



FAQ EasyBuilderPro

Gestion de recettes par base de données

KEPFrance SAS
Z.A. Belle Aire
3 rue Vasco de Gama
17440 AYTRE
Tél : 05 46 07 44 40
Fax : 05 46 07 44 44
Site internet : www.kepfrance.fr
E-mail commercial : commercial@kepfrance.fr
E-mail technique : technique@kepfrance.fr

1. Introduction	3
1.1 Nouvelle gestion.....	3
1.2 Déclaration recette.....	3
1.3 Charger / Récupérer une base de données recette	6
2. Créer une base de données recette	7
2.1 Paramétrer la structure de la recette	7
2.2 Eléments de recettes	8
2.3 Object Afficheur de recettes	10
2.4 Editeur base de données recette.	11
3. Utiliser l’afficheur de recette.....	12
3.1 Visualiser la base de données	12
3.2 Modifier une recette en ligne sur le pupitre.....	14
3.3 Charger une recette dans l’automate.....	15
3.4 Recherche de recette dans une macro.....	16

1. Introduction

1.1 Nouvelle gestion

EasyBuilderPro propose, en plus de l'ancienne gestion des recettes par registres RW et RW_A, une nouvelle manière de gérer les différentes recettes contenues dans le pupitre. Les recettes sont maintenant stockées dans une base de données, ceci permet une manipulation des données plus intuitive et offre la possibilité d'utiliser des fonctions avancées, telles que le tri par colonne, la recherche par nom, éléments, via des requêtes SQL.

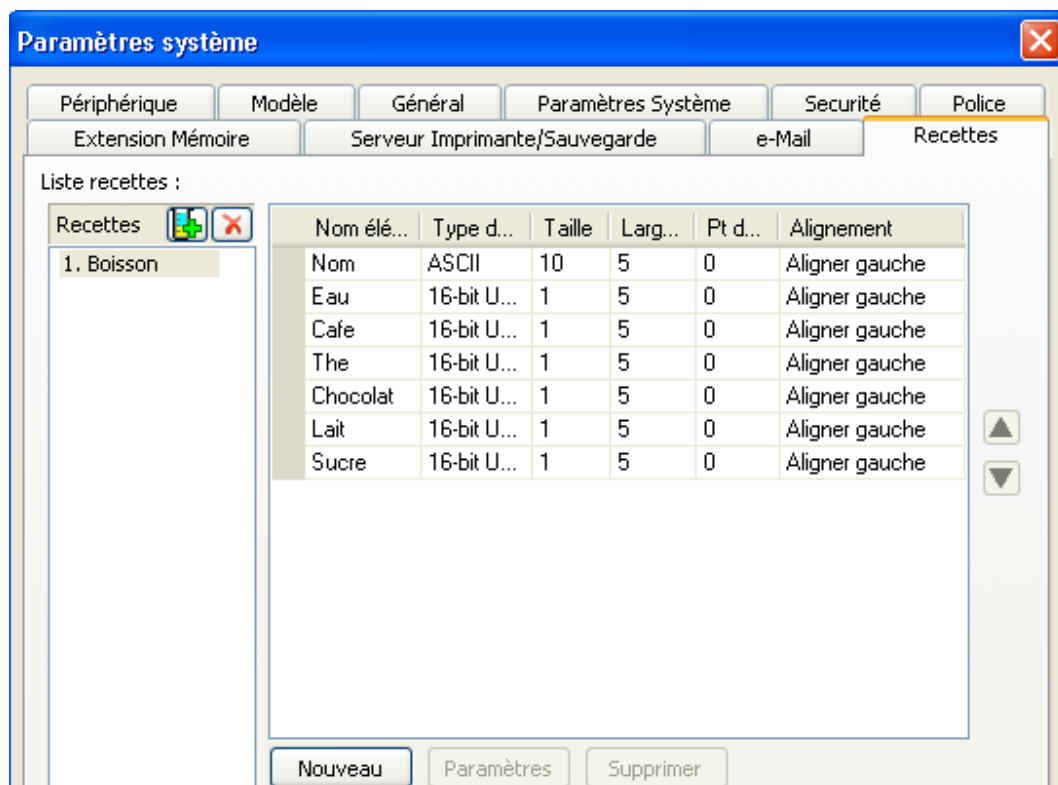
Le système gère automatiquement la place occupée par les différents éléments de la recette, il n'est donc plus nécessaire de calculer les adresses occupées, ni de prévoir des emplacements vides dans le cas où le nombre d'éléments augmente.

