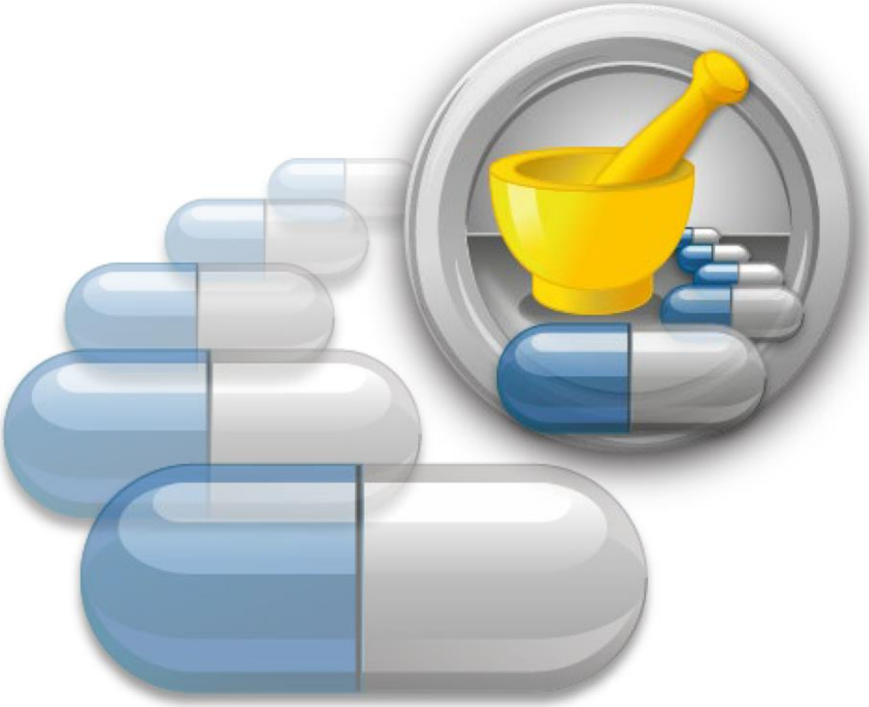
FT PharmaSuite®



EBR PHASES

RELEASE 10.02.00

USER MANUAL

PUBLICATION PSEB-UM005C-EN-E-SEPTEMBER-2021
Supersedes publication PSEB-UM005B-EN-E

Rockwell Automation

Contact Rockwell

See contact information provided in your maintenance contract.

Copyright Notice

Trademark Notices

Other Trademarks

Warranty

This product is warranted in accordance with the product license. The product's performance may be affected by system configuration, the application being performed, operator control, maintenance, and other related factors. Rockwell Automation is not responsible for these intervening factors. The instructions in this document do not cover all the details or variations in the equipment, procedure, or process described, nor do they provide directions for meeting every possible contingency during installation, operation, or maintenance. This product's implementation may vary among users.

This document is current as of the time of release of the product; however, the accompanying software may have changed since the release. Rockwell Automation, Inc. reserves the right to change any information contained in this document or the software at any time without prior notice. It is your responsibility to obtain the most current information available from Rockwell when installing or using this product.

Contents

Chapter 1 **EBR Phases** 1

Typographical Conventions 2

Chapter 2 **Get Choice Value** 3

Execution 3

Phase Design 4

Process Parameters 4

Output Variables 13

Chapter 3 **Get Process Value** 17

Execution 17

Phase Design 18

Process Parameters 19

Output Variables 30

Chapter 4 **Get Text Value** 33

Execution 33

Phase Design 34

Process Parameters 34

Output Variables 42

Chapter 5 **Show Document** 45

Execution 45

Phase Design 46

Process Parameters 46

Output Variables 53

Chapter 6 Show Instruction Text 55
Execution 55
Phase Design 55
Process Parameters 56
Output Variables........ 62

Chapter 7 Show URL........ 65
Execution 65
Phase Design 66
Process Parameters 66
Output Variables........ 73

Chapter 8 Upload Image 75
Execution 75
Phase Design 78
Process Parameters 78
Output Variables........ 86

Chapter 9 Upload PDF........ 89
Execution 89
Phase Design 93
Process Parameters 93
Output Variables........ 102

Chapter 10 Create Workflow........ 105
Execution 105
Phase Design 106
Process Parameters 107
Output Variables........ 118

Chapter 11 Write Context Data........ 123
Execution 123
Phase Design 124
Process Parameters 124

Output Variables 135

Chapter 12 Send User Notification 139

Execution 139

Phase Design 140

Process Parameters 140

Output Variables 150

Index 153

Figures

Figure 1: Get choice value during execution 3

Figure 2: Get choice value after phase completion 4

Figure 3: Get choice value in the Navigator 4

Figure 4: Post-completion exception of Get choice value.................................... 4

Figure 5: Option List editor.. 5

Figure 6: Get process value during execution 18

Figure 7: User-triggered exception of Get process value....................................18

Figure 8: Get process value after phase completion 18

Figure 9: Get process value in the Navigator 18

Figure 10: Post-completion exception of Get process value 18

Figure 11: Get text value during execution..33

Figure 12: User-triggered exception of Get text value 33

Figure 13: Get text value after phase completion...33

Figure 14: Get text value in the Navigator..34

Figure 15: Post-completion exception of Get text value 34

Figure 16: Show document during execution 46

Figure 17: Show document after phase completion 46

Figure 18: Show document in the Navigator 46

Figure 19: Show instruction text during execution 55

Figure 20: Show instruction text after phase completion 55

Figure 21: Show instruction text in the Navigator 55

Figure 22: Show URL during execution 65

Figure 23: Show URL after phase completion..65

Figure 24: Show URL in the Navigator 66

Figure 25: Upload image during execution - Automatic loading mode 76

Figure 26: Upload image during execution - Selection mode 76

Figure 27: User-triggered exception of Upload image 76

Figure 28: File selection dialog of Upload image 77
Figure 29: Upload image after phase completion..77
Figure 30: Upload image in the Navigator...77
Figure 31: Post-completion exception of Upload image 77
Figure 32: Upload PDF during execution - Automatic loading mode 90
Figure 33: Upload PDF during execution - Selection mode 90
Figure 34: User-triggered exception of Upload PDF 91
Figure 35: File selection dialog of Upload PDF 91
Figure 36: Upload PDF after phase completion 92
Figure 37: Upload PDF in the Navigator 92
Figure 38: Post-completion exception of Upload PDF.......................................92
Figure 39: Create workflow during execution 106
Figure 40: User-triggered exception of Create workflow................................. 106
Figure 41: Create workflow after phase completion 106
Figure 42: Create workflow in the Navigator 106
Figure 43: Master Workflow Selection editor 109
Figure 44: Work Center Selection editor 110
Figure 45: Station Selection editor... 111
Figure 46: Write context data during execution 123
Figure 47: User-triggered exception of Write context data 124
Figure 48: Write context data after phase completion 124
Figure 49: Write context data in the Navigator 124
Figure 50: Send user notification during execution - Timer notification................ 139
Figure 51: Send user notification during execution - Timestamp notification 139
Figure 52: Send user notification after phase completion   - Timer notification 140
Figure 53: Send user notification after phase completion - Timestamp notification .. 140
Figure 54: Send user notification in the Navigator - Timer notification................. 140
Figure 55: Send user notification in the Navigator - Timestamp notification 140

Chapter 1

EBR Phases

The EBR phases of PharmaSuite represent a collection of phases that can be used for various recipe or workflow purposes within the framework of PharmaSuite for Production Execution with EBR. They provide functions to retrieve or register processing data, such as values, documents, or images.

The following phases are available:

- Get Choice Value (page 3)
- Get Process Value (page 17)
- Get Text Value (page 33)
- Show Document (page 45)
- Show Instruction Text (page 55)
- Show URL (page 65)
- Upload Image (page 75)
- Upload PDF (page 89)
- Create Workflow (page 105)
- Write Context Data (page 123)
- Send User Notification (page 139)

This section contains important information about using the EBR phases in master recipes, workflows, or building blocks. Please read this section carefully, because it provides a solid background for all operations you may wish to perform with your system.

Typographical Conventions

This documentation uses typographical conventions to enhance the readability of the information it presents. The following kinds of formatting indicate specific information:

Bold typeface

Designates user interface texts, such as

- window and dialog titles
- menu functions
- panel, tab, and button names
- box labels
- object properties and their values (e.g., status).

Italic typeface

Designates technical background information, such as

- path, folder, and file names
- methods
- classes.

CAPITALS

Designate keyboard-related information, such as

- key names
- keyboard shortcuts.

`Monospaced typeface`

Designates code examples.

Chapter 2

Get Choice Value

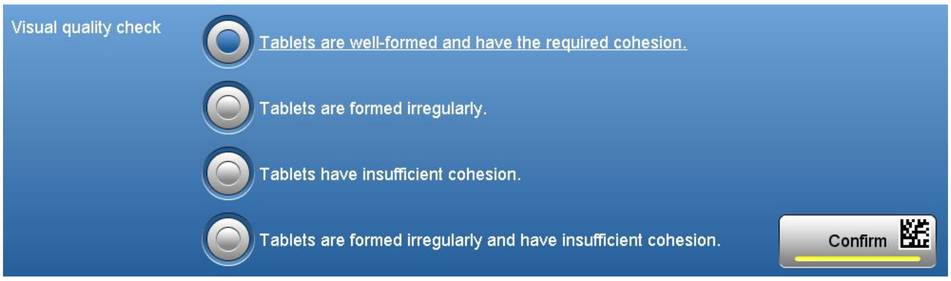
The **Get choice value** phase allows an operator to record a choice from a pre-defined list of options.

It can be used for processing requirements, such as:

- Recording of visual appearance during product test
  During the inspection of a product sample, the visual appearance of the sample can be selected from a pre-defined list (e.g. Transparent, Cloudy, Dark).
- Recording of production resources from a pre-defined list with a preset default option
  Operator documents which tool was used when entering the property tag.
- Recording of an operator decision
  Operator documents with **Yes** or **No** whether a certain precondition applies. The operator decision determines which of the two alternative subsequent process steps of a selection branch will become active.

Execution

The **Get choice value** phase records one option selected by an operator from a pre-defined list of options and can match it against an expected value. If configured so, it displays the default option by pre-selecting it and the expected option by underlining it. After phase completion, it provides a post-completion exception to correct the value selected during processing.
After completion the phase displays the recorded value, both in the Execution Window and the Navigator. Additionally, the Navigator provides access to the post-completion exception.



Visual quality check

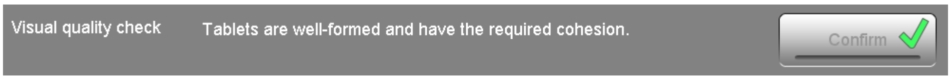
● Tablets are well-formed and have the required cohesion.

○ Tablets are formed irregularly.

○ Tablets have insufficient cohesion.

○ Tablets are formed irregularly and have insufficient cohesion.

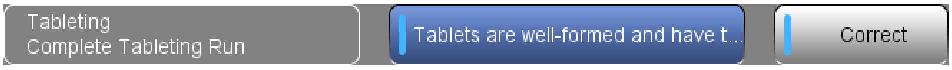
Figure 1: Get choice value during execution



Visual quality check

Tablets are well-formed and have the required cohesion.

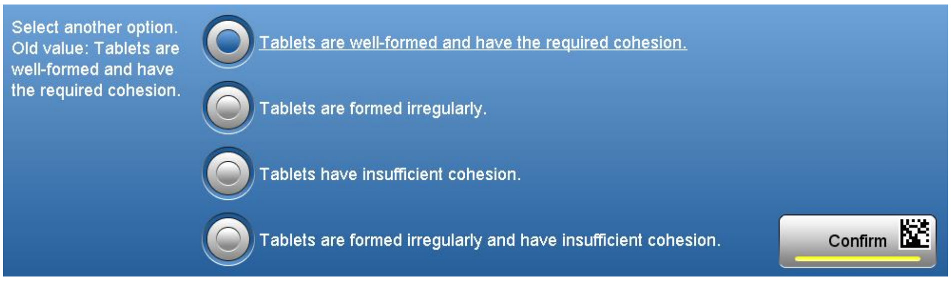
Figure 2: Get choice value after phase completion



Tableting
Complete Tableting Run

Tablets are well-formed and have t...

Figure 3: Get choice value in the Navigator



Select another option.
Old value: Tablets are well-formed and have the required cohesion.

- ● Tablets are well-formed and have the required cohesion.
- ○ Tablets are formed irregularly.
- ○ Tablets have insufficient cohesion.
- ○ Tablets are formed irregularly and have insufficient cohesion.

Figure 4: Post-completion exception of Get choice value

Phase Design

The characteristics of the **Get choice value** phase are defined via process parameters and their attributes.

Its user interface is designed in three columns. The first column is available for instruction texts. The second column displays the definable list of options. The rightmost column provides the **Confirm** button.

Exception handling during execution is controlled by a risk assessment classification and an exception message that are both defined by the recipe author in the exception's process parameter.

Process Parameters

The following process parameters are available to configure the phase's behavior during execution:

Instruction

Represents the instruction text that is visible on the preview, the active, and the completed view of the phase.

Attribute	Type	Comment
Column 1	HTML text	Instruction text to be displayed. Maximum length is 2000 characters (including HTML tags).
Column 2	HTML text	Not used.

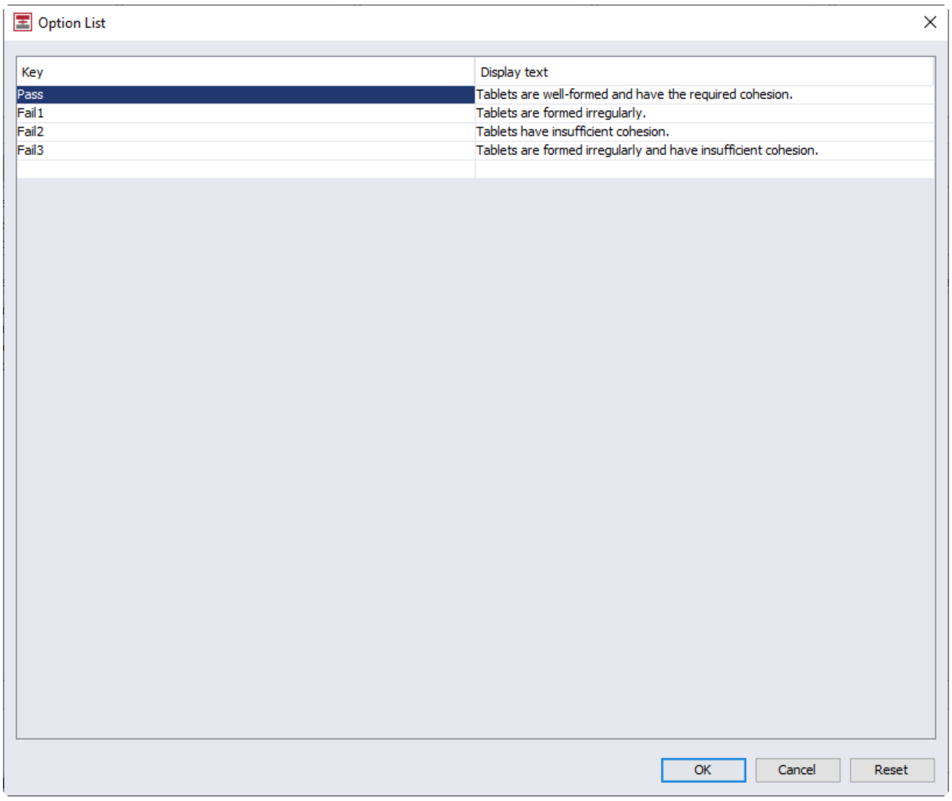
Attribute	Type	Comment
Column 3	HTML text	Not used.

List of options

Defines the options available for selection during execution.

Attribute	Type	Comment
Options	Text (structured)	Defines the available options as key/display text value pairs. Both keys and display texts are unique within a phase.

For entering the key/display text pairs the system provides an Option List editor.



Key	Display text
Pass	Tablets are well-formed and have the required cohesion.
Fail1	Tablets are formed irregularly.
Fail2	Tablets have insufficient cohesion.
Fail3	Tablets are formed irregularly and have insufficient cohesion.

Figure 5: Option List editor

Expected value configuration

Defines if the actual option selected during execution must be checked against an expected value.

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If so, ensure that the **Expected value key** attribute of the **Expected value definition** process parameter (page 6) is set.
Display	Flag	Controls if an expected value is displayed during execution. The expected option is underlined. Ensure that the **Expected value key** attribute of the **Expected value definition** process parameter (page 6) is set.
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: **None**, **Low**, **Low (mandatory comment)**, **Medium**, **Medium (mandatory comment)**, **High**, **High (mandatory comment)**. Default setting: **High**.
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 250 characters.

Expected value definition

Defines the key of the option required as expected value (page 6) if the respective check is enabled.

Attribute	Type	Comment
Expected value	String	Defines the expected value.
Default value	String	Defines the pre-selected item in the list of options.

One-click completion

Defines if the phase is confirmed and thus completed automatically when an operator selects an option during execution. If you have configured an expected value (page 6), the phase only completes automatically when the selected option matches the expected value.

Attribute	Type	Comment
Enabled	Flag	Controls if the phase is automatically completed when an option has been selected.

Correct value

Represents a post-completion exception that is accessible from the Navigator.
The exception allows an operator to correct the option selected while the phase was active.
It covers incidents when the operator has selected an incorrect option on account of a reading error, but has confirmed and completed the phase before detecting the error.

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: **None**, **Low**, **Low (mandatory comment)**, **Medium**, **Medium (mandatory comment)**, **High**, **High (mandatory comment)**. Default setting: **High**.
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 2000 characters.

Instruction links

In addition to the permanent process parameters that are always present, the **Get choice value** phase provides instruction links as optional process parameters, which you can insert if required.

You can add up to ten instruction link parameters.

TIP

Instruction links are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.

ADDING INSTRUCTION LINKS

1. Click the **Add parameter** button.
   The system opens an option list that holds all optional parameter types available for the phase.

2. Select the **Instruction Link** type.
   The system opens the **Add Instruction Link** dialog to define the instruction link's identifier.

3. Type an identifier and click the **OK** button.
   The system adds the instruction link parameter and the first link definition row to the list of parameters.
   Instruction link parameters are generally inserted below all other parameters. Where within the block of instruction link parameters the system adds a new link parameter depends on the current selection in the Parameter Panel:

   - If no parameter is selected, the system adds the new instruction link parameter as last parameter.
   - If an instruction link parameter is selected, the system adds the new instruction link parameter below the selected one.
   - If any other parameter is selected, the system adds the new instruction link parameter as first parameter of the instruction link parameter block.

   TIPS
   Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa.
   The identifier of the instruction link parameter is shown as **Identifier** of the link's instruction text parameter.

4. Specify the instruction text to be displayed and mark the link texts by enclosing them in curly brackets.

5. Specify the list of link definitions. Each row of the list defines one hyperlink. The button bar above the list provides the following functions:

   ⊞ adds a new row to the list. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.

🗑 deletes the currently selected rows.

⬆ moves the currently selected row one row up.

⬇ moves the currently selected row one row down.

Removing instruction links

1. In the list of parameters, select the instruction link parameter you wish to remove.
2. Click the **Remove parameter** button.
   The system asks you to confirm the action and then removes the instruction link parameter.

The following process parameters are available to configure the phase's behavior during execution:

Instruction text with links

Defines the text of the optional instruction link that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Attribute	Type	Comment
Instruction text	HTML text	Instruction text to be displayed. For any text enclosed in curly brackets you can define a hyperlink with the **Instruction link definition** process parameter (page 9). Example: Refer to {SOP1270} for guidance. Maximum length is 2000 characters (including HTML tags).

Instruction link definition

Defines all links to be available within the instruction text defined with the **Instruction text with links** process parameter (page 9). You can either access a file on the web, such as your intranet, by using the web access syntax or a file on your file system, by using the file access syntax.
Examples:

```
https://rockwellautomation.com
file:///c:/SOP/SOP1492.pdf
```

Attribute	Type	Comment
Link text	Text	Text to be used as link. For any text enclosed in curly brackets within the instruction text you can define a link with the **Link URL** attribute. Including the brackets in the link text is optional. Maximum length is 80 characters.
Link URL	Text	URL of the file to be displayed. The link opens the external application assigned to the file type by the operating system. Maximum length is 256 characters.

Instruction tables

In addition to the permanent process parameters that are always present, the **Get choice value** phase provides instruction tables as optional process parameters, which you can insert if required.

You can add up to ten instruction tables with up to 50 table rows.

TIP

Instruction tables are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.

ADDING INSTRUCTION TABLES

1. Click the **Add parameter** button.
   The system opens an option list that holds all optional parameter types available for the phase.
2. Select the **Instruction Table** type.
   The system opens the **Add Instruction Table** dialog to define the instruction table's identifier.
3. Type an identifier and click the **OK** button.
   The system adds the parameter definition and the first table row to the list of parameters.
   Instruction table parameters are generally inserted below all other parameters.

Where within the block of instruction table parameters the system adds a new table depends on the current selection in the Parameter Panel:

- If no parameter is selected, the system adds the new instruction table as last parameter.
- If an instruction table parameter is selected, the system adds the new instruction table below the selected one.
- If any other parameter is selected, the system adds the new instruction table as first parameter of the instruction table parameter block.

TIPS

Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa.

The identifier of the instruction table is shown as **Identifier** of the table's definition parameter. The identifiers of the individual table rows (**Row-1**, **Row-2**, etc.) are system-defined and not editable.

4. Specify the overall appearance of the table:
   - Select the number of columns to define the layout.
   - Set the width of the first column. If you do not set it to narrow, all columns have equal widths. If you set the first column to narrow, the remaining columns will have equal widths.
   - Define if your table needs to have borders. You can either show all borders of the table and its cells or none.
5. Specify the table rows and their content. The button bar above the rows table provides the following functions:

   ⊞ adds a new row to the table. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.

   🗑 deletes the currently selected rows.

   ⬆ moves the currently selected row one row up.

   ⬇ moves the currently selected row one row down.

TIP

Please note that the system always retains the consecutive numbering of the rows. If you reorder the table rows or delete rows, the row identifiers are updated accordingly.

REMOVING INSTRUCTION TABLES

1. In the list of parameters, select the instruction table parameter you wish to remove.
2. Click the **Remove parameter** button.
   The system asks you to confirm the action and then removes the instruction table with its definition and rows.

The following process parameters are available to configure the phase's behavior during execution:

Instruction table definition

Defines the appearance of the optional instruction table that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Attribute	Type	Comment
Table layout	Choice list	Defines the layout of the instruction table holding the instruction texts. Available settings: **1 column**, **2 columns**, **3 columns**, **4 columns**, **5 columns**. Default setting: **1 column**.
First column narrow	Boolean	Defines if the first column of the table shall be narrow.
Show all borders	Boolean	Defines if the borders of the table shall be visible.

Instruction table text

Specifies the instruction texts to be displayed in the individual cells of the instruction table.

Attribute	**Type**	**Comment**
Column 1	HTML text	Instruction text to be displayed in a column. **Restriction:** Maximum length is 2000 characters (including HTML tags).
Column 2	HTML text	
Column 3	HTML text	
Column 4	HTML text	
Column 5	HTML text	

Output Variables

Instead of specifying a fixed value to be displayed or used during execution, you can also use an expression created in the Expression editor to draw the output of another phase or operation or the calculated result of several outputs as value into a parameter attribute. When you reference outputs in this manner you need to be aware of the following restrictions:

- Only when a component has been processed does it provide an output that can be fed into another component as attribute value. For this reason, you can never reference an output of a component that is a strict successor of the component in which you try to use the output.
- Branches and loops, however, require special notice in this context, since they are only potentially passed through and/or completed during processing, so their outputs are not reliably available. Thus, you can reference any such potentially available outputs, but need to be aware of the fact that the provided value may be **Undefined** so that the component into which you are feeding the output must be able to deal with such an **Undefined** input value.

The **Get choice value** phase provides the following output variables:

Option key

- Data type: String, used for displaying a pre-defined sequence of characters, such as "**NEXT_ITEM**" or "**COMPLETED**".
- Usage: The output variable provides the key value of the selected option.

Option text

- Data type: String, used for displaying a pre-defined sequence of characters, such as "**Passed without issues**" or "**Passed with issues**".
- Usage: The output variable provides the display text of the selected option.

Identifier

- Data type: String, used for displaying a pre-defined sequence of characters, such as "**Read Instruction**".
- Usage: The output variable provides the identifier of the phase.

Instance count

- Data type: Long, used for integral numbers:
  12345
- Usage: The output variable provides the count of the number of instances the phase has been processed, for example in a loop. The count is also increased when the phase is skipped from an operator's perspective, since the phase is still executed, but as a hidden phase.
  The count variable of a phase that has not been executed provides 0 as output value.

Start time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
  To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the start time of the phase.

Completion time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
  To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the completion time of the phase.

TIP

To calculate a duration from two timestamps and display it in a specific format, you need to use two conversion functions on the calculation:

- **Convert to Unitless Number** (**convertTo**) takes the calculated duration and converts it into the duration's value for one of its units (e.g., minutes or seconds).
- **Convert to String for Display** (**convertToDisplayString)** takes the converted value and displays it as string to which you can add the unit, also as string.

Example:

Sample Phase with Start time = 14-Nov-2014@10:15
Sample Phase with Completion time = 14-Nov-2014@11:47
The duration is to be displayed in minutes.

```
convertToDisplayString
  (convertTo
      ({Sample Phase}.{Completion time}-{Sample Phase}.{Start time},
"min")
  )
 + " min"
```

As result of the expression, the system displays "**92 min**".

Chapter 3

Get Process Value

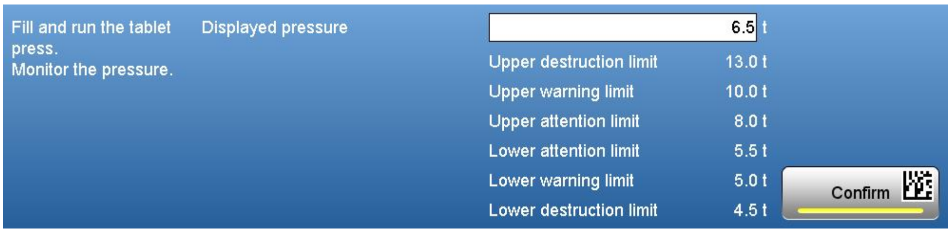
The **Get process value** phase allows an operator to capture process-related parameters.

It can be used for processing requirements, such as:

- Manual entry of room temperature
  The room temperature must range between 20°C and 22°C. These boundary values can be defined as limits and corresponding limit violations can be tracked as exceptions.
- Recording of pH values
  The pH value of a material needs to be adjusted to a specified range. When capturing the pH value, the value can be checked against a specified range. Depending on the result, the building block can trigger loops within the recipe to further adjust the pH value.
- Recording of manually entered weighing values
  The relation between actual quantities and planned quantities is essential for the final product quality. Manually entered weighing values can be checked against three limit ranges (e.g. Warning limit, Control limit, Out of specification limit).

