R&W Recipe Designer
Recipe Management for SIMATIC WinCC

Version 4.0 SP1 Hotfix 2

Recipe Designer CS
Important note:
The software is made available to the user as it stands. All and any risks regarding the quality and the results obtained through its use remain with the user. Any relevant safety measures to insure against improper use must be taken by the user.

We shall not accept any responsibility for any damage which may ensue, either directly or indirectly, e.g. loss of profits, loss of orders or impairments of business of any kind.

© 2006 R&W Industrieautomation GmbH
Graf-Heinrich-Str. 20
57627 Hachenburg
Tel.: 02662 941434
Fax: 02662 941441
www.r-u-w.de Email: info@r-u-w.de

No part of this documentation may be duplicated or made available to third parties without specific permission in writing from R&W Industrieautomation GmbH.

We have taken great care in compiling all the details in this documentation and tested them for compliance with the described hardware and software. Nevertheless, deviations from these cannot be fully discounted. We shall accept no legal responsibility or liability for damage which may occur for this reason. We will add any necessary corrections to subsequent editions.

Windows, Windows NT and MS-DOS are either registered trade marks or trade marks of the Microsoft Corporation in the U.S.A and/or other countries.
IBM and VGA are registered trade marks of International Business Machines, Inc.
All other brand names are trade marks of their owners.

Version 4.00.01.02 01/ 2006
1. Introduction

The Recipe Designer is an add-on application for SIMATIC WinCC. The recipe handling system for WinCC enables archiving of parameterization data of a machine or system in a database. Functions are provided for loading, saving, deleting and printing as well as organizing article master data in the recipe database.

The database has the following structure:

The data sets for the parameterization of a machine, system or partial system are indexed via an product-No. or name. A corresponding comment makes it possible to describe the product more specifically. Each product is assigned a recipe containing the actual parameterization data (recipe data) of the corresponding product. This database structure makes it possible to assign recipes to more than one product-No.

![Fig. 1.1: Assignment of product to recipes](image)

The databases are registered as ODBC databases and are controlled via the database language SQL (Sybase SQL Anywhere 5.0; for WinCC 5.0 SP2 and higher Sybase Adaptive Server Anywhere 7.0). This database format enables the analysis or editing of the databases from external database applications.
2. Manner of Operation

2.1 The Basic Concept

The Recipe Designer configures project-related recipe templates. Recipe templates are independent data sets that define the structure of a recipe type for a WinCC project. The data sets of such a recipe template contain information, for example, on the tags associated with the recipe, the runtime functions, multi-language text elements and interfaces to WinCC runtime or to the AS. A runtime module of the R&W recipe handling system accesses these data sets in order to define the behavior and the manner of operation while the WinCC project is running.

2.2 WinCC Explorer

The Recipe Designer is a configuration application that is fully integrated in WinCC. In the WinCC Explorer, the Recipe Designer can be selected and started in the WinCC editor tree, just as any other editor such as the Graphics Designer.

The Recipe Designer automatically creates a path in the WinCC project where the project-related configuration data (templates) for recipes are stored. The recipe templates are displayed on the right side in the project navigation window of the WinCC Explorer.

Fig. 2.1: Integration of the Recipe Designer in the WinCC Explorer
2.3 Configuration Features

2.3.1 The Editor of the Recipe Designer

On the start page of the Recipe Designer, the basic properties of the recipe template, such as the name of the recipe template, comments, name of the database with the recipes in WinCC runtime and the path of the database with the recipes in WinCC runtime are parameterized.

The R&W recipe handling system also supports WinCC applications with a server option. This enables configuration of local recipe handling systems in a WinCC project with one server and multiple clients, in order to provide all clients with a global recipe structure (e.g. common article master data). In addition, a master database can be defined in the Recipe Designer for each client in any remote computer.

The Recipe Designer features the functions “Load”, “Save” and “Save as” for the handling of recipe templates, so that recipe templates can also be used for other projects by copying, supplementing, etc.

![Fig. 2.2: Main page of the Recipe Designer](image-url)
2.3.2 Functions of the Recipe Designer

This button starts the dialog for configuration of the WinCC tags associated with the recipe (see Ch. 2.3.2). When the tags dialog is opened, a conformity test of the tags is performed (see Ch. 2.3.3).

This button opens the dialog of the runtime layout for configuring the appearance of the runtime module (see Ch. 2.3.5).