La déclaration des recettes se fait en deux temps : la création de la table de recettes, puis la saisie des éléments. La visualisation et la manipulation des recettes sur le pupitre se fait directement via l'objet visualisateur de recette. Cet objet permet l'affichage de l'ensemble des recettes sous forme d'un tableau. Les requêtes SQL se font via des commandes macros.

1.2 Déclaration recette

A. Structure de la recette.

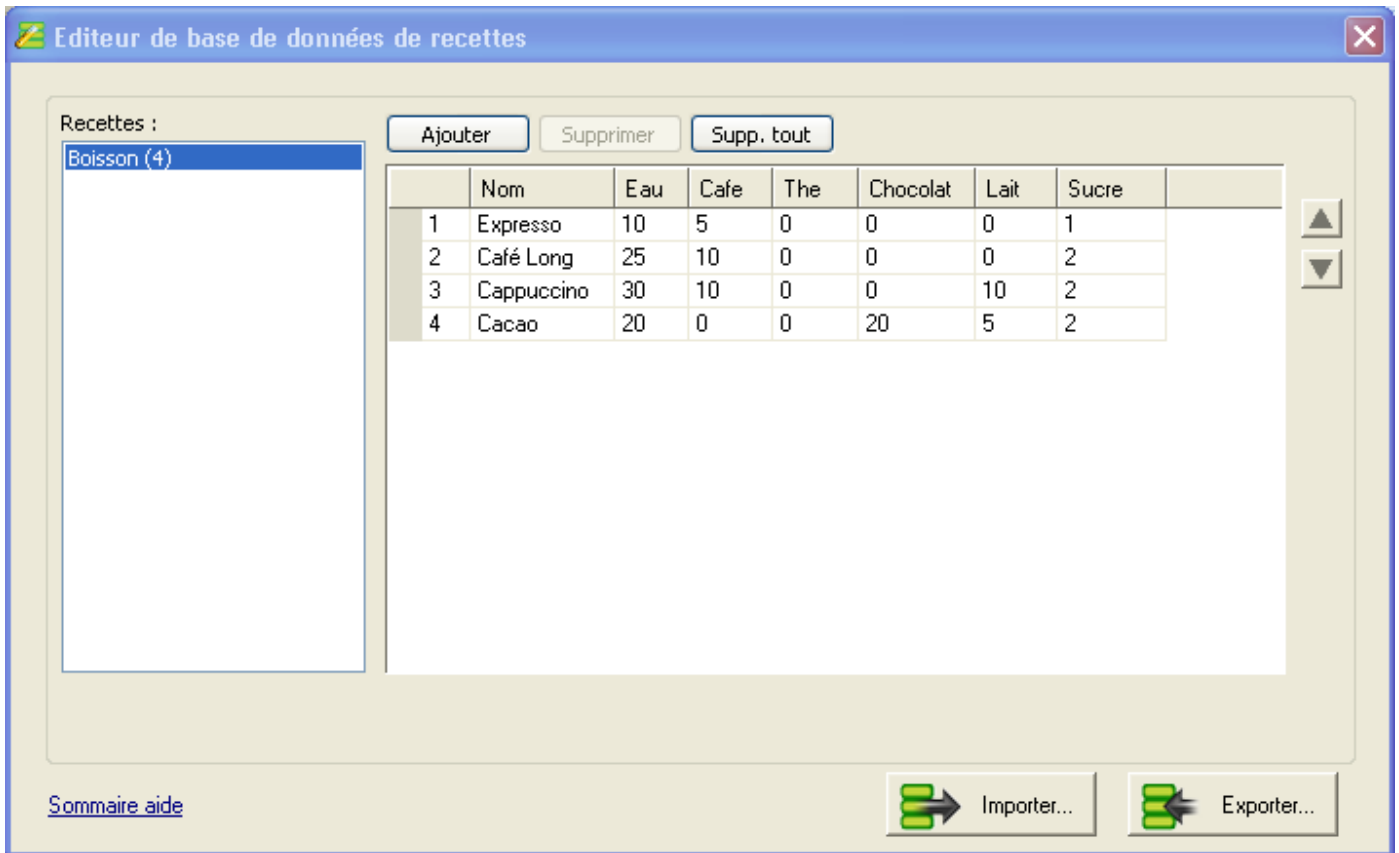
Le menu Edition > Paramètres Système contient un nouvel onglet Recettes.



Il est possible depuis cet onglet de créer plusieurs tables (structures) de recettes différentes. Une fois tous les éléments créés, cliquer sur « exporter fichier définition des recettes » pour sauvegarder la structure de la recette (format RDEF). Ceci permet d'importer cette structure de recette d'un projet.

