

LISTEN.
THINK.
SOLVE.®

PharmaSuite®



EBR PHASES
RELEASE 8.4
USER MANUAL

PUBLICATION PSEB-UM003D-EN-E-DECEMBER-2017
Supersedes publication PSEB-UM003C-EN-E



Allen-Bradley • Rockwell Software

Rockwell
Automation

Contact Rockwell See contact information provided in your maintenance contract.

Copyright Notice © 2017 Rockwell Automation Technologies, Inc. All rights reserved.
This document and any accompanying Rockwell Software products are copyrighted by Rockwell Automation Technologies, Inc. Any reproduction and/or distribution without prior written consent from Rockwell Automation Technologies, Inc. is strictly prohibited. Please refer to the license agreement for details.

Trademark Notices FactoryTalk, PharmaSuite, Rockwell Automation, Rockwell Software, and the Rockwell Software logo are registered trademarks of Rockwell Automation, Inc.

The following logos and products are trademarks of Rockwell Automation, Inc.:

FactoryTalk Shop Operations Server, FactoryTalk ProductionCentre, FactoryTalk Administration Console, FactoryTalk Automation Platform, and FactoryTalk Security.

Operational Data Store, ODS, Plant Operations, Process Designer, Shop Operations, Rockwell Software CPGSuite, and Rockwell Software AutoSuite.

Other Trademarks ActiveX, Microsoft, Microsoft Access, SQL Server, Visual Basic, Visual C++, Visual SourceSafe, Windows, Windows 7 Professional, Windows Server 2008, Windows Server 2012, and Windows Server 2016 are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Adobe, Acrobat, and Reader are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

ControlNet is a registered trademark of ControlNet International.

DeviceNet is a trademark of the Open DeviceNet Vendor Association, Inc. (ODVA).

Ethernet is a registered trademark of Digital Equipment Corporation, Intel, and Xerox Corporation.

OLE for Process Control (OPC) is a registered trademark of the OPC Foundation.

Oracle, SQL*Net, and SQL*Plus are registered trademarks of Oracle Corporation.

All other trademarks are the property of their respective holders and are hereby acknowledged.

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.

Chapter 1	EBR Phases.....	1
	Typographical Conventions	2
Chapter 2	Get Choice Value	3
	Execution	4
	Phase Design	5
	Process Parameters	5
	Output Variables	8
Chapter 3	Get Process Value	11
	Execution	12
	Phase Design	13
	Process Parameters	13
	Output Variables	19
Chapter 4	Get Text Value	23
	Execution	24
	Phase Design	25
	Process Parameters	25
	Output Variables	28
Chapter 5	Show Document.....	31
	Execution	32
	Phase Design	32
	Process Parameters	33
	Output Variables	34

Chapter 6	Show Instruction Text	37
	Execution	37
	Phase Design	38
	Process Parameters	38
	Output Variables	39
Chapter 7	Show URL	41
	Execution	41
	Phase Design	42
	Process Parameters	42
	Output Variables	44
Chapter 8	Upload Image	47
	Execution	47
	Phase Design	50
	Process Parameters	50
	Output Variables	54
Chapter 9	Upload PDF	57
	Execution	57
	Phase Design	61
	Process Parameters	61
	Output Variables	65
Index	69

Figure 1: Get choice value during execution.....	4
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	6
Figure 6: Get process value during execution	12
Figure 7: User-triggered exception of Get process value	12
Figure 8: Get process value after phase completion	12
Figure 9: Get process value in the Navigator	12
Figure 10: Post-completion exception of Get process value	12
Figure 11: Get text value during execution	24
Figure 12: User-triggered exception of Get text value.....	24
Figure 13: Get text value after phase completion	24
Figure 14: Get text value in the Navigator	24
Figure 15: Post-completion exception of Get text value	24
Figure 16: Show document during execution	32
Figure 17: Show document after phase completion	32
Figure 18: Show document in the Navigator	32
Figure 19: Show instruction text during execution	37
Figure 20: Show instruction text after phase completion	37
Figure 21: Show instruction text in the Navigator	37
Figure 22: Show URL during execution	41
Figure 23: Show URL after phase completion	41
Figure 24: Show URL in the Navigator	42
Figure 25: Upload image during execution - Automatic loading mode	48
Figure 26: Upload image during execution - Selection mode	48
Figure 27: User-triggered exception of Upload image.....	48

