Copyright © RES Software Development B.V. All rights reserved. Commercial Computer Software documentation/data - Restricted Rights. RES ® and RES ONE ® are registered trademarks and service marks of RES Software B.V. internationally. The software licensed by RES Software B.V. or RES Software, Inc. is covered by patents, any patents pending, granted to and/or owned by RES Software Development B.V. and as identified on www.res.com/legal-statements.

Disclaimer

While care has been taken by RES to ensure that the information contained in this document is correct and complete, it is possible that this is not the case. RES provides the information “as is”, without any warranty of any kind. To the maximum extent permitted by applicable law, RES 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 information. RES may change or remove this document at any time without notice and shall not be responsible for any consequence(s) arising therefrom. RES is not responsible for any contributions by third parties to this information.
Contents

Chapter 1: Introduction 1
1.1 RES Training ......................................................... 2
1.2 RES Support ......................................................... 3

Chapter 2: About RES ONE Automation 4
2.1 Components ................................................... 5
2.2 Architecture ................................................... 6
2.3 Licensing ....................................................... 8
2.4 Video Tutorials .................................................. 9

Chapter 3: Setup 11
3.1 Installation procedure ........................................ 11
3.1.1 Prerequisites ................................................. 12
3.2 Database ......................................................... 15
3.2.1 Microsoft SQL Server .................................. 16
3.2.2 Microsoft Azure SQL .................................. 20
3.2.3 Oracle ......................................................... 20
3.2.4 IBM DB2 ....................................................... 21
3.2.5 MySQL ......................................................... 21
3.2.6 Encryption using AES-256 ......................... 22
3.2.7 Install database drivers using RES ONE Automation 22
3.3 Licensing ......................................................... 23
3.3.1 Pre-2014 Licensing Model ......................... 25
3.3.2 Licensing process ........................................ 26
3.3.3 License server registration process .............. 28
3.3.4 License usage collection and reporting ........ 29
3.4 RES ONE Identity Director ............................. 31
3.5 PoSh Automation .............................................. 32
3.6 Task Visibility .................................................. 34
3.7 Add-ons .......................................................... 34
3.8 Global Settings ................................................ 35
3.8.1 Site settings ................................................ 37
3.8.2 Dispatcher Detection settings .................. 38
3.8.3 Connection settings .................................. 40
3.8.4 Dispatcher Proxy settings ......................... 41
3.8.5 Dispatcher Cache settings ......................... 41
3.8.6 Dispatcher WebAPI settings ..................... 42
3.8.7 Agent Cache settings .................................. 42
3.8.8 Scheduling settings ..................................... 43
3.8.9 Job Notification settings ......................... 45
3.8.10 Security settings ...................................... 45
3.8.11 Snapshot Intelligence settings ................. 47
3.8.12 Version Control settings ......................... 48
3.8.13 Password Security Policy settings ............ 48
3.8.14 Other Settings ............................................ 50

Chapter 4: Jobs 52
4.1 Scheduling ....................................................... 52

Copyright © RES Software Development B.V. All rights reserved.
<table>
<thead>
<tr>
<th>Chapter 6: Topology</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1 Consoles</td>
</tr>
<tr>
<td>6.2 Dispatchers</td>
</tr>
<tr>
<td>6.2.1 Dispatcher settings</td>
</tr>
<tr>
<td>6.2.2 Master caching</td>
</tr>
<tr>
<td>6.2.3 Web API</td>
</tr>
<tr>
<td>6.2.4 Housekeeping tasks of Dispatcher</td>
</tr>
<tr>
<td>6.2.5 Reporting to RES ONE License Server</td>
</tr>
<tr>
<td>6.2.6 Telemetry tasks of Dispatcher</td>
</tr>
<tr>
<td>6.3 Agents</td>
</tr>
<tr>
<td>6.3.1 Agent settings</td>
</tr>
<tr>
<td>6.3.2 Agent Variables</td>
</tr>
<tr>
<td>6.3.3 Agent Trusts</td>
</tr>
<tr>
<td>6.3.4 Open a remote Console on an Agent</td>
</tr>
<tr>
<td>6.3.5 Install Agents embedded using Prepare for Image</td>
</tr>
<tr>
<td>6.3.6 Install Agents embedded using Prepared4Embedded</td>
</tr>
<tr>
<td>6.3.7 Snapshot Intelligence</td>
</tr>
<tr>
<td>6.3.8 Agent+</td>
</tr>
<tr>
<td>6.3.9 RES ONE Automation Agents for Unix/Linux</td>
</tr>
<tr>
<td>6.3.10 Agents for Apple Mac OS X</td>
</tr>
<tr>
<td>6.3.11 Bare Metal OS deployment using WDS and RES ONE Automation</td>
</tr>
<tr>
<td>6.4 Teams</td>
</tr>
<tr>
<td>6.4.1 Team Folders</td>
</tr>
<tr>
<td>6.4.2 Team settings</td>
</tr>
<tr>
<td>6.4.3 Team Variables</td>
</tr>
<tr>
<td>6.4.4 Team members</td>
</tr>
<tr>
<td>6.4.5 Team Trusts</td>
</tr>
<tr>
<td>6.4.6 Team Permissions</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 7: Administration</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1 Information</td>
</tr>
<tr>
<td>7.2 Security</td>
</tr>
<tr>
<td>7.2.1 Login accounts</td>
</tr>
<tr>
<td>7.2.2 Change my password</td>
</tr>
<tr>
<td>7.2.3 Administrative Roles</td>
</tr>
<tr>
<td>7.2.4 Assigning administrative roles to login accounts</td>
</tr>
<tr>
<td>7.2.5 Secure delegation</td>
</tr>
<tr>
<td>7.3 Components</td>
</tr>
<tr>
<td>7.3.1 Saving components as MSI files</td>
</tr>
<tr>
<td>7.3.2 Upgrading your RES ONE Automation environment</td>
</tr>
<tr>
<td>7.4 Audit Trail</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 8: General functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.1 Building Blocks</td>
</tr>
<tr>
<td>8.1.1 Creating Building Blocks</td>
</tr>
<tr>
<td>8.1.2 Importing Building Blocks</td>
</tr>
<tr>
<td>8.1.3 The structure of Building Blocks</td>
</tr>
<tr>
<td>8.2 Command Bar</td>
</tr>
<tr>
<td>8.3 Conditions</td>
</tr>
<tr>
<td>8.3.1 Settings</td>
</tr>
<tr>
<td>8.3.2 Conditions based on existence of file or folder</td>
</tr>
<tr>
<td>8.3.3 Conditions based on parameters</td>
</tr>
<tr>
<td>8.3.4 Conditions based on registry settings</td>
</tr>
<tr>
<td>8.4 Evaluators</td>
</tr>
<tr>
<td>8.4.1 Settings</td>
</tr>
<tr>
<td>8.4.2 Scenario: Install applications only on Agents with sufficient disk space</td>
</tr>
<tr>
<td>8.4.3 Scenario: Add Active Directory users to Distribution Lists</td>
</tr>
</tbody>
</table>
8.5 Functions ........................................................................................................... 246
8.5.1 Overview ..................................................................................................... 247
8.6 Instant Reports ................................................................................................. 255
8.6.1 Where to find Instant Reports in the Console ......................................................... 255
8.6.2 Instant Report format .................................................................................. 255
8.6.3 Customizing the contents and presentation .................................................. 256
8.7 Parameters ......................................................................................................... 257
8.7.1 Where to find parameters .......................................................................... 258
8.7.2 Parameter types .......................................................................................... 259
8.7.3 Input settings ............................................................................................... 260
8.7.4 Input Mask .................................................................................................. 261
8.7.5 Creating parameters automatically ............................................................. 263
8.7.6 Linking parameters ..................................................................................... 264
8.7.7 Securing Modules, Projects and Run Books using parameters ....................... 266
8.8 Pattern matching, wildcards and operators ..................................................... 267
8.9 Trusts ................................................................................................................ 269
8.9.1 Where to find Trusts in the Console ............................................................ 270
8.9.2 Configuring Trusts ..................................................................................... 271
8.10 Version Control ................................................................................................ 272

Chapter 9: Appendixes ........................................................................................................... 274

9.1 Best Practices ............................................................................................................. 274
9.1.1 Bare Metal OS Deployment Using WDS and RES ONE Automation ..................... 274
9.2 Command-line options .................................................................................... 287
9.2.1 Adding or editing a Resource ........................................................................ 289
9.2.2 Changing Agent connection settings ........................................................... 291
9.2.3 Changing Dispatcher connection settings .................................................. 291
9.2.4 Creating a Building Block for a Run Book .................................................... 292
9.2.5 Exporting Job Results ................................................................................ 293
9.2.6 Exporting licenses ...................................................................................... 294
9.2.7 Importing Building Blocks ........................................................................... 295
9.2.8 Installing Agents ........................................................................................ 296
9.2.9 Installing Consoles ..................................................................................... 298
9.2.10 Installing Dispatchers ................................................................................ 299
9.2.11 Installing RES ONE Automation ............................................................... 300
9.2.12 Scheduling Jobs ......................................................................................... 303
9.2.13 Solving a Console lockout .......................................................................... 304
9.3 Registry settings ..................................................................................................... 305
9.3.1 Frequency at which Console checks for new updates ..................................... 305
9.3.2 Location of Dispatcher Resource cache folder .............................................. 305
9.3.3 Master caching Dispatcher .......................................................................... 306
9.3.4 Tracing ......................................................................................................... 307
9.4 Logging .............................................................................................................. 309
9.4.1 Event logs ..................................................................................................... 309
9.4.2 Telemetry ...................................................................................................... 312
9.5 Complete list of collected data provided to RES .................................................. 312
9.6 Flowcharts .......................................................................................................... 313
9.6.1 Dispatcher Discovery ................................................................................... 313
9.7 RES ONE Automation Glossary ........................................................................... 314
9.8 Compatibility Matrix ........................................................................................... 316
9.8.1 Supported Microsoft Windows operating systems ....................................... 316
9.8.2 Supported Unix/Linux operating systems .................................................. 318
9.8.3 Supported Directory Services ...................................................................... 318
9.8.4 Supported Database systems ........................................................................ 319
9.8.5 Supported Mail Servers .............................................................................. 319
9.8.6 Supported Application Virtualization ............................................................ 319
9.8.7 Supported Hypervisors ................................................................................ 320
9.8.8 Supported Provisioning Services ................................................................... 320
9.8.9 Supported Microsoft Windows PowerShell Scripting .................................... 320
Chapter 1: Introduction

Welcome to the RES ONE Automation 10.2 Administration Guide. This document provides detailed information about the installation and configuration of RES ONE Automation 10.2 features and components.

For additional documents and information, please refer to our website http://res.com and to our Success Center http://success.res.com.

For feedback about the RES ONE Automation Administration Guide, please contact the RES Software documentation team at documentation@res.com.

[Generated on 28-9-2017]
1.1 RES Training

RES has developed a mix of learning materials to help our customers and channel partners get the most out of our products. Our goal is to give you a choice in how you learn; whether that is in the classroom, online tutorials and virtual workshops, or downloading our self-study kits. Please visit the Academy at RES Success Center (http://success.res.com) or go to http://res.com/support/training to find more information on Training.

**RES Academy** - RES Academy provides an engaging way to learn about RES products and technologies. It consists of short video tutorials, including practice questions, informative links and more. These tutorials cover a broad range of subjects: from planning, installing and configuring an environment to using the functionality of the RES product.

**Workshops** - For customers and partners RES organizes free interactive online workshops. These one-hour events are intended for experienced users of our software and deal with specific use cases and troubleshooting. Due to the interactive nature of these workshops, the number of seats per session is limited.

**Training Classes** - For partners and customers, RES has developed several technical courses that deal with the installation and configuration of RES ONE Workspace, RES ONE Automation and RES ONE Identity Director. These technical courses are offered by RES Authorized Learning Centers (RALCs).

**RES Certification** - RES offers a certification program designed to validate IT professionals with the technical capabilities and expertise needed to effectively use the RES product portfolio, giving companies the confidence that their IT employees have the skills and experience needed to be successful.
1.2 RES Support

At RES, our core business is to help heighten productivity in your organization. RES Support helps us to achieve this goal, and has been embedded in the core principles of our company since it was founded. RES is dedicated to supporting everyone who uses or wants to use its proven products with RES Support, which elevates our enterprise solutions above and beyond technology.

Support - If you are experiencing difficulties with any of our products, you may find the solution in our Knowledge Base (Success Center > Support) or you can contact RES Support directly (Success Center > Click Contact Us).

Product Upgrades and Service Releases - To upgrade your product to the latest standard, you can install Product Upgrade Packs from http://res.com and Service Releases from the Success Center > Downloads.

Solution Assurance - To protect your investment, it is mandatory that you purchase one initial year of Solution Assurance with each license purchase. Solution Assurance unlocks access to Technical Support, Product Updates and Upgrades and the Knowledge Base. Solution Assurance is extended automatically, unless you specify otherwise. For more information: http://res.com/support.

Early Adopter Program - Participants of the Early Adopter Program are actively involved in taking RES solutions to the next level. The Early Adopter Program unlocks access to interim releases of our products. These releases are production-ready and allow you to test drive and explore new functionality.

RES Community - RES invites you to become part of our community to share best practices and tips with fellow IT professionals, find solutions and more (Success Center > Q&A).

Please visit the RES Success Center (http://success.res.com) for more information on Support.
Chapter 2: About RES ONE Automation

RES ONE Automation increases IT productivity and efficiency by dividing and reducing the workload across all IT skill sets.

Increase Operational Efficiency
With RES ONE Automation, you can streamline complex operations and help free IT staff for strategic projects. You can increase productivity with extensive out-of-the-box knowledge while minimizing ad-hoc, unsecure and complex scripting and provide a central point of automation.

Improve Compliance and Security
With RES ONE Automation, you can control who can make changes to the IT environment with real-time audit capabilities and provide better governance. Patented trust security enables enterprises to automate with increased security and efficiency.

Gain Greater Strategic Value
RES ONE Automation saves costs by enabling quicker and more accurate deployments while leveraging existing technology investments. It eliminates single point of dependencies and empowers anyone to manage anything from simple to complex change requests, with the control required by the expert.
2.1 Components

RES ONE Automation uses the following components:

- The Console
- The Management Portal
- The Datastore
- The Dispatcher
- The Agent

**Console**

The Console is the central point of administration of your environment. Among many other things, you can use it to:

- Create new Datastores and connect to existing ones
- Deploy components
- Configure all settings relevant to your RES ONE Automation environment
- Schedule Jobs

**Management Portal**

The Management Portal is the web-based application for scheduling Jobs in your RES ONE Automation environment. You can also manage Library items, for example Run Books, Project, Modules, Resources and variables, and create and import Building Blocks.

The Management Portal works alongside the Console.

- The availability of a new Public API using the Swagger UI framework.

**Datastore**

The Datastore stores all relevant settings and Job history in your environment. It can also store Resources that are needed to execute certain Jobs.

**Dispatcher**

The Dispatcher is a service that directs the information flow between the Datastore and the Agents.

- A Dispatcher can cache Resources, thereby lightening the load on the Datastore.
- Dispatchers can be chained together to work effectively in different network topologies.
- The availability of a Web API enables you to build and deploy RESTful and SOAP web services and use the Dispatcher to integrate Job scheduling in external systems.

**Agent**

An Agent is a service that executes the Tasks on devices running Microsoft Windows, Unix, Linux or Mac OS X.

- You can deploy Agents+ running on Microsoft Windows from the Console.
- You need to install Agents running on Unix, Linux or Mac OS X separately.
2.2 Architecture

The RES ONE Automation architecture model can be represented as follows:

![Architecture Diagram]

**Process**

When you deploy, repair or remove a Console, Dispatcher or Agent, the Console that you are logged on to extracts the necessary data from the Datastore and then connects directly to the target computer to deploy, repair or remove the requested component.

When you create Modules, Projects or Run Books, the Console stores all data in the Datastore, including the Tasks and Resources involved. When you schedule Jobs with these Modules, Projects or Run Books, the Console (or Management Portal) stores this schedule in the Datastore.

A Dispatcher manages all communication between the Datastore and an Agent. If a new Job is available for the Agent, the Dispatcher will download all necessary data from the Datastore and store it.

- If a master caching Dispatcher is configured, the Dispatcher will download the Resources from the master caching Dispatcher, not from the Datastore directly.

In turn, the Agent checks (polls) the Dispatcher at regular intervals for relevant changes to Job schedules, Agent properties, global settings, etc. In case of the Agent+, the Dispatcher notifies the Agent+ if there are any changes. Once the push connection has been established, the Dispatcher sends a push notification to the Agent+, so polling is no longer necessary. When a new Job is available, the Agent or Agent+ will download all necessary data from the Dispatcher and perform the Job. If a Dispatcher is too busy, the Agent or Agent+ will try to contact a different Dispatcher.
Ports

The RES ONE Automation components use the following default ports to communicate:

<table>
<thead>
<tr>
<th>TCP Port</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>443</td>
<td>Used by the web browser of the administrator or user to connect to the web server that hosts the Management Portal if SSL is used.</td>
</tr>
</tbody>
</table>
| <vendor default> | The following components connect to the Datastore:  
  - Console  
  - Management Portal via IIS  
  - Dispatcher  
  Microsoft SQL Server, Microsoft Azure SQL, Oracle, MySQL and IBM DB2 use different default ports. See the vendor documentation for the correct port numbers. |
| 3163 | Used for poll communication between Dispatchers and Agent(s) or Agent(s)+. This port is fixed and cannot be changed.  
  - By default, port 3163 is used also used for communication between Dispatchers. You can change the port between Dispatcher and master caching Dispatcher, to make a master caching Dispatcher work exclusively for caching. See also Master caching Dispatcher. |
| 3162* | Used for push communication between Dispatchers and Agent(s)+. *This port number can be changed. |

To secure communication between:

- A Datastore based on Microsoft SQL Server and the components that connect to it (Consoles and Dispatchers), you can use **Database protocol encryption**.
- Dispatchers and Agents, you can configure the global setting **Protocol Encryption Dispatcher**, irrespective of the database type used.
2.3 Licensing

When RES ONE Automation is installed, you need licenses to use the functionality. Only licensed Agents can execute Jobs.

- When RES ONE Automation is installed, 75 evaluation licenses are provided automatically. These are valid for 45 days. Please contact backoffice@res.com if you want to extend this evaluation period.
- To use RES ONE Automation after the trial period, you need RES ONE Automation licenses, which can be obtained from your local reseller.
- Licenses are distributed based on the deployment date of Agents. When Agents are removed, the spare licenses are redistributed automatically.

At Setup > Licensing (see page 23), you can view and manage your RES ONE Automation licenses.

Please refer to the RES ONE Automation Upgrade Guide for licensing attention points when upgrading and downgrading your environment.
2.4 Video Tutorials

To help you quickly on your way in our product, we have created a number of easy to use online video tutorials. Just click the links below to watch the videos.

The following video tutorials are available for RES ONE Automation:

**Installation and Setup**
- BareMetal Deployment using WDS
- Creating the Datastore
- Deploying the Components
- Installation
- Licensing Model & Connectors

**Jobs**
- Schedule Jobs & Examining Job Results
- Reapply Job History
- Snapshot Intelligence

**Library**
- An Introduction to Parameters
- AutoCreate Parameters
- Citrix XenApp deployment
- Conditions
- Creating Modules - 1: FoxIt Installation
- Creating Modules - 2: iTunes Installation
- Creating Modules - 3: OpenOffice Installation
- Creating Projects
- Creating Run Books
- Executing Unix/Linux Tasks
- Evaluators
- Input Settings
Chapter 2: About RES ONE Automation

- Launch Windows
- Linked Parameters
- Manually Create Parameters
- Microsoft Service Center Configuration Manager
- Parse Variables, Parameters and Functions
- Using Global Variables

Topology
- Creating Teams
- RES ONE Automation Architecture
- Planning a RES ONE Automation environment

Administration
- Managing the Datastore
Chapter 3: Setup

The Setup menu allows you to configure:

- **Database** (on page 15) connection settings
- **Licensing** (see page 23)
- **RES ONE Identity Director integration** (on page 30)
- **Task Visibility** (on page 34)
- **Add-ons** (on page 34)
- **Global Settings** (see page 35)

3.1 Installation procedure

**Prerequisites**

Before you install RES ONE Automation components, make sure to meet the following prerequisites:

- **Hardware** (on page 12)
- **Software** (on page 12)
- **Database** (see page 15)

**Procedure in short**

RES ONE Automation is designed to install the various components in a certain order:

1. Install RES ONE Automation using the RES ONE Automation Installer. This installs the first Console. From there, all the other components can be installed.
2. Create the Datastore.
3. Install the Management Portal and connect to Datastore.
4. Set up and configure your environment:
   - Set up security access for the Console and Management Portal (at **Administration > Security**).
   - Configure the global settings (at **Setup > Global Settings**) for tuning your RES ONE Automation site. It is important to configure the global settings **before** you install Dispatcher and Agents, because most of these settings are related to Dispatchers and Agents. If necessary, you can make exceptions to the global settings for individual Dispatchers and Agents after their installation.
   - Import and register the licenses (at **Setup > Licensing**).
5. Deploy at least one Dispatcher.
6. Deploy Agents.
7. If necessary, deploy additional Consoles.
8. If necessary, deploy additional RES ONE Connectors to integrate with third party software.

3.1.1 Prerequisites

Hardware

- All hardware must comply with the Microsoft Hardware Compatibility List (HCL).
- The installation of a Console takes 32MB of disk space.
- The installation of a Dispatcher takes 84MB of disk space.
- The installation of a legacy A takes 10MB of disk space.
- The installation of an Agent+ takes 18MB of disk space.
- The initial installation of the RES ONE Automation environment takes 211MB of disk space.
- Verify that the database server that hosts the RES ONE Automation Datastore and the devices on which the Dispatchers are installed have sufficient disk space to store Resources. Resources can also be stored on file servers.

Software

See the Compatibility Matrix (on page 316) for compatible operating systems.

Additionally, the following software is required:

- Microsoft Windows Installer 3.1 and higher.
- Microsoft .NET Framework 4.0 (Full version) or higher on the machine running the RES ONE Automation Installer.
- Microsoft .NET Framework 4.5.2 (Full version) or higher on all machines hosting a Dispatcher and all machines running an Agent+.
- When upgrading from a previous version of RES ONE Automation: RES Automation Manager 2011 SR3 or higher. Please refer to the RES ONE Automation Upgrade Guide.

Specific prerequisites for Management Portal

<table>
<thead>
<tr>
<th>Software installed on system</th>
<th>Prerequisites</th>
</tr>
</thead>
</table>
| RES ONE Automation           | • RES ONE Automation 10.2  
|                              | • The installation package and update Module for the Management Portal. These files can be downloaded at [http://success.res.com](http://success.res.com), at the Downloads section. |
| .NET                         | Microsoft .NET Framework 4.5.2 or higher |
|                              | **Note:** An active Internet connection is needed, in case the .NET Framework is not yet installed. The RES ONE Automation Management Portal Installer will download and install the framework via this connection. |
| Operating System             | We recommend installing the RES ONE Automation Management Portal on a web server. For evaluation purposes, you can also install it on a workstation. We do not recommend installing it on a domain controller. |
| Database Client              | Database client used for the RES ONE Automation Datastore. AES-256 encrypted databases are also supported by the Management Portal. |
|                              | **Supported Database systems** (on page 319) |

Supported Microsoft Windows operating systems (on page 316)
## Chapter 3: Setup

### Software installed on system

<table>
<thead>
<tr>
<th>Browser</th>
<th>Prerequisites</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supported browsers:</td>
<td></td>
</tr>
<tr>
<td>• Microsoft Internet Explorer 10 or higher</td>
<td></td>
</tr>
<tr>
<td>• Microsoft Edge 15 or higher</td>
<td></td>
</tr>
<tr>
<td>• Mozilla Firefox 47 or higher</td>
<td></td>
</tr>
<tr>
<td>• Google Chrome 52 or higher</td>
<td></td>
</tr>
<tr>
<td>• Opera 12 or higher</td>
<td></td>
</tr>
<tr>
<td>• Safari 9.0 (Mac only) or higher</td>
<td></td>
</tr>
</tbody>
</table>

JavaScript enabled

### Web server

<table>
<thead>
<tr>
<th>Permissions</th>
<th>Prerequisites</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internet Information Services (IIS)</td>
<td>IIS 7.5 or higher</td>
</tr>
</tbody>
</table>

### Microsoft Windows features installed on Web server

<table>
<thead>
<tr>
<th>Necessary features for Microsoft Windows Server 2008 R2</th>
<th>Necessary features for Microsoft Windows Server 2012 / 2012 R2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Microsoft .NET Framework</strong></td>
<td>• Version 4.5.2 or higher</td>
</tr>
<tr>
<td>• To support Microsoft Internet Explorer 10, install all available Microsoft Windows updates for .NET Framework 4.5.2 first.</td>
<td>• ASP.NET 4.5</td>
</tr>
</tbody>
</table>

### Web Server (IIS)

<table>
<thead>
<tr>
<th>Web Server</th>
<th>Application Development</th>
<th>Health and Diagnostics</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Common HTTP Features</td>
<td>• ASP.NET</td>
<td>• HTTP Logging</td>
</tr>
<tr>
<td>• Static Content</td>
<td>• .NET Extensibility</td>
<td>• Request Monitor</td>
</tr>
<tr>
<td>• Default Document</td>
<td>• ISAPI Extensions</td>
<td></td>
</tr>
<tr>
<td>• HTTP Errors</td>
<td>• ISAPI Filters</td>
<td></td>
</tr>
<tr>
<td>• HTTP Redirection</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Copyright © RES Software Development B.V. All rights reserved. 13
Chapter 3: Setup

<table>
<thead>
<tr>
<th>Microsoft Windows features installed on Web server</th>
<th>Necessary features for Microsoft Windows Server 2008 R2</th>
<th>Necessary features for Microsoft Windows Server 2012 / 2012 R2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security</td>
<td>• Basic Authentication</td>
<td>• Basic Authentication</td>
</tr>
<tr>
<td></td>
<td>• Windows Authentication</td>
<td>• Windows Authentication</td>
</tr>
<tr>
<td></td>
<td>• Client Certificate Mapping Authentication</td>
<td>• Client Certificate Mapping Authentication</td>
</tr>
<tr>
<td></td>
<td>• IIS Client Certificate Mapping Authentication</td>
<td>• IIS Client Certificate Mapping Authentication</td>
</tr>
<tr>
<td></td>
<td>• Request Filtering</td>
<td>• Request Filtering</td>
</tr>
<tr>
<td>Performance</td>
<td>• Static Content Compression</td>
<td>• Static Content Compression</td>
</tr>
<tr>
<td></td>
<td>• Dynamic Content Compression</td>
<td>• Dynamic Content Compression</td>
</tr>
<tr>
<td>Management Tools</td>
<td>• IIS Management Console</td>
<td>• IIS Management Console</td>
</tr>
<tr>
<td></td>
<td>• Management Service</td>
<td></td>
</tr>
</tbody>
</table>

Other requirements

When deploying components from the Console on machines running Microsoft Windows Vista or higher, disable the User Account Control feature, as this feature does not support silent installations.

Other requirements for Agents:

- Allow RES ONE Automation traffic in the Microsoft Windows Firewall. Please refer to the document Antivirus Recommendations RES ONE Enterprise for recommended exclusions for real-time antivirus scanning.
- Remote Registry service enabled
- Simple File Sharing (or Use Sharing Wizard) disabled
- IPC$ enabled

Notes

• Not all RES ONE Automation Tasks may be functional on all operating systems.
• See the Compatibility Matrix for a complete overview of all products supported by RES ONE Automation.
3.2 Database

The node Setup > Database shows the connection settings to the current Datastore. At this node, you can connect to another Datastore or create a new one. After a fresh installation of RES ONE Automation, the Console must be pointed to an existing Datastore, or a new Datastore must be created.

When creating a new Datastore, your RES ONE Automation environment uses AES-256 encryption (see page 21). For existing environments that do not use AES-256 encryption, a conversion tool is available at http://success.res.com.

Configuration
- To connect to another existing Datastore, edit the connection settings and click Connect.
- To create a new Datastore, click Create to open the Datastore Wizard.
- After creating a new Datastore or connecting to an existing one, the Console will restart and any administrative roles will be applied.
- Alternatively, it is also possible to connect to a database that has not been created by RES ONE Automation, but by a database administrator on the database console. This is useful if a RES ONE Automation administrator does not have the correct permissions to create NEW databases, but is allowed to manage EXISTING ones. For example, an administrator with only DBO permissions is not allowed to create new databases, but can connect RES ONE Automation to a database created by someone else who does have sufficient permissions (DBA). This database can then be used as a RES ONE Automation Datastore. When connecting to a database that has not been created by RES ONE Automation, RES ONE Automation will ask for confirmation first. When confirmed, RES ONE Automation will create the required tables.

Supporting database types
RES ONE Automation supports the following database types:
- Microsoft SQL Server (on page 16)
- Microsoft Azure SQL (on page 19)
- MySQL (on page 21)
- Oracle (on page 20)
- IBM DB2 (on page 20)

Notes
- When creating a new Datastore and a connection to an existing Datastore is not available, the full installation of RES ONE Automation and Microsoft .NET Framework 4.5.2 (Full version) or higher are required on the computer from which a new Datastore is created. In that case, it is not possible to create a new Datastore from a computer on which only the Console component has been installed.
- When creating a new Datastore and a connection to an existing Datastore is available, RES ONE Automation will use the components in the existing Datastore to create the new one.
- When connecting to a non-RES ONE Automation database, do not connect to an existing non-RES ONE Automation database that already contains data. Use empty databases only.
3.2.1 Microsoft SQL Server

Supported Microsoft SQL Server databases

- Microsoft SQL Server 2016
- Microsoft SQL Server 2014
- Microsoft SQL Server 2012
- Microsoft SQL Server 2008 R2
- Microsoft SQL Server 2008
- Microsoft SQL Server 2005

Prerequisites

- Microsoft SQL Native Client if you want to use protocol encrypted (see page 19) communication between the Datastore and the Consoles and Dispatchers or if WMC.exe needs to be started from the command line.

Other

- When connecting to a Datastore based on Microsoft SQL Server, RES ONE Automation uses SQL Server Authentication (see page 17) by default. You can also use Windows Authentication (see page 17).
- You can configure Microsoft SQL Server and IBM DB2 to use a different port (see page 19) than the default one.
SQL Server authentication

- **Mixed Mode** authentication
- A named SQL Server System Administrator login ID, or an alternative account with server role "sysadmin" and user mappings to master, model, msdb and tempdb
- Microsoft .NET Framework 2.0 or higher

Windows authentication

Microsoft recommends using **Windows authentication** when you connect to a Microsoft SQL Server. This is more secure than **SQL Server authentication**.

- Depending on the configuration of your database server, you can use Windows authentication on server-level or on database-level. If you switch between authentication modes on server-level, other databases on this server will also be affected.
- You can only use Windows authentication if all RES ONE Automation components are member of a domain in the same AD forest or of a trusted domain (typically single-tenant sites). Windows authentication is not supported in environments with disjointed AD connectivity (typically multi-tenant sites).
- Windows Authentication is not supported on Domain Controllers and on Microsoft Windows Small Business Server. This only affects components that connect to the Datastore (Consoles and Dispatchers), not Agents.
- For troubleshooting purposes, use the Windows Event log to validate a successful connection.

New installations

1. In Microsoft Active Directory, create a Group for service accounts.
2. Create an Active Directory User that is a member of this service accounts group.
3. Create the following policy:
   1. Log on as a service for the service accounts group.
   2. Add the service accounts group to the local administrators group.
4. Link the policy to the OU that contains the devices running the Consoles and/or Dispatchers.
5. Open Microsoft SQL Server Management Studio.
   1. In the Security folder, create a new login.
   2. Click Search and then Object Types.
   3. Add the service accounts group that you created earlier.
   4. Add Domain Admins (or any other group of administrators that uses the RES ONE Automation Console).
6. Create a new default database with the following settings:
   - Size 150MB, AutoGrow 25MB
   - Log 75MB, AutoGrow 10MB
7. Open the properties of the service accounts group.
8. On the User Mapping tab, select the database that you just created.
9. In Database Role Membership, select the db_owner role.
10. All users who need access to the RES ONE Automation Console need at least the following rights on the Datastore:
    - DB_Datareader
    - DB_Datawriter: To adjust these rights:
      1. Create an Active Directory group.
      2. Add all users who need access to the Console to this Active Directory group.
Chapter 3: Setup

3. Add this group in the Security node on the SQL server.
4. Under User Mapping select the RES ONE Automation Datastore and select the roles `db_datareader` and `db_datawriter`.

11. Add the account that is going to create the database tables and add the role DBO.
   - Alternatively, when you use accounts from another domain:
     1. Add Domain Admins (or any other group of administrators that uses the RES ONE Automation Console) and the service account group to a domain local group.
     2. In Microsoft SQL Server Management studio, add the domain local group to the database as `db_owner`.

12. Install RES ONE Automation with a user that has the role DBO.
13. After installation, start the RES ONE Automation Console.
14. When prompted, do NOT create a new database, but connect to the one that you just created.
15. Provide the required information and select Windows Authentication.
16. Specify the Service Account in the format: DOMAIN\username.
17. Click OK.
18. When you connect to the database, confirm to create the required tables.

**Existing installations using SQL Server authentication**

1. On the Microsoft SQL Server, switch the authentication mode for the RES ONE Automation Datastore from mixed mode authentication to Windows Authentication.
2. Follow the steps as described above, but skip the step where you create a new database.
3. Start the RES ONE Automation Console and at Setup > Database change Database authentication to Windows Authentication.
4. Provide the service account credentials and click Connect. The Console will now restart.
5. Repair all Consoles and Dispatchers in the RES ONE Automation environment. Every new Console and Dispatcher that you deploy will use the provided service account credentials.
   - If you change the service account at a later stage, you need to repair all Consoles and Dispatchers.
Specify alternative ports for Microsoft SQL Server and IBM DB2

You can configure Microsoft SQL Server and IBM DB2 database servers to listen on a different port than the default one (1433 for Microsoft SQL Server, 50000 for IBM DB2). You need to specify this port number in RES ONE Automation from the moment it is used on the database server.

