



CREW DATABASE MANAGEMENT

QUICKSTART

Connect
Ideas.
Shape
solutions.



Table of contents

Document description	3
Application guidelines	3
Database connection	4
Enable the connection	4
Connection parameters	5
Predefined Database.....	5
Custom database	6
Data Types.....	6
Exporting.....	6
Tags	7
History alarm buffer.....	8
Data log	9
Recipe Archive.....	10
Importing	10



Document description

This documents will describe the way of getting the connection and exchange datas between the application developed with Crew and the database.

Application guidelines

The Crew runtime does include the ODBC communication protocol (Open Database Connection) and use the ODBC services in order to access the database.

In case of an HMI of the EW1xx series, the database has to run on an external server PC belonging to the same LAN network.

The server PC can be reached via a pure PING command from the HMI.



If the CREW application has been developed for a PC based device (Scada) both the Crew runtime and the database can run on the same PC.

The Ip address in order to reach the database is the localhost address (127.0.0.1).

Via this Ip different applications running on the same PC can exchange data each other.





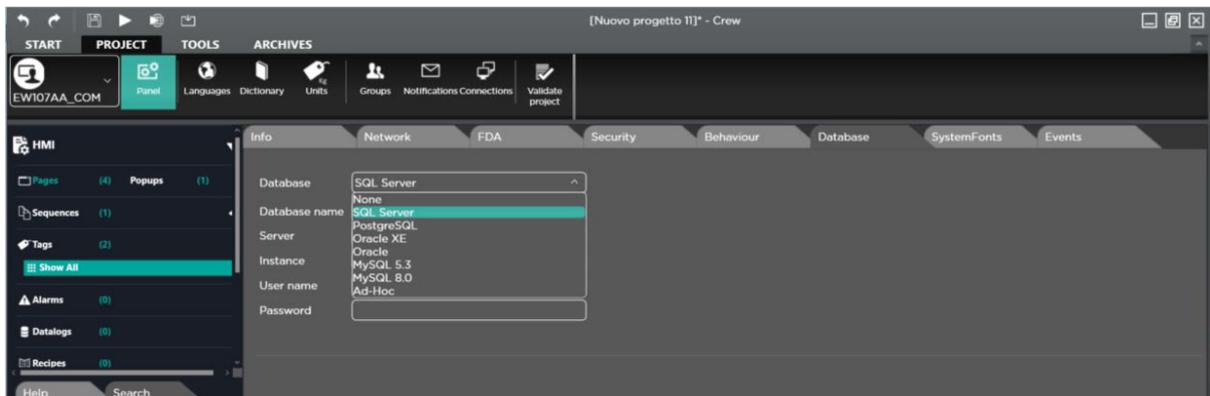
Database connection

Enable the connection

In order to enable the database connection it is necessary goto the path:

PROJECT - PANEL - DATABASE

and enable the database type to be connected.



The implemented software connectors are:

- SQL server
- PostgreSQL
- Oracle XE
- Oracle
- MySQL

In case of a different database the user can define manually the database connection string.
All the database implementing ODBC can be freely accessed.



Connection parameters

Predefined Database

The parameter values to be inserted are the same for all the different database technologies:

The screenshot shows a configuration form with the following fields and values:

- Database: SQL Server (dropdown)
- Database name: tempdb
- Server: BIZZOZERO-764
- Instance: SQLEXPRESS
- User name: diego
- Password: Dbizzozero1
- Prefix table: tab
- Language: English (United Kingdom) (dropdown)
- Server Address: BIZZOZERO-764\SQLEXPRESS
- Tags table name: Tags

Below the form, there is a summary of tables and datalogs:

