at night coding? Get CodeCharge, go home, get rest.
Assigned To	george
Finish Date	03.02.2003
Assigned By	3
Start Date	02.02.2003
<input type="button" value="Confirmer"/> <input type="button" value="Supprimer"/>	

Il y a plusieurs méthodes possibles pour traiter de la question. Merci de vous référer aux explications qui suivent :

1 - Créez une Requête qui contient plusieurs tables pouvant être utilisée en tant que source de données par le formulaire Enregistrement, comme vous l'avez déjà fait avec la grille sur la page Liste de Tâches. Malheureusement, les requêtes qui contiennent plusieurs tables ne peuvent pas être mises à jour de par leur nature, et donc le formulaire Enregistrement entier peut s'arrêter de fonctionner. En d'autres termes, si vous avez indiqué que vous vouliez utiliser une requête contenant une table *tasks* et *employees* dans votre formulaire Enregistrement, alors si vous attribuez une tâche à quelqu'un d'autre, le programme ne pourra pas savoir si vous vouliez mettre à jour la table *tasks* avec le nouveau *employee_id*, ou si vous vouliez mettre à jour la table *employees* et changer le nom de l'employé.

Ainsi, si vous utilisez plusieurs tables comme source de données pour le formulaire Enregistrement, vous devrez également définir les opérations *Custom Insert*, *Custom Update* et *Custom Delete* dans les propriétés du formulaire Enregistrement pour indiquer quels champs de la base de données devront être mis à jour avec les valeurs correspondantes saisies sur la page. Cette méthode semble représenter beaucoup d'effort juste pour afficher une valeur supplémentaire sur la page.

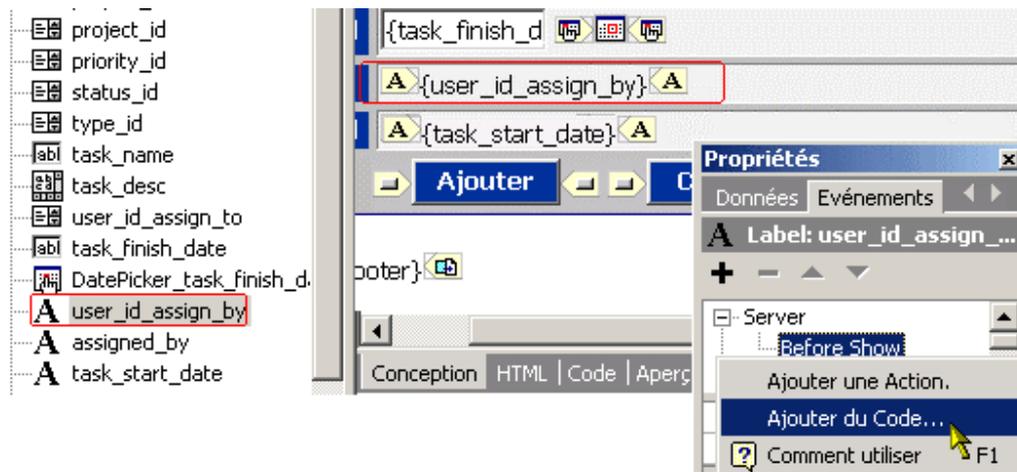
2 - Utilisez une Procédure Événementielle pour insérer du Code Personnalisé (*Custom Code*) où vous pourrez produire la valeur désirée par programmation. Cette méthode est très flexible, dans la mesure où elle vous permet d'étendre le code généré par l'ajout de votre propre code. L'étape suivante décrit en détail cette méthode.

Utiliser l'Événement Before Show pour Changer la Valeur d'une



Etiquette

- 1 - En mode **Conception**, sélectionnez l'Etiquette *user_id_assign_by*,
 - 2 - Dans la fenêtre **Propriétés**, cliquez sur l'onglet **Evénements**.
 - 3 - Faites un clic droit sur l'événement **Before Show** et sélectionnez **Ajouter du Code...** .
- CodeCharge Studio va alors automatiquement passer en mode **Visualisation du Code**.