- For Microsoft SQL Server, separate server name and port number with a comma (,), as in: `<dbserver>, <port number>`.
- For IBM DB2, separate server name and port number with a colon (:), as in: `<dbserver>: <port number>`.

New environments

In the Datastore Wizard, when you select a database server (for Microsoft SQL Servers only), specify the port number in the **Server** field in the format as shown above.

Existing environments

1. At Setup > Database, specify the port number in the **Database server** field in the format as shown above.
2. Click **Connect**.
3. Perform a repair of all Dispatchers.

For IBM DB2 database server, you can only connect to an existing database.

Configure Protocol Encryption

When you use a Datastore based on Microsoft SQL Server, you can optionally use protocol encryption to secure communication between the Datastore and the components that connect to it.

Prerequisites

- Microsoft SQL Native Client on the devices running the Consoles and Dispatchers.

Configuration

- When you create a new Datastore, enable **Force Protocol Encryption** in the Datastore Wizard.
- When you connect to an existing Datastore or change the protocol encryption settings of the current Datastore, enable **Force Protocol Encryption** at Setup > Database.
  - You need to enable protocol encryption for any individual Console that connects to the Datastore.
  - You need to repair any available Dispatchers before the setting takes effect. This forces the Dispatchers to reload the connection parameters to the Datastore.
  - When enabled, at Topology > Console and Dispatcher, the property **Database Encryption** will be shown as Yes.

The SSL parameters that the Microsoft SQL Server database uses to encrypt communication to Dispatchers and Consoles depend on the Security Certificate used by this database.
3.2.2 Microsoft Azure SQL

Supported databases
- Microsoft Azure SQL

Other Prerequisites

Database
- An existing Microsoft Azure SQL database
- Microsoft Azure database account

Consoles and Dispatchers
Microsoft SQL Server Native Client (ODBC driver)

3.2.3 Oracle

Supported Oracle databases
- Oracle 12c
- Oracle 11.2
- Oracle 11.1

Prerequisites

Database
- Oracle DBA credentials. Use uppercase characters only.

Consoles and Dispatchers
- Oracle OLEDB provider
  - Oracle win64_11gR2_client with Patch 10245351 or higher (x64)
  - Oracle win32_11gR2_client with Patch 10245350 or higher (x86)
3.2.4 IBM DB2

Supported IBM DB2 databases
- IBM DB2 10.5
- IBM DB2 9.8
- IBM DB2 9.7
- IBM DB2 9.5

Prerequisites
Database
- An existing database and database user.
- The database user must be able to access a table space with a page size of at least 8KB

Consoles and Dispatchers
- IBM DB2 OLEDB provider (x64/x86)

Configuration
1. Create a local user on the DB2 Server, for example: RESONEAutomation. This account will be used to connect to the RES ONE Automation Datastore.
2. Manually create a database on the DB2 Server, for example: ROADB.
3. Create a table space for the user RESONEAutomation in the ROADB database. Specify a buffer pool of at least 8KB.
4. Create a schema in the ROADB database, for example: roa.
5. Add the user you created in step 1 to the database ROADB.
6. Assign the applicable authorities, and add the schema and table space you created in steps 3 and 4 to this user. Make sure you assign the proper privileges.

You can configure IBM DB2 database servers to listen on a different port than the default port 50000.

See also
- Specify alternative ports for Microsoft SQL Server and IBM DB2 (on page 19)

3.2.5 MySQL

Supported MySQL databases
- MySQL 5.6
- MySQL 5.0

Prerequisites
Database
- MySQL DBA credentials

Consoles and Dispatchers
- MYSQL ODBC Driver
  - 5.3.8
  - 5.2.4 (x64/x86) - ANSI only
  - 5.1.12 (x64/x86)
Chapter 3: Setup

3.2.6 Encryption using AES-256

When creating a new Datastore, RES ONE Automation automatically uses AES-256 encryption to protect the security sensitive data (for example, credentials/passwords and security context). At Administration > Information (and Setup > Database), the type of encryption is displayed: 'This database is configured to use AES-256 encryption'.

If your existing environment uses legacy encryption and you want to use AES-256 encryption, you can download the RES ONE Automation AES-256 Conversion Tool from the Success Center (at http://success.res.com). Before using this Conversion Tool, all Dispatchers, Consoles and Agents in your environment must be running RES ONE Automation 10.2 or higher.

If your environment uses a Datastore with AES-256 encryption, downgrading to RES ONE Automation version 10.0 or lower is not supported. Also, (re)connecting a legacy Console (version 10.0 or lower) to an AES-256 encrypted environment is not supported.

The communication between Dispatchers and Agents is automatically secured by protocol encryption (see page 39) (SSL).

In migration scenarios, it is possible to import Building Blocks to fill your Datastore. See also Building Blocks using AES-256 encryption (on page 225).

Agents for Unix/Linux

AES-256 encryption is not supported for Agents running on the following platforms:

- Red Hat Enterprise Linux 4.5 (x86/x64)
- Sun Solaris 10 for SPARC
- Sun Solaris 10 for Intel

3.2.7 Install database drivers using RES ONE Automation

Consoles and Dispatchers require the installation of certain database drivers on the computers on which they are deployed. You can do this using RES ONE Automation:

1. Manually install the appropriate database driver on a computer.
2. Deploy a Dispatcher on this computer.
3. Deploy an Agent on all computers on which you want to deploy a Console or a Dispatcher.
4. Install the database driver on the target computers using the Task Invoke Windows Installer or Perform Unattended Installation.
5. After this, you can deploy Consoles and Dispatchers on the target computers.

You can also use this method to migrate an existing RES ONE Automation environment from a Microsoft SQL database to a MySQL, Oracle or IBM DB2 database. Before you do this, create a Building Block of your Library as a backup.
3.3 Licensing

The node **Setup > Licensing** shows how many RES ONE Automation licenses are available and how many are claimed.

License requirements

The following license model applies:

<table>
<thead>
<tr>
<th>Agent type</th>
<th>Number of required licenses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows client</td>
<td>0.5 (For example, 4 Windows clients require 2 licenses, 5 Windows clients require 3 licenses).</td>
</tr>
<tr>
<td>Mac OS X</td>
<td>0.5</td>
</tr>
<tr>
<td>Windows server (including terminal servers)</td>
<td>6</td>
</tr>
<tr>
<td>Unix/Linux</td>
<td>6</td>
</tr>
</tbody>
</table>

When the RES ONE Identity Director integration is enabled and validated, you will receive 12 additional license points for RES ONE Identity Director (RES ONE Identity Director licenses are shown under the section **Connectors**).

In RES ONE Automation environments that upgraded from a previous version and that (also) used pre-2014 licenses, a **different pre-2014 licensing model** (see page 25) applies.

**Licensing tab**

- To import and activate new licenses, click **License Wizard** and follow the steps.
- To view the details of a license, select it in the list and click **View**. This will open the RES ONE Automation License window, which will display the details of the selected license.
- To delete a license (for example, because it is an evaluation license), select it in the list and click **Delete**. You cannot delete Site IDs.

See **Licensing Process** (see page 26) for information on how to register and activate your licenses.
License Server tab

For administration purposes, you need to report the actual license usage to RES backoffice every month for all RES products using Datacenter Server and Desktop - ASP and Corporate licenses. To automate this reporting process, you can use the RES ONE License Server.

Important: This feature will only become available when the RES ONE License Server is released.

- To implement the RES ONE License Server, enable and click Registration Wizard and follow the steps in the registration process.
- The Status overview shows the latest date/time and status of the following actions:
  - when your site was registered to the RES ONE License Server (register)
  - which Dispatcher has most recently collected the license usage (collect)
  - which Dispatcher has most recently reported the license peak usage to the RES ONE License Server (report)
  - if communication to the RES ONE License Server is still possible (heartbeat)
- To view or change the Dispatcher role of the available Dispatchers, click Dispatcher Settings Overview.
- To view the collection and reporting of the license usage, click Export Data and Export Reports.

See RES ONE License Server registration process (on page 27) for more information on how to implement the RES ONE License Server.

See License usage collection and reporting (on page 29) for more information on the collected data.
3.3.1 Pre-2014 Licensing Model

In the Pre-2014 Licensing Model, the license count for clients and servers is different and connectors are required to execute certain advanced Tasks. Typically, this model applies to environments with licenses generated before February 2014.

Consider the following scenarios:

- If your environment uses only legacy licenses, the Pre-2014 Licensing Model applies.
- If your environment uses a mix of legacy licenses and Datacenter Server and/or Desktop licenses, the Pre-2014 Licensing Model applies. In this scenario, connectors do not require licenses and are freed up.
- If your environment uses only Datacenter Server and/or Desktop licenses, the new Licensing Model applies. See Licensing Model.

License requirements for Agents (pre-February 2014)

<table>
<thead>
<tr>
<th>Agent type</th>
<th>Number of required licenses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows client</td>
<td>0.5 (i.e. 2 Windows clients require 1 license, 3 Windows clients require 2 licenses).</td>
</tr>
<tr>
<td>Mac OS X</td>
<td>0.5</td>
</tr>
<tr>
<td>Windows server</td>
<td>4</td>
</tr>
<tr>
<td>Windows terminal server</td>
<td>8</td>
</tr>
<tr>
<td>Unix/Linux</td>
<td>4</td>
</tr>
<tr>
<td>RES HyperDrive Virtual Appliance</td>
<td>0</td>
</tr>
</tbody>
</table>

License requirements for connectors (pre-February 2014)

In the pre-February 2014 Licensing Model, certain Tasks require connectors. These claim additional licenses:

<table>
<thead>
<tr>
<th>Task type</th>
<th>Required Connector type</th>
<th>Required Connector licenses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Statement</td>
<td>Database Server connector</td>
<td>1</td>
</tr>
<tr>
<td>Secure Shell Commands</td>
<td>Remote Host connector</td>
<td>1</td>
</tr>
<tr>
<td>VMware ESX Virtual Machine</td>
<td>Virtualization Host connector</td>
<td>4</td>
</tr>
<tr>
<td>Exchange Mailbox</td>
<td>Mail Server connector</td>
<td>16</td>
</tr>
<tr>
<td>Active Directory</td>
<td>Directory Server connector</td>
<td>32</td>
</tr>
<tr>
<td>Database Servers, Mail Servers, Directory Servers</td>
<td>Small Business Server connector</td>
<td>33</td>
</tr>
<tr>
<td>Web Service</td>
<td>Web Service Hosts connector</td>
<td>1</td>
</tr>
</tbody>
</table>

Notes

- If you have an Enterprise Agreement, your license requirements may deviate from the requirements mentioned above. Please contact your reseller for more information.
- When you add a Small Business connector to execute Tasks on a computer, licenses that are claimed by other connectors that execute Tasks on the same computer are not freed up automatically. The Small Business connector needs to be assigned to these Tasks manually, after which you can delete the redundant connectors.
3.3.2 Licensing process

The licensing process consists of two steps:

1: Import the licenses
1. When you order your RES ONE Automation licenses, they will be sent as an "RES ONE Automation License file" by e-mail. Save this file to an accessible location.
2. In the Console, at Setup > Licensing, open the License Wizard and select Add RES ONE Automation license.
3. Follow the prompts to import the licenses to the Datastore. All new licenses should be imported at once.
4. At the end of the process, you will be prompted to register your licenses.
   - Evaluation licenses can be registered immediately, but can also be used unregistered until the end of the evaluation period.
   - Corporate licenses need to be registered immediately.
   - Production licenses can be registered within 30 days.

When you have completed the import process, your licenses and all relevant information will be shown in the Licensing node.

2: Register and activate the licenses

When you have imported all your RES ONE Automation licenses, you need to register and activate them. You do not need to do this per license: when you start the registration process with the License Wizard, RES ONE Automation will scan for any license that has not been activated and activate all of them at once. When registering your licenses, a Site ID will be linked to your licenses. The Site ID contains all relevant information about your organization and your RES ONE Automation environment.

1. In the Console, open the License Wizard and select Register RES ONE Automation license(s) and request activation.
2. Follow the prompts to register and activate your RES ONE Automation Licenses.
   - The name of your RES ONE Automation Site must be unique in your environment, because it links your licenses to your business.
3. The last step of the License Wizard is to activate your licenses. Licenses can only be registered and activated through the License Wizard.

When you have activated your licenses, the Licensing node will display an overview of your licenses, including license type, RES ONE Automation edition, site status and the number of licenses that are required and claimed.

Note

If you import any additional RES ONE Automation licenses at a later stage, they can be registered and activated using the procedure as described above. Additional licenses must be registered under the same Site name.
Manual activation

In certain situations, automatic activation of licenses may fail. For example, this may occur when the RES license activation server cannot be reached and certain license types are used. In these situations, RES ONE Automation licenses still need to be activated manually.

1. In the Console, open the License Wizard and select Register RES ONE Automation license(s) and request activation.
2. Follow the prompts to register and activate your RES ONE Automation Licenses. When you register your licenses, only one text file will be generated for all licenses. This file needs to be sent to RES, as a request for license activation. There are three ways to do this:
   - **Web**: e-mail the activation request to RES from the RES website.
   - **E-mail**: e-mail the activation request to RES directly from an e-mail client on the computer running the Console (requires a configured MAPI-compliant e-mail client).
   - **Save to file**: save the activation request as a text file that you send to activation@res.com.
3. After the activation request has been processed, RES will send the activation file to the mail recipient that you specified during registration. The activation file will be sent within 24 hours during workdays, and enables you to activate all licenses.
4. Save the activation file that you received by e-mail to an accessible location.
5. In the Console, open the License Wizard again and select Activate RES ONE Automation license(s).
6. Follow the prompts to activate your RES ONE Automation licenses.

⚠️ **Warning**

License files and activation files contain crucial information. Do not edit these files, because it will render the files useless.
3.3.3 License server registration process

Note
This feature becomes available when the RES ONE License Server is released.

The license server registration process consists of two steps:

1: Register to RES ONE License Server

To automate the reporting process via the RES ONE License Server, you need to enable the RES ONE License Server and register your RES ONE Automation site. The Console needs Internet access for a successful registration and Microsoft .NET Framework 4.0 (Full version) or higher must be installed.

1. In the Console, at Setup > Licensing, on the License Server tab, enable the License Server (for Datacenter Server and Desktop - ASP and Corporate licenses only).

2. Open the Registration Wizard.

3. Follow the prompts to configure the communication to the RES ONE License Server. The default URL and port number of the RES ONE License Server are provided. You can change the URL and port number if needed.

4. At the end of the process you will be prompted to authenticate with a valid RES Success Center Salesforce account. If you log in with a valid account, a verification code is shown. Please enter the verification code and click Register.

Your site is registered successfully to the RES ONE License Server.

2: Assign Licensing role to Dispatcher

To have control over which Dispatcher will report the license peak usage to the RES ONE License Server, you need to assign the Licensing role to one or more Dispatchers.

1. In the Console, at Topology > Dispatchers > Settings tab, enable the Licensing role.

Your Dispatcher is configured to report to the RES ONE License Server (see page 189). If no Dispatchers have the Licensing role or if all Dispatchers with the Licensing role are offline, a warning message will be shown when the Console user logs in. It is possible to configure a proxy server to route the Internet traffic from the Dispatchers to the RES ONE License Server via the global setting Dispatcher Proxy (see page 40).
3.3.4 License usage collection and reporting

Note
This feature becomes available when the RES ONE License Server is released.

At Setup > Licensing, on the License Server tab:

- Use the Export data button to view the collected license usage
- Use the Export reports button to view the reported license peak usage

Select the time span and filename and location to export the information to a CSV file.

Collection and reporting

Every hour, the Dispatchers will automatically collect the license usage of ASP and Corporate licenses and save it in the Datastore. By default, all Dispatchers are allowed to collect this information, but only one will actually do the collection. If you have assigned the Licensing role to one or more Dispatchers, only the assigned Dispatchers are allowed to do the collection.

After you have successfully registered the product and assigned the Licensing role to one or more Dispatchers, the assigned Dispatchers will also automatically report the license peak usage to the RES ONE License Server every first day of the month. Only the assigned Dispatchers are allowed to communicate with the RES ONE License Server and report the license peak usage. Furthermore, every hour the communication to the RES ONE License Server is tested to see if the server is available (heartbeat).

It is possible to configure a proxy server to route the Internet traffic from the Dispatchers to the RES ONE License Server via the global setting Dispatcher Proxy (see page 40).

Export data

The CSV file (export-license-data.csv) shows when (UTC date and time) and by who (Dispatcher name) the information was collected for the specified time span and includes the following measurements:

- Total number of used license points
- Number of servers (such as: Windows Servers, Unix/Linux systems)
- Number of workstations (such as: Windows, Mac OS X clients)

The license points are rounded up. For example, 191 workstations will result in 96 license points. By default, the time span from the start of the current month to the current date is shown. The list is separated by semi-colons (";").
Export reports

The CSV file (export-license-usage.csv) shows the following information per report for the specified time span:

- Applicable product (environment ID - as registered in the RES ONE License Server)
- Date and time of calculating (fact date) and sending (date sent) the report
- License types (license info) used in the product
- Summary of the collected information
- Period, year and month, of the report
- Status of sending the report to the RES ONE License Server: OK or Failed
- Collected information of peak moment, which includes:
  - Total number of used license points at peak moment
  - Date and time of peak moment
  - Number of servers at peak moment
  - Number of workstations at peak moment

By default, the time span from 6 months before the current date to the current date is shown. The list is separated by semi-colons (",").
3.4 RES ONE Identity Director

At Setup > RES ONE Identity Director, you can integrate RES ONE Identity Director (or RES ONE Service Store) into your RES ONE Automation environment.

When you have configured RES ONE Identity Director integration AND your administrative role has Modify access to the Run Books node, a RES ONE Identity Director Wizard becomes available at Library > Run Books, which allows you to create new services in RES ONE Identity Director based on Run Books and service templates. See Creating RES ONE Identity Director services based on Run Books (on page 167).

As soon as you have enabled and validated the RES ONE Identity Director integration, you will receive 12 complimentary license points for RES ONE Identity Director. The RES ONE Identity Director licenses are made available under the section Connectors in the Licensing overview, regardless of any other licenses that you may already have.

Configuration

• Specify the Catalog host name and port. The catalog host is the machine on which the Catalog Services run. The Catalog Services are used by RES ONE Automation to query which RES ONE Identity Director templates are available. The default port of the Catalog host is 8081.

• Specify the Publication name of the service publication that contains the RES ONE Identity Director templates. If necessary, also specify the Publication password.

• When configuring RES ONE Identity Director Integration, use the Test now button to test whether you can actually connect to the specified catalog host. If communication is possible, a message “Connection successful” will be shown.

Notes

• The user’s access permissions for the RES ONE Identity Director node does not influence the availability of the RES ONE Identity Director Wizard.

• The RES ONE Identity Director licenses are not made available during the trial period and when the pre-2014 licensing model applies.
3.5 PoSh Automation

PoSh Automation is the free starter package offering basic RES ONE Automation functionality. This edition contains the Windows PowerShell Script Task and offers the complement of other functionality such as Job scheduling, Building Blocks and Teams. With PoSh Automation, you can use RES ONE Automation for an unlimited period with an unlimited number of licenses to deploy Agents. To unlock the full functionality of RES ONE Automation, activate a production license and automatically switch from PoSh to Full mode retaining all existing Datastore content.

When installing RES ONE Automation 10.2 or higher and creating a new Datastore, select the option PoSh Automation to run in PoSh mode. To access the PoSh Automation edition, use the default user name res and password res. This user account is also used to access the RES ONE Automation Management Portal if public access is configured. You can specify a different PoSh password in the Security node. At Setup > Licensing, on the Edition tab, you can manually switch between the PoSh Edition and Full Edition in Evaluation mode (if no production licenses have been added and the RES ONE Identity Director integration has been disabled).

Keep in mind that:

- PoSh Automation uses public access. Therefore, the RES ONE Automation login accounts and administrative roles are not available in PoSh mode. Any global Password Security Policy settings that are configured will be ignored in PoSh mode.
- All Agents need to be updated to the latest version of RES ONE Automation in order to support the PoSh mode. Therefore, when switching to PoSh mode, all Agents are updated automatically.
- In PoSh mode, the setup options RES ONE Identity Director integration, Task Visibility and Add-ons are not available.
- In PoSh mode, the global settings WebAPI state, RunBookWho, Password Security Policy and Auto Update Agent are grayed out and cannot be changed.

Configuration

- In PoSh mode, you can only select and configure the PowerShell Task. All other Tasks, including the Add-on Tasks, are disabled.
- PoSh allows all Tasks to be imported via Building Blocks (and stored in the Datastore), but only enables the PowerShell Tasks.
- At Library, when configuring a Module, Project or Run Book, the unsupported Tasks are visible but grayed out (disabled for editing, only viewing and deleting).
**Scheduling**

- In PoSh mode, you cannot (re)schedule unsupported Tasks (that are imported via Building Blocks).
- At Jobs, when (re)scheduling a Job, the unsupported Tasks are removed and will not be shown on the Resulting Tasks tab. If a Job only contains unsupported Tasks, the Job itself will be removed from the Resulting Tasks tab. If no resulting Jobs are applicable, the Job cannot be (re)scheduled. If you schedule a Job based on a Run Book and no supported Tasks are applicable, the Run Book will be aborted when (re)scheduled.

**Notes**

- When switching from PoSh to Full mode, all unsupported Tasks will be available again. The previous configuration will be restored. Please note that when rescheduling a Job from the Job History, only the actually processed Tasks will be rescheduled (i.e. the unsupported Tasks will not reappear).
- When switching from PoSh to Full mode, the PoSh login account will no longer be applicable. Any previously configured security will be restored.
- When switching from Full to PoSh mode, and viewing the Job History, you will see the Tasks actually processed (i.e. Job details from the unsupported Tasks are available).
- When switching from Full to PoSh mode, and recurring Jobs are already scheduled and running, the Jobs scheduled before switching are executed according to the Full mode, all Jobs scheduled after switching will be executed with the PowerShell Tasks only.
3.6 Task Visibility

At Setup > Task Visibility, you can configure which Tasks are visible in the Select Task window for all Console users. The Select Task window is shown when adding a new Task to a Module.

When the Task Visibility option is selected a preview of all available Tasks in the Select Task window is shown on the right-hand side. To configure, right-click the option Task Visibility and select Edit (or click the Edit button in the command bar). In the Edit Task Visibility window, select the check box to show a specific Task or clear the check box to hide a specific Task. For an up-to-date preview of all available Tasks in the Select Task window, make sure the Edit Task Visibility window is closed.

Please be aware that the option Task Visibility is not related to the security of your RES ONE Automation environment. When you hide a Task, existing Modules and scheduled Jobs can still contain these hidden Tasks. Also, Modules with hidden Tasks may still be configured, deleted or duplicated from that Module. For example, when hiding the Task Shutdown Computer, you are still able to configure the shutdown options from the Task Reboot Computer. In addition, Building Blocks have no relation with the configuration of Task Visibility. When you import a Building Block containing a Module, you can run the Modules even if specific Tasks are hidden on the system where it is imported.

Please note that new Tasks (when installed via an Add-on) are visible by default. Upgraded Tasks will keep their configuration on Task Visibility.

3.7 Add-ons

At Setup > Add-ons, you can see which Add-ons are installed in your RES ONE Automation environment and you are able to delete one or more Add-ons from the Datastore. Please note that the user needs Modify permissions on the Add-ons node to install or delete Add-ons.

Overview

When selecting the option Add-ons, an overview of all installed Add-ons is shown on the right-hand side. The column Date Modified shows the latest installation date of the Add-on. The column In Use shows whether any of the Tasks related to that Add-on is currently being used in a Module or not.

Add-on Usage

To see the impact before deleting the Add-on, select one or more Add-ons, right-click and select Add-on Usage. The Add-on Usage window gives you an overview of all Modules that contain a Task from the selected Add-on. To save the results as a CSV file, use the Export Data to File button.

Note

When exporting the Add-on Usage to a CSV file, any double quotation mark (") and/or semi-colon (;) used in a Module name will be removed from the result.

Delete

To delete, select one or more Add-ons in the Add-ons node, right-click and select Delete (or click the Delete button in the command bar). When deleting, the Tasks and related help file are removed from the Datastore. Please note that the configured Modules that use these Tasks are not removed. The same applies to the installed Quick Start Guide and Building Blocks.
3.8 Global Settings

With Global Settings, you can custom-fit the RES ONE Automation environment to your network environment. These settings allow you to define the general behavior of your RES ONE Automation environment, and the default behavior of Dispatchers, Teams and Agents.

For many of these settings, you can create exceptions for specific Dispatchers, Teams and Agents at:

- Topology > Dispatchers
- Topology > Teams
- Topology > Agents

<table>
<thead>
<tr>
<th>Section</th>
<th>Setting</th>
<th>Global Setting</th>
<th>Dispatcher Setting</th>
<th>Team Setting</th>
<th>Agent Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Site</td>
<td>Site name</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Site identifier</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dispatcher Detection</td>
<td>Dispatcher discovery</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Dispatcher locations</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Dispatcher recovery</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Exclude from Dispatcher list</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Connection</td>
<td>Bandwidth throttling (Dispatcher -&gt; Agent)</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Resource download throttling (Datastore -&gt; Dispatcher)</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Protocol Encryption Dispatcher</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dispatcher push communication</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dispatcher Proxy</td>
<td>Dispatcher Proxy</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dispatcher Cache</td>
<td>Dispatcher cache timing</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dispatcher cache availability</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dispatcher cache duration</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Dispatcher WebAPI</td>
<td>WebAPI state</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Dispatcher Roles</td>
<td>Housekeeping</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Section</td>
<td>Setting</td>
<td>Global Setting</td>
<td>Dispatcher Setting</td>
<td>Team Setting</td>
<td>Agent Setting</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>----------------</td>
<td>--------------------</td>
<td>--------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Licensing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Telemetry</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Agent Cache (see page 42)</td>
<td>Agent resource folder Windows</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Agent resource folder Unix/Linux/Mac OS X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scheduling (see page 43)</td>
<td>Launch timeout</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Launch window</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Update of detailed Task result</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Default schedule status</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Job Notification (see page 44)</td>
<td>Send mail message</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Send SNMP trap</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security (see page 45)</td>
<td>Protect clear of audits</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Remote Console</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RunBookWho</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Trusts Security</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Limit Job Export</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Limit Job Execution</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Allow agents to be added to a team programmatically</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Snapshot Intelligence (see page 47)</td>
<td>When use of snapshot is detected</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>When the last change path is reapplied</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Version Control (see page 48)</td>
<td>Use automatic versioning</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Version comment</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Password Security Policy (see page 48)</td>
<td>Password age</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Password complexity</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Section Setting

<table>
<thead>
<tr>
<th>Setting</th>
<th>Global Setting</th>
<th>Dispatcher Setting</th>
<th>Team Setting</th>
<th>Agent Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password history</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minimum password length</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Account lockout threshold</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other settings</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(see page 49)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Synchronize clock</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Identification Method</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wake-on-LAN</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Auto Update Agent</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

### Notes
- The Agent setting Agent resource folder Windows is only shown on Windows Agents.
- The Agent setting Agent resource folder Unix/Linux/Mac OS X is only shown on Unix/Linux or MAC OS X Agents.

### 3.8.1 Site settings

Global Settings allow you to custom-fit the RES ONE Automation environment to your network environment. These settings allow you to define the general behavior of your RES ONE Automation environment, and the default behavior of Dispatchers and Agents.

#### Site settings

<table>
<thead>
<tr>
<th>Site settings</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Site name</td>
<td>Represents a readable name of your RES ONE Automation site. By default, it contains the database and server name, but this name can be changed.</td>
</tr>
<tr>
<td>Site identifier</td>
<td>The Global Unique Identifier (GUID) that uniquely identifies your RES ONE Automation site.</td>
</tr>
</tbody>
</table>
3.8.2 Dispatcher Detection settings

Global Settings allow you to custom-fit the RES ONE Automation environment to your network environment. These settings allow you to define the general behavior of your RES ONE Automation environment, and the default behavior of Dispatchers and Agents.

### Dispatcher Detection settings

<table>
<thead>
<tr>
<th>Dispatcher discovery</th>
<th>Specifies the Agents’ search method for Dispatchers. Agents can discover Dispatchers by sending a multicast signal or by using a Dispatcher address list.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dispatcher locations</td>
<td>Specifies which Dispatchers should be used by the Agent. Agents can use any discovered Dispatchers to retrieve a complete list of all Dispatchers and use these as well or only use the Dispatchers it discovered directly.</td>
</tr>
</tbody>
</table>
| Dispatcher recovery  | Specifies which action Agents should take if no Dispatchers can be contacted:  
  - If you select Revert to cached list of all known Dispatchers, any other Dispatchers known to the Agent may be used if no Dispatcher can be contacted. You can prevent this by selecting Retry discovery. |

Dispatcher discovery and Dispatcher locations are closely related. Use their options to create a combination that specifies how and where an Agent should search for a Dispatcher:

- **Autodetect and Retrieve Complete Dispatcher Address List After Discovery.** This combination lets the Agent search for Dispatchers in its own network segment only. It does this by sending a multicast signal, which cannot travel over routers. The discovered Dispatchers will tell the Agent which other Dispatcher locations are known in the Datastore. The Agent will randomly connect to these Dispatchers if communication over IP port 3163 is possible.

- **Autodetect and Only Use Discovered Dispatchers.** This combination lets the Agent search for Dispatchers in its own network segment only. It does this by sending a multicast signal, which cannot travel over routers. The Agent will only communicate with the discovered Dispatchers, and randomly connect to these Dispatchers. This is the recommended combination for LAN connections.

- **Use Dispatcher Address List and Retrieve Complete Dispatcher Address List After Discovery.** This combination lets the Agent search for Dispatchers using a Dispatcher address list; not via a multicast signal. Best practice is to select First try autodetect to autodetect Dispatchers. If this fails, the specified Dispatchers will be used. The discovered Dispatchers will tell the Agent which other Dispatcher locations are known in the Datastore. In this way, the Agent will retrieve all Dispatchers. The Agent will randomly connect to these Dispatchers if communication over IP port 3163 is possible.
• **Use Dispatcher Address List and Only Use Discovered Dispatchers.** This combination lets the Agent search for Dispatchers using a Dispatcher address list; not via a multicast signal. Best practice is to select **First try autodetect** to autodetect Dispatchers. If this fails, the specified Dispatchers will be used. This setting is particularly useful for laptops: When a laptop logs on at the office, the Agent will automatically detect a Dispatcher. When a laptop is used at a different location, the Dispatcher address list will be used. The Agent will randomly connect to these Dispatchers if communication over IP port 3163 is possible. This is the recommended combination for WAN connections.

---

**Notes**

- To detect Dispatchers, an Agent uses multicast 224.1.1.50 to port 3163. If Agents should only communicate with Dispatchers in their own subnet within a WAN, use **Dispatcher Discovery: Autodetect** for each subnet. This allows you to add Dispatchers to particular sites without having to change configuration settings.
- When you configure the Dispatcher discovery settings, the Dispatcher address list must be created manually, by clicking **Add** in the Dispatcher discovery window and specifying the Dispatcher addresses.
- When you change any of the Dispatcher detection settings, either at global level or at Agent level, the Agent will store the previous Dispatcher detection settings as the Last Known Good Configuration (LKGC). These are the settings with which it is certainly possible to connect to a Dispatcher.
- The Agent will only revert to the LKGC if it is not possible to connect to any Dispatcher immediately after changing the Dispatcher detection settings.
- The Agent will not revert to the LKGC if it is not possible to connect to any Dispatcher at a later stage (for example, because the Dispatchers get offline).
- When an Agent has reverted to the LKGC, it will retry the new settings every 5 minutes. The Agent will keep trying with these settings until it can connect to a Dispatcher or until it gets new Dispatcher detection settings.
### 3.8.3 Connection settings

Global Settings allow you to custom-fit the RES ONE Automation environment to your network environment. These settings allow you to define the general behavior of your RES ONE Automation environment, and the default behavior of Dispatchers and Agents.

<table>
<thead>
<tr>
<th>Connection settings</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Bandwidth throttling**             | Controls the amount of data that can be downloaded per second by an Agent from a Dispatcher.  
  - If you configure different values for this setting at Agent and Dispatcher level, the lowest value is used.  
  - If you disable this setting, the maximum download speed is used, depending on your network topology, available CPU performance and number of concurrent downloads.  
  Application deployment can cause a decrease in overall network performance, due to excessive bandwidth usage. RES recommends setting bandwidth throttling to the maximum download speed when deploying applications with RES ONE Automation. |
| **Resource download throttling**     | Controls the amount of Resource data that can be downloaded per second by a Dispatcher from the Datastore.  
  - If you disable this setting, the maximum download speed is used, depending on your network topology, available CPU performance and number of concurrent downloads.                                                                                                                                                                                                                      |
| **Protocol Encryption Dispatcher**   | Specifies whether communication between Dispatchers and Agents is secured. The communication uses TLS (Transport Layer Security) 1.2 if possible. The highest TLS version will be negotiated. The supported versions are TLS 1.2, 1.1, 1.0 and SSL (Secure Socket Layer) 3.0.  
  - This setting functions irrespective of database type used.  
  - Any changes to this setting take effect immediately.  
  - Each Dispatcher generates a self-signed certificate to encrypt communication with Agents. The most important properties of the self-signed certificate are:  
    - Encryption algorithm: RSA  
    - Hash algorithm: SHA1  
    - Key size: 1024 bits  
  - For Datastores encrypted with AES-256, the communication is automatically secured by protocol encryption. The global setting is not available in an AES-256 environment.                                                                                                                                 |
| **Dispatcher push communication**    | Specifies the port number used for push communication between the Dispatchers and the Agents+. The default port number is set to 3162.                                                                                                                                                                                                                           |
### 3.8.4 Dispatcher Proxy settings

At Setup > Global Settings, you can define the general settings (see page 35) of your RES ONE Automation environment, and the default settings of Dispatchers and Agents.