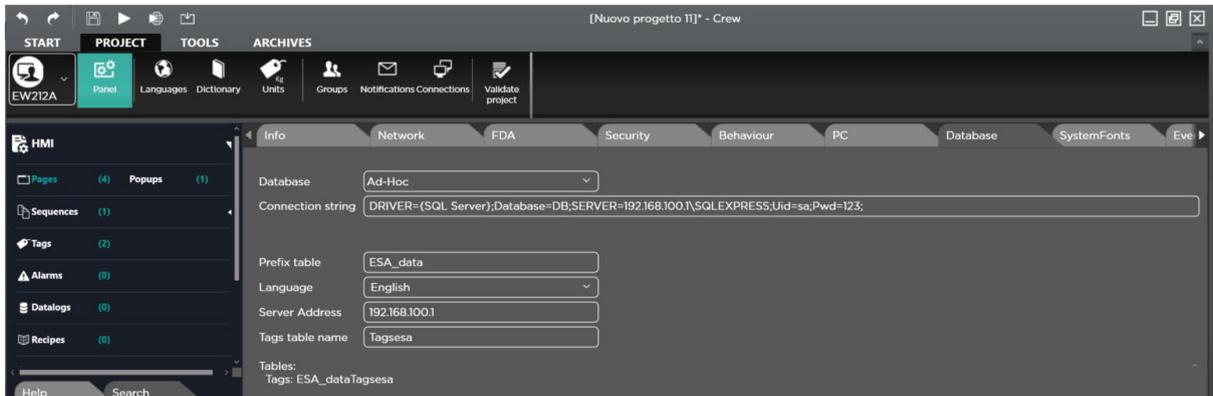
- Tables:
 - Tags: tabTags
- Datalogs:
 - PLCProductionData: tabPLCProductionData

- | | |
|------------------|--|
| Database: | Database type |
| Database name: | Name of the database to be accessed |
| Server: | PC name where the database is running.
In case both the Crew runtime and the database are running on the same device the local host address has to be inserted. |
| Instance: | Database instance |
| User name: | Crew will access the database with this user name |
| Password: | Crew will access the database with this password |
| Prefix table: | This prefix will be added to all the tables where the Crew data will be saved. |
| Server address: | In most of the real application use the same IP address of the server.
A different IP can be required in case the database is running in a virtual machine. |
| Tags table name: | The table where the tag values will be exported. |



Custom database

In case of a custom database it is necessary to insert manually the connection string. For example in order to access the SQL database via this free programming string the user has to enter the parameters as below:



All the required parameters for the connection are inside the connection string.

Data Types

In this section all the different data that can be exchanged with the database are deeply described. All the data are grouped in different families and for each family a dedicated table is created.

In the Crew project the user can enable which data has to be exported, imported and the export/import policy.

Exporting

The data families that can be exported are listed below:

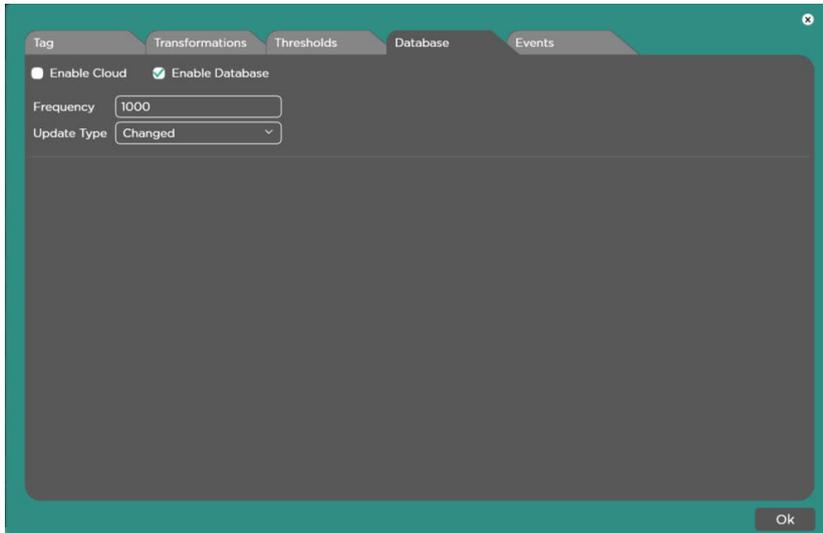
- Tags
- History alarm buffer
- Datalog
- Recipe archive

Each data family is described below.