This button opens the dialog for configuration of the text elements of the runtime module for various languages (see Ch. 2.3.6).

This button opens the dialog for configuration of the interface tags between the runtime module and WinCC (see Ch. 2.3.7).

This button opens the dialog for configuration of the printing of recipes in the runtime module (see Ch. 2.3.8).

This button starts a database update for the recipe database (see Ch. 2.4.1).

This button opens a dialog for editing the recipe database (see Ch. 2.4.2).

- Template comment: This input field is used for a description of the recipe template. The maximum length of the description is 255 characters.

- Name of recipe database: This input field is used for defining the name of the recipe database. The name must be no longer than 255 characters and must conform to the Windows file name conventions.

- Data directory of recipe database: This input field is used for defining the directory for the recipe database. The path must be no longer than 255 characters and must conform to the Windows directory name conventions. The path can be configured by means of a browse button.

- Master database: This input field is used for defining the complete path of the master database that can be accessed in the runtime module with a client-server configuration by multiple recipe handling systems. The path must be no longer than 255 characters and must conform to the Windows directory name conventions. The path can be configured by means of a browse button.

- Enable manual input of an article name: This check box specifies whether the manual input of an article name is allowed when saving or creating a new article, or whether the article name can be selected only via the master article of the master database.
2.3.3 Configuration of Tags

The tags associated with a recipe are defined in a tags selection dialog. The selected recipe tags are displayed in a list with the name, data type, AS address and a description of the tag in German, English or a third language.

![Recipe tag selection](image1)

**Fig. 2.3: Recipe tag selection**

The tags dialog supports the following functions:

![Recipe tag selection, right side](image2)

**Fig. 2.4: Recipe tag selection, right side**

The tags dialog supports the following functions:
Add: When adding a tag, the Recipe Designer directly accesses all tags contained in the WinCC project and supports simple and multiple selection as well as search masks for selecting tags. It is also possible to define new tags to be added to the recipe, and the new tags are automatically implemented in the WinCC project.

New tags are added to the end of the list. The function is started by pressing the [Add] button.

Remove: This function removes tags from a list of recipe tags. This function can also be started via the hotkey [Del] or the context menu [Right].

Move: The order of the tags in the list can be changed by moving single tags upward or downward. This function can also be started with the [Up] and [Down] keys and the hotkeys [Num+, Num-].

Bit selection: For 8, 16 and 32-bit tags without a prefix, a bit number can be selected in the “Bit” list column, so that the tag is treated as a binary partial tag of the recipe.

Note that all 8, 16 and 32-bit tags are saved in the recipe database of the runtime module, even if only individual bits appear in the recipe.

Edit: For each recipe tag, a description in German, English or any other third language can be edited (Fig.2.3). One can switch between languages via buttons. To edit a description field, select the field by double-clicking it. When editing in a foreign language, the corresponding reference text is suggested in the base language and can then be accepted.

At the right of the list of recipe tags (Fig. 2.4) there are additional input fields for:

- Unit: Text for the physical unit of a numeric tag.

- Status TRUE: Text describing the “logical 1 status” of a binary tag or a single-bit tag.

- Status FALSE: Text describing the “logical 0 status” of a binary tag or a single-bit tag.

Input in these fields are of significance only for printing the recipe with the runtime module of the recipe handling system. The values of numeric tags that have a physical unit here are printed with this unit. The values of binary tags that are configured with a TRUE-FALSE status here are printed with the respective status text. Otherwise the value is printed as “0” or “1”.

Import/Export: The import/export function enables the import and export of tag descriptions from/to text files in CSV format. This makes it possible to edit and import translations conveniently in external applications such as Excel. These functions are started with the [Text Import] and [Text Export] buttons.
In the tags dialog, various functions can also be started with the right mouse button via a context menu for the selected tag:

<table>
<thead>
<tr>
<th>Cut</th>
<th>Copy</th>
<th>Paste</th>
<th>Paste as bit tags</th>
<th>Delete</th>
<th>Get tag from WinCC</th>
</tr>
</thead>
</table>

Fig. 2.5: Context menu for tag selection

- **Cut:** Cuts tag to the clipboard for later insertion at another location.
- **Copy:** Copies a tag to the clipboard for later insertion at another location as a single bit tag, for example.
- **Paste:** Inserts a tag that was placed in the clipboard with the cut or copy function. The tag is inserted directly before the selected position.
- **Paste as bit tags:** Inserts a tag from the clipboard as an 8, 16 or 32 single bit tag. This is only possible with 8, 16 and 32 bit tags without a prefix.
- **Delete:** Deletes a tag. The tag is not placed in the clipboard, but is instead deleted.
- **Insert from WinCC:** Inserts one or more tags from the selection dialog of all tags contained in the WinCC project; supports single and multiple selection as well as search masks for selecting WinCC tags. The tags are inserted here directly before the selected position, as opposed to the [Add] button.

