

Working with Variables

Environment variables give you a centralized way to define, review, and modify values that are used in multiple places across CTP. For example, assume you have a path that is used multiple times. If you define it as a literal value and later want to change that path value, you will have to change it numerous times. If you define it as a variable, you can update it once, and your changes will be automatically propagated to all relevant areas.

Targeted Variable Management

You can configure variables at the system, environment, component, or component instance level. To add a variable at a specific level:

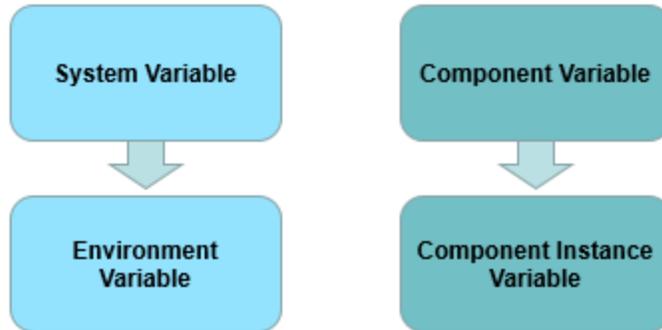
1. Go to the appropriate level variable table.
2. If needed, make the variable table editable.
3. Click the + icon at the bottom of the variables table.
4. Specify the new variables, then save your changes.

Global Variable Management

If you prefer to review, add, and modify all variables related to a specific environment from a centralized page, you can do so from the Manage Instances page, which is described in [Managing Component Instances](#).

Variable Inheritance

Component instance variables can inherit values from component variables. Environment variables can inherit from system variables. Environment and system variable values are not connected to component and component instance variable values.



All variable names are visible at the system level and shared between all environments on the same system. All variable names visible at the component level are shared between all instances of the same component.

Overriding Variables

Each lower level entity can override values from the upper level entity. For example, if the PATH variable was defined at the system level, you could change its value (but not the variable name) and the environment level. In addition, you can define new environment-level variables that are not present at the system level; those variable names will be added to the system without values.

Any variables that are not overridden will inherit their values from the system (if looking at variables for an environment) or component (if looking at variables for a component instance).

Component Instance Variables

Override	Name	Value
<input type="checkbox"/>	Version	
<input checked="" type="checkbox"/>	ENDPOINT	http://ws1.parasoft.com/glue/store-01
<input checked="" type="checkbox"/>	WSDL	http://soatest.parasoft.com/store-01.wsdl
<input type="checkbox"/>	ENDPOINT_ORIGINAL	http://ws1.parasoft.com/glue/calculator

From the Environment Information Panel

To override an inherited variable value from the environment information panel:

- Add the new value in the text field.

EM Demo ParaBank

Here's a description of my environment.

Tags: + -

tag1 tag2

Variables:

Key	Value
Password
Something_to_override	123456
Version	1.0

If you later want to remove return to the inherited variable (e.g., remove the override), click in that field, then click the **Revert** button in the right of that field.

Variables:

Key	Value
Password
Something_to_override	12345678 ↺
Version	1.0

Revert to value inherited from system

From the Manage Instances Page

To override an inherited variable value from the manage instances page (described in [Managing Component Instances](#)):

1. Open the appropriate variable editor. (if it is not visible, click **Show details** or **Expand all**), then click the **Edit** button.

SOAtest Test Executor

SOAtest Test Executor

▶ Variables 

▶ Instances (2) 

SOAtest Test Executor Component # 1 Show

SOAtest Test Executor

▼ Variables  

Key	Default value
ENDPOINT	http://ws1.parasoft.com/glue/store-01
Version	
WSDL	http://soatest.parasoft.com/store-01.wsdl

2. Specify the desired values.
3. If you want to see a modified value rendered as plain text before you save the changes, click the **Apply** icon.

SOAtest Test Executor

SOAtest Test Executor

▼ Variables  

Key 

From the Component Instance Wizard

Variable values for a specific component instance can be overridden in the Variables page of the component instance wizard:

Component Instance Variables

Override	Name	Value
<input type="checkbox"/>	Version	
<input checked="" type="checkbox"/>	ENDPOINT	http://ws1.parasoft.com/glue/store-01
<input type="checkbox"/>	WSDL	http://soatest.parasoft.com/store-01.wsdl
<input type="checkbox"/>	ENDPOINT_ORIGINAL	http://ws1.parasoft.com/glue/calculator
<input type="checkbox"/>	WSDL_1	http://soaptest.parasoft.com/calculator.wsdl
<input type="checkbox"/>	ENDPOINT_1	http://dragon.parasoft.com:9080/calculator_proxy
<input type="checkbox"/>	env_var	value
<input type="checkbox"/>	ENDPOINT_2	http://parabank.parasoft.com/parabank/services/store-01
<input type="checkbox"/>	WSDL_2	http://parabank.parasoft.com/parabank/services/store-01

Note that any overrides made at this point will apply only to this specific component instance.

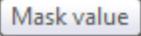
Masking/Unmasking Variables

When variable values are editable, you can mask or unmask them (if they were not yet saved).

In system/environment information panels, you can mask a value by clicking in that field, then clicking the **Mask** icon in the right of the field.

Variables:  

Key	Value
Password	blahblah 
Version	1.0



In other areas, you can mask a value by using the key icon to the right of the variable editor.

Variables Table:

Name	Value
ENDPOINT	http://parabank.parasoft.com/parabank/servic... 
PASSWORD 
USERNAME	admin 

The exception is when editing variables that are inherited from SOAtest tests or within CTP (e.g., environment variables inherited from a system). These variables will inherit the parent's masking setting. If the parent variable is masked, it cannot be unmasked at the inherited level. If you want to unmask it, you must do so at the parent level.