Figure 28: File selection dialog of Upload image.....	49
Figure 29: Upload image after phase completion	49
Figure 30: Upload image in the Navigator	49
Figure 31: Post-completion exception of Upload image	49
Figure 32: Upload PDF during execution - Automatic loading mode.....	58
Figure 33: Upload PDF during execution - Selection mode	58
Figure 34: User-triggered exception of Upload PDF	59
Figure 35: File selection dialog of Upload PDF	59
Figure 36: Upload PDF after phase completion.....	60
Figure 37: Upload PDF in the Navigator.....	60
Figure 38: Post-completion exception of Upload PDF	60

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 [11](#))
- Get Text Value (page [23](#))
- Show Document (page [31](#))
- Show Instruction Text (page [37](#))
- Show URL (page [41](#))
- Upload Image (page [47](#))
- Upload PDF (page [57](#))

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 <ul style="list-style-type: none">■ window and dialog titles■ menu functions■ panel, tab, and button names■ box labels■ object properties and their values (e.g. status).
<i>Italic typeface</i>	Designates technical background information, such as <ul style="list-style-type: none">■ path, folder, and file names■ methods■ classes.
CAPITALS	Designate keyboard-related information, such as <ul style="list-style-type: none">■ key names■ keyboard shortcuts.
Monospaced typeface	Designates code examples.

TIP

Instructions in this manual are based on Windows 7. Select the appropriate commands if you are using a different operating system.

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.

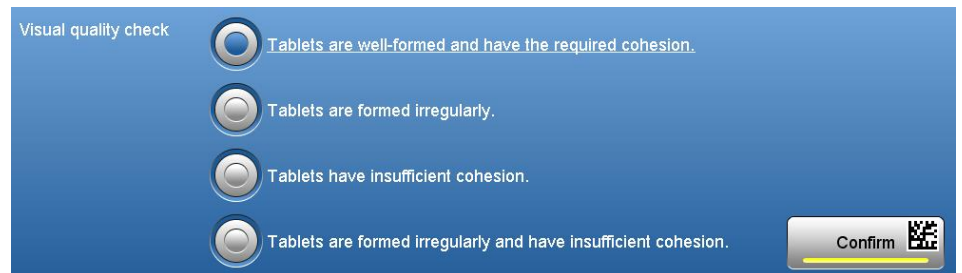


Figure 1: Get choice value during execution

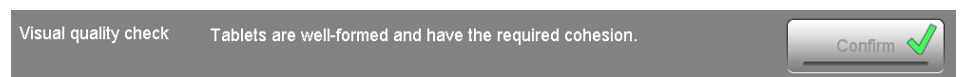


Figure 2: Get choice value after phase completion



Figure 3: Get choice value in the Navigator

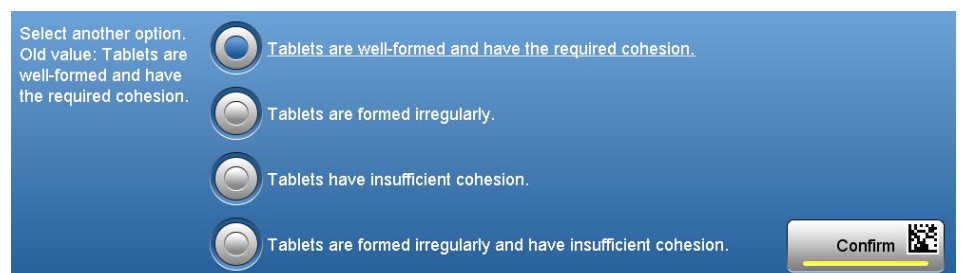


Figure 4: Post-completion exception of Get choice value

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.
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.

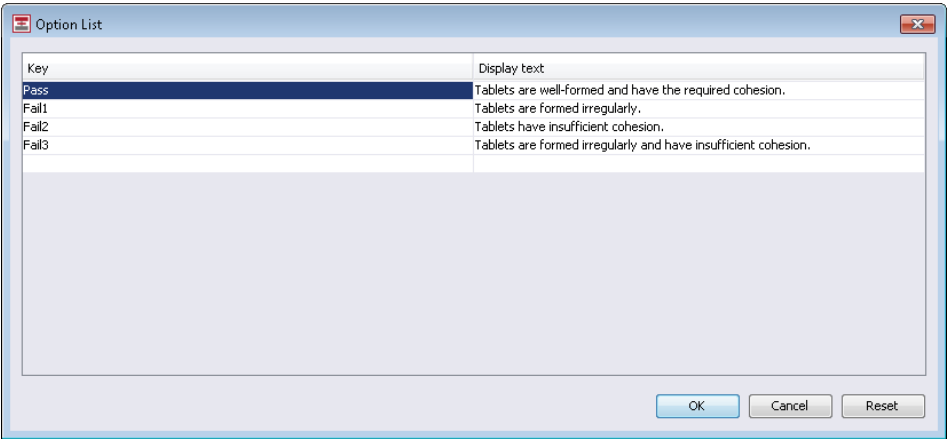


Figure 5: Option List editor

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 7) 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 7) is set.
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: None , 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.