**Diagnostics:** This button provides a diagnostic function for testing the connection of the recipe tags to the automation system. This function is described in detail in Chapter 2.3.3 Connection Diagnostics.
2.3.3.1 Tags with „Validity-Check“

For special applications of a recipe management it is provided to force a so called “validity-check” for special recipe tags, before loading a recipe to the AS.

This is explained as follows:
A recipe shall be loaded to the process only if the values of special tags of the recipe to be loaded, match with the values of the tag in the WinCC process. In case the values are matching, the recipe can be loaded to the AS as usual. Otherwise an error message “Currently no permission to load” occurs.

This validity-check will be carried out for all recipe tags with the prefix “!_“ ( e.g. “!_varname“ ). There can be defined an binary interface tag to suppress the validity-check. This interface tag must have the same name as the interface tag „release for loading recipes“ (see chap. 2.3.8 Interface Tags), but with the prefix “!_“. If this interface tag does not exist, the validity-check will not be carried out. If the interface tag has the value “false”, the validity-check will be carried out for all “!_” - tags”. If the value of the interface tag is “true”, the recipe will be loaded without validity-check.

If a recipe is loaded to the WinCC-process from the offline-editor via the button "download" and the validity-check has failed, the concerning tags will be highlighted with red background color.
2.3.4 Conformity Check

The conformity check for the recipe tags is performed when the tags dialog is started. Tags already existing in the recipe are tested to determine whether all properties of the tags still conform to the tags in the WinCC project. If the AS address or the data type of a tag is changed in WinCC, for example, this is detected and the following dialog appears:

![Conformity Check: Address different](image)

This dialog can be closed with the following acknowledgments.

- **Accept**: The change made to the tag in WinCC is accepted in the tag configuration in the recipe template.
- **Accept all**: All other deviations in the tags in WinCC are accepted in the tag configuration of the recipe template.
- **Skip**: The change made to the tag in WinCC is ignored and is **not** accepted in the tag configuration of the recipe template.
- **Skip all**: All other deviations in the tags in WinCC are ignored and are **not** accepted in the tag configuration of the recipe template.

After the conformity check has been carried out, the tag dialog is started. If not all tags have been adapted by means of the conformity test, or recipe tags exist that are no longer present in WinCC, these tags are displayed with a check mark in the tags dialog.

**Important!**

As long as there are discrepancies with the WinCC project, this recipe template will enable no data transfer functions in the runtime module.
2.3.5 Connection Diagnostics

The connection diagnostics tests the connection between the recipe tags and the automation system (AS). The background of this test function is as follows:

If a recipe is saved while the runtime module of the recipe handling system is running, all tags associated with the recipe are requested by the data manager of WinCC runtime in order to maintain the current tag values. The data manager of WinCC requests the data via the connection channel from the AS (e.g. via MPI). But if tags in the address configured in WinCC are not contained in the AS, the data manager reports to the recipe handling system that the data were not successfully requested by the AS, with the result that the recipe cannot be saved. Such connection problems also exist, for example, if the value of an AS address could not be processed by the conversion to WinCC format. This is the case, if for example, a conversion to S5 time format is used and the value of the AS address does not contain a valid BCD time format value.

Since the search for such connection problems can be very tedious and time-consuming, this is simplified significantly by the diagnostic function.

Fig. 2.7: Connection diagnostics dialog window

The connection diagnostics dialog window contains the following functions:

- **Database:** This field displays the database path of the current template. This is important, because the connection diagnostics is based on the recipe structure of an already existing recipe database. The tags in the template can deviate from those of the database if changes in the template were not implemented in the database by means of a database update. This also shows that the diagnosis is necessary only on the WinCC target system, where all databases and their paths or even their remote paths of a client-server system are present. The diagnostic function therefore supports the implementation of the recipe handling system.
Tag list: The tag list "Tags with connection failure to AS" displays the result of the diagnosis after conducting the diagnostic run. All tags for which connection problems were detected are listed here.