Une fois en mode **Visualisation du Code**, si vous générez du Perl, vous devriez voir le fichier *tasks_maint_events.pl*, avec les lignes de code suivantes :

```
# -----  
# Write your own code here.  
# -----
```

Remplacez le texte :

```
# Write your own code here.
```

Par :

```
$DBIntranetDB = clsDBIntranetDB->new();  
if ( $tasks->{EditMode} )  
{  
    $tasks->{user_id_assign_by}->SetValue(CCDlookUp("emp_name", "employees", " emp_id=" . $DBIntranetDB->ToSQL($tasks->{user_id_assign_by}->GetValue(), $ccsInteger,$DBIntranetDB));  
}  
else  
{  
    $tasks->{user_id_assign_by}->SetValue(CCDlookUp("emp_name", "employees", " emp_id=" .  
    $DBIntranetDB->ToSQL(CCGetSession("UserID"), $ccsInteger, $DBIntranetDB));  
}
```

Le code du dessus est composé des éléments suivants :



tasks : le nom du formulaire enregistrement sur la page.

EditMode : propriété du formulaire, qui indique si l'enregistrement est éditable. En fonction de la valeur de cette propriété, nous affichons soit le nom de la personne qui a initialement soumis la tâche (mode Edition), soit le nom de la personne qui soumet la tâche (mode Insertion).

user_id_assign_by : le nom de l'Etiquette dans la Grille, et en même temps le nom du champ base de données qui a été utilisé pour créer cette Etiquette, et qui est désormais sa source de données.

SetValue : une méthode d'un objet (dans ce cas l'Etiquette), qui a été utilisée pour modifier la valeur de l'objet.

\$tasks->{user_id_assign_by}->SetValue : méthode *SetValue* entièrement qualifiée, qui dit au programme à quel objet il se réfère. En d'autres termes, c'est la méthode *SetValue* qui affecte le champ *user_id_assign_by*, qui en retour dépend de la grille *tasks*.

CCDlookup : fonction de CodeCharge qui supporte la récupération d'une valeur de base de données basée sur un nom de champ, un nom de table, et une condition. Ici, cette fonction récupère le Nom de l'Employé (*emp_name*) dans la table *employees* en utilisant la condition suivante : la clé (*emp_id*) est égale à la valeur courante de l'Etiquette.

ToSQL : Fonction de CodeCharge qui convertit une valeur dans un format supporté par la base de données. Cette fonction requiert un paramètre qui dit si une valeur doit être convertie en un nombre (entier) ou en du texte. Dans ce cas, cette fonction convertit la valeur courante de l'Etiquette en un nombre pouvant être utilisé avec la fonction *CCDlookup*. Il est conseillé de toujours utiliser cette fonction avec *CCDlookup*.

\$DBIntranetDB : le nom de l'objet qui définit la connexion base de données que vous voulez utiliser dans la fonction *CCDLookup*.

CCGetSession("UserID") : fonction de CodeCharge qui retourne l'ID de l'utilisateur connecté au système.

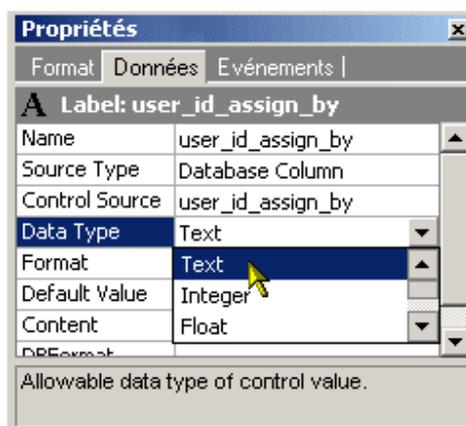
Le code se lit approximativement comme suit :

If a record is being edited : Attribue le nom de la personne qui a initialement soumis la question à l'Etiquette *user_id_assign_by*, par la recherche du nom de l'employé dans la table *employees* en utilisant la fonction *CCDLookup*, qui utilise la connexion *IntranetDB* et la valeur de l'Etiquette *user_id_assign_by*.