Execution

The **Get process value** phase records process values entered during execution and can match them against configurable limits. It supports checking against up to three limit ranges.
As long as the phase is active, it provides a user-triggered exception to override the process value if it was inserted automatically and the input box is configured to be not editable.
After phase completion, it provides a post-completion exception to correct the value recorded during processing.
After completion the phase displays the recorded value, both in the Execution Window and the Navigator. Additionally, the Navigator provides access to the post-completion exception.



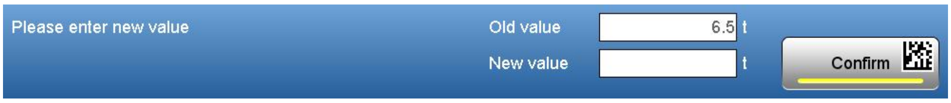
Fill and run the tablet press.
Monitor the pressure.

Displayed pressure

6.5 t

Upper destruction limit	13.0 t
Upper warning limit	10.0 t
Upper attention limit	8.0 t
Lower attention limit	5.5 t
Lower warning limit	5.0 t
Lower destruction limit	4.5 t

Figure 6: Get process value during execution

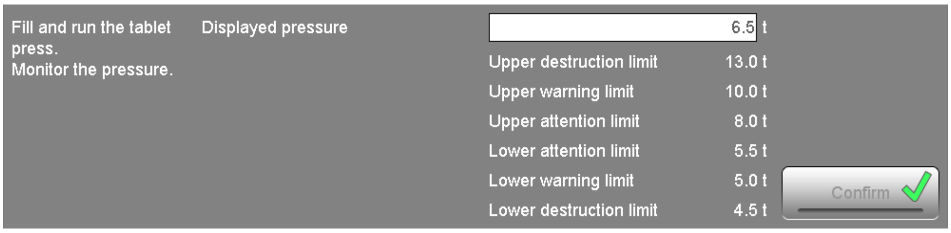


Please enter new value

Old value 6.5 t

New value t

Figure 7: User-triggered exception of Get process value



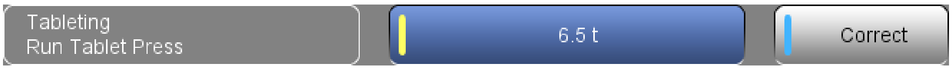
Fill and run the tablet press.
Monitor the pressure.

Displayed pressure

6.5 t

Upper destruction limit	13.0 t
Upper warning limit	10.0 t
Upper attention limit	8.0 t
Lower attention limit	5.5 t
Lower warning limit	5.0 t
Lower destruction limit	4.5 t

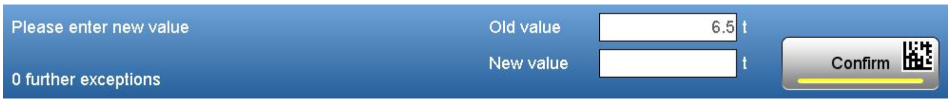
Figure 8: Get process value after phase completion



Tableting
Run Tablet Press

6.5 t

Figure 9: Get process value in the Navigator



Please enter new value

0 further exceptions

Old value 6.5 t

New value t

Figure 10: Post-completion exception of Get process value

Phase Design

The characteristics of the **Get process value** phase are defined via process parameters and their attributes.

Its user interface is designed in four columns. The first two columns are available for instruction texts. The third column displays an input box to display the default value and provides space to display the pre-defined limit ranges. When the phase is active the input box is available for entering a value, provided it is set to be editable (page 22). The rightmost column provides the **Confirm** button.

Exception handling during execution is controlled by a risk assessment classification and an exception message that are both defined by the recipe author in the exception's process parameter.

Process Parameters

The following process parameters are available to configure the phase's behavior during execution:

Instruction

Represents the instruction text that is visible on the preview, the active, and the completed view of the phase.

Attribute	Type	Comment
Column 1	HTML text	Instruction text to be displayed. Maximum length is 2000 characters (including HTML tags).
Column 2	HTML text	Instruction text to be displayed. Maximum length is 2000 characters (including HTML tags).
Column 3	HTML text	Not used.

Value configuration

Attribute	Type	Comment
UoM	Unit of measure	Must match a unit of measure available within PharmaSuite. The limit values defined with the **Limit definition** process parameter (page 22) are based on this unit of measure.
Value editable	Flag	Controls if the displayed value is editable during execution. Default setting: **Yes**

TIP

Limit values with more than 7 digits are truncated at the end in the Phase Preview.

Limit configuration

During execution, the actual process value entered in the input box is checked against the configured limits when the operator moves the focus away from the box that holds the value, for example by tapping the **Confirm** button. If the checks for the respective limit ranges are enabled, they are performed in the following order:

1. LLL-HHH
2. LL-HH
3. L-H.

L-H configuration

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If so, ensure that the **L limit** and **H limit** attributes of the **Limit definition** process parameter (page 22) are set.
Display	Flag	Controls if the limit range is displayed during execution.
Lower limit name	Text	Defines the name of the lower limit displayed during execution.
Upper limit name	Text	Defines the name of the upper limit displayed during execution.
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: **None**, **Low**, **Low (mandatory comment)**, **Medium**, **Medium (mandatory comment)**, **High**, **High (mandatory comment)**. Default setting: **High**.
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 2000 characters.

LL-HH configuration

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If so, ensure that the **LL limit** and **HH limit** attributes of the **Limit definition** process parameter (page 22) are set.
Display	Flag	Controls if the limit range is displayed during execution if the check is enabled.
Lower limit name	Text	Defines the name of the lower limit displayed during execution.
Upper limit name	Text	Defines the name of the upper limit displayed during execution.
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: **None**, **Low**, **Low (mandatory comment)**, **Medium**, **Medium (mandatory comment)**, **High**, **High (mandatory comment)**. Default setting: **High**.
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 2000 characters.

LLL-HHH configuration

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If so, ensure that the **LLL limit** and **HHH limit** attributes of the **Limit definition** process parameter (page 22) are set.
Display	Flag	Controls if the limit range is displayed during execution if the check is enabled.
Lower limit name	Text	Defines the name of the lower limit displayed during execution.
Upper limit name	Text	Defines the name of the upper limit displayed during execution.

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: **None**, **Low**, **Low (mandatory comment)**, **Medium**, **Medium (mandatory comment)**, **High**, **High (mandatory comment)**. Default setting: **High**.
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 2000 characters.

Limit definition

Limits can be defined as **Absolute** values where you set the exact limit yourself or as **Relative** values where the system calculates the limit by either adding a defined quantity to a given reference value or by subtracting the quantity from the reference value.
The limits are calculated according to the following definitions.

Limit	Absolute value definition	Relative value definition
HHH limit	HHH	Reference value + HHH
HH limit	HH	Reference value + HH
H limit	H	Reference value + H
L limit	L	Reference value - L
LL limit	LL	Reference value - LL
LLL limit	LLL	Reference value - LLL

When defining the attribute values you need to make sure that

- the unit of measure must be of the same system of measurement as the one used for the **Value configuration** process parameter (page 19) (e.g. weight: mg, kg, pound; length: mm, m, inch),
- the limits are strictly sequential and do not overlap, so that
  LLL limit < LL limit < L limit < Reference value < H limit < HH limit < HHH limit.

You can define a default value to be shown in the value box and configure if the default value is editable during execution. This way you can use an expression to draw the output of another phase into the value box and even record an exception if an operator needs to edit it.

Attribute	Type	Comment
LLL limit	MeasuredValue	Define the values of the upper limits (including the values themselves). Limit values with more than 7 digits are truncated at the end in the Phase Preview.
LL limit	MeasuredValue	
L limit	MeasuredValue	
Reference value	MeasuredValue	Defines the reference value in case of a limit range of the **Relative** limit type.
H limit	MeasuredValue	Define the values of the upper limits (including the values themselves). Limit values with more than 7 digits are truncated at the end in the Phase Preview.
HH limit	MeasuredValue	
HHH limit	MeasuredValue	
L-H type	Choice list	Define the types of the limit range (**Absolute**, **Relative**). During execution, the phase always calculates and displays absolute values. Default setting: **Absolute**.
LL-HH type	Choice list	
LLL-HHH type	Choice list	
Default value	MeasuredValue	Defines the default value.

Post - Correct value

Represents a post-completion exception that is accessible from the Navigator.
The exception allows an operator to correct the value entered while the phase was active.
It covers incidents when the operator has entered an incorrect value on account of a reading error, but has confirmed and completed the phase before detecting the error.

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: **None**, **Low**, **Low (mandatory comment)**, **Medium**, **Medium (mandatory comment)**, **High**, **High (mandatory comment)**. Default setting: **High**.
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 2000 characters.

Override value

Represents a user-triggered exception that is accessible from the Exception Window. The exception allows an operator to override the value even if it is set to read-only for regular execution, which you achieve by unselecting the **Value editable** attribute of the **Value configuration** process parameter (page 19).
It covers incidents when a reading error causes a calculated process value to fail the limits defined with the **Limit definition** process parameter (page 22), but the actual value is within the required range so that the process can be continued.

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: **None**, **Low**, **Low (mandatory comment)**, **Medium**, **Medium (mandatory comment)**, **High**, **High (mandatory comment)**. Default setting: **High**.
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 2000 characters.

Instruction links

In addition to the permanent process parameters that are always present, the **Get process value** phase provides instruction links as optional process parameters, which you can insert if required.

You can add up to ten instruction link parameters.

TIP

Instruction links are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.

ADDING INSTRUCTION LINKS

1. Click the **Add parameter** button.
   The system opens an option list that holds all optional parameter types available for the phase.

2. Select the **Instruction Link** type.
   The system opens the **Add Instruction Link** dialog to define the instruction link's identifier.

3. Type an identifier and click the **OK** button.
   The system adds the instruction link parameter and the first link definition row to the list of parameters.
   Instruction link parameters are generally inserted below all other parameters. Where within the block of instruction link parameters the system adds a new link parameter depends on the current selection in the Parameter Panel:

   - If no parameter is selected, the system adds the new instruction link parameter as last parameter.
   - If an instruction link parameter is selected, the system adds the new instruction link parameter below the selected one.
   - If any other parameter is selected, the system adds the new instruction link parameter as first parameter of the instruction link parameter block.

> TIPS
> Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa.
> The identifier of the instruction link parameter is shown as **Identifier** of the link's instruction text parameter.

4. Specify the instruction text to be displayed and mark the link texts by enclosing them in curly brackets.

5. Specify the list of link definitions. Each row of the list defines one hyperlink. The button bar above the list provides the following functions:

   ⊞ adds a new row to the list. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.

   🗑 deletes the currently selected rows.

   ⬆ moves the currently selected row one row up.

   ⬇ moves the currently selected row one row down.

Removing instruction links

1. In the list of parameters, select the instruction link parameter you wish to remove.

2. Click the **Remove parameter** button.
   The system asks you to confirm the action and then removes the instruction link parameter.

The following process parameters are available to configure the phase's behavior during execution:

Instruction text with links

Defines the text of the optional instruction link that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Attribute	Type	Comment
Instruction text	HTML text	Instruction text to be displayed. For any text enclosed in curly brackets you can define a hyperlink with the **Instruction link definition** process parameter (page 26). Example: Refer to {SOP1270} for guidance. Maximum length is 2000 characters (including HTML tags).

Instruction link definition

Defines all links to be available within the instruction text defined with the **Instruction text with links** process parameter (page 26). You can either access a file on the web, such as your intranet, by using the web access syntax or a file on your file system, by using the file access syntax.
Examples:

```
https://rockwellautomation.com
file:///c:/SOP/SOP1492.pdf
```

Attribute	Type	Comment
Link text	Text	Text to be used as link. For any text enclosed in curly brackets within the instruction text you can define a link with the **Link URL** attribute. Including the brackets in the link text is optional. Maximum length is 80 characters.

Attribute	Type	Comment
Link URL	Text	URL of the file to be displayed. The link opens the external application assigned to the file type by the operating system. Maximum length is 256 characters.

Instruction tables

In addition to the permanent process parameters that are always present, the **Get process value** phase provides instruction tables as optional process parameters, which you can insert if required.

You can add up to ten instruction tables with up to 50 table rows.

> **TIP**
>
> Instruction tables are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.

ADDING INSTRUCTION TABLES

1. Click the **Add parameter** button.
   The system opens an option list that holds all optional parameter types available for the phase.
2. Select the **Instruction Table** type.
   The system opens the **Add Instruction Table** dialog to define the instruction table's identifier.
3. Type an identifier and click the **OK** button.
   The system adds the parameter definition and the first table row to the list of parameters.
   Instruction table parameters are generally inserted below all other parameters. Where within the block of instruction table parameters the system adds a new table depends on the current selection in the Parameter Panel:
   - If no parameter is selected, the system adds the new instruction table as last parameter.
   - If an instruction table parameter is selected, the system adds the new instruction table below the selected one.
   - If any other parameter is selected, the system adds the new instruction table as first parameter of the instruction table parameter block.

> **TIPS**
> Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa.
>
> The identifier of the instruction table is shown as **Identifier** of the table's definition parameter. The identifiers of the individual table rows (**Row-1**, **Row-2**, etc.) are system-defined and not editable.

4. Specify the overall appearance of the table:
   - Select the number of columns to define the layout.
   - Set the width of the first column. If you do not set it to narrow, all columns have equal widths. If you set the first column to narrow, the remaining columns will have equal widths.
   - Define if your table needs to have borders. You can either show all borders of the table and its cells or none.
5. Specify the table rows and their content. The button bar above the rows table provides the following functions:

   ⊞ adds a new row to the table. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.

   🗑 deletes the currently selected rows.

   ⬆ moves the currently selected row one row up.

   ⬇ moves the currently selected row one row down.

> **TIP**
> Please note that the system always retains the consecutive numbering of the rows. If you reorder the table rows or delete rows, the row identifiers are updated accordingly.

REMOVING INSTRUCTION TABLES

1. In the list of parameters, select the instruction table parameter you wish to remove.
2. Click the **Remove parameter** button.
   The system asks you to confirm the action and then removes the instruction table with its definition and rows.

The following process parameters are available to configure the phase's behavior during execution:

Instruction table definition

Defines the appearance of the optional instruction table that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Attribute	Type	Comment
Table layout	Choice list	Defines the layout of the instruction table holding the instruction texts. Available settings: **1 column**, **2 columns**, **3 columns**, **4 columns**, **5 columns**. Default setting: **1 column**.
First column narrow	Boolean	Defines if the first column of the table shall be narrow.
Show all borders	Boolean	Defines if the borders of the table shall be visible.

Instruction table text

Specifies the instruction texts to be displayed in the individual cells of the instruction table.

Attribute	Type	Comment
Column 1	HTML text	Instruction text to be displayed in a column. **Restriction:** Maximum length is 2000 characters (including HTML tags).
Column 2	HTML text	
Column 3	HTML text	
Column 4	HTML text	
Column 5	HTML text	

Output Variables

Instead of specifying a fixed value to be displayed or used during execution, you can also use an expression created in the Expression editor to draw the output of another phase or operation or the calculated result of several outputs as value into a parameter attribute. When you reference outputs in this manner you need to be aware of the following restrictions:

- Only when a component has been processed does it provide an output that can be fed into another component as attribute value. For this reason, you can never reference an output of a component that is a strict successor of the component in which you try to use the output.
- Branches and loops, however, require special notice in this context, since they are only potentially passed through and/or completed during processing, so their outputs are not reliably available. Thus, you can reference any such potentially available outputs, but need to be aware of the fact that the provided value may be **Undefined** so that the component into which you are feeding the output must be able to deal with such an **Undefined** input value.

The **Get process value** phase provides the following output variables:

Value

- Data type: MeasuredValue, used for displaying numeric values qualified by a unit of measure.
- Usage: The output variable provides the complete process value as a **MeasuredValue** object.

Unit of measure

- Data type: String, used for displaying a pre-defined sequence of characters, such as "**NEXT_ITEM**" or "**COMPLETED**".
- Usage: The output variable provides the unit of measure of the process value.

Identifier

- Data type: String, used for displaying a pre-defined sequence of characters, such as "**Read Instruction**".
- Usage: The output variable provides the identifier of the phase.

Instance count

- Data type: Long, used for integral numbers:
  12345
- Usage: The output variable provides the count of the number of instances the phase has been processed, for example in a loop. The count is also increased when the phase is skipped from an operator's perspective, since the phase is still executed, but as a hidden phase.
  The count variable of a phase that has not been executed provides 0 as output value.

Start time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
  To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the start time of the phase.

Completion time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
  To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the completion time of the phase.

TIP

To calculate a duration from two timestamps and display it in a specific format, you need to use two conversion functions on the calculation:

- **Convert to Unitless Number** (**convertTo**) takes the calculated duration and converts it into the duration's value for one of its units (e.g., minutes or seconds).
- **Convert to String for Display** (**convertToDisplayString)** takes the converted value and displays it as string to which you can add the unit, also as string.

Example:

Sample Phase with Start time = 14-Nov-2014@10:15
Sample Phase with Completion time = 14-Nov-2014@11:47
The duration is to be displayed in minutes.

```
convertToDisplayString
  (convertTo
      ({Sample Phase}.{Completion time}-{Sample Phase}.{Start time},
"min")
  )
 + " min"
```

As result of the expression, the system displays "**92 min**".

Chapter 4

Get Text Value

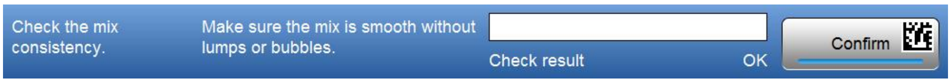
The **Get text value** phase allows an operator to record text (a string) during execution.

It can be used for processing requirements, such as:

- Recording of visual appearance during product test
  During the inspection of a product sample, the visual appearance of the sample can be documented (e.g. transparent, cloudy).
- Checking the expected representation of a recipe at an equipment unit
  On the display of a piece of equipment, the machine recipe is visualized. When setting up the equipment unit, the quality of the recipe's representation can be checked against an expected string (e.g. difficult to read).
- Recording of production resources
  Operator documents which tool was used when entering the property tag.

Execution

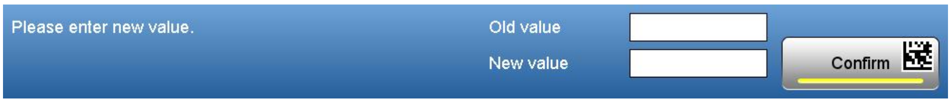
The **Get text value** phase records a string of text entered during execution and can match it against an expected value.
As long as the phase is active, it provides a user-triggered exception to override the text value if it was inserted automatically and the input box is configured to be not editable.
After phase completion, it provides a post-completion exception to correct the value recorded during processing.
After completion the phase displays the recorded value, both in the Execution Window and the Navigator. Additionally, the Navigator provides access to the post-completion exception.


Check the mix consistency.

Make sure the mix is smooth without lumps or bubbles.

Check result OK

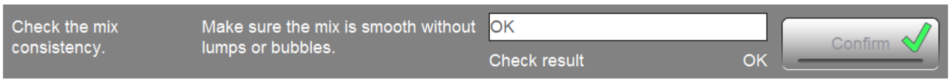
Figure 11: Get text value during execution


Please enter new value.

Old value

New value

Figure 12: User-triggered exception of Get text value

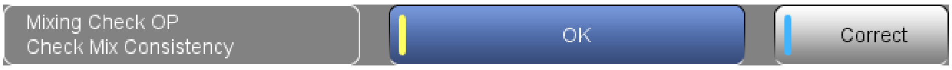

Check the mix consistency.

Make sure the mix is smooth without lumps or bubbles.

OK

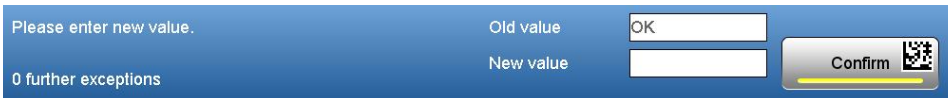
Check result OK

Figure 13: Get text value after phase completion



Mixing Check OP
Check Mix Consistency

Figure 14: Get text value in the Navigator



Please enter new value.

0 further exceptions

Old value: OK

New value:

Figure 15: Post-completion exception of Get text value

Phase Design

The characteristics of the **Get text value** phase are defined via process parameters and their attributes.
Its user interface is designed in four columns. The first two columns are available for instruction texts. The third column displays a text box to display the default text value and provides space to display a pre-defined expected value. When the phase is active the text box is available for entering a value, provided it is set to be editable (page 35). The rightmost column provides the **Confirm** button.
Exception handling during execution is controlled by a risk assessment classification and an exception message that are both defined by the recipe author in the exception's process parameter.

Process Parameters

The following process parameters are available to configure the phase's behavior during execution:

Instruction

Represents the instruction text that is visible on the preview, the active, and the completed view of the phase.

Attribute	Type	Comment
Column 1	HTML text	Instruction text to be displayed. Maximum length is 2000 characters (including HTML tags).
Column 2	HTML text	Instruction text to be displayed. Maximum length is 2000 characters (including HTML tags).
Column 3	HTML text	Not used.

Expected value configuration

Defines if the actual value entered during execution must be checked against an expected value.

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If so, ensure that the **Expected value** attribute of the **Expected value definition** process parameter (page 35) is set.
Display	Flag	Controls if an expected value is displayed during execution.
Expected value name	Text	Defines the name of the expected value.
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: **None**, **Low**, **Low (mandatory comment)**, **Medium**, **Medium (mandatory comment)**, **High**, **High (mandatory comment)**. Default setting: **High**.
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 2000 characters.

Expected value definition

Defines the text required as expected value (page 35) if the respective check is enabled. You can define a default value to be shown as expected value and configure if the default value is editable during execution. This way you can use an expression to draw the output of another phase into the text value box and even record an exception if an operator needs to edit it.

Attribute	Type	Comment
Expected value	Text	Defines the expected value. Maximum length is 256 characters.
Default value	Text	Defines the default value. Maximum length is 256 characters.

Attribute	Type	Comment
Value editable	Flag	Controls if the displayed value is editable during execution. Default setting: **Yes**

Post - Correct value

Represents a post-completion exception that is accessible from the Navigator.
The exception allows an operator to correct the value entered while the phase was active.
It covers incidents when the operator has entered an incorrect value on account of a reading error, but has confirmed and completed the phase before detecting the error.

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: **None**, **Low**, **Low (mandatory comment)**, **Medium**, **Medium (mandatory comment)**, **High**, **High (mandatory comment)**. Default setting: **High**.
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 2000 characters.

Override value

Represents a user-triggered exception that is accessible from the Exception Window.
The exception allows an operator to override the value even if it is set to read-only for regular execution, which you achieve by unselecting the **Value editable** attribute of the **Expected value definition** process parameter (page 35).
It covers incidents when a reading error causes an automatically established value to deviate from the expected value, but the actual value is correct so that the process can be continued.

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: **None**, **Low**, **Low (mandatory comment)**, **Medium**, **Medium (mandatory comment)**, **High**, **High (mandatory comment)**. Default setting: **High**.
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 2000 characters.

Instruction links

In addition to the permanent process parameters that are always present, the **Get text value** phase provides instruction links as optional process parameters, which you can insert if required.

You can add up to ten instruction link parameters.

TIP

Instruction links are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.

ADDING INSTRUCTION LINKS

1. Click the **Add parameter** button.
   The system opens an option list that holds all optional parameter types available for the phase.
2. Select the **Instruction Link** type.
   The system opens the **Add Instruction Link** dialog to define the instruction link's identifier.
3. Type an identifier and click the **OK** button.
   The system adds the instruction link parameter and the first link definition row to the list of parameters.
   Instruction link parameters are generally inserted below all other parameters. Where within the block of instruction link parameters the system adds a new link parameter depends on the current selection in the Parameter Panel:

- If no parameter is selected, the system adds the new instruction link parameter as last parameter.
- If an instruction link parameter is selected, the system adds the new instruction link parameter below the selected one.
- If any other parameter is selected, the system adds the new instruction link parameter as first parameter of the instruction link parameter block.

TIPS

Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa.
The identifier of the instruction link parameter is shown as **Identifier** of the link's instruction text parameter.

4. Specify the instruction text to be displayed and mark the link texts by enclosing them in curly brackets.
5. Specify the list of link definitions. Each row of the list defines one hyperlink. The button bar above the list provides the following functions:

   ⊞ adds a new row to the list. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.

   🗑 deletes the currently selected rows.

   ⬆ moves the currently selected row one row up.

   ⬇ moves the currently selected row one row down.

Removing instruction links

1. In the list of parameters, select the instruction link parameter you wish to remove.
2. Click the **Remove parameter** button.
   The system asks you to confirm the action and then removes the instruction link parameter.

The following process parameters are available to configure the phase's behavior during execution:

Instruction text with links

Defines the text of the optional instruction link that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Attribute	Type	Comment
Instruction text	HTML text	Instruction text to be displayed. For any text enclosed in curly brackets you can define a hyperlink with the **Instruction link definition** process parameter (page 39). Example: Refer to {SOP1270} for guidance. Maximum length is 2000 characters (including HTML tags).

Instruction link definition

Defines all links to be available within the instruction text defined with the **Instruction text with links** process parameter (page 38). You can either access a file on the web, such as your intranet, by using the web access syntax or a file on your file system, by using the file access syntax.

Examples:

```
https://rockwellautomation.com
file:///c:/SOP/SOP1492.pdf
```

Attribute	Type	Comment
Link text	Text	Text to be used as link. For any text enclosed in curly brackets within the instruction text you can define a link with the **Link URL** attribute. Including the brackets in the link text is optional. Maximum length is 80 characters.
Link URL	Text	URL of the file to be displayed. The link opens the external application assigned to the file type by the operating system. Maximum length is 256 characters.

Instruction tables

In addition to the permanent process parameters that are always present, the **Get text value** phase provides instruction tables as optional process parameters, which you can insert if required.

You can add up to ten instruction tables with up to 50 table rows.

TIP

Instruction tables are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.

ADDING INSTRUCTION TABLES

1. Click the **Add parameter** button.
   The system opens an option list that holds all optional parameter types available for the phase.
2. Select the **Instruction Table** type.
   The system opens the **Add Instruction Table** dialog to define the instruction table's identifier.
3. Type an identifier and click the **OK** button.
   The system adds the parameter definition and the first table row to the list of parameters.
   Instruction table parameters are generally inserted below all other parameters. Where within the block of instruction table parameters the system adds a new table depends on the current selection in the Parameter Panel:
   - If no parameter is selected, the system adds the new instruction table as last parameter.
   - If an instruction table parameter is selected, the system adds the new instruction table below the selected one.
   - If any other parameter is selected, the system adds the new instruction table as first parameter of the instruction table parameter block.

   TIPS
   Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa.

   The identifier of the instruction table is shown as **Identifier** of the table's definition parameter. The identifiers of the individual table rows (**Row-1**, **Row-2**, etc.) are system-defined and not editable.
4. Specify the overall appearance of the table:
   - Select the number of columns to define the layout.
   - Set the width of the first column. If you do not set it to narrow, all columns have equal widths. If you set the first column to narrow, the remaining columns will have equal widths.

- Define if your table needs to have borders. You can either show all borders of the table and its cells or none.