Status: This display field shows the status during a diagnostic run and a success or error message after its completion.

"Start" button: This button starts the diagnostic run.

"Close" button: This button closes the dialog window and returns to the tag dialog. After closing the window the problematic tags in the tag list of the tags dialog are marked with a special icon:

![Tag_XY](image)

Requirements for a diagnostic run:

- The WinCC project must be in runtime mode.

- The runtime database for the recipe handling system must exist. The database is automatically created when the runtime window of the recipe handling system opens the first time. The recipe structure in this database is automatically created from the corresponding template.

- The recipe structure in the recipe database should correspond to the tags of the template. This is only the case if the recipe database has been newly created or if a database update is conducted after making changes to the template.
2.3.6 Runtime Layout

This dialog specifies which functions should be available in the main menu of the runtime module. Here one can also define the appearance of the dialogs and selection lists in WinCC Runtime.

2.3.6.1 "Runtime Window" Tab

![Runtime Layout Dialogue Window]

Fig. 2.8: "Runtime Window" tab in the "Runtime Layout" dialog window

- **Functions in runtime:** This is used to configure which buttons should be shown in the runtime window. Available functions include Load, Save, Delete, Print, New, Assign and Offline Editor.
  
  For each button function a permission level of the WinCC User Administrator can be set. The function is accessible in runtime mode, only if a user having the appropriate permission level is currently logged in.

- **Suppress runtime window and show only offline editor window:** If this check box is selected, the runtime window is not shown while the module is running, but only the offline editor window. This setting is desirable if all operations such as loading, saving etc. are performed in the offline editor window.

- **Runtime window X position** This specifies the X position of the upper left corner of the runtime window.

- **Runtime window Y position** This specifies the Y position of the upper left corner of the runtime window.
2.3.6.2 "Editor Window" Tab

Functions in offline editor: This is used to configure which buttons should be shown in the offline window. The available functions include New, Load, Save, Delete, Print, Assign, Download to SPS, Upload from SPS. For each button function a permission level of the WinCC User Administrator can be set. The function is accessible in runtime mode, only if a user having the appropriate permission level is currently logged in.

Fix editor window: If this check box is selected, the offline editor window is fixed during runtime, i.e. the window cannot be moved and always remains in the foreground.

X position offline editor window: This specifies the X position of the upper left corner of the offline editor window during runtime.

Y position offline editor window: This specifies the Y position of the upper left corner of the offline editor window during runtime.

Width of offline editor window: This specifies the width of the offline editor window during runtime.

Height of offline editor window: This specifies the height of the offline editor window during runtime.

Font size for offline editor table: This specifies the font size in pixels of the offline editor table during runtime.
2.3.6.3 "Choice Lists" Tab

![Fig. 2.10: "Choice Lists" tab in the "Runtime Layout" dialog window](image)

**Selection list for dialogs Load, Save, Delete, Print:**
This is used to specify the column widths of the selection lists of the dialog window for loading, saving, deleting and printing. The column widths can be changed by dragging the column headers. The "Default" button sets the default values for the column widths.

**Selection list of the dialog Assign:**
This is used to configure the column widths of the selection list of the Assign dialog window. The column widths can be changed by dragging the column headers. The "Default" button sets the default value for the column widths.
2.3.7 Runtime Text Elements

All text elements occurring in the selection menus and dialog windows of the runtime module in German, English and another foreign language are configured in a text table.