<table>
<thead>
<tr>
<th>Dispatcher Proxy settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dispatcher Proxy</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

If disabled, the Dispatchers will check if there are proxy settings configured at user level (for example, proxy settings via GPO using a service account). If so, these settings will be used to communicate with the RES ONE License Server. If not, the Dispatcher will communicate with the RES ONE License Server directly via the Internet and not via a proxy server.

**Notes**

- The proxy server configuration in RES ONE Automation will override any other proxy server settings you have set locally when communicating with the RES ONE License Server.
- For Agent+, IPv4 and IPv6 addresses are supported. For the legacy Agent, only IPv4 is supported.

### 3.8.5 Dispatcher Cache settings

Global Settings allow you to custom-fit the RES ONE Automation environment to your network environment. These settings allow you to define the general behavior of your RES ONE Automation environment, and the default behavior of Dispatchers and Agents.

<table>
<thead>
<tr>
<th>Dispatcher Cache settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dispatcher cache timing</td>
</tr>
<tr>
<td>Dispatcher cache availability</td>
</tr>
<tr>
<td>Dispatcher cache duration</td>
</tr>
</tbody>
</table>
3.8.6  Dispatcher WebAPI settings

At Setup > Global Settings, you can define the general settings (see page 35) of your RES ONE Automation environment, and the default settings of Dispatchers and Agents.

Dispatcher WebAPI settings

<table>
<thead>
<tr>
<th>WebAPI state</th>
<th>Specifies whether the Web API is disabled or enabled on Dispatchers. By default, the Dispatcher Web API is disabled for all Dispatchers.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• If Enabled, the Dispatcher Web API communicates over port 80.</td>
</tr>
<tr>
<td></td>
<td>• SSL certificate thumbprint: SSL certificate must be installed on the Dispatcher</td>
</tr>
<tr>
<td></td>
<td>• If you are hosting the web services on a different address (URL) and/or port number, you can specify an alternative base address and/or custom port number.</td>
</tr>
</tbody>
</table>

It is not necessary to manually configure these settings in the Dispatcher configuration file. The settings in the Dispatcher configuration file will be ignored when deploying the Dispatcher.

See also Web API for information on how to access the Web API and an overview of all available resources and actions.

Note: In PoSh mode, this functionality is not available.

3.8.7  Agent Cache settings

Global Settings allow you to custom-fit the RES ONE Automation environment to your network environment. These settings allow you to define the general behavior of your RES ONE Automation environment, and the default behavior of Dispatchers and Agents.

Agent Cache

<table>
<thead>
<tr>
<th>Agent resource folder Windows</th>
<th>Specifies the folder where Resources that are used by the Agents for Windows should be cached.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>By default, these Resources are stored in the folder &lt;Installation Path&gt;\Workspace. Because this is not always desirable, you can specify an alternative location, for example C:\Windows\Temp. You are allowed to use environment variables. If this folder does not exist, it will be created by the operating system.</td>
</tr>
<tr>
<td>Agent resource folder Unix/Linux/Mac OS X</td>
<td>Specifies the folder where Resources that are used by the Agents for Unix/Linux or Apple Mac OS X should be cached.</td>
</tr>
<tr>
<td></td>
<td>The default folder in which Resources are cached differs for each Unix/Linux or Apple Mac OS X operating system version. To make it easier to retrieve downloaded Resources, you can also specify an alternative location, for example /var/temp. If this folder does not exist, it will be created by the operating system.</td>
</tr>
</tbody>
</table>
### 3.8.8 Scheduling settings

Global Settings allow you to custom-fit the RES ONE Automation environment to your network environment. These settings allow you to define the general behavior of your RES ONE Automation environment, and the default behavior of Dispatchers and Agents.

| **Scheduling** | **Launch timeout** | Specifies the time that is allowed to elapse before an Agent needs to be started with the execution of a scheduled Job. If the Agent is unable to start the execution of the Job before the timeout expires, the Job will be canceled. For example, this option makes it possible to prevent situations in which Jobs remain scheduled on offline Agents. The default value is 1 week. |
| **Launch Window** | Specifies the time period during which an Agent is allowed to execute a Job. Depending on the **Scheduling** option, if outside launch window, the Job will fail, continue or wait for the next launch window, if it has not been executed during the launch window. For example, this option makes it possible to ensure that Tasks are executed at quiet times and not at peak times during the working day. |
| **Update of detailed Task result** | Specifies when the detailed results of Tasks are to be updated at **Scheduling**. |
| **Default Schedule Status** | Specifies the default schedule status of new Jobs. Normally, when a Job is scheduled, it is enabled by default. In certain situations, it may be desirable to have a safety net, to prevent the accidental scheduling of Jobs. |

| **Launch timeout** | Specifies the time that is allowed to elapse before an Agent needs to be started with the execution of a scheduled Job. If the Agent is unable to start the execution of the Job before the timeout expires, the Job will be canceled. For example, this option makes it possible to prevent situations in which Jobs remain scheduled on offline Agents. The default value is 1 week. |

| **Launch Window** | Specifies the time period during which an Agent is allowed to execute a Job. Depending on the **Scheduling** option, if outside launch window, the Job will fail, continue or wait for the next launch window, if it has not been executed during the launch window. For example, this option makes it possible to ensure that Tasks are executed at quiet times and not at peak times during the working day. |
| **Launch Window** | Specifies the time period during which an Agent is allowed to execute a Job. Depending on the **Scheduling** option, if outside launch window, the Job will fail, continue or wait for the next launch window, if it has not been executed during the launch window. For example, this option makes it possible to ensure that Tasks are executed at quiet times and not at peak times during the working day. |
| **Update of detailed Task result** | Specifies when the detailed results of Tasks are to be updated at **Scheduling**. |
| **Default Schedule Status** | Specifies the default schedule status of new Jobs. Normally, when a Job is scheduled, it is enabled by default. In certain situations, it may be desirable to have a safety net, to prevent the accidental scheduling of Jobs. |

- **Scheduled**: If selected, the new Job is enabled. This means that the Job will be executed at the time that was specified during Job scheduling. This is the default setting.
- **On Hold**: If selected, the Job is put “on hold”. Jobs that are put on hold will not be scheduled until manually resuming the Job (either by right-clicking the Job in the **Scheduling** node and selecting **Resume** or by editing the Job and selecting **Scheduled**). This global setting also applies to rescheduled Jobs.
Chapter 3: Setup

Behavior of launch window

- Jobs that are executed at a time that falls within the launch window will not be aborted when the launch window expires.
- Run Book Jobs in a Run Book that is executed during a time that falls within the launch window will be executed until the launch window expires: the next Run Book Job in the Run Book will fail.
- All system-initiated Job schedules will fail outside the launch window. This includes:
  - Reapply change path
  - Add/remove Agents from Teams
  - Command line schedules
  - RES ONE Workspace Integration
  - RES ONE Identity Director Integration
  - Dispatcher WebAPI schedules
- When an Agent is offline during the launch window, the Job will be canceled when the configured launch timeout expires. If the Agent gets online again before the timeout expires and within the launch window, the Job will be executed by the Agent.
- If you schedule an After every boot Job on an Agent in combination with the option If outside launch window: Wait for next window, the Job will be executed by the Agent after every reboot. If this moment falls outside the launch window of the Agent, this Job will be kept in queue until the next launch window.

There is an exception to this behavior though:

Suppose the following Jobs are scheduled on an Agent:

- Job 1: an After every boot Job that counts how many times the machine on which the Agent is running was rebooted.
- Job 2: a Run Book that contains a Task Reboot Computer.

The Agent will first execute Job 1, and set the reboot count to 1. After this, it will execute Job 2. When the Agent reboots during Job 2, you would expect that the Dispatcher creates two new Jobs, Job 1 and Job 2. However, this is not the case: after the reboot during Job 2, the Dispatcher checks whether Jobs have already been created from the After every boot Job. If this is the case, no further Jobs are created.

This behavior is to prevent Agents from entering a loop in which the Agent keeps rebooting and also applies if the option or when Agent is resumed after hibernate, sleep or standby is selected.
### 3.8.9 Job Notification settings

At **Setup > Global Settings**, you can define the general settings (see page 35) of your RES ONE Automation environment, and the default settings of Dispatchers and Agents.

<table>
<thead>
<tr>
<th>Job Notification</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Send mail message</strong></td>
</tr>
<tr>
<td>Specifies to whom a notification e-mail message is sent about the status of every Job that is executed. This setting is closely related to the Task <strong>Send E-mail</strong> and is disabled by default.</td>
</tr>
<tr>
<td>• When the option SSL Enabled is selected, the notification e-mail message will be sent over SSL. The communication of the SMTP server only supports Explicit SSL.</td>
</tr>
</tbody>
</table>

| **Send SNMP trap**                     |
| Specifies to which address notification SNMP traps are sent about the status of every Job that is executed. SNMP traps can be sent to specific IP (IPv4 and IPv6) addresses or broadcasted on the local network.Notifications are sent by a Dispatcher. This setting is closely related to the Task **Send SNMP trap** and is disabled by default. |

**Note**: When sending a request to an IPv6 destination, the SNMP Version should be set to SNMPv2c.

### 3.8.10 Security settings

Global Settings allow you to custom-fit the RES ONE Automation environment to your network environment. These settings allow you to define the general behavior of your RES ONE Automation environment, and the default behavior of Dispatchers and Agents.

<table>
<thead>
<tr>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Protect clear of audits</strong></td>
</tr>
<tr>
<td>You can optionally protect the Audit Trail with a password.</td>
</tr>
</tbody>
</table>

| **Remote Console**              |
| Specifies whether the Remote Console functionality is available. With this functionality, you can open remote Consoles on online Agents. See Open a remote Console on an Agent (on page 196). The option Remote Console is disabled by default. You can also set permissions on the Remote Console functionality. |

| **RunBookWho**                  |
| Specifies whether the RunBookWho functionality is available when configuring Run Book Jobs. |
| • **Enabled**: The option Use Run Book Parameter is available when configuring Run Book Jobs. With this option, you can parameterize the Who field with the parameter $[RunBookWho]$, to specify which Team(s) or Agent(s) should perform the Job. This is particularly useful when using the Run Book in a RES ONE Identity Director service. See Run Books (on page 159). |
| • With the setting Enhanced security, you are only allowed to schedule a Run Book with a $[RunBookWho]$ parameter configured in the Who field when using a login with RES ONE Automation Authentication. Before enabling this option, you need to reschedule all existing Run Book Jobs that are scheduled before RES ONE Automation 2015 SR3 and all existing Run Book Jobs that are scheduled using a login with Windows Authentication. Otherwise, these Jobs will fail. |
Security

- **Disabled**: the functionality will no longer be available for new Run Book Jobs. In existing Run Book Jobs, the functionality will still be available until a different sub-option is selected and saved. In that case, the parameter $\{RunBookWho\}$ will not be removed from the Run Book, but it can no longer be used for Job scheduling purposes.

Trusts Security

Specifies the behavior of Trusts in your RES ONE Automation environment. With Trusts, you can create "trusted" relationships between Agents, Modules and Resources, and so determine whether an Agent can execute a Job with these Modules and Resources. This further increases security in an environment and is especially useful in multi-tenant sites serving multiple customers. See Trusts (see page 269).

- **Disabled**: This setting disables Trusts. Use this setting if it is not or no longer necessary to use Trusts in your environment. When disabled, it will not be possible to configure Trusts or to assign administrative role permissions to them. The Trusts tabs on Agents, Team folders, Teams, Resource folders, Resources, Module folders and Modules will be hidden in the Console, as well as all Trusts columns in administrative roles. Any existing Trusts will no longer be applied at Job execution. Trusts Security is disabled by default.

- **Disabled, configure only**: This setting disables Trusts, but allows Console users that log on with administrative role Full Access to configure Trusts and to assign administrative role permissions to them: the Trusts tabs on Agents, Team folders, Teams, Resource folders, Resources, Module folders and Modules, and the Trusts columns in administrative roles will only be available for these users.

This setting is considered to be best practice when fine-tuning Trusts before enabling it in your environment: Any configured Trusts will NOT be applied at Job execution, and any Console users without the administrative role Full Access will not be able to configure Trusts. Once you have configured all relevant Trusts, you can safely enable Trusts Security.

The setting **Disabled, configure only** also applies in environments in which no login accounts exist and the default administrative role Full Access applies.

- **Enabled**: This setting enables Trusts. When Trusts Security is enabled, the Trusts tabs on Agents, Team folders, Teams, Resource folders, Resources, Module folders and Modules, and the Trusts columns in administrative roles will become available to all Console users (provided that their administrative role grants access to these items). Any Trusts in your environment will be applied at Job execution, and any administrative role permissions that you have assigned will also be applied when Console users log on with these administrative roles.
### Security

**Limit Job Export**

Specifies whether the Export Job Results functionality is limited to the access permissions of a Console user's administrative role. When enabled, Console users can only export the results of Jobs that are executed by Agents to which they have access (Read/Modify). Console users must have access to all Agents in the Job: if the Console user does not have access to one or more Agents in the Job, the results of this Job cannot be exported.

- If the Console user has insufficient permissions, any Job export functionality will be hidden in the Console, and neither will it be possible to export the results of these Jobs using a command line.
- At Administration > Security, if the option Limit task details when read permissions are set has been selected in the Console user's administrative role (Library > Modules node), and the Console user only has Read access to the Modules node, he can only export limited information.

The setting Limit Job Export is disabled by default.

**Limit Job Execution**

Specifies which Jobs Console users can see in the Jobs nodes, based on the Console user's administrative role(s).

When enabled, Console users can only view Jobs that are executed by Agents to which they have access (Read/Modify). In the Jobs node, the number of hidden job results will be shown as a text. Console users must have access to all Agents in the Job: if the Console user does not have access to one or more Agents in the Job, the Job will not be shown.

The setting Limit Job Execution is disabled by default.

**Allow agents to be added to a team programmatically**

Specifies whether Agents can be added to a Team via the Console, but also:

- During unattended installations of an Agent (by applying properties to its MSI file).
- Via a registry value.

If the setting is disabled, it will only be possible to add Agents to a Team via the Console. The setting Allow agents to be added to a team programmatically is enabled by default.

### 3.8.11 Snapshot Intelligence settings

Global Settings allow you to custom-fit the RES ONE Automation environment to your network environment. These settings allow you to define the general behavior of your RES ONE Automation environment, and the default behavior of Dispatchers and Agents.

**Snapshot Intelligence**

**When use of snapshot is detected**

Specifies what should happen if RES ONE Automation detects that a snapshot has been applied to an Agent. See Snapshot Intelligence (on page 197).

**When the last change path is reapplied**

Specifies whether specific types of Task should be skipped whenever a change path is reapplied automatically as a result of Snapshot Intelligence.

- Always skip shutdown Tasks for Agents that boot from a snapshot (read-only virtual disks that are used by for example Citrix Provisioning Servers), to prevent these Agents from ending up in a loop. Do not skip shutdown Tasks for regular Agents on which reboots will be necessary, for example after software installation.
3.8.12 Version Control settings

Global Settings allow you to custom-fit the RES ONE Automation environment to your network environment. These settings allow you to define the general behavior of your RES ONE Automation environment, and the default behavior of Dispatchers and Agents.

**Version Control**

<table>
<thead>
<tr>
<th>Use automatic versioning</th>
<th>Specifies whether or not versions (see page 272) of objects should be increased automatically, without user interaction.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• When set to Yes, the revision number will be increased automatically.</td>
</tr>
<tr>
<td></td>
<td>• When set to No, the Version Control window becomes available when saving changes to Modules, Projects, Run Books and Resources.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Version comment</th>
<th>Specifies whether a version comment is mandatory when updating an object.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• When set to Disabled, the comment cannot be edited and will be set to “Updated version.”</td>
</tr>
<tr>
<td></td>
<td>• When set to Optional, a comment is optional and may be empty. When selected, the newVersion Control window becomes available when saving changes to Modules, Projects, Run Books and Resources.</td>
</tr>
<tr>
<td></td>
<td>• When set to Mandatory, a comment must be specified. When selected, the new Version Control window becomes available when saving changes to Modules, Projects, Run Books and Resources.</td>
</tr>
</tbody>
</table>

3.8.13 Password Security Policy settings

Global Settings allow you to custom-fit the RES ONE Automation environment to your network environment. These settings allow you to define the general behavior of your RES ONE Automation environment, and the default behavior of Dispatchers and Agents.

- Use Password Security Policy settings to manage the behavior of password security on login accounts using RES ONE Automation Authentication.
- These global settings are only available at global level and cannot be set at Dispatcher, Team or Agent-level.
- These global settings cannot be set for individual login accounts. However, the administrator can choose to not apply the global Password Security Policy for individual login accounts (see page 213).

**Password Security Policy**

<table>
<thead>
<tr>
<th>Password age</th>
<th>Specifies the number of days a password can be used before a Console user is required to change it.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• The default value is 60 days.</td>
</tr>
<tr>
<td></td>
<td>• If you enable the Password age setting, the current date will be set as the “last password change date” for all existing RES ONE Automation logins. In environments with any login accounts, this can take up some time.</td>
</tr>
<tr>
<td></td>
<td>• When the password will expire within 5 days, the user gets a message to change their password manually. When the password has actually expired, the Change my password window will open automatically.</td>
</tr>
<tr>
<td></td>
<td>• When the password is changed by the administrator, the current date will also be set as the “last password change date” for the selected login</td>
</tr>
</tbody>
</table>
### Password Security Policy

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
</table>
| Password complexity          | Specifies the number of required character types in the password (uppercase, lowercase, numbers and symbols). This setting has a fixed value of 3 character types.  
  - You can use any of the following symbols: ~ ! @ # $ % ^ & * _ + = \ - | |
| Password history             | Specifies the number of passwords that will be stored; this determines how frequently old passwords can be reused.  
  - The default value is 6 passwords.  
  - RES ONE Automation uses FIPS-compliant cryptography to store the password history. |
| Minimum password length      | Specifies the minimum number of characters that need to be used in new passwords. The default value is 8 characters. The maximum password length is 50 characters. |
| Account lockout threshold    | Specifies the number of password retries before a user account will be locked. The default value is 6 retry attempts. |

#### Warning

When enabling the global Password Security Policy, it will become effective immediately. The password complexity check will be applied upon next password change.

**Best practice:**

When implementing the global Password Security Policy, first exclude the login accounts and then enable the global Password Security Policy.

#### Notes

- All global Password Security Policy settings are reported in the corresponding Instant Report. All changes made on the settings will be logged in the Audit Trail.
- Interaction via command line, Web API and integration with other RES products, are also included in the Password Security Policy. The only exception is, when using the Console via the Web API and when upgrading RES ONE Automation, the account will not be locked out when entering too many incorrect passwords.
### 3.8.14 Other Settings

At Setup > Global Settings, you can define the general settings (see page 35) of your RES ONE Automation environment, and the default settings of Dispatchers and Agents.

<table>
<thead>
<tr>
<th>Other settings</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Synchronize clock</strong></td>
<td>Synchronizes the clock on all Agents with the Datastore. This setting is disabled by default.</td>
</tr>
<tr>
<td><strong>Identification Method</strong></td>
<td>Specifies how Consoles, Dispatchers and Agents are identified in your RES ONE Automation environment. When you deploy a Console, Dispatcher or Agent, it receives a unique RES ONE Automation-generated GUID by which it is identified. In certain scenarios, this identification method can lead to obsolete Consoles, Dispatchers and Agents (e.g. when you reinstall the operating system on a computer, when virtual machines use the same image, or when you deploy RES ONE Automation components on multiple computers using imaging). To prevent this, you can change their identification method. The selected identification method applies to all Consoles, Dispatchers and Agents:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Unique RES ONE Automation-generated computer GUID</strong>: Selected by default.</td>
</tr>
<tr>
<td></td>
<td>• <strong>MAC address of the first enabled network interface card</strong>: If a Console, Dispatcher or Agent has multiple network cards, RES ONE Automation will use the MAC address of the first enabled network card, based on the order as defined on the Console, Dispatcher and Agent by Microsoft Windows. You can find this order in Microsoft Windows by clicking Start &gt; Control Panel &gt; Network Connections &gt; Advanced &gt; Advanced Settings.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Computer domain name &amp; NetBIOS name</strong>: Identifies a Console, Dispatcher or Agent by the domain name or workgroup name and NetBIOS name of the computer on which it runs.</td>
</tr>
<tr>
<td><strong>Wake-on-LAN</strong></td>
<td>Specifies which broadcast address should be used when selecting the option <em>Use Wake-on-LAN when Job starts</em> during Job scheduling.</td>
</tr>
<tr>
<td></td>
<td>• The default global IPv4 broadcast address is 255.255.255.255.</td>
</tr>
<tr>
<td></td>
<td>• However, using a subnet broadcast address may be useful in environments in which Wake-on-LAN does not function correctly when using a global broadcast address. When using the subnet broadcast address, RES ONE Automation will broadcast the “wake-up” packet to the last known subnet broadcast address of the Agent that executes the Job (for example, 172.31.255.255).</td>
</tr>
<tr>
<td></td>
<td>• The default global IPv6 broadcast address is set to ff02::1 (i.e. all nodes on the local network segment).</td>
</tr>
<tr>
<td></td>
<td>• The default port that RES ONE Automation uses for Wake-on-LAN is port 3163. However, you can select an alternative UDP port number if necessary.</td>
</tr>
</tbody>
</table>
**Auto Update Agent**

Specifies whether Agents are automatically updated.

A Dispatcher manages all communication between the Datastore and an Agent by contacting the Datastore at regular intervals. If a new Job or a new version of RES ONE Automation is available for the Agent, the Dispatcher will download all necessary data from the Datastore and store it. In its turn, the Agent contacts the Dispatcher at regular intervals. If a new Job or a new version of RES ONE Automation is available, the Agent will download all necessary data from the Dispatcher and perform the Job or update to the latest version. This may lead to issues in large environments with many Agents that all update at the same time. With this setting you can manage this behavior. The options are:

- **Enabled**: at upgrade of RES ONE Automation, all Agents will be updated instantly.
- **Disabled until Agent restart**: at upgrade of RES ONE Automation, the selected Agents will not be updated until the Agents restart or reboot.
- **Disabled entirely**: at upgrade of RES ONE Automation, the Agents will not be updated after the Agents restart or reboot.

**Note**: In PoSh mode, this functionality is not available. By default, all Agents are updated automatically in order to support the PoSh mode.

---

**Warning**

When using MAC address of the first enabled network interface card as identification method, it is important that no network adapters are enabled or disabled after the Console and the Agent have been registered successfully for the first time. Enabling or disabling network adapters may affect the MAC address selected by the Console and the Agent to identify themselves the next time they are restarted. For reliable results, it is important that the Console and the Agent select the same MAC address to identify themselves each time they start up.

**How does the MAC address selection work?**

The Agent selects the MAC address of the first enabled Ethernet adapter (including Bluetooth); if none were found, it will use the MAC address of the enabled Wireless LAN adapter. The order of the list of adapters used for selection is determined by Microsoft Windows.
Chapter 4: Jobs

The Jobs node of the Console allows you to manage the Jobs in your environment. You can use this node to:

- Schedule Jobs
- View the details of active Jobs
- View the results of a Job

When you schedule a Job, it is shown at Jobs > Scheduling. When a Job is executed, it moves from Scheduling to Activity. When a Job has finished, you can view its results at Job History.

4.1 Scheduling

The Tasks that have been set up in your RES ONE Automation environment can be executed by scheduling a Job. A Job combines your Modules/Projects/Run Books (What) and Agents/Teams (Who) with a specific time (When). In effect, when you schedule a Job, you are saying: “the following Job should be performed by the following Agents at the following moment.”

- As long as a scheduled Job is not being executed, you can Edit it, Delete it or put the Job On Hold.
- When a Job is executed, it will move from Scheduling to Activity.
- When a Job has finished, you can view its results at Job History.

4.1.1 Where to find Scheduling

<table>
<thead>
<tr>
<th>What</th>
<th>Where</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scheduling a Job</td>
<td>• At Jobs &gt; Scheduling, click New in the command bar.</td>
</tr>
<tr>
<td></td>
<td>- or -</td>
</tr>
<tr>
<td></td>
<td>• At Library &gt; Modules/Projects/Run Books, right-click a Module/Project/Run Book and click Schedule Job.</td>
</tr>
<tr>
<td></td>
<td>- or -</td>
</tr>
<tr>
<td></td>
<td>• At Topology &gt; Agents/Teams, right-click one or more Agents/Teams and click Schedule Job.</td>
</tr>
<tr>
<td>Rescheduling a Job</td>
<td>• At Jobs &gt; Job History, right-click a Job and click Reschedule Job.</td>
</tr>
<tr>
<td></td>
<td>- or -</td>
</tr>
<tr>
<td></td>
<td>• At Library &gt; Modules/Projects/Run Books, open a Module/Project/Run Book, click the Job History tab, right-click a Job and click Reschedule Job.</td>
</tr>
<tr>
<td></td>
<td>- or -</td>
</tr>
<tr>
<td></td>
<td>• At Topology &gt; Agents/Teams, open an Agent/Team, click the Job History tab, right-click a Job and click Reschedule Job.</td>
</tr>
<tr>
<td>Reapplying the Job history</td>
<td>• At Topology &gt; Agents/Teams, open an Agent/Team, click the Job History tab, select a Job and click Reapply Job History.</td>
</tr>
</tbody>
</table>
4.1.2  Scheduling settings

You configure the settings of a Job in the New/Edit Job window, which opens when adding or editing a Job. This window consists of the following tabs:

- **Properties tab**: Use this tab to specify:
  - **When** (on page 54) the Job should be executed
  - **What** (on page 60) should be executed
  - **Who** (on page 60) should execute the Job
  - If a Job should run in parallel (see page 63) with other Jobs
- **Resulting Tasks** and **Resulting Agents tabs** (not available for Run Books): Use these tabs to view which Tasks will be executed and by which Agents. If you have edited any of these Tasks or Agents after the Job was scheduled (for example, when you edited the Tasks or Agents used in a Recurring Job), you can update these changes in the Job:
  - Click **Reload Resulting Tasks** on the Resulting Tasks tab to update changes to Tasks.
  - Click **Reload Resulting Agents** on the Resulting Agents tab to update changes to Agents.
- **Resulting Jobs tab** (only available for Run Books): Use this tab to view which Jobs will be executed and by which Agents. If you have edited any of these Jobs or Agents after the Run Book was scheduled, you can update these changes by clicking **Reload Resulting Jobs**.
- **Job Parameters tab**: Use this tab to review the Job parameter values. This tab is only shown if the Input settings of the used parameters specify that new values are required when scheduling the Job.
  - If the parameters on the tab have not been reviewed yet, a warning sign will be shown.
  - If necessary, you can resize the New/Edit Job window to fit the contents of the value fields. This window adjusts automatically if the Job parameters contain Module parameter conditions. See Parameters (on page 257).
  - Select **Use parameter values from file** if you want to input text parameter values from a CSV file, instead of providing these values manually. This makes it possible to feed information into Tasks from a spreadsheet. See Using parameter values from a CSV file (on page 61).

### Notes

- As long as a scheduled Job is not being executed, for example because it is not scheduled to run immediately, you can **Edit** it, **Delete** it or put the Job **On Hold**.
- When a Job is executed, it will move from Scheduling to Activity.
- When a Job has finished, you can view its results at Job History.
- You can also schedule Jobs directly from the Agents and Teams nodes at Topology, and from the Modules, Projects and Run Books nodes at Library.
- You can also schedule a Job unattended (see page 302), for example if you want to schedule a Job using a third-party product.
- Enabling or disabling the Auto-Refresh option to automatically update items affects ALL nodes in which this functionality is available (all Jobs and Topology nodes).
4.1.3 Scheduling a Job: When, What, Who

When

When you schedule a Job, the first thing you need to determine is when the Job should be executed.

Configuration

- **Immediately**: select this option to execute a Job immediately.
  - **Schedule offline Agents for next boot**: select this option to postpone the execution of the Job for all offline Agents. RES ONE Automation will automatically determine whether Agents are online or offline and create Immediately and/or After next boot Jobs, depending whether online or offline Agents are found. When one of the offline Agents gets online, the After next boot Job will be executed and remains active until all specified Agents have completed the Job. If you do not select this option, the Job will fail on all offline Agents, depending on the Launch timeout settings (at Setup > Global Settings).
  - **or when Agent is resumed after hibernate, sleep or standby**: select this option when scheduling Jobs on Agents running on Windows laptops: When a laptop goes into hibernation, sleep or standby, the Agent running on the laptop goes offline. However, when the laptop is resumed and the Agent comes back online again, any Jobs that are scheduled for next boot on offline Agents will not be executed, because the laptop has not booted. As many users rarely turn off their laptop, but put it in hibernation, sleep or standby instead, this can cause a backlog in Jobs that need to be executed on these Agents. By selecting the option Schedule offline Agents for next boot in combination with the option or when Agent is resumed after hibernate, sleep or standby, Jobs scheduled on these Agents will also be executed.
  - The option or when Agent is resumed after hibernate, sleep or standby is also available for After next boot and After every boot Jobs. The option is not supported on non-Windows machines (Unix/Linux and Mac OS X). When selected, the non-Windows Agent will execute the Job after a reboot of the machine.
- **Scheduled**: select this option to execute the Job on a specific date and time, for example at night or during the weekend when network traffic is at its lowest.
- **Recurring**: select this option to execute the Job recurrently at specific intervals. Scheduling a Job recurrently can be useful for recurring maintenance Tasks such as defragmentation or disk space queries. For more granular control over your recurrence schedule, consider using Schedule using Cron (see below).
  - When you edit a Recurring Job that is set up with a schedule that is Effective Immediately, the schedule will be reset to that moment. This means that the Job will be executed again immediately, after which it is executed recurrently at the specified intervals.
- **Select Schedule using Cron** to schedule a Job recurrently using a cron expression (on page 58). Once scheduled, the Job is considered a recurring Job.
- **After next boot**: select this option to execute the Job after the next boot of the specified Agents.
- **After every boot**: select this option to execute the Job after every boot of the specified Agents. When you schedule a Job After every boot to a Team, this will be a dynamic schedule. If the Team is enabled, the Job will automatically run on all Agents added to that Team before or after the Job has been scheduled.
- **After new Agent registered**: select this option to execute the Job after new Agents get online.
- **Use Wake-On-LAN when Job starts**: select this option to boot Agents automatically when the Job starts, by broadcasting a "wake-up" packet to the Agent. This option forces the Agent to use UTC time. By default, RES ONE Automation uses a global broadcast address, 255.255.255.255 and UDP port 3163 for Wake-on-LAN. If necessary, you can set up different Wake-on-LAN settings at Setup > Global Settings. See Using Wake-on-LAN (on page 59). The option Use Wake-On-LAN when Job starts is only available for Immediate, Scheduled and Recurring Jobs.
If outside launch window: select this option to specify what should happen to the Job when it is scheduled at a time that falls outside of the launch window. A launch window specifies the time period during which Agents are allowed to execute Jobs. The availability of the option If outside launch window depends on the special permissions of your administrative role (on page 217). You can set up a launch window at Global Settings (see page 35) and set exceptions for individual Teams and Agents. See Launch window options (on page 56) for an overview of all possible combinations.

- **Fail the Job**: select this option to fail the Job when it is scheduled during a time that falls outside of the launch window of the Agent.
- **Override window and continue**: select this option to schedule the Job irrespective of the launch window of the Agent. This option is only available if the administrative role of the administrator permits the use of this functionality.
- **Wait for next window**: select this option to keep the Job in queue until the next launch window when it is scheduled during a time that falls outside of the launch window of the Agent. If the global setting Launch timeout is set to less than one week, a warning message will be shown that this combination may cause the Job to fail on Agents that have infrequent launch windows.

- **Maximum Job duration**: use this field to configure a timeout that specifies the maximum Job duration. This can be useful to prevent situations in which Jobs cannot be completed if an Agent that executes a Task goes offline during execution, for example because it is serviced, rebooted, etc. Because this Task will remain active, the Job will not finish. By specifying a maximum Job duration, you can prevent these situations: if a timeout has been specified, the Job will be canceled when the timeout expires.

  - **Maximum Job duration** is disabled by default. When enabled, a default value of 10 minutes will be shown.

  - In Run Books, if a Run Book Job is not completed within the specified timeout, it will change the status of any active Agent to Timed out. Subsequent Run Book Jobs will continue as normal. The end result of the Run Book will be Completed with errors.

- **Timing**: specify in this field whether to use local time of the Agent or UTC time of the Datastore. The latter option is useful in large environments, where the Datastore is located in a different time zone than the Agent.

  - **Scheduled**: clear this option to (temporarily) put the Job On Hold. The Job will not be executed until the Scheduled option is selected again.

  - **Select Schedule in parallel with other jobs** to allow the Job to run in parallel (see page 63) instead of serialized.