B. Enregistrements.

L'outil d'édition de base de données recettes permet d'ajouter, modifier et supprimer des recettes. Il se trouve dans le menu outil > éditeur base de données recettes.



Cliquer sur ajouter pour créer une nouvelle recette, chaque élément est accessible en édition (ici les champs Nom, eau, café, thé, chocolat, lait et sucre).

Les flèches haut et bas permettent de déplacer une recette dans la liste.

Cliquer sur exporter pour enregistrer le fichier de recette. Il est ainsi possible de charger cette base de données dans un pupitre. Le format de la base de données de recette est .DB. Il est possible d'éditer ce fichier avec un éditeur de base de données.

Le logiciel « SQLite Database Browser » se trouve en téléchargement gratuit sur l'espace client du site www.kepfrance.fr.

Cliquer sur importer pour aller chercher un fichier de structure RDEF (voir point précédent) ou directement une base de données DB.

C. Visualisateur de recette.

Nom	Eau	Cafe	The	Chocolat	Lait	Sucre
Expresso	10	5	0	0	0	1
Cappuccino	30	10	0	0	10	2
Cafe Long	25	10	0	0	0	2
Cacao	20	0	0	20	5	2

L'objet visualisateur de recettes permet d'afficher sur le pupitre une table contenant l'ensemble des recettes. L'utilisateur peut l'utiliser pour choisir la recette souhaité et la charger, modifier via les registres de contrôle associés (voir chapitre 3).

D. Fonction de recherche dans la base de données.

Quatre fonctions portant sur les recettes ont été ajoutées dans les macros :

1. RecipeGetData : cette fonction permet de récupérer une recette de la base de données.
2. RecipeQuery : cette fonction permet d'interroger la base de données selon une requête SQL.
Le résultat de la recherche est sous forme d'une table contenant l'ensemble des recettes répondant aux critères de recherche.
3. RecipeQueryGetData : cette fonction permet de récupérer une recette de la liste des résultats obtenue via la recherche faite par la fonction précédente RecipeQuery..
4. RecipeQueryGetRecordID : cette fonction permet de récupérer le numéro d'emplacement d'une recette à partir du résultat de la fonction 2.

Ces fonctions sont détaillées dans la suite de ce document.

1.3 Charger / Récupérer une base de données recette

Le transfert des recettes dans le pupitre se fait directement depuis l'utility manager de la même façon que pour transférer un projet, les historiques d'événements/courbes ...

Charger

☐ Firmware eMMI9000 SERIES

☐ Projet

☐ RW

☐ RW_A

☒ Base données recette C:\Recettes.db Sélectionner...

☐ Données

☐ Ecran de demurrage

Connexion

☐ Ethernet ☒ Câble USB

☒ Relancer MMI après transfert

☐ RAZ recette ☐ RAZ base de données recette ☐ Supprimer écran de démarrage

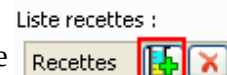
☐ RAZ historique ☐ RAZ données ☐ RAZ journal opération

Charger Stop Paramètres... Quitter

2. Créer une base de données recette

2.1 Paramétrer la structure de la recette

Aller dans Edition > Paramètres Système, puis aller dans l'onglet Recettes. Cliquer sur l'icône pour ajouter une nouvelle structure de recette.



Note : Règles à respecter :

- Le nom d'une structure de recette est unique
- Seuls les caractères alphanumériques sont supportés (ceci exclus les caractères spéciaux comme les accents, cyrilliques).
- Le nombre maximum de structure de recettes est 100 par projet.

Cliquer sur **nouveau** pour ajouter un nouvel élément dans la recette. (Note : le nouvel élément créé sera du même type que l'élément actuellement sélectionné).

Cliquer sur **paramètres** pour pouvoir modifier cet élément :

Information élément recettes

Nom :

Type d'affichage :

Taille élément (mots) :

Largueur d'affichage (car) :

Pt décimale :

Alignement :

Nom	Nom de l'élément de recette. Caractères alphanumérique et « _ » uniquement.
Type d'affichage	Format de l'élément (chaîne de caractères, flottant, signé, non signé ...). *
Taille élément (mots)	Nombre de caractères, pour le type chaîne de caractères. (max 255).
Largueur d'affichage (car)	Nombre de caractères visibles dans l'objet visualisateur de recettes
Pt décimal	Nombre de chiffre à droite de la décimale
Alignement	Justification du texte (aligné à gauche, centré, aligné à droite)

* 14 types en tout :

- 16 bits BCD
- 32 bits BCD
- 16 bits Hex
- 32 bits Hex
- 16 bits Binary
- 32 bits Binary
- 16 bits Unsigned

- 32bits Unsigned
- 16 bits Signed
- 32 bits Signed
- 32 bits Float
- ASCII
- Unicode
- Inversion poids fort / faible

Cliquer sur **supprimer** pour effacer l'élément sélectionné.