5. Specify the table rows and their content. The button bar above the rows table provides the following functions:

   ⊞ adds a new row to the table. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.

   🗑 deletes the currently selected rows.

   ⬆ moves the currently selected row one row up.

   ⬇ moves the currently selected row one row down.

   > TIP
   > Please note that the system always retains the consecutive numbering of the rows. If you reorder the table rows or delete rows, the row identifiers are updated accordingly.

Removing instruction tables

1. In the list of parameters, select the instruction table parameter you wish to remove.
2. Click the **Remove parameter** button.
   The system asks you to confirm the action and then removes the instruction table with its definition and rows.

The following process parameters are available to configure the phase's behavior during execution:

Instruction table definition

Defines the appearance of the optional instruction table that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Attribute	Type	Comment
Table layout	Choice list	Defines the layout of the instruction table holding the instruction texts. Available settings: **1 column**, **2 columns**, **3 columns**, **4 columns**, **5 columns**. Default setting: **1 column**.

Attribute	Type	Comment
First column narrow	Boolean	Defines if the first column of the table shall be narrow.
Show all borders	Boolean	Defines if the borders of the table shall be visible.

Instruction table text

Specifies the instruction texts to be displayed in the individual cells of the instruction table.

Attribute	Type	Comment
Column 1	HTML text	Instruction text to be displayed in a column. **Restriction:** Maximum length is 2000 characters (including HTML tags).
Column 2	HTML text	
Column 3	HTML text	
Column 4	HTML text	
Column 5	HTML text	

Output Variables

Instead of specifying a fixed value to be displayed or used during execution, you can also use an expression created in the Expression editor to draw the output of another phase or operation or the calculated result of several outputs as value into a parameter attribute. When you reference outputs in this manner you need to be aware of the following restrictions:

- Only when a component has been processed does it provide an output that can be fed into another component as attribute value. For this reason, you can never reference an output of a component that is a strict successor of the component in which you try to use the output.
- Branches and loops, however, require special notice in this context, since they are only potentially passed through and/or completed during processing, so their outputs are not reliably available. Thus, you can reference any such potentially available outputs, but need to be aware of the fact that the provided value may be **Undefined** so that the component into which you are feeding the output must be able to deal with such an **Undefined** input value.

The **Get text value** phase provides the following output variables:

Value

- Data type: String, used for displaying a pre-defined sequence of characters, such as "**NEXT_ITEM**" or "**COMPLETED**".
- Usage: The output variable provides the text value entered during execution as string of characters.

Identifier

- Data type: String, used for displaying a pre-defined sequence of characters, such as "**Read Instruction**".
- Usage: The output variable provides the identifier of the phase.

Instance count

- Data type: Long, used for integral numbers:
  12345
- Usage: The output variable provides the count of the number of instances the phase has been processed, for example in a loop. The count is also increased when the phase is skipped from an operator's perspective, since the phase is still executed, but as a hidden phase.
  The count variable of a phase that has not been executed provides 0 as output value.

Start time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
  To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the start time of the phase.

Completion time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
  To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the completion time of the phase.

TIP

To calculate a duration from two timestamps and display it in a specific format, you need to use two conversion functions on the calculation:

- **Convert to Unitless Number** (**convertTo**) takes the calculated duration and converts it into the duration's value for one of its units (e.g., minutes or seconds).
- **Convert to String for Display** (**convertToDisplayString)** takes the converted value and displays it as string to which you can add the unit, also as string.

Example:

Sample Phase with Start time = 14-Nov-2014@10:15
Sample Phase with Completion time = 14-Nov-2014@11:47
The duration is to be displayed in minutes.

```
convertToDisplayString
   (convertTo
        ({Sample Phase}.{Completion time}-{Sample Phase}.{Start time},
"min")
   )
 + " min"
```

As result of the expression, the system displays "**92 min**".

Chapter 5

Show Document

The **Show document** phase allows to display a variety of documents.

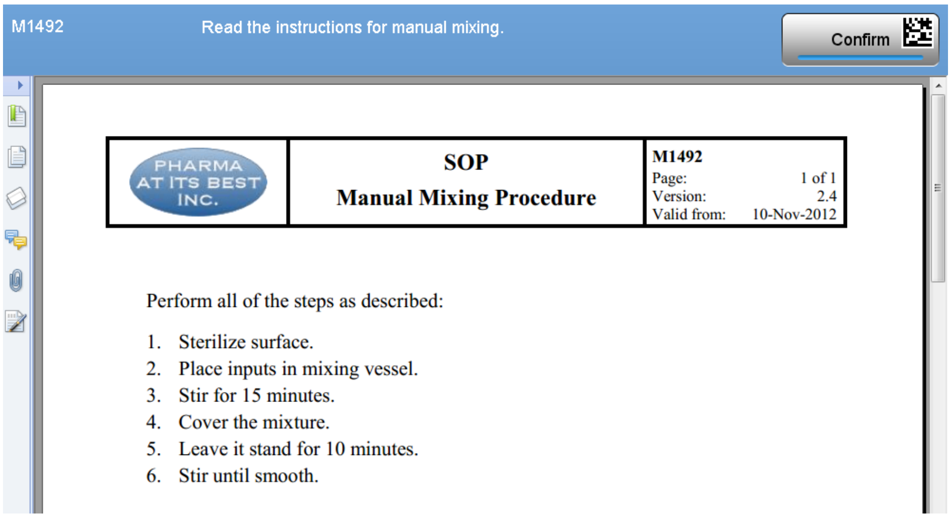
It can be used to provide complex information to an operator, such as:

- Display an SOP
  An SOP is stored within a central DMS. This very SOP can be displayed to the operator during execution.
- Show an instruction video
  The instruction of a GMP-critical process step is available as a video. The video can be shown during execution.

> TIP
>
> Please note that the **Show document** phase uses **Work instruction** objects of Process Designer as file basis. So any file you wish to display during execution must be available in Process Designer and be referenced with the file name it has in Process Designer.

Execution

The **Show document** phase can show information or instructions in various formats, such as PDF documents, images, or video files, to an operator. It presents the referenced work instruction in an HTML control.
After completion the phase displays the name of the referenced work instruction in the Navigator.



M1492 Read the instructions for manual mixing.

SOP

Manual Mixing Procedure

M1492

Page: 1 of 1

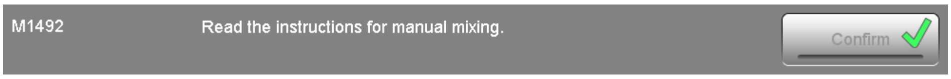
Version: 2.4

Valid from: 10-Nov-2012

Perform all of the steps as described:

1. Sterilize surface.
2. Place inputs in mixing vessel.
3. Stir for 15 minutes.
4. Cover the mixture.
5. Leave it stand for 10 minutes.
6. Stir until smooth.

Figure 16: Show document during execution



M1492 Read the instructions for manual mixing.

Figure 17: Show document after phase completion

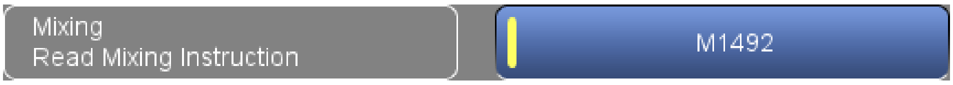


Figure 18: Show document in the Navigator

Phase Design

The characteristics of the **Show document** phase are defined via process parameters and their attributes.

Its user interface is designed in three columns that are extended by a second row that holds a full-width HTML control when the phase becomes active. The left column of the first row displays the name of the referenced work instruction while the second column shows an instruction text that can tell the user what to do with the referenced file. The rightmost column provides the **Confirm** button.

When the phase is completed, it reverts back to its initial one-row and three-column layout.

Process Parameters

The following process parameters are available to configure the phase's behavior during execution:

Instruction

Represents the instruction text that is visible on the preview, the active, and the completed view of the phase.

Attribute	Type	Comment
Column 1	HTML text	Instruction text to be displayed. Maximum length is 2000 characters (including HTML tags).
Column 2	HTML text	Not used.
Column 3	HTML text	Not used.

Document

Defines the name of the file to be displayed during execution.

Attribute	Type	Comment
Work instruction	Text	Name of a FactoryTalk ProductionCentre work instruction object. The document will be shown within an HTML container.

Instruction links

In addition to the permanent process parameters that are always present, the **Show document** phase provides instruction links as optional process parameters, which you can insert if required.

You can add up to ten instruction link parameters.

TIP

Instruction links are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.

➕ Adding instruction links

1. Click the **Add parameter** button.
   The system opens an option list that holds all optional parameter types available for the phase.
2. Select the **Instruction Link** type.
   The system opens the **Add Instruction Link** dialog to define the instruction link's identifier.

3. Type an identifier and click the **OK** button.
   The system adds the instruction link parameter and the first link definition row to the list of parameters.
   Instruction link parameters are generally inserted below all other parameters. Where within the block of instruction link parameters the system adds a new link parameter depends on the current selection in the Parameter Panel:

   - If no parameter is selected, the system adds the new instruction link parameter as last parameter.
   - If an instruction link parameter is selected, the system adds the new instruction link parameter below the selected one.
   - If any other parameter is selected, the system adds the new instruction link parameter as first parameter of the instruction link parameter block.

   > **TIPS**
   > Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa.
   > The identifier of the instruction link parameter is shown as **Identifier** of the link's instruction text parameter.

4. Specify the instruction text to be displayed and mark the link texts by enclosing them in curly brackets.

5. Specify the list of link definitions. Each row of the list defines one hyperlink. The button bar above the list provides the following functions:

   ⊞ adds a new row to the list. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.

   🗑 deletes the currently selected rows.

   ⬆ moves the currently selected row one row up.

   ⬇ moves the currently selected row one row down.

Removing instruction links

1. In the list of parameters, select the instruction link parameter you wish to remove.
2. Click the **Remove parameter** button.
   The system asks you to confirm the action and then removes the instruction link parameter.

The following process parameters are available to configure the phase's behavior during execution:

Instruction text with links

Defines the text of the optional instruction link that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Attribute	Type	Comment
Instruction text	HTML text	Instruction text to be displayed. For any text enclosed in curly brackets you can define a hyperlink with the **Instruction link definition** process parameter (page 49). Example: Refer to {SOP1270} for guidance. Maximum length is 2000 characters (including HTML tags).

Instruction link definition

Defines all links to be available within the instruction text defined with the **Instruction text with links** process parameter (page 49). You can either access a file on the web, such as your intranet, by using the web access syntax or a file on your file system, by using the file access syntax.

Examples:

```
https://rockwellautomation.com
file:///c:/SOP/SOP1492.pdf
```

Attribute	Type	Comment
Link text	Text	Text to be used as link. For any text enclosed in curly brackets within the instruction text you can define a link with the **Link URL** attribute. Including the brackets in the link text is optional. Maximum length is 80 characters.
Link URL	Text	URL of the file to be displayed. The link opens the external application assigned to the file type by the operating system. Maximum length is 256 characters.

Instruction tables

In addition to the permanent process parameters that are always present, the **Show document** phase provides instruction tables as optional process parameters, which you can insert if required.

You can add up to ten instruction tables with up to 50 table rows.

TIP

Instruction tables are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.

ADDING INSTRUCTION TABLES

1. Click the **Add parameter** button.
   The system opens an option list that holds all optional parameter types available for the phase.
2. Select the **Instruction Table** type.
   The system opens the **Add Instruction Table** dialog to define the instruction table's identifier.
3. Type an identifier and click the **OK** button.
   The system adds the parameter definition and the first table row to the list of parameters.
   Instruction table parameters are generally inserted below all other parameters. Where within the block of instruction table parameters the system adds a new table depends on the current selection in the Parameter Panel:
   - If no parameter is selected, the system adds the new instruction table as last parameter.
   - If an instruction table parameter is selected, the system adds the new instruction table below the selected one.
   - If any other parameter is selected, the system adds the new instruction table as first parameter of the instruction table parameter block.

   TIPS

   Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa.

   The identifier of the instruction table is shown as **Identifier** of the table's definition parameter. The identifiers of the individual table rows (**Row-1**, **Row-2**, etc.) are system-defined and not editable.

4. Specify the overall appearance of the table:
   - Select the number of columns to define the layout.
   - Set the width of the first column. If you do not set it to narrow, all columns have equal widths. If you set the first column to narrow, the remaining columns will have equal widths.
   - Define if your table needs to have borders. You can either show all borders of the table and its cells or none.
5. Specify the table rows and their content. The button bar above the rows table provides the following functions:

   ⊞ adds a new row to the table. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.

   🗑 deletes the currently selected rows.

   ⬆ moves the currently selected row one row up.

   ⬇ moves the currently selected row one row down.

> TIP
> Please note that the system always retains the consecutive numbering of the rows. If you reorder the table rows or delete rows, the row identifiers are updated accordingly.

Removing Instruction Tables

1. In the list of parameters, select the instruction table parameter you wish to remove.
2. Click the **Remove parameter** button.
   The system asks you to confirm the action and then removes the instruction table with its definition and rows.

The following process parameters are available to configure the phase's behavior during execution:

Instruction table definition

Defines the appearance of the optional instruction table that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Attribute	Type	Comment
Table layout	Choice list	Defines the layout of the instruction table holding the instruction texts. Available settings: **1 column**, **2 columns**, **3 columns**, **4 columns**, **5 columns**. Default setting: **1 column**.
First column narrow	Boolean	Defines if the first column of the table shall be narrow.
Show all borders	Boolean	Defines if the borders of the table shall be visible.

Instruction table text

Specifies the instruction texts to be displayed in the individual cells of the instruction table.

Attribute	Type	Comment
Column 1	HTML text	Instruction text to be displayed in a column. **Restriction:** Maximum length is 2000 characters (including HTML tags).
Column 2	HTML text	
Column 3	HTML text	
Column 4	HTML text	
Column 5	HTML text	

Output Variables

Instead of specifying a fixed value to be displayed or used during execution, you can also use an expression created in the Expression editor to draw the output of another phase or operation or the calculated result of several outputs as value into a parameter attribute. When you reference outputs in this manner you need to be aware of the following restrictions:

- Only when a component has been processed does it provide an output that can be fed into another component as attribute value. For this reason, you can never reference an output of a component that is a strict successor of the component in which you try to use the output.
- Branches and loops, however, require special notice in this context, since they are only potentially passed through and/or completed during processing, so their outputs are not reliably available. Thus, you can reference any such potentially available outputs, but need to be aware of the fact that the provided value may be **Undefined** so that the component into which you are feeding the output must be able to deal with such an **Undefined** input value.

The **Show document** phase provides the following output variables:

Identifier

- Data type: String, used for displaying a pre-defined sequence of characters, such as "**Read Instruction**".
- Usage: The output variable provides the identifier of the phase.

Instance count

- Data type: Long, used for integral numbers:
  12345
- Usage: The output variable provides the count of the number of instances the phase has been processed, for example in a loop. The count is also increased when the phase is skipped from an operator's perspective, since the phase is still executed, but as a hidden phase.
  The count variable of a phase that has not been executed provides 0 as output value.

Start time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
  To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.

- Usage: The output variable provides the start time of the phase.

Completion time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
  To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the completion time of the phase.

TIP

To calculate a duration from two timestamps and display it in a specific format, you need to use two conversion functions on the calculation:

- **Convert to Unitless Number** (**convertTo**) takes the calculated duration and converts it into the duration's value for one of its units (e.g., minutes or seconds).
- **Convert to String for Display** (**convertToDisplayString)** takes the converted value and displays it as string to which you can add the unit, also as string.

Example:

Sample Phase with Start time = 14-Nov-2014@10:15
Sample Phase with Completion time = 14-Nov-2014@11:47
The duration is to be displayed in minutes.

```
convertToDisplayString
   (convertTo
        ({Sample Phase}.{Completion time}-{Sample Phase}.{Start time},
"min")
   )
 + " min"
```

As result of the expression, the system displays "**92 min**".

Chapter 6

Show Instruction Text

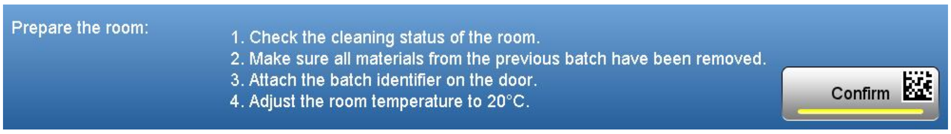
The **Show instruction text** phase allows to display a specific instruction related to the process step the operator is executing.

It can be used to provide an operator with directions such as:

- Description of how to assemble equipment.
- Description of how to sample the product.
- Information about specifics of a process activity.

Execution

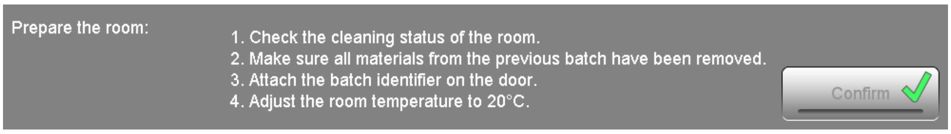
The **Show instruction text** phase displays textual instructions to an operator. After completion it displays a blank detail information button in the Navigator.



Prepare the room:

1. Check the cleaning status of the room.
2. Make sure all materials from the previous batch have been removed.
3. Attach the batch identifier on the door.
4. Adjust the room temperature to 20°C.

Figure 19: Show instruction text during execution



Prepare the room:

1. Check the cleaning status of the room.
2. Make sure all materials from the previous batch have been removed.
3. Attach the batch identifier on the door.
4. Adjust the room temperature to 20°C.

Figure 20: Show instruction text after phase completion

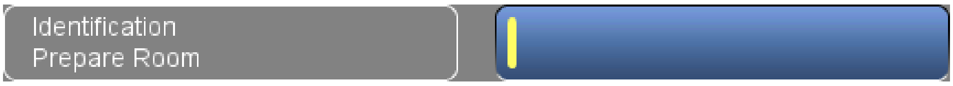


Figure 21: Show instruction text in the Navigator

Phase Design

The characteristics of the **Show instruction text** phase are defined via process parameters and their attributes.

Its user interface is designed to hold up to four columns. The number and width of the columns is controlled by the **Layout** process parameter (page 56). The rightmost column provides the **Confirm** button.

Process Parameters

The following process parameters are available to configure the phase's behavior during execution:

Instruction

Represents the instruction text that is visible on the preview, the active, and the completed view of the phase.

Attribute	Type	Comment
Column 1	HTML text	Instruction text to be displayed. Maximum length is 2000 characters (including HTML tags). Availability and width of the columns depends on the **Layout** process parameter (page 56).
Column 2	HTML text	
Column 3	HTML text	

Layout

Defines the column layout of the phase to cater for various text layouts.

Attribute	Type	Comment
Type	Choice list	Defines the layout of the column(s) holding the instruction texts: **1** column, **2** columns (with narrow first column and wide second column), **2** columns (with wide first column and narrow second column), or **3** columns). Default setting: **1 column.**

Instruction links

In addition to the permanent process parameters that are always present, the **Show instruction text** phase provides instruction links as optional process parameters, which you can insert if required.

You can add up to ten instruction link parameters.

TIP

Instruction links are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.

ADDING INSTRUCTION LINKS

1. Click the **Add parameter** button.
   The system opens an option list that holds all optional parameter types available for the phase.

2. Select the **Instruction Link** type.
   The system opens the **Add Instruction Link** dialog to define the instruction link's identifier.

3. Type an identifier and click the **OK** button.
   The system adds the instruction link parameter and the first link definition row to the list of parameters.
   Instruction link parameters are generally inserted below all other parameters. Where within the block of instruction link parameters the system adds a new link parameter depends on the current selection in the Parameter Panel:

   - If no parameter is selected, the system adds the new instruction link parameter as last parameter.

   - If an instruction link parameter is selected, the system adds the new instruction link parameter below the selected one.

   - If any other parameter is selected, the system adds the new instruction link parameter as first parameter of the instruction link parameter block.

> **TIPS**
> Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa.
> The identifier of the instruction link parameter is shown as **Identifier** of the link's instruction text parameter.

4. Specify the instruction text to be displayed and mark the link texts by enclosing them in curly brackets.

5. Specify the list of link definitions. Each row of the list defines one hyperlink. The button bar above the list provides the following functions:

   ⊞ adds a new row to the list. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.

   🗑 deletes the currently selected rows.

   ⬆ moves the currently selected row one row up.

   ⬇ moves the currently selected row one row down.

Removing instruction links

1. In the list of parameters, select the instruction link parameter you wish to remove.
2. Click the **Remove parameter** button.
   The system asks you to confirm the action and then removes the instruction link parameter.

The following process parameters are available to configure the phase's behavior during execution:

Instruction text with links

Defines the text of the optional instruction link that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Attribute	Type	Comment
Instruction text	HTML text	Instruction text to be displayed. For any text enclosed in curly brackets you can define a hyperlink with the **Instruction link definition** process parameter (page 58). Example: Refer to {SOP1270} for guidance. Maximum length is 2000 characters (including HTML tags).

Instruction link definition

Defines all links to be available within the instruction text defined with the **Instruction text with links** process parameter (page 58). You can either access a file on the web, such as your intranet, by using the web access syntax or a file on your file system, by using the file access syntax.
Examples:

```
https://rockwellautomation.com
file:///c:/SOP/SOP1492.pdf
```

Attribute	Type	Comment
Link text	Text	Text to be used as link. For any text enclosed in curly brackets within the instruction text you can define a link with the **Link URL** attribute. Including the brackets in the link text is optional. Maximum length is 80 characters.
Link URL	Text	URL of the file to be displayed. The link opens the external application assigned to the file type by the operating system. Maximum length is 256 characters.

Instruction tables

In addition to the permanent process parameters that are always present, the **Show instruction text** phase provides instruction tables as optional process parameters, which you can insert if required.

You can add up to ten instruction tables with up to 50 table rows.

TIP

Instruction tables are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.

ADDING INSTRUCTION TABLES

1. Click the **Add parameter** button.
   The system opens an option list that holds all optional parameter types available for the phase.
2. Select the **Instruction Table** type.
   The system opens the **Add Instruction Table** dialog to define the instruction table's identifier.
3. Type an identifier and click the **OK** button.
   The system adds the parameter definition and the first table row to the list of parameters.
   Instruction table parameters are generally inserted below all other parameters.

Where within the block of instruction table parameters the system adds a new table depends on the current selection in the Parameter Panel:

- If no parameter is selected, the system adds the new instruction table as last parameter.
- If an instruction table parameter is selected, the system adds the new instruction table below the selected one.
- If any other parameter is selected, the system adds the new instruction table as first parameter of the instruction table parameter block.

> **TIPS**
> Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa.
>
> The identifier of the instruction table is shown as **Identifier** of the table's definition parameter. The identifiers of the individual table rows (**Row-1**, **Row-2**, etc.) are system-defined and not editable.

4. Specify the overall appearance of the table:
   - Select the number of columns to define the layout.
   - Set the width of the first column. If you do not set it to narrow, all columns have equal widths. If you set the first column to narrow, the remaining columns will have equal widths.
   - Define if your table needs to have borders. You can either show all borders of the table and its cells or none.
5. Specify the table rows and their content. The button bar above the rows table provides the following functions:

   ⊞ adds a new row to the table. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.

   🗑 deletes the currently selected rows.

   ⬆ moves the currently selected row one row up.

   ⬇ moves the currently selected row one row down.

> **TIP**
> Please note that the system always retains the consecutive numbering of the rows. If you reorder the table rows or delete rows, the row identifiers are updated accordingly.

REMOVING INSTRUCTION TABLES

1. In the list of parameters, select the instruction table parameter you wish to remove.
2. Click the **Remove parameter** button.
   The system asks you to confirm the action and then removes the instruction table with its definition and rows.

The following process parameters are available to configure the phase's behavior during execution:

Instruction table definition

Defines the appearance of the optional instruction table that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Attribute	Type	Comment
Table layout	Choice list	Defines the layout of the instruction table holding the instruction texts. Available settings: **1 column**, **2 columns**, **3 columns**, **4 columns**, **5 columns**. Default setting: **1 column**.
First column narrow	Boolean	Defines if the first column of the table shall be narrow.
Show all borders	Boolean	Defines if the borders of the table shall be visible.

Instruction table text

Specifies the instruction texts to be displayed in the individual cells of the instruction table.

Attribute	Type	Comment
Column 1	HTML text	Instruction text to be displayed in a

Attribute	Type	Comment
Column 2	HTML text	
Column 3	HTML text	
Column 4	HTML text	
Column 5	HTML text	

Output Variables

Instead of specifying a fixed value to be displayed or used during execution, you can also use an expression created in the Expression editor to draw the output of another phase or operation or the calculated result of several outputs as value into a parameter attribute. When you reference outputs in this manner you need to be aware of the following restrictions:

- Only when a component has been processed does it provide an output that can be fed into another component as attribute value. For this reason, you can never reference an output of a component that is a strict successor of the component in which you try to use the output.
- Branches and loops, however, require special notice in this context, since they are only potentially passed through and/or completed during processing, so their outputs are not reliably available. Thus, you can reference any such potentially available outputs, but need to be aware of the fact that the provided value may be **Undefined** so that the component into which you are feeding the output must be able to deal with such an **Undefined** input value.

The **Show instruction text** phase provides the following output variables:

Identifier

- Data type: String, used for displaying a pre-defined sequence of characters, such as "**Read Instruction**".
- Usage: The output variable provides the identifier of the phase.

Instance count

- Data type: Long, used for integral numbers:
  12345
- Usage: The output variable provides the count of the number of instances the phase has been processed, for example in a loop. The count is also increased when

the phase is skipped from an operator's perspective, since the phase is still executed, but as a hidden phase.
The count variable of a phase that has not been executed provides 0 as output value.

Start time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
  To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the start time of the phase.

Completion time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
  To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the completion time of the phase.

TIP

To calculate a duration from two timestamps and display it in a specific format, you need to use two conversion functions on the calculation:

- **Convert to Unitless Number** (**convertTo**) takes the calculated duration and converts it into the duration's value for one of its units (e.g., minutes or seconds).
- **Convert to String for Display** (**convertToDisplayString)** takes the converted value and displays it as string to which you can add the unit, also as string.

Example:

Sample Phase with Start time = 14-Nov-2014@10:15
Sample Phase with Completion time = 14-Nov-2014@11:47
The duration is to be displayed in minutes.

```
convertToDisplayString
  (convertTo
      ({Sample Phase}.{Completion time}-{Sample Phase}.{Start time},
"min")
  )
 + " min"
```

As result of the expression, the system displays "**92 min**".