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 the calculated result of several phase outputs as value into a parameter attribute. When you reference phase outputs in this manner you need to be aware of the following restrictions:

- Only if a phase has been processed does it provide an output that can be fed into another phase as attribute value. For this reason, you can never reference an output of a phase that is a strict successor of the phase 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 phase to 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"**.

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	

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	Correct
-------------------------------	-------	---------

Figure 9: Get process value in the Navigator

Please enter new value	Old value	6.5 t
0 further exceptions	New value	

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 16). 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 16) 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 16) 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.

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.

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 16) 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.

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 16) 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.

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 14) (e.g. weight: mg, kg, pound; length: mm, m, inch),
- the limits are strictly sequential and do not overlap, so that

$$\text{LLL limit} < \text{LL limit} < \text{L limit} < \text{Reference value} < \text{H limit} < \text{HH limit} < \text{HHH limit}$$

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 14).
 It covers incidents when a reading error causes a calculated process value to fail the limits defined with the **Limit definition** process parameter (page 16), 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.

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 the calculated result of several phase outputs as value into a parameter attribute. When you reference phase outputs in this manner you need to be aware of the following restrictions:

- Only if a phase has been processed does it provide an output that can be fed into another phase as attribute value. For this reason, you can never reference an output of a phase that is a strict successor of the phase 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 phase to 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".

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.



Figure 11: Get text value during execution



Figure 12: User-triggered exception of Get text value

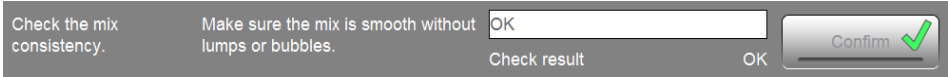


Figure 13: Get text value after phase completion



Figure 14: Get text value in the Navigator



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 27). 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 27) 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 26) 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.
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 27).

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.

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 the calculated result of several phase outputs as value into a parameter attribute. When you reference phase outputs in this manner you need to be aware of the following restrictions:

- Only if a phase has been processed does it provide an output that can be fed into another phase as attribute value. For this reason, you can never reference an output of a phase that is a strict successor of the phase 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 phase to 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"**.

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.

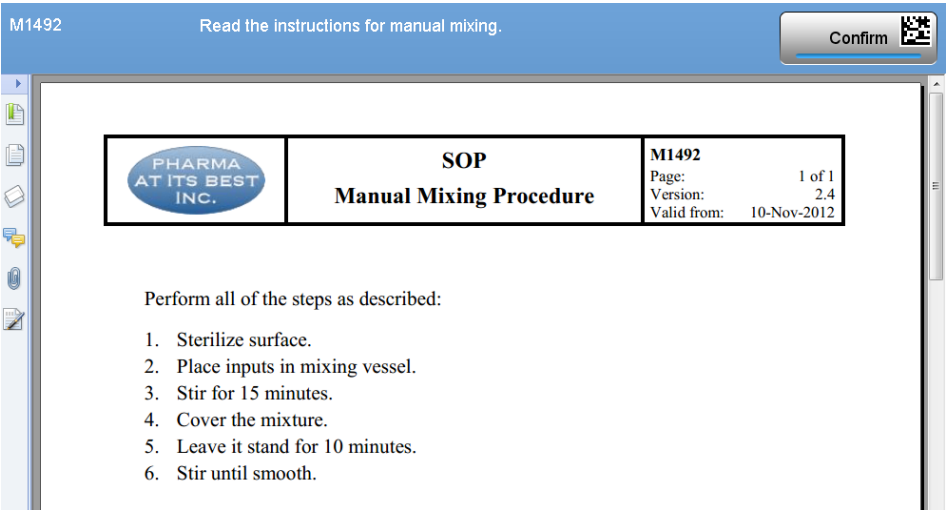


Figure 16: Show document during execution

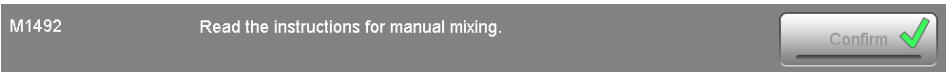


Figure 17: Show document after phase completion



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.