Exporter l'architecture de recette :

Cliquer sur **Exporter fichier définition des recettes** pour sauvegarder l'architecture au format RDEF afin de pouvoir utiliser ce format de ce recettes dans d'autres projets.

L'exemple de recette de ce document est celui d'une machine à boisson chaude.
L'architecture de la recette est la suivante :

Élément	Nom de l'élément	Format
1	NOM	Chaine ASCII de 30 caractères (15 mots)
2	CAFE	Entier 16 bits non signé (1 mot)
3	THE	Entier 16 bits non signé (1 mot)
4	CACAO	Entier 16 bits non signé (1 mot)
5	SUCRE	Entier 16 bits non signé (1 mot)
6	LAIT	Entier 16 bits non signé (1 mot)
7	EAU	Entier 16 bits non signé (1 mot)

Une recette a donc une longueur de 21 mots consécutifs.

2.2 Eléments de recettes

Aller dans le menu librairie et cliquer sur enregistrement recette pour ouvrir le tableau suivant.

Recettes : sont affichées ici les différentes architectures de recettes déclarées.

Ajouter / Supprimer : Cliquer sur ajouter pour créer un nouvel enregistrement. Cliquer sur supprimer pour effacer l'enregistrement sélectionné.

Cliquer sur les flèches pour déplacer l'enregistrement sélectionné.

Editer Recette ✕

Recettes :

Boisson (10)

Ajouter

Supprimer

	NOM	CAFE	THE	CACAO	SUCRE	LAIT	EAU
▶ 1	Cappuccino_25ml	5	0	0	2	3	15
2	Cappuccino_50ml	10	0	0	4	6	30
3	The_25ml	0	5	0	2	3	15
4	The_50ml	0	10	0	4	6	30
5	Cafe_au_lait_25ml	6	0	0	1	2	15
6	Cafe_au_lait_50ml	12	0	0	2	4	30
7	Chocolat_25ml	0	0	6	1	2	15
8	Chocolat_50ml	0	0	12	2	4	30
9	Expresso_sucre	3	0	0	1	0	5
10	Expresso	3	0	0	0	0	5

Exporter...

Importer...

Type donnée = '16-bit Unsigned'
Pt décimale. = '0'

OK

Annuler

▲

▼

Exporter / Importer : Cliquer sur exporter pour sauvegarder la base de données courante au format BD. Cliquer sur importer pour ouvrir un fichier RDEF ou BD existant.

Information sur l'élément sélectionné

Note : Règles à respecter :

- Chaque recette peut contenir au maximum 10 000 enregistrements.
- A l'importation d'un fichier RDEF ou BD, la structure actuelle ainsi que la base de données courante sera écrasée par la nouvelle.
- Pour effacer la base de données de recette du le pupitre lors du chargement du projet, cocher la case RAZ base de données recette. La case RAZ recette va remettre à zéro les registres RW et RW_A (ancienne gestion de recettes).

Chargement

☒ Ethernet ☐ Câble USB Mot de passe :

◀ **IP** **Nom** ▶

Nom MMI: 192.168.0.78 (Default HMI)

☒ Firmware ☒ Fichiers police
* Obligatoire si mise à jour du firmware ou premier téléchargement

☐ Utiliser écran de démarrage personnalisé

☒ RAZ recette ☒ RAZ journal événements ☒ RAZ journal données
☒ RAZ base données recette ☒ RAZ journal opération

☒ Rebooter MMI après le transfert
☐ Utiliser les paramètres courants de chargement après la compilation

2.3 Object Afficheur de recettes

L'objet afficheur de recettes permet l'affichage de la base de données sur le pupitre sous forme de tableau. Il est nécessaire d'avoir au moins un enregistrement pour pouvoir utiliser cet objet.