Show URL

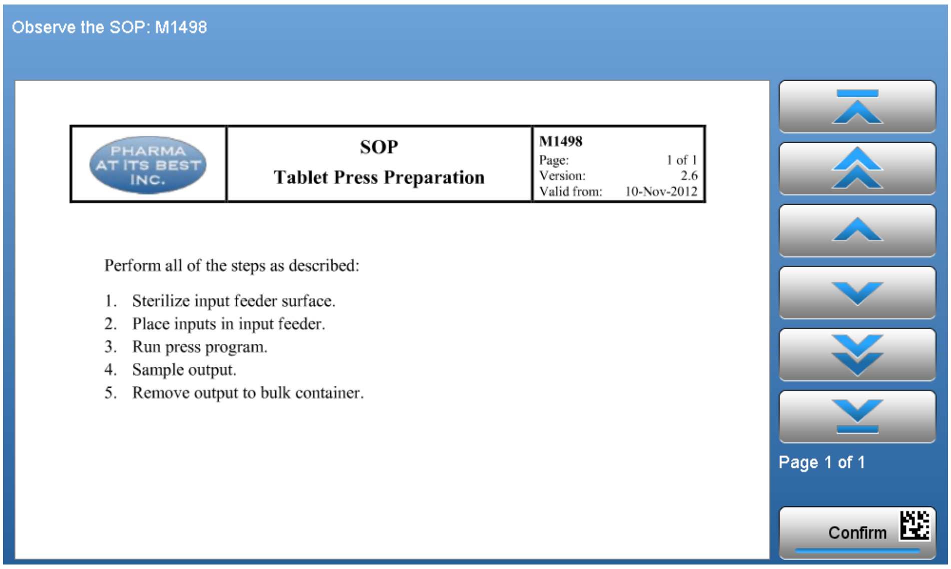
The **Show URL** phase allows to display PDF documents accessible via a URL.

It can be used to provide complex information to an operator, such as:

- Display an SOP
  An SOP is stored on the intranet or the file system. This very SOP can be displayed to the operator during execution.

Execution

The **Show URL** phase can show information or instructions in PDF format. A URL defines the location of the document to be displayed, thus providing access to PDF files by file link or web link (page 67).
After completion it displays the URL of the linked document. The Navigator displays the name of the phase.



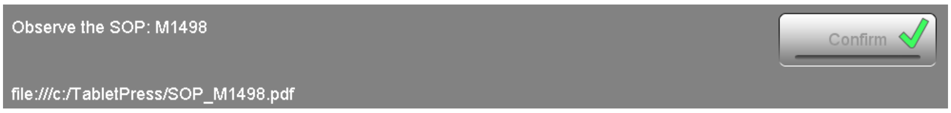
Observe the SOP: M1498

PHARMA AT ITS BEST INC.	SOP Tablet Press Preparation	M1498 Page: 1 of 1 Version: 2.6 Valid from: 10-Nov-2012

Perform all of the steps as described:

1. Sterilize input feeder surface.
2. Place inputs in input feeder.
3. Run press program.
4. Sample output.
5. Remove output to bulk container.

Figure 22: Show URL during execution



Observe the SOP: M1498

file:///c:/TabletPress/SOP_M1498.pdf

Figure 23: Show URL after phase completion

Identification
Prepare Tableting Unit

Prepare Tableting Unit

Figure 24: Show URL in the Navigator

Phase Design

The characteristics of the **Show URL** phase are defined via process parameters and their attributes.

Its user interface is designed in two columns that span two rows when the phase is active. The merged columns of the first row provide space for textual instructions. The rightmost column provides the **Confirm** button. When the phase is active, the **Confirm** button is shown at the bottom of the second row, while the first row only holds the instruction text. The first column of the second row is dedicated to displaying the pre-defined PDF file. To the right of the displayed document, the phase provides navigation buttons to page or scroll through the document: **First page**, **Previous page**, **Scroll up**, **Scroll down**, **Next page**, **Last page**. Below the document navigation buttons, it shows the page number of the currently displayed page and the total number of pages.

When the phase is completed, the **Confirm** button appears again in the second column of the first row and the second row only holds the pre-defined URL used to access the PDF document.

Exception handling during execution is controlled by a risk assessment classification and an exception message that are both defined by the recipe author in the exception's process parameter.

Process Parameters

The following process parameters are available to configure the phase's behavior during execution:

Instruction

Represents the instruction text that is visible on the preview, the active, and the completed view of the phase.

Attribute	Type	Comment
Column 1	HTML text	Instruction text to be displayed. Maximum length is 2000 characters (including HTML tags).
Column 2	HTML text	Not used.
Column 3	HTML text	Not used.

Document

Defines the URL that links to the PDF file to be displayed during execution. You can either access a file on the web, such as your intranet, by using the web access syntax (`http://...`) or a file on your file system, by using the file access syntax (`file:///...`), as defined with Java by RFC 2396: Uniform Resource Identifiers (URI): Generic Syntax, amended by RFC 2732: Format for Literal IPv6 Addresses in URLs.

TIP

When using a URL to display a PDF file during execution you need to make sure that the network location to which the URL links is accessible from the execution work station on the shop floor.

Attribute	Type	Comment
URL	Text	URL of the document to be displayed. The document will be shown within a PDF viewer.

Loading failed

Represents a system-triggered exception that is displayed in the Exception Window. It provides a way to proceed if there is a technical issue that prevents the document from loading.

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: **None**, **Low**, **Low (mandatory comment)**, **Medium**, **Medium (mandatory comment)**, **High**, **High (mandatory comment)**. Default setting: **High**.
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 250 characters.

Instruction links

In addition to the permanent process parameters that are always present, the **Show URL** phase provides instruction links as optional process parameters, which you can insert if required.

You can add up to ten instruction link parameters.

TIP

Instruction links are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.

ADDING INSTRUCTION LINKS

1. Click the **Add parameter** button.
   The system opens an option list that holds all optional parameter types available for the phase.
2. Select the **Instruction Link** type.
   The system opens the **Add Instruction Link** dialog to define the instruction link's identifier.
3. Type an identifier and click the **OK** button.
   The system adds the instruction link parameter and the first link definition row to the list of parameters.
   Instruction link parameters are generally inserted below all other parameters. Where within the block of instruction link parameters the system adds a new link parameter depends on the current selection in the Parameter Panel:
   - If no parameter is selected, the system adds the new instruction link parameter as last parameter.
   - If an instruction link parameter is selected, the system adds the new instruction link parameter below the selected one.
   - If any other parameter is selected, the system adds the new instruction link parameter as first parameter of the instruction link parameter block.

   TIPS
   Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa.
   The identifier of the instruction link parameter is shown as **Identifier** of the link's instruction text parameter.
4. Specify the instruction text to be displayed and mark the link texts by enclosing them in curly brackets.

5. Specify the list of link definitions. Each row of the list defines one hyperlink. The button bar above the list provides the following functions:

   ⊞ adds a new row to the list. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.

   🗑 deletes the currently selected rows.

   ⬆ moves the currently selected row one row up.

   ⬇ moves the currently selected row one row down.

Removing instruction links

1. In the list of parameters, select the instruction link parameter you wish to remove.
2. Click the **Remove parameter** button.
   The system asks you to confirm the action and then removes the instruction link parameter.

The following process parameters are available to configure the phase's behavior during execution:

Instruction text with links

Defines the text of the optional instruction link that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Attribute	Type	Comment
Instruction text	HTML text	Instruction text to be displayed. For any text enclosed in curly brackets you can define a hyperlink with the **Instruction link definition** process parameter (page 70). Example: Refer to {SOP1270} for guidance. Maximum length is 2000 characters (including HTML tags).

Instruction link definition

Defines all links to be available within the instruction text defined with the **Instruction text with links** process parameter (page 69). You can either access a file on the web, such as your intranet, by using the web access syntax or a file on your file system, by using the file access syntax.
Examples:

```
https://rockwellautomation.com
file:///c:/SOP/SOP1492.pdf
```

Attribute	Type	Comment
Link text	Text	Text to be used as link. For any text enclosed in curly brackets within the instruction text you can define a link with the **Link URL** attribute. Including the brackets in the link text is optional. Maximum length is 80 characters.
Link URL	Text	URL of the file to be displayed. The link opens the external application assigned to the file type by the operating system. Maximum length is 256 characters.

Instruction tables

In addition to the permanent process parameters that are always present, the **Show URL** phase provides instruction tables as optional process parameters, which you can insert if required.

You can add up to ten instruction tables with up to 50 table rows.

TIP

Instruction tables are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.

Adding instruction tables

1. Click the **Add parameter** button.
   The system opens an option list that holds all optional parameter types available for the phase.

2. Select the **Instruction Table** type.
   The system opens the **Add Instruction Table** dialog to define the instruction table's identifier.

3. Type an identifier and click the **OK** button.
   The system adds the parameter definition and the first table row to the list of parameters.
   Instruction table parameters are generally inserted below all other parameters. Where within the block of instruction table parameters the system adds a new table depends on the current selection in the Parameter Panel:

   - If no parameter is selected, the system adds the new instruction table as last parameter.
   - If an instruction table parameter is selected, the system adds the new instruction table below the selected one.
   - If any other parameter is selected, the system adds the new instruction table as first parameter of the instruction table parameter block.

> TIPS
> Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa.
>
> The identifier of the instruction table is shown as **Identifier** of the table's definition parameter. The identifiers of the individual table rows (**Row-1**, **Row-2**, etc.) are system-defined and not editable.

4. Specify the overall appearance of the table:

   - Select the number of columns to define the layout.
   - Set the width of the first column. If you do not set it to narrow, all columns have equal widths. If you set the first column to narrow, the remaining columns will have equal widths.
   - Define if your table needs to have borders. You can either show all borders of the table and its cells or none.

5. Specify the table rows and their content. The button bar above the rows table provides the following functions:

   ⊞ adds a new row to the table. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.

   🗑 deletes the currently selected rows.

   ⬆ moves the currently selected row one row up.

⬇ moves the currently selected row one row down.

> **TIP**
> Please note that the system always retains the consecutive numbering of the rows. If you reorder the table rows or delete rows, the row identifiers are updated accordingly.

Removing instruction tables

1. In the list of parameters, select the instruction table parameter you wish to remove.
2. Click the **Remove parameter** button.
   The system asks you to confirm the action and then removes the instruction table with its definition and rows.

The following process parameters are available to configure the phase's behavior during execution:

Instruction table definition

Defines the appearance of the optional instruction table that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Attribute	Type	Comment
Table layout	Choice list	Defines the layout of the instruction table holding the instruction texts. Available settings: **1 column**, **2 columns**, **3 columns**, **4 columns**, **5 columns**. Default setting: **1 column**.
First column narrow	Boolean	Defines if the first column of the table shall be narrow.
Show all borders	Boolean	Defines if the borders of the table shall be visible.

Instruction table text

Specifies the instruction texts to be displayed in the individual cells of the instruction table.

Attribute	Type	Comment
Column 1	HTML text	Instruction text to be displayed in a column. **Restriction:** Maximum length is 2000 characters (including HTML tags).
Column 2	HTML text	
Column 3	HTML text	
Column 4	HTML text	
Column 5	HTML text	

Output Variables

Instead of specifying a fixed value to be displayed or used during execution, you can also use an expression created in the Expression editor to draw the output of another phase or operation or the calculated result of several outputs as value into a parameter attribute. When you reference outputs in this manner you need to be aware of the following restrictions:

- Only when a component has been processed does it provide an output that can be fed into another component as attribute value. For this reason, you can never reference an output of a component that is a strict successor of the component in which you try to use the output.
- Branches and loops, however, require special notice in this context, since they are only potentially passed through and/or completed during processing, so their outputs are not reliably available. Thus, you can reference any such potentially available outputs, but need to be aware of the fact that the provided value may be **Undefined** so that the component into which you are feeding the output must be able to deal with such an **Undefined** input value.

The **Show URL** phase provides the following output variables:

Identifier

- Data type: String, used for displaying a pre-defined sequence of characters, such as "**Read Instruction**".
- Usage: The output variable provides the identifier of the phase.

Instance count

- Data type: Long, used for integral numbers:
  12345

- Usage: The output variable provides the count of the number of instances the phase has been processed, for example in a loop. The count is also increased when the phase is skipped from an operator's perspective, since the phase is still executed, but as a hidden phase.
  The count variable of a phase that has not been executed provides 0 as output value.

Start time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
  To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the start time of the phase.

Completion time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
  To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the completion time of the phase.

TIP

To calculate a duration from two timestamps and display it in a specific format, you need to use two conversion functions on the calculation:

- **Convert to Unitless Number** (**convertTo**) takes the calculated duration and converts it into the duration's value for one of its units (e.g., minutes or seconds).
- **Convert to String for Display** (**convertToDisplayString)** takes the converted value and displays it as string to which you can add the unit, also as string.

Example:

Sample Phase with Start time = 14-Nov-2014@10:15
Sample Phase with Completion time = 14-Nov-2014@11:47
The duration is to be displayed in minutes.

```
convertToDisplayString
  (convertTo
      ({Sample Phase}.{Completion time}-{Sample Phase}.{Start time},
"min")
  )
 + " min"
```

As result of the expression, the system displays "**92 min**".

Chapter 8

Upload Image

The **Upload image** phase allows an operator to upload an image to document processing-related information.

It can be used for processing requirements, such as:

- Recording of visual appearance during product test
  During the inspection of a product sample, a picture of the sample can be taken and uploaded.
- Supporting the operator with graphical instructions
  When the operator starts a specific processing step, the phase automatically displays the required illustration.

Execution

The **Upload image** phase loads an image file from a local directory or a network file location. It supports JPG, GIF, and PNG image formats.

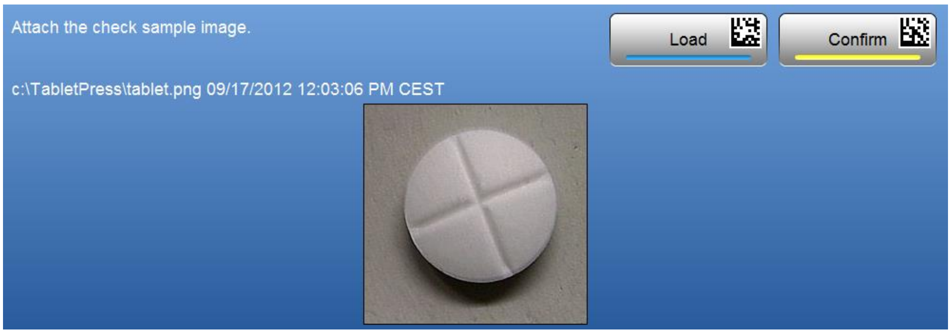
Different phase modes enable the usage in various situations that can occur during processing:

- In the **Selection** mode, the operator manually selects an image file to be uploaded.
- In the **Loading** mode, the operator triggers the upload of an already pre-defined image file.
- In the **Automatic loading** mode, the phase loads a pre-defined image file automatically.
- In the **Automatic completion** mode, the phase loads a pre-defined image file and is completed automatically without any operator interaction.

The system provides a file selection dialog to access the file system during the **Selection** mode.

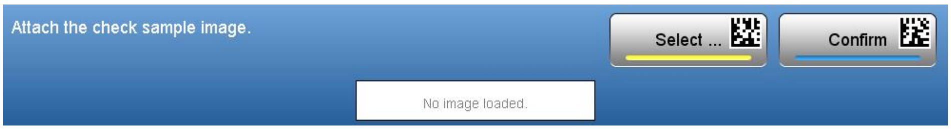
As long as the phase is active, it provides a user-triggered exception to select another image if the image loaded during processing is unsuitable for use. The system then displays the file selection dialog.
After phase completion, it provides a post-completion exception to replace the loaded image by another one. The system then displays the file selection dialog.
After completion the phase displays the full directory path of the image, the timestamp when the image was loaded, and the image itself.

The Navigator displays the file name of the image and provides access to the post-completion exception.


Attach the check sample image.

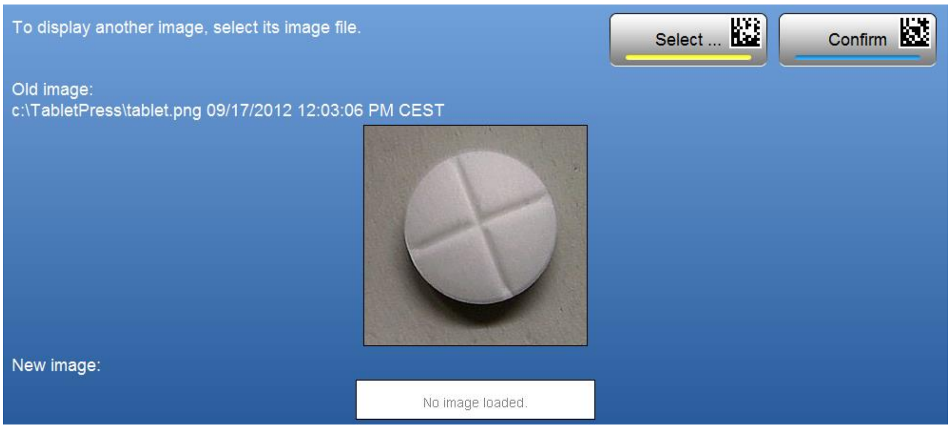
c:\TabletPress\tablet.png 09/17/2012 12:03:06 PM CEST

Figure 25: Upload image during execution - Automatic loading mode


Attach the check sample image.

No image loaded.

Figure 26: Upload image during execution - Selection mode

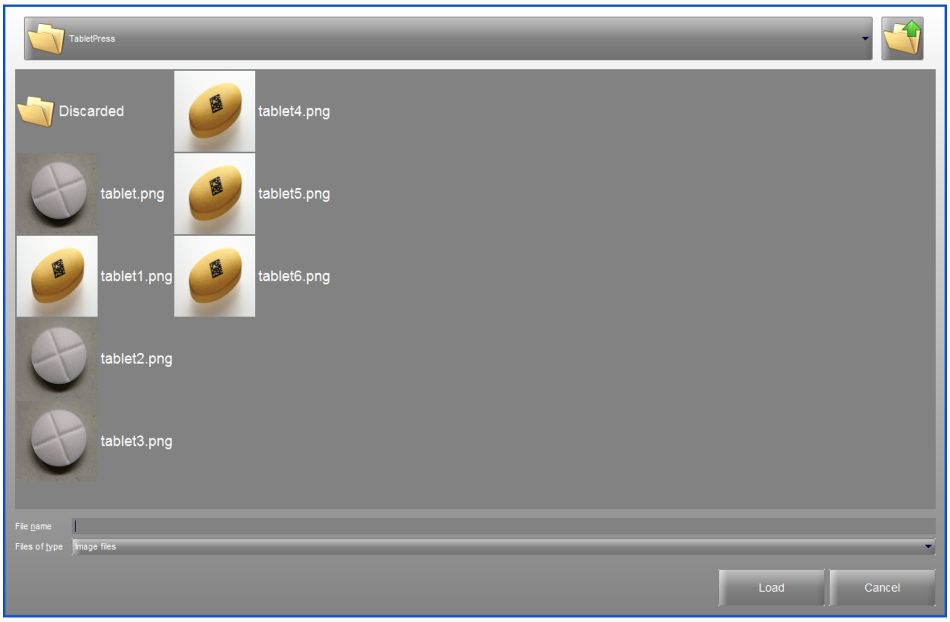

To display another image, select its image file.

Old image:
c:\TabletPress\tablet.png 09/17/2012 12:03:06 PM CEST

New image:

No image loaded.

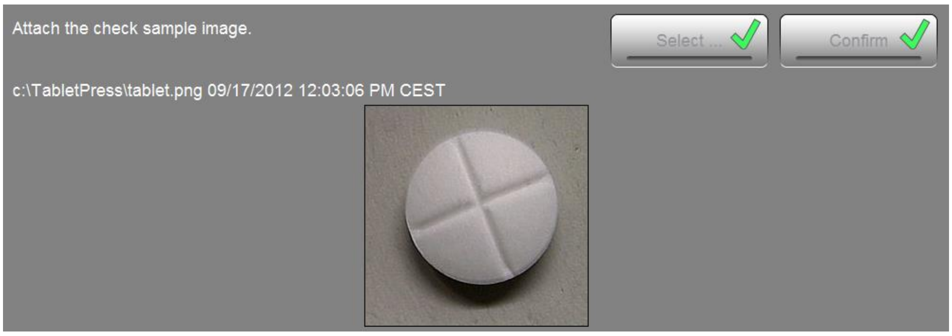
Figure 27: User-triggered exception of Upload image



File name

Files of type Image files

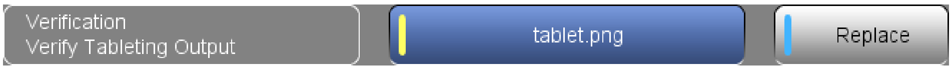
Figure 28: File selection dialog of Upload image



Attach the check sample image.

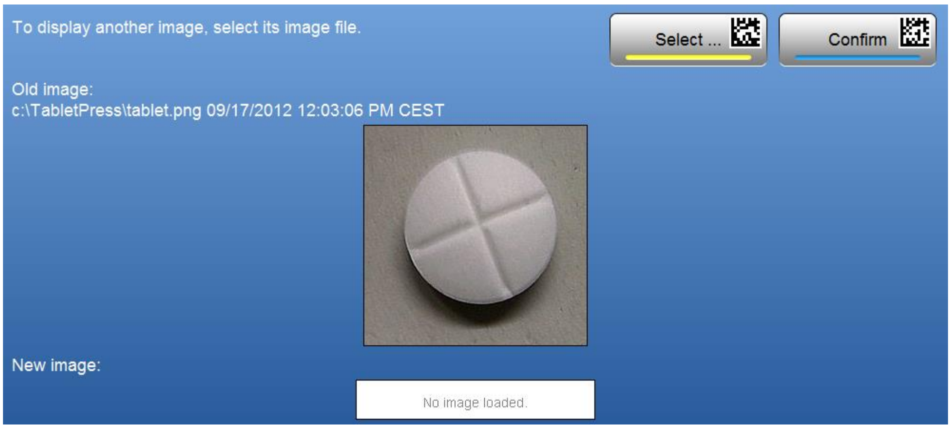
c:\TabletPress\tablet.png 09/17/2012 12:03:06 PM CEST

Figure 29: Upload image after phase completion



Verification
Verify Tableting Output

Figure 30: Upload image in the Navigator



To display another image, select its image file.

Old image:
c:\TabletPress\tablet.png 09/17/2012 12:03:06 PM CEST

New image:

No image loaded.

Figure 31: Post-completion exception of Upload image

Phase Design

The characteristics of the **Upload image** phase are defined via process parameters and their attributes.

Its user interface is designed in three columns that span two rows when the phase is active. The left column of the first row is available for instruction texts. The second column displays the **Load** or the **Select** button, which one depends on the mode defined for the phase. The rightmost column provides the **Confirm** button. The second row is dedicated to the image and its context information and thus displays a placeholder as long as no image has been loaded. Once the image has been selected or loaded automatically, it also shows the full directory path of the image file along with the timestamp when the image was loaded.

Exception handling during execution is controlled by a risk assessment classification and an exception message that are both defined by the recipe author in the exception's process parameter.

Process Parameters

The following process parameters are available to configure the phase's behavior during execution:

Instruction

Represents the instruction text that is visible on the preview, the active, and the completed view of the phase.

Attribute	Type	Comment
Column 1	HTML text	Instruction text to be displayed. Maximum length is 2000 characters (including HTML tags).
Column 2	HTML text	Not used.
Column 3	HTML text	Not used.

Mode

Defines which amount of operator interaction the phase expects during execution. The path and file definitions required for all modes except the **Selection** mode are set with the attributes of the **File location** parameter (page 79).

Attribute	Type	Comment
Mode	Choice list	Defines the processing mode. **Selection** (default): Operator selects an image file. **Loading (pre-defined file)**: Operator triggers phase to load a pre-defined image file. **Automatic loading (pre-defined file)**: Phase automatically loads a pre-defined image file. **Automatic completion (pre-defined file)**: Phase automatically loads a pre-defined image file and is completed.

File location

Defines the network or local directory the system accesses to load the image file.

Attribute	Type	Comment
Directory path	Text	Defines the directory path of the image file to be loaded. Environment variables are supported (e.g. %USERNAME%). If the **Mode** process parameter (page 78) is set to **Selection**, the system defaults the file selection dialog to this path. The operator can navigate to another directory.
File name	Text	Defines the name of the image file to be loaded. If the **Mode** process parameter (page 78) is set to **Selection**, the system uses the name as filter criterion in the **Files of type** box of the file selection dialog. Wildcards are supported to restrict the number of displayed image files. Example: img*.jpg displays any image files that start with img. For all other options available with the **Mode** process parameter (page 78), the file name must be unique.

Mandatory upload check

Defines whether an image must be loaded or not to complete the phase regularly.

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If so, the phase can only be completed after an image has been loaded or an exception has been registered.
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: **None**, **Low**, **Low (mandatory comment)**, **Medium**, **Medium (mandatory comment)**, **High**, **High (mandatory comment)**. Default setting: **High**.
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 250 characters. **Mandatory** if the **Enabled** attribute is set to **Yes**.

Select manually

Represents a user-triggered exception that is accessible from the Exception Window. The exception allows an operator to select an image manually even when the phase is set to loading a pre-defined image file.
It covers incidents when the pre-defined file is not available or unsuitable.

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: **None**, **Low**, **Low (mandatory comment)**, **Medium**, **Medium (mandatory comment)**, **High**, **High (mandatory comment)**. Default setting: **High**.
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 250 characters.

Replace file

Represents a post-completion exception that is accessible from the Navigator.
The exception allows an operator to select an image for loading after the phase has been completed.
It covers incidents when the phase is run in the **Automatic completion** mode, but the loaded image turns out to be unsuitable.

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: **None**, **Low**, **Low (mandatory comment)**, **Medium**, **Medium (mandatory comment)**, **High**, **High (mandatory comment)**. Default setting: **High**.
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 250 characters.

Instruction links

In addition to the permanent process parameters that are always present, the **Upload image** phase provides instruction links as optional process parameters, which you can insert if required.

You can add up to ten instruction link parameters.

TIP

Instruction links are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.

ADDING INSTRUCTION LINKS

1. Click the **Add parameter** button.
   The system opens an option list that holds all optional parameter types available for the phase.
2. Select the **Instruction Link** type.
   The system opens the **Add Instruction Link** dialog to define the instruction link's identifier.

3. Type an identifier and click the **OK** button.
   The system adds the instruction link parameter and the first link definition row to the list of parameters.
   Instruction link parameters are generally inserted below all other parameters.
   Where within the block of instruction link parameters the system adds a new link parameter depends on the current selection in the Parameter Panel:

   - If no parameter is selected, the system adds the new instruction link parameter as last parameter.
   - If an instruction link parameter is selected, the system adds the new instruction link parameter below the selected one.
   - If any other parameter is selected, the system adds the new instruction link parameter as first parameter of the instruction link parameter block.

   > TIPS
   > Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa.
   > The identifier of the instruction link parameter is shown as **Identifier** of the link's instruction text parameter.

4. Specify the instruction text to be displayed and mark the link texts by enclosing them in curly brackets.
5. Specify the list of link definitions. Each row of the list defines one hyperlink. The button bar above the list provides the following functions:

   ⊞ adds a new row to the list. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.

   🗑 deletes the currently selected rows.

   ⬆ moves the currently selected row one row up.

   ⬇ moves the currently selected row one row down.

Removing instruction links

1. In the list of parameters, select the instruction link parameter you wish to remove.
2. Click the **Remove parameter** button.
   The system asks you to confirm the action and then removes the instruction link parameter.