---

**Note**

When downgrading to an older version of RES ONE Automation, you need to open the existing Jobs scheduled After every boot to a Team, click on Reload Resulting Agents and click OK. Otherwise, the existing Jobs scheduled After every boot will never run on Agents, since older versions of RES ONE Automation require these Jobs to have been scheduled to Agents at schedule time.
### Launch window options

<table>
<thead>
<tr>
<th>Jobs scheduled immediately</th>
<th>Fail the Job</th>
<th>Jobs that are scheduled during a time that falls outside of the launch window of the Agent will fail.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Override window and continue</td>
<td>Jobs will be scheduled irrespective of the launch window of the Agent. This option is only available if the administrative role of the administrator allows the use of this functionality.</td>
<td></td>
</tr>
<tr>
<td>Wait for next window</td>
<td>Jobs that are scheduled during a time that falls outside of the launch window of the Agent will be kept in queue until the next launch window. If the global setting Launch timeout is set to less than one week, a warning message will be shown that this combination may cause the Job to fail on Agents that have infrequent launch windows.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Jobs scheduled at a specific time</th>
<th>Fail the Job</th>
<th>Jobs that are scheduled during a time that falls outside of the launch window of the Agent will fail.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Override window and continue</td>
<td>Jobs will be scheduled irrespective of the launch window of the Agent. This option is only available if the administrative role of the administrator allows the use of this functionality.</td>
<td></td>
</tr>
<tr>
<td>Wait for next window</td>
<td>Jobs that are scheduled during a time that falls outside of the launch window of the Agent will be kept in queue until the next launch window. If the global setting Launch timeout is set to less than one week, a warning message will be shown that this combination may cause the Job to fail on Agents that have infrequent launch windows.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Jobs scheduled recurring</th>
<th>Fail the Job</th>
<th>Jobs that are scheduled during a time that falls outside of the launch window of the Agent will fail.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Override window and continue</td>
<td>Jobs will be scheduled irrespective of the launch window of the Agent. This option is only available if the administrative role of the administrator allows the use of this functionality.</td>
<td></td>
</tr>
<tr>
<td>Wait for next window</td>
<td>Jobs that are scheduled during a time that falls outside of the launch window of the Agent will be kept in queue until the next launch window. If the global setting Launch timeout is set to less than one week, a warning message will be shown that this combination may cause the Job to fail on Agents that have infrequent launch windows.</td>
<td></td>
</tr>
</tbody>
</table>
### Jobs scheduled using Cron

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fail the Job</td>
<td>Jobs that are scheduled during a time that falls outside of the launch window of the Agent will fail.</td>
</tr>
<tr>
<td>Override window and continue</td>
<td>Jobs will be scheduled irrespective of the launch window of the Agent. This option is only available if the administrative role of the administrator allows the use of this functionality.</td>
</tr>
<tr>
<td>Wait for next window</td>
<td>Jobs that are scheduled during a time that falls outside of the launch window of the Agent will be kept in queue until the next launch window. If the global setting Launch timeout is set to less than one week, a warning message will be shown that this combination may cause the Job to fail on Agents that have infrequent launch windows.</td>
</tr>
</tbody>
</table>

### Jobs scheduled after next boot

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fail the Job</td>
<td>Jobs that are scheduled during a time that falls outside of the launch window of the Agent will fail.</td>
</tr>
<tr>
<td>Override window and continue</td>
<td>Jobs will be scheduled irrespective of the launch window of the Agent. This option is only available if the administrative role of the administrator allows the use of this functionality.</td>
</tr>
<tr>
<td>Wait for next window</td>
<td>After the boot of the Agent, any Jobs that are scheduled during a time that falls outside of the launch window of the Agent will be kept in queue until the next launch window. If the global setting Launch timeout is set to less than one week, a warning message will be shown that this combination may cause the Job to fail on Agents that have infrequent launch windows.</td>
</tr>
</tbody>
</table>

### Jobs scheduled after every boot

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fail the Job</td>
<td>Jobs that are scheduled during a time that falls outside of the launch window of the Agent will fail.</td>
</tr>
<tr>
<td>Override window and continue</td>
<td>Jobs will be scheduled irrespective of the launch window of the Agent. This option is only available if the administrative role of the administrator allows the use of this functionality.</td>
</tr>
<tr>
<td>Wait for next window</td>
<td>After every boot of the Agent, any Jobs that are scheduled during a time that falls outside of the launch window of the Agent will be kept in queue until the next launch window. If the global setting Launch timeout is set to less than one week, a warning message will be shown that this combination may cause the Job to fail on Agents that have infrequent launch windows.</td>
</tr>
</tbody>
</table>
### Jobs scheduled after new Agent registers

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fail the Job</strong></td>
<td>Jobs that are scheduled during a time that falls outside of the launch window of the Agent will fail.</td>
</tr>
<tr>
<td><strong>Override window and continue</strong></td>
<td>Jobs will be scheduled irrespective of the launch window of the Agent. This option is only available if the administrative role of the administrator allows the use of this functionality.</td>
</tr>
<tr>
<td><strong>Wait for next window</strong></td>
<td>After the Agent registers, any Jobs that are scheduled during a time that falls outside of the launch window of the Agent will be kept in queue until the next launch window. If the global setting <strong>Launch timeout</strong> is set to less than one week, a warning message will be shown that this combination may cause the Job to fail on Agents that have infrequent launch windows.</td>
</tr>
</tbody>
</table>

### Cron expression

When using the option **Schedule using Cron**, you can execute the Job at a specific interval using a cron expression. The cron expression makes it possible to use more advanced options. For example, scheduling every 'first' 'Friday' of every '2' months at 23:45h. This translates to the cron expression: `0 45 23 * 1/2 FRI#1 *`

In short, the cron expression is a string that consists of five to seven fields separated by white space that represents a set of times.

The following cron formats are supported:

- `MINUTES HOURS DAYS MONTHS DAYS-OF-WEEK`
- `MINUTES HOURS DAYS MONTHS DAYS-OF-WEEK YEARS`
- `SECONDS MINUTES HOURS DAYS MONTHS DAYS-OF-WEEK`
- `SECONDS MINUTES HOURS DAYS MONTHS DAYS-OF-WEEK YEARS`

The following values are supported:

<table>
<thead>
<tr>
<th>Field name</th>
<th>Allowed values</th>
<th>Allowed special characters</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seconds</td>
<td>0 - 59</td>
<td>* , - /</td>
<td>Optional</td>
</tr>
<tr>
<td>Minutes</td>
<td>0 - 59</td>
<td>* , - /</td>
<td></td>
</tr>
<tr>
<td>Hours</td>
<td>0 - 23</td>
<td>* , - /</td>
<td></td>
</tr>
<tr>
<td>Day of month</td>
<td>1 - 31</td>
<td>* , - / ? L W</td>
<td></td>
</tr>
<tr>
<td>Month</td>
<td>1 - 12 or JAN-DEC</td>
<td>* , - /</td>
<td></td>
</tr>
<tr>
<td>Day of week</td>
<td>0 - 6 or SUN-SAT</td>
<td>* , - / ? L #</td>
<td></td>
</tr>
<tr>
<td>Year</td>
<td>0001 - 9999</td>
<td>* , - /</td>
<td>Optional</td>
</tr>
</tbody>
</table>

For more information, please refer to existing documentation on cron expressions (for example, [http://en.wikipedia.org/wiki/Cron](http://en.wikipedia.org/wiki/Cron)).
Chapter 4: Jobs

Exceptional behavior to the Job scheduling combination “After every boot” and “Wait for next window”

Exceptional behavior

If you schedule an After every boot Job on an Agent in combination with the option if outside launch window: Wait for next window, the Job will be executed by the Agent after every reboot. If this moment falls outside the launch window of the Agent, this Job will be kept in queue until the next launch window.

There is an exception to this behavior though:

Suppose the following Jobs are scheduled on an Agent:

- Job 1: An After every boot Job that counts how many times the machine on which the Agent is running was rebooted.
- Job 2: A Run Book that contains a Task Reboot Computer.

The Agent will first execute Job 1, and set the reboot count to 1. After this, it will execute Job 2. When the Agent reboots during Job 2, you would expect that the Dispatcher creates two new Jobs, Job 1 and Job 2. However, this is not the case: after the reboot during Job 2, the Dispatcher checks whether Jobs have already been created from the After every boot Job. If this is the case, no further Jobs are created.

This behavior is to prevent Agents from entering a loop in which the Agent keeps rebooting and also applies if the option or when Agent is resumed after hibernate, sleep or standby is selected.

Using Wake-on-LAN

When a Job is scheduled to use Wake-on-LAN (WOL), all Dispatchers broadcast a WOL packet containing 16 times the MAC address of the Agent(s) that need to wake up. For WOL to work, the receiving computer’s BIOS needs to support WOL and it needs to be enabled in the BIOS. Broadcasts need to be supported on the (VLAN) switches between the Dispatcher(s) and the Agent(s).

RES ONE Automation follows the standard WOL Magic Packet procedure with packet type UDP. This procedure is described at http://en.wikipedia.org/wiki/Wake-on-LAN.

The default global broadcast address is 255.255.255.255. Using a subnet broadcast address may be useful in environments in which Wake-on-LAN does not function correctly when using a global broadcast address. When using the subnet broadcast address, RES ONE Automation will broadcast the “wake-up” packet to the last known subnet broadcast address of the Agent that executes the Job (for example, 172.31.255.255).

The default port that RES ONE Automation uses for Wake-on-LAN is port 3163, but you can select an alternative UDP port number if necessary. See Global Settings (see page 35).
What

After specifying **when** a Job is scheduled to be executed, you can specify **what** needs to be executed: which Modules and/or Projects or Run Books.

**Configuration**

- To specify what needs to be executed, click the **What** field in the **Add/Edit Job** window. The **Select Run Book/Project/Module** window opens. From this window, you can select a Run Book, a Project or **one or more** Modules. You can also add new Modules, Projects or Run Books and search for Modules, Projects and Run Books.

Who

After specifying what needs to be executed, you can specify which Agents and/or Teams need to execute the Job.

**Configuration**

- When selecting Agents that should perform the Job, you can filter on Agents that are member of a Team or on Agents that executed a specific Project or Module.
- You can filter the results further by searching on all Agents, only licensed and online Agents, offline Agents, or no Agents and include Teams in the search results.
- As an alternative to selecting Agents manually, you can add multiple Agents at once by copying them from a list of Agents separated by semi-colons (";"), comma’s (","), or tabs and pasting them directly into the **Who** field.
Using parameter values from a CSV file

When scheduling a Job, the option Use parameter values from file makes it possible to input Text parameter and Password parameter values from a CSV file, instead of providing these values manually. This makes it possible to schedule Tasks using information obtained from third-party sources, such as change management systems. The same CSV file can be used for several Tasks, so that one spreadsheet can provide many different Tasks with parameter values. This makes it possible to input text parameter values in bulk into large numbers of Tasks.

Example scenario: creating large numbers of users in bulk

Each new school year, hundreds of new students enroll. For each new student, the administrator needs to run a Module that contains the Tasks Create Active Directory User and Manage Active Directory User. The Tasks are configured with Module parameters such as $[FullName], $[UserLogonName], $[CreateIn], $[passwd], etc. For each new student, the administrator needs to schedule the Module and provide the parameter values.

With the option Use parameter values from file, the Module no longer needs to be scheduled separately for each student. Instead, a CSV file refers to the relevant parameters and then provides the relevant information for each student. For example:

```
FullName,UserLogonName,passwd, CreateIn
Amanda Cavendish,cavendisha,<password>,"OU=EngLit,OU=Students,OU=Campus1,DC=university,DC=local"
Dan Harris,harrisd,<password>,"OU=Maths,OU=Students,OU=Campus2,DC=university,DC=local"
Andrew Williams,williamsa,<password>,"OU=German,OU=Students,OU=Campus2,DC=university,DC=local"
Milosz Nowicki,nowickim,<password>,"OU=EngLit,OU=Students,OU=Campus1,DC=university,DC=local"
```

If the values in a column in the CSV file do not correspond with the desired values for a specific parameter, you can select a different column of values for a specific parameter by right clicking the parameter, selecting Select input column and choosing the relevant column.

RES ONE Automation will execute the Module once for each row in the CSV file. So, if the CSV contains 400 rows of user information, then RES ONE Automation will execute the Module 400 times and so create all 400 users - while the administrator only had to schedule the Module once.

Extra option for Run Books

For Run Books only, you can choose how RES ONE Automation should use the parameter values from the CSV file:

- **Use on Task level** results in more Tasks per Job. This is the fastest option, because certain Tasks will be executed simultaneously. However, if one Task fails, the entire Job fails.
- **Use on Job level** results in fewer Tasks per Job. Tasks will be spread out over more Jobs, which are executed in turn. This means that if one Job fails, RES ONE Automation will move on to the next Job. This option also provides more detailed information in the Job results.

Example

Suppose you have a CSV file with entries for user1 and user2. You also have a Run Book with 3 Run Book Jobs:
Chapter 4: Jobs

- Run Book Job 1 contains a Module Query Disk Space on a file server
- Run Book Job 2 contains a Module Create Active Directory user with parameters
- Run Book Job 3 contains a Project with 2 Modules that create a home folder on a file server and set permissions on this home folder. This Project contains parameters.

When you schedule the Run Book and specify the CSV file on the Job Parameters tab, choose how RES ONE Automation should use the parameter values from the CSV file:

- On Task level
- On Job level

When the Run Book has been executed, the Job results will be:

**On Task level:**

<table>
<thead>
<tr>
<th>Job</th>
<th>Modules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job 1</td>
<td>Module Query Disk Space on file server</td>
</tr>
<tr>
<td>Job 2</td>
<td>Module Create Active Directory user for user1</td>
</tr>
<tr>
<td>Job 2</td>
<td>Module Create Active Directory user for user2</td>
</tr>
<tr>
<td>Job 3</td>
<td>Module Create home folder for user1</td>
</tr>
<tr>
<td>Job 3</td>
<td>Module Set permissions on home folder for user1</td>
</tr>
<tr>
<td>Job 3</td>
<td>Module Create home folder for user2</td>
</tr>
<tr>
<td>Job 3</td>
<td>Module Set permissions on home folder for user2</td>
</tr>
</tbody>
</table>

**On Job level:**

<table>
<thead>
<tr>
<th>Job</th>
<th>Modules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job 1</td>
<td>Module Query Disk Space on file server</td>
</tr>
<tr>
<td>Job 2</td>
<td>Module Create Active Directory user for user1</td>
</tr>
<tr>
<td>Job 3</td>
<td>Module Create home folder for user1</td>
</tr>
<tr>
<td>Job 3</td>
<td>Module Set permissions on home folder for user1</td>
</tr>
<tr>
<td>Job 4</td>
<td>Module Create Active Directory user for user2</td>
</tr>
<tr>
<td>Job 5</td>
<td>Module Create home folder for user2</td>
</tr>
<tr>
<td>Job 5</td>
<td>Module Set permissions on home folder for user2</td>
</tr>
</tbody>
</table>

**Notes**

- The option Use parameter values from file can only be used for Text parameters and Password parameters. Other types of parameters (such as list or credentials) cannot be filled with values imported from a CSV file. Note that parameters can be embedded, so the values of a list parameter can be a set of text parameters - and those text parameters can be filled from a CSV file.
- RES ONE Automation only supports comma-separated CSV files. Other formats, such as tab-separated or semi-colon separated are not supported.
- Empty rows in the CSV file are skipped.
- Use double quotes (“”) around parameter values that include comma’s (such as the Active Directory paths used in the example above).
- Input settings are not evaluated if a parameter's value is imported from a CSV file.
- In Run Books, it is not possible to use parameter values from a CSV file in conditions based on parameters whose values are based on other parameters. See Conditions (on page 233).
- In Run Books, when selecting the option Use Run Book parameter, it is not possible to use parameter values from a CSV file.
Schedule Job in parallel

With parallel processing, you are able to run multiple Jobs simultaneously when scheduled to the Agent+. This makes it possible to start another Job on the same machine without waiting for the previous Job to finish. This could be useful if you, for instance, want to back up the Oracle database (Job 1) and back up the Security Event Log (Job 2) on Server A. Or, you continuously monitor the state of a print queue and if the Job fails, stop the spooler service, clear the stuck job and restart the service again. These Jobs could run in parallel.

At Jobs > Scheduling, when configuring a new Job, select the option Schedule in parallel with other jobs to allow this Job to be scheduled in parallel instead of serialized (default). At Jobs > Scheduling, Activity and Job History, you can view whether the Job has run in parallel or not. It is also possible to schedule the Job in parallel via the command line option /parallel.

Keep in mind that:

- The Tasks in the Module, Project or Run Book will always run serialized in the order as configured in that Job.
- When multiple Jobs are scheduled, in parallel and serialized, the following applies:
  - When a serialized Job is scheduled to run on an Agent+, the Agent+ will wait for all other running Jobs (parallel or serialized) to finish before starting the serialized Job.
  - When a serialized Job is started on an Agent+, no other Jobs (parallel or serialized) will be started on that Agent until the serialized Job has finished.
  - For example, if you schedule a serialized Job at 9 AM, and two parallel Jobs at 10 AM, the parallel Jobs will only start when the serialized Job has finished. [More examples]

More examples of parallel processing in combination with serialized Jobs:

**Example 1**

You schedule:

- one serialized Job (A) and two parallel Jobs (B and C) at 9 AM

The order of execution will be:

- Job A starts at 9 AM
- Job A ends at 9.30 AM
- Job B and C start at 9.30 AM in parallel

**Example 2**

You schedule:

- five serialized Jobs at 9 AM

The order of execution will be:

- Job 1 starts at 9 AM
- Job 1 ends at 9.05 AM
- Job 2 starts at 9.05 AM
- Job 2 ends at 9.10 AM and Job 3 starts, etc.

**Example 3**

You schedule:

- one serialized Job (A) at 9 AM
- two parallel Jobs (B and C) at 9:05 AM
- two serialized Jobs (D and E) at 9.10 AM
Chapter 4: Jobs

The order of execution will be:

- Job A starts at 9 AM
- Job A ends at 9.30 AM
- Job B and C start at 9.30 AM in parallel
- Job B and C end at 9:45 AM
- Job D (or E) starts at 9.45 AM in serial

Example 4

You schedule:

- five parallel Jobs at 9 AM

The order of execution will be:

- All Jobs start at 9 AM in parallel

Example 5:

Currently a parallel Job (A) is running

You schedule:

- one serialized Job (B) immediately
- one parallel Job (C) immediately

The order of execution will be:

- Job A continues execution.
- When Job A ends, Job B starts.
- When Job B ends, Job C starts.

More scenarios that could give conflicts when run in parallel:

Example 1

When scheduling the Task **Perform File Operations** with the option **Move Files/Folders** in parallel with another Task that uses these files and folders on the same server.

Example 2

When scheduling multiple **Download Resource** Tasks that connect to the same fileshare.

Example 3

When scheduling the Task **Perform File Operations** with the option **Edit/Create INI file**, in parallel with the Task to download a Resource with the option **Parse environment variables, parameters and functions** selected.

- The following Tasks are excluded, by default, from parallel processing:
  - Task **Install Windows Package**
  - Task **Reboot Computer**
  - Task **Shutdown Computer**

Note

- The legacy Agent and Agents for Unix/Linux and Mac OS X do not support parallel processing.
4.2 Activity

The Activity node gives an overview of all active Jobs that are being executed.

4.2.1 Where to find Activity

<table>
<thead>
<tr>
<th>What</th>
<th>Where</th>
</tr>
</thead>
<tbody>
<tr>
<td>Viewing information about an active Job</td>
<td>• At Jobs &gt; Activity, select an active Job and click View in the command bar.</td>
</tr>
<tr>
<td>Aborting a Job</td>
<td>• At Jobs &gt; Activity, select an active Job and click Abort in the command bar.</td>
</tr>
</tbody>
</table>

4.2.2 Forcing an abort of a Job that is already in an aborting state

In certain situations, for example when you schedule a Job on a Team that contains offline Agents, the Job can remain in an aborting state when it is aborted manually.

To allow you to tackle these situations: use the abort forcing functionality. So when you abort a Job that is already in the aborting state, it is aborted immediately, without waiting for a response from the Agent that is executing the Job.

This functionality is available by right-clicking a Job that is in an aborting state and clicking Abort.
4.3 Job History

The Job History gives an overview of all Jobs that were executed in your RES ONE Automation environment. You can view when a Job was started, how long it ran, what it was about, which Agent(s) performed it, which status it received, who triggered the Job, and whether it was part of a Run Book.

4.3.1 Where to find Job results

<table>
<thead>
<tr>
<th>What</th>
<th>Where</th>
</tr>
</thead>
<tbody>
<tr>
<td>Viewing all Job results</td>
<td>•  Jobs &gt; Job History.</td>
</tr>
<tr>
<td>Viewing the results of a specific Job</td>
<td>•  At Jobs &gt; Job History, select the relevant Job and click View in the command bar.</td>
</tr>
<tr>
<td>Exporting Job results</td>
<td>•  At Jobs &gt; Job History, select the relevant Job(s), right-click the selection and click Export Job Results.</td>
</tr>
<tr>
<td></td>
<td>•  At Library &gt; Modules, create a Module with a Task Export Job Results and schedule the Job.</td>
</tr>
<tr>
<td>Rescheduling a Job</td>
<td>•  At Jobs &gt; Job History, right-click the relevant Job and click Reschedule Job.</td>
</tr>
<tr>
<td>Deleting Job results</td>
<td>•  At Jobs &gt; Job History, select the relevant Job(s) and click Delete in the command bar.</td>
</tr>
</tbody>
</table>

4.3.2 Viewing Job results

If a Job has finished, you can see its results at the Job History node. This node shows basic information about all Jobs that have been executed in your RES ONE Automation site. This basic information includes the description of the Job, its duration and, very important: its status. The Job status clearly shows whether a Job has finished correctly.

A finished Job can have any of the following statuses:

<table>
<thead>
<tr>
<th>Status</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aborted</td>
<td>The Job was aborted by an administrator during its execution.</td>
</tr>
<tr>
<td>Canceled</td>
<td>The Job was canceled before it was executed.</td>
</tr>
<tr>
<td>Completed</td>
<td>The Job has completed without any errors.</td>
</tr>
<tr>
<td>Completed with errors</td>
<td>The Job has completed, but not all Tasks were performed correctly. On the Task’s Properties tab, Stop Job on error was not enabled.</td>
</tr>
<tr>
<td>Failed</td>
<td>The Job did not finish correctly. Stop Job on error was enabled and an error occurred.</td>
</tr>
</tbody>
</table>

Configuration

• You can search for a specific Job by pressing CTRL+F. Press F3 to go to the next occurrence of the search phrase.
• The column Who shows which Agents or Teams executed a Job. If this was a Run Book Job, this column shows See details. The details show which Run Book Job was executed by which Agent.
• If the Job was triggered by a Team, this usually means that the Job was performed because a Team member was added to or removed from the Team.
Although the basic Job results give you a basic overview of the completed Jobs, the provided information is limited. It is also not possible to compare the various Tasks and/or Agents. It is therefore often necessary to view the detailed results of a Job. To view the detailed Job results, select the Job and click View.

**Notes**

- The Job history is not affected when you change or delete Modules, Projects or Run Books; only when you change or delete Resources.
- Because of database changes, the Job History will become unavailable if you downgrade to a version older than RES ONE Automation 2015. Please consider exporting the Job History prior to downgrading. Should you decide to upgrade once again after downgrading, the entire Job History will be available again.
- Enabling or disabling the Auto-Refresh option to automatically update items affects ALL nodes in which this functionality is available (all Jobs and Topology nodes).

**Detailed Job results**

The Job window, which opens when viewing the detailed results of a Job, shows several tabs that allow you to view the details of a specific Job:

**Properties**

This tab shows a summary of the Job with information about the status of the Job, when the Job started and when it stopped, how long the Job ran, what was executed and by whom, etc.

- If a Job was triggered by a Team or a Run Book, or if it is part of a Recurring Job, the Show Job Invoker button is shown. Click this button to jump to the relevant Team, Run Book or Recurring Job. This button Show Job Invoker is only shown if your administrative role allows you to view the Scheduling node or the Teams node. Please note, that for Jobs that were Scheduled using Cron, the button is only available if the cron schedule has not yet expired.
- If your administrative role allows you to view the Teams node, but not the Team that triggered the Job, the button Show Job Invoker will be shown, but no Job invoker details will be shown when you click this button.

**Job Parameters**

This tab is only shown if parameters were used in the Job. It shows the parameters and the values they had during execution of the Job.

**Tasks**

This tab shows all Modules and Tasks that were part of the Job, including whether they completed successfully, only partially, or if they failed. To see the details of a specific Module or Task, select it and click Details.

- **Modules**: if you view the details of a Module, the Job Properties window shows which Agent(s) executed the Module. To view the details for a specific Agent, select it and click Details. This allows you to view the details of any parameters or conditions in the Module that were applied to the Agent.
- **Tasks**: if you view the details of a regular Task, the Task Job window shows a brief overview of all Agents that executed the Task. To see the details of the Task for a specific Agent, select it and click Details.
- Clear the option Show Agents that are marked obsolete for this Job by Snapshot Intelligence to hide Agents for which this Job has become obsolete because the Agent has reverted to a snapshot. See Snapshot Intelligence (on page 197).
Chapter 4: Jobs

- **Query Tasks**: if you view the details of a Query Task, the Task Job window shows the Query results per Agent.
  - Click **Swap rows and columns** to change the view.
  - Click the **Similarities** button to hide or mark similarities. For certain Query Tasks, it is possible to filter out or mark similarities between two Agents. This allows you to compare the results of different Agents. For example, if you execute a Task **Query Installed programs** on multiple Agents, you can filter out or mark all similar programs. Although only basic information about the Tasks is shown, this approach allows you to compare the results of the Agents. In this way it is very easy to see differences between Agents. Marked similarities will also be shown in an Instant Report of this view.
  - Clear the option **Show Agents that are marked obsolete for this Job by Snapshot Intelligence** to hide Agents for which this Job has become obsolete because the Agent has reverted to a snapshot.
  - To see the details of the Query Task for a specific Agent, select it and click **Details**.
  - You can also schedule a Job based on the results of the Query Task. For example, if the Job results show that a number of Agents have an outdated version of an application, select these Agents, right-click them and select **Schedule Job**. The window **Add/Edit Job** opens with the selected Agents already entered in the **Who** field. This allows you to take action on the Query Task results very quickly and easily.
  - You can also save the results as a TXT file or a CSV file. See **Saving the Job results of a Query Task as a TXT file or a CSV file** (on page 74).

**Agents**

This tab shows all Agents that executed the Job, including whether they completed the Job successfully, only partially, or if they failed.

- Clear the option **Show Agents that are marked obsolete for this Job by Snapshot Intelligence** to hide Agents for which this Job has become obsolete because the Agent has reverted to a snapshot.
- To see the details of a specific Agent, select it and click **Details**.
  - **Modules**: if you view the details of a Module, the **Job Properties** window shows the details of any parameters or conditions in the Module that were applied to the Agent.
  - **Tasks**: if you view the details of a regular Task, the **Task** window shows the settings of the executed Task.
  - **Query Tasks**: if you view the details of a Query Task, the **Task** window opens at the Result tab, showing the Query results for the Agent.
    - Click **Swap rows and columns** to change the view.
    - Certain Query Tasks can contain evaluators. For this type of Queries, the option **Only show data returned by evaluator** allows you to filter the Query results. See **Evaluators** (on page 237).
    - You can also save the results as a TXT file or a CSV file.

**Job History**

This tab only shows information if the scheduled Job is still active as a recurring Job.

**Audit Trail**

This tab gives a short overview about when the Job was created, what was executed and by whom.
4.3.3 Reapplying the Job History of Agents and Teams

Traditionally, it has been time-consuming and difficult to ensure uniform maintenance in your IT environment. The setup that in theory should have been ideal for your organization always needs subsequent fine-tuning before the optimum working IT environment has been obtained where each user has the right access and resources at the right time. If you subsequently need to add new computers to this setup, recreating that fine-tuning process after the initial setup can be challenging. By reapplying the Job history of an Agent or Team, this challenge becomes simple and less time-consuming.

RES ONE Automation stores information about all the Jobs executed on an Agent or Team, including how those Jobs were configured at the moment of scheduling. This means that, even when several persons are responsible for maintaining your RES ONE Automation environment, it is always clear what actions were executed by which Agents. By reapplying the Job of an Agent, you do not simply copy the applications and configuration settings from one computer to another, because that will never result in an exact replica computer. The applications and configuration settings on the original Agent were obtained through a specific sequence of actions, and by reapplying this Job history, you can reproduce this sequence of actions on other Agents, so that you obtain the exact same results. This is useful, for example, if a new Agent should have an identical same setup as an existing Agent, with the same applications installed, the same system updates, configuration settings, etc.

When you reapply the Job history of an Agent or Team, you can select which Tasks in the Job history should be reapplied. This allows you to filter out erroneous or surplus Tasks, failed Tasks, reboots and shutdowns, etc. Only those actions that you select will be reapplied, which speeds up the reapplication process and leads to optimum results.

When you have created identical copies of an Agent by reapplying its job history, you want to make sure that the copies stay identical to the original. You can do so by adding the new Agent to the same Team as the original Agent. By performing maintenance on Teams instead of individual Agents, you can attain uniform maintenance in your IT environment. This not only makes maintenance simple, reliable and time-saving, but also ensures that you can reapply the Job history continually on Agents in your IT environment.

Configuration

To reapply the Job history of an Agent or Team:

1. Open the properties of the relevant Agent or Team (at Topology > Agents or Topology > Teams). The Agent Properties window or the New/Edit Team window opens.
2. Click the Job History tab and click Reapply Job History. The Reapply Tasks window opens.
3. Use the various options to select the Tasks that should be reapplied. For example, you can exclude Tasks individually or based on Job status. It is also possible to exclude all Reboot Computer and Shutdown Computer Tasks.
   - Optionally, you can create a new Module with the selected Tasks.
4. After selecting the Tasks, click OK. The New/Edit Job window opens.
5. Click the Who field to select the appropriate Agents or Team and execute the Job.

Notes

- When a Job history is reapplied, any parameters are used with the same value as in the original Jobs: new input is not requested.
- Conditions are re-evaluated when a Job history is reapplied.
- Recurring Jobs (including scheduled using Cron) cannot be reapplied.
- Changing or deleting a Module or Project does NOT affect the Job history (changing or deleting Resources DOES affect the Job history).
4.3.4 Rescheduling a Job

Rescheduling a Job works in a similar way as reapplying the Job history of an Agent, but for an individual Job. By default, a rescheduled Job is executed on the same Agents or Teams, but different Agents or Teams can be selected if necessary.

Rescheduling a Job is useful for Jobs that should be repeated with some regularity, such as a query of the disk space available on Agents. Because you can also target other Agents or Teams than the original ones, it is also useful, for example, for trying out a Job on one Agent before executing it on a set of Agents and Teams.

RES ONE Automation stores the Job in the Job results exactly as it was performed at execution. This information does not change if the configuration of the used Modules, Projects or Run Books is changed afterwards. Tasks containing parameters and/or functions will be stored by RES ONE Automation with the actual values and data that were applied at the time of Job execution. Therefore, when you reschedule a Job, the Job will be repeated in the form it had when it was originally scheduled, with the Tasks configured as they were then. Any later changes to Modules/Projects will not be reflected in a rescheduled Job.

Configuration

To reschedule a Job:

1. At Jobs > Job History, right-click the relevant Job(s) and choose Reschedule Job. The Reschedule Tasks window opens.
2. Use the various options to select the Tasks that should be reapplied. For example, you can exclude Tasks individually or based on Job status. It is also possible to exclude all Reboot Computer and Shutdown Computer Tasks.
   - Optionally, you can create a new Module with the selected Tasks.
3. After selecting the Tasks, click OK. The New/Edit Job window opens.
   - If necessary, click the Who field to select other Agents or Team and execute the Job.
   - If you made any changes to the Modules or Projects after the original Job was executed, and you want the rescheduled Job to reflect the most recent changes, click Reload Resulting Tasks and Reload Resulting Agents on respectively the Resulting Tasks and Resulting Agents tabs of the Job. This updates the Job to reflect these changes.
4. Click OK to reschedule the Job.

Notes

- A rescheduled Job that uses a Resource will use the current version of that Resource. If the Resource was edited after the original execution of the Job, the rescheduled Job will use the edited version.
- Depending on the input settings, new input will be requested for any parameters used in a rescheduled Job.
- Conditions are re-evaluated when a Job is rescheduled.
- Besides rescheduling Jobs, you can also reapply the entire Job history of an Agent or Team to another Agent or Team. See Reapplying the Job History of Agents and Teams (on page 68).
### 4.3.5 Exporting Job results to XML

Besides by executing a Task **Export RES ONE Automation Results** (see page 136), you can also manually export the results of a Job from the Console as an XML file. Exporting the Job results to an XML file allows you or others to view Job results without using the Console. This is useful, for example if someone wants to review the results of a Job on a computer on which a Console has not been installed. You can also use it save specific Job results as an XML file before you delete these Job results from the Datastore.

There are various ways in which Job results can be exported to XML:

- **Exporting Job results immediately** (on page 71)
- **Exporting the Job results of a specific Job using a command line** (on page 71)
- **Exporting the Job results of an active Job using a generic command line** (on page 72)
- **Exporting the Job results of a previous Job using a generic command line** (on page 73)

#### Exporting Job results immediately

To export the Job results immediately:

1. At **Jobs > Job History**, select the relevant Job(s), right-click the selection and click **Export Job Results**. The Export Job Results window opens.

2. To change the save location of the XML file, click ![folder](image) in the **Save location** field. You can replace folder names with their associated environment string by right-clicking the field.

3. Select **Export overview results** to include a general overview of the Job results for all Tasks that were included in the Job.

4. Select **Export detailed results** to include a detailed overview of the Job results for all Tasks that were included in the Job. If you also select **Export overview results**, RES ONE Automation will create one XML file that first gives a general overview of the Job results and then a detailed overview.

5. Select **Export results per Agent** to create a separate XML file for each Agent that executed the Job, including the name of the Agent. This is useful when you want to keep a separated administration for each Agent.

6. Click **Export**.

#### Notes
- When you select **Export results per Agent**, any existing XML files with the same name will be overwritten without warning.
- When exporting the results of a Job, the Variable are included in the XML files for the Agents and/or Teams.

#### Exporting the Job results of a specific Job using a command line

To export the Job results of a specific Job using a command line:

1. At **Jobs > Job History**, select the relevant Job(s), right-click the selection and click **Export Job Results**. The Export Job Results window opens.

2. To change the save location of the XML file, click ![folder](image) in the **Save location** field. You can replace folder names with their associated environment string by right-clicking the field.

3. Select **Export overview results** to include a general overview of the Job results for all Tasks that were included in the Job.

4. Select **Export detailed results** to include a detailed overview of the Job results for all Tasks that were included in the Job. If you also select **Export overview results**, RES ONE Automation
will create one XML file that first gives a general overview of the Job results and then a detailed overview.

5. The Command line box shows the command line that will be used to generate the export file. Click Copy Command Line and close the Export Job Results window.

6. Configure a Task Execute Command, paste the command line on its Settings tab, and schedule a Job with this Task. Alternatively, you can run the command line using a Microsoft Windows command prompt.

Notes

- When exporting the results of a Job, the Variable are included in the XML files for the Agents and/or Teams.
- It is not possible to export the results of multiple Jobs using the Command line box.
- You can only execute a command line that exports Job results on Agents on which a Console has been installed.