If a new record is being created : Attribue l'utilisateur connecté à l'Etiquette *user_id_assign_by* en récupérant son nom dans la table *employees* en utilisant la fonction *CCDLookup* qui utilise la connexion *IntranetDB* et la fonction *CCGetSession("UserID")*, qui permet d'obtenir l'ID de l'utilisateur connecté.

Maintenant que vous avez modifié par programmation la valeur de l'Etiquette *user_id_assign_by* pour produire le Nom de l'Employé à la place de son ID, vous allez devoir indiquer que ce champ est désormais un champ Texte, et non plus un champ Numérique.

- 1 - Cliquez sur l'onglet **Données** dans la fenêtre **Propriétés**.
- 2 - Sélectionnez *Text* en tant que **Data Type** (Type de Données).





Etape 3

Ajouter un Champ Caché "Assigned By" pour la Mise à Jour Automatique de Nouvelles Tâches

Vous avez précédemment utilisé l'événement *Before Show* pour afficher le nom de la personne qui attribue une tâche. Cependant, les champs Etiquette ne peuvent pas être mis à jour de par leur nature, alors bien que le nom de l'employé soit affiché sur la page, il n'est pas écrit dans la base de données. Dans la mesure où nous voulons que la base de données enregistre le nom ou l'ID de la personne qui soumet une tâche, nous allons devoir ajouter la logique de programmation pour effectuer cela.

1 - Ajoutez un champ **Caché** à votre page depuis l'onglet **Formulaires** de la **Boîte à Outils** (*Toolbox*).

Ce type de champ n'est pas visible dans le navigateur, mais il sera utilisé pour stocker une valeur et mettre à jour la base de données.

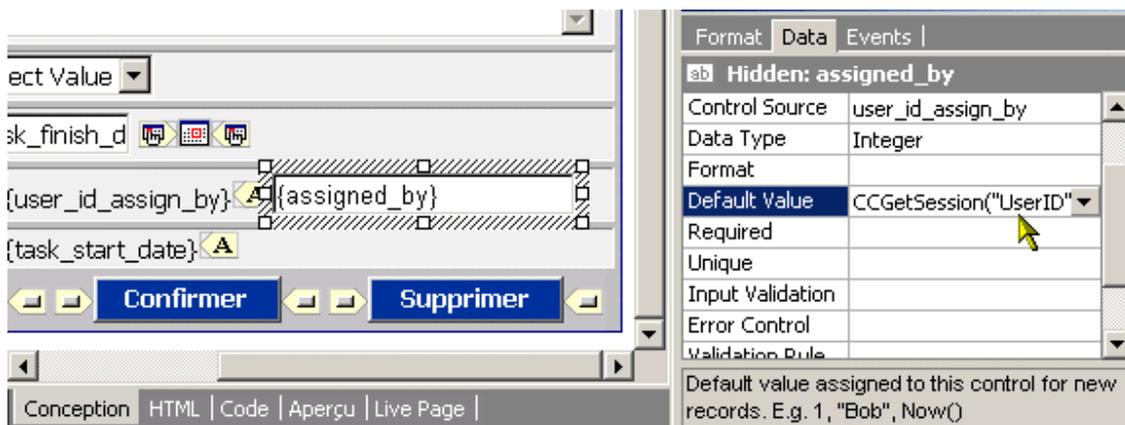
2 - Configurez le nouveau champ en paramétrant ses propriétés comme indiqué ci-dessous :

Name (Nom) : *assigned_by* - le nom du champ Caché qui vient d'être ajouté. Cela peut être n'importe quel nom.

Control Source : *user_id_assign_by* - le champ/colonne de la base de données qui sera utilisé(e) pour retrouver la valeur du champ et qui sera mis(e) à jour avec la nouvelle valeur en cas de changement.

Data Type (Type de Données) : *Integer* - le type de la valeur liée au *Control Source*. Nos ID utilisateur/employé sont numériques.

Default (par défaut) : *CCGetSession("UserID")* - *CCGetSession("UserID")* est une fonction de CodeCharge qui récupère l'ID de l'utilisateur connecté au système. Ainsi, vous pouvez simplement spécifier que vous voulez enregistrer l'ID de l'utilisateur connecté dans le champ *user_id_assign_by* pour chaque nouvelle tâche qui est soumise.