The following process parameters are available to configure the phase's behavior during execution:

Instruction text with links

Defines the text of the optional instruction link that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Attribute	Type	Comment
Instruction text	HTML text	Instruction text to be displayed. For any text enclosed in curly brackets you can define a hyperlink with the **Instruction link definition** process parameter (page 83). Example: Refer to {SOP1270} for guidance. Maximum length is 2000 characters (including HTML tags).

Instruction link definition

Defines all links to be available within the instruction text defined with the **Instruction text with links** process parameter (page 83). You can either access a file on the web, such as your intranet, by using the web access syntax or a file on your file system, by using the file access syntax.

Examples:

```
https://rockwellautomation.com
file:///c:/SOP/SOP1492.pdf
```

Attribute	Type	Comment
Link text	Text	Text to be used as link. For any text enclosed in curly brackets within the instruction text you can define a link with the **Link URL** attribute. Including the brackets in the link text is optional. Maximum length is 80 characters.
Link URL	Text	URL of the file to be displayed. The link opens the external application assigned to the file type by the operating system. Maximum length is 256 characters.

Instruction tables

In addition to the permanent process parameters that are always present, the **Upload image** phase provides instruction tables as optional process parameters, which you can insert if required.

You can add up to ten instruction tables with up to 50 table rows.

> **TIP**
>
> Instruction tables are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.

Adding instruction tables

1. Click the **Add parameter** button.
   The system opens an option list that holds all optional parameter types available for the phase.
2. Select the **Instruction Table** type.
   The system opens the **Add Instruction Table** dialog to define the instruction table's identifier.
3. Type an identifier and click the **OK** button.
   The system adds the parameter definition and the first table row to the list of parameters.
   Instruction table parameters are generally inserted below all other parameters. Where within the block of instruction table parameters the system adds a new table depends on the current selection in the Parameter Panel:
   - If no parameter is selected, the system adds the new instruction table as last parameter.
   - If an instruction table parameter is selected, the system adds the new instruction table below the selected one.
   - If any other parameter is selected, the system adds the new instruction table as first parameter of the instruction table parameter block.

> **TIPS**
>
> Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa.
>
> The identifier of the instruction table is shown as **Identifier** of the table's definition parameter. The identifiers of the individual table rows (**Row-1**, **Row-2**, etc.) are system-defined and not editable.

4. Specify the overall appearance of the table:
   - Select the number of columns to define the layout.
   - Set the width of the first column. If you do not set it to narrow, all columns have equal widths. If you set the first column to narrow, the remaining columns will have equal widths.
   - Define if your table needs to have borders. You can either show all borders of the table and its cells or none.
5. Specify the table rows and their content. The button bar above the rows table provides the following functions:

   ⊞ adds a new row to the table. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.

   🗑 deletes the currently selected rows.

   ⬆ moves the currently selected row one row up.

   ⬇ moves the currently selected row one row down.

   > **TIP**
   > Please note that the system always retains the consecutive numbering of the rows. If you reorder the table rows or delete rows, the row identifiers are updated accordingly.

REMOVING INSTRUCTION TABLES

1. In the list of parameters, select the instruction table parameter you wish to remove.
2. Click the **Remove parameter** button.
   The system asks you to confirm the action and then removes the instruction table with its definition and rows.

The following process parameters are available to configure the phase's behavior during execution:

Instruction table definition

Defines the appearance of the optional instruction table that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Attribute	Type	Comment

Attribute	Type	Comment
Table layout	Choice list	Defines the layout of the instruction table holding the instruction texts. Available settings: **1 column**, **2 columns**, **3 columns**, **4 columns**, **5 columns**. Default setting: **1 column**.
First column narrow	Boolean	Defines if the first column of the table shall be narrow.
Show all borders	Boolean	Defines if the borders of the table shall be visible.

Instruction table text

Specifies the instruction texts to be displayed in the individual cells of the instruction table.

Attribute	Type	Comment
Column 1	HTML text	Instruction text to be displayed in a column. **Restriction**: Maximum length is 2000 characters (including HTML tags).
Column 2	HTML text	
Column 3	HTML text	
Column 4	HTML text	
Column 5	HTML text	

Output Variables

Instead of specifying a fixed value to be displayed or used during execution, you can also use an expression created in the Expression editor to draw the output of another phase or operation or the calculated result of several outputs as value into a parameter attribute. When you reference outputs in this manner you need to be aware of the following restrictions:

- Only when a component has been processed does it provide an output that can be fed into another component as attribute value. For this reason, you can never reference an output of a component that is a strict successor of the component in which you try to use the output.

- Branches and loops, however, require special notice in this context, since they are only potentially passed through and/or completed during processing, so their outputs are not reliably available. Thus, you can reference any such potentially available outputs, but need to be aware of the fact that the provided value may be **Undefined** so that the component into which you are feeding the output must be able to deal with such an **Undefined** input value.

The **Upload image** phase provides the following output variables:

Image full path

- Data type: String, used for displaying a pre-defined sequence of characters, such as "**C:\Data\Execution\report.pdf**" or "**C:\Data\Processing\img.png**".
- Usage: The output variable provides the full path and file name of the uploaded image.

Image timestamp

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
  To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the modification time of the uploaded image.

Identifier

- Data type: String, used for displaying a pre-defined sequence of characters, such as "**Read Instruction**".
- Usage: The output variable provides the identifier of the phase.

Instance count

- Data type: Long, used for integral numbers:
  12345
- Usage: The output variable provides the count of the number of instances the phase has been processed, for example in a loop. The count is also increased when the phase is skipped from an operator's perspective, since the phase is still executed, but as a hidden phase.
  The count variable of a phase that has not been executed provides 0 as output value.

Start time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
  To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the start time of the phase.

Completion time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
  To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the completion time of the phase.

TIP

To calculate a duration from two timestamps and display it in a specific format, you need to use two conversion functions on the calculation:

- **Convert to Unitless Number** (**convertTo**) takes the calculated duration and converts it into the duration's value for one of its units (e.g., minutes or seconds).
- **Convert to String for Display** (**convertToDisplayString)** takes the converted value and displays it as string to which you can add the unit, also as string.

Example:

Sample Phase with Start time = 14-Nov-2014@10:15
Sample Phase with Completion time = 14-Nov-2014@11:47
The duration is to be displayed in minutes.

```
convertToDisplayString
  (convertTo
      ({Sample Phase}.{Completion time}-{Sample Phase}.{Start time},
"min")
  )
 + " min"
```

As result of the expression, the system displays "**92 min**".

Chapter 9

Upload PDF

The **Upload PDF** phase allows an operator to upload a PDF file, to display its content, and to document the content in the batch report.

It can be used for processing requirements, such as:

- Attaching documentation to the batch report
  During execution, the operator can upload a PDF file that provides information about test results of a sample.
- Supporting the operator with instructions that need to be recorded in the batch report
  When the operator starts a specific processing step, the phase automatically uploads and displays the required PDF file.

Execution

The **Upload PDF** phase loads a PDF file from a local directory or a network file location.

Different phase modes enable the usage in various situations that can occur during processing:

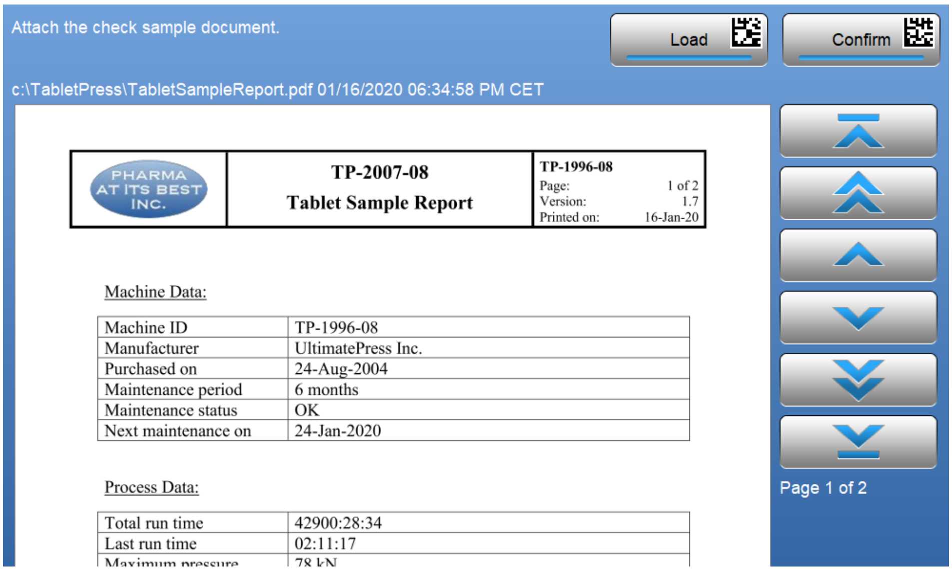
- In the **Selection** mode, the operator manually selects a PDF file to be uploaded.
- In the **Loading** mode, the operator triggers the upload of an already pre-defined PDF file.
- In the **Automatic loading** mode, the phase loads a pre-defined PDF file automatically.
- In the **Automatic completion** mode, the phase loads a pre-defined PDF file and is completed automatically without any operator interaction.

The system provides a file selection dialog to access the file system during the **Selection** mode. The dialog displays readable PDF files as thumbnails and encrypted or otherwise unsuitable PDF files with a corresponding icon.

As long as the phase is active, it provides a user-triggered exception to select another PDF file if the file loaded during processing is unsuitable for use. The system then displays the file selection dialog.
After phase completion, it provides a post-completion exception to replace the loaded PDF file with another one. The system then displays the file selection dialog.
After completion the phase displays the full directory path of the PDF, the timestamp

when the PDF was loaded, and the document itself.

The Navigator displays the file name of the PDF and provides access to the post-completion exception.



Attach the check sample document.

c:\TabletPress\TabletSampleReport.pdf 01/16/2020 06:34:58 PM CET

	TP-2007-08 Tablet Sample Report	TP-1996-08 Page: 1 of 2 Version: 1.7 Printed on: 16-Jan-20

Machine Data:

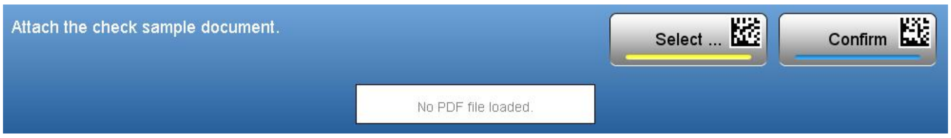
Machine ID	TP-1996-08
Manufacturer	UltimatePress Inc.
Purchased on	24-Aug-2004
Maintenance period	6 months
Maintenance status	OK
Next maintenance on	24-Jan-2020

Process Data:

Total run time	42900:28:34
Last run time	02:11:17
Maximum pressure	78 kN

Page 1 of 2

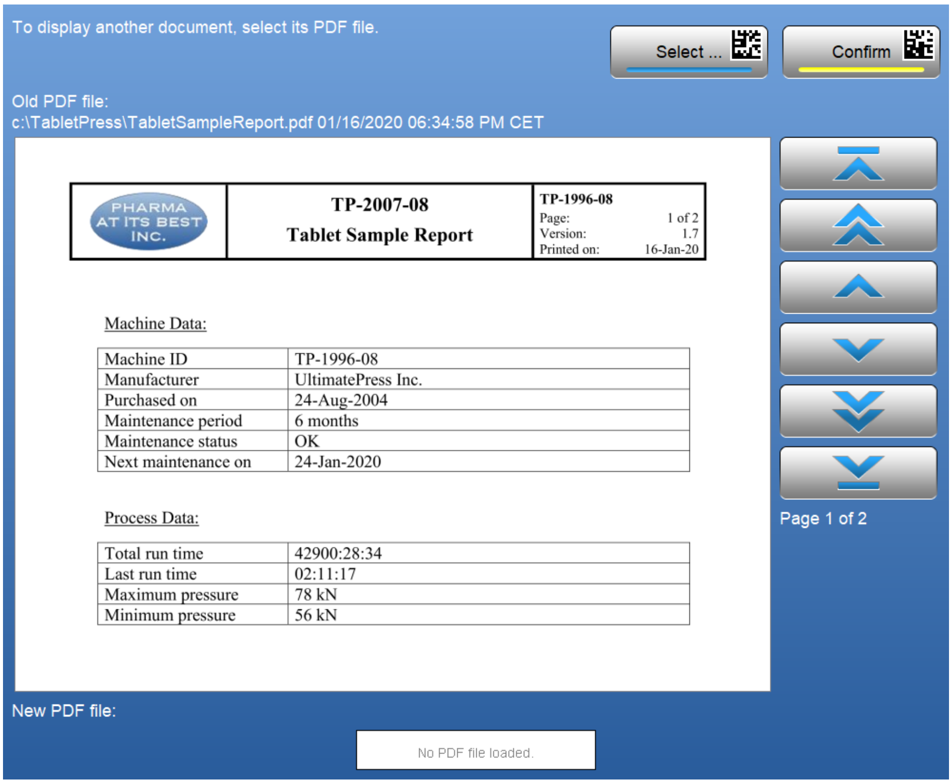
Figure 32: Upload PDF during execution - Automatic loading mode



Attach the check sample document.

No PDF file loaded.

Figure 33: Upload PDF during execution - Selection mode



To display another document, select its PDF file.

Old PDF file:
c:\TabletPress\TabletSampleReport.pdf 01/16/2020 06:34:58 PM CET

	TP-2007-08 Tablet Sample Report	TP-1996-08 Page: 1 of 2 Version: 1.7 Printed on: 16-Jan-20

Machine Data:

Machine ID	TP-1996-08
Manufacturer	UltimatePress Inc.
Purchased on	24-Aug-2004
Maintenance period	6 months
Maintenance status	OK
Next maintenance on	24-Jan-2020

Process Data:

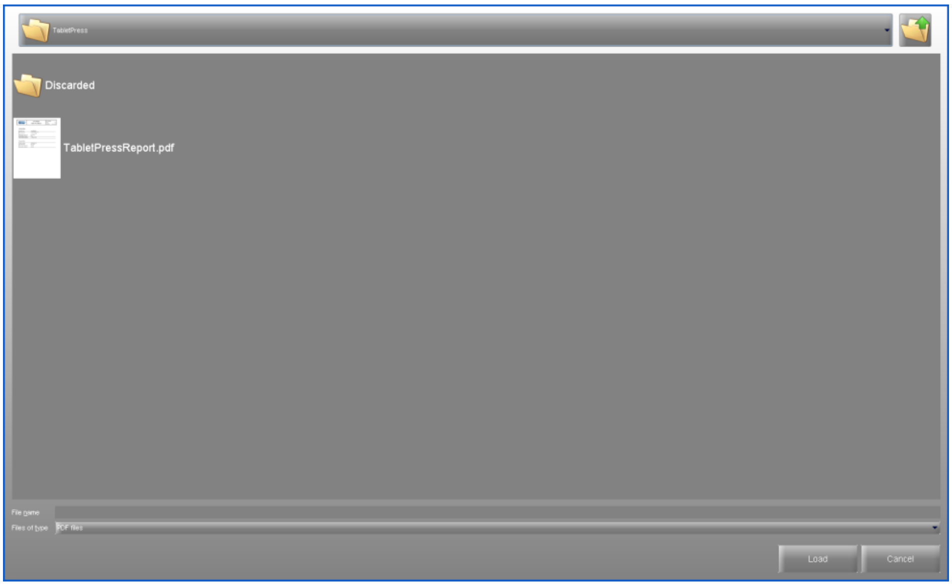
Total run time	42900:28:34
Last run time	02:11:17
Maximum pressure	78 kN
Minimum pressure	56 kN

Page 1 of 2

New PDF file:

No PDF file loaded.

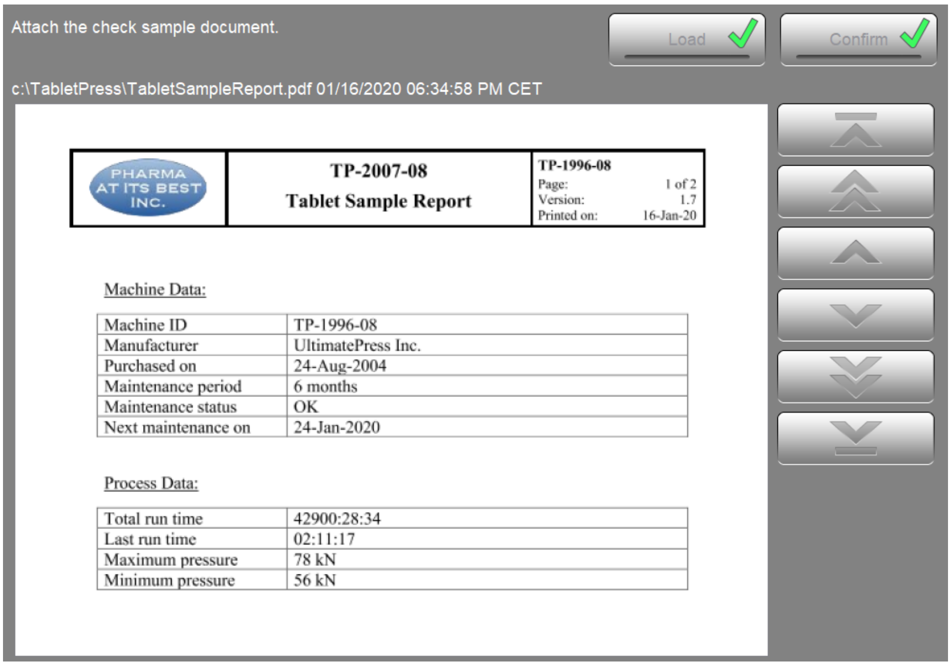
Figure 34: User-triggered exception of Upload PDF



File name

Files of type PDF files

Figure 35: File selection dialog of Upload PDF



Attach the check sample document.

c:\TabletPress\TabletSampleReport.pdf 01/16/2020 06:34:58 PM CET

PHARMA AT ITS BEST INC.	TP-2007-08 Tablet Sample Report	TP-1996-08 Page: 1 of 2 Version: 1.7 Printed on: 16-Jan-20

Machine Data:

Machine ID	TP-1996-08
Manufacturer	UltimatePress Inc.
Purchased on	24-Aug-2004
Maintenance period	6 months
Maintenance status	OK
Next maintenance on	24-Jan-2020

Process Data:

Total run time	42900:28:34
Last run time	02:11:17
Maximum pressure	78 kN
Minimum pressure	56 kN

Figure 36: Upload PDF after phase completion

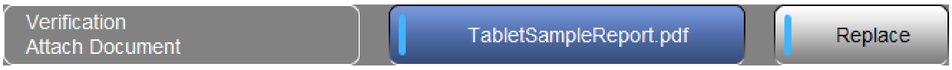
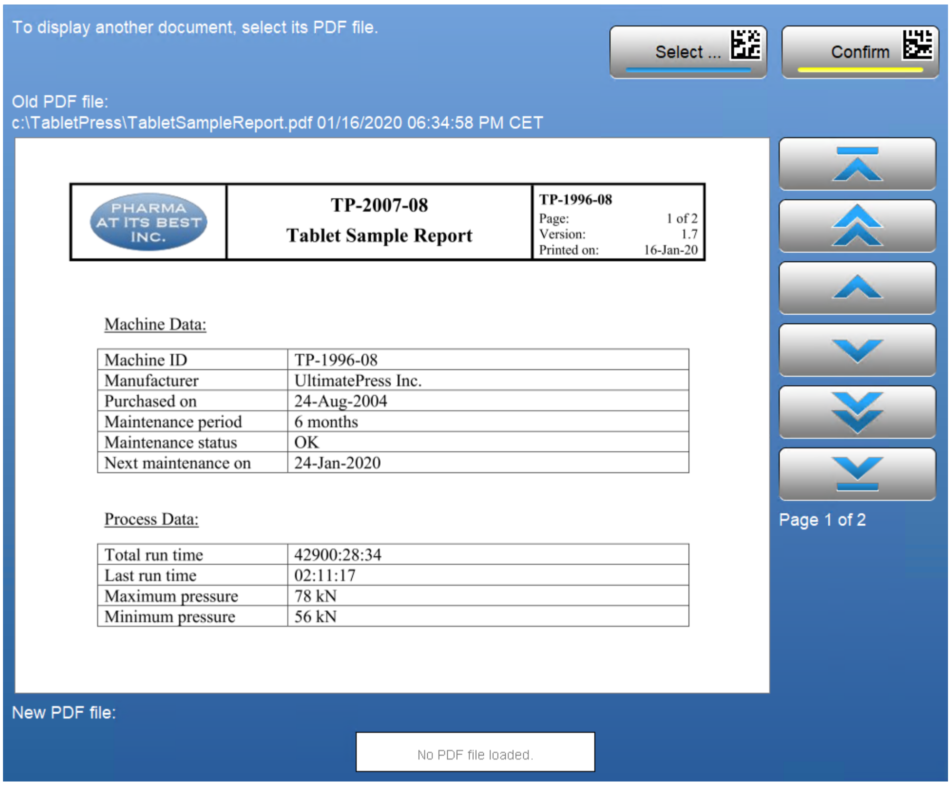


Figure 37: Upload PDF in the Navigator



To display another document, select its PDF file.

Old PDF file:
c:\TabletPress\TabletSampleReport.pdf 01/16/2020 06:34:58 PM CET

PHARMA AT ITS BEST INC.	TP-2007-08 Tablet Sample Report	TP-1996-08 Page: 1 of 2 Version: 1.7 Printed on: 16-Jan-20

Machine Data:

Machine ID	TP-1996-08
Manufacturer	UltimatePress Inc.
Purchased on	24-Aug-2004
Maintenance period	6 months
Maintenance status	OK
Next maintenance on	24-Jan-2020

Process Data:

Total run time	42900:28:34
Last run time	02:11:17
Maximum pressure	78 kN
Minimum pressure	56 kN

Page 1 of 2

New PDF file:

No PDF file loaded.

Figure 38: Post-completion exception of Upload PDF

Phase Design

The characteristics of the **Upload PDF** phase are defined via process parameters and their attributes.

Its user interface is designed in three columns that span two rows when the phase is active. The left column of the first row is available for instruction texts. The second column displays the **Load** or the **Select** button, which one depends on the mode defined for the phase. The rightmost column provides the **Confirm** button. The second row is dedicated to the PDF file and its context information and thus displays a placeholder as long as no file has been loaded. Once the PDF file has been selected or loaded automatically, it also shows the full directory path of the PDF file along with the timestamp when the file was loaded. To the right of the displayed document, the phase provides navigation buttons to page or scroll through the document: **First page**, **Previous page**, **Scroll up**, **Scroll down**, **Next page**, **Last page**. Below the document navigation buttons, it shows the page number of the currently displayed page and the total number of pages. In the completed view, only the first page of the document is shown, the navigation buttons are disabled, and there is no page number.

Exception handling during execution is controlled by a risk assessment classification and an exception message that are both defined by the recipe author in the exception's process parameter.

Process Parameters

The following process parameters are available to configure the phase's behavior during execution:

Instruction

Represents the instruction text that is visible on the preview, the active, and the completed view of the phase.

Attribute	Type	Comment
Column 1	HTML text	Instruction text to be displayed. Maximum length is 2000 characters (including HTML tags).
Column 2	HTML text	Not used.
Column 3	HTML text	Not used.

Mode

Defines which amount of operator interaction the phase expects during execution. The path and file definitions required for all modes except the **Selection** mode are set with the attributes of the **File location** parameter (page 94).

Attribute	Type	Comment
Mode	Choice list	Defines the processing mode. **Selection** (default): Operator selects a PDF file. **Loading (pre-defined file)**: Operator triggers phase to load a pre-defined PDF file. **Automatic loading (pre-defined file)**: Phase automatically loads a pre-defined PDF file. **Automatic completion (pre-defined file)**: Phase automatically loads a pre-defined PDF file and is completed.

File location

Defines the network or local directory the system accesses to load the PDF file.

Attribute	Type	Comment
Directory path	Text	Defines the directory path of the image file to be loaded. Environment variables are supported (e.g. %USERNAME%). If the **Mode** process parameter (page 94) is set to **Selection**, the system defaults the file selection dialog to this path. The operator can navigate to another directory.

Attribute	Type	Comment
File name	Text	Defines the name of the image file to be loaded. If the **Mode** process parameter (page 94) is set to **Selection**, the system uses the name as filter criterion in the **Files of type** box of the file selection dialog. Wildcards are supported to restrict the number of displayed PDF files. Example: rep*.pdf displays all PDF files starting with rep. For all other options available with the **Mode** process parameter (page 94), the file name must be unique.

Mandatory upload check

Defines whether a PDF file must be loaded or not to complete the phase regularly.

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If so, the phase can only be completed after a PDF file has been loaded or an exception has been registered.
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: **None**, **Low**, **Low (mandatory comment)**, **Medium**, **Medium (mandatory comment)**, **High**, **High (mandatory comment)**. Default setting: **High**.
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 250 characters. **Mandatory** if the **Enabled** attribute is set to **Yes**.

Select manually

Represents a user-triggered exception that is accessible from the Exception Window.
The exception allows an operator to select a PDF file manually even when the phase is set to loading a pre-defined PDF file.
It covers incidents when the pre-defined file is not available or unsuitable.

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: **None**, **Low**, **Low (mandatory comment)**, **Medium**, **Medium (mandatory comment)**, **High**, **High (mandatory comment)**. Default setting: **High**.
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 250 characters.

Replace file

Represents a post-completion exception that is accessible from the Navigator.
The exception allows an operator to select a PDF file for loading after the phase has been completed.
It covers incidents when the phase is run in the **Automatic completion** mode, but the loaded document turns out to be unsuitable.

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: **None**, **Low**, **Low (mandatory comment)**, **Medium**, **Medium (mandatory comment)**, **High**, **High (mandatory comment)**. Default setting: **High**.
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 250 characters.

Instruction links

In addition to the permanent process parameters that are always present, the **Upload PDF** phase provides instruction links as optional process parameters, which you can insert if required.

You can add up to ten instruction link parameters.

> **TIP**
>
> Instruction links are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.

Adding instruction links

1. Click the **Add parameter** button.
   The system opens an option list that holds all optional parameter types available for the phase.
2. Select the **Instruction Link** type.
   The system opens the **Add Instruction Link** dialog to define the instruction link's identifier.
3. Type an identifier and click the **OK** button.
   The system adds the instruction link parameter and the first link definition row to the list of parameters.
   Instruction link parameters are generally inserted below all other parameters.
   Where within the block of instruction link parameters the system adds a new link parameter depends on the current selection in the Parameter Panel:
   - If no parameter is selected, the system adds the new instruction link parameter as last parameter.
   - If an instruction link parameter is selected, the system adds the new instruction link parameter below the selected one.
   - If any other parameter is selected, the system adds the new instruction link parameter as first parameter of the instruction link parameter block.

   > **TIPS**
   > Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa.
   > The identifier of the instruction link parameter is shown as **Identifier** of the link's instruction text parameter.
4. Specify the instruction text to be displayed and mark the link texts by enclosing them in curly brackets.

5. Specify the list of link definitions. Each row of the list defines one hyperlink. The button bar above the list provides the following functions:

   ⊞ adds a new row to the list. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.

   🗑 deletes the currently selected rows.

   ⬆ moves the currently selected row one row up.

   ⬇ moves the currently selected row one row down.