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 the calculated result of several phase outputs as value into a parameter attribute. When you reference phase outputs in this manner you need to be aware of the following restrictions:

- Only if a phase has been processed does it provide an output that can be fed into another phase as attribute value. For this reason, you can never reference an output of a phase that is a strict successor of the phase 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 phase to 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".

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.



Figure 19: Show instruction text during execution



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 38). 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 38).
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 .

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 the calculated result of several phase outputs as value into a parameter attribute. When you reference phase outputs in this manner you need to be aware of the following restrictions:

- Only if a phase has been processed does it provide an output that can be fed into another phase as attribute value. For this reason, you can never reference an output of a phase that is a strict successor of the phase 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 phase to 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 43).

After completion it displays the URL of the linked document. The Navigator displays the name of the phase.

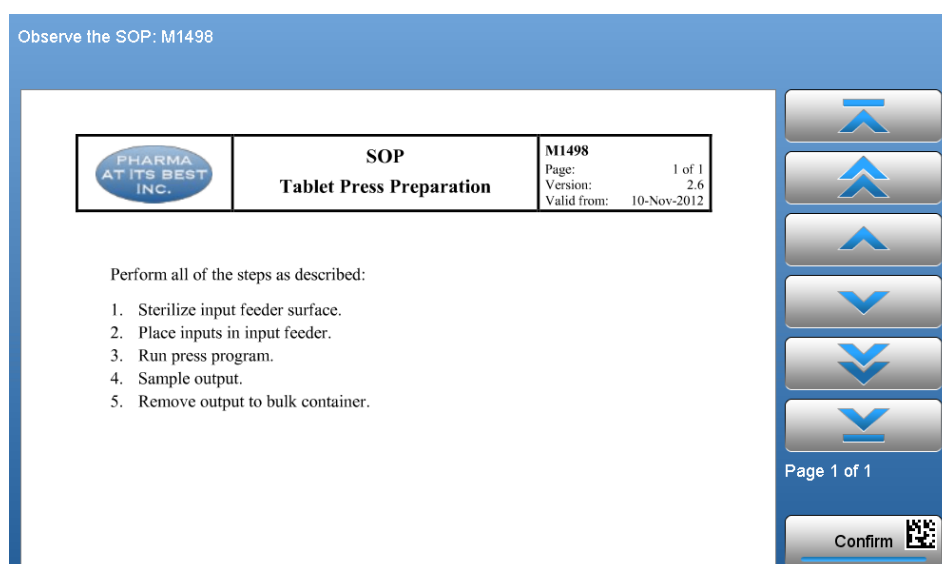


Figure 22: Show URL during execution



Figure 23: Show URL after phase completion



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.

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 the calculated result of several phase outputs as value into a parameter attribute. When you reference phase outputs in this manner you need to be aware of the following restrictions:

- Only if a phase has been processed does it provide an output that can be fed into another phase as attribute value. For this reason, you can never reference an output of a phase that is a strict successor of the phase 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 phase to 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".

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.