<table>
<thead>
<tr>
<th>Text element</th>
<th>German</th>
<th>English</th>
<th>Foreign</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title of runtime window</td>
<td>Rezeptur-Verwaltung</td>
<td>Recipe-Management</td>
<td></td>
</tr>
<tr>
<td>Title of dialog &quot;Load&quot;</td>
<td>Artikel laden</td>
<td>Load product</td>
<td></td>
</tr>
<tr>
<td>Title of dialog &quot;Save&quot;</td>
<td>Artikel speichern</td>
<td>Save product</td>
<td></td>
</tr>
<tr>
<td>Title of dialog &quot;Delete&quot;</td>
<td>Artikel löschen</td>
<td>Delete product</td>
<td></td>
</tr>
<tr>
<td>Title of dialog &quot;Print&quot;</td>
<td>Artikel drucken</td>
<td>Print product</td>
<td></td>
</tr>
<tr>
<td>Title of dialog &quot;New product&quot;</td>
<td>Neuer Artikel</td>
<td>New product</td>
<td></td>
</tr>
<tr>
<td>Title of dialog &quot;Assignment&quot;</td>
<td>Zuordnung editieren</td>
<td>Edit assignment</td>
<td></td>
</tr>
<tr>
<td>Title of dialog &quot;Offline editor&quot;</td>
<td>Offline Editor</td>
<td>Offline Editor</td>
<td></td>
</tr>
<tr>
<td>Title of dialog &quot;Change data&quot;</td>
<td>Daten ändern</td>
<td>Modify Data</td>
<td></td>
</tr>
<tr>
<td>Title of dialog &quot;Select product&quot;</td>
<td>Artikel auswählen</td>
<td>Select product</td>
<td></td>
</tr>
<tr>
<td>Title of dialog &quot;Save as&quot;</td>
<td>Speichern unter...</td>
<td>Save as...</td>
<td></td>
</tr>
<tr>
<td>Title of dialog &quot;Load to Editor&quot;</td>
<td>Laden in Editor</td>
<td>Load editor</td>
<td></td>
</tr>
<tr>
<td>Button OK</td>
<td>OK</td>
<td>OK</td>
<td></td>
</tr>
<tr>
<td>Button Cancel</td>
<td>Abbrechen</td>
<td>Cancel</td>
<td></td>
</tr>
<tr>
<td>Column header 1 list</td>
<td>Artikel-Nr.</td>
<td>Product no.</td>
<td></td>
</tr>
<tr>
<td>Column header 2 list</td>
<td>Kommentar</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>Column header 3 list</td>
<td>Rezeptur</td>
<td>Recipe</td>
<td></td>
</tr>
<tr>
<td>Column header 4 list</td>
<td>Datum</td>
<td>Date</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 2.11: "Runtime Text Elements" dialog window

The "Runtime Text Elements" dialog supports the following functions:

- **Edit:** The text elements of the runtime module can be edited directly for German, English and a third language.

- **Default:** All German and English texts can be reset to default settings by pressing this button.

- **Import/Export:** The import/export function enables the import and export of the runtime text elements from/to text files in CSV file format. This makes it possible to conveniently edit and import translations in external applications such as Excel.
2.3.8 Interfaces

The "Interface" dialog window enables the definition of interface tags by which the runtime module of the recipe handling system should communicate with WinCC runtime or with the AS. Tags from the WinCC project are selected as interface tags. If a selection field remains empty, the interface will not be used by the runtime module.

Fig. 2.12: "Interface Tags" dialog

The following interface tags are available:

- **Current article number:**
  This interface tag returns the article number of the article (recipe) loaded by the runtime module for display in WinCC Runtime. The tag type (number or text string) can be selected here. In addition, the maximum length of the article number can be specified.

- **Mask:**
  This input field is to define a masked edit of the article number in the runtime module. That masked edit provides the user of the runtime module to enter an article number in the correct form. The mask can contain any literal, non-mask character, which will be displayed. Special mask characters are used to denote input spots in the edit, based on the following key:

  - #  Numeric character (0-9)
  - A  Alpha-numeric character (0-9 and a-Z)
  - &  Any ASCII character
  - ?  tAlphabetic character (a-Z)
  - U  Accepts a-Z, forces to A-Z (uppercase)
  - L  tAccepts a-Z, forces to a-z (lowercase)
  - \  Escape character
Here is a possible example of a mask:

Mask:  #UU#  ### - #UU## - #UU#  Mask prompt character:  _

Then in runtime the edit field looks like this:

Empty edit field for an article number:

```
________
```

Entered formatted article number:

```
GE57 416 - 2<K02 - 0A00
```

Masks consist of 'mask characters' and literals. Literals are characters that appear unchanged in the mask, while mask characters specify a spot in the mask edit that accepts certain characters. At runtime the mask characters are replaced with the prompt character. Use the \' \' Escape character if you would like one of the special mask characters to be interpreted as a literal character.

 thận article comment:
This interface tag returns the article comment of the article (recipe) loaded by the runtime module for display in WinCC Runtime.