Exporting the Job results of an active Job using a generic command line

To export the Job results of an active Job using a generic command line:

1. At Jobs > Job History, select the relevant Job(s), right-click the selection and click Export Job Results. The Export Job Results window opens.

2. To change the save location of the XML file, click in the Save location field. You can replace folder names with their associated environment string by right-clicking the field.

3. Select Export overview results to include a general overview of the Job results for all Tasks that were included in the Job.

4. Select Export detailed results to include a detailed overview of the Job results for all Tasks that were included in the Job. If you also select Export overview results, RES ONE Automation will create one XML file that first gives a general overview of the Job results and then a detailed overview.

5. Select Replace GUID with function and Add command line parameter to wait for Job to finish.
   - With Replace GUID with function, you can create a generic command line that you can use in, for example, a Task Execute Command.
   - With Add command line parameter to wait for Job to finish, you can create a generic command line that waits with exporting Job results until all running Tasks have been executed. This option only works when combined with the option Replace GUID with function.

6. The Command line box shows the command line that will be used to generate the export file. Click Copy Command Line and close the Export Job Results window.

7. Use this Task in a Job that contains the Module(s) or Project(s) of which you want to export the Job results. Alternatively, if a previous Job is still running, you can export the results of this Job by scheduling a new Job that only contains a Task Execute Command.

Notes

- When exporting the results of a Job, the Variable are included in the XML files for the Agents and/or Teams.
- It is not possible to export the results of multiple Jobs using the Command line box.
- You can only execute a command line that exports Job results on Agents on which a Console has been installed.
- The function @[MASTERJOBGUID] cannot be used for Run Books.
- When used in combination with the Task Execute Command and the timeout option Terminate process, the Task will be terminated but not the sub processes. This generates Job results exported to an XML file. The timeout option Terminate process tree will also stop the sub processes and does not generate an XML file.
Chapter 4: Jobs

Exporting the Job results of a previous Job using a generic command line

Because RES ONE Automation remembers the GUID of each last opened or scheduled Job, you can schedule a Job that exports the Job results of a previous Job:

1. At Jobs > Job History, select the relevant Job(s), right-click the selection and click Export Job Results. The Export Job Results window opens.

2. To change the save location of the XML file, click in the Save location field. You can replace folder names with their associated environment string by right-clicking the field.

3. Select Export overview results to include a general overview of the Job results for all Tasks that were included in the Job.

4. Select Export detailed results to include a detailed overview of the Job results for all Tasks that were included in the Job. If you also select Export overview results, RES ONE Automation will create one XML file that first gives a general overview of the Job results and then a detailed overview.

5. Select Replace GUID with function. This allows you to create a generic command line that you can use in, for example, a Task Execute Command.

6. The Command line box shows the command line that will be used to generate the export file. Click Copy Command Line and close the Export Job Results window.

7. Configure a Task Execute Command and paste the command line on its Settings tab.


9. Clear the Input option Show previous value for this parameter. The parameter $[MASTERJOBGUID] is a special parameter: it works similar to the function @[MASTERJOBGUID], and is used by RES ONE Automation to capture the GUID of the previous Job.

10. On the Settings tab, replace the function @[MASTERJOBGUID] in the command line with a parameter $[MASTERJOBGUID].

11. Schedule a Job with the Task. During execution of the Job, the value of the parameter $[MASTERJOBGUID] in the command line will be replaced by the GUID of the previous Job.

Notes
- When exporting the results of a Job, the Variable are included in the XML files for the Agents and/or Teams.
- It is not possible to export the results of multiple Jobs using the Command line box.
- You can only execute a command line that exports Job results on Agents on which a Console has been installed.
- The function @[MASTERJOBGUID] cannot yet be used for Run Books.
4.3.6 Saving the Job results of a Query Task as a TXT file or a CSV file

If the Job that was executed contained a Query Task, you can save the results of this Task as a TXT file or a CSV file. This allows you to view the Job results outside RES ONE Automation, for example in a spreadsheet.

To save the general results of a Query for all Agents:

1. Open the relevant Job.
2. Click the Tasks tab, select the Query and click Details.
3. In the Task Job window, click \( \text{save} \).
4. In the window that opens, type a name for the file and select whether to save it as a TXT file or as a CSV file.

To save the detailed results of a Query for a specific Agent:

1. Open the relevant Job.
2. Click the Agents tab, select an Agent and click Details.
3. In the Agent Job window, select the Query and click Details.
4. In the Task window, click \( \text{save} \).
5. In the window that opens, type a name for the file and select whether to save it as a TXT file or as a CSV file.

Note

The regional settings of your user account determine the list separator used in the CSV file. This setting is stored in the registry value of the registry key 'HKEY_CURRENT_USER\Control Panel\International\sList'.
4.4 Scheduling in the Management Portal

When you have installed the RES ONE Automation Management Portal (alongside the Console) you can also schedule Jobs and view Job results from the Management Portal. For more information, see also the Help in the Management Portal. As a summary, this is how it looks like in the Management Portal:

Scheduling

In the Management Portal, on the Jobs page, you can manage the following:

On the Scheduling tab, you can schedule Jobs on Agents and Teams and view information about:

- **Recent Activity**: last 5 Jobs that were performed
- **Current Activity**: all active or aborting Jobs (currently being performed)
- **Scheduled**: all Jobs that are scheduled and/or on hold

On the Job History tab, you can view the results of a Job (or directly at Recent Activity).

A scheduled or active Job can have any of the following statuses:
Chapter 4: Jobs

<table>
<thead>
<tr>
<th>Status</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active</td>
<td>The Job is active (being performed).</td>
</tr>
<tr>
<td>Scheduled</td>
<td>The Job is scheduled.</td>
</tr>
<tr>
<td>Requested</td>
<td>The Job request is sent to the Dispatcher to be scheduled.</td>
</tr>
<tr>
<td>Scheduled Recurring</td>
<td>The Job is scheduled and recurring.</td>
</tr>
<tr>
<td>Aborting</td>
<td>The Job is being aborted by an administrator during its execution.</td>
</tr>
<tr>
<td>On Hold</td>
<td>The scheduled Job is put on hold.</td>
</tr>
</tbody>
</table>

The following actions are available from the † context menu for scheduled or active Jobs:

- Audit Trail
- Delete
- Put On Hold or Resume (scheduled Jobs)
- Abort (active or aborting Jobs)

To view or change the configuration details of a Job, click the relevant Job.

To schedule a new Job, click New. The same configuration fields apply.
Chapter 4: Jobs

Job History

In the Management Portal, on the Jobs page, on the Job History tab, you can view when Jobs were started, how long they ran, what they were about, which Agent(s) performed them, which status they received, who triggered the Jobs, and whether they were part of a Run Book.

A finished Job can have any of the following statuses:

<table>
<thead>
<tr>
<th>Status</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>✗ Aborted</td>
<td>The Job was aborted by an administrator during its execution.</td>
</tr>
<tr>
<td>✗ Canceled</td>
<td>The Job was canceled before it was executed.</td>
</tr>
<tr>
<td>✔ Completed</td>
<td>The Job has completed without any errors.</td>
</tr>
<tr>
<td>🔴 Completed With Errors</td>
<td>The Job has completed, but not all Tasks were performed correctly. On the Task's Properties tab, Continue Job on error was enabled.</td>
</tr>
<tr>
<td>✗ Failed</td>
<td>The Job did not finish correctly. Stop Job on error was enabled and an error occurred.</td>
</tr>
</tbody>
</table>

The following actions are available from the ‡ context menu for finished Jobs:

- Audit Trail
- Delete
- Reschedule
- Export (results in XML)

To view the Job history details, click the relevant Job.

For Query Tasks, you can also export the detailed results to a CSV or TXT file.
Chapter 5: Library

The Library of the Console consists of:

- **Modules** (on page 78)
- **Projects** (on page 154)
- **Run Books** (on page 159)
- **Resources** (on page 169)
- **Variables** (on page 176)

You can use the following elements in Modules, Projects and Run Books:

- **Parameters** (on page 257)
- **Functions** (on page 246)
- **Conditions** (on page 233)
- **Evaluators** (on page 237)

### 5.1 Modules

If you want to perform (a series of) Task(s), the first step is to create a **Module**. A Module is a container for one or more Tasks that you can execute on Agents. Modules hold information about the way in which these Tasks should be executed, such as the order of execution, parameters, conditions, and/or evaluators. A Module can contain multiple Tasks. The ability to use a series of Tasks is very useful if one Task is directly related to the other one. To execute the Tasks in a Module, you need to schedule a Job with the Module of which they are part of.

For example: A Module of an installation of a Microsoft Installer Package (an MSI file), followed by a reboot of the target machine. This Module would consist of two Tasks:

- a Task **Invoke Windows Installer** with all related settings (which MSI file, etc.).

Followed by:

- a Task **Shutdown Computer** with the related settings (reboot).
5.1.1 Managing Modules

At Library > Modules:

- On the Folders tab of the Modules node, you can create a folder structure (see page 80) for all your Modules and place the Modules that you have created in the designated folders. Grouping Modules in folders can be useful in large environments and multi-tenant RES ONE Automation sites to easily locate any Modules you have created and also to create folders for various needs.
  - You can add, edit or delete folders by using the three folder buttons on the lower right side of the screen.
  - If you delete a folder, its contents will be deleted as well.
- The Modules tab shows exactly the same list of Modules, but alphabetically listed and without the folder structure.
- To create a duplicate of a Module, select it and click Duplicate in the command bar. Creating duplicates of a Module is useful when you need to create a new Module that only slightly differentiates from an already existing Module. By changing the values and settings of a duplicated Module, you can customize it to your needs. This saves configuration time.
- To quickly create a new Module for a specific Task, click Quick Task in the command bar. This opens the Select Task window immediately; a new Module will be created automatically. The Select Task window allows you to select Tasks from various libraries. After configuring the selected Task, it will be added to the new Module. See Tasks (on page 85) for an overview of all libraries and the Tasks they contain.
- When saving changes to the Module, the Version Control window will be opened. This window makes it possible to configure the versioning (see page 272) of changes to the Module. The availability of this window depends on the global setting Use silent versioning (see page 48).
- After modifying a Module, a notification window will open if the Module is used in a scheduled Job. Provided you have sufficient permissions to the Scheduling node, you can use this window to open the scheduled Job and reload the relevant Tasks, if necessary.

When configuring a Module:

- Use the Properties tab to specify a name and an optional description for the Module.
- Use the Tasks tab to manage the Tasks in the Module.
- Use the Module Parameters tab to manage the parameters (on page 257) in all Tasks in the Module.
- Use the Usage tab to view in which Project(s), Run Book(s) and/or Team(s) the Module is used.
- Use the Job History tab to view details of Jobs in which the Module was used and to reschedule these Jobs if necessary.
- Use the Versioning tab to view a list of changes made to the Module. This includes all known versions of the Module.
- Use the Trusts tab to define trusted Agents and Teams for the Module. This tab is only shown if Trusts Security (see page 269) has been enabled.
- Use the Permissions tab to view the level of access that Console users have to the Module.
5.1.2 Module Folders

By grouping Modules in folders (At Library > Modules) you can:

- configure permissions on groups of Modules. See Administrative Roles (“Security” on page 213).
- configure trusted Agents and Teams for a Module folder. See Trusts (see page 269).

Configuration

- When configuring a Module folder, use the Trusts tab to define trusted Agents and Teams.
- Use the search bar to search for specific objects.
- You can use the following settings when configuring Trusts:
  - Do not trust: If this setting is selected, the selected item is not trusted. For example, if you are configuring Trusts on a specific Agent and specify that the Module Delete RES ONE Automation Results is not trusted by the Agent, the Module will fail when used in a Job that is executed by the Agent.
  - Trust: If this setting is selected, the selected item is trusted. This is the default setting. Depending on whether a full Trust exists with the item, it is possible to use the item in a Job. For example, if you are configuring Trusts on a specific Module and specify that Agent ONE is trusted by the Module, the Agent is allowed to use the Module in a Job.
  - Inherit trust: If no particular settings are selected (blank check box), the item will inherit the Trusts from the above lying item. Trusts that are inherited from an above lying item are grayed out and show the type and name of this item, except when this is the top-level folder: The Trusts of each top-level folder use the default setting (Trust). These Trusts are shown in black and it is not possible to change their settings.
  - All items that are shown on the various Trusts tabs are subject to the access permissions of the administrative role(s) of the Console user.

Select Show only items that trust this ... to filter on items that already trust the Agent, Team, Team folder, Module, Module folder, Resource or Resource folder that you are editing. When Trusts are enabled, Agents can only execute Jobs with Modules and Resources when a full Trust exists between them (i.e. the Module or Resource is trusted by the Agent AND the Agent is trusted by the Module or Resource). By using a filter, it becomes easier to create full Trusts.

- Filtering the Trusts tab may take some time, especially in large environments with many Agents, Modules and Resources. When selecting the option, a popup window is therefore shown, showing the progress. If necessary, you can cancel the filtering.
- If an item does not trust the item that you are editing, it will not be shown. Folders will be shown with an overlay icon.
- This option requires Microsoft .NET Framework 4.0 (Full version) or higher to be installed on the machine that runs the Console.

Note

Availability of the Trusts tab depends on the global setting Trusts Security. See Trusts (see page 269).
5.1.3 Module Properties tab

At Library > Modules, you can configure and manage Modules. Modules are containers for the Tasks that you can execute on Agents. Modules hold information about how these Tasks should be executed, such as Task settings, order of execution of Tasks, parameters and conditions. When configuring Modules, use the Properties tab to specify a name and an optional description for the Module.

**Configuration**

- The Module name is the name that is shown on the Folders/Modules tab of the Modules node. When creating a Module, it is therefore wise to choose a name that reflects its contents.
- The Description that you can optionally specify is only shown on the Modules tab. In the Description field, you can use hyperlinks. Console users can directly browse to these hyperlinks from the Console. The following links are supported:
  - http://
  - https://
  - file://
- If a Module is disabled, you can no longer directly schedule it in a Job. However, if a Module is disabled, it is still enabled in Projects and Run Books. It is, however, not possible to create new Projects and Run Books containing that Module.
- If you disable a Module in a Project, it is only disabled in that Project, but not globally. That means you can still schedule it.
- Use the option Leave used Resources in Agent cache after Module is finished only for Resources that are frequently used by the Agents and that rarely change. This allows Agents to execute Tasks faster, because they do not need to download the Resources from a Dispatcher first.

5.1.4 Tasks tab

At Library > Modules, you can configure and manage Modules. Modules are containers for the Tasks that you can execute on Agents. Modules hold information about how these Tasks should be executed, such as Task settings, order of execution of Tasks, parameters and conditions. When configuring Modules, use the Tasks tab to manage the Tasks in the Module.

Tasks are actions that you can perform on an Agent. With a Task you can, for example, install software on Agents, manage users and groups in Active Directory, query security risks on Agents or query mailboxes on Microsoft Exchange servers.

**Configuration**

- When adding a new Task to a Module, you can select Tasks from the various libraries in the Select Task window. See Tasks (on page 85) for an overview of all libraries and the Tasks they contain.
  - You can use the Instant Search field when selecting Tasks. To find a specific Task, simply start typing part of its name. As you type, the list will be filtered to show only items that contain the search term. Alternatively, you can just browse the tree of available Tasks.
  - When selecting a Task for a Module, you can open all libraries in the Select Task window at once: right-click and select Expand all. To close all open libraries, right-click and select Collapse all.
  - You can also select and open items in the Select Task window using the keyboard. Use TAB and ENTER to go to libraries and topics. To select a Task in a topic, select the relevant topic and type 1 to open the first Task, type 2 for the second Task, etc.
  - To create a duplicate of a Task, select it and click Duplicate. Creating a duplicate of a Task is useful when you need to configure a new Task that only slightly differentiates from an already
existing Task in the Module. By changing the values and settings of a duplicated Task, you can customize it to your needs. This saves configuration time.

- You can add multiple Tasks to a Module.
- To create sets of Tasks, right-click the Tasks tab and click Add Separator. Separators are a visual aid to enhance readability in Modules that contain many Tasks; they do not affect the execution order of Tasks, but make it possible to configure visible sections of Tasks that, for example, are related to each other. In addition, it is possible to add a descriptive text to a set of Tasks.
  - The name of a separator can contain a maximum of 255 characters.
  - The description of a separator can contain a maximum of 4000 characters. The use of hyperlinks is supported.
- To change the order of execution of Tasks in the Module, select a Task and use the arrow buttons. Likewise, use the arrow buttons to move any separators up or down the list of Tasks.
- To set conditions on a Task, select the Task and click Conditions. Task conditions determine whether a Task in a Module should be executed, skipped or failed. See Conditions (on page 233).
- You can copy and paste Tasks from one Module to another. If a Task contains parameters, only their placeholders will be copied, as text: the corresponding Module parameters will not be copied, but have to be created manually.
- Alternatively, if you want to create a new Module quickly for a specific Task, right-click the Modules node and click Quick Task. This opens the Select Task window immediately; a new Module will be created automatically. After configuring the selected Task, it will be added to this Module. This functionality is also available from the Action menu.

5.1.5 Module Parameters tab

At Library > Modules, you can configure and manage Modules. Modules are containers for the Tasks that you can execute on Agents. Modules hold information about how these Tasks should be executed, such as Task settings, order of execution of Tasks, parameters and conditions. When configuring Modules, use the Module Parameters tab to manage the parameters in all Tasks in the Module. See parameters (on page 257).

The Module Parameters tab gives an overview of all parameters of all Tasks in the Module. With Module parameters, you can parameterize various settings in a Task. Because each Task is part of a Module, you can use the same Module parameter in all Tasks of the Module: This is useful if similar settings in multiple Tasks share the same value (for example, a specific user name or password).

Configuration

- If you delete Module parameters that are used in one or more Tasks and/or other parameters, you have the option to replace all references to the parameter with its current value.
- Use the arrow buttons to change the order in which Module parameters are presented at input moment.
- Click Cleanup to delete Module parameters that are not used by any Task in the Module.
5.1.6 Module Usage tab

At Library > Modules, you can configure and manage Modules. Modules are containers for the Tasks that you can execute on Agents. Modules hold information about how these Tasks should be executed, such as Task settings, order of execution of Tasks, parameters and conditions. When configuring Modules, use the Usage tab to view in which Project(s), Run Book(s) and/or Team(s) the Module is used.

Besides the Project(s) and/or Run Books in which the Module is used, the Usage tab also shows in which Team(s) it is used if Teams use the Module for Automatic Job Scheduling. See Teams (on page 205).

Configuration
- To jump to a Project, Run Book or Team, select it and click Open.

5.1.7 Module Job History tab

At Library > Modules, you can configure and manage Modules. Modules are containers for the Tasks that you can execute on Agents. Modules hold information about how these Tasks should be executed, such as Task settings, order of execution of Tasks, parameters and conditions. When configuring Modules, use the Job History tab to view details of Jobs in which the Module was used and to reschedule these Jobs if necessary.

Configuration
- To reschedule (part of) a Job, right-click the relevant Job and click Reschedule Job.

5.1.8 Module Versioning tab

At Library > Modules, you can configure and manage Modules. Modules are containers for the Tasks that you can execute on Agents. Modules hold information about how these Tasks should be executed, such as Task settings, order of execution of Tasks, parameters and conditions. When configuring Modules, use the Versioning tab to view a list of changes made to the Module, including all known versions of the Module.

Configuration
- The Current version field specifies the latest saved version number of the Module. It is possible to change this version number manually by clicking ..., regardless of the global setting Use automatic versioning (see page 48).
- The Version column specifies all versions of the Module.
- The Comment column specifies the comment that Console users made about individual versions.
- Use the drop-down field to customize the number of Audit Trail results that are shown per page.
- Use the navigation buttons to browse through the various Audit Trail pages.
5.1.9 **Module Trusts**

Use the **Trusts** tab to define trusted Agents and Teams for the Module. A Module can have explicit Trusts or inherit its Trusts from its Module folder. See **Trusts** (see page 269).

- Use the search bar to search for specific objects.
- You can use the following settings when configuring Trusts:
  - **Do not trust**: If this setting is selected, the selected item is not trusted. For example, if you are configuring Trusts on a specific Agent and specify that the Module **Delete RES ONE Automation Results** is not trusted by the Agent, the Module will fail when used in a Job that is executed by the Agent.
  - **Trust**: If this setting is selected, the selected item is trusted. This is the default setting. Depending on whether a full Trust exists with the item, it is possible to use the item in a Job. For example, if you are configuring Trusts on a specific Module and specify that Agent **ONE** is trusted by the Module, the Agent is allowed to use the Module in a Job.
  - **Inherit trust**: If no particular settings are selected (blank check box), the item will inherit the Trusts from the above lying item. Trusts that are inherited from an above lying item are grayed out and show the type and name of this item, except when this is the top-level folder: The Trusts of each top-level folder use the default setting (**Trust**). These Trusts are shown in black and it is not possible to change their settings.
  - All items that are shown on the various **Trusts** tabs are subject to the access permissions of the administrative role(s) of the Console user.
  - Select **Show only items that trust this ...** to filter on items that already trust the Agent, Team, Team folder, Module, Module folder, Resource or Resource folder that you are editing. When Trusts are enabled, Agents can only execute Jobs with Modules and Resources when a full Trust exists between them (i.e. the Module or Resource is trusted by the Agent **AND** the Agent is trusted by the Module or Resource). By using a filter, it becomes easier to create full Trusts.
  - Filtering the **Trusts** tab may take some time, especially in large environments with many Agents, Modules and Resources. When selecting the option, a popup window is therefore shown, showing the progress. If necessary, you can cancel the filtering.
  - If an item does not trust the item that you are editing, it will not be shown. Folders will be shown with an overlay icon.
  - This option requires Microsoft .NET Framework 4.0 (Full version) or higher to be installed on the machine that runs the Console.
  - When you create a duplicate of a Module for which trusts have been defined, you will be prompted whether the trusts should be included in the duplicate.

---

**Note**

*Availability of the **Trusts** tab depends on the global setting **Trusts Security**. See **Trusts** (see page 269).*
5.1.10  **Module Permissions tab**

At Library > Modules, you can set up and manage Modules. Modules are containers for the Tasks that you can execute on Agents. Modules hold information about how these Tasks should be executed, such as Task settings, order of execution of Tasks, parameters and conditions. When configuring Modules, use the Permissions tab to view the level of access that Console users have to the Module.

**Configuration**

- Use the option Do not allow this Module to be scheduled programmatically to allow the Module from being invoked via the RES ONE Automation Console only. When selected, it will no longer be possible to invoke the Module:
  - During unattended installations of an Agent (by applying properties to its MSI file).
  - Via RES ONE Workspace.

5.1.11  **Tasks**

The core of any RES ONE Automation environment is the ability to perform Tasks. Tasks are actions that can be performed by an Agent. With a Task you can, for example, install software on Agents or manage users and groups in Active Directory, but also query security risks on Agents or query mailboxes on Microsoft Exchange servers. Some Tasks are part of an Add-on and can be installed via RES ONE Connectors. In general, all Tasks are performed in the same way. Tasks are configured as part of a Module.

The various query Tasks make it possible to acquire several types of information about Agent machines. The query Task can be combined with another Task that, for example, can be based on the results of that query Task.

You can use the Instant Search field for selecting Tasks. To find a specific Task, simply start typing part of its name. As you type, the list will be filtered to show only items that contain the search term. Alternatively, you can just browse the tree of available Tasks. Via Setup > Task Visibility, you are able to show or hide certain Tasks from the Select Task window.

Tasks are grouped by functionality in libraries:

- The library Recently Used contains the 5 most recently used Tasks.
- The library Configuration contains Tasks that relate to the configuration of a computer, such as applying registry settings, managing service parameters, and performing file operations.
- The library Provisioning contains Tasks that relate to routine administrative management, such as Active Directory Tasks, Exchange Mailbox Tasks, and the provisioning of applications and services.
- The library PC Lifecycle Management contains Tasks that relate to managing PC lifecycles, such as applying actions via IBM Endpoint Manager and distributing LANDesk software.
- The library Virtualization contains Tasks that relate to management of virtual applications and machines, such as Microsoft Hyper-V, Microsoft App-V, and VMware machines.
- The library Security contains Tasks that relate to security, such as querying security risks, installing security updates, and managing local groups and users.
- The library System State contains Tasks that relate to the system state of a computer, such as defragmenting disks, and querying events logs and installed programs.
- The library Advanced contains advanced Tasks, such as sending SMTP traps and executing Microsoft Windows PowerShell scripts.
- The library RES ONE Automation contains Tasks that relate to maintaining the RES ONE Automation environment, such as deploying components and deleting Job results.
The library Unix/Linux contains Tasks that can be executed by Agents running on Linux and Unix.

The library Apple Mac OS contains Tasks that can be executed by Agents running on Mac OS X.

The library Software as a Service contains Tasks that relate to provisioning of software services, such as cloud solutions.

The library Infrastructure as a Service contains Tasks that relate to infrastructure services, such as creating Virtual Servers via IBM SoftLayer.

The library Service Desk Integration contains Tasks that relate to integrating service desk services, such as BMC Remedy, IBM Maximo and ServiceNow.

The library Mobile Device Management contains Tasks that relate to managing mobile devices, such as adding devices via AirWatch, Fiberlink and MobileIron.

When you have selected and configured a Task, it is shown on the Tasks tab of the Module. If necessary, you can now add another Task to the Module. All added Tasks will be shown on the Tasks tab. You can change the order of execution with the arrow buttons.

When selecting a Task for a Module, you can open all libraries in the Select Task window at once: right-click and select Expand all. To close all open libraries, right-click and select Collapse all.

You can also select and open items in the Select Task window using the keyboard. Use TAB and ENTER to go to libraries and topics. To select a Task in a topic, select the relevant topic and type 1 to open the first Task, type 2 for the second Task, etc.

You can copy and paste Tasks in a Module or from one Module to another. If a Task contains parameters, only their placeholders will be copied, as text; the corresponding Module parameters will not be copied, but have to be created manually.

You can add multiple Tasks to a Module.

Tasks specific for Add-ons

The Tasks specific for Add-ons are available for several library categories, such as Mobile Device Management and Virtualization. You can install the Add-ons manually via RES ONE Connectors. With these RES ONE Connectors you can increase your automation capabilities based on different technologies or vendors.

Each Add-on contains one or more Tasks grouped per object. For example, the RES ONE Connector (formerly known as Automation Pack) for Mobile Device Management contains the Add-on for AirWatch. The AirWatch Add-on contains the Task Add AirWatch Device, Wipe AirWatch Device, etc. To open the Help for the Add-on, view or edit the Task of the Add-on in the Console and press F1.

Go to Setup > Add-ons for an overview of all installed Add-ons and to remove the Add-ons from the Datastore.

Go to http://res.com/connectors to download the RES ONE Connectors.

Add-ons for Core Pack

The RES ONE Automation Core Pack is automatically installed in your RES ONE Automation environment and contains Tasks such as Write Event Logs and Unix/Linux User Management. To open the Help for the Core Pack, view or edit the specific Task and press F1.
Error control for Tasks

Error control for Tasks in a Module

If a Module contains several Tasks, error control determines what should happen if one of those Tasks fails.

- With the option Stop Job on error, RES ONE Automation will stop if a Task fails. The remaining Tasks in the Module will not be started.
- With the option Continue Job on error, RES ONE Automation will move on to the next Task in the Module.

You can set this per Task, on the Task’s Properties tab.

Error control for actions in a Task

Tasks related to Active Directory, Exchange mailboxes, VMware ESX Virtual Machine and Files can contain several actions, for example moving and renaming a number of files. For these Tasks, an extra option Ignore errors is available on the Task’s Settings tab. With this option selected, RES ONE Automation will continue executing the configured actions, even if one of those actions fails. For example, if one of the files cannot be moved, this will produce an error, but RES ONE Automation will proceed with the next files to be moved. Without this option, the entire Task will fail if one action fails.

Note

The default value of error control for Tasks is Stop Job on error, except for the following Tasks, in which the default value is Continue Job on error:

- Query Computer Properties
- Query Disk Space
- Query Installed programs
- Query Service Properties
- Query TCP/IP Properties
Task Security context

By default, RES ONE Automation uses the Local System Account to execute a Task. However, the execution of certain Tasks requires credentials, for example to access a file share or an Exchange mailbox. Such credentials go in the Task's Security Context field.

- The user name should include the domain: domain\user.
- In Tasks that include a Domain field, this domain is filled out automatically as the default domain in the Security Context. This can be changed to another domain if necessary.
- Select Remember this for other Tasks to use the credentials in other Tasks as well.

Changing the security context password for all RES ONE Automation Tasks at once

If many Tasks in your RES ONE Automation environment use the same security context credentials, and the password changes, it can be tedious to change the password for these Tasks manually. To change the password for all Tasks at once:

1. Export the Library settings (Resources, Modules, Projects, Run Books) to a Building Block. This Building Block serves as a Building Block.
2. At Library > Modules, open a Module and change the security context password of one specific Task.
3. Export the Library settings (Resources, Modules, Projects, Run Books) to a Building Block.
4. Open the Building Block in a text editor.
5. Use the Search functionality to search for the changed security context password of the Task. The password is encrypted.
6. Copy the password and use the Search and Replace functionality to replace the encrypted passwords with the one from the changed Task.
7. Save the changed Building Block (in XML format).
8. In the Console, import the Building Block again into the Library node. When prompted, select the original GUID options.
9. When the Building Block is imported, check your changes to see whether the actions succeeded. If you are not happy with the result, you can import the original Building Block and revert to your old settings.
Active Directory Computer (Create, Manage, Delete, Query)

Use the Active Directory Computer Tasks to create an Active Directory Computer account; or to change, delete or query all Active Directory computer accounts:

- in a specific Active Directory folder (optionally including subfolders).
- with a computer name that matches a specific wildcard pattern.
- with a specific value for a specific Computer property.
- with a specific number of days since last login.

With Create Active Directory Computer, you can create an Active Directory Computer account with a specific name in a specific Organizational Unit, and you can determine which users or groups can join this computer to a domain.

With Manage Active Directory Computer, you can set and change the properties and group memberships of Active Directory Computer accounts. For example, you can target a group of Active Directory Computer accounts and set their location; or you can add a new group membership to all target Active Directory Computer accounts. New group memberships can be added or existing group memberships can be removed. Alternatively, set a list of existing group memberships to replace the existing memberships of the target Active Directory Computer accounts. You can also set a group as the primary group.

With Delete Active Directory Computer, you can delete Active Directory Computer accounts. Combined with the filter on number of days since last login, for example, you can use this Task to clean up Active Directory Computers that have not been used for a given length of time.

With Query Active Directory Computer, you can obtain overviews of all Active Directory computer accounts, filtered by Organizational Unit, by computer name, property value and/or number of days since last login.

Configuration

- If you are going to run the Task on a Domain controller, you can leave the Domain controller field on the Settings tab empty and select Local Agent (domain controller) instead.
- On the Settings tab, computer name refers to the pre-Windows 2000 name.
- In the Filter by property field, use wildcards only in combination with the operators LIKE and NOT LIKE.

Warning

- When you configure a Task to change or delete Active Directory Computers that match a set of criteria, there is a risk that more Active Directory Computer accounts are targeted than expected. By selecting the option Fail this Task if the number of items affected exceeds [x], you can create a safety net that prevents undesired results.
- The Task Query Active Directory Computer may take a long time if the queried Organizational Unit contains many items. This is particularly the case if the query retrieves the Number of days since last login, the Computer property Last logon date, or the Computer property Last logon server.
- It is possible to abort long running Active Directory Query Tasks. It may take around 10 seconds for the abort to be detected by the Agent, after which the Job will fail with status Aborted.
- Long running Active Directory Tasks that update, move or delete Active Directory objects cannot be aborted - the Task will always run until it is completed.

Notes

- The execution speed of Active Directory Queries may depend on the number of additional Active Directory properties that are to be reported in the query. On the Computer Properties tab, you can select a maximum number of 90 Active Directory properties.
- When browsing for an Organizational Unit, the list of Organizational Units that is shown depends on the Security context and the Domain controller. If the Domain controller field is empty, the list of Organizational Units depends on the Domain.
**Tips**

- When you browse for a specific OU, this information is automatically pasted in the relevant field and takes the following format: OU=T,OU=Amsterdam,OU=Netherlands,DC=d-energy,DC=local. However, if you set a parameter in this field, the data will have to be provided at the moment of input - in the correct format. To provide an example of the format, copy a sample path and paste it as the default value for the relevant parameter. With the Input setting *Show previous value*, the example will be shown whenever input is required for the parameter.
- In Tasks to set Active Directory properties for Active Directory objects, the variable %username% will be resolved as the name of the user under which the RES ONE Automation Agent runs. If you want to refer to the actual user name of the target Active Directory object instead, for example when you are setting a user's home drive using the Task *Manage Active Directory User*, use the variable %accountname%. This will be resolved as the pre-Windows 2000 user name of the Active Directory User, Computer, Group or Object that is being managed. This option is not available for Organizational Units.

---

**Active Directory Group (Create, Manage, Delete, Query)**

Use the Active Directory Group Tasks to create an Active Directory Group; or to change, delete and query all Active Directory Groups:

- in a specific Active Directory folder (optionally including subfolders).
- with a group name that matches a specific wildcard pattern.
- with a specific value for a specific Active Directory Computer property.
- without any members.

With *Create Active Directory Group*, you can create an Active Directory Group with a specific name in a specific Organizational Unit, and you can determine the group scope and type.

With *Manage Active Directory Group*, you can set and change the group properties, members and the group memberships of Active Directory Groups.

- On the *Members* tab, you can add or remove users as members of the group. You can also create a new list of members to replace the entire existing list of members.
- On the *Member of* tab, you can make the target group member of other groups, or remove membership of other groups. You can also create a new list of groups to replace the existing memberships of the target Active Directory Group.

With *Delete Active Directory Group*, you can delete Active Directory Groups. Combined with the filter *Only apply to groups without members*, for example, you can use this Task to clean up empty Active Directory Groups.

With *Query Active Directory Group*, you can obtain overviews of all Active Directory Groups, filtered by Organizational Unit, by group name, property value and/or by lack of members.