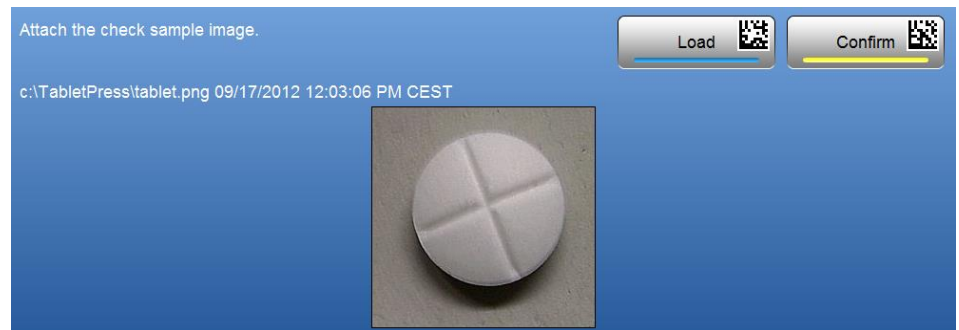


Figure 25: Upload image during execution - Automatic loading mode



Figure 26: Upload image during execution - Selection mode



Figure 27: User-triggered exception of Upload image

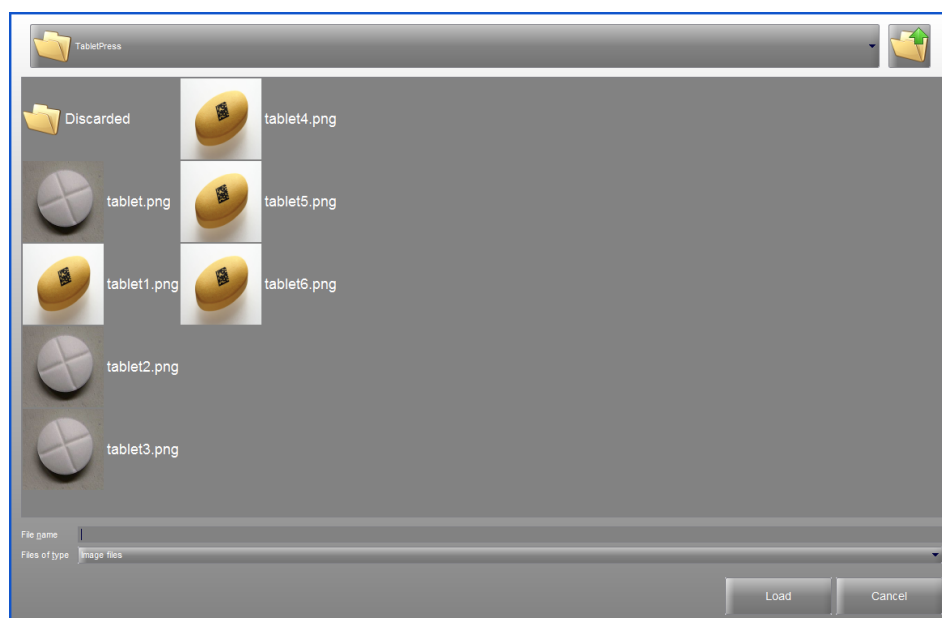


Figure 28: File selection dialog of Upload image

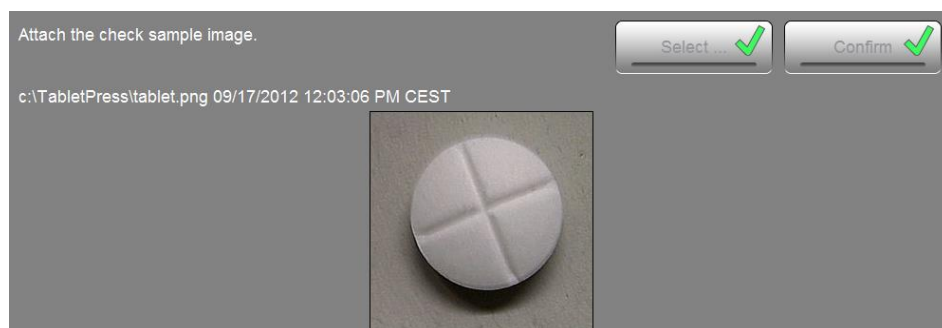


Figure 29: Upload image after phase completion



Figure 30: Upload image in the Navigator

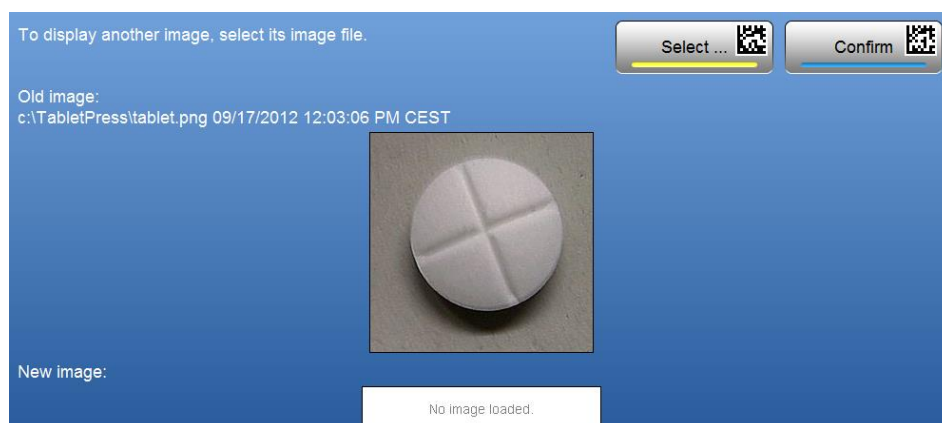


Figure 31: Post-completion exception of Upload image

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 51).

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

File location

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

Attribute	Type	Comment
Directory path	Text	<p>Defines the directory path of the image file to be loaded. Environment variables are supported (e.g. %USERNAME%).</p> <p>If the Mode process parameter (page 51) is set to Selection, the system defaults the file selection dialog to this path. The operator can navigate to another directory.</p>

Attribute	Type	Comment
File name	Text	<p>Defines the name of the image file to be loaded.</p> <p>If the Mode process parameter (page 51) 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.</p> <p>Example: img*.jpg displays any image files that start with img.</p> <p>For all other options available with the Mode process parameter (page 51), the file name must be unique.</p>

Mandatory upload check

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

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

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.