Il est possible de modifier la couleur des éléments suivants :

Titre

Contrôle de sélection : couleur recette sélectionnée

Nom	Eau	Cafe	The	Chocolat	Lait	Sucre
Expresso	10	5	0	0	0	1
Cappuccino	30	10	0	0	10	2
Cafe Long	25	10	0	0	0	2
Cacao	20	0	0	20	5	2

Profil : cadre et fond

Grille

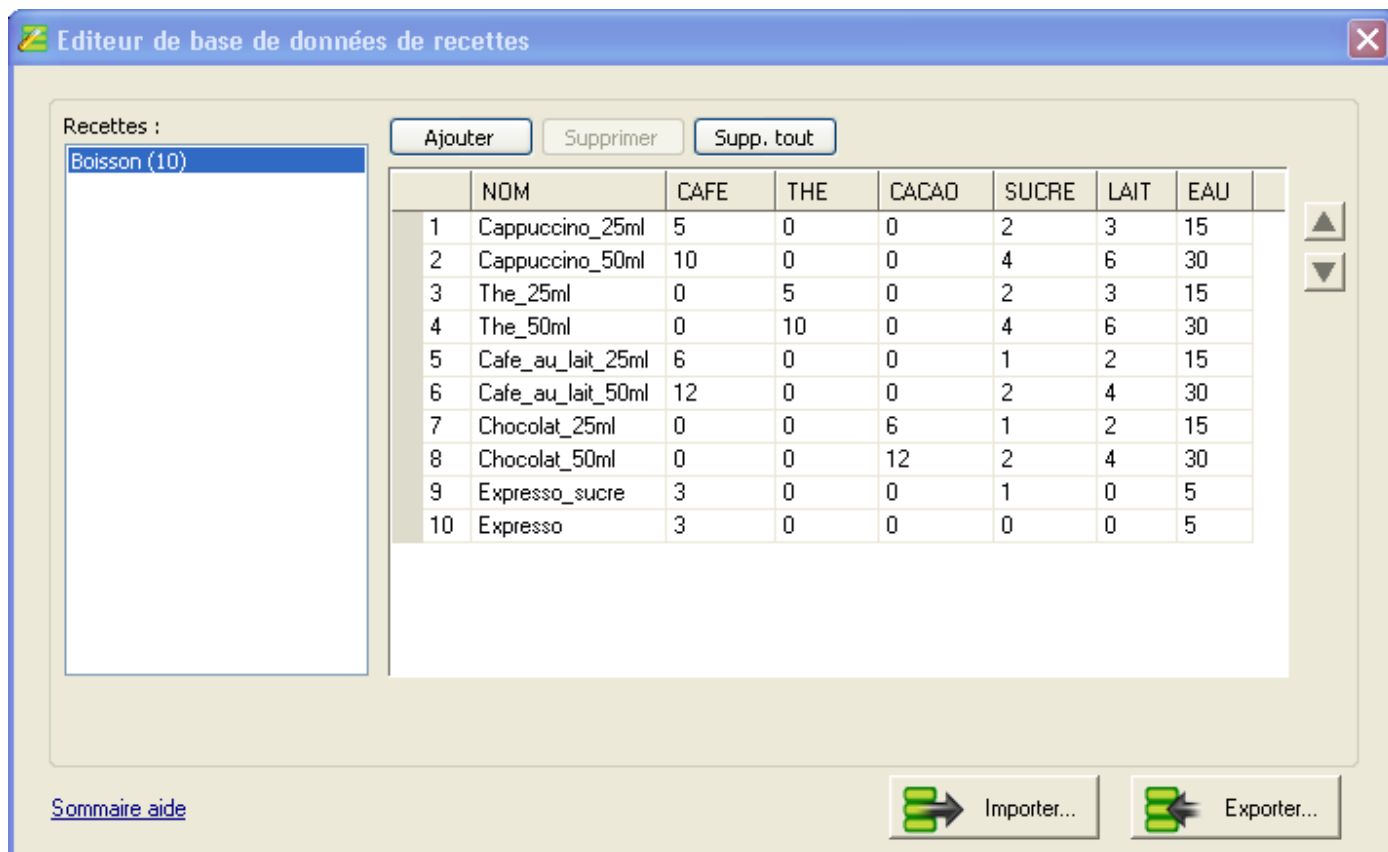
Table recette : Sélectionner la base de données de recette à visualiser.	Table recette Nom recette : Boisson
Titre : Configurer la couleur du bandeau de titre.	Titre ☐ Transparant Couleur :
Profil : Configurer la couleur du cadre et du fond de l'objet.	Profil ☐ Transparant Cadre : Fond d'écran :
Grille : Configurer la couleur des traits de séparation des lignes et colonnes.	Grille ☒ Activer Couleur :
Contrôle de sélection : Configurer la couleur du fond de la recette sélectionnée	Contrôle sélection Couleur :
Tri par défaut : Configurer le tri à l'ouverture de la page : nom de la colonne et sens de tri. Par la suite, l'utilisateur peut trier les recettes comme il le souhaite en cliquant sur le bandeau de titre.	Tri par défaut ☒ Activer Trier par : NOM ☐ Montant ☒ Descendant

2.4 Editeur base de données recette.

L'éditeur se trouve dans le menu Outils > Editeur base de données recette.