Ajouter un Champ Caché "Date Created" au Formulaire Enregistrement

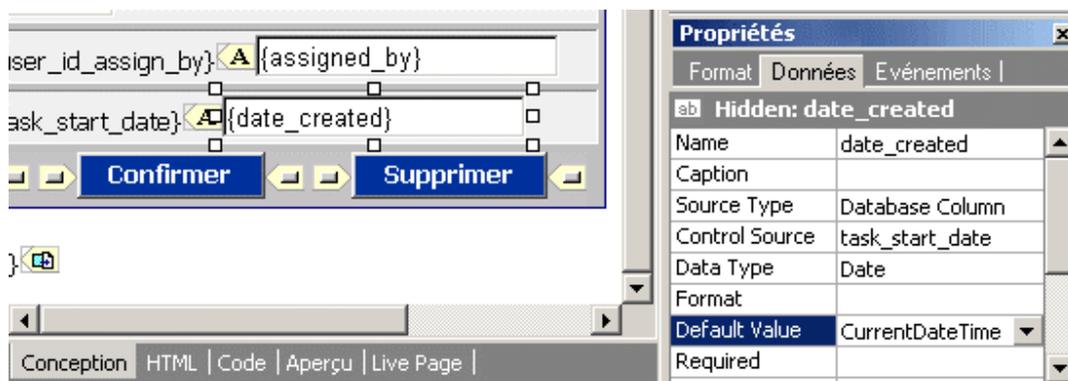
1 - Configurez le nouveau champ comme indiqué ci-dessous :

Name (Nom) : *date_created*

Control Source : *task_start_date*

Data Type (Type de Données) : *Date*

Default (Par Défaut) : *CurrentDateTime* - La valeur *CurrentDateTime* vous permet d'attribuer automatiquement la date et l'heure courantes aux nouvelles tâches. La propriété Par Défaut n'affecte pas les enregistrements existants, donc la date et l'heure des tâches existantes ne seront pas modifiées lors de mises à jour ultérieures.



1 - Cliquez sur le champ `task_start_date` dans l'**Explorateur de Projet**.

2 - Dans la fenêtre **Propriétés**, paramétrez **Default Value** (Valeur Par Défaut) à `CurrentDateTime`.

C'est ainsi que le champ Etiquette peut afficher la date, bien que le champ Caché ne soit pas visible pour l'utilisateur.

Tester les Champs Etiquette et Caché

Pour terminer :

1 - Passez en mode **Live Page**.

2 - Sélectionnez ou ajoutez une Tâche, et vous voyez alors votre Etiquette afficher le nom de la personne qui a attribué la tâche.

3 - La version de base de votre Gestionnaire de Tâches est désormais finalisée. Ne pas oublier de sauvegarder !

[Employees](#) [Priorities](#) [Projects](#) [Statuses](#) [Tasks](#)

Ajouter/Editer Tasks

Project	My Project
Priority	Highest
Status	Closed
Type	Task
Name	Finish my project
Desc	Implement "Assigned By" Label to show values from another table
Assigned To	alexander
Finish Date	
Assigned By	George Pennington
Start Date	08.04.2003 16:48:07
Ajouter	



Etape 4

Programmation du Formulaire Enregistrement

Maintenant que vous avez créé une application simple de Gestion de Tâches, vous souhaitez certainement l'étendre pour qu'elle soit à la fois plus pratique et plus utile... Dans cette section, vous allez avoir un aperçu sur la manière d'implémenter des applications pratiques et sophistiquées par l'ajout de code et d'actions, ceci afin d'offrir des fonctionnalités supplémentaires à votre Gestionnaire de Tâches.

Vous allez apprendre à :

- Envoyer des notifications d'email à la personne à qui la tâche est attribuée,
- Autoriser seulement la personne à qui la tâche est attribuée à la modifier.