REMOVING INSTRUCTION LINKS

1. In the list of parameters, select the instruction link parameter you wish to remove.
2. Click the **Remove parameter** button.
   The system asks you to confirm the action and then removes the instruction link parameter.

The following process parameters are available to configure the phase's behavior during execution:

Instruction text with links

Defines the text of the optional instruction link that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Attribute	Type	Comment
Instruction text	HTML text	Instruction text to be displayed. For any text enclosed in curly brackets you can define a hyperlink with the **Instruction link definition** process parameter (page 99). Example: Refer to {SOP1270} for guidance. Maximum length is 2000 characters (including HTML tags).

Instruction link definition

Defines all links to be available within the instruction text defined with the **Instruction text with links** process parameter (page 98). You can either access a file on the web, such as your intranet, by using the web access syntax or a file on your file system, by using the file access syntax.
Examples:

```
https://rockwellautomation.com
file:///c:/SOP/SOP1492.pdf
```

Attribute	Type	Comment
Link text	Text	Text to be used as link. For any text enclosed in curly brackets within the instruction text you can define a link with the **Link URL** attribute. Including the brackets in the link text is optional. Maximum length is 80 characters.
Link URL	Text	URL of the file to be displayed. The link opens the external application assigned to the file type by the operating system. Maximum length is 256 characters.

Instruction tables

In addition to the permanent process parameters that are always present, the **Upload PDF** phase provides instruction tables as optional process parameters, which you can insert if required.

You can add up to ten instruction tables with up to 50 table rows.

TIP

Instruction tables are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.

Adding instruction tables

1. Click the **Add parameter** button.
   The system opens an option list that holds all optional parameter types available for the phase.

2. Select the **Instruction Table** type.
   The system opens the **Add Instruction Table** dialog to define the instruction table's identifier.

3. Type an identifier and click the **OK** button.
   The system adds the parameter definition and the first table row to the list of parameters.
   Instruction table parameters are generally inserted below all other parameters. Where within the block of instruction table parameters the system adds a new table depends on the current selection in the Parameter Panel:

   - If no parameter is selected, the system adds the new instruction table as last parameter.
   - If an instruction table parameter is selected, the system adds the new instruction table below the selected one.
   - If any other parameter is selected, the system adds the new instruction table as first parameter of the instruction table parameter block.

TIPS
Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa.

The identifier of the instruction table is shown as **Identifier** of the table's definition parameter. The identifiers of the individual table rows (**Row-1**, **Row-2**, etc.) are system-defined and not editable.

4. Specify the overall appearance of the table:

   - Select the number of columns to define the layout.
   - Set the width of the first column. If you do not set it to narrow, all columns have equal widths. If you set the first column to narrow, the remaining columns will have equal widths.
   - Define if your table needs to have borders. You can either show all borders of the table and its cells or none.

5. Specify the table rows and their content. The button bar above the rows table provides the following functions:

   ⊞ adds a new row to the table. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.

   🗑 deletes the currently selected rows.

   ⬆ moves the currently selected row one row up.

⬇ moves the currently selected row one row down.

> TIP
> Please note that the system always retains the consecutive numbering of the rows. If you reorder the table rows or delete rows, the row identifiers are updated accordingly.

Removing instruction tables

1. In the list of parameters, select the instruction table parameter you wish to remove.
2. Click the **Remove parameter** button.
   The system asks you to confirm the action and then removes the instruction table with its definition and rows.

The following process parameters are available to configure the phase's behavior during execution:

Instruction table definition

Defines the appearance of the optional instruction table that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Attribute	Type	Comment
Table layout	Choice list	Defines the layout of the instruction table holding the instruction texts. Available settings: **1 column**, **2 columns**, **3 columns**, **4 columns**, **5 columns**. Default setting: **1 column**.
First column narrow	Boolean	Defines if the first column of the table shall be narrow.
Show all borders	Boolean	Defines if the borders of the table shall be visible.

Instruction table text

Specifies the instruction texts to be displayed in the individual cells of the instruction table.

Attribute	Type	Comment
Column 1	HTML text	Instruction text to be displayed in a column. **Restriction**: Maximum length is 2000 characters (including HTML tags).
Column 2	HTML text	
Column 3	HTML text	
Column 4	HTML text	
Column 5	HTML text	

Output Variables

Instead of specifying a fixed value to be displayed or used during execution, you can also use an expression created in the Expression editor to draw the output of another phase or operation or the calculated result of several outputs as value into a parameter attribute. When you reference outputs in this manner you need to be aware of the following restrictions:

- Only when a component has been processed does it provide an output that can be fed into another component as attribute value. For this reason, you can never reference an output of a component that is a strict successor of the component in which you try to use the output.
- Branches and loops, however, require special notice in this context, since they are only potentially passed through and/or completed during processing, so their outputs are not reliably available. Thus, you can reference any such potentially available outputs, but need to be aware of the fact that the provided value may be **Undefined** so that the component into which you are feeding the output must be able to deal with such an **Undefined** input value.

The **Upload PDF** phase provides the following output variables:

PDF full path

- Data type: String, used for displaying a pre-defined sequence of characters, such as "**C:\Data\Execution\report.pdf**" or "**C:\Data\Processing\img.png**".
- Usage: The output variable provides the full path and file name of the uploaded image.

PDF timestamp

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
  To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the modification time of the uploaded image.

Identifier

- Data type: String, used for displaying a pre-defined sequence of characters, such as "**Read Instruction**".
- Usage: The output variable provides the identifier of the phase.

Instance count

- Data type: Long, used for integral numbers:
  12345
- Usage: The output variable provides the count of the number of instances the phase has been processed, for example in a loop. The count is also increased when the phase is skipped from an operator's perspective, since the phase is still executed, but as a hidden phase.
  The count variable of a phase that has not been executed provides 0 as output value.

Start time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
  To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the start time of the phase.

Completion time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
  To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the completion time of the phase.

TIP

To calculate a duration from two timestamps and display it in a specific format, you need to use two conversion functions on the calculation:

- **Convert to Unitless Number** (**convertTo**) takes the calculated duration and converts it into the duration's value for one of its units (e.g., minutes or seconds).
- **Convert to String for Display** (**convertToDisplayString)** takes the converted value and displays it as string to which you can add the unit, also as string.

Example:

Sample Phase with Start time = 14-Nov-2014@10:15
Sample Phase with Completion time = 14-Nov-2014@11:47
The duration is to be displayed in minutes.

```
convertToDisplayString
   (convertTo
        ({Sample Phase}.{Completion time}-{Sample Phase}.{Start time},
"min")
   )
 + " min"
```

As result of the expression, the system displays "**92 min**".

Chapter 10

Create Workflow

The **Create workflow** phase allows to create a workflow during the execution of an order or workflow.

It can be used for processing requirements, such as:

- During order execution the container to be used is contaminated and needs to be cleaned. An ad-hoc workflow to clean the container is created and appended to the order.
- The flow rate of a filter is too low and the filter needs to be rinsed. An ad-hoc workflow is created to rinse the filter.

Execution

In addition to the instruction text, the **Create Workflow** phase displays the following data:

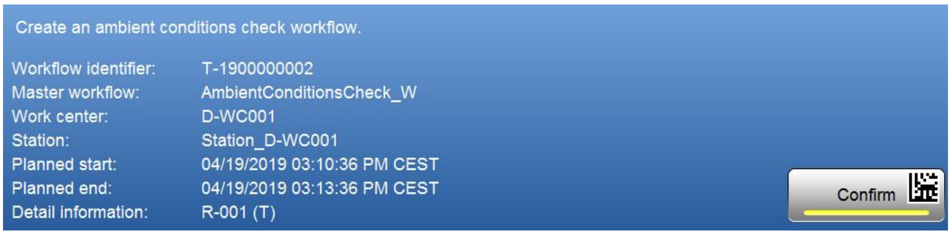
- the identifier of the master workflow to be used for generating the workflow to be executed
- the work center where the generated workflow is to be executed
- the station at which the generated workflow is to be executed
- the planned start date and time for the generated workflow
- the planned end date and time for the generated workflow
- detail information for the generated workflow.

> TIP
> Please note that planned start and end as well as the detail information are displayed as columns of the list of startable workflow steps in the **Start Workflow Processing** Selection Window of PharmaSuite for Production Execution.

The system displays only those labels and their data for which input has been defined in the respective process parameters and can already be evaluated.

The operator can view the data defined for the workflow to be created. As long as the phase is active, it provides a user-triggered exception to skip the creation of the workflow.

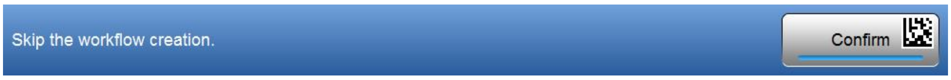
After completion the phase displays the defined data.
The navigator displays the identifier of the created workflow.



Create an ambient conditions check workflow.

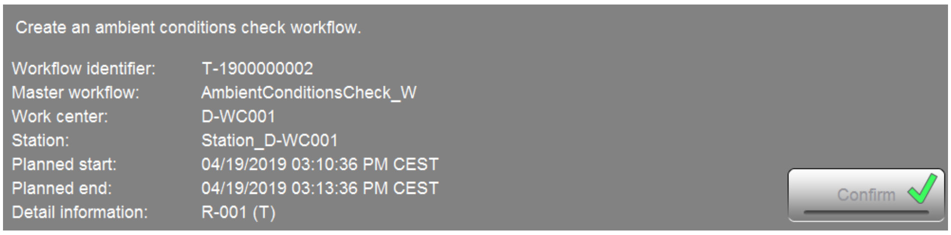
Workflow identifier: T-1900000002
Master workflow: AmbientConditionsCheck_W
Work center: D-WC001
Station: Station_D-WC001
Planned start: 04/19/2019 03:10:36 PM CEST
Planned end: 04/19/2019 03:13:36 PM CEST
Detail information: R-001 (T)

Figure 39: Create workflow during execution



Skip the workflow creation.

Figure 40: User-triggered exception of Create workflow



Create an ambient conditions check workflow.

Workflow identifier: T-1900000002
Master workflow: AmbientConditionsCheck_W
Work center: D-WC001
Station: Station_D-WC001
Planned start: 04/19/2019 03:10:36 PM CEST
Planned end: 04/19/2019 03:13:36 PM CEST
Detail information: R-001 (T)

Figure 41: Create workflow after phase completion

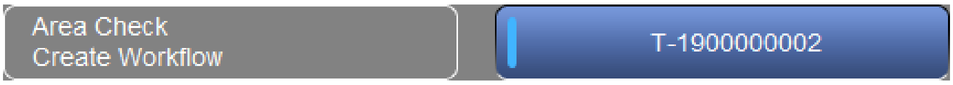


Figure 42: Create workflow in the Navigator

Phase Design

The characteristics of the **Create workflow** phase are defined via process parameters and their attributes.
Its user interface is designed in three columns that span several rows. When the phase is active, the merged columns of the first row provide space for textual instructions.
In the following rows of the left and center columns, the phase displays data defined for the workflow.
The right column provides the **Confirm** button.

When the phase is completed, it shows the same three-column, multi-row layout. Exception handling during execution is controlled by a risk assessment classification and an exception message that are both defined by the recipe author in the exception's process parameter.

Process Parameters

The following process parameters are available to configure the phase's behavior during execution:

Instruction

Represents the instruction text that is visible on the preview, the active, and the completed view of the phase.

Attribute	Type	Comment
Text	HTML text	Instruction text to be displayed. Maximum length is 2000 characters (including HTML tags).

Mode

Defines if the phase expects operator interaction during execution.

Attribute	Type	Comment
Mode	Choice list	Defines the processing mode. **Manual completion** (default): Operator confirms the phase. **Automatic completion**: Phase automatically tries to confirm the phase.

Workflow definition

Defines the identifier of the workflow. You can reference the output of a preceding phase as identifier or use the **Create Workflow ID** function of the Expression editor.

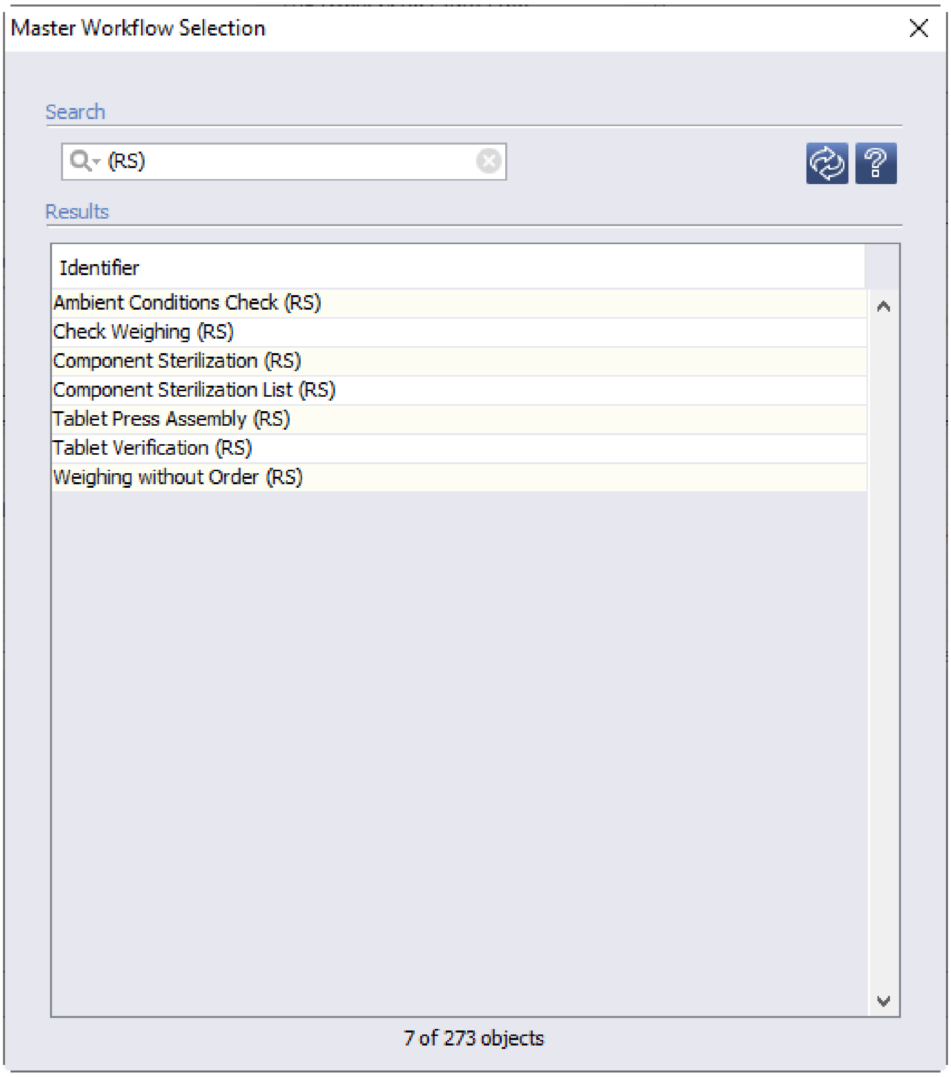
Attribute	Type	Comment
Identifier	String	Optional parameter to define a unique workflow identifier for the workflow to be created. If empty, a workflow identifier with the configured default prefix is generated. Maximum length is 12 characters.

Master workflow

Defines which master workflow is to be used. A master workflow has to be either **Scheduled** or **Approved** to be available for selection. During execution, the system will use the currently approved version of the master workflow for generating the new workflow.

Attribute	Type	Comment
Identifier	String	Defines the master workflow identifier without version to be used for generating the workflow. Maximum length is 50 characters.

For selecting the master workflow from the list of available master workflows, the system provides a Master Workflow Selection editor.



Master Workflow Selection

Search

(RS)

Results

Identifier
Ambient Conditions Check (RS)
Check Weighing (RS)
Component Sterilization (RS)
Component Sterilization List (RS)
Tablet Press Assembly (RS)
Tablet Verification (RS)
Weighing without Order (RS)

7 of 273 objects

Figure 43: Master Workflow Selection editor

Append workflow

Defines if the system automatically appends the workflow to the currently active unit procedure of the order from which it is created. If enabled, it prevents the system from presenting this question when the workflow is started.

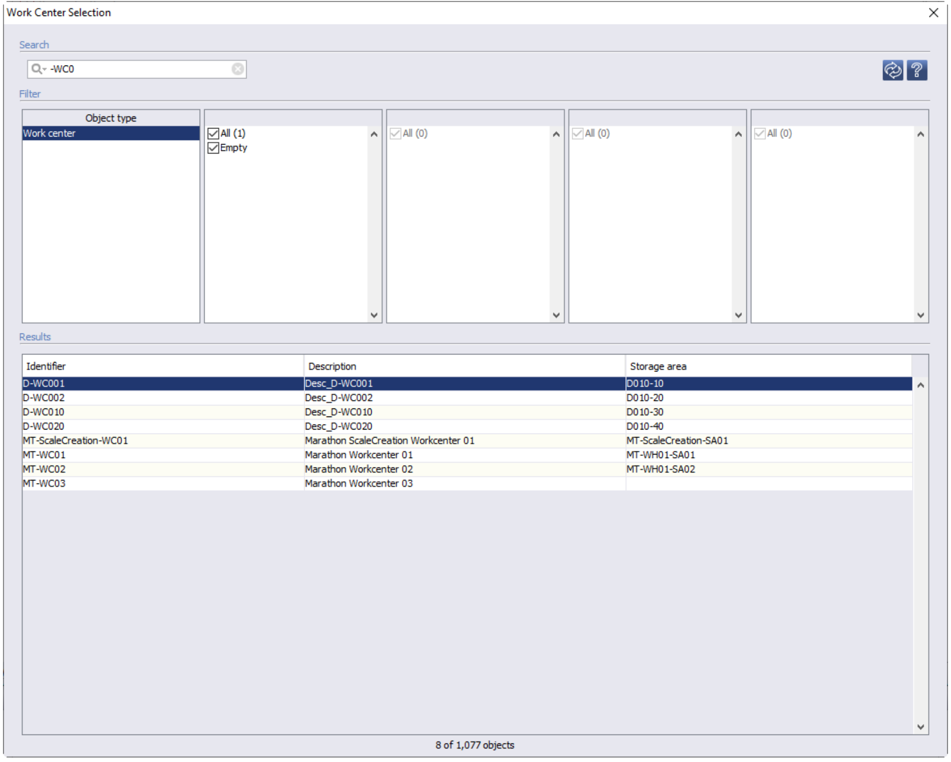
Attribute	Type	Comment
Enabled	Flag	Controls if the created workflow is automatically appended to the current unit procedure. If the phase runs in the context of a workflow, the parameter is ignored. Default setting: **Yes**

Work center

Defines the work center at which the workflow is to be executed.

Attribute	Type	Comment
Identifier	String	Optional parameter to define the work center to which the created workflow is dispatched. Maximum length is 20 characters.

For selecting the work center from the list of available work centers, the system provides a Work Center Selection editor.



Work Center Selection

Search: -WC0

Filter

Object type: Work center

☑ All (1)
☑ Empty

☑ All (0)

☑ All (0)

☑ All (0)

Results

Identifier	Description	Storage area
D-WC001	Desc_D-WC001	D010-10
D-WC002	Desc_D-WC002	D010-20
D-WC010	Desc_D-WC010	D010-30
D-WC020	Desc_D-WC020	D010-40
MT-ScaleCreation-WC01	Marathon ScaleCreation Workcenter 01	MT-ScaleCreation-SA01
MT-WC01	Marathon Workcenter 01	MT-WH01-SA01
MT-WC02	Marathon Workcenter 02	MT-WH01-SA02
MT-WC03	Marathon Workcenter 03	

8 of 1,077 objects

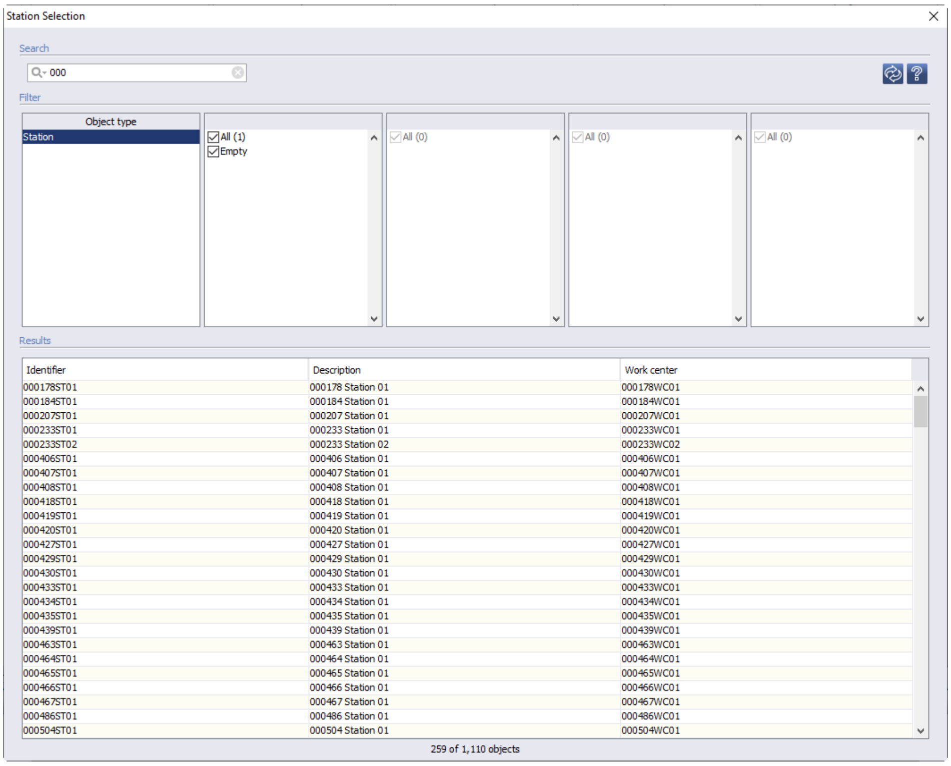
Figure 44: Work Center Selection editor

Station

Defines the station at which the workflow is to be executed.

Attribute	Type	Comment
Identifier	String	Optional parameter to define the station to which the created workflow is dispatched. Maximum length is 20 characters.

For selecting the station from the list of available work centers, the system provides a Station Selection editor.



Identifier	Description	Work center
000178ST01	000178 Station 01	000178WC01
000184ST01	000184 Station 01	000184WC01
000207ST01	000207 Station 01	000207WC01
000233ST01	000233 Station 01	000233WC01
000233ST02	000233 Station 02	000233WC02
000406ST01	000406 Station 01	000406WC01
000407ST01	000407 Station 01	000407WC01
000408ST01	000408 Station 01	000408WC01
000418ST01	000418 Station 01	000418WC01
000419ST01	000419 Station 01	000419WC01
000420ST01	000420 Station 01	000420WC01
000427ST01	000427 Station 01	000427WC01
000429ST01	000429 Station 01	000429WC01
000430ST01	000430 Station 01	000430WC01
000433ST01	000433 Station 01	000433WC01
000434ST01	000434 Station 01	000434WC01
000435ST01	000435 Station 01	000435WC01
000439ST01	000439 Station 01	000439WC01
000463ST01	000463 Station 01	000463WC01
000464ST01	000464 Station 01	000464WC01
000465ST01	000465 Station 01	000465WC01
000466ST01	000466 Station 01	000466WC01
000467ST01	000467 Station 01	000467WC01
000486ST01	000486 Station 01	000486WC01
000504ST01	000504 Station 01	000504WC01

259 of 1,110 objects

Figure 45: Station Selection editor

Start workflow automatically

Defines if the system automatically starts the workflow.

TIP

Please note that if no definitions were made for either the **Work center** parameter (page 110) or the **Station** parameter (page 110), the system starts the workflow at the current work center.

Attribute	Type	Comment
Enabled	Flag	Controls if the created workflow will be available in the Cockpit. Default setting: **No**

Planned start

The planned start date and time is displayed as column of the list of startable workflow steps in the **Start Workflow Processing** Selection Window of PharmaSuite for Production Execution, unless the workflow is set to start automatically (page 111).

Attribute	Type	Comment
Timestamp	Timestamp	Optional information.

Planned end

The planned end date and time is displayed as column of the list of startable workflow steps in the **Start Workflow Processing** Selection Window of PharmaSuite for Production Execution, unless the workflow is set to start automatically (page 111).

Attribute	Type	Comment
Timestamp	Timestamp	Optional information.

Detail information

The detail information is displayed as column of the list of startable workflow steps in the **Start Workflow Processing** Selection Window of PharmaSuite for Production Execution, unless the workflow is set to start automatically (page 111).

Attribute	Type	Comment
Value	String	Optional information.

Skip workflow creation

Represents a user-triggered exception that is accessible from the Exception Window. The exception allows an operator to confirm the phase without creating a workflow. It covers incidents when the processing situation on the shop floor does not require the planned workflow to be created.

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: **None**, **Low**, **Low (mandatory comment)**, **Medium**, **Medium (mandatory comment)**, **High**, **High (mandatory comment)**. Default setting: **High**.
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 250 characters.

Instruction links

In addition to the permanent process parameters that are always present, the **Create workflow** phase provides instruction links as optional process parameters, which you can insert if required.

You can add up to ten instruction link parameters.

> **TIP**
>
> Instruction links are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.

Adding instruction links

1. Click the **Add parameter** button.
   The system opens an option list that holds all optional parameter types available for the phase.
2. Select the **Instruction Link** type.
   The system opens the **Add Instruction Link** dialog to define the instruction link's identifier.
3. Type an identifier and click the **OK** button.
   The system adds the instruction link parameter and the first link definition row to the list of parameters.
   Instruction link parameters are generally inserted below all other parameters.
   Where within the block of instruction link parameters the system adds a new link parameter depends on the current selection in the Parameter Panel:
   - If no parameter is selected, the system adds the new instruction link parameter as last parameter.
   - If an instruction link parameter is selected, the system adds the new instruction link parameter below the selected one.
   - If any other parameter is selected, the system adds the new instruction link parameter as first parameter of the instruction link parameter block.

   > **TIPS**
   >
   > Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa.
   > The identifier of the instruction link parameter is shown as **Identifier** of the link's instruction text parameter.
4. Specify the instruction text to be displayed and mark the link texts by enclosing them in curly brackets.

5. Specify the list of link definitions. Each row of the list defines one hyperlink. The button bar above the list provides the following functions:

   ⊞ adds a new row to the list. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.

   🗑 deletes the currently selected rows.

   ⬆ moves the currently selected row one row up.

   ⬇ moves the currently selected row one row down.

REMOVING INSTRUCTION LINKS

1. In the list of parameters, select the instruction link parameter you wish to remove.
2. Click the **Remove parameter** button.
   The system asks you to confirm the action and then removes the instruction link parameter.

The following process parameters are available to configure the phase's behavior during execution:

Instruction text with links

Defines the text of the optional instruction link that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Attribute	Type	Comment
Instruction text	HTML text	Instruction text to be displayed. For any text enclosed in curly brackets you can define a hyperlink with the **Instruction link definition** process parameter (page 115). Example: Refer to {SOP1270} for guidance. Maximum length is 2000 characters (including HTML tags).