Default article number of the auto-load function:
A default article number can be entered here and activated by means of a control box. The default article number is used with the runtime auto-load function if the desired article number of the auto-load function does not exist. The default article number can therefore be regarded as "non-recipe". In case of an auto-load failure, a message is normally issued in the runtime module and instead of the "Finished loading" signal an "Auto load error" signal is sent via the interface tag. If a default article was successfully loaded, however, no message is issued, but only an "Auto load error" instead.

Release:
WinCC runtime or the AS can issue the release for loading a recipe via this interface tag.

Acknowledgment of active RH runtime window:
If the runtime window of the recipe handling system is active, this status can be reported to WinCC runtime or the AS via this interface tag.

Detection of a batch start:
This interface tag is used for sending a signal to WinCC Runtime or the AS after loading of a new recipe, for example to signal the beginning of a new batch.

Detection of a batch end:
This interface tag is used for sending a signal to WinCC Runtime or the AS before loading of a new recipe, for example to signal the end of a batch.

Loading of recipe completed:
This interface tag is used to send a signal to WinCC Runtime or the AS to signal that the loading of a recipe is completed.

Automatic loading failed:
This interface tag is used to send a signal to WinCC Runtime or the AS to signal an error in the auto-load function of the R&W recipe handling system.
Signal propagation times:
These input fields make it possible to specify the runtime behavior of the interface signals in a runtime diagram. The tag duration of the runtimes of the interface signals ensure that the signals are active long enough in order to be detected for further processing of the signals in WinCC (e.g. by alarm logging or global script).
2.3.9 Report Layout

The "Report Layout" dialog window is used to specify the layout for printing of a recipe of the recipe handling system runtime module.

![Report Layout dialog window](image)

The report layout features the following options:

- **Margins:**
  The widths of the "left", "right", "top" and "bottom" margins of the printout can be entered here in millimeters or changed by means of the small up-down keys between 0 – 50 mm (see Fig. 2.13).

- **Tabs:**
  "Tabs" refers to the three tab positions of the text elements in the side header of the printout (see Fig. 2.13).

- **Column widths:**
  The widths of column 1 (tag description) and column 2 (tag value) of the tags table can be set between 0 – 200 mm. The width of the third column (tag name) automatically adjusts to the right margin. If the column for the tag name is hidden, then column 2 (tag value) automatically adjusts to the right margin (see Fig. 2.13).
Fig. 2.14: Dimensions in print view

**Preview:** In a small preview window one can see how the configured dimensions change the appearance of the printout.

**Font:** The font type and size for the texts in the tags table of the printout can be configured here.

**Show tag names:** In the third column of the tags table the WinCC tag name of the respective recipe tag can be printed. Since this is not always desirable, this option can be selected or deselected via the check box.

**Sub-headers:** In the right side of the dialog window there is a list of the current recipe tags. The printout of the recipe tags can be sub-divided by subject by inserting sub-headers here (see Fig. 2.13) in order to give the printout a clearer structure. The sub-headers can be edited by double-clicking them with the right mouse button. Buttons make it possible to switch between languages. When editing in a foreign language the corresponding reference text is displayed in the base language and can then be accepted.

In addition, the sub-headers are used for the editor table in the offline editor.
2.4 Database Functions

2.4.1 Database Update

This function is used for maintaining runtime databases of the R&W Recipe Designer, which is necessary in the following event:

A WinCC application with R&W Recipe Designer is already in operation and numerous recipes have been archived in the recipe database. New process parameters are now to be entered into the recipe handling system. For this purpose, the recipe template has been adapted with the Recipe Designer.

The database update function can be used to adapt all recipes in the recipe database to the new range of tags. All recipe tags that were removed from the template are then deleted from all recipe data sets in the database! New recipe tags in the template are added to all recipe data sets in the database with the default value zero!

A database update is allowed only if no runtime module of the WinCC project is active. Therefore, the runtime module must first be deactivated before a database update can be started.

A database update is desired or necessary...

...if tags have been added to or deleted from the template.
...if tags in the template have been changed (address, data type)
...if text elements such as tag comment, status text or physical unit have been changed.
...if sub-headers in the report layout have been added, deleted or changed.

The recipe database that is updated using the database update function must exist in order for the update to be carried out. The database is newly created by the runtime module the first time it is started.