Ajouter du Code dans l'Événement After Insert pour Envoyer des Emails

- 1 - Sélectionnez le formulaire *tasks* dans la page *tasks_maint* en le sélectionnant depuis l'**Explorateur de Projet**, ou en cliquant n'importe où dans la légende du formulaire.
- 2 - Dans la fenêtre **Propriétés**, cliquez sur l'onglet **Événements**.
- 3 - Sélectionnez l'événement **After Insert**.
- 4 - Cliquez sur le bouton **[+]**.
- 5 - Sélectionnez **Ajouter du Code...**



Une fois en mode **Visualisation du Code**, remplacez le commentaire généré :

```
// Write your own code here.
```



Par le code suivant :

```
$DBIntranetDB = clsDBIntranetDB->new();

$from_name = CCDlookUp("emp_name", "employees", "emp_id=" . CCGetSession("UserID") , $DBIntranetDB);
$from_email = CCDlookUp("email", "employees", "emp_id=" . CCGetSession("UserID"), $DBIntranetDB);
$to_email= CCDlookUp("email", "employees", "emp_id=" .
$tasks->{user_id_assign_to}->GetValue(), $DBIntranetDB);

$subject = "New task for you";
$message = "The following task was submitted:<br><br>" .
        "Task ID: " . CCDlookUp("max(task_id)", "tasks", "user_id_assign_by=" .
        CCGetSession("UserID"), $DBIntranetDB) . "<br><br>" .

$tasks->{task_desc}->GetValue();

if(open(SENDMAIL, "| sendmail $from_name"))
{
    print SENDMAIL "From: $from_email\nTo: $to_email\nSubject:
    $subject\n\n$message\n.\n";
    close(SENDMAIL);
}
```

Comme vous avez pu le constater, le code du dessus envoie des emails aux utilisateurs à qui de nouvelles tâches sont attribuées. Ce qui suit est une explication du code :

```
$from_name = CCDlookUp("emp_name", "employees", "emp_id=" . CCGetSession("UserID") , $DBIntranetDB);
```

Paramètre *from_name* à la valeur du champ *emp_name* pour l'utilisateur connecté.

```
$from_email = CCDlookUp("email", "employees", "emp_id=" . CCGetSession("UserID"), $DBIntranetDB);
```

Paramètre *from_email* à la valeur du champ *email* dans la table *employees* où *emp_id* correspond à l'utilisateur connecté. La fonction *CCDlookUp* est utilisée pour récupérer une valeur de base de données, alors que *CCGetSession("UserID")* récupère l'ID de l'utilisateur connecté.

```
$to_email= CCDlookUp("email", "employees", "emp_id=" . $tasks->{user_id_assign_to}->GetValue(),
$DBIntranetDB);
```

Paramètre *to_email* à l'email de la personne à qui est attribuée la tâche. La fonction *CCDlookUp* est utilisée ici pour récupérer l'adresse email appropriée.

```
$subject = "New task for you";
```



L'objet de l'email à envoyer.

```
$message = "The following task was submitted:<br><br>" . "Task ID: " . CCDlookUp("max(task_id)", "tasks",  
"user_id_assign_by=" . CCGetSession("UserID"), $DBIntranetDB) . "<br><br>" . $tasks->{task_desc}-  
>GetValue();
```

La variable *message* contient le corps de l'email qui sera envoyé. La fonction *CCDlookUp* est utilisée pour récupérer le plus grand ID de tâche soumis par l'utilisateur connecté (en supposant que les ID de tâche sont créés par incrémentation).

```
if(open(SENDMAIL, "| sendmail $from_name"))  
{  
    print SENDMAIL "From: $from_email\nTo: $to_email\nSubject: $subject\n\n$message\n.\n";  
    close(SENDMAIL);  
}
```

Envoie l'email en utilisant les variables créées dans le code du dessus.

Utiliser l'Événement After Update pour Envoyer des Emails

Vous avez précédemment ajouté le code nécessaire qui envoie une notification d'email à la personne à qui la tâche est attribuée, une fois l'enregistrement d'une nouvelle tâche effectuée dans le système. Maintenant, il nous faut implémenter une fonctionnalité similaire dans l'événement *After Update* pour informer la personne, à qui la tâche est attribuée, lorsqu'une tâche existante est mise à jour et réattribuée à quelqu'un d'autre.

