

# RES Wisdom 2009 Feature Profile

## Parameters

Parameters are placeholders in a Task for the values in various fields, such as text, a file path, credentials, etc. The actual value of that field is then provided when the Task is used in a Job, or imported as a Building Block. Parameters have a number of uses:

- Configure parts of a Task to remain the same, and use parameters to allow other parts of the Task to be customized when it is actually used. For example, the settings with which a local user is created remain the same, but the Task is scheduled with a different user name and full name each time. This makes it possible to create generic Modules, Projects and Run Books that can be customized at the moment of use or when imported as a Building Block.
- Allow users with a specific Security Role to schedule Jobs and customize some fields, but prevent them from changing the standard configured fields.
- Restrict the type of input provided at the moment of use. For example, use a List parameter to present a list of options from which to choose, making it impossible to enter other data in that field. As a result, you can control the kind of information with which a Task is performed.
- Information may need to be repeated in several fields in Modules and Projects contained in a Project or Run Book. For example, a Project can contain 4 Modules that all require the same user name to be filled out. By inserting linked parameters in those user name fields, input will be requested once and will fill all fields.
- Provide default values for certain fields, with the option to overrule these defaults at the moment of use.
- Use a parameter as the basis for a Condition, so that the value of the parameter determines whether a Module, Project or Run Book should be executed or not. For example, you can configure a Condition so that a Module is skipped if the person scheduling it does not provide the correct value for a password parameter.

This Feature Profile on RES Wisdom parameters explains the different types of parameters and their use. For configuration guidance, please refer to the RES Wisdom Help, available from the RES Wisdom Management Console.

### **GENERIC MODULES, PROJECTS AND RUN BOOKS**

Parameters can be used to create Modules, Projects and Run Books that can be customized when they are scheduled as a Job, or when they are imported as part of a Building Block.

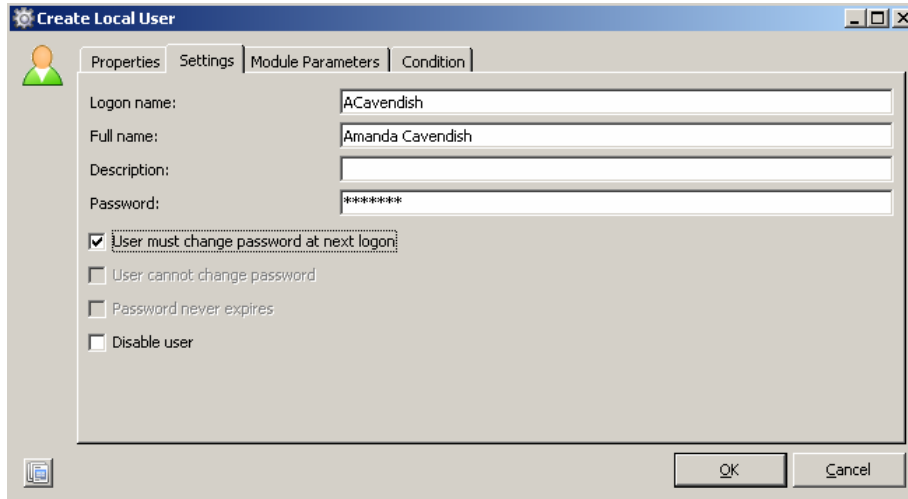
#### **Customizing a Task when it is scheduled as a Job**

With parameters, Tasks can be configured flexibly, so that they can be re-used efficiently.