A la différence de l'éditeur du point 2.2, celui-ci ne nécessite pas qu'un projet EBPro soit ouvert.

Les fonctions sont similaires entre les deux éditeurs.

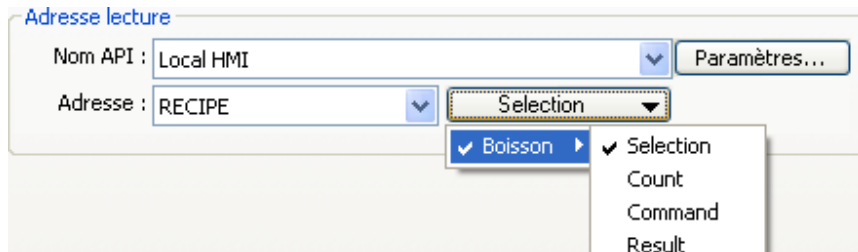


Cliquer sur Importer pour ouvrir un fichier RDEF ou DB.

3. Utiliser l'afficheur de recette

3.1 Visualiser la base de données

Pour manipuler la base de données (ajouter, modifier ou supprimer une recette, connaître le nombre de recette, etc), il faut utiliser les registres : Recipe.Selection, Recipe.Count, Recipe.Command et Recipe.Result.



Voici le détail de ces nouveaux registres.

Selection :

Ce registre contient l'index de la recette dans la base de données. Les recettes sont numérotées à partir du chiffre zéro. Ainsi, la première recette aura pour index 0, la seconde aura pour index 1, la troisième aura pour index 2 etc...

Count :

Ce registre contient le nombre total de recettes contenues dans la base de données.

Command :

Ce registre est à utiliser pour ajouter, modifier ou supprimer une recette. Selon la valeur écrite dans ce registre, le pupitre va faire une action propre. Les valeurs possibles sont :

Valeur	Action
1	Ajouter une nouvelle recette. La recette sera stockée en dernière position. (Son index sera égal à la valeur actuelle de Count).
2	Mise à jour de la recette sélectionnée. La recette sélectionnée sera remplacée par les nouvelles valeurs.
3	Suppression de la recette sélectionnée.
4	Suppression de toutes les recettes.

L'utilisateur est libre de la manière dont l'écriture se fait dans le registre Recipe.Command : objet Action Mot, Entrée Numérique, Menu Déroulant ...

Result :

Ce registre est mis à jour après l'exécution d'une commande. Il indique le résultat de cette commande. Les valeurs possibles sont :

Valeur	Résultat
1	La commande a été correctement exécutée.
2	Le numéro d'index indiqué n'existe pas.
4	Commande inconnue
8	Le nombre maximal de recette est atteint, impossible d'en ajouter une recette supplémentaire.

Pour rappel, le nombre maximal de recette est de 10 000.

Ingrédient de recette :

Les différents éléments de recette sont également accessibles via le registre Recipe.xxx. Un filtre est appliqué en fonction du format. Par exemple, un élément déclaré en temps que chaîne de caractères ASCII n'est visible que pour les objets afficheur et saisie ASCII. Un élément numérique sera accessible par tous les objets numériques

Adresse lecture

Nom API : Local HMI

Adresse : RECIPE

Selection

✓ Boisson

NOM

Adresse lecture

Nom API : Local HMI

Adresse : RECIPE

SUCRE

✓ Boisson

Selection

Count

Command

Result

CAFE

THE

CACAO

✓ SUCRE

LAIT

EAU

Notification

☐ Actif

Exemple de base de données avant et après écriture dans le mot Command.

Exemple : 1 : ajout

Base de données avant commande ajout								Base de données après commande ajout							
Index	Nom	Cafe	The	Cacao	Sucre	Lait	Eau	Index	Nom	Cafe	The	Cacao	Sucre	Lait	Eau
0	Expresso	5	0	0	1	0	10	0	Expresso	5	0	0	1	0	10
1	Chocolat	0	0	10	2	20	10	1	Chocolat	0	0	10	2	20	10
-	-	-	-	-	-	-	-	2	Thé_25ml	0	7	0	2	0	25

Exemple : 2 : modification (recette sélectionnée en bleu)