Instruction link definition

Defines all links to be available within the instruction text defined with the **Instruction text with links** process parameter (page 114). You can either access a file on the web, such as your intranet, by using the web access syntax or a file on your file system, by using the file access syntax.

Examples:

```
https://rockwellautomation.com
file:///c:/SOP/SOP1492.pdf
```

Attribute	Type	Comment
Link text	Text	Text to be used as link. For any text enclosed in curly brackets within the instruction text you can define a link with the **Link URL** attribute. Including the brackets in the link text is optional. Maximum length is 80 characters.
Link URL	Text	URL of the file to be displayed. The link opens the external application assigned to the file type by the operating system. Maximum length is 256 characters.

Instruction tables

In addition to the permanent process parameters that are always present, the **Create workflow** phase provides instruction tables as optional process parameters, which you can insert if required.

You can add up to ten instruction tables with up to 50 table rows.

TIP

Instruction tables are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.

Adding instruction tables

1. Click the **Add parameter** button.
   The system opens an option list that holds all optional parameter types available for the phase.

2. Select the **Instruction Table** type.
   The system opens the **Add Instruction Table** dialog to define the instruction table's identifier.
3. Type an identifier and click the **OK** button.
   The system adds the parameter definition and the first table row to the list of parameters.
   Instruction table parameters are generally inserted below all other parameters. Where within the block of instruction table parameters the system adds a new table depends on the current selection in the Parameter Panel:
   - If no parameter is selected, the system adds the new instruction table as last parameter.
   - If an instruction table parameter is selected, the system adds the new instruction table below the selected one.
   - If any other parameter is selected, the system adds the new instruction table as first parameter of the instruction table parameter block.

TIPS

Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa.

The identifier of the instruction table is shown as **Identifier** of the table's definition parameter. The identifiers of the individual table rows (**Row-1**, **Row-2**, etc.) are system-defined and not editable.

4. Specify the overall appearance of the table:
   - Select the number of columns to define the layout.
   - Set the width of the first column. If you do not set it to narrow, all columns have equal widths. If you set the first column to narrow, the remaining columns will have equal widths.
   - Define if your table needs to have borders. You can either show all borders of the table and its cells or none.
5. Specify the table rows and their content. The button bar above the rows table provides the following functions:

   ⊞ adds a new row to the table. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.

   🗑 deletes the currently selected rows.

   ⬆ moves the currently selected row one row up.

⬇ moves the currently selected row one row down.

> **TIP**
> Please note that the system always retains the consecutive numbering of the rows. If you reorder the table rows or delete rows, the row identifiers are updated accordingly.

Removing instruction tables

1. In the list of parameters, select the instruction table parameter you wish to remove.
2. Click the **Remove parameter** button.
   The system asks you to confirm the action and then removes the instruction table with its definition and rows.

The following process parameters are available to configure the phase's behavior during execution:

Instruction table definition

Defines the appearance of the optional instruction table that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Attribute	Type	Comment
Table layout	Choice list	Defines the layout of the instruction table holding the instruction texts. Available settings: **1 column**, **2 columns**, **3 columns**, **4 columns**, **5 columns**. Default setting: **1 column**.
First column narrow	Boolean	Defines if the first column of the table shall be narrow.
Show all borders	Boolean	Defines if the borders of the table shall be visible.

Instruction table text

Specifies the instruction texts to be displayed in the individual cells of the instruction table.

Attribute	Type	Comment
Column 1	HTML text	Instruction text to be displayed in a column. **Restriction**: Maximum length is 2000 characters (including HTML tags).
Column 2	HTML text	
Column 3	HTML text	
Column 4	HTML text	
Column 5	HTML text	

Output Variables

Instead of specifying a fixed value to be displayed or used during execution, you can also use an expression created in the Expression editor to draw the output of another phase or operation or the calculated result of several outputs as value into a parameter attribute. When you reference outputs in this manner you need to be aware of the following restrictions:

- Only when a component has been processed does it provide an output that can be fed into another component as attribute value. For this reason, you can never reference an output of a component that is a strict successor of the component in which you try to use the output.
- Branches and loops, however, require special notice in this context, since they are only potentially passed through and/or completed during processing, so their outputs are not reliably available. Thus, you can reference any such potentially available outputs, but need to be aware of the fact that the provided value may be **Undefined** so that the component into which you are feeding the output must be able to deal with such an **Undefined** input value.

The **Create workflow** phase provides the following output variables:

Workflow identifier

- Data type: String, used for displaying a pre-defined sequence of characters, such as "**SL0002854**" or "**0010**".

- Usage: The output variable provides the identifier of the created workflow taken from the **Workflow definition** process parameter (page 108) or the generated default value.

Detail information

- Data type: String, used for displaying a pre-defined sequence of characters, such as "**SL0002854**" or "**0010**".
- Usage: The output variable provides the detail information about the workflow taken from the **Detail information** process parameter (page 112).

Planned start

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
  To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the planned start time of the workflow taken from the **Planned start** process parameter (page 111).

Planned end

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
  To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the planned end time of the workflow taken from the **Planned end** process parameter (page 112).

Creation result

- Data type: String, used for displaying a pre-defined sequence of characters, such as "**GOOD**" or "**PLANNED**".
- Usage: The output variable states if a workflow was created or the phase was skipped.
  - The value is `CREATED` if the creation of the workflow was successful.
  - The value is `SKIPPED` if no workflow was created and the phase was skipped by the **Skip workflow creation** user-triggered exception.

Identifier

- Data type: String, used for displaying a pre-defined sequence of characters, such as "**Read Instruction**".
- Usage: The output variable provides the identifier of the phase.

Instance count

- Data type: Long, used for integral numbers:
  12345
- Usage: The output variable provides the count of the number of instances the phase has been processed, for example in a loop. The count is also increased when the phase is skipped from an operator's perspective, since the phase is still executed, but as a hidden phase.
  The count variable of a phase that has not been executed provides 0 as output value.

Start time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
  To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the start time of the phase.

Completion time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
  To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the completion time of the phase.

TIP

To calculate a duration from two timestamps and display it in a specific format, you need to use two conversion functions on the calculation:

- **Convert to Unitless Number** (**convertTo**) takes the calculated duration and converts it into the duration's value for one of its units (e.g., minutes or seconds).
- **Convert to String for Display** (**convertToDisplayString)** takes the converted value and displays it as string to which you can add the unit, also as string.

Example:

Sample Phase with Start time = 14-Nov-2014@10:15
Sample Phase with Completion time = 14-Nov-2014@11:47
The duration is to be displayed in minutes.

```
convertToDisplayString
  (convertTo
      ({Sample Phase}.{Completion time}-{Sample Phase}.{Start time},
"min")
  )
 + " min"
```

As result of the expression, the system displays "**92 min**".

Chapter 11

Write Context Data

The **Write context data** phase allows to store processing data from an order or workflow for later use.

It can be used for processing requirements, such as:

- Store the equipment identifier of a filter to allow a filter test workflow to check that it runs the test against the planned filter.
- Store and increase a counter value in a loop for a number of used containers until a planned number of containers per type has been filled.

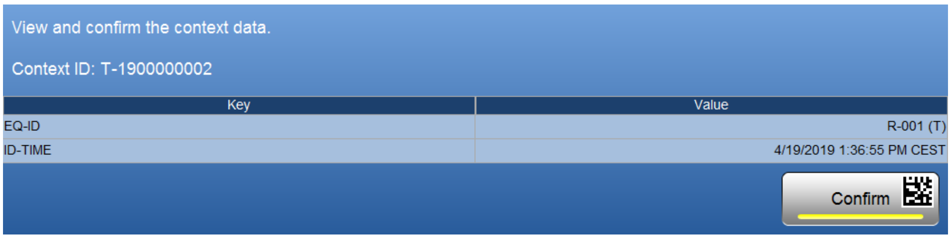
Execution

In addition to the instruction text, the **Write context data** phase displays the following data:

- the identifier of the context to which the context data refers, such as a workflow identifier
- a table that lists the individual key and value pairs of the data to be written.

The operator can view the data to be written. As long as the phase is active, it provides a user-triggered exception to override the values defined for the individual keys.

After completion, the phase displays the data written to the context.
The Navigator displays the identifier of the context.

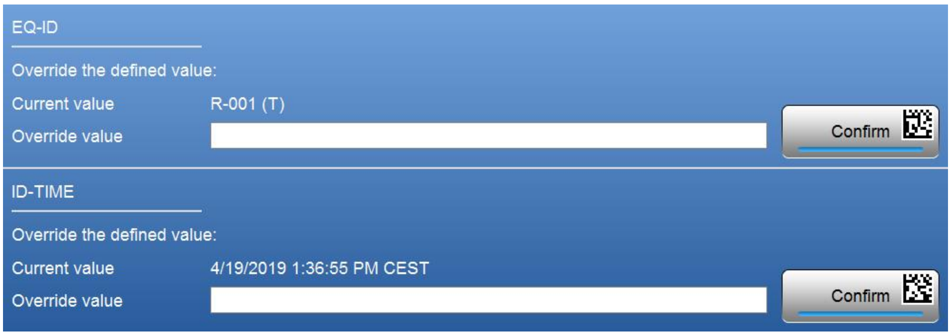


View and confirm the context data.

Context ID: T-1900000002

Key	Value
EQ-ID	R-001 (T)
ID-TIME	4/19/2019 1:36:55 PM CEST

Figure 46: Write context data during execution



EQ-ID

Override the defined value:

Current value R-001 (T)

Override value

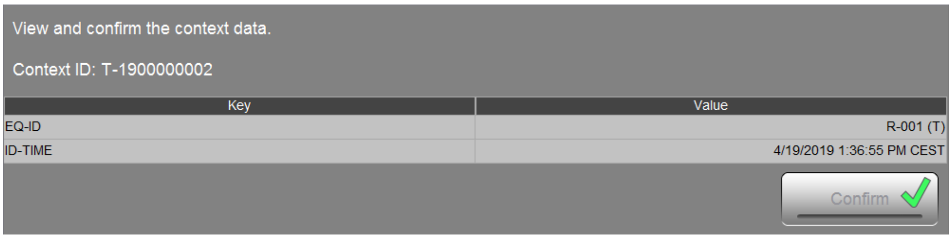
ID-TIME

Override the defined value:

Current value 4/19/2019 1:36:55 PM CEST

Override value

Figure 47: User-triggered exception of Write context data

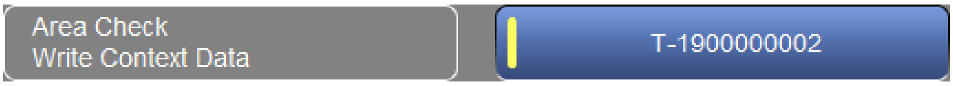


View and confirm the context data.

Context ID: T-1900000002

Key	Value
EQ-ID	R-001 (T)
ID-TIME	4/19/2019 1:36:55 PM CEST

Figure 48: Write context data after phase completion



Area Check
Write Context Data

T-1900000002

Figure 49: Write context data in the Navigator

Phase Design

The characteristics of the **Write context data** phase are defined via process parameters and their attributes.
Its user interface is designed in three columns that span several rows. When the phase is active, the merged columns of the first row provide space for textual instructions.
In the following row, the phase displays the context identifier to which the data refers.
The next rows display a tabular view of key and value pairs of the data, spanning all columns.
The right column of the bottom row contains the **Confirm** button.

When the phase is completed, it shows the same three-column, multi-row layout.
Exception handling during execution is controlled by a risk assessment classification and an exception message that are both defined by the recipe author in the exception's process parameter.

Process Parameters

The following process parameters are available to configure the phase's behavior during execution:

Instruction

Represents the instruction text that is visible on the preview, the active, and the completed view of the phase.

Attribute	Type	Comment
Text	HTML text	Instruction text to be displayed. Maximum length is 2000 characters (including HTML tags).

Mode

Defines if the phase expects operator interaction during execution.

Attribute	Type	Comment
Mode	Choice list	Defines the processing mode. **Manual completion** (default): Operator confirms the phase. **Automatic completion**: Phase is automatically completed after the data has been written.

Context definition

Defines the identifier of the context to which the context data refers, such as the identifier of a workflow. You can reference the output of a preceding phase or, in the case of a workflow, use the **Create Workflow ID** function provided by the Expression editor to generate a workflow identifier.

Attribute	Type	Comment
Identifier	String	Defines the context identifier for all key/value pairs. Maximum length is 250 characters.

Override value

Represents a user-triggered exception that is accessible from the Exception Window. The exception allows an operator to override a value defined with a key and value pair of a bundle parameter.
It covers incidents when a defined value is unsuitable and needs to be adjusted.

Attribute	Type	Comment

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: **None**, **Low**, **Low (mandatory comment)**, **Medium**, **Medium (mandatory comment)**, **High**, **High (mandatory comment)**. Default setting: **High**.
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 250 characters.

Context and key check

Represents a system-triggered exception that is displayed in the Exception Window to record incidents when for the context identifier and key there is already a data set, which would consequently be updated.

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. Default setting: **Yes**
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: **None**, **Low**, **Low (mandatory comment)**, **Medium**, **Medium (mandatory comment)**, **High**, **High (mandatory comment)**. Default setting: **High**.
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 250 characters.

Parameter bundles

In addition to the permanent process parameters that are always present, the **Write context data** phase provides parameter bundles as optional process parameters, which you can insert if required.

You can add process parameter bundles for up to 50 values of seven different data types (**BigDecimal**, **Boolean**, **Duration**, **Long**, **MeasuredValue**, **String**, **Timestamp**) to the **Write context data** phase.

Adding parameter bundles

1. Click the **Add parameter** button.
   The system opens an option list that holds all data types available for the value.
2. Select the type.
   The system opens the **Add <Data Type>** dialog to define the value's identifier.
3. Type an identifier and click the **OK** button.
   The system adds all process parameters of the bundle to the list of parameters.

Removing parameter bundles

1. In the list of parameters, select the header row that contains the identifier of the bundle you wish to remove.
2. Click the **Remove parameter** button.
   The system asks you to confirm the action and then removes the value bundle.

The following process parameters are available to configure the phase's behavior during execution:

BigDecimal - Master (bundle identifier)

Defines the key and BigDecimal value pair to be stored for the context.

Attribute	Type	Comment
Key	String	Defines the key that can be used with the context identifier to retrieve the value. Maximum length is 250 characters.
Value	BigDecimal	Defines the value for the key and context identifier.

Boolean - Master (bundle identifier)

Defines the key and Boolean value pair to be stored for the context.

Attribute	Type	Comment
Key	String	Defines the key that can be used with the context identifier to retrieve the value. Maximum length is 250 characters.
Value	Boolean	Defines the value for the key and context identifier.

Duration - Master (bundle identifier)

Defines the key and Duration value pair to be stored for the context.

Attribute	Type	Comment
Key	String	Defines the key that can be used with the context identifier to retrieve the value. Maximum length is 250 characters.
Value	Duration	Defines the value for the key and context identifier.

Long - Master (bundle identifier)

Defines the key and Long value pair to be stored for the context.

Attribute	Type	Comment
Key	String	Defines the key that can be used with the context identifier to retrieve the value. Maximum length is 250 characters.
Value	Long	Defines the value for the key and context identifier.

MeasuredValue - Master (bundle identifier)

Defines the key and MeasuredValue value pair to be stored for the context.

Attribute	Type	Comment
Key	String	Defines the key that can be used with the context identifier to retrieve the value. Maximum length is 250 characters.
Value	MeasuredValue	Defines the value for the key and context identifier.

String - Master (bundle identifier)

Defines the key and String value pair to be stored for the context.

Attribute	Type	Comment
Key	String	Defines the key that can be used with the context identifier to retrieve the value. Maximum length is 250 characters.
Value	String	Defines the value for the key and context identifier.

Timestamp - Master (bundle identifier)

Defines the key and Timestamp value pair to be stored for the context.

Attribute	Type	Comment
Key	String	Defines the key that can be used with the context identifier to retrieve the value. Maximum length is 250 characters.
Value	Timestamp	Defines the value for the key and context identifier.

Instruction links

In addition to the permanent process parameters that are always present, the **Write context data** phase provides instruction links as optional process parameters, which you can insert if required.

You can add up to ten instruction link parameters.

TIP

Instruction links are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.

➕ Adding instruction links

1. Click the **Add parameter** button.
   The system opens an option list that holds all optional parameter types available for the phase.

2. Select the **Instruction Link** type.
   The system opens the **Add Instruction Link** dialog to define the instruction link's identifier.

3. Type an identifier and click the **OK** button.
   The system adds the instruction link parameter and the first link definition row to the list of parameters.
   Instruction link parameters are generally inserted below all other parameters. Where within the block of instruction link parameters the system adds a new link parameter depends on the current selection in the Parameter Panel:

   - If no parameter is selected, the system adds the new instruction link parameter as last parameter.
   - If an instruction link parameter is selected, the system adds the new instruction link parameter below the selected one.
   - If any other parameter is selected, the system adds the new instruction link parameter as first parameter of the instruction link parameter block.

   TIPS
   Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa.
   The identifier of the instruction link parameter is shown as **Identifier** of the link's instruction text parameter.

4. Specify the instruction text to be displayed and mark the link texts by enclosing them in curly brackets.

5. Specify the list of link definitions. Each row of the list defines one hyperlink. The button bar above the list provides the following functions:

   ➕ adds a new row to the list. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.

🗑 deletes the currently selected rows.

⬆ moves the currently selected row one row up.

⬇ moves the currently selected row one row down.

Removing instruction links

1. In the list of parameters, select the instruction link parameter you wish to remove.
2. Click the **Remove parameter** button.
   The system asks you to confirm the action and then removes the instruction link parameter.

The following process parameters are available to configure the phase's behavior during execution:

Instruction text with links

Defines the text of the optional instruction link that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Attribute	Type	Comment
Instruction text	HTML text	Instruction text to be displayed. For any text enclosed in curly brackets you can define a hyperlink with the **Instruction link definition** process parameter (page 131). Example: Refer to {SOP1270} for guidance. Maximum length is 2000 characters (including HTML tags).

Instruction link definition

Defines all links to be available within the instruction text defined with the **Instruction text with links** process parameter (page 131). You can either access a file on the web, such as your intranet, by using the web access syntax or a file on your file system, by using the file access syntax.
Examples:

```
https://rockwellautomation.com
file:///c:/SOP/SOP1492.pdf
```

Attribute	Type	Comment
Link text	Text	Text to be used as link. For any text enclosed in curly brackets within the instruction text you can define a link with the **Link URL** attribute. Including the brackets in the link text is optional. Maximum length is 80 characters.
Link URL	Text	URL of the file to be displayed. The link opens the external application assigned to the file type by the operating system. Maximum length is 256 characters.

Instruction tables

In addition to the permanent process parameters that are always present, the **Write context data** phase provides instruction tables as optional process parameters, which you can insert if required.

You can add up to ten instruction tables with up to 50 table rows.

TIP

Instruction tables are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.

ADDING INSTRUCTION TABLES

1. Click the **Add parameter** button.
   The system opens an option list that holds all optional parameter types available for the phase.
2. Select the **Instruction Table** type.
   The system opens the **Add Instruction Table** dialog to define the instruction table's identifier.
3. Type an identifier and click the **OK** button.
   The system adds the parameter definition and the first table row to the list of parameters.
   Instruction table parameters are generally inserted below all other parameters.

Where within the block of instruction table parameters the system adds a new table depends on the current selection in the Parameter Panel:

- If no parameter is selected, the system adds the new instruction table as last parameter.
- If an instruction table parameter is selected, the system adds the new instruction table below the selected one.
- If any other parameter is selected, the system adds the new instruction table as first parameter of the instruction table parameter block.

TIPS

Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa.

The identifier of the instruction table is shown as **Identifier** of the table's definition parameter. The identifiers of the individual table rows (**Row-1**, **Row-2**, etc.) are system-defined and not editable.

4. Specify the overall appearance of the table:
   - Select the number of columns to define the layout.
   - Set the width of the first column. If you do not set it to narrow, all columns have equal widths. If you set the first column to narrow, the remaining columns will have equal widths.
   - Define if your table needs to have borders. You can either show all borders of the table and its cells or none.
5. Specify the table rows and their content. The button bar above the rows table provides the following functions:

   ⊞ adds a new row to the table. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.

   🗑 deletes the currently selected rows.

   ⬆ moves the currently selected row one row up.

   ⬇ moves the currently selected row one row down.

TIP

Please note that the system always retains the consecutive numbering of the rows. If you reorder the table rows or delete rows, the row identifiers are updated accordingly.

Removing instruction tables

1. In the list of parameters, select the instruction table parameter you wish to remove.
2. Click the **Remove parameter** button.
   The system asks you to confirm the action and then removes the instruction table with its definition and rows.

The following process parameters are available to configure the phase's behavior during execution:

Instruction table definition

Defines the appearance of the optional instruction table that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Attribute	Type	Comment
Table layout	Choice list	Defines the layout of the instruction table holding the instruction texts. Available settings: **1 column**, **2 columns**, **3 columns**, **4 columns**, **5 columns**. Default setting: **1 column**.
First column narrow	Boolean	Defines if the first column of the table shall be narrow.
Show all borders	Boolean	Defines if the borders of the table shall be visible.

Instruction table text

Specifies the instruction texts to be displayed in the individual cells of the instruction table.

Attribute	Type	Comment
Column 1	HTML text	Instruction text to be displayed in a column. **Restriction:** Maximum length is 2000 characters (including HTML tags).
Column 2	HTML text	
Column 3	HTML text	
Column 4	HTML text	
Column 5	HTML text	

Output Variables

Instead of specifying a fixed value to be displayed or used during execution, you can also use an expression created in the Expression editor to draw the output of another phase or operation or the calculated result of several outputs as value into a parameter attribute. When you reference outputs in this manner you need to be aware of the following restrictions:

- Only when a component has been processed does it provide an output that can be fed into another component as attribute value. For this reason, you can never reference an output of a component that is a strict successor of the component in which you try to use the output.
- Branches and loops, however, require special notice in this context, since they are only potentially passed through and/or completed during processing, so their outputs are not reliably available. Thus, you can reference any such potentially available outputs, but need to be aware of the fact that the provided value may be **Undefined** so that the component into which you are feeding the output must be able to deal with such an **Undefined** input value.

The **Write context data** phase provides the following output variables:

Context identifier

- Data type: String, used for displaying a pre-defined sequence of characters, such as "**SL0002854**" or "**0010**".
- Usage: The output variable provides the context identifier that was used to store the keys and values and is taken from the **Context definition** process parameter (page 125).

Identifier

- Data type: String, used for displaying a pre-defined sequence of characters, such as "**Read Instruction**".
- Usage: The output variable provides the identifier of the phase.

Instance count

- Data type: Long, used for integral numbers:
  12345
- Usage: The output variable provides the count of the number of instances the phase has been processed, for example in a loop. The count is also increased when the phase is skipped from an operator's perspective, since the phase is still executed, but as a hidden phase.
  The count variable of a phase that has not been executed provides 0 as output value.

Start time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
  To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the start time of the phase.

Completion time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
  To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the completion time of the phase.

TIP

To calculate a duration from two timestamps and display it in a specific format, you need to use two conversion functions on the calculation:

- **Convert to Unitless Number** (**convertTo**) takes the calculated duration and converts it into the duration's value for one of its units (e.g., minutes or seconds).
- **Convert to String for Display** (**convertToDisplayString)** takes the converted value and displays it as string to which you can add the unit, also as string.

Example:

Sample Phase with Start time = 14-Nov-2014@10:15
Sample Phase with Completion time = 14-Nov-2014@11:47
The duration is to be displayed in minutes.

```
convertToDisplayString
  (convertTo
      ({Sample Phase}.{Completion time}-{Sample Phase}.{Start time},
"min")
  )
 + " min"
```

As result of the expression, the system displays "**92 min**".

Send User Notification

The **Send user notification** phase allows to inform an operator on the shop floor about important data that is required to complete the current process in time and with the appropriate priority or to support the operator with taking the right decisions.

It can be used for processing requirements, such as:

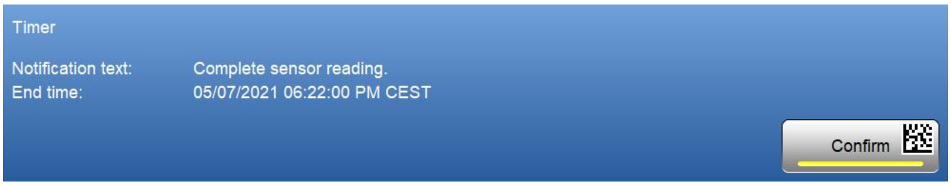
- Send a timestamp to a filling operation to display a countdown timer by which the material must be used and the filling process stopped.
- Send a timestamp by which the cleaning process will be finished.

Execution

In addition to the instruction text, the **Send user notification** phase displays the following data:

- the notification text to be displayed in the Notification Panel
- the definition in the form of a timestamp when the notification raises an alarm:
  - For a notification of the **Timer** type, it is the end time to which the timer counts down.
  - For a notification of the **Timestamp** type, it is the time by which an activity needs to have been completed.

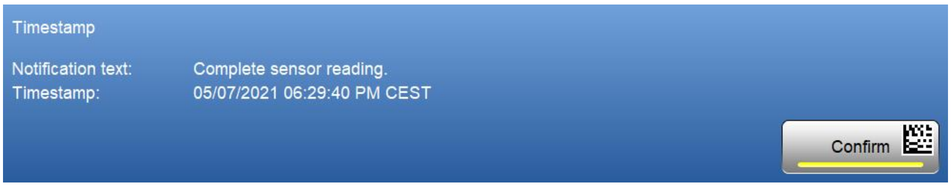
After completion, the phase also displays the notification text and the time definition. The Navigator displays the bundle type-specific value of the notification.



Timer

Notification text: Complete sensor reading.
End time: 05/07/2021 06:22:00 PM CEST

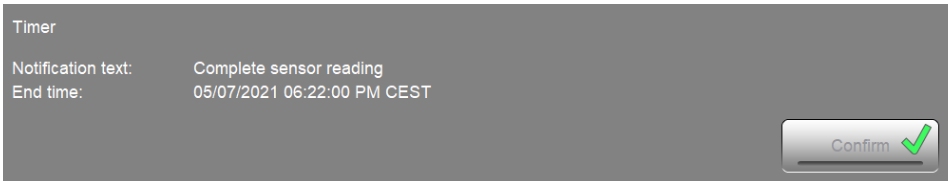
Figure 50: Send user notification during execution - Timer notification



Timestamp

Notification text: Complete sensor reading.
Timestamp: 05/07/2021 06:29:40 PM CEST

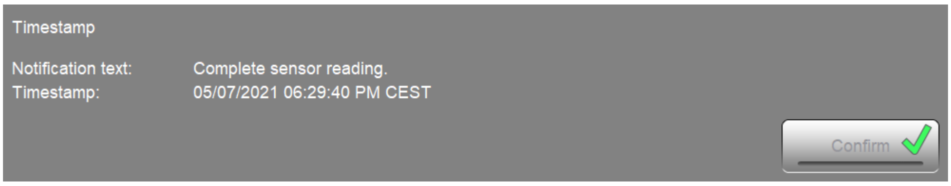
Figure 51: Send user notification during execution - Timestamp notification



Timer

Notification text: Complete sensor reading
End time: 05/07/2021 06:22:00 PM CEST

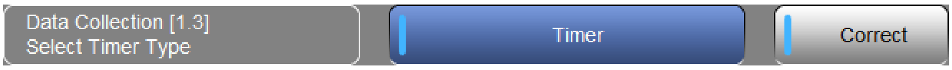
Figure 52: Send user notification after phase completion - Timer notification



Timestamp

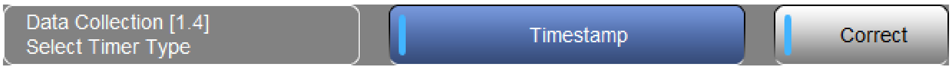
Notification text: Complete sensor reading.
Timestamp: 05/07/2021 06:29:40 PM CEST

Figure 53: Send user notification after phase completion - Timestamp notification



Data Collection [1.3]
Select Timer Type

Figure 54: Send user notification in the Navigator - Timer notification



Data Collection [1.4]
Select Timer Type

Figure 55: Send user notification in the Navigator - Timestamp notification

Phase Design

The characteristics of the **Send user notification** phase are defined via process parameters and their attributes.
Its user interface is designed in three columns that span several rows. When the phase is active, the merged columns of the first row provide space for textual instructions.
In the following rows, the phase displays the notification text and the time definition that will be displayed on the Notification Panel.
The right column of the bottom row contains the **Confirm** button.