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 the calculated result of several phase outputs as value into a parameter attribute. When you reference phase outputs in this manner you need to be aware of the following restrictions:

- Only if a phase has been processed does it provide an output that can be fed into another phase as attribute value. For this reason, you can never reference an output of a phase that is a strict successor of the phase 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 phase to 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"**.

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 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 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.

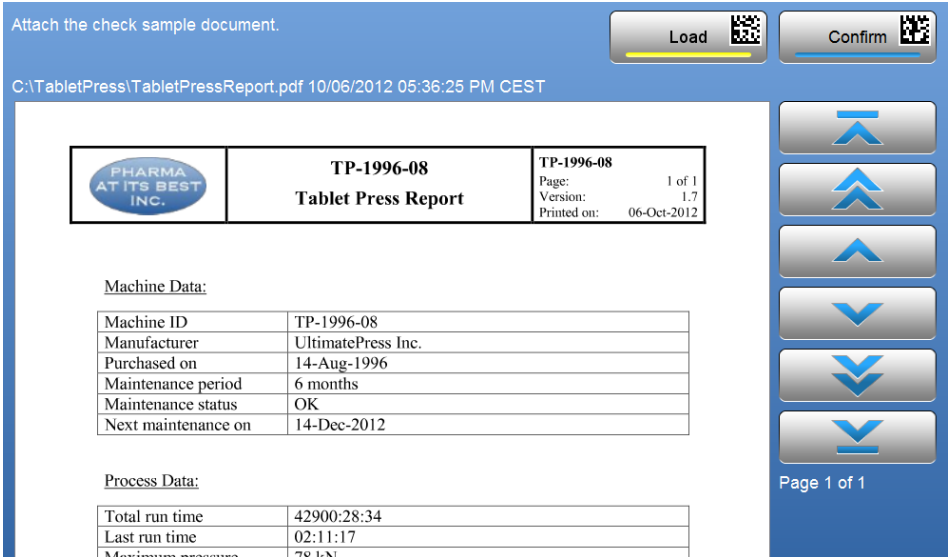


Figure 32: Upload PDF during execution - Automatic loading mode




Figure 33: Upload PDF during execution - Selection mode

To display another document, select its PDF file.

Old PDF file:
C:\TabletPress\TabletPressReport.pdf 10/06/2012 05:36:25 PM CEST

Select ... Confirm



TP-1996-08	TP-1996-08
Tablet Press Report	Page: 1 of 1 Version: 1.7 Printed on: 06-Oct-2012

Machine Data:

Machine ID	TP-1996-08
Manufacturer	UltimatePress Inc.
Purchased on	14-Aug-1996
Maintenance period	6 months
Maintenance status	OK
Next maintenance on	14-Dec-2012

Process Data:

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

New PDF file:

No PDF file loaded.

Page 1 of 1

Figure 34: User-triggered exception of Upload PDF

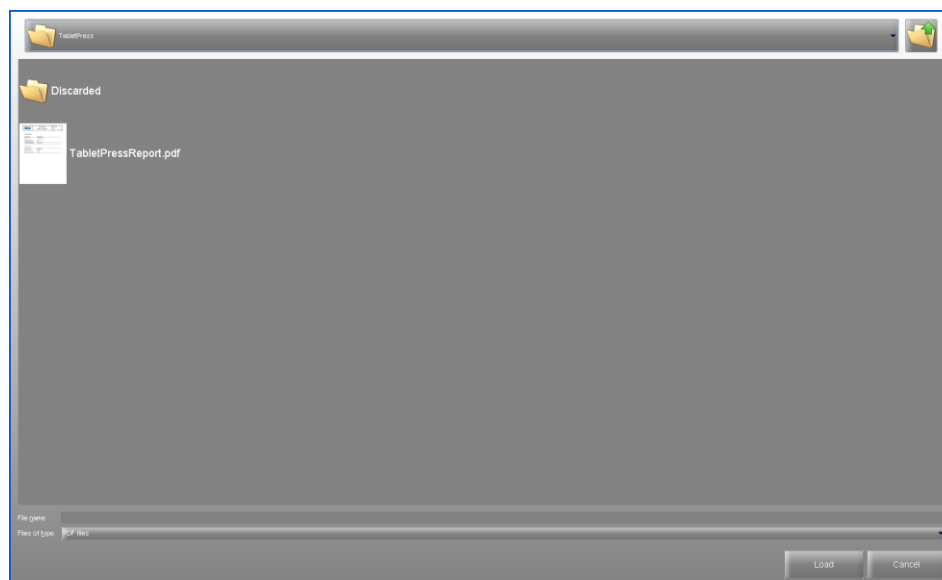


Figure 35: File selection dialog of Upload PDF

Attach the check sample document.

