

# 配方数据库功能简介

## 应用简介

众所周知，配方在自动化领域有着不可替代的重要性，而我们威纶通为了客户提供了有效快捷建立配方数据库的解决方案。本章带大家一起了解我们的配方数据库的部分功能。

支持 eb 版本：EBproV6.04.02.676\_20201116

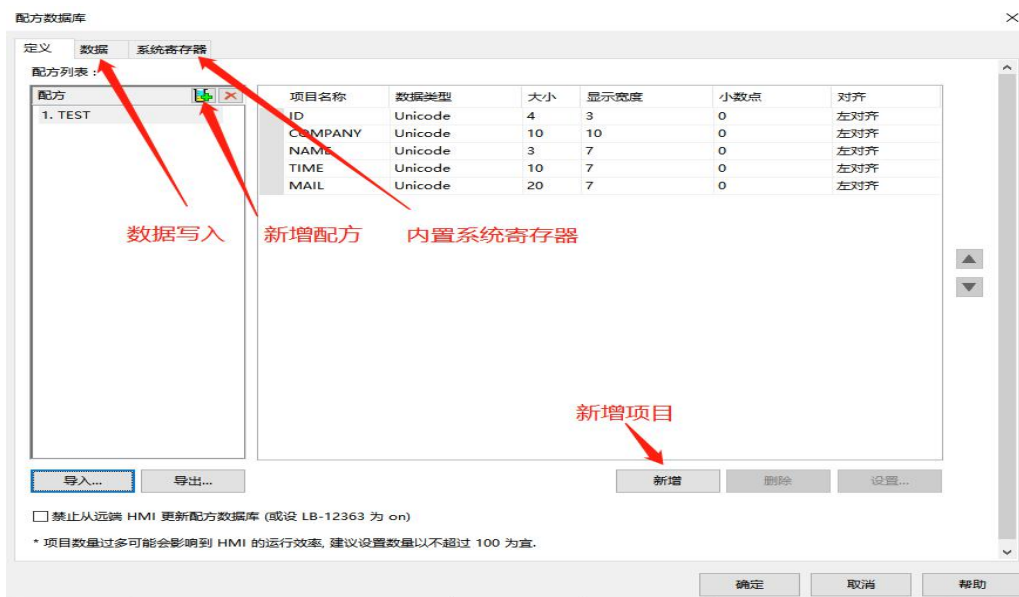
支持 HMI OS 版本：20151228 或更新版本的 OS

支持的机型：IE/CMT/EMT 系列

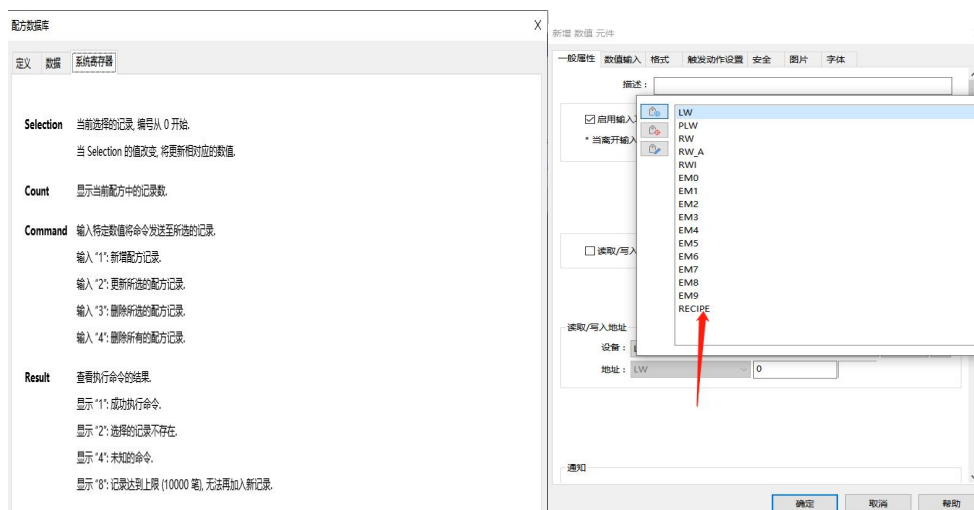
## 操作说明

首先，我们的一个配方数据最多可达 1000 个配方项目，配方项目名称仅支持英文字母或数字。注意：一个配方数据库最多可使用 2000 个字的数据，若超过此限制，我们的工程将无法成功编译。下图为配方数据库基本窗口。

- 1、在资料/历史栏目中点击配方数据库；
- 2、点击新增配方生成配方；
- 3、新增多个所需项目并定义数据类型与长度；
- 4、在数据中填入所需的实际数值。



在系统寄存器的选项中我们可以看到：



当我们将配方数据库生成的时候，数值元件里面会多出一个“RECIPE”的地址，可以在里面选择：

- 1、 selection： 输入编号来读取配方；
- 2、 command： 写入 1（新增当前数据为配方）、写入 2（更新当前所选配方的数据）、写入 3（删除所选配方）、写入 4（删除所有的配方）。

## 配方搜索

在设计时，当配方较多时，为了快速获得我们所需要的配方数据，我们还提供了宏指令来进行数据的快速定位功能。例：

- 1、 RecipeQuery("SELECT \* FROM TEST WHERE ID = %(a)", total\_row)
  - (1) TEST 为所建配方自定义的名称；
  - (2) ID 为所建配方自定义的项目；
  - (3) A 为项目下的数据，需要赋值。

此条命令：找寻 TEST 配方，项目 ID 下的为 a 中数据的个数,存放于 total\_row。

- 2、 RecipeQueryGetRecordID(data\_id,row\_number)

此条命令：获取所搜寻数据的在项目中的编号。写入至 data\_id 中。

结合前面所说的 selection（输入编号来读取配方），我们将搜索出来 data\_id（如果有多笔）写入，就可以快速的获得我们想要的的数据。

```

1:
2: macro_command main()
3: char a[4]=" " //定义查询数据
4: char b[20]=" " //定义查询后的编号显示
5: SetData(b[0], "Local HMI", LW, 100, 20) //将显示框清零
6: short c=100
7: short d=0
8: short total_row=0 //定义编号查询到的数量
9: short row_number=0
10: short id=0 //定义在配方中的实际编号
11: bool bishu=false
12: bool chaid=false
13: GetData(a[0], "Local HMI", LW, 50, 4) //获取查询的数据
14: bishu=RecipeQuery("SELECT * FROM TEST WHERE ID = %(a)", total_row) //找寻TEST配方，项目ID下的为a中数据的个数，存放于total_row。
15: SetData(total_row, "Local HMI", LW, 60, 1) //存放于lw60
16: if bishu then
17: for row_number=0 to total_row-1 step 1
18: chaid=RecipeQueryGetRecordID(id,row_number)
19: SetData(id, "Local HMI", RECIPE, "TEST.Selection") //获取所搜寻数据的在项目中的编号。写入至data_id中
20: DEC2ASCII(id, b[0], 2)
21: SetData(b[0], "Local HMI", LW, c, 2) //依次显示在lw100后20个字中
22: DELAY(0)
23: c=c+2
24: next row_number
25: end if
26: end macro_command

```

因此，通过上述操作，我们不仅可以知道该数据在配方中出现了多少次，也能准确知道其出现的位置，并通过编号将其读出，大大减少了我们的工作量。