Tags

In the tag programming window the user can define if the tag has to be exported to the database (the value is written into the tag dedicated table) and the exporting policy.



It is mandatory to enable the option Enable Database and define the exporting policy
From the ones listed below:

- Frequency: the exporting is based on a fixed timing in milliseconds (Frequency)
- Changed: the tag is exported to the database in case the value is changed.
- Thresholdabove: the tag is exported to the database in case of a value higher then the treshold value.
- Thresholdbelow: the tag is exported to the database in case of a value lower then the threshold value

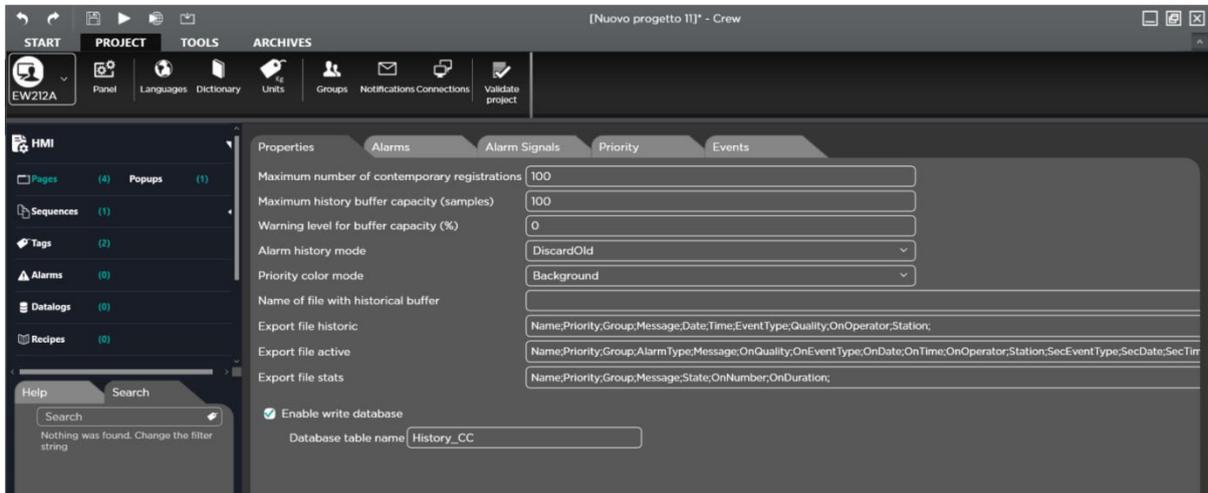
The tag value is saved into the tag dedicated table.

In case the table does not exist will be automatically created at the first export.



History alarm buffer

In the alarm programming windows (PROJECT – ALARM – PROPERTIES) the user has to enable the access to the database and define the dedicated table name.



The alarm history buffer is exported automatically every time the buffer is copied to the retentive memory.

This exporting is based on an embedded 3 minutes time out.