**Configuration**

- If you are going to run the Task on a Domain controller, you can leave the *Domain controller* field on the *Settings* tab empty and select *Local Agent (domain controller)* instead.
- On the *Settings* tab, group name refers to the pre-Windows 2000 name.
- In the *Filter by property* field, use wildcards only in combination with the operators LIKE and NOT LIKE.
- By default, group memberships defined on the *Member of* tab are added to any existing group memberships. To replace the existing list with the list defined in the Task, select *Replace all existing group memberships*. 
Warning

When you configure a Task to change or delete Active Directory Groups that match a set of criteria, there is a risk that more Active Directory Groups are targeted than expected. By selecting the option **Fail this Task if the number of items affected exceeds [x]**, you can create a safety net that prevents undesired results.

Notes

- The execution speed of Active Directory Queries may depend on the number of additional Active Directory properties that are to be reported in the query. On the **Computer Properties** tab, you can select a maximum number of 90 Active Directory properties. It may take around 10 seconds for the abort to be detected by the Agent, after which the Job will fail with status **Aborted**. Long running **Active Directory** Tasks that update, move or delete Active Directory objects cannot be aborted - the Task will always run until it is completed.

- When browsing for an Organizational Unit, the list of Organizational Units that is shown depends on the Security context and the Domain controller. If the Domain controller field is empty, the list of Organizational Units depends on the Domain.

- It is possible to abort long running **Active Directory Query** Tasks. It may take around 10 seconds for the abort to be detected by the Agent, after which the Job will fail with status **Aborted**. Long running **Active Directory** Tasks that update, move or delete Active Directory objects cannot be aborted - the Task will always run until it is completed.

- When configuring a Task **Manage Active Directory Group**, and you select the option **Replace all existing members** on the **Members** tab, **RES ONE Automation** will not replace the members for which the specified group has been marked as primary group.

Tips

- **When configuring a Task Manage Active Directory Group**, use Multi-select List parameters when specifying members on the **Members** tab and **Members of** tab. This makes it easier to specify these members when scheduling the Task in a Job.

- When you browse for a specific OU, this information is automatically pasted in the relevant field and takes the following format: OU=IT,OU=Amsterdam,OU=Netherlands,DC=dn-energy,DC=local. However, if you set a parameter in this field, the data will have to be provided at the moment of input - in the correct format. To provide an example of the format, copy a sample path and paste it as the default value for the relevant parameter. With the Input setting **Show previous value**, the example will be shown whenever input is required for the parameter.

- In Tasks to set Active Directory properties for Active Directory objects, the variable %username% will be resolved as the name of the user under which the **RES ONE Automation Agent** runs. If you want to refer to the actual user name of the target Active Directory object instead, for example when you are setting a user's home drive using the Task **Manage Active Directory User**, use the variable %accountname%. This will be resolved as the pre-Windows 2000 user name of the Active Directory User, Computer, Group or Object that is being managed. This option is not available for Organizational Units.
Active Directory Object (Move, Query)

With **Move Active Directory Object**, you can move target Active Directory Object(s):

- of a specific Active Directory Object type or types.
- in a specific Active Directory folder (optionally including subfolders).
- with a name that matches a specific wildcard pattern.

With **Query Active Directory Object**, you can obtain overviews of all Active Directory Objects, filtered by Organizational Unit, by (logon) name and/or by property value. By default, the Query results show the folder, name, account name and type of each Active Directory object, but the object properties included in the Query results is fully configurable.

**Configuration**

- If you are going to run the Task on a Domain controller, you can leave the **Domain controller** field on the **Settings** tab empty and select **Local Agent (domain controller)** instead.

**Warning**

When you configure a Task to move Active Directory Objects that match a set of criteria, there is a risk that more Active Directory Objects are targeted than expected. By selecting the option **Fail this Task if the number of items affected exceeds [x]**, you can create a safety net that prevents undesired results.

**Notes**

- The execution speed of Active Directory Queries may depend on the number of additional Active Directory properties that are to be reported in the query. On the **Computer Properties** tab, you can select a maximum number of 90 Active Directory properties. It is possible to abort long running **Active Directory Query** Tasks. It may take around 10 seconds for the abort to be detected by the Agent, after which the Job will fail with status **Aborted**. Long running **Active Directory** Tasks that update, move or delete Active Directory objects cannot be aborted - the Task will always run until it is completed.
- When browsing for an Organizational Unit, the list of Organizational Units that is shown depends on the Security context and the Domain controller. If the Domain controller field is empty, the list of Organizational Units depends on the Domain.
- It is possible to abort long running **Active Directory Query** Tasks. It may take around 10 seconds for the abort to be detected by the Agent, after which the Job will fail with status **Aborted**. Long running **Active Directory** Tasks that update, move or delete Active Directory objects cannot be aborted - the Task will always run until it is completed.

**Tips**

- When you browse for a specific OU, this information is automatically pasted in the relevant field and takes the following format: `OU=IT,OU=Amsterdam,OU=Netherlands,DC=d-energy,DC=local`. However, if you set a parameter in this field, the data will have to be provided at the moment of input - in the correct format. To provide an example of the format, copy a sample path and paste it as the default value for the relevant parameter. With the Input setting **Show previous value**, the example will be shown whenever input is required for the parameter.
- In Tasks to set Active Directory properties for Active Directory objects, the variable `%username%` will be resolved as the name of the user under which the RES ONE Automation Agent runs. If you want to refer to the actual user name of the target Active Directory object instead, for example when you are setting a user's home drive using the **Task Manage Active Directory User**, use the variable `%accountname%`. This will be resolved as the pre-Windows 2000 user name of the Active Directory User, Computer, Group or Object that is being managed. This option is not available for Organizational Units.
Active Directory Organizational Unit (OU) (Create, Manage, Delete, Query)

Use the Active Directory Organizational Unit (OU) Tasks to create an Active Directory Organizational Unit, or to change, delete or query all Organizational Units:

- in a specific Active Directory folder (optionally including subfolders).
- with an OU name that matches a specific wildcard pattern.
- without any objects.

With **Create Active Directory Organizational Unit**, you can create an Active Directory Organizational Unit with a specific name in a specific Active Directory folder.

With **Manage Active Directory Organizational Unit**, you can set and change the properties of Organizational Units:

- Description
- Street
- City
- State/province
- Zip/Postal Code
- Country/region
- Name (only for single OUs)

For example: you can target a group of Active Directory Organizational Units and change their address and location.

With **Delete Active Directory Organizational Unit**, you can delete Active Directory Organizational Units. Combined with the filter on Active Directory Organizational Units without any objects, for example, you can use this Task to clean up unused Active Directory Organizational Units.

With **Query Active Directory Organizational Unit**, you can obtain overviews of all Active Directory Organizational Units, filtered by Active Directory folder, OU property, OU name or empty folder. Per Agent, the detailed Job results show all the queried Organizational Unit Properties.

The query results show the total number of Organizational Units. Per Organizational Unit, the detailed results show all the queried Organizational Unit properties.

**Configuration**

- If you are going to run the Task on a Domain controller, you can leave the **Domain controller** field on the **Settings** tab empty and select **Local Agent (domain controller)** instead.
- In the **Filter by property** field, use wildcards only in combination with the operators LIKE and NOT LIKE.

**Warning**

When you configure a Task to change or delete Active Directory Organizational Units that match a set of criteria, there is a risk that more Active Directory Organizational Units are targeted than expected. By selecting the option **Fail this Task if the number of items affected exceeds [x]**, you can create a safety net that prevents undesired results.
Notes

- The execution speed of Active Directory Queries may depend on the number of additional Active Directory properties that are to be reported in the query. On the Computer Properties tab, you can select a maximum number of 90 Active Directory properties. It is possible to abort long running Active Directory Query Tasks. It may take around 10 seconds for the abort to be detected by the Agent, after which the Job will fail with status Aborted. Long running Active Directory Tasks that update, move or delete Active Directory objects cannot be aborted - the Task will always run until it is completed.

- When browsing for an Organizational Unit, the list of Organizational Units that is shown depends on the Security context and the Domain controller. If the Domain controller field is empty, the list of Organizational Units depends on the Domain.

- The filter on empty Organizational Units also targets Organizational Units that only contain other Organizational Units that are also empty. For example, you could have an Organizational Unit A, which contains Organizational Unit B, which in turn contains Organizational Unit C. If none of these Organizational Units contain any other objects, then all three Organizational Units will be targeted by a Task that filters on empty Organizational Units.

- It is possible to abort long running Active Directory Query Tasks. It may take around 10 seconds for the abort to be detected by the Agent, after which the Job will fail with status Aborted. Long running Active Directory Tasks that update, move or delete Active Directory objects cannot be aborted - the Task will always run until it is completed.

Tip

When you browse for a specific OU, this information is pasted automatically in the relevant field and takes the following format: OU=IT,OU=Amsterdam,OU=Netherlands,DC=d-energy,DC=local. However, if you set a parameter in this field, the data will have to be provided at the moment of input - in the correct format. To provide an example of the format, copy a sample path and paste it as the default value for the relevant parameter. With the Input setting Show previous value, the example will be shown whenever input is required for the parameter.
Active Directory User (Create, Manage, Delete, Query)

Use the Active Directory User Tasks to create an Active Directory user; or to change, delete or query all users:

- in a specific Active Directory folder (optionally including subfolders).
- with a user logon name that matches a specific wildcard pattern.
- with specific values for a specific property.
- with a specific number of days since last login.

With Create Active Directory User, you can create an Active Directory User in a specific Active Directory folder, and provide basic properties: logon name, password and password settings, and names.

With Manage Active Directory User, you can

- set and change all User properties of Active Directory users.
- set primary and other group memberships.
- set primary and other SMTP e-mail addresses.

For example: you can target a group of Active Directory users and change their e-mail addresses.

With Delete Active Directory User, you can delete Active Directory users, filtered by Organizational Unit, logon name, User property value and number of days since last login. This makes it possible, for example, to clean up unused Active Directory user accounts.

With Query Active Directory User, you can obtain overviews of all Active Directory users, filtered by Organizational Unit, logon name, user property value and number of days since last login. Per Agent, the detailed Job results show all queried user properties.

Configuration

- If you are going to run the Task on a Domain controller, you can leave the Domain controller field on the Settings tab empty and select Local Agent (domain controller) instead.
- On the Settings tab, user logon name refers to the pre-Windows 2000 name.
- In the Filter by property field, use wildcards only in combination with the operators LIKE and NOT LIKE.
- For Active Directory users, you can define telephone numbers in different categories: home phone, mobile phone, etc. In Active Directory, a user can have several phone numbers in each category. However, with the Task Manage Active Directory User you can set only one phone number per category - so, one home phone number, one mobile phone, etc.
- The functionality on the E-mail addresses tab is not supported for Microsoft Exchange Server 2010. To use this functionality for Microsoft Exchange Server 2010, use the Task Manage Exchange Mailbox (see page 111) instead.
- If you want to remove a primary group or primary e-mail address, set another as primary first.
- You can only add e-mail addresses for Active Directory Users with existing Exchange e-mailboxes.
- You can use and combine Windows variables when specifying e-mail addresses. The options are provided when you right-click the E-mail addresses field, but you can also type them in as required. See Using Windows variables in e-mail addresses (on page 98).
- By default, group memberships and e-mail addresses defined on the Member of and E-mail addresses tabs are added to any existing group memberships and e-mail addresses. To replace existing lists with the list defined in the Task, select Replace all existing group memberships or Replace all existing e-mail addresses.
- In a Task Query Active Directory User, when filtering on the property value Member of, you can specify multiple groups. Use LIKE or NOT LIKE if users should (or should not) be member of
at least ONE of the specified groups. When using any other separator, the user has to be a member of ALL specified groups.

**Warning**

When you configure a Task to change or delete Active Directory Users that match a set of criteria, there is a risk that more Active Directory Users are targeted than expected. By selecting the option *Fail this Task if the number of items affected exceeds [x]*, you can create a safety net that prevents undesired results.

**Notes**

- The execution speed of Active Directory Queries may depend on the number of additional Active Directory properties that are to be reported in the query. On the Computer Properties tab, you can select a maximum number of 90 Active Directory properties. It is possible to abort long running Active Directory Query Tasks. It may take around 10 seconds for the abort to be detected by the Agent, after which the Job will fail with status Aborted. Long running Active Directory Tasks that update, move or delete Active Directory objects cannot be aborted - the Task will always run until it is completed.

- When browsing for an Organizational Unit, the list of Organizational Units that is shown depends on the Security context and the Domain controller. If the Domain controller field is empty, the list of Organizational Units depends on the Domain.

- In a Task Manage Active Directory User, when setting the Active Directory property Home folder to create a home folder for a user on a share, the user as specified in the Security Context field requires permissions to access the share and create a folder.

- In a Task Manage Active Directory User, when setting the Active Directory property Home folder, use the following format:
  - Home folder = Local path:
    - Local path = <drive letter:\folder> (When left empty, the 'homeDirectory' and 'homeDrive' Active Directory properties are cleared.)
  - Home folder = Connect drive:
    - Connect drive = <drive letter>
    - Connect drive to = <\server\share\folder> (When left empty, the 'homeDrive' Active Directory property is set to this drive letter and the 'homeDirectory' Active Directory property does not change.)

- In a Task Manage Active Directory User you can add the Active Directory User account options *User must change password at next logon* and *User cannot change password*. However, if you add both options, this results in a conflicting combination: it will lead to a situation in which the user will be prompted to change the password at each log on, after which it will not be possible to log on. To prevent this situation, only one of the options will be selected after adding both Active Directory User account options. Although it is still possible to select both options at that point, this makes it clearer that the selection is a conflicting combination. We therefore recommend not to use this combination.

- In a Task Query Active Directory User, when selecting the option *Filter number of days since last login* (for example, greater than 90 days), it can take a while to run this Task. When this option is selected, first all Domain Controllers of the specified domain will be queried and then the Last-Logon attribute of each user on each Domain Controller will be queried. The reason because is that Microsoft does not replicate the Last-Logon user attribute for all Domain Controllers. For more information, go to [http://msdn.microsoft.com/en-us/library/ms676823%28v=vs.85%29.aspx](http://msdn.microsoft.com/en-us/library/ms676823%28v=vs.85%29.aspx).

- It is possible to abort long running Active Directory Query Tasks. It may take around 10 seconds for the abort to be detected by the Agent, after which the Job will fail with status Aborted. Long running Active Directory Tasks that update, move or delete Active Directory objects cannot be aborted - the Task will always run until it is completed.
**Tips**

- When you browse for a specific OU, this information is automatically pasted in the relevant field and takes the following format: OU=IT,OU=Amsterdam,OU=Netherlands,DC=dc-energy,DC=local. However, if you set a parameter in this field, the data will have to be provided at the moment of input - in the correct format. To provide an example of the format, copy a sample path and paste it as the default value for the relevant parameter. With the Input setting **Show previous value**, the example will be shown whenever input is required for the parameter.

- In Tasks to set Active Directory properties for Active Directory objects, the variable `%username%` will be resolved as the name of the user under which the RES ONE Automation Agent runs. If you want to refer to the actual user name of the target Active Directory object instead, for example when you are setting a user's home drive using the Task **Manage Active Directory User**, use the variable `%accountname%`. This will be resolved as the pre-Windows 2000 user name of the Active Directory User, Computer, Group or Object that is being managed. This option is not available for Organizational Units.

- When using environment variables, they sometimes should not be translated to their value on a specific Agent, but should remain as variables. If you place double percentage signs (%) around a variable, the variable is not translated to a value, but remains a variable. For example, with double percentage signs, `%%WINDIR%%` is translated to `%WINDIR%` and not to C:\windows.
Using environment variables in a Task “Manage Active Directory User”

Suppose you want to set the user profile path for all Active Directory users as a variable %profilepath%, so that this path can be defined depending on the Operating System: C:\Documents and Settings\%username% on Windows XP machines, and C:\users\%username% on Windows 7 machines.

To achieve this, first use the Task Set Environment Variables to create the system environment variable %profilepath% with the correct values on Windows XP and Windows 7 machines. Then configure the Task Manage Active Directory User to set %profilepath% as the User property user profile path for all users.

As a result, all user profile paths in Active Directory are: %profilepath%. Due to this variable, the user profile path is resolved depending on the Operating System under which a user logs on: C:\Documents and Settings\%username% or C:\users\%username%.

Using Windows variables in e-mail addresses

You can use and combine any of the following Microsoft Windows variables when specifying e-mail addresses. The options are provided when you right-click the E-mail addresses field, but you can also type them in as required.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Stands for</th>
<th>Example for user Dan P. Harris</th>
</tr>
</thead>
<tbody>
<tr>
<td>%g</td>
<td>Given name (first name)</td>
<td>%g yields dan</td>
</tr>
<tr>
<td>%[x]g</td>
<td>Uses x number of letters of the given name.</td>
<td>%1g yields d</td>
</tr>
<tr>
<td>%i</td>
<td>Middle initial</td>
<td>%i yields p</td>
</tr>
<tr>
<td>%s</td>
<td>Surname (last name)</td>
<td>%s yields harris</td>
</tr>
<tr>
<td>%d</td>
<td>Display name</td>
<td>%d yields Dan Harris (or Dan P. Harris. depending on the configured display name)</td>
</tr>
<tr>
<td>%[x]s</td>
<td>Uses x number of letters of the surname.</td>
<td>%3s yields har</td>
</tr>
<tr>
<td>%m</td>
<td>Exchange alias</td>
<td></td>
</tr>
</tbody>
</table>

Example of combined variables (for Dan P. Harris):

- %1g%s yields dharris
- %s%1g yields harrisd
- %s%g yields harrisdan
- %1g%i%s yields dpharris
Baseline Security (MBSA) (Query)

Use the Task Query Baseline Security (MBSA) to identify various risks on Agents, such as Microsoft Windows vulnerabilities, weak passwords, IIS and SQL vulnerabilities, and missing security updates. The Task is based on the Microsoft Baseline Security Analyzer technology (MBSA).

Configuration

• The option Query weak passwords checks the locally stored account information for weaknesses.
• The option Query SQL vulnerabilities also checks for MSDE issues.

Viewing the results

To view the results, open the completed Job at Jobs > Job History, go to the tab Tasks and double click to open the Query Baseline Security. The initial report shows the results per issue. Other views are also available:

• To view the results per Agent, click Swap rows and columns.
• To view the a more detailed list of security issues, click Computers per issue. From this overview, you can schedule a new Job, if necessary.
• To open the data in the MBSA application (provided the application is available on your computer), click Show these results in Microsoft Baseline Security Analyzer.

Tips

• If the detailed Job results show that Microsoft Windows security updates or SDK components security updates need to be installed on certain Agents, select the security issue and click Computers per issue.
• In the list of security updates:
  • click on a security update to view a description on the website of Microsoft.
  • use Click here to download to download the security update. You can then store this file as a Resource and schedule a Task Install Microsoft Update to install the update.

Saving and exporting information

• In each window, you can create a PDF file of the current view using the Instant Report button. The Instant Report PDF file can be saved.
• To export the results, open the completed Job at Jobs > Job History, go to the tab Tasks and double click to open the Query Baseline Security. Using the Export button, you can export the results to the following formats:
  • TXT (Tab delimited)
  • CSV (Comma delimited)
  • MBSA (MBSA version 2 or higher): This format can be opened on computers with Microsoft Baseline Security Analyzer 2.0 or higher installed.
  • XML (MBSA version 1.x): This format can be opened on computers with lower versions of Microsoft Baseline Security Analyzer installed. It can also be viewed as a regular XML file.

Note

If multiple Agents were queried when the Task was executed, and you export the Job results to XML or MBSA, RES ONE Automation will create one file per Agent.
Certificate (Import, Delete, Query)

Use the Tasks **Import** and **Delete Certificate** to install and remove certificates, certificate revocation lists and certificate trust lists on Agents. This Task is useful when certificates need to be used for client/server authentication, signing, etc., and helps you maintain an up-to-date storage of digital certificates, which is crucial for the security of the software in your environment.

Use the Task **Query Certificate** to search for certificates, certificate revocation Lists and certificate trust lists on Agents. Listing certificate stores and the certificates they contain can be very useful, especially when you want to search a certificate store for a specific certificate that needs to be used for client/server authentication, signing, etc. You can apply filters to search for specific information. For example, when you enable the expiration filter, you can track certificates that are about to expire.

**Note**

It can be useful to combine a Task **Import Certificate** with a Task **Query Certificate**, for example to discover which certificates will expire within the next month. This allows you to obtain new certificates and add them to the relevant stores, based on the query results.

Citrix Published Applications (Query)

Use the Task **Query Citrix Published Applications** to get a detailed overview of all or specific Citrix Published Applications on Agents.

For example, after you have installed a new Citrix server, you can use this Task to compare the published applications that run on all your Citrix servers, in order to ensure consistency. You can also use this Task to trace specific published applications that are installed on Agents; based on the results, you can publish an application that turned out to be missing on an Agent.

**Note**

If you schedule a recurring Job with this Task and view the results of this Job, the **History** tab in the detailed results window will display changes as they occur when comparing the query results with previous query results. Depending on the configuration of the query, this allows you to track changes in installed applications, printer drivers, services, shares, etc.
Citrix Workflow (Invoke)

Use the Task *Invoke Citrix Workflow* to execute Citrix workflows on Agents.

Citrix Workflow Studio is an infrastructure process automation platform that integrates components across the Citrix Delivery Center, including Citrix XenApp, XenDesktop, XenServer and Netscaler. Citrix Workflow Studio enables you to create, schedule, run, and manage workflows related to administrative tasks that span multiple products.

Using RES ONE Automation to handle Citrix workflows makes it easy to schedule Citrix workflows on many computers (Agents). The same version of the Citrix workflow is executed in the same manner across the entire set of Agents, ensuring consistent results. You can schedule Citrix workflows in RES ONE Automation by creating a workflow in Citrix Workflow Studio, exporting it as a ZIP file and storing it as a RES ONE Automation Resource.

When a Task *Invoke Citrix Workflow* is executed, RES ONE Automation monitors the Citrix Workflow and reports the workflow status of the executed workflow in the Job results.

Configuration

- The workflow resource is the Resource that contains the Citrix workflow (ZIP format).
- The account that you specify in the Security context field must have sufficient permissions to **deploy**, **schedule** and **start** Citrix workflows on the Agent(s).
- Optionally, you can specify a file that RES ONE Automation should grab during execution of the Task, for example a file with specific information that is created during execution of the workflow. The contents of this file are shown on the **Grab File** tab in the Job history. You can replace folder names with their associated environment string by right-clicking the field.
- Select **Wait for completion timeout** and specify a timeout if it is not necessary or desirable that RES ONE Automation waits for the Citrix Workflow to be completed before completing the Task. This option is useful if completion of the Citrix Workflow takes a long time, but it is not necessary to wait for its results before continuing with the next Task in a Job. When selected, RES ONE Automation will fail the Tasks that are not completed within the preset timeout period, while the Citrix Workflow will continue.
- A period of time after which an error condition is raised if some event has not occurred. A common example is sending a message. If the receiver does not acknowledge the message within some preset timeout period, a transmission error is assumed to have occurred.
- Any Citrix Workflow properties need to be specified on the **Workflow Properties** tab. When specifying Workflow properties, you can use parameters, functions and variables.

Notes

The Task *Invoke Citrix Workflow* requires the following software on an Agent:

- Citrix Workflow Studio 2.0.1 runtime
- Citrix Workflow Studio 2.0.1 Activity Libraries
Command (Execute)

Use the Task **Execute Command** to run a command on Agents. This allows you to schedule any action that can be performed by a command line. For example, you can use this Task to execute scripts or to automate the export of the results of a Job. See **Exporting Job results to XML** (on page 71).

**Configuration**

**Settings tab**

- In the **Command line** field, you can:
  - also refer to a Resource, if necessary. When inserting a Resource link by right-clicking the command line box, the GUID of the selected Resource will be used in the command line. The Resource will be added to the command line box as $Workspace<GUID>, where <GUID> is the unique identifier of the Resource. When the Task is executed, the Resource will be downloaded to the Agent’s cache automatically and will be called from that location.
  - use the function @[SCRIPT] as a placeholder for the script file. When the Task is executed by an Agent, the function @[SCRIPT] in the command line will be replaced by the path to the script file in the temporary folder on the Agent.
  - use paths. When using paths in the command line or script, encapsulate them with double quotes. For example, if you want to use a command line or script with a path for instance: C:\Program Files (x86)\RES Software\Automation Manager\Agent\agent.exe, encapsulate this path with double quotes: "C:\Program Files (x86)\RES Software\Automation Manager\Agent\agent.exe". By default Microsoft Windows has 8.3 naming convention enabled, which means that when an application is installed that uses a long path or filename, Microsoft Windows will create an 8.3 name for it. However, it is possible to disable 8.3 names. This means that Microsoft Windows will not create an 8.3 name for long path or file names. If you then specify a long path or file name in a Task **Execute Command**, Microsoft Windows will fail to resolve the 8.3 name and thus the Task will fail.
  - In the optional **Security Context**, select **Load user profile** if the execution of the command requires information that is stored in the user's profile, for example the location of the user's Home Directory.
  - The Task **Execute Command** can be an advanced Task. If you enter a complex command line, RES ONE Automation cannot interpret its validity. Therefore, always consider selecting **Execute command using the Windows command interpreter** and **Redirect standard output and standard error to log files**, and specifying whether the Task should fail if an error log is generated. Use these options in combination with **Task fails/succeeds if the executable returns the following exit code** to base the result of the Task on a specific exit code.
  - The option **Set parameter with exit code** allows you to place an exit code into a parameter, for use in another Task in the Module. For example, suppose you have configured a Module with a Task **Execute Command** and a Task that installs certain software. By using the option **Set parameter with exit code** in the Task **Execute Command**, you can use this parameter in a condition in the Task that installs software, to determine whether this Task should be executed.
  - With the option **Set parameter with standard output** you can place the standard output into a parameter, for use in another Task in the Module (for example, a Query). The standard output parameter can contain a maximum of 4MB. This option is only available when the option **Execute command using the Windows command interpreter** is selected.
• In the field **Timeout command execution after**, you can specify a maximum number of 9999 minutes (about 166 hours and 40 minutes). You can use parameters, functions and variables.
  
  • When 0 timeout is detected (when 0 was entered or when parameter, functions, variables translate to non-numeric values) the fallback will be the maximum timeout period (9999 minutes).

• In certain situations, Agents can continue to execute the remaining commands in the script when the timeout expired. You can prevent situations like these by selecting one of the following options (by default, none of the options are selected):
  
  - Select **Terminate process** to exit any active process when the timeout expires.
  - Select **Terminate process tree** to exit any active process and any associated processes when the timeout expires.

• In the optional field **Grab log file**, select a text file to be collected from the target machine. This file will be shown as an extra tab in the Job results named **Grabbed Log**.

**Script tab**

• Scripts can be typed in directly. You can use RES ONE Automation functions, RES ONE Automation parameters and environment Variables in Secure Shell commands. These Variables, functions and parameters will be parsed when the Task is executed.

• Microsoft Windows commands are internal commands without their own executable (for example, the DIR command).

• Use the **File extension of script** field to specify the file extension that RES ONE Automation should use. This extension is used to save the script as a script file in the specified format when the Task is executed. RES ONE Automation will save this file in the temporary folder on the Agent that executes the Task.

• Use the **Open in editor** button to open an external editor that is associated with the specified file extension in the **File extension of script** field. This makes it easier to create complex scripts and troubleshoot existing ones. After creating or editing the script, it will be copied from the external editor to the **Script** tab. For more information on how to configure file associations, see [http://msdn.microsoft.com/en-us/library/windows/desktop/cc144175%28v=vs.85%29.aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/cc144175%28v=vs.85%29.aspx).

  • The **Script** tab has a limit of 64KB. If you use an external editor to edit a script, RES ONE Automation will disregard any characters that exceed this limit.

• When executing this Task, log files will be generated and included in the detailed Job results. Before the Agent uploads these output log, error log or grabbed log files to the Dispatcher and make them available, the Agent will first compress the file if it has a size less than 100 MB.

• In the following situations, the compression of that file will not take place. As a result, the file will not be uploaded:
  
  - The file size cannot be determined
  - The file is empty
  - The file is larger than 100 MB

• The Job results will report if a file could not be uploaded because the file size could not be determined, or if the file size was larger than 100 MB. Empty files will not be reported. If a file cannot be uploaded, the outcome of the Task is not affected: the Task will still complete successfully, but the log file(s) that were not uploaded are not available in the Console.
Importing a RES ONE Workspace Building Block into a RES ONE Workspace environment using a Task “Execute Command”

In RES ONE Workspace, you can import a Building Block into a RES ONE Workspace environment by appending the command line parameter /add <path + buildingblock.xml> to the executable of the RES ONE Workspace Console, pwrtech.exe. If the RES ONE Workspace Console is installed on a machine on which also a RES ONE Automation Agent is running, you can automate this task using a Task Execute Command.

Configuration

1. Configure a Task Execute Command with the following command line:
   "%RESPFDIR%\pwrtech.exe" /add <path + buildingblock.xml> (for example, "%RESPFDIR%\pwrtech.exe" /add h:\xml\security_network.xml).
2. In the Security context field, use an account that has the role Technical Manager.
3. Select the option Execute command using the Windows command interpreter.
4. Schedule the Task on a RES ONE Automation Agent that is running on a machine on which also the RES ONE Workspace Console is installed.

Computer (Inventory)

Use the Task Inventory Computer to gather information about the hardware and operating system that is installed on Agents. For example, this allows you to determine which Agents already have a DVD/CD drive and on which Agents you need to install one.

Note

If you schedule a recurring Job with this query and view the results of this Job, the Job History tab in the detailed query results window will display changes as they occur when comparing the query results with previous query results. Depending on the configuration of the query, this allows you to track changes in installed applications, printer drivers, services, shares, etc.
Computer (Reboot, Shutdown)

Use the Tasks Reboot and Shutdown Computer to reboot or shut down Agents automatically. A restart is often required after the installation of new software or when a computer has been moved to a different domain.

Configuration

- In the Tasks Reboot Computer, the option Reboot computer after shutdown is enabled by default.
- In the Tasks Shutdown Computer, the option Reboot computer after shutdown is disabled by default.
- With the option Display Duration, you can specify how long the shutdown dialog box message will be shown in the Console. You can display a time duration in seconds between 1-999 seconds. The specified time will also affect the moment at which the shutdown timeout starts.
- With the option Wait for Agent to come back online, you can specify whether a Job should wait after a reboot with continuing with the next Task in the Job until the Agent has come back online. This option is enabled by default. It can be useful to disable the option in scenarios in which the computer on which the Agent is running switches to a different OS after a reboot and a different Agent comes online. This prevents RES ONE Automation from waiting until the original Agent comes back online.
- With the option Shutdown timeout, you can specify how long it will take before any running applications that are still running during the execution of this Job are forcibly closed. You can display a time duration in seconds between 1-999 minutes.

Computer (Uptime)

Use the Task Uptime Computer to view information about the boot time and uptime of Agents. This is useful to determine whether you are dealing with a memory leak. Based on the results of this Query, you can decide whether an Agent should be rebooted. You can automate this sequence of events by combining the Task with a Task Reboot Computer.
Computer Properties (Manage)

Use the Task Manage Computer Properties to manage the computer properties of Agents. These properties identify a computer on the network.

Configuration

- Use Change Computer Security Identifier to apply a new SID to the computer. This is useful if you use disk duplication software and you need to ensure the uniqueness of an SID. The computer needs to be rejoined to the domain after it has rebooted. Changing computer SIDs is not yet supported for x64-systems.

- Use Change primary DNS Suffix to apply a new primary DNS suffix to the computer. The default setting for the local primary DNS suffix of the computer is the same as the Active Directory domain name. If you change the DNS suffix, the domain membership of the computer will not be affected. If you enter a DNS suffix different from the Active Directory domain name, you must register the new full computer name in the Active Directory domain.

- For the option Change (physical) NetBIOS name > based on list of MAC addresses, upload a Resource that is a comma-separated file (CSV) containing a list of MAC addresses and their intended NetBIOS names. To create a valid CSV file for this option:
  - separate the segments of the MAC address by colons (:). Other separators will not work.
  - separate the MAC address and the intended NetBIOS name by a comma (,). Other separators will not work.
  - provide a valid NetBIOS name.

Example

00:50:58:8E:75:A4, sales10
00:50:63:8D:73:EA, sales11
00:28:61:8D:59:9C, marketing08
00:0A:29:AF:1B:A6, QA09

When the Task Manage Computer Properties is executed on an Agent, RES ONE Automation will identify the Agent’s MAC address and will change the computer’s NetBIOS name to the matching name given in the CSV file. This can be useful if you have a supply of anonymous computers in storage, identified only by their MAC address, and you want to give each computer a department-related name when it is delivered to a user.

Notes

- You can only join a computer to a different domain or workgroup if you change its computer name first (because a computer can only be a member of one domain at a time).
- If you join a computer to a workgroup, but it is already a member of a domain, the computer is automatically removed from the domain.
- If you add a computer to a different domain, it needs to be rebooted twice for the policy settings to take effect: the first reboot copies the policy settings from the domain to the computer; the second reboot causes the policy settings to take effect. You can schedule these reboots by following the Task Manage Computer Properties with a Task Reboot Computer.
- If changing the NetBIOS name of a computer, please note that it can have a maximum of 15 characters.
- After successfully changing the computer name, the computer should be rebooted immediately afterwards to avoid losing access for domain users. You can schedule this reboot by following the Task Manage Computer Properties with a Task Reboot Computer.
Computer Properties (Query)

Use the Task **Query Computer Properties** to gather information about the computer properties of Agents. For example, you can combine this Task with the Task **Manage Computer Properties**, to join a computer to another domain or workgroup based on the Query results.

**Note**

If you schedule a recurring Job with this Task and view the results of this Job, the **Job History** tab in the detailed results window will display changes as they occur when comparing the query results with previous query results. Depending on the configuration of the query, this allows you to track changes in installed applications, printer drivers, services, shares, etc.

Database Statement (Execute, Query)

- Use the Task **Execute Database Statement** to run a preconfigured SQL (Structured Query Language) script on a database. This allows you to maintain your database with RES ONE Automation. For example, this allows you to schedule a recurring cleanup Task of the database.
- Use the Task **Query Database Statement** to run an SQL (Structured Query Language) query on a database. This allows you to query your database with RES ONE Automation.