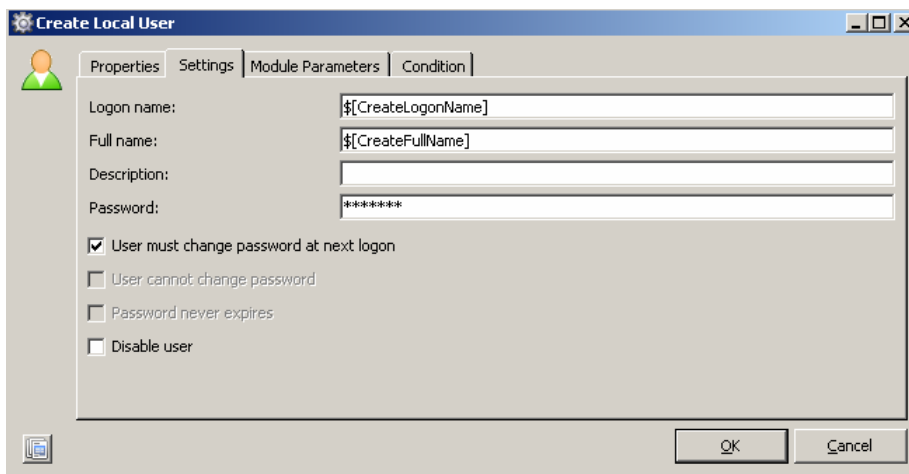
In any RES Wisdom environment, there will be Tasks that are used many times over, with just minor changes to their configuration. For example, you may have a Module to create

a new local user with the Task **Create Local User**. Each time you use this Module, the password and password settings remain the same, but you provide a different user name.

For example, to create a new local user ACavendish, you could configure the Task as follows:



The next day, you need to create the local user ELeonard. You *could* edit the original Module, save it, and schedule it. Next time you have to create a local user, you do the same again. However, it is more efficient to insert parameters in the Task:



Each parameter must have the **Input** setting **Input new value when scheduling Job** (which is the default, found on the parameters' **Input** tabs). Now, when you need to create a new local user, you simply schedule the Module. RES Wisdom will then prompt you to provide the logon name and full name with which the Task should be executed.



#### Tip

When scheduling a Job, the option is available to **Use parameter values from file**. This makes it possible to import parameter values from a CSV file, instead of entering these values manually. This makes it possible to input parameter values in bulk into large numbers of Tasks. See the RES Wisdom Help (available from the Management Console) for more information.

## Customizing a Task when it is imported as part of a Building Block

Building Blocks can transport Modules, Projects and Run Books from one RES Wisdom environment to another. This is useful, for example, for purposes of system integration. In such cases, the configuration of some Tasks may need to be changed to the new environment. This can be achieved efficiently using parameters.

For example, an imported Run Book is likely to contain Tasks that require administrator credentials in their Security Context fields. Those credentials will need to be changed for the receiving environment. This can be set up efficiently with parameters: replace the Security Contexts in the original Tasks with credentials parameters, all linked to one Run Book parameter for these credentials. This parameter must have the parameter **Input** setting **When importing Building Block** (which is the default, found on the parameters' **Input** tabs). Then, each time the Run Book is transported to a different environment in a Building Block, RES Wisdom will prompt for a new value for the Run Book parameter for the administrator credentials.



### Tip

By default, input is requested when a Job is scheduled, and when a Building Block is imported. However, you may want new values to be requested when the parameter is imported as part of a Building Block, but not when it is used in a scheduled Job. To achieve this, provide a default value for the credentials parameter and clear the input option **When scheduling Job**.

## SECURE DELEGATION

With parameters, control over the execution and configuration of Tasks can be delegated, so that specific users can schedule a Job and provide the parameter values, but cannot change the rest of the configuration of that Task.

As a result, for example, a specific user scheduling the Job "Create Local User" can provide a user name for the new user, but cannot change the fixed configuration of the password and its settings.

### Configuration

- Configure the Tasks with parameters in the fields that should remain editable for users scheduling Jobs.
- Create a Security Role with read-only access to **Modules, Projects, Run Books, Agents** and **Teams**, and full access to **Job Execution**.
- Assign this Security Role to users. These users will be able to schedule Modules, Projects and Run Books, and can provide values for all parameters in the input window. However, they cannot edit any of the other fields.

Security Roles and Users are configured at **Infrastructure > Datastore > Security**. For specific configuration guidance, please refer to the RES Wisdom Help, which is available from the RES Wisdom Management Console.