Fig. 2.15: "Database Update" dialog window
2.4.2 Edit Database

This function enables editing of recipe data in the recipe database. It can be used, for example, to initialize values for recipe tags that were added by means of a database update. Editing of a tag value in the database also changes the tag in all recipes in the database!

Fig. 2.16: "Edit Database" dialog window

![Edit Database dialog window](image)
3. Engineering in the WinCC Project

The functions of the runtime module of the recipe handling system can be integrated easily in WinCC Project. There are three ways to do this:

- Dialog-based control in a runtime window
- Autoload function for loading recipes in the background
- Autosave function for saving recipes in the background
- Start of Recipe Viewer to list all Recipes of a database
- Quit the Runtime window by an external call

To handle the recipe management, the WinCC Global Script provides standard functions. This functions are implemented in WinCC with the installation of Recipe Designer 3.0 CS. After the installation it may be necessary to operate the function “Regenerate Header” in the menu entry “Tools” of the Global Script Editor, before the Recipe Designer standard functions are compilable in the script coding!

3.1 Opening the Runtime Window

The runtime module of the recipe handling system is started via a command line in the C script of WinCC. For example, the start can be configured on a button object in the WinCC image at the event “mouse click” in a C script:

```c
#include "apdefap.h"
void OnClick(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName)
{
    RD_StartRuntime( "Anlage_1" );  //Start Runtime-Window
}
```

Syntax of the standard function:

```c
void RD_StartRuntime( char* lpszTemplateName )

char* lpszTemplateName       //Pointer to name of template without suffix
```
3.2 Closing the Runtime Window

Normally, the runtime window is closed automatically or manually by the operator after executing an operation (e.g. loading of a recipe). The runtime window remains completely unaffected upon a display change in WinCC. The function "RecipeRTQuit" is available for closing the runtime window with an event from WinCC (e.g. upon display change). This function is an EXE application that sends a message to a specific runtime window, causing the window to close. In this case, any data transfer still in progress will be completed before the window is closed.

The function is started via a command line in the C script. For example, the start can be configured on a button object in the WinCC image at the event “mouse click” in a C script.

```c
#include "apdefap.h"
void OnClick(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName) {
    RD_QuitRuntime( "Anlage_1" ); // Quit Runtime-Window
}
```

Syntax of the standard function:

```c
void RD_QuitRuntime( char* lpszTemplateName )
char* lpszTemplateName // Pointer to name of template without suffix
```

3.3 Starting the AutoLoad Function

The auto-load function of the runtime module of the recipe handling system is started via a command line in the C script of WinCC. For example, the start can take place in a global action of the global script in the WinCC project. The action can be triggered by a process tag, for example. This is an easy way to implement auto-loading by the automation system.

```c
#include "apdefap.h"
int gscAction( void ) {
    RD_AutoLoad( "Anlage_1", "Produkt 12345" ); // Autoload
}
```

Syntax of the standard function:

```c
void RD_AutoLoad( char* lpszTemplateName, char* lpszRecipe )
char* lpszTemplateName // Pointer to name of template without suffix
char* lpszRecipe // Pointer to name of product to be loaded
```
3.4 Starting the AutoSave Function

The auto-save function of the runtime module of the recipe handling system is started via a command line in the C script of WinCC. For example, the start can take place in a global action of the global script in the WinCC project. The action can be triggered by a process tag, making it easy to implement the auto-save function by the automation system.

The following example illustrates auto-saving of the current article. The current article is taken from the interface tag and placed in the command line. Therefore, the auto-save function always saves the article in the recipe database that is specified by the AS or by WinCC via a string tag (process tag).

```c
#include "apdefap.h"

int gscAction( void )
{
    char szProduct[255];
    //Get actual product from a WinCC tag
    strcpy( szProduct, GetTagChar("Var_Artikelname") );

    RD_AutoSave( "Anlage_1", szProduct );  //Autosave
}
```

Syntax of the standard function:

```c
void RD_AutoSave( char* lpszTemplateName, char* lpszRecipe )
char* lpszTemplateName     //Pointer to name of Template without suffix
char* lpszRecipe            //Pointer to name of product to be saved
```

3.5 Starting the Recipe Viewer

The Recipe Viewer provides the viewing of all recipes of the database in one list. For example, the start can be configured on a button object in the WinCC image at the event "mouse click" in a C script:

```c
#include "apdefap.h"

void OnClick(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName)
{
    RD_RecipeViewer( char* lpszTemplateName );  //Start Recipe Viewer
}
```

Syntax of the standard function:

```c
void RD_RecipeViewer( char* lpszTemplateName )
char* lpszTemplateName    //Pointer to name of template without suffix
```
4. Installation

4.1 System Requirements

The R&W Recipe Designer is designed to run with SIMATIC WinCC V4.02 through V6.0 SP4. The operating system MS Windows NT 4.0, Windows 2000 SP3 or Windows XP SP2 is required.