**Configuration of a Task Execute Database Statement**

**Settings tab**

- If you select **Use SQL script from Resource** on the **Settings** tab, the relevant SQL script must be available as a Resource.
- If you run the SQL script on a [Microsoft SQL Server]-based Datastore, select in the **Database authentication** field whether to connect to this Datastore using **SQL Server Authentication** or **Windows Authentication**. This field is only enabled when selecting Microsoft SQL Server in the **Database type** field.
  - When selecting **Windows Authentication**, specify logins in the format Domain\User name or Domain\Group name.
- It is very useful to use Module parameters in these Tasks. This allows you to make it generic: when you schedule the Task, you will then be prompted for example for the database server, the database name and the login credentials.

**Script tab**

- Scripts can be typed in directly.
- Use the **File extension of script** field to specify the file extension that RES ONE Automation should use. This extension is used to save the script as a script file in the specified format when the Task is executed. RES ONE Automation will save this file in the temporary folder on the Agent that executes the Task.
- Use the **Open in editor** button to open an external editor that is associated with the specified file extension in the **File extension of script** field. This makes it easier to create complex scripts and troubleshoot existing ones. After creating or editing the script, it will be copied from the external editor to the **Script** tab. If no associated editor can be found, the script will be opened in Notepad. For more information on how to configure file associations, see [http://msdn.microsoft.com/en-us/library/windows/desktop/cc144175(v=vs.85).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/cc144175(v=vs.85).aspx).
- Log and error files are included in the detailed Job results.
Notes

- You can run SQL scripts on any database that is supported by RES ONE Automation.
- Per Task, your SQL script can perform actions on multiple databases, as long as they are located on the same database server.
- When executing this Task, log files will be generated. Before the Agent uploads these output log, error log or grabbed log files to the Dispatcher and make them available, the Agent will first compress the file if it has a size less than 100 MB. In the following situations, the compression of that file will not take place. As a result, the file will not be uploaded:
  - The file size cannot be determined
  - The file is empty
  - The file is larger than 100 MB
- The Job results will report if a file could not be uploaded because the file size could not be determined, or if the file size was larger than 100 MB. Empty files will not be reported. If a file cannot be uploaded, the outcome of the Task is not affected: the Task will still complete successfully, but the log file(s) that were not uploaded are not available in the Console.
- The Scripts tab has a limit of 64KB. If you use an external editor to edit a script, RES ONE Automation will disregard any characters that exceed this limit.

Disk Fragmentation (Defragment, Analyze)

Use the Task **Defragment Disks** to defragment and optimize local drives and system drives. If you schedule this Task as a regular Job, you can schedule automatic disk defragmentation at regular intervals.

Use the Task **Analyze Disk Fragmentation** to gather information about the degree of fragmentation of the drives on Agents.

Tips

- You cannot defragment volumes that are marked as dirty, which indicates possible corruption. Therefore, it is recommended to schedule a Task **Analyze Disk Fragmentation** before you schedule a Task **Defragment Disks**. This also enables you to automate the defragmentation and optimization of drives.
- Optimization of folders is restricted to NTFS file systems, because FAT systems do not allow changes in the address of the first cluster of directories.
Disk Space (Query)

Use the Task **Query Disk Space** to check the available disk space and folder space usage on Agents. If you select to query a specific folder, this allows you to check the amount of space it uses and the size of the files it contains.

For example, you can use this Task to check the available disk space on a server at regular intervals, in order to prevent it from running out of disk space. You can also use the Task to check the size of the folder in which an application has been installed. You can set the Task **Invoke Windows Installer** to install the application automatically, depending on the outcome of the Query.

Configuration

- When querying on used folder space, you can specify the Start folder by clicking , by specifying the path to the folder manually or by using parameters and/or functions. You can specify parameters and functions manually or by right-clicking the Start folder field and clicking **Insert Parameters** or **Insert Functions**. See **Parameters** (on page 257) and **Functions** (on page 246).
- When setting an evaluator based on disk, you can use environment variables, parameters and functions to specify the disk. This is useful because mapped drive letters may vary per Agent. For example, if you use the environment variables `%systemdrive%`, the Agent that executes the Task will decide what drive to use.
- When viewing the Job results of a Task **Query Disk Space**, select **Show in KB** to view the Job results of all Agents in KB. When a Task **Query Disk Space** is executed, the amount of free disk space of Agents can be reported in GB, in MB or in KB, depending on the amount of disk space that is available on each Agent. This makes it difficult to compare the Job results of multiple Agents when they are exported to a CSV file. The option **Show in KB** allows for an easy comparison in a CSV file.

<table>
<thead>
<tr>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>If you schedule a recurring Job with this Task and view the results of this Job, the Job History tab in the detailed results window will display changes as they occur when comparing the query results with previous query results. Depending on the configuration of the query, this allows you to track changes in installed applications, printer drivers, services, shares, etc.</td>
</tr>
</tbody>
</table>
E-mail (Send)

Use the Task Send E-mail to send e-mails, including attachments, using SMTP.

This Task is especially useful when combined with other Tasks in a Module. For example, if a Module contains a Task Create Exchange Mailbox, you can use the Task Send E-mail to send an e-mail to this mailbox once it has been created. You can also use this Task to send e-mail notifications when another Task fails, by configuring conditions (see page 233).

This Task is closely related to the global setting Send mail message (see page 35):

- With the global setting Send mail message, you can specify whether RES ONE Automation sends an e-mail for every Job that is executed.
- With the Task Send E-mail, you can send an e-mail only for the Job in which the Task is included.

Configuration

- The server is the mail server that sends the e-mail. By default, SMTP uses Port 25.
- When the option SSL Enabled is selected, the e-mail message is sent over SSL when executed on the Agent+. When executed on a legacy Agent (and the option SSL is enabled), the Task will fail. The communication of the SMTP server only supports Explicit SSL.
- In the Security Context field, provide the credentials for the e-mail address specified in the Sender field.
- You can use wildcard characters when attaching files.

Environment Variables (Set, Delete)

Use the Tasks Set Environment Variables and Delete Environment Variables to add, change or delete environment variables on Agents.

Configuration

For the option Append the value data if value already exists, add a semi-colon (;) in front of the value data that you want to append to the existing value. Without the semi-colon, the value will be replaced.

Notes

- If the environment variables have been added or changed successfully, the system changes will be immediately available for new processes on Agent(s) that run the Task. Existing processes will continue to use the "old" environment variables.
- Job environment variables only exist during execution of the Job. After the Job has finished, all Job environment variables will be discarded.
- System environment variables are persistent, even when the Job has finished. Job environment variables are useful to parameterize batch files.

Tips

- It can be useful to combine this Task with the Task Query Environment Variables, to gather information about environment variables on Agents and to take action based on the query results.
- When using environment variables, they sometimes should not be translated to their value on a specific Agent, but should remain as variables. If you place double percentage signs (%) around a variable, the variable is not translated to a value, but remains a variable. For example, with double percentage signs, %%WINDIR%% is translated to %WINDIR% and not to C:\windows.
Environment Variables (Query)

Use the Task **Query Environment Variables** to gather information about environment variables on Agents.

**Configuration**

- Select **System environment variables** to query how the environment variables will be set up for new processes.
- Select **Process environment variables** to query how the environment variables are actually set up, including the source of configuration (Job or process-related environment variables).

Event Logs (Query, Write)

Use the Event Logs Tasks to view or create entries in the Microsoft Event Log.

- Use the Task **Query Event Logs** to view the event logs of Agents for errors and warnings. Performing this Task ensures that historical event logs are stored, whereas this data is normally lost when the Microsoft Event Viewer reaches its maximum size. See also **Appendix: Event logs** (on page 309) for a complete list of available Event IDs and messages.
- Use the Task **Write Event Log** to write an entry in an Agent’s Application Event Log, Security Event Log or System Event Log.
  - In the **Log name** field, provide the name of the Microsoft Event Log in which the event should be written. If the name provided does not match an existing log on the Agent, the system will create a new log with the specified name.
  - In the **Entry type** field, please type one of the following texts:
    - Error
    - Information
    - FailureAudit
    - SuccessAudit
    - Warning
Exchange Mailbox (Create, Manage, Move, Export, Disable)

- Use the Tasks Create, Manage and Move to create, manage and move mailboxes on Microsoft Exchange servers.
- Use the Task Export Exchange Mailbox to export mailboxes on Microsoft Exchange servers to a PST file. This is useful when you want to back up or archive Exchange mailboxes, for example before deleting mailboxes or for legal purposes.
- Use the Task Disable Exchange Mailbox to disable mailboxes on Microsoft Exchange servers. This will remove all Exchange attributes from the corresponding Active Directory user accounts. The user account itself will be retained. By default, Exchange will retain disabled mailboxes for 30 days. It is possible to reconnect disabled mailboxes to an Active Directory user account within this retention period.

With these Tasks, you can perform these actions in a consistent manner, without the risk of inconsistencies in the data entered manually in Exchange itself.

Configuration

- When configuring a Task Create Exchange Mailbox, the User name on the Settings tab is the user logon name (pre-Windows 2000).
- When configuring a Task Manage Exchange Mailbox to manage mailboxes on Microsoft Exchange Server 2010 or higher, use the E-Mail Addresses tab to add primary and other SMTP e-mail addresses or remove these e-mail address for Active Directory users with existing Exchange e-mailboxes. When using Microsoft Exchange Server 2003 or 2007 (only applicable for current Agent) you need to this by configuring a Task Manage Active Directory User.
  - You can only add e-mail addresses for Active Directory Users with existing Exchange e-mailboxes. When specifying e-mail addresses, you can use and combine Windows variables. The options are provided when you right-click the E-mail addresses field, but you can also type them in as required. See Using Windows variables in e-mail addresses (on page 98).
  - If you want to remove a primary e-mail address, set another as primary first.
  - By default, any specified e-mail addresses are added to any existing e-mail addresses. To replace existing lists with the list defined in the Task, select Replace all existing SMTP e-mail addresses.
- When configuring a Task Export Exchange Mailbox, in the PST file name field on the Mail Properties tab, provide a location where the PST file can be stored. Click \[\] to browse to a location. If you specify a location where a PST file of the same name already exists, the old file will be replaced by the new one.
- Microsoft Exchange Server 2010 and 2013 offer the capability to use Database Availability Groups (DAGs). DAGs are fully supported in RES ONE Automation. When configuring Exchange Mailbox Tasks in this scenario, specify as Exchange server the relevant cluster node (Exchange server) in the DAG, NOT the name of the DAG.

**Warning**

When you configure a Task Create, Manage, Move, Export or Disable Exchange Mailbox, there is a risk that more Exchange mailboxes are targeted than expected. By selecting the option Fail this Task if the number of items affected exceeds \[x\], you can create a safety net that prevents undesired results.

**Notes**

- Tasks Move Exchange Mailbox that move mailboxes from older versions of Microsoft Exchange Server to Microsoft Exchange Server 2010 or higher or vice versa should only be scheduled on Agents running Microsoft Exchange Server 2010 or higher.
### Tips

- When configuring a Task **Create Exchange Mailbox**, you can configure a condition that sets a parameter, and make the Exchange folder in which the mailbox should be created dependent on the surname of the user. The value of the parameter can then determine whether the Task should be executed or skipped.
- For a Task **Export Exchange Mailbox**, when you specify \%username% in the PST file name, RES ONE Automation will automatically convert this variable to the name of the mailbox user.

### Permissions and prerequisites to create, manage, move or disable Exchange mailboxes

<table>
<thead>
<tr>
<th>Exchange version on Agent:</th>
<th>Basic requirements:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exchange Administrator Tools</td>
<td><strong>Account used in Task:</strong></td>
</tr>
<tr>
<td></td>
<td>• Permissions on the Exchange Server(s).</td>
</tr>
<tr>
<td>Exchange Server 2003 (not supported by Agent+, only for Agents)</td>
<td><strong>Account used in Task:</strong></td>
</tr>
<tr>
<td></td>
<td>• Mailbox permissions on the Exchange Server(s). You can set up these permissions using the Exchange System Manager.</td>
</tr>
<tr>
<td></td>
<td>• Security role “Exchange Full Administrator”.</td>
</tr>
<tr>
<td></td>
<td>** Installed on Agent:**</td>
</tr>
<tr>
<td></td>
<td>• CDOEXM (part of Exchange Management Components).</td>
</tr>
<tr>
<td>Exchange Server 2007 (not supported by Agent+, only for Agents)</td>
<td><strong>Account used in Task:</strong></td>
</tr>
<tr>
<td></td>
<td>• Mailbox permissions on the Exchange Server(s). You can set up these permissions using the Exchange Management Console or the Exchange Management Shell.</td>
</tr>
<tr>
<td></td>
<td>• Security role “Exchange Organization Administrators”.</td>
</tr>
<tr>
<td></td>
<td><strong>Account requirements for the Task Move Exchange Mailbox:</strong></td>
</tr>
<tr>
<td></td>
<td>• Security role “Exchange Server Administrators”.</td>
</tr>
<tr>
<td></td>
<td>• Security role “Local Administrators” (on the Agent and on the Exchange server).</td>
</tr>
<tr>
<td>Exchange Server 2010 / 2013 / 2016</td>
<td><strong>Account used in Task:</strong></td>
</tr>
<tr>
<td></td>
<td>• Mailbox permissions on the Exchange Server(s). You can set up these permissions using the Exchange Management Console or the Exchange Management Shell.</td>
</tr>
<tr>
<td></td>
<td>• Management role group “Organization Management”.</td>
</tr>
<tr>
<td></td>
<td><strong>Other requirements:</strong></td>
</tr>
<tr>
<td></td>
<td>• To use the browse functionality in the Exchange Mailbox Tasks to browse to Exchange Servers and mailbox stores from the Console, the corresponding Exchange Management Tools need to be installed on the computer on which the Console is running.</td>
</tr>
</tbody>
</table>
### Permissions and prerequisites to export Exchange mailboxes

<table>
<thead>
<tr>
<th>Exchange version on Agent:</th>
<th>Basic requirements:</th>
</tr>
</thead>
</table>
| Exchange Administrator Tools | **Account:**  
  • Mailbox permissions on the Exchange Server(s)  
  • "Send as" and "Receive as" permissions on the mail store containing the mailbox to be exported. (Please note that the Administrator does not have these permissions by default.)  

**Installed on Agent:**  
• This Task also runs on Agents running just Microsoft Outlook 2000 or higher, installed in Corporate/Workgroup Mode and/or configured to connect using the Exchange Server (not POP3 or IMAP). |

| Exchange Server 2003 (not supported by Agent+, only for Agents) | **Account:**  
  • Mailbox permissions on the Exchange Server(s). You can set up these permissions using the Exchange System Manager.  
  • Security role "Exchange Full Administrator”  
  • "Send as" and "Receive as" permissions on the mail store containing the mailbox to be exported. (Please note that the Administrator does not have these permissions by default.)  

**Installed on Agent:**  
• CDOEXM (part of Exchange Management Components)  
• This Task also runs on Agents running just Microsoft Outlook 2000 or higher, installed in Corporate/Workgroup Mode and/or configured to connect using the Exchange Server (not POP3 or IMAP). |

| Exchange Server 2007 (not supported by Agent+, only for Agents) | **Account:**  
  • Mailbox permissions on the Exchange Server(s). You can set up these permissions using the Exchange Management Console or the Exchange Management Shell.  
  • Security role "Exchange Organization Administrators".  

**Installed on Agent:**  
• 32-bit computer with the following installed:  
  • 32-bit version of Microsoft Outlook 2000 (v6.0) or higher |

| Exchange Server 2010 | **Account:**  
  • Management role group “Organization Management”  
  • Management role 'Mailbox Import Export'  

**Installed on Agent:**  
• 64-bit computer with the following installed:  
  • Microsoft Exchange Server 2010 role  
  • 64-bit version of Microsoft Outlook 2010  
  • Administration Tools |
### Exchange version on Agent

**Exchange Server 2010 SP1**

<table>
<thead>
<tr>
<th>Basic requirements:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Account:</strong></td>
</tr>
<tr>
<td>• Management role group “Organization Management”</td>
</tr>
<tr>
<td>• Management role &quot;Mailbox Import Export&quot;</td>
</tr>
<tr>
<td><strong>Installed on Agent:</strong></td>
</tr>
<tr>
<td>• 64-bit computer with the following installed:</td>
</tr>
<tr>
<td>• Microsoft Exchange Server 2010 role</td>
</tr>
<tr>
<td>• 64-bit version of Microsoft Outlook 2010</td>
</tr>
<tr>
<td>• Administration Tools</td>
</tr>
<tr>
<td><strong>Share:</strong></td>
</tr>
<tr>
<td>• Read/Write permissions for the group &quot;Exchange Trusted Subsystem&quot;</td>
</tr>
</tbody>
</table>

**Exchange Server 2013 / 2016**

<table>
<thead>
<tr>
<th>Basic requirements:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Account:</strong></td>
</tr>
<tr>
<td>• Management role group “Organization Management”</td>
</tr>
<tr>
<td>• Management role &quot;Mailbox Import Export&quot;</td>
</tr>
<tr>
<td><strong>Installed on Agent:</strong></td>
</tr>
<tr>
<td>• 64-bit computer with the following installed:</td>
</tr>
<tr>
<td>• Microsoft Exchange Server 2013/2016 role</td>
</tr>
<tr>
<td>• 64-bit version of Microsoft Outlook 2013/2016</td>
</tr>
<tr>
<td>• Administration Tools</td>
</tr>
<tr>
<td><strong>Share:</strong></td>
</tr>
<tr>
<td>• Read/Write permissions for the group &quot;Exchange Trusted Subsystem&quot;</td>
</tr>
</tbody>
</table>

---

**Example Create Exchange Mailbox**

**Example scenario**

Suppose you have two Exchange servers. On server 1, you create mailboxes for users with surnames starting with A-M. On server 2, you create mailboxes for users with surnames N-Z. User names consist of only the user's surname. This can be set up in RES ONE Automation using a Project containing two Modules:

- The Module “Mailbox on Server 1” contains a Task to create an Exchange Mailbox on server 1; and the Module parameter $[surname]$ is set in the user name field.
- The Module “Mailbox on Server 2” contains a Task to create an Exchange Mailbox on server 2; and the Module parameter $[surname]$ is set in the user name field.
- The Project parameter $[surname]$ is linked to both Module parameters $[surname]$.

To configure the value of the parameter $[surname]$ as the basis for the condition to skip either Module 1 or Module 2:

1. Configure a condition for the Module “Mailbox on server 1”:
   1. Select the Module “Mailbox on Server 1” on the Project’s Modules tab.
   2. Click **Condition** and click **Add > Parameter**.
   3. Configure the condition: `surname LIKE [n-z]*` and click **OK**.
   4. Under Action, configure **If condition is true, skip this Module**.
   5. Click **OK** to return to the Modules tab.

2. Configure a condition for the Module “Mailbox on server 2”:
1. Select the Module "Mailbox on Server 2" on the Project’s Modules tab.
2. Click Condition and click Add > Parameter.
3. Configure the condition: surname LIKE [a-m]* and click OK.
4. Under Action, configure If condition is true, skip this Module.
5. Click OK to return to the Modules tab.

If you provide the surname Zimmerman as value for the parameter $[surname]$, then Module 1 will not meet the condition and will be skipped. Module 2 will meet the condition, and will be executed, so the mailbox is created on server 2.

If you provide the surname Johnson, it works the other way around: Module 1 is executed and Module 2 is skipped.

Exchange Mailbox (Query)

Use the Task Query Exchange Mailbox to obtain an overview of the number of mailboxes, their total size, and the number of Exchange servers in a specific domain. For example, perform this Task at set intervals to monitor the total growth of mailboxes in your environment. Click through the results to see detailed information per user, mailbox, server, etc.

Configuration

Use the filters to drill down to more specific information, for example to view how many mailboxes a specific OU has, and what size they add up to.

Note

Permissions and prerequisites to query Exchange mailboxes

The Agent(s) running this Task must be able to connect to the specified domain.

<table>
<thead>
<tr>
<th>Exchange version on Agent:</th>
<th>Basic requirements:</th>
</tr>
</thead>
</table>
| Exchange Administrator Tools | **Account used in Task:**  
|                              | • Permissions on the Exchange Server(s).  
|                              | • "Send as" and "Receive as" permissions on the mail store containing the mailbox. (Please note that the Administrator does not have these permissions by default.) |
| Exchange Server 2003        | **Account used in Task:**  
| (not supported by Agent+, only for current Agents) | • Mailbox permissions on the Exchange Server(s). You can set up these permissions using the Exchange Server System Manager.  
|                              | • "Send as" and "Receive as" permissions on the mail store containing the mailbox. (Please note that the Administrator does not have these permissions by default.)  
|                              | • Security role "Exchange View-Only Administrators". |
| Account used in Task:       | **Installed on Agent:**  
|                              | • CDOEXM (part of Exchange Management Components) |
| Exchange Server 2007        | **Account used in Task:**  
| (not supported by Agent+, only for current Agents) | • Mailbox permissions on the Exchange Server(s). You can set up these permissions using the Exchange Management Console or the Exchange Management Shell.  
|                              | • Security role "Exchange Organization Administrators" or "Exchange View-Only Administrators". |
| Exchange Server 2010 / 2013 / 2016 | **Account used in Task:**  
|                                | • Mailbox permissions on the Exchange Server(s). You can set up these permissions using the Exchange Management Console or the Exchange Management Shell.  
|                                | • Membership of the management role group "Organization Management" or "View-Only Organization Management". |
| Account used in Task:       | **Installed on Agent:**  
|                                | • When filtering by server, Exchange Management Tools need to be installed to be able to browse to Microsoft Exchange servers. |
File Permissions, Printer Permissions, Registry Permissions, Share Permissions (Set)

Use the Tasks Set File Permissions, Set Printer Permissions, Set Registry Permissions and Set Share Permissions to add, change, replace and remove permissions on files, folders, printers, registry settings and shares.

**Configuration**

- **Set NTFS file/folder security** allows you to change security permissions and access rights on files and folders for specific users.
- **Set Printer Permissions** allows you to manage security permissions and access rights on specific printers for specific users and groups.
- **Set Registry Permissions** allows you to change security permissions and access rights on registry keys for specific users.
- **Set Share Permissions** allows you to change security permissions and access rights on shares for specific users.
- When configuring these Tasks to set permissions, you can use the following additional options:
  - With the option **Remove all existing permissions**, the permissions of all existing users and groups are removed.
  - With the option **Replace permissions on child objects**, the group or user that you have configured at permissions is added to the already existing users or groups.
  - When selecting both options, the permissions of all existing users and groups are replaced by the permissions that you have added.

**Notes**

- Security for files and folders is limited to the NTFS file system.
- It can be useful to combine these Task with a Task Query File Permissions, Query Printer Permissions, Query Registry Permissions or Query Share Permissions, to trace which security settings are active in your network and to change these settings, depending on the query results.

File Permissions, Printer Permissions, Registry Permissions, Share Permissions (Query)

Use the Tasks Query File Permissions, Query Printer Permissions, Query Registry Permissions and Query Share Permissions to retrieve security permissions on files, folders, shares and registry settings on Agents.

Combined with the Task Set File Permissions, this Task allows you to maintain up-to-date and pertinent security settings on the files, folders, shares and registries in your network.

**Notes**

- Security of files and folders is limited to the NTFS file system.
- When configuring a Task Query Share Permissions, please note that Agents run under the local system account, which means that the local system account must have the appropriate permissions to the share that you want to query.
Files (Perform Operations)

Use the Task Perform File Operations to move, copy, rename and delete files and folders; to create folders; and to create and edit INI files. File operations can be performed on local folders and disks on Agents.

After running this Task, you can view its log file on the Log tab in its Job History.

Configuration

• **Edit/Create INI file**: it is also possible to configure files with other extensions, but the configured file will still be treated as an INI file.
• If you try to copy, move or create a file or folder to a destination path that does not exist, any folders missing from that path will be created automatically.
• Specify a backslash (\) at the end of the source path if you want RES ONE Automation to perform the file operation only with the contents of the specified folder.

Wildcards

Contrary to other places where wildcards are allowed, the Task File Operations interprets wildcards in source file and folder paths within the backslashes that define the different levels of the path. This means that you can use an expression that includes several wildcards.

For example, C:\Documents and Settings\*\*\temp\* resolves as:

• all files and folders in C:\Documents and Settings\Default User\Local Settings\Temp
• all files and folders in all users' My Documents\temp

Files (Query)

Use the Task Query Files to check file information on Agents.

For example, you can use this Task to check whether a specific version of a DLL file is available where necessary. If not, you can configure the Task Download Resources to obtain the correct file, so that it can be installed. The Microsoft Installed Programs List does not always list applications that have been installed through a command line. This Task checks the relevant EXE file on Agents.

When scanning the version of a file, you can query the file version or the product version.

When scanning the contents of a file, the following file types are supported: ANSI, UTF8, UTF8 with BOM, UTF16-BE and UTF16-LE.

---

**Note**

Separate multiple entries with a semi-colon (;).

---

Hosts (Discover)

Use the Task Discover Hosts to search for IP addresses within a specified IP range. The Task will return all computers and other network devices that it can contact, including their MAC address.

This is useful, for example, when you need to deploy new Agents and you want to check which computers can be pinged. Because you can execute this Task on Agents in remote subnets, you can also ping devices outside your own subnet.

For the Agent+, IPv4 and IPv6 addresses are supported. For the legacy Agent, only IPv4 is supported.
Installed Programs (Query)

Use the Task **Query Installed Programs** to obtain an overview of applications that have been installed on Agents and check where/whether these applications have been installed. For example, this allows you to determine which computers lack Microsoft Office and subsequently install Microsoft Office on these computers, using the Task **Invoke Windows Installer**.

**Note**

If you schedule a recurring Job with this Task and view the results of this Job, the **Job History** tab in the detailed results window will display changes as they occur when comparing the query results with previous query results. Depending on the configuration of the query, this allows you to track changes in installed applications, printer drivers, services, shares, etc.

Job Postpone

Use the Task **Postpone Job** to postpone the execution of a Job. This temporary delay can be determined by the Console user on the Agent that executes the Job and/or by a pre-defined time period.

- Allowing Console users to decide whether the execution of a Job should continue can be useful if it is more convenient that the Job is executed at a later stage, for example if the bandwidth at the user's location is not sufficient to install certain software or if the Console user is presenting a company-wide presentation. In this scenario, the Task **Postpone Job** should always be the first Task that is executed in a Job.

- Specifying how long an Agent should wait before continuing with the Job can be useful if a RES ONE Automation Task fails because a previous Task has not yet been completed. For example, when a dcpromo with answerfile takes a long time before it is finished, RES ONE Automation may start its succeeding Task too early. Because dcpromo reboots the system when it has finished, this causes the succeeding Task that has already started to fail. In this scenario, schedule the Task **Postpone Job** in between these two Tasks.

**Configuration**

- When selecting **Let console user continue Job**, a popup window will be shown in the session of Console users. They can use this window to continue the Job immediately, if necessary.
  - Select **Allow console user to cancel Job** to allow Console users to cancel the Job. This will abort the active Job.
  - Select **Skip Task if no console user is logged on** to skip the Task **Postpone Job** and execute the remaining Tasks in the Job immediately if no console users are logged on to the Agent.
  - Select **Automatically start execution after** to set a time limit after which the Job will be executed. This will execute the Job after the set time limit, and ignores any postpone actions of the Console user.
LANDesk (Distribute Software)

Use the Task **Distribute Software with LANDesk** to invoke LANDesk software distribution tasks on Agents on which a LANDesk agent is running. LANDesk allows you to distribute, configure and manage software across virtual, distributed and mobile environments.

You can also use this Task to delegate control over the distribution of LANDesk software packages, for example in scenarios in which help desk employees are allowed to distribute packages, but are not allowed to access the LANDesk Management Console itself.

RES ONE Automation supports integration with LANDesk 9.5.

**Configuration**

- Use the **MBSDK Web Service** field to specify the MBSDK Web Service that should be used. If the computer on which you are running the Console also has a LANDesk agent installed, RES ONE Automation will automatically detect the MBSDK Web Service in your network.
- Use the **Security context** field to specify the required credentials to access the MBSDK Web Service. It is possible to use a global Variable or a Module parameter in this field. However, it is not possible to use a global Variable as a value for the parameter, or a parameter as a value for the parameter.
- Use the **Distribution package** field to select the software package that should be distributed to the target computer. This package contains the files and settings necessary to install a specific software package.
- Use the **Push delivery method** field to specify how the distribution package will be delivered to the Agent. This list will be populated with the delivery methods available on the LANDesk server.
- Choose whether RES ONE Automation should **Wait for the software distribution to finish** before continuing with a subsequent Task or Job on the Agent.

**Note**

- This Task requires a LANDesk agent to be installed on the target computer.
- For more information, see the LANDesk documentation.

Local Group (Create, Manage, Delete, Query)

- Use the Tasks **Create**, **Manage** and **Delete Local Group** to create, manage and delete local groups on Agents.
- Use the Task **Query Local Group** to view which local groups exist on Agents.
- When specifying users and/or groups, use the format **Domain\User**. User Principal Names (user@domain) are not supported in RES ONE Automation.

**Tip**

Combine the Tasks **Create Local Group** and **Manage Local Group** with the Task **Query Local Group** to facilitate the maintenance of the Local Groups in your environment. For example, based on the query results, you can create a local group, or change the properties of an existing group.
Local User (Create, Manage, Delete, Query)

- Use the Tasks Create Local User, Manage Local User and Delete Local User to create, manage and delete local users on Agents. For example, use the Task Manage Local User with the option Change local user properties to change the local administrator password on multiple computers periodically.
- Use the Task Query Local User to view which local users exist on Agents.
- When specifying users, use the format Domain\User. User Principal Names (user@domain) are not supported in RES ONE Automation.

Tip
It can be useful to combine the Tasks Create Local User, Manage Local User and Delete Local User with a Task Query Local User, to obtain an overview of the details of local users that exist on Agents. For example, this allows you to change the properties of a local user or remove a local user from a local group, based on the query results.

Mac OS X Automator (Invoke)

Use the Task Invoke Mac OS X Automator Workflow to invoke an Automator workflow on Agents running on Apple Mac OS X.

With Automator, you can create custom workflows and deliver automation in Apple Mac OS X to customers without the necessity of writing any scripts, routines, or computer code.

- The Automator workflow must first be added as a Resource before you can select it in the Workflow field.
- When the Automator workflow has been selected, an overview of all editable variables relating to this workflow will be shown. When editing variables, you can use placeholders to specify their value.
- When selecting Enable logging, the Job history will include a log of the Task and of the Automator output.
- If more than one user is logged on to the same Apple Mac OS X machine, the specified Automator workflow will be executed for the user who logged on first.

Known limitations
- At least one user needs to be logged on. If no users are logged on, the Task will fail.
- If more than one user is logged on to the same Apple Mac OS X machine, the specified Automator workflow will be executed for the user who logged on first.
- When specifying values for variables, any paths that you specify must be absolute, not relative.

Best Practice
1. Create the workflow on an Apple Mac OS X machine.
2. Save the workflow to a network location.
3. Because workflows are always saved as document.workflow, change the name of the workflow to a more descriptive one. This is useful to differentiate between the various workflow files.
4. Create a Resource in RES ONE Automation with the workflow file.
Mac OS X Command (Execute)

Use the Task **Execute Mac OS X Command** to run commands on Agents running on Apple Mac OS X. This allows you to schedule any action that can be performed by a command line. For example, you can use this Task to execute scripts.

**Configuration**
- Scripts that you run on a RES ONE Automation Agent for Apple Mac OS X using the Task **Execute Apple Mac OS X Command** cannot include reboot and shutdown commands. Use the Tasks **Reboot Mac OS X Computer** or **Shutdown Mac OS X Computer** instead.

**Settings tab**
- Use the option **Insert resource link** from the context menu in the **Command line** box to specify a Resource. When inserting a Resource link, the GUID of the selected Resource will be used in the command line. The Resource will be added to the command line box as $Workspace{<resource GUID>}. When the Task is executed, the Resource will be downloaded to the Agent's cache automatically and will be called from that location.
- The option **Set parameter with exit code** allows you to place an exit code into a parameter, for use in another Task in the Module. For example, suppose you have configured a Module with this Task and another Task that installs certain software. By using the option **Set parameter with exit code** in the first Task, you can use this parameter in a condition in the Task that installs software, to determine whether it should be executed.
- The option **Set parameter with standard output** allows you to place the standard output into a parameter, for use in another Task in the Module (for example, a Query).
- In the field **Timeout command execution after**, you can specify a maximum number of 9999 minutes (about 166 hours and 40 minutes). You can use parameters, functions and variables.
  - In certain situations, Agents can continue to execute the remaining commands in the script when the timeout expired. You can prevent situations like these by selecting the option **Terminate process when timeout expires**.

**Script tab**
- Scripts can be typed in directly.
- Use the **File extension of script** field to specify the file extension that RES ONE Automation should use. This extension is used to save the script as a script file in the specified format when the Task is executed. RES ONE Automation will save this file in the temporary folder on the Agent that executes the Task.
- Use the **Open in editor** button to open an external editor that is associated with the specified file extension in the **File extension of script** field. This makes it easier to create complex scripts and troubleshoot existing ones. After creating or editing the script, it will be copied from the external editor to the **Script** tab. If no associated editor can be found, the script will be opened in Notepad. For more information on how to configure file associations, see [http://msdn.microsoft.com/en-us/library/windows/desktop/cc144175(v=vs.85).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/cc144175%28v=vs.85%29.aspx).
- Log and error files are included in the detailed Job results.