- 1 - Cliquez sur le formulaire *tasks* dans l'**Explorateur de Projet**.
- 2 - Dans la fenêtre **Propriétés**, sélectionnez l'onglet **Événements**.
- 3 - Ajoutez le **Custom Code** (Code Personnalisé) dans l'événement **After Update** :

```
$DBIntranetDB = clsDBIntranetDB->new();  
  
if (CCGetSession("UserID") != $tasks->{user_id_assign_to}->GetValue())  
{  
    $from_name = CCDlookUp("emp_name", "employees", "emp_id=" . CCGetSession("UserID"), $DBIntranetDB);  
    $from_email = CCDlookUp("email", "employees", "emp_id=" . CCGetSession("UserID"), $DBIntranetDB);  
    $to_email= CCDlookUp("email", "employees", "emp_id=" . $tasks->{user_id_assign_to}->GetValue(), $DBIntranetDB);  
    $subject = "A task was assigned to you";  
    $message = "The following task was assigned to you:<br><br>" . "Task ID: " . CCGetFromGet("task_id", "") .  
    "<br><br>" . $tasks->{task_desc}->GetValue();  
    if(open(SENDMAIL, "| sendmail $from_name")) {  
        print SENDMAIL "From: $from_email\nTo: $to_email\nSubject: $subject\n\n$message\n.\n";  
        close(SENDMAIL);  
    }  
}
```

Les principales différences entre le code ci-dessus et celui utilisé dans l'événement *After Insert* sont les suivantes :



- 1 - Une condition *if* a été ajoutée pour envoyer un email seulement si un utilisateur attribue une tâche à une autre personne.
- 2 - *task_id* est récupéré depuis l'URL utilisant la fonction *CCGetFromGet*. Nous pouvons utiliser cette méthode parce que des tâches peuvent être mises à jour seulement si l'utilisateur est arrivé à la page en cours par une URL qui contient un ID de tâche à mettre à jour. Une telle URL ressemble à : http://localhost/TaskManager/tasks_maint.cgi?task_id=9

Test de l'Envoi d'Email

Avant de tester l'envoi d'email :

- 1 - Vous devez ajouter de nouveaux utilisateurs à votre base de données avec des adresses email correctes, ou modifier les utilisateurs existants en changeant leur adresse email :

A - Vous pouvez le faire en ouvrant la base de données *Intranet.mdb* qui se trouve dans votre répertoire *Projet*.

B - Autrement, vous pouvez utiliser le Gestionnaire de Tâches. Allez à la page *Employés* pour y visualiser les emails utilisateur et les modifier.

- 2 - Une fois vos utilisateurs configurés avec le test d'emails, sauvegardez votre projet et passez en mode **Live Page** pour tester votre système.

Remarque : Vous aurez besoin de Microsoft Access 2000 ou une version supérieure pour éditer manuellement le fichier base de données. Si votre code email fonctionne correctement, vous devez alors vous retrouver sur la page *Liste de Tâches* après l'ajout ou la modification d'une tâche, et la personne à qui la tâche a été attribuée doit recevoir un email.

Etape 5

Implémenter la Sécurité d'Enregistrement dans l'Événement After Initialize

Votre Gestionnaire de Tâches est désormais pratiquement complet, seule une fonctionnalité pouvant s'avérer importante manque, à savoir la Sécurité. Dans l'état actuel du gestionnaire, n'importe qui peut modifier et supprimer n'importe quelle tâche. Vous voudrez certainement limiter l'accès pour que seuls les employés autorisés puissent mettre à jour leurs tâches. Il y a plusieurs façons de procéder, et nous allons examiner plusieurs d'entre elles.