## CONTROLLED INPUT IN TASK FIELDS

Instead of simply allowing any text to be input as a parameter value at the moment of use, a number of parameter types can be used to determine what kind of input is allowed. As a result, you can control the kind of information with which a Task is performed:

- with a **list** parameter, the value can be restricted to one from a list of provided options. For example, in a Task **Execute SQL Statement**, you could provide a list of Database servers.
- with a **multi-select list** parameter, the value can be restricted to one or more from a list of provided options. For example, in a Task **Manage Active Directory User**, you could provide a list of Active Directory Groups of which the user can be made member.
- with a **multi-line text** parameter, the value can be a text of several lines, containing line breaks. This can be useful, for example, for a message box text, Active Directory User Notes, or for the value for a multi-string Registry value.
- the input provided for **text** parameters can be restricted with the input option **Use input mask**. For example, the text field for an Active Directory User's telephone number can be restricted to numbers in a set pattern such as (000) 000-0000. This will result in the following prompt: (\_\_\_\_) \_\_\_\_-\_\_\_\_, in which only numbers can be entered. The mask options are described in the RES Wisdom Help.

## UNIFORM INPUT FOR MULTIPLE, LINKED PARAMETERS

### Module parameters

Often, sets of Tasks are executed that contain the same information in various fields. For example, several Tasks in a Module can refer to the same user name. Instead of repeating the same information in all those fields each time the Module is used, the fields can be filled using one Module parameter. Input is requested once, and then used in all those fields.

### Project and Run Book parameters

When several Modules are combined into a Project, the Module parameters will continue to function. However, several of those Modules may have fields that require the same information, such as a user name. In such cases, you can create a Project parameter and link it to the Module parameters that should all receive the same input. With such a linked Project parameter, input is requested only once, and fills all the fields containing one of the linked Module parameters.

For example, a Project could contain four Modules that all require the same user name to be filled out. By filling the Task fields with a Module parameter, and then creating a Project parameter linked to those Module parameters, all four fields can be filled with the value provided for the single Project parameter.

Similarly, a Run Book may contain several Modules and Projects that share information, which can be filled efficiently by using a Run Book parameter linked to the relevant Module and Project parameters.

## USING PARAMETER VALUES TO SKIP OR EXECUTE MODULES, PROJECTS AND JOBS

Conditions can be set on Modules, Projects and Run Books, to determine whether a specific part should be failed, executed or skipped, and whether any following parts should be executed or skipped. One of the things on which Conditions can be based, are parameters.

For example, the execution of the first Task in a Module can have a Condition based on the value of a Password parameter. If the person scheduling the Job does not provide the correct value for this parameter, then this Task and all remaining Tasks in the Module should be skipped. This effectively protects the Module from execution by unauthorized people who do not know the correct password.

Another example is a Module containing two identical Tasks to create an Exchange Mailbox for a user. The only difference between the two Tasks is that one is configured to create the mailbox on mail server A, and the other is configured to do so on mail server B. The Module Condition is set on the value of the parameter `$(surname)` that is used in the Task. If the surname of the user starts with A-M, the first Task must be executed and the second one skipped. If the surname starts with N-Z, the first Task must be skipped and the second one executed. (The pattern for this is: `LIKE [a-m]*` for one Task, and `LIKE [n-z]*` for the other.)



### Note

Besides determining whether specific parts of a Module, Project or Run Book should be skipped, failed or executed, a Condition can also be used to set a different value for a specific parameter. This makes it possible to change a parameter value without user input, depending on one or more of the many available Condition expressions.

### Disclaimer

Whilst every care has been taken by RES Software to ensure that the information contained in this publication is correct and complete, it is possible that this is not the case. RES Software provides the publication "as is", without any warranty for its soundness, suitability for a different purpose or otherwise. RES Software is not liable for any damage which has occurred or may occur as a result of or in any respect related to the use of this publication. RES Software may change or terminate this publication at any time without further notice and shall not be responsible for any consequence(s) arising there from. Subject to this disclaimer, RES Software is not responsible for any contributions by third parties to this publication.

### Copyright Notice

Copyright © 1998-2008 RES Software, The Netherlands. RES®, PowerFuse®, Wisdom®, Orchestra®, Insight® and the RES logo are either registered trademarks or trademarks of RES Software in Europe, the United States and other countries. Microsoft and Windows are either registered trademarks of Microsoft Corporation in the United States and/or other countries. All other product and company names mentioned may be trademarks and/or service marks of their respective owners.

Copyright © RES manuals, training materials and software 1998-2008 Real Enterprise Solutions Development BV, The Netherlands. Patents Pending.

Any rights not expressly granted herein are reserved by RES Software or Real Enterprise Solutions Development BV.