When the phase is completed, it shows the same three-column, multi-row layout.
Exception handling during execution is controlled by a risk assessment classification and an exception message that are both defined by the recipe author in the exception's process parameter.

Process Parameters

The following process parameters are available to configure the phase's behavior during execution:

Instruction

Represents the instruction text that is visible on the preview, the active, and the completed view of the phase.

Attribute	Type	Comment
Text	HTML text	Instruction text to be displayed. Maximum length is 2000 characters (including HTML tags).

Mode

Defines if the phase expects operator interaction during execution.

Attribute	Type	Comment
Mode	Choice list	Defines the processing mode. **Manual completion** (default): Operator confirms the phase. **Automatic completion**: Phase is automatically completed after the data has been written.

Parameter bundles

In addition to the permanent process parameters that are always present, the **Send user notification** phase provides parameter bundles as optional process parameters, which you can insert if required.

You can add exactly one process parameter bundle of the two different data types (**Timer Notification**, **Timestamp Notification**) to the **Send user notification** phase.

ADDING PARAMETER BUNDLES

1. Click the **Add parameter** button.
   The system opens an option list that holds all data types available for the value.
2. Select the type.
   The system opens the **Add <Data Type>** dialog to define the value's identifier.
3. Type an identifier and click the **OK** button.
   The system adds all process parameters of the bundle to the list of parameters.

REMOVING PARAMETER BUNDLES

1. In the list of parameters, select the header row that contains the identifier of the bundle you wish to remove.
2. Click the **Remove parameter** button.
   The system asks you to confirm the action and then removes the value bundle.

The following process parameters are available to configure the phase's behavior during execution:

Timer - Master (bundle identifier)

Defines the identifier and text value pair for the notification, along with the path to the component for which the notification is displayed.

Attribute	Type	Comment
Notification identifier	String	Together with the order or workflow identifier, it represents a unique value and can be used to update the notification text and/or value of the same bundle type within a further notification. Maximum length is 80 characters.
Reference path	String	Defines the visibility of the notification. Allowed values are ■ Empty ■ Unit procedure ■ Unit procedure / Operation Maximum length is 1024 characters. TIP A notification update needs to use the same reference path and notification type as the initial notification.
Notification text	String	Defines the text to be displayed in the target Production Execution Client in the first column of the notification panel. Maximum length is 1024 characters. TIP Please note that HTML tags will be shown as plain text and not interpreted by the notification panel.

Timer - End time

Defines the time to which the timer counts down.

Attribute	Type	Comment
Timestamp	Timestamp	Defines the timestamp by which the countdown timer must reach 0. Can be defined with a Date/Time Picker editor or an expression.

Timestamp - Master (bundle identifier)

Defines the identifier and text value pair for the notification, along with the path to the component for which the notification is displayed.

Attribute	Type	Comment
Notification identifier	String	Together with the order or workflow identifier, it represents a unique value and can be used to update the notification text and/or value of the same bundle type within a further notification. Maximum length is 80 characters.
Reference path	String	Defines the visibility of the notification. Allowed values are ■ Empty ■ Unit procedure ■ Unit procedure / Operation Maximum length is 1024 characters. TIP A notification update needs to use the same reference path and notification type as the initial notification.
Notification text	String	Defines the text to be displayed in the target Production Execution Client in the first column of the notification panel. Maximum length is 1024 characters. TIP Please note that HTML tags will be shown as plain text and not interpreted by the notification panel.

Timestamp - Timestamp

Defines the time after which an alarm is raised.

Attribute	Type	Comment
Timestamp	Timestamp	Defines the timestamp to be displayed with the notification. Can be defined with a Date/Time Picker editor or an expression.

Timestamp - Alarm

Defines if an alarm is raised in the notification panel when the point in time defined with the timestamp has passed.

Attribute	Type	Comment
Enabled	Boolean	Defines whether the notification will be displayed as an alarm or not. Default setting: **No**.

Timestamp - Pinned

Defines the position of a notification in the list displayed in the notification panel.

Attribute	Type	Comment
Enabled	Boolean	Defines whether the notification will be pinned at the top of the notification panel list or not. Default setting: **No**.

Instruction links

In addition to the permanent process parameters that are always present, the **Send user notification** phase provides instruction links as optional process parameters, which you can insert if required.

You can add up to ten instruction link parameters.

TIP

Instruction links are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.

⊞ ADDING INSTRUCTION LINKS

1. Click the **Add parameter** button.
   The system opens an option list that holds all optional parameter types available for the phase.

2. Select the **Instruction Link** type.
   The system opens the **Add Instruction Link** dialog to define the instruction link's identifier.

3. Type an identifier and click the **OK** button.
   The system adds the instruction link parameter and the first link definition row to the list of parameters.
   Instruction link parameters are generally inserted below all other parameters. Where within the block of instruction link parameters the system adds a new link parameter depends on the current selection in the Parameter Panel:

   - If no parameter is selected, the system adds the new instruction link parameter as last parameter.
   - If an instruction link parameter is selected, the system adds the new instruction link parameter below the selected one.
   - If any other parameter is selected, the system adds the new instruction link parameter as first parameter of the instruction link parameter block.

   > TIPS
   > Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa.
   > The identifier of the instruction link parameter is shown as **Identifier** of the link's instruction text parameter.

4. Specify the instruction text to be displayed and mark the link texts by enclosing them in curly brackets.

5. Specify the list of link definitions. Each row of the list defines one hyperlink. The button bar above the list provides the following functions:

   ⊞ adds a new row to the list. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.

   🗑 deletes the currently selected rows.

   ⬆ moves the currently selected row one row up.

   ⬇ moves the currently selected row one row down.

REMOVING INSTRUCTION LINKS

1. In the list of parameters, select the instruction link parameter you wish to remove.
2. Click the **Remove parameter** button.
   The system asks you to confirm the action and then removes the instruction link parameter.

The following process parameters are available to configure the phase's behavior during execution:

Instruction text with links

Defines the text of the optional instruction link that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Attribute	Type	Comment
Instruction text	HTML text	Instruction text to be displayed. For any text enclosed in curly brackets you can define a hyperlink with the **Instruction link definition** process parameter (page 146). Example: Refer to {SOP1270} for guidance. Maximum length is 2000 characters (including HTML tags).

Instruction link definition

Defines all links to be available within the instruction text defined with the **Instruction text with links** process parameter (page 146). You can either access a file on the web, such as your intranet, by using the web access syntax or a file on your file system, by using the file access syntax.
Examples:

```
https://rockwellautomation.com
file:///c:/SOP/SOP1492.pdf
```

Attribute	Type	Comment
Link text	Text	Text to be used as link. For any text enclosed in curly brackets within the instruction text you can define a link with the **Link URL** attribute. Including the brackets in the link text is optional. Maximum length is 80 characters.
Link URL	Text	URL of the file to be displayed. The link opens the external application assigned to the file type by the operating system. Maximum length is 256 characters.

Instruction tables

In addition to the permanent process parameters that are always present, the **Write context data** phase provides instruction tables as optional process parameters, which you can insert if required.

You can add up to ten instruction tables with up to 50 table rows.

TIP

Instruction tables are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.

ADDING INSTRUCTION TABLES

1. Click the **Add parameter** button.
   The system opens an option list that holds all optional parameter types available for the phase.
2. Select the **Instruction Table** type.
   The system opens the **Add Instruction Table** dialog to define the instruction table's identifier.
3. Type an identifier and click the **OK** button.
   The system adds the parameter definition and the first table row to the list of parameters.
   Instruction table parameters are generally inserted below all other parameters.

Where within the block of instruction table parameters the system adds a new table depends on the current selection in the Parameter Panel:

- If no parameter is selected, the system adds the new instruction table as last parameter.
- If an instruction table parameter is selected, the system adds the new instruction table below the selected one.
- If any other parameter is selected, the system adds the new instruction table as first parameter of the instruction table parameter block.

TIPS
Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa.

The identifier of the instruction table is shown as **Identifier** of the table's definition parameter. The identifiers of the individual table rows (**Row-1**, **Row-2**, etc.) are system-defined and not editable.

4. Specify the overall appearance of the table:
   - Select the number of columns to define the layout.
   - Set the width of the first column. If you do not set it to narrow, all columns have equal widths. If you set the first column to narrow, the remaining columns will have equal widths.
   - Define if your table needs to have borders. You can either show all borders of the table and its cells or none.
5. Specify the table rows and their content. The button bar above the rows table provides the following functions:

   ⊞ adds a new row to the table. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.

   🗑 deletes the currently selected rows.

   ⬆ moves the currently selected row one row up.

   ⬇ moves the currently selected row one row down.

TIP
Please note that the system always retains the consecutive numbering of the rows. If you reorder the table rows or delete rows, the row identifiers are updated accordingly.

REMOVING INSTRUCTION TABLES

1. In the list of parameters, select the instruction table parameter you wish to remove.
2. Click the **Remove parameter** button.
   The system asks you to confirm the action and then removes the instruction table with its definition and rows.

The following process parameters are available to configure the phase's behavior during execution:

Instruction table definition

Defines the appearance of the optional instruction table that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Attribute	Type	Comment
Table layout	Choice list	Defines the layout of the instruction table holding the instruction texts. Available settings: **1 column**, **2 columns**, **3 columns**, **4 columns**, **5 columns**. Default setting: **1 column**.
First column narrow	Boolean	Defines if the first column of the table shall be narrow.
Show all borders	Boolean	Defines if the borders of the table shall be visible.

Instruction table text

Specifies the instruction texts to be displayed in the individual cells of the instruction table.

Attribute	Type	Comment
Column 1	HTML text	Instruction text to be displayed in a column. **Restriction:** Maximum length is 2000 characters (including HTML tags).
Column 2	HTML text	
Column 3	HTML text	
Column 4	HTML text	
Column 5	HTML text	

Output Variables

Instead of specifying a fixed value to be displayed or used during execution, you can also use an expression created in the Expression editor to draw the output of another phase or operation or the calculated result of several outputs as value into a parameter attribute. When you reference outputs in this manner you need to be aware of the following restrictions:

- Only when a component has been processed does it provide an output that can be fed into another component as attribute value. For this reason, you can never reference an output of a component that is a strict successor of the component in which you try to use the output.
- Branches and loops, however, require special notice in this context, since they are only potentially passed through and/or completed during processing, so their outputs are not reliably available. Thus, you can reference any such potentially available outputs, but need to be aware of the fact that the provided value may be **Undefined** so that the component into which you are feeding the output must be able to deal with such an **Undefined** input value.

The **Send user notification** phase provides the following output variables:

Timer notification

If you have added a Timer Notification process parameter bundle to the list of parameters, the system provides a corresponding output variable for it.

TIP
Similar to the convention used for naming a bundle's process parameters, its output variables are prefixed with its bundle identifier.

The following output variable is available for a Timer Notification bundle:

End time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
  To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the defined value of the countdown timer end time.

Timestamp notification

If you have added a Timestamp Notification process parameter bundle to the list of parameters, the system provides a corresponding output variable for it.

> TIP
> Similar to the convention used for naming a bundle's process parameters, its output variables are prefixed with its bundle identifier.

The following output variable is available for a Timestamp Notification bundle:

Timestamp

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
  To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the defined timestamp value.

Identifier

- Data type: String, used for displaying a pre-defined sequence of characters, such as "**Read Instruction**".
- Usage: The output variable provides the identifier of the phase.

Instance count

- Data type: Long, used for integral numbers:
  12345
- Usage: The output variable provides the count of the number of instances the phase has been processed, for example in a loop. The count is also increased when the phase is skipped from an operator's perspective, since the phase is still executed, but as a hidden phase.

The count variable of a phase that has not been executed provides 0 as output value.

Start time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
  To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the start time of the phase.

Completion time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
  To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the completion time of the phase.

TIP

To calculate a duration from two timestamps and display it in a specific format, you need to use two conversion functions on the calculation:

- **Convert to Unitless Number** (**convertTo**) takes the calculated duration and converts it into the duration's value for one of its units (e.g., minutes or seconds).
- **Convert to String for Display** (**convertToDisplayString)** takes the converted value and displays it as string to which you can add the unit, also as string.

Example:

Sample Phase with Start time = 14-Nov-2014@10:15
Sample Phase with Completion time = 14-Nov-2014@11:47
The duration is to be displayed in minutes.

```
convertToDisplayString
  (convertTo
      ({Sample Phase}.{Completion time}-{Sample Phase}.{Start time},
"min")
  )
 + " min"
```

As result of the expression, the system displays "**92 min**".

Index

A

Alarm - Timestamp notification (Send user notification, parameter) • 144

Append workflow (Create workflow, parameter) • 109

B

Bundle identifier (Send user notification, parameter)

- Timer • 142
- Timestamp • 143

Bundle identifier (Write context data, parameter)

- BigDecimal • 127
- Boolean • 128
- Duration • 128
- Long • 128
- MeasuredValue • 129
- String • 129
- Timestamp • 129

C

Completion time (output)

- Create workflow • 120
- Get choice value • 14
- Get process value • 31
- Get text value • 43
- Send user notification • 152
- Show document • 54
- Show instruction text • 63
- Show URL • 74
- Upload image • 88
- Upload PDF • 103
- Write context data • 136

Context and key check (Write context data, parameter) • 126

Context definition (Write context data, parameter) • 125

Context identifier (Write context data, output) • 135

Conventions (typographical) • 2

Correct value (Get choice value, parameter) • 7

Create workflow • 105

- Output variable • 118
- Process parameter • 107

Creation result (Create workflow, output) • 119

D

Detail information (Create workflow, output) • 119

Detail information (Create workflow, parameter) • 112

Document (parameter)

- Show document • 47
- Show URL • 67

E

End time - Timer notification (Send user notification, parameter) • 143

End time (Send user notification, output) • 151

Expected value configuration (parameter)

- Get choice value • 6
- Get text value • 35

Expected value definition (parameter)

- Get choice value • 6
- Get text value • 35

F

File location (parameter)

- Upload image • 79
- Upload PDF • 94

G

Get choice value • 3

- Output variable • 13
- Process parameter • 4

Get process value • 17

- Output variable • 30
- Process parameter • 19

Get text value • 33

- Output variable • 42
- Process parameter • 34

I

Identifier (output)
- Create workflow • 120
- Get choice value • 14
- Get process value • 30
- Get text value • 43
- Send usre notification • 151
- Show document • 53
- Show instruction text • 62
- Show URL • 73
- Upload image • 87
- Upload PDF • 103
- Write context data • 136

Image full path (Upload image, output) • 87

Image timestamp (Upload image, output) • 87

Instance count (output)
- Create workflow • 120
- Get choice value • 14
- Get process value • 31
- Get text value • 43
- Send user notification • 151
- Show document • 53
- Show instruction text • 62
- Show URL • 73
- Upload image • 87
- Upload PDF • 103
- Write context data • 136

Instruction (parameter)
- Create workflow • 107
- Get choice value • 4
- Get process value • 19
- Get text value • 34
- Send user notification • 141
- Show document • 46
- Show instruction text • 56
- Show URL • 66
- Upload image • 78
- Upload PDF • 93
- Write context data • 125

Instruction link definition (parameter)
- Create workflow • 115
- Get choice value • 9
- Get process value • 26
- Get text value • 39
- Send user notification • 131
- Show document • 49
- Show instruction text • 58
- Show URL • 70
- Upload image • 83
- Upload PDF • 99
- Write context data • 131

Instruction table definition (parameter)
- Create workflow • 117
- Get choice value • 12
- Get process value • 29
- Get text value • 41
- Send user notification • 149
- Show document • 52
- Show instruction text • 61
- Show URL • 72
- Upload image • 85
- Upload PDF • 101
- Write context data • 134

Instruction table text (parameter)
- Create workflow • 118
- Get choice value • 12
- Get process value • 29
- Get text value • 42
- Send user notification • 149
- Show document • 52
- Show instruction text • 61
- Show URL • 72
- Upload image • 86
- Upload PDF • 102
- Write context data • 134

Instruction text with links (parameter)
- Create workflow • 114
- Get choice value • 9
- Get process value • 26
- Get text value • 38
- Send user notification • 146
- Show document • 49

Show instruction text • 58
Show URL • 69
Upload image • 83
Upload PDF • 98
Write context data • 131

L

Layout (Show instruction text, parameter) • 56
Limit configuration (Get process value, parameter) • 20
Limit definition (Get process value, parameter) • 22
List of options (Get choice value, parameter) • 5

M

Mandatory upload check (parameter)
Upload image • 80
Upload PDF • 95
Master workflow (Create workflow, parameter) • 108
Mode (parameter)
Create workflow • 107
Send user notification • 141
Upload image • 78
Upload PDF • 94
Write context data • 125

O

One-click completion (Get choice value, parameter) • 7
Option key (Get choice value, output) • 13
Option text (Get choice value, output) • 14
Output value (Get text value, output) • 43
Output variable
Completion time (Create workflow) • 120
Completion time (Get choice value) • 14
Completion time (Get process value) • 31
Completion time (Get text value) • 43
Completion time (Send usre notification) • 152
Completion time (Show document) • 54
Completion time (Show instruction text) • 63
Completion time (Show URL) • 74
Completion time (Upload image) • 88
Completion time (Upload PDF) • 103
Completion time (Write context data) • 136
Context identifier (Write context data) • 135
Create workflow • 118
Creation result (Create workflow) • 119
Detail information (Create workflow) • 119
End time (Send user notification) • 151
Get choice value • 13
Get process value • 30
Get text value • 42
Identifier (Create workflow) • 120
Identifier (Get choice value) • 14
Identifier (Get process value) • 30
Identifier (Get text value) • 43
Identifier (Send usre notification) • 151
Identifier (Show document) • 53
Identifier (Show instruction text) • 62
Identifier (Show URL) • 73
Identifier (Upload Image) • 87
Identifier (Upload PDF) • 103
Identifier (Write context data) • 136
Image full path (Upload image) • 87
Image timestamp (Upload image) • 87
Instance count (Create workflow) • 120
Instance count (Get choice value) • 14
Instance count (Get process value) • 31
Instance count (Get text value) • 43
Instance count (Send user notification) • 151
Instance count (Show document) • 53
Instance count (Show instruction text) • 62
Instance count (Show URL) • 73
Instance count (Upload image) • 87
Instance count (Upload PDF) • 103
Instance count (Write context data) • 136
Option key (Get choice value) • 13
Option text (Get choice value) • 14
Output value (Get text value) • 43
PDF full path (Upload PDF) • 102
PDF timestamp (Upload PDF) • 103
Planned end (Create workflow) • 119
Planned start (Create workflow) • 119
Send user notification • 150
Show document • 53

Show instruction text • 62
Show URL • 73
Start time (Create workflow) • 120
Start time (Get choice value) • 14
Start time (Get process value) • 31
Start time (Get text value) • 43
Start time (Send user notification) • 152
Start time (Show document) • 53
Start time (Show instruction text) • 63
Start time (Show URL) • 74
Start time (Upload image) • 88
Start time (Upload PDF) • 103
Start time (Write context data) • 136
Timestamp (Send user notification) • 151
Unit of measure (Get process value) • 30
Upload image • 86
Upload PDF • 102
Value (Get process value) • 30
Workflow identifier (Create workflow) • 118
Write context data • 135
Override value (parameter)
Get process value • 24
Write context data • 125

P

Parameter bundle
Write context data • 127
PDF full path (Upload PF, output) • 102
PDF timestamp (Upload PDF, output) • 103
Pinned - Timestamp notification (Send user notification, parameter) • 144
Planned end (Create workflow, output) • 119
Planned end (Create workflow, parameter) • 112
Planned start (Create workflow, output) • 119
Planned start (Create workflow, parameter) • 111
Post - Correct value (parameter)
Get process value • 23
Get text value • 36
Process parameter • 140
Alarm - Timestamp notification (Send user notification) • 144
Append workflow (Create workflow) • 109
Bundle identifier - BigDecimal (Write context data) • 127
Bundle identifier - Boolean (Write context data) • 128
Bundle identifier - Duration (Write context data) • 128
Bundle identifier - Long (Write context data) • 128
Bundle identifier - MeasuredValue (Write context data) • 129
Bundle identifier - String (Write context data) • 129
Bundle identifier - Timer (Send user notification, parameter) • 142
Bundle identifier - Timestamp (Send user notification, parameter) • 143
Bundle identifier - Timestamp (Write context data) • 129
Context and key check (Write context data) • 126
Context definition (Write context data) • 125
Correct value (Get choice value) • 7
Create workflow • 107
Detail information (Create workflow) • 112
Document (Show document) • 47
Document (Show URL) • 67
End time - Timer notification (Send user notification) • 143
Expected value configuration (Get choice value) • 6
Expected value configuration (Get text value) • 35
Expected value definition (Get choice value) • 6
Expected value definition (Get text value) • 35
File location (Upload image) • 79
Get choice value • 4
Get process value • 19
Get text value • 34
Image location (Upload PDF) • 94
Instruction (Create workflow) • 107
Instruction (Get choice value) • 4
Instruction (Get process value) • 19
Instruction (Get text value) • 34
Instruction (Send user notification) • 141
Instruction (Show document) • 46
Instruction (Show instruction text) • 56
Instruction (Show URL) • 66

Instruction (Upload image) • 78
Instruction (Upload PDF) • 93
Instruction (Write context data) • 125
Instruction link definition (Create workflow) • 115
Instruction link definition (Get choice value) • 9
Instruction link definition (Get process value) • 26
Instruction link definition (Get text value) • 39
Instruction link definition (Send user notification) • 131
Instruction link definition (Show document) • 49
Instruction link definition (Show instruction text) • 58
Instruction link definition (Show URL) • 70
Instruction link definition (Upload image) • 83
Instruction link definition (Upload PDF) • 99
Instruction link definition (Write context data) • 131
Instruction table definition (Create workflow) • 117
Instruction table definition (Get choice value) • 12
Instruction table definition (Get process value) • 29
Instruction table definition (Get text value) • 41
Instruction table definition (Send user notification) • 149
Instruction table definition (Show document) • 52
Instruction table definition (Show instruction text) • 61
Instruction table definition (Show URL) • 72
Instruction table definition (Upload image) • 85
Instruction table definition (Upload PDF) • 101
Instruction table definition (Write context data) • 134
Instruction table text (Create workflow) • 118
Instruction table text (Get choice value) • 12
Instruction table text (Get process value) • 29
Instruction table text (Get text value) • 42
Instruction table text (Send user notification) • 149
Instruction table text (Show document) • 52
Instruction table text (Show instruction text) • 61
Instruction table text (Show URL) • 72
Instruction table text (Upload image) • 86
Instruction table text (Upload PDF) • 102
Instruction table text (Write context data) • 134
Instruction text with links (Create workflow) • 114
Instruction text with links (Get choice value) • 9
Instruction text with links (Get process value) • 26
Instruction text with links (Get text value) • 38
Instruction text with links (Send user notification) • 146
Instruction text with links (Show document) • 49
Instruction text with links (Show instruction text) • 58
Instruction text with links (Show URL) • 69
Instruction text with links (Upload image) • 83
Instruction text with links (Upload PDF) • 98
Instruction text with links (Write context data) • 131
Layout (Show instruction text) • 56
Limit configuration (Get process value) • 20
Limit definition (Get process value) • 22
List of options (Get choice value) • 5
Mandatory upload check (Upload image) • 80
Mandatory upload check (Upload PDF) • 95
Master workflow (Create workflow) • 108
Mode (Create workflow) • 107
Mode (Send user notification) • 141
Mode (Upload image) • 78
Mode (Upload PDF) • 94
Mode (Write context data) • 125
One-click completion (Get choice value) • 7
Override value (Get process value) • 24
Pinned - Timestamp notification (Send user notification) • 144
Planned end (Create workflow) • 112
Planned start (Create workflow) • 111
Post - Correct value (Get process value) • 23
Post - Correct value (Get text value) • 36
Replace file (Upload image) • 81
Replace file (Upload PDF) • 96
Select manually (Upload image) • 80
Select manually (Upload PDF) • 96
Send user notification • 140
Show document • 46
Show instruction text • 56
Show URL • 66
Skip workflow creation (Create workflow) • 112
Start workflow automatically (Create workflow) • 111
Station (Create workflow) • 110
Timestamp - Timestamp notification (Send user notification) • 144
Upload image • 78

Upload PDF • 93
Value configuration (Get process value) • 19
Work center (Create workflow) • 110
Workflow definition (Create workflow) • 108
Write context data • 124
Write context data (Override value) • 125

R

Replace file (parameter)
Upload image • 81
Upload PDF • 96

S

Select manually (parameter)
Upload image • 80
Upload PDF • 96
Send user notification • 139
Output variable • 150
Show document • 45
Output variable • 53
Process parameter • 46
Show instruction text • 55
Output variable • 62
Process parameter • 56
Show URL • 65
Output variable • 73
Process parameter • 66
Skip workflow creation (Create workflow, parameter) • 112
Start time (output)
Create workflow • 120
Get choice value • 14
Get process value • 31
Get text value • 43
Send user notification • 152
Show document • 53
Show instruction text • 63
Show URL • 74
Upload image • 88
Upload PDF • 103
Write context data • 136
Start workflow automatically (Create workflow, parameter) • 111
Station (Create workflow, parameter) • 110

T

Timestamp - Timestamp notification (Send user notification, parameter) • 144
Timestamp (Send user notification, output) • 151

U

Unit of measure (Get process value, output) • 30
Upload image • 75
Output variable • 86
Process parameter • 78
Upload PDF • 89
Output variable • 102
Process parameter • 93

V

Value (Get process value, output) • 30
Value configuration (Get process value, parameter) • 19

W

Work center (Create workflow, parameter) • 110
Workflow definition (Create workflow, parameter) • 108
Workflow identifier (Create workflow, output) • 118
Write context data • 123
Output variable • 135
Process parameter • 124