- 1 - Cliquez sur la page *tasks_maint* dans l'**Explorateur de Projet**.
- 2 - Dans la fenêtre **Propriétés**, sélectionnez l'onglet **Événements**.
- 3 - Ajoutez **Custom Code** (Code Personnalisé) à l'événement **After Initialize** de la page. Procédez comme suit :
- 4 - Une fois en mode **Visualisation du Code**, remplacez le commentaire généré :

```
# Write your own code here.
```



Par ce code :

```
$DBIntranetDB = clsDBIntranetDB->new();
$current_task = CCGetParam("task_id", "");
$task_user_id = CCDlookUp("user_id_assign_to", "tasks", "task_id=" .
$DBIntranetDB->ToSQL($current_task, $ccsInteger), $DBIntranetDB);

if ( ($current_task != 0) && (CCGetSession("UserID") != $task_user_id) )
{
    $tasks->{Visible} = 0;
    # $Redirect = "tasks_list.cgi";
}
```

Le code du dessus vous permet de tester les différentes méthodes pour implémenter la sécurité d'enregistrement :

Ne pas montrer la Tâche (formulaire Enregistrement) sur la page si la tâche sélectionnée n'appartient pas à l'utilisateur connecté. Un utilisateur non autorisé verra une page blanche.

Vous pouvez cacher n'importe quel formulaire sur une page en lui attribuant une valeur 0 à la propriété *Visible* (visible) du formulaire. Le code *\$current_task != 0* dans la condition *if* indique que le code sera seulement exécuté si l'utilisateur essaie de modifier une tâche existante et qui ne lui est pas attribuée. La condition *if* s'assure également que tous les utilisateurs peuvent créer de nouvelles tâches. Vous pouvez tester cette fonctionnalité en insérant le code du dessus, en passant en mode **Live Page**, et en essayant de modifier une tâche qui ne vous est pas attribuée ; vous verrez alors une page vide. Bien que cette fonctionnalité puisse ne pas être très subtile, elle illustre comment cacher des formulaires sur une page. Ajoutez un autre formulaire Enregistrement sur la page, qui ne puisse pas être mis à jour et qui aient juste des champs Etiquette qui montrent les informations. Une fois que vous avez vos deux formulaires sur la page, vous pouvez cacher chaque formulaire par programmation en utilisant des critères d'exclusion.

Rediriger des utilisateurs non autorisés sur une autre page. Seuls les utilisateurs à qui les tâches sont attribuées peuvent visualiser la page.

Vous pouvez mettre en application et tester cette fonctionnalité en modifiant légèrement le code du dessus comme montré ci-dessous :

```
$DBIntranetDB = clsDBIntranetDB->new();
$current_task = CCGetParam("task_id", "");
$task_user_id = CCDlookUp("user_id_assign_to", "tasks", "task_id=" .
$DBIntranetDB->ToSQL($current_task, $ccsInteger), $DBIntranetDB);

if ( ($current_task != 0) && (CCGetSession("UserID") != $task_user_id) )
{
    # $tasks->{Visible} = 0;
    $Redirect = "tasks_list.cgi";
}
```



Ce code montre que vous devriez commenter la ligne précédemment active, et activer la ligne qui commence avec *\$Redirect*. *\$Redirect* est une variable utilisée par CodeCharge Studio qui détermine si la page en cours doit être redirigée sur une autre page, par exemple, dans le cas où un utilisateur n'aurait pas de session ouverte. Cette variable peut être utilisée seulement sur les pages ayant un accès restreint et nécessitant une authentification. Vous pouvez simplement attribuer la page de destination à la variable *\$Redirect* et la page sera alors automatiquement redirigée. Testez cette fonctionnalité en modifiant le code comme montré, en passant en mode **Live Page**, et enfin en essayant de modifier une tâche qui ne vous est pas attribuée.

CONCLUSION

Ce didacticiel vous a permis d'utiliser le Générateur d'Application pour créer un Gestionnaire de Tâches. Bien que de nombreuses autres caractéristiques et améliorations puissent être ajoutées à votre application, maintenant vous vous êtes familiarisé(e) avec l'interface de CodeCharge Studio et un grand nombre de ses fonctionnalités.

Pour plus d'informations sur des points spécifiques, merci de vous référer au Guide Utilisateur de CodeCharge Studio.