**Notes**
- The **Scripts** tab has a limit of 64KB. If you use an external editor to edit a script, RES ONE Automation will disregard any characters that exceed this limit.
- If the use of parameters, functions and/or variables in the field **Timeout command execution after** results in non-numeric values (text) when the Task is executed, the timeout will use a fall back value of 1 minute.
Mac OS X Computer (Reboot, Shutdown, Inventory, Uptime)

- Use the Tasks **Reboot Mac OS X Computer** and **Shutdown Mac OS X Computer** to reboot or shut down Agents running on Apple Mac OS X. These Tasks can also be useful after the execution of a Task **Execute Mac OS X Command**. See **Mac OS X Command (Execute)** (on page 122).
- Use the Task **Inventory Mac OS X Computer** to gather information about the hardware and operating system that is installed on Apple Mac OS X computers. For example, this allows you to determine which Agents for Apple Mac OS X already have a DVD/CD drive and on which Agents you need to install one.
- Use the Task **Uptime Mac OS X Computer** to query the boot time and uptime of Agents running on Apple Mac OS X. This is useful to determine whether you are dealing with a memory leak. Based on the results of this Query, you can decide whether an Agent should be rebooted. You can automate this sequence of events by combining the Task with a Task **Reboot Mac OS X Computer**.

**Configuration**

- In the Tasks **Reboot Mac OS X Computer**, the option **Reboot computer after shutdown** is enabled by default.
- In the Tasks **Shutdown Mac OS X Computer**, the option **Reboot computer after shutdown** is disabled by default.
- With the option **Wait for Agent to come back online**, you can specify whether a Job should wait after a reboot with continuing with the next Task in the Job until the Agent has come back online. This option is enabled by default. It can be useful to disable the option in scenarios in which the computer on which the Agent is running switches to a different OS after a reboot and a different Agent comes online. This prevents RES ONE Automation from waiting until the original Agent comes back online.

Mac OS X Parameters (Query)

Use the Task **Query Mac OS X Parameters** to view the values that Module parameters had when a Job was executed on one or more Mac OS X Agents. This Task is particularly useful in a Module where the value of one or more parameters is set depending on conditions or evaluators.

The results of this Task are presented per Agent and can be sorted per parameter. This Task is also available for **Windows** (see page 130) and **Unix/Linux** (see page 146) Agents.

**Configuration**

- If no parameters are shown, this means that no Module parameters are available (yet). When no Module parameters are available, the **Evaluators** tab is grayed out.
- By default, all Module parameters are reported in this Query. To exclude specific Module parameters from the Query results, clear **Show all parameters in Query summary** and then clear specific check boxes.

**Note**

You can only use this Task for Module parameters. If the Module is included in a Project with Project parameters, the Project parameters will overrule the values of the Module parameters.
Chapter 5: Library

Mac OS X Resource (Download)

Use the Task **Download Mac OS X Resource** to download Resources to Agents running on Apple Mac OS X.

**Configuration**

When you select the option **Enable logging**, the detailed Job history will include a log.

<table>
<thead>
<tr>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>It is only possible to download Resources that are stored in the Datastore. Resources stored on a fileshare or RES ONE Automation Resource Packages are not supported.</td>
</tr>
</tbody>
</table>

Mac OS X Disk Space (Query)

Use the Task **Query Mac OS X Disk Space** to check the available disk space and folder space usage on Agents running on Apple Mac OS X. If you select to query a specific folder, this allows you to check the amount of space it uses and the size of the files it contains.

For example, you can use this Task to check the available disk space on a server at regular intervals, in order to prevent it from running out of disk space. You can also use the Task to check the size of the folder in which an application has been installed. You can set the Task **Invoke Windows Installer** to install the application automatically, depending on the outcome of the Query.

**Configuration**

- When querying on used folder space, you can specify the Start folder by clicking , by specifying the path to the folder manually or by using parameters and/or functions. You can specify parameters and functions manually or by right-clicking the Start folder field and clicking **Insert Parameters** or **Insert Functions**. See **Parameters** (on page 257) and **Functions** (on page 246).
- When setting an evaluator based on disk, you can use environment variables, parameters and functions to specify the disk. This is useful because mapped drive letters may vary per Agent. For example, if you use the environment variables `%systemdrive%`, the Agent that executes the Task will decide what drive to use.
- When viewing the Job results of a Task **Query Disk Space**, select **Show in KB** to view the Job results of all Agents in KB. When a Task **Query Disk Space** is executed, the amount of free disk space of Agents can be reported in GB, in MB or in KB, depending on the amount of disk space that is available on each Agent. This makes it difficult to compare the Job results of multiple Agents when they are exported to a CSV file. The option **Show in KB** allows for an easy comparison in a CSV file.

<table>
<thead>
<tr>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>If you schedule a recurring Job with this Task and view the results of this Job, the <strong>Job History</strong> tab in the detailed results window will display changes as they occur when comparing the query results with previous query results. Depending on the configuration of the query, this allows you to track changes in installed applications, printer drivers, services, shares, etc.</td>
</tr>
</tbody>
</table>
Chapter 5: Library

Mac OS X Installed Programs (Query)

Use the Task Query Installed Programs to obtain an overview of applications that have been installed on Mac OS X Agents and check where/whether these applications have been installed. For example, this allows you to determine which computers lack Microsoft Office and subsequently install Microsoft Office on these computers.

Note

If you schedule a recurring Job with this Task and view the results of this Job, the Job History tab in the detailed results window will display changes as they occur when comparing the query results with previous query results. Depending on the configuration of the query, this allows you to track changes in installed applications, printer drivers, services, shares, etc.

Mac OS X Software Package (Install, Remove)

• Use the Task Install Mac OS X Software Package to install disk images (DMG file) and installer packages (PKG file) on Mac OS X Agents.
• Use the Task Remove Mac OS X Software Package to remove installed programs (APP file) from Mac OS X Agents.

Configuration

• Settings tab of Task Install Mac OS X Software Package:
  • The Installation options area is unavailable when installing packages of type *.DMG. This is the default setting.
    • DMG files may only contain one application (*.APP), which is always copied to /Applications.
  • Use the Installation target field to specify the installation location of the package. The default target is "/". Other options are for instance /Applications/Utilities or /Volumes/OtherDrive.
    • PKG packages determine which target(s) are allowed. Please refer to the documentation of the installer package for a list of all supported targets.
  • Use the Package Parameters field to specify any additional parameters required for the installation. Please refer to the documentation of the installer package for a list of all supported parameters.

• Settings tab of Task Remove Mac OS X Software Package:
  • Use the Application path field to specify the complete path to the application that needs to be removed (for instance /Applications/iPhoto.app).
  • You can only remove applications that are not active. Select Terminate application if running to forcibly close the application if it is running.

Notes

• When specifying installation targets, paths, etc., please note that all Mac OS X Tasks are case-sensitive.
• Regular expressions are supported.

Tip

To retrieve which targets are allowed for a PKG package, apply -volinfo to the package in a command line, while working on a Mac OS X machine.
**Message Box (Show)**

Use the Task *Show Message Box* to create a message that will be displayed on Agents. This is useful, for example, if you want to notify your users about an Exchange server that is not working, the duration of a maintenance operation or a new application that has become available.

**Microsoft App-V Client (Invoke)**

Use the Task *Invoke Microsoft App-V Client* to preload and publish Microsoft App-V application caches to Agents.

RES ONE Automation supports Microsoft Application Virtualization (App-V) 4.5 and 4.6.

**Configuration**

- This Task requires credentials that have permissions to modify the user profiles directory. (The RES ONE Automation Agent service typically runs under the Local System account, which has no privileges to access the user profiles directory.)
- With the option *Clear application cache*, all preloaded applications will be deleted from the application cache, and the published applications will also be cleared from the **Start** menu and **Desktop**.

**Tip**

It can be useful to combine this Task with a Task *Query Microsoft App-V Client*, to obtain an overview of all preloaded applications on the Agents. For example, this allows you to preload a missing Microsoft App-V application or remove the application cache, based on the query results.

**Microsoft App-V Client (Query)**

Use the Task *Query Microsoft App-V Clients* to obtain an overview of the Microsoft App-V applications that are loaded on Agents. This allows you to create a report of the cached applications and the log files that are used by Microsoft App-V.

For example, you can check whether Adobe Acrobat is preloaded on all Agents. If not, you can configure a Task *Invoke Microsoft App-V Client* to preload the application based on the Query results.

RES ONE Automation supports Microsoft Application Virtualization (App-V) 4.5 and 4.6.

**Configuration**

- To query the cached applications of a specific user on an Agent, specify a user account in the **Security context** field.
- To query all cached applications on an Agent, leave the **Security context** field blank.

**Notes**

- Only Agents on which SoftGrid Client 3.1 or higher is installed can perform this Task. For more information about Microsoft App-V, see the Microsoft App-V website.
- If you schedule a recurring Job with this Task and view the results of this Job, the History tab in the detailed results window will display changes as they occur when comparing the query results with previous query results. Depending on the configuration of the query, this allows you to track changes in installed applications, printer drivers, services, shares, etc.
Microsoft System Center Configuration Manager (Distribute Software)

Use the Task Distribute Software with Microsoft System Center Configuration Manager to deploy software distribution packages on Agents on which a Microsoft System Center Configuration Manager client is running. Using packages, you can deliver, install, and configure software to meet the individual needs of your clients.

You can also use this Task to delegate control over the deployment of Configuration Manager packages, for example in scenarios in which help desk employees are allowed to deploy packages, but are not allowed to access the Configuration Manager Console itself.

Configuration

- Use the **Management Server** field to specify the Configuration Manager server that should be used. If the computer on which you are running the Console also has a Configuration Manager client installed, RES ONE Automation will automatically detect the Configuration Manager server in your network.
- The option **Autodetect management server on client** is selected by default. When selected, the Agent will automatically use the configured Configuration Manager server for that client in your network. Clear this option if you want to override the client-configured management server with the value of the **Management Server** field.
- Use the **Security context** field to specify the required credentials to access the Configuration Manager server. At least an administrator account with the "Application Administrator role" is needed. It is possible to use a global Variable or a Module parameter in this field. However, it is not possible to use a global Variable as a value for the parameter, or a parameter as a value for the parameter.
- Use the **Version** field to specify the Configuration Manager. When access is possible, RES ONE Automation automatically detects the version of the server (pre-2012 or 2012 and higher). Although it is possible to specify a different version than the one that was automatically detected, this is not recommended.
- Use the **Deployment** field to select whether to deploy Packages or Applications. This option will only be active when **Version** is set to Microsoft System Center Configuration Manager. It will be grayed out and set to Package when **Version** is set to Microsoft System Center Configuration Manager (pre 2012), because deployment of applications is not supported on those versions.
  - **Packages**: Use the **Package** field to select the software distribution package that should be advertised to the target computer.
  - Use the **Program** field to select the command that tells the client what action to execute once the package is received. For example, install or uninstall software, perform upgrades, or apply hotfixes.
  - **Applications**: Use the **Application** field to select the application that should be deployed on the target computer.
  - Use the **Action** field to select the action that should be performed by the client once the application is received.
• Use the **Distribute to** field to select the collection of clients to which the software distribution package or application should be made available.
• Select **Distribute to local Agent only (faster)** to make the software distribution package or application only available to the Agent that executes the Task. If a collection for this Agent does not yet exist, RES ONE Automation will create it automatically.
  • Choose whether RES ONE Automation should **Wait for the software distribution to finish** before continuing with a subsequent Task or Job on the Agent, including:
  • the option **Timeout for checking the "SMS Agent Host" service to be started** to delay the start of the "SMS Agent Host" service. By default, the timeout period is set to 5 minutes and can be changed to a number in the range of 1 - 99 minutes. When the "SMS Host Agent" service is started within the defined timeout period, the Task will continue. Otherwise the Task will fail.
  • If **Distribute to local Agent only (faster)** is selected for deployment of an application, the application will not be found when executing a Task **Query Microsoft Configuration Manager Client**.

---

### Notes
- This Task requires Microsoft System Center Configuration Manager client to be installed on the target computer.
- Do not execute this Task on more than one RES ONE Automation Agent unless the option **Distribute to local Agent only (faster)** has been selected.
- For supported versions of Microsoft System Center Configuration Manager, see the **Compatibility Matrix RES ONE Automation** ("Compatibility Matrix" on page 316).
- For more information about Microsoft System Center Configuration Manager, see the Microsoft documentation.

---

**Microsoft System Center Configuration Manager (Query Client)**

Use the Task **Query Microsoft Configuration Manager Client** to create status reports of Microsoft System Center Configuration Manager clients or to obtain an overview of the software distribution packages that were advertised to them.

**Configuration**

• Select **Query Client Status** to create status reports of Configuration Manager clients, including client version, cache options, service states, etc.
• Select **Query Client Advertised Packages** to obtain an overview of all software distribution packages that have ever been advertised to the target computer, including local status of the advertisement, last run date and exit code.
• Select **Query Deployed Applications** to obtain an overview of all applications that have ever been deployed to the target computer, including software version, install time and install state. Applications that have been deployed with a Task **Distribute Software with Microsoft System Center Configuration Manager** in which the option **Distribute to local Agent only (faster)** was selected, will not be shown in the Query results.

---

### Notes
- This Task requires Microsoft System Center Configuration Manager client to be installed on the target computer.
- For more information, see the Microsoft documentation.
**Microsoft Update (Install)**

Use the Task **Install Microsoft Update** to install Microsoft Hotfixes and Service Packs on Agents.

**Configuration**
- The relevant MSI or MSU file must be available as a Resource.

**Notes**
- The Agent may need to reboot after installing the update. You can schedule this reboot by including the Task **Shutdown Computer** after the Task **Install Microsoft Update**.
- Create a separate Task for each update that you want to install.

**Tip**

It can be useful to base this Task on the results of the Task **Query Installed Programs**, which obtains an overview of all installed programs on Agents. For example, this allows you to install Microsoft Office 2003 Service Pack 1 on specific Agents, based on the results of the Query.

**Microsoft Windows Product Key (Set, Query)**

- Use the Task **Set Microsoft Windows Product Key** to change the Microsoft Windows product key. This is useful if your license agreement(s) with Microsoft have expired or changed.
- Use the Task **Query Microsoft Product Keys** to gather information about Microsoft Windows or Office product keys. You can use this information, for example, to verify that you use genuine copies of Windows in your RES ONE Automation environment; or to check which product keys are used by which Agents.

Only genuine Microsoft software is fully supported by Microsoft and helps safeguard your IT environment against for example spyware and malware. Read more about this on the Microsoft site.

**Notes**

Microsoft Windows Genuine Advantage Validation can only be performed on Agents running on Windows XP or Windows Vista and requires the installation of the Windows Genuine Advantage Validation Tool (LegitCheckControl.cab).

**Tip**

It can be useful to combine a Task **Set Microsoft Product Key** with a Task **Query Microsoft Product Keys**, to obtain an overview of all current product keys and to change these keys, depending on the query results.
Parameters (Query)

Use the Task Query Parameters to view the values that Module parameters had when a Job was executed on one or more Agents. This Task is particularly useful in a Module where the value of one or more parameters is set depending on conditions or evaluators.

The results of this Task are presented per Agent and can be sorted per parameter. This Task is also available for Unix/Linux (see page 146) and Mac OS X (see page 124) Agents.

Configuration

- If no parameters are shown, this means that no Module parameters are available (yet). When no Module parameters are available, the Evaluators tab is grayed out.
- By default, all Module parameters are reported in this Query. To exclude specific Module parameters from the Query results, clear Show all parameters in Query summary and then clear specific check boxes.

Note

You can only use this Task for Module parameters. If the Module is included in a Project with Project parameters, the Project parameters will overrule the values of the Module parameters.

Printer (Add, Remove, Query)

Use the Tasks Add Printer, Remove Printer and Query Printer to add or remove printers on Agents.

- With Add Printer, you can add printers and, if necessary, install their associated printer drivers on Agents.
- With Remove Printer, you can remove these printers from Agents.
- With Query Printer, you can query the printers that are installed on Agents. You can filter by printer name and driver name.

Configuration

- When adding a local printer, the list of port names corresponds to the port names that are known on the computer on which you are running the Console.
- When adding a network printer, it is obligatory to specify a TCP/IP port that is installed on the Agent. If it does not yet exist, a standard TCP/IP port will be installed by RES ONE Automation when the Task is executed. For the Agent+, IPv4 and IPv6 addresses are supported. For the legacy Agent, only IPv4 is supported.

Notes

- A standard TCP/IP Port that is installed by RES ONE Automation uses the RAW printing protocol.
- A Task Remove Printer only removes printers from Agents; it does not remove printer drivers or ports.
- RES ONE Automation can only install printer drivers that are digitally signed by an approved certification authority and only on Agents running on Microsoft Windows 7 or higher.

Tip

You can set the Tasks Add Printer and Remove Printer to install printers and printer drivers or remove these printers, depending on the query results.
Chapter 5: Library

Printer Driver (Add, Remove, Query)

- Use the Tasks Add Printer Driver and Remove Printer Driver to add and remove printer drivers on Agents.
- Use the Task Query Printer Driver to query the printer drivers that are installed on Agents. You can filter by driver name, manufacturer name and printer driver provider.

Note
RES ONE Automation can only install printer drivers that are digitally signed by an approved certification authority and only on Agents running on Microsoft Windows 7 or higher.

Tip
You can set the Tasks Add and Remove Printer Driver to install or remove printer drivers, depending on the query results.

Registry Settings (Apply)

Use the Task Apply Registry Settings to change settings in the system registries of Agent computers.

Warning
Careless registry editing can cause irreversible damage! Approach this task with caution, and only after you have made a backup.

Configuration

- In all fields where the name of a registry key or value needs to be provided, you can copy the name of the relevant key or value from the local computer or from another Agent. This is less error prone than typing it in manually. To copy a key name or value, right-click the relevant (sub)tree and choose Open [key name] to open the window Pick keys/values from registry. Here, you can add keys or values to the Task.
  - To add a key with its value, select the (default) name, and click Add.
  - To add a key without its value, select the key in the browse tree and click Add.
- In the same window you can also click Connect to browse the registry of another Agent and copy the information from there. To type the name in manually instead of copying it, right-click the (sub)tree in the window Apply Registry Settings and choose New. With either method, ensure that you add not only the key, but also any relevant values.
- To delete a registry key or value, add it to the Task, then right-click and choose Toggle "remove this key and subkeys" or Toggle "remove this value". Keys and values toggled for deletion are marked with ☞. If Delete is selected, the registry key will only be deleted from the settings of the Task, and not from the registry of the Agent when the Task is executed. You can also use Toggle "remove this value" to reset default registry keys to their default value (value not set).
- If the ☞ icon is not shown on the key or value, it will not be deleted.
- By default, when a key is deleted from the registry, all its subkeys and values are deleted too, even if they are not shown in the Task window. To set exceptions for a specific key or value, add it to the Task, then right-click it and choose Toggle "remove this key and subkeys" or Toggle "remove this value".
- When importing registry files, Microsoft Windows interprets specific parts differently than RES ONE Automation. This can lead to unexpected results if you use RES ONE Automation to export
files that contain deletion exceptions, and then import the file into an external registry server. For predictable results, always use RES ONE Automation to export and import registry settings.

• When importing registry files, choose how the merge should handle keys that exist in both the original registry file and in the file being imported:
  • with Replace existing data, the new Registry file overwrites the old one completely.
  • with Perform incremental merge with existing data, new keys from the imported file are added, but existing keys are left untouched.
  • with Perform differential merge with existing data, only keys that exist in both files are kept. If both files contain the same key with the same value, but with different value data, then the value data from the imported file overwrites the value data of the original.
• To use pattern matching on the name of a registry key, add it to the Task, then right-click and choose Toggle "use pattern matching on keyname". This allows you to perform specific actions with the key, such as adding keys and values to other keys, or removing these keys and values.

**Note**

If you use wildcard characters in the name of registry keys, and do not use the option Toggle "use pattern matching on keyname", the name of the registry key will be interpreted literally. For example, the asterisk (*) will be just an asterisk, not a wildcard character.

**Tip**

• Module parameters can replace the names of registry subkeys and values in the Task. Right-click the relevant key or value and choose Replace Key Name with parameter or Replace Value Name with parameter.
• To rename a key, add it to the Task and rename it, so that a duplicate key will be created. If this has been successful, execute another Task in which the original key is deleted.

### Performing Actions using Pattern Matching on Registry Key Names

**How to use pattern matching in the names of registry keys**

1. Add a new key in the relevant Registry location, and use the relevant wildcard(s) in the name of the key.
2. Right-click the key and choose Toggle "use pattern matching on keyname". The selected key is marked with a blue star to indicate that the name will be interpreted according to the pattern matching rules.

**Example 1**

Suppose you want to delete all references to locally cached copies of user profiles on a number of Agents. These profiles all start with S-1-5-21 and are all located in HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList.

1. Right-click HKEY_LOCAL_MACHINE and click Open HKEY_LOCAL_MACHINE. This opens the window Pick keys/values from registry.
2. Browse to the key HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList and click Add.
3. Click OK to confirm that the selected key has been added to the Task.
4. Click Close to close the window Pick keys/values from registry.
5. In the window Apply Registry Settings, right-click the key ProfileList, choose New > Key.
6. Type S-1-5-21* as the name of the registry key. Click elsewhere to fix the new name.
7. Right-click this key and choose Toggle "use pattern matching on keyname". The selected key is marked with a blue star.
8. Right-click the same key and choose Toggle “remove this key and all subkeys”. The selected key is marked for removal.

As a result, when you schedule a Job with this Task on a set of Agents, RES ONE Automation will remove all keys that match the pattern and that appear in the registry key
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList, including its subkeys and values.

Example 2

Suppose you want to set a value for Adobe Acrobat Professional, but there are several versions of that suite in use. By using pattern matching on the name of the registry key name, you can create a registry key that matches all versions. For example, when you create a registry key HKEY_LOCAL_MACHINE\SOFTWARE\Adobe\Adobe Acrobat\*.\Language, this key matches both HKEY_LOCAL_MACHINE\SOFTWARE\Adobe\Adobe Acrobat\10.0\Language and HKEY_LOCAL_MACHINE\SOFTWARE\Adobe\Adobe Acrobat\11.0\Language.

Registry Settings (Query)

Use the Task Query Registry Settings to compare and report the registry settings on different Agents. For example, you can check whether a Microsoft App-V Client is run at startup by querying the registry key HKLM/software/Microsoft/Windows/current/version/run.

Configuration

In the field Key, you can copy the name of the relevant key or value from the local computer or from another Agent. This is less error prone than typing it in manually. To copy a key name, click ... From the window Add/Change Registry to query, click Connect to browse the registry of another Agent and copy the information from there.

Remote Terminal Server Logons (Change, Query)

- Use the Task Change Remote Terminal Server Logons to enable or disable remote connections to a terminal server. This can be useful if you need to perform maintenance work.
- Use the Task Query Remote Terminal Server Logons to check whether a terminal server accepts remote connections.

Tip

It can be useful to combine a Task Change Remote Terminal Server Logons with a Task Query Remote Terminal Server Logons, to check whether a terminal server accepts remote logons and change this setting, based on the query results.
RES ONE Workspace (Refresh)

Use the Task Refresh RES ONE Workspace to force a refresh of each active user workspace on the target computer. Multiple sessions on the same computer (for example a terminal server) will be refreshed at a 10-second interval. This Task is useful, for example, to refresh user workspaces on laptops or to notify users in a user workspace that new applications are available.

Note

This Task will always complete. If RES ONE Workspace is not installed on the target computer, this will be displayed in the log.

RES ONE Automation Component (Deploy, Repair, Remove, Update)

- Use the Tasks Deploy, Repair and Remove RES ONE Automation Component to use an Agent as a proxy to deploy, repair or remove Consoles, Dispatchers and Agents.
- Use the Task Update RES ONE Automation Agent to force Agents to update to the latest version. If a new version of RES ONE Automation is available for the Agent, a Dispatcher will download all necessary data from the Datastore and store it. In its turn, the Agent will download all necessary data from the Dispatcher and update to the latest version. This may lead to issues in large environments with many Agents that all update at the same time. This behavior can be managed with the global setting Auto Update Agents. By executing a Task Update RES ONE Automation Agent, you can force Agents to update to the latest version, irrespective of this global setting.

Configuration

- Tasks Deploy, Repair and Remove RES ONE Automation Component:
  - As target computer, you can also type the host name, the fully-qualified domain name (FQDN) or the IP (IPv4 or IPv6) address of the target computer(s). The Scan Computers function does not support IPv6 address ranges. For legacy Agents, IPv6 is not supported.
  - These Tasks require administrator credentials to connect to the target computer(s).

Notes

- When repairing, removing or updating an Agent, the Agent that executes the Task must be different Agent than the target Agent: if the Task would be executed by an Agent that is also target computer, the Agent that executes the Task will also go offline and the Job will never finish.
- When repairing or removing a Dispatcher, your RES ONE Automation environment needs to contain at least two Dispatchers: one that is repaired or removed and one that remains. During the execution of the Task, the Dispatcher service will be stopped. If the Dispatcher is the only Dispatcher in your RES ONE Automation environment, this means that the Agent that executes the Task will also go offline and the Job will never finish.
- It is not possible to deploy, repair or remove Agents+ (see page 202), Agents for Unix/Linux (see page 203) or Agents for Mac OS X (see page 204) with this Task.
RES ONE Automation Dispatchers (Discover)

Use the Task Discover RES ONE Automation Dispatchers to check which Dispatchers are available to Agents.

This allows you to query which Dispatchers are available to an Agent with the current Agent settings, but also to query which Dispatchers are available to an Agent if you were to change the Agent settings. This allows you to test new settings before implementing them.

Configuration

The settings Dispatcher discovery and Dispatcher locations specify how and where an Agent should search for a Dispatcher.

- **Autodetect** and **Retrieve Complete Dispatcher Address List After Discovery** lets an Agent search for available Dispatchers in its own LAN by sending a multicast signal. The discovered Dispatchers will tell the Agent which other Dispatchers are known in the Datastore. In this way, the Agent will be aware of all Dispatcher locations. If connection is possible, the Agent will randomly connect to these Dispatchers to retrieve information.

- **Autodetect** and **Only Use Discovered Dispatchers** lets an Agent search for available Dispatchers in its own LAN by sending a multicast signal. The Agent will only communicate with the Dispatchers it can find with the multicast signal. The Agent randomly uses the available Dispatchers to retrieve information.

- **Use Dispatcher Address List and Retrieve Complete Dispatcher Address List After Discovery** lets an Agent only search for Dispatchers specified in the Dispatcher address list. Best practice is to select **First try autodetect** to autodetect Dispatchers. If this fails, the specified Dispatchers will be used. This setting is particularly useful for laptops: When a laptop logs on at the office, the Agent will automatically detect a Dispatcher. When a laptop is used at a different location, the Dispatcher address list will be used. If connection is possible, the Agent will randomly connect to these Dispatchers and all other Dispatchers that are retrieved.

- **Use Dispatcher Address List and Only Use Discovered Dispatchers** lets an Agent only search for the Dispatchers that you specified in the Dispatcher address list. Specify at least one Dispatcher in the list. Select the **First try autodetect** check box to autodetect Dispatchers first. If this fails, the specified Dispatchers will be used. If connection is possible, the Agent will randomly connect to these Dispatchers.

---

**Note**

- To detect Dispatchers, an Agent uses multicast 224.1.1.50 to port 3163. If Agents should only communicate with Dispatchers in their own subnet within a WAN, use **Dispatcher Discovery: Autodetect** for each subnet. This allows you to add Dispatchers to particular sites without having to change configuration settings.

- For the communication between Agent+ and Dispatcher, IPv4 and IPv6 are supported. For the communication between legacy Agent and Dispatcher, only IPv4 is supported.
RES ONE Automation Results (Delete, Export)

Use the RES ONE Automation Results Tasks to delete or to export Job results from the Datastore automatically.

- With Delete RES ONE Automation Results, you can clean up the Datastore from Job results that are no longer relevant, for example because they have become obsolete through Snapshot Intelligence or because they are the results of recurring Jobs.
- With Export RES ONE Automation Results, you can export Job results to XML files. This allows you to backup Job results for review purposes. This Task is especially useful in combination with the Task Delete RES ONE Automation Results, in which you first back up certain Job results by exporting them to zipped XML files, and then clean up the Datastore by deleting them.

In these Tasks, you can delete or export Job results from the Datastore, based on a combination of criteria:

- The type of Job that was performed (recurring/non-recurring)
- The relevance of the Job results (actual Job results/Job results that have become obsolete through Snapshot Intelligence)
- The Agent(s)/Team(s) that executed the Job
- The period in which the Jobs were executed

Configuration

- When specifying Job type, Recurring includes Jobs that were scheduled as Recurring, Schedule using Cron and After every boot; Non-recurring includes Jobs that were executed only once.
- When specifying Agents:
  - select Agent(s) to include the Job results of the specified Agent(s) only. This includes Jobs that were scheduled on Teams of which the Agent(s) are a member.
  - select All Agents to include the Job results of all Agent(s).
  - select Team Jobs to include the results of Jobs that were scheduled on the specified Team(s) only.
  - select Team members to include the results of Jobs that were scheduled on Agents that are member of the specified Team(s) only. Team members are resolved when the Task Delete RES ONE Automation Results or Export RES ONE Automation Results is executed.
- When specifying the Period, Local time means the local time on the Agent that executes the Job.
- In the Task Export RES ONE Automation Results:
  - With Export overview results, you can create an XML file with a general overview of the Job results.
  - With Export detailed results, you can create an XML file with a detailed overview of the Job results. If you also select Export overview results, RES ONE Automation will create one XML file that first gives a general overview of the Job results and then a detailed overview.
  - With Export results per Agent, you can create a separate XML file for each Agent that executed the Job, including the name of the Agent. This is useful when you want to keep a separated administration for each Agent.
  - Job results are exported to XML files, after which they are saved in a ZIP file. The number of XML files that are created depends on the scope criteria.
  - By default, the ZIP file of the Job results is saved on the local hard drive of the Agent that executes the Job.
    - When using environment variables in the save location, it is not recommended to use %userprofile%, because Agents generally use the local service account to execute Jobs.
    - You can verify whether the desired Job results are exported by comparing the XML files with the Job results in the Job History node or on the Job History tab of the specified Agent(s) or Team(s).
• When a Task Delete RES ONE Automation Results has been executed, its results show how many Job results were deleted and how many Agent Job results: if a Job was scheduled on multiple Agents and you delete the Job results of one of these Agents, this does not delete the Job results for the entire Job, but only the Job results for that Agent.

**Warnings**

• Execution of these Tasks may cause a heavy load on your database server, depending on the scope of the Tasks and the amount of Job results in the Datastore.

• Automatic deletion of Job results may affect change paths for Snapshot Intelligence.

**RES ONE Automation Team Membership (Change)**

Use the Task Change RES ONE Automation Team Membership to add or remove Agents to Teams. This Task is typically used in combination with a Task Query in which an evaluator determines whether the succeeding Task Change RES ONE Automation Team Membership should be executed, for example to add explicit members to rule-based Teams.

**Example**

Suppose your RES ONE Automation environment contains three Teams, a Team for machines that are a web server, a Team for machines on which Microsoft Office is installed and a Team for machines in New York. Agents should be added to these Teams, based on their properties.

Because you cannot tackle this situation by using Team rules, you need to create a Module that contains the following:

1. A Task Query Service Properties with an evaluator for a web service.
2. A Task Change RES ONE Automation Team Membership with a condition that executes the Task if the status of the previous Task is Completed. This Task adds the Agent to the Team Web Server.
3. A Task Query Installed Programs with an evaluator for Microsoft Office.
4. A Task Change RES ONE Automation Team Membership with a condition that executes the Task if the status of the previous Task is Completed. This Task adds the Agent to the Team Office.
6. A Task Change RES ONE Automation Team Membership with a condition that executes the Task if the status of the previous Task is Completed. This Task adds the Agent to the Team New York.

**Configuration**

• You can add or remove Agents to or from multiple Teams at once.

• When setting a specific Team as a primary Team, the Team option Automatically set as primary team for new members in any other Teams will be overruled.

• If the Agent is added to multiple Teams for which the Team option Automatically set as primary team for new members has been set, the last Team in the list for which the option has been set will be the primary Team.

• The button Set as primary team/Do not set as primary team is grayed out when removing Agents from a Team.

**Notes**

• If you add an Agent to a Team of which it is already a member or if you remove an Agent from a Team of which it is not a member, the Task will be ignored.

• If you remove an Agent from a Team whose membership is rule-based, the Task will fail.
Resource (Download)

Use the Task Download Resource to download Resources to Agents, for example to pre-cache large files or to send a specific version of a DLL file to Agents on which it is needed.

Configuration

- When you select the option Enable logging, the detailed Job history will include a log.
- By default, Resources are downloaded to the Agent cache, but you can specify a different destination folder for the Resource.
- You can use environment variables when specifying the destination folder.
- Resource packages will only be extracted when downloaded if a destination folder has been specified.
- If the network connection fails during execution of the Task, the download will be retried 10 times. If the network connection cannot be reestablished before the 10th attempt, the Job status will be set to "Failed Halted".

Tips

- When a Task uses a Resource, it will first check whether that Resource is available in the Agent cache. If the file is available (and it is the same version as the one stored centrally), the Task will use the cached Resource instead of downloading it from the Datastore. This makes it possible to pre-cache a Resource to the relevant Agents using the Task Download Resource, so that another Task, such as the Task Invoke Windows Installer, can use it later.
- After use, downloaded Resources that were placed in a custom location (not the Agent cache) can be removed with the Task Perform File Operations.

Secure Shell (SSH) Commands (Execute)

SSH is a network protocol that allows you to connect to a shell on remote devices via a secured connection and to execute commands on these devices. It can be used to access shell accounts on Linux and Unix systems, and on various devices, such as Ethernet switches.

The Task Execute Secure Shell (SSH) Commands allows you to build your own, custom RES ONE Automation Tasks that use Secure Shell commands. These custom Tasks can be scheduled on any Agent, who will then connect to the specified computer and execute the specified commands.

For example, if the servers in a server farm are equipped with a remote access card, you can use this Task to power down most servers in the server farm during the weekend and reboot these servers when necessary. This can save considerable costs.

Prerequisites

- Secure Shell commands can be executed on devices on which the SSH service is running.
- For this Task, .NET Framework 2.0 must be installed on the Agent running this Task. When running this Task on an Agent+, Microsoft .NET Framework 4.5.2 (Full version) is required.

Configuration

Settings tab

- You can use user/password authentication or user/certificate authentication.
- You can specify Secure Shell commands manually (on the Commands tab), but you can also use Secure Shell commands that have been stored