4.2 Installation

1.) Before you begin the installation of the R&W Recipe Designer, make sure that all other applications, especially SIMATIC WinCC, have been closed.

2.) Place the Recipe Designer CD-ROM in the CD-ROM drive. Setup will begin automatically after a few seconds. Once the installation has started, the CD must not be removed from the drive.

3.) Starting Setup.exe manually (alternative)
   a) In the start menu, select "Settings", "Control panel" and then "Software". In the "Software properties" window select the "Install/Uninstall" tab and press the "Install" button. In the following window you can select the drive and the path to the setup program.
   b) In the start menu, select "Execute...". In the following window you can select the drive and the path to the setup program and start the program.
      Note: In order to properly install the R&W Recipe Designer with Windows NT, the user must have write permission for the registry (administrator permission).

4.) Setup will guide you through the entire installation of the R&W Recipe Designer.

5.) During setup, you can choose between standard and user-defined installation.
   a) In the standard installation, the runtime module "R&W Recipe Designer RT" is installed. This installation type is intended for computers for which only one runtime license was purchased.

   b) In the user-defined installation, the components "Recipe Designer CS" (configuration module) and "Recipe Designer RT" (runtime module) can be selected separately for installation. The installation and use of the respective module requires the purchase of the corresponding software license.

      Unauthorized use of the software without a software license is punishable by law!

6.) During the installation, system files of WinCC are automatically modified.
4.3 Adding Components / Uninstalling

Adding Components:

To add components, the R&W Recipe Designer must first be completely uninstalled and then reinstalled with the desired components.

Uninstalling:

1.) In the start menu, select "Settings", "Control panel".
2.) In the "Control panel" window, select "Software".
3.) In the "Software properties" window, select the "Install/Uninstall" tab.
4.) In the list of installed software, select "R&W Recipe Designer V4.0" and click the "Add/Remove" button.
5.) During the uninstall routine, the system files are automatically purged by WinCC.
## 5. Overview

### R&W Recipe Designer 4.0 SP1 Hotfix 2 Overview

#### Features:

- Full integration in WinCC (Recipe Designer is started from the editor tree in the WinCC Explorer)

- Runtime module for WinCC for starting the runtime function from all possible object events in WinCC displays or other events

- Freely definable interfaces to the WinCC process via internal or process tags

- Data storage in separate SQL databases, as all databases of the WinCC system (up to WinCC 5.0 SP1: Sybase SQL-Anywhere 5.0, WinCC 5.0 SP2 – 5.1 SP1: Sybase Adaptive Server Anywhere 7.0, from WinCC 6.0: Microsoft SQL Server 2000)

- Unlimited\(^1\) number of tags per recipe in the database

- Unlimited\(^2\) number of data sets (recipes) in the database

- Standardized ODBC/SQL interface to the recipe data for further processing e.g. with Microsoft Excel® or Microsoft Access®

- Network-capable database concept

- Support of WinCC client-server architecture (configuration of master and client recipe databases for common data)

- Support of WinCC language switching

- Easy-to-operate for the developer – saves time and engineering costs

- Easy-to-use for the operator

- Can be used with WinCC V4.02, V5.0 SP1, V5.0 SP2, V5.1, V5.1 SP1, V6.0, V6.0 SP1, V6.0 SP2, V6.0 SP3, V6.0 SP4

\(^1\) depending on computer performance, max. 64,000, tested 10,000

\(^2\) depending on hard drive capacity and Sybase SQL driver limitations

#### Supplier:

R&W Industrieautomation GmbH
Graf-Heinrich-Str. 20
D-57627 Hachenburg
Tel.: 02662 941434
Fax: 02662 941441
E-mail: info@r-u-w.de
Web: www.r-u-w.de