Base de données avant commande ajout								Base de données après commande ajout							
Index	Nom	Cafe	The	Cacao	Sucre	Lait	Eau	Index	Nom	Cafe	The	Cacao	Sucre	Lait	Eau
0	Expresso	5	0	0	1	0	10	0	Expresso	5	0	0	1	0	10
1	Chocolat	0	0	10	2	20	10	1	Chocolat	0	0	10	2	30	0

Exemple : 3 : suppression (recette sélectionnée en bleu)

Base de données avant commande ajout								Base de données après commande ajout							
Index	Nom	Cafe	The	Cacao	Sucre	Lait	Eau	Index	Nom	Cafe	The	Cacao	Sucre	Lait	Eau
0	Expresso	5	0	0	1	0	10	0	Chocolat	0	0	10	2	20	10
1	Chocolat	0	0	10	2	20	10	-	-	-	-	-	-	-	-

Exemple : 4 : suppression de toute la base

Base de données avant commande ajout								Base de données après commande ajout							
Index	Nom	Cafe	The	Cacao	Sucre	Lait	Eau	Index	Nom	Cafe	The	Cacao	Sucre	Lait	Eau
0	Expresso	5	0	0	1	0	10	-	-	-	-	-	-	-	-
1	Chocolat	0	0	10	2	20	10	-	-	-	-	-	-	-	-

Gestion de recettes par base de données

3.2 Modifier une recette en ligne sur le pupitre

Pour modifier une recette directement depuis le pupitre, créer autant d'objet que d'éléments à modifier.
Dans notre exemple il faut :

Elément	Nom de l'élément	Exemple d'objets à paramétrer
1	NOM	Entrée ASCII
2	CAFE	Entrée numérique
3	THE	Entrée numérique
4	CACAO	Entrée numérique
5	SUCRE	Entrée numérique
6	LAIT	Entrée numérique
7	EAU	Entrée numérique

Créer ensuite un objet Visualisateur de recette.

Enfin créer un objet Action mot sur le registre Recipe.Command, pour forcer la valeur 2.

Mettre mot Propriétés de l'objet

Général Sécurité Forme Label Profil

Description :

Adresse écrite

Nom API : Local HMI

Adresse : RECIPE 16-bit Unsigned

☐ Ecrire après l'appui du bouton

Notification

☐ Actif

Attribut

Mode : Ecrire valeur constante

Mettre la valeur : 2

Note : la base de données est automatiquement sauvegardée lors d'une modification. Il n'est plus nécessaire d'utiliser le bit système LB 9029 pour enregistrer les recettes.

3.3 Charger une recette dans l'automate

Pour transférer une recette dans l'automate, utiliser l'objet transfert de données.

Cet objet permet de copier un bloc de registre dans un autre. Configurer le bloc source Recipe, et configurer l'adresse de destination de votre automate.

Note : le nombre de mots copiés dépend de la longueur de votre recette.

Dans notre exemple, une recette comprend 21 mots consécutifs.

The screenshot shows a software window titled "Nouveau Transfert de données (interrupteur) Objet". It has four tabs: "Général", "Sécurité", "Forme", and "Label". The "Général" tab is active. Inside the window, there is a "Description :" label followed by an empty text box. Below this, there are two sections. The first section, "Adresse source", contains "Nom API :" with a dropdown menu showing "Local HMI" and a "Paramètres..." button, and "Adresse :" with a dropdown menu showing "RECIPE" and another dropdown menu showing "Boisson". The second section, "Adresse destination", contains "Nom API :" with a dropdown menu showing "MODBUS RTU" and a "Paramètres..." button, and "Adresse :" with a dropdown menu showing "4x" and a text box containing "400".

3.4 Recherche de recette dans une macro

Les recettes étant stockées dans une base de données, les requêtes de recherche se font via des commandes SQL. Quatre fonctions prédéfinies sont disponibles pour manipuler la base de données et les résultats des requêtes.

1. RecipeGetData : cette fonction permet de récupérer une recette de la base de données.
2. RecipeQuery : cette fonction permet d'interroger la base de données selon une requête SQL.
Le résultat de la recherche est sous forme d'une table contenant l'ensemble des recettes répondant aux critères de recherche.
3. RecipeQueryGetData : cette fonction permet de récupérer une recette de la liste des résultats obtenue via la recherche faite par la fonction précédente RecipeQuery..
4. RecipeQueryGetRecordID : cette fonction permet de récupérer le numéro d'emplacement d'une recette à partir du résultat de la fonction 2.