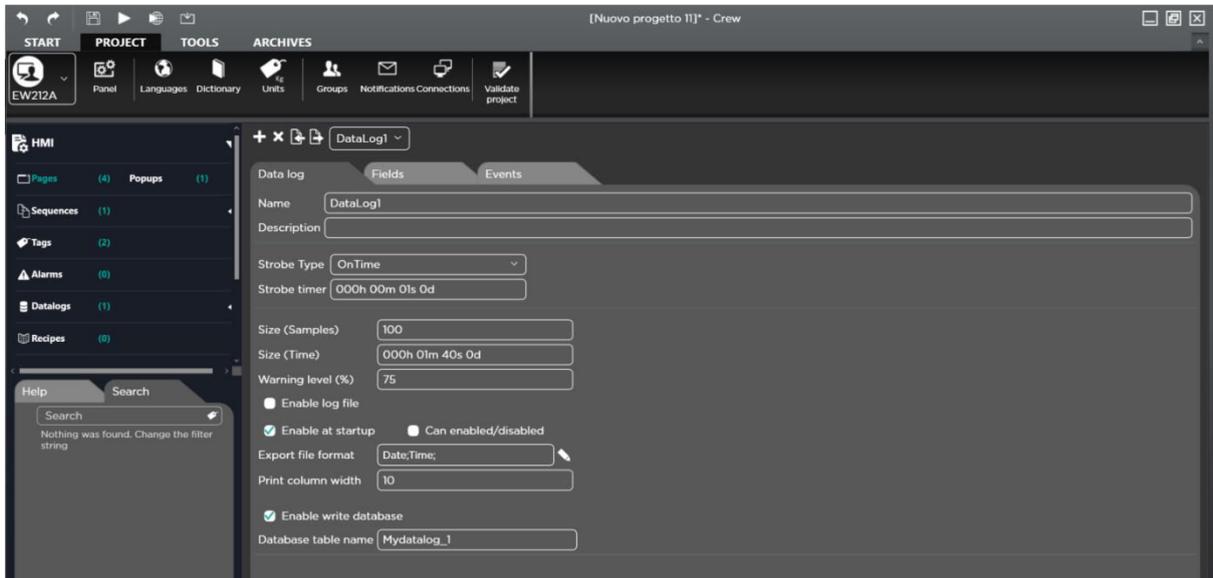
The user can force manually the exporting via the dedicated function:

HistoryExportDatabase



Data log

The user has to enable the database access in the dedicated window (PROJECT – DATALOG – DATALOG) and insert the datalog table name.



The datalog buffer is exported automatically every time the buffer is copied to the retentive memory.

This exporting is based on an embedded 3 minutes time out.

The user can force manually the exporting via the dedicated function:

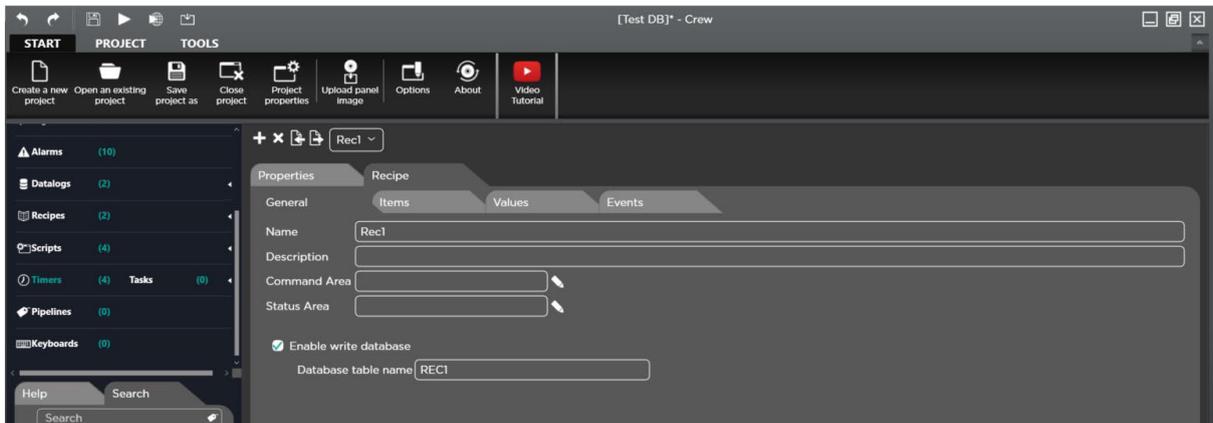
SamplesExportDatabase



Recipe Archive

The user has to enable the database support in the recipe setting window and enter the table name as usual.

The path is: PROJECT – RECIPES – GENERAL.



The recipe archive is exported to the specific database table when the exporting function is fired.

RecipeExportDatabase

Importing

The complete recipe archive can be imported from the database table to the Crew integrated archive.

This functionality can be implemented in case the customer does require a centralized management of the recipe archive.

The predefined function to be used is:

RecipeImportDatabase



Connect
ideas.
shape
solutions.

[ESA S.p.A. | www.esa-automation.com](http://www.esa-automation.com) |