Load  Confirm 

C:\TabletPress\TabletPressReport.pdf 10/06/2012 05:36:25 PM CEST

	TP-1996-08 Tablet Press Report	TP-1996-08 Page: 1 of 1 Version: 1.7 Printed on: 06-Oct-2012
---	---	--

Machine Data:

Machine ID	TP-1996-08
Manufacturer	UltimatePress Inc.
Purchased on	14-Aug-1996
Maintenance period	6 months
Maintenance status	OK
Next maintenance on	14-Dec-2012

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 1

Figure 36: Upload PDF after phase completion



Verification
Verify Output (Doc) (2)

TabletPressReport.pdf


Replace

Figure 37: Upload PDF in the Navigator

To display another document, select its PDF file.

Select ...  Confirm 

Old PDF file:
C:\TabletPress\TabletPressReport.pdf 10/06/2012 05:36:25 PM CEST

	TP-1996-08 Tablet Press Report	TP-1996-08 Page: 1 of 1 Version: 1.7 Printed on: 06-Oct-2012
---	---	--

Machine Data:

Machine ID	TP-1996-08
Manufacturer	UltimatePress Inc.
Purchased on	14-Aug-1996
Maintenance period	6 months
Maintenance status	OK
Next maintenance on	14-Dec-2012

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 1

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.

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 62).

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

File location

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

Attribute	Type	Comment
Directory path	Text	<p>Defines the directory path of the image file to be loaded. Environment variables are supported (e.g. %USERNAME%).</p> <p>If the Mode process parameter (page 62) is set to Selection, the system defaults the file selection dialog to this path. The operator can navigate to another directory.</p>

Attribute	Type	Comment
File name	Text	<p>Defines the name of the image file to be loaded.</p> <p>If the Mode process parameter (page 62) 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.</p> <p>Example: rep*.pdf displays all PDF files starting with rep.</p> <p>For all other options available with the Mode process parameter (page 62), the file name must be unique.</p>

Mandatory upload check

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

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

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.

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 the calculated result of several phase outputs as value into a parameter attribute. When you reference phase outputs in this manner you need to be aware of the following restrictions:

- Only if a phase has been processed does it provide an output that can be fed into another phase as attribute value. For this reason, you can never reference an output of a phase that is a strict successor of the phase 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 phase to 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".

C

- Completion time (output)
 - Get choice value • 9
 - Get process value • 20
 - Get text value • 29
 - Show document • 35
 - Show instruction text • 40
 - Show URL • 45
 - Upload image • 55
 - Upload PDF • 66
- Conventions (typographical) • 2
- Correct value (Get choice value, parameter) • 7

D

- Document (parameter)
 - Show document • 33
 - Show URL • 43

E

- Expected value configuration (parameter)
 - Get choice value • 6
 - Get text value • 26
- Expected value definition (parameter)
 - Get choice value • 7
 - Get text value • 27

F

- File location (parameter)
 - Upload image • 51
 - Upload PDF • 62

G

- Get choice value • 3
 - Output variable • 8
 - Process parameter • 5
- Get process value • 11
 - Output variable • 19

- Process parameter • 13

- Get text value • 23
 - Output variable • 28
 - Process parameter • 25

I

- Identifier (output)
 - Get choice value • 8
 - Get process value • 19
 - Get text value • 29
 - Show document • 34
 - Show instruction text • 39
 - Show URL • 44
 - Upload image • 54
 - Upload PDF • 65
- Image full path (Upload image, output) • 54
- Image timestamp (Upload image, output) • 54
- Instance count (output)
 - Get choice value • 9
 - Get process value • 20
 - Get text value • 29
 - Show document • 34
 - Show instruction text • 39
 - Show URL • 44
 - Upload image • 55
 - Upload PDF • 66
- Instruction (parameter)
 - Get choice value • 5
 - Get process value • 13
 - Get text value • 25
 - Show document • 33
 - Show URL • 42
 - Upload image • 50
 - Upload PDF • 61
- Instruction (Show instruction text, parameter) • 38

L

- Layout (Show instruction text, parameter) • 38

Limit configuration (Get process value, parameter) • 14
 Limit definition (Get process value, parameter) • 16
 List of options (Get choice value, parameter) • 5

M

Mandatory upload check (parameter)
 Upload image • 52
 Upload PDF • 63
 Mode (parameter)
 Upload image • 51
 Upload PDF • 62