Nom	RecipeGetData
Syntaxe	RecipeGetData(destination, registre_recette, index_rec)
Description	Lecture de la valeur d'un ingrédient. La valeur est stockée dans la variable destination. Registre_recette correspond à un ingrédient de la recette (ex. Recipe.CAFE). Index_rec est le numéro d'index de la recette concernée.
Exemple	<pre>macro_command main() short data=0 char str[20] short index bool resultat index=0 resultat=RecipeGetData(data, "Boisson.CAFE", index) // lecture de l'ingrédient CAFE de la recette 0 index=1 resultat=RecipeGetData(str[0], "Boisson.NOM", index) // lecture du nom de la recette 1 end macro_command</pre>

Nom	RecipeQuery
Syntaxe	RecipeQuery(requete_SQL, nb_resultat)
Description	Exécute la requête SQL placée en paramètre. Le nombre de résultat trouvé sera stocké dans le paramètre nb_resultat. La commande SQL peut être une chaîne de caractère statique ou un variable déclaré en temps que table de caractères. Ex : RecipeQuery(“SELECT * FROM Boisson WHERE CAFE>0”, nb_resultat) ou : RecipeQuery(requete[0], nb_resultat) Cette fonction retourne également un tableau contenant les index des recettes répondant à la requête. Dans le cas où nb_resultat est différent de 0. Ce tableau contient nb_resultat éléments indexée de 0 à nb_resultat-1. Pour connaître les index, l'utilisateur doit utiliser la fonction RecipeQueryGetRecordID. La requête SQL doit commencer par SELECT * FROM, puis doit être suivie d'une condition de recherche.
Exemple	<pre>macro_command main() int nb_resultat=0 char requete[100]="SELECT * FROM Boisson WHERE THE>0" bool resultat resultat=RecipeQuery("SELECT * FROM Boisson WHERE CAFE>0", nb_resultat) // le nombre de recette à base de café sera stocké dans nb_resultat resultat=RecipeQuery(requete[0], nb_resultat) // le nombre de recette à base de thé sera stocké dans nb_resultat end macro_command</pre>

Nom	RecipeQueryGetRecordID
Syntaxe	RecipeQueryGetRecordID(destination, index_resultat)
Description	Cette fonction est à utiliser après la fonction RecipeQuery. Elle permet de connaître, en fonction de la case du le tableau de résultat, l'index de la recette répondant aux critères de recherche de la requête posée par RecipeQuery.
Exemple	<pre> macro_command main() short index_rec=0 short nb_resultat=0 short colonne=0 bool resultat_requete=0 resultat_requete=RecipeQuery("SELECT * FROM Boisson WHERE CAFE>0", nb_resultat) // le nombre total de recette de à base de café est stocké dans nb_resultat if resultat_requete then for colonne=0 to nb_resultat-1 RecipeQueryGetRecordID(index_rec, colonne) next colonne end if end macro_command </pre>

Exemple concret.

Prenons la base de données suivante :

Index	Eléments de la recette BOISSON						
	NOM	CAFE	THE	CACAO	SUCRE	LAIT	EAU
0	Cappuccino_25ml	5	0	0	2	3	15
1	Cappuccino_50ml	10	0	0	4	6	30
2	The_25ml	0	5	0	2	3	15
3	The_50ml	0	10	0	4	6	30
4	Cafe_au_lait_25ml	6	0	0	1	2	15
5	Cafe_au_lait_50ml	12	0	0	2	4	30
6	Chocolat_25ml	0	0	6	1	2	15
7	Chocolat_50ml	0	0	12	2	4	30
8	Expresso_sans_sucre	3	0	0	1	0	5
9	Expresso_sucre	3	0	0	0	0	5

Si on recherche toutes les boissons à base de café, la requête SQL adéquate est :

SELECT * FROM Boisson WHERE CAFE > 0

Que l'on peut traduire par :

SELECTIONNER tous DE LA TABLE Boisson OU L'INGREDIENT CAFE est supérieur à 0

Pour lancer la recherche, il faut utiliser la fonction RecipeQuery :

resultat = RecipeQuery('SELECT * FROM Boisson WHERE CAFE>0', nb_resultat)

Variable	Type	Valeur après exécution
resultat	Booléen (bool)	0 si la requête à échoué* 1 si la requête à réussi
nb_resultat	Entier 16/32 bits (short/int)	6

* une requête peut échouer si le nom de la table ou de l'ingrédient n'existe pas dans la base de données.

Cette fonction va également mettre en mémoire un tableau les index des recettes répondant aux critères de recherche. Ce tableau contient autant de case qu'il a de résultat. Ce tableau est accessible via la fonction RecipeQueryGetRecordID.

Dans cet exemple, ce tableau est :

Numéro de case	Contenu tableau (index recette répondant aux critères)
0	0
1	1
2	5
3	6
4	8
5	9