O

One-click completion (Get choice value, parameter) • 7
 Option key (Get choice value, output) • 8
 Option text (Get choice value, output) • 8
 Output value (Get text value, output) • 29
 Output variable
 Completion time (Get choice value) • 9
 Completion time (Get process value) • 20
 Completion time (Get text value) • 29
 Completion time (Show document) • 35
 Completion time (Show instruction text) • 40
 Completion time (Show URL) • 45
 Completion time (Upload image) • 55
 Completion time (Upload PDF) • 66
 Get choice value • 8
 Get process value • 19
 Get text value • 28
 Identifier (Get choice value) • 8
 Identifier (Get process value) • 19
 Identifier (Get text value) • 29
 Identifier (Show document) • 34
 Identifier (Show instruction text) • 39
 Identifier (Show URL) • 44
 Identifier (Upload Image) • 54
 Identifier (Upload PDF) • 65
 Image full path (Upload image) • 54
 Image timestamp (Upload image) • 54
 Instance count (Get choice value) • 9
 Instance count (Get process value) • 20
 Instance count (Get text value) • 29

Instance count (Show document) • 34
 Instance count (Show instruction text) • 39
 Instance count (Show URL) • 44
 Instance count (Upload image) • 55
 Instance count (Upload PDF) • 66
 Option key (Get choice value) • 8
 Option text (Get choice value) • 8
 Output value (Get text value) • 29
 PDF full path (Upload PDF) • 65
 PDF timestamp (Upload PDF) • 65
 Show document • 34
 Show instruction text • 39
 Show URL • 44
 Start time (Get choice value) • 9
 Start time (Get process value) • 20
 Start time (Get text value) • 29
 Start time (Show document) • 35
 Start time (Show instruction text) • 40
 Start time (Show URL) • 45
 Start time (Upload image) • 55
 Start time (Upload PDF) • 66
 Unit of measure (Get process value) • 19
 Upload image • 54
 Upload PDF • 65
 Value (Get process value) • 19
 Override value (Get process value, parameter) • 18

P

PDF full path (Upload PF, output) • 65
 PDF timestamp (Upload PDF, output) • 65
 Post - Correct value (parameter)
 Get process value • 18
 Get text value • 27
 Process parameter
 Correct value (Get choice value) • 7
 Document (Show document) • 33
 Document (Show URL) • 43
 Expected value configuration (Get choice value) • 6
 Expected value configuration (Get text value) • 26
 Expected value definition (Get choice value) • 7
 Expected value definition (Get text value) • 27

File location (Upload image) • 51
Get choice value • 5
Get process value • 13
Get text value • 25
Image location (Upload PDF) • 62
Instruction (Get choice value) • 5
Instruction (Get process value) • 13
Instruction (Get text value) • 25
Instruction (Show document) • 33
Instruction (Show instruction text) • 38
Instruction (Show URL) • 42
Instruction (Upload image) • 50
Instruction (Upload PDF) • 61
Layout (Show instruction text) • 38
Limit configuration (Get process value) • 14
Limit definition (Get process value) • 16
List of options (Get choice value) • 5
Mandatory upload check (Upload image) • 52
Mandatory upload check (Upload PDF) • 63
Mode (Upload image) • 51
Mode (Upload PDF) • 62
One-click completion (Get choice value) • 7
Override value (Get process value) • 18
Post - Correct value (Get process value) • 18
Post - Correct value (Get text value) • 27
Replace file (Upload image) • 53
Replace file (Upload PDF) • 64
Select manually (Upload image) • 53
Select manually (Upload PDF) • 64
Show document • 33
Show instruction text • 38
Show URL • 42
Upload image • 50
Upload PDF • 61
Value configuration (Get process value) • 14

R

Replace file (parameter)
Upload image • 53
Upload PDF • 64

S

Select manually (parameter)
Upload image • 53
Upload PDF • 64
Show document • 31
Output variable • 34
Process parameter • 33
Show instruction text • 37
Output variable • 39
Process parameter • 38
Show URL • 41
Output variable • 44
Process parameter • 42
Start time (output)
Get choice value • 9
Get process value • 20
Get text value • 29
Show document • 35
Show instruction text • 40
Show URL • 45
Upload image • 55
Upload PDF • 66

U

Unit of measure (Get process value, output) • 19
Upload image • 47
Output variable • 54
Process parameter • 50
Upload PDF • 57
Output variable • 65
Process parameter • 61

V

Value (Get process value, output) • 19
Value configuration (Get process value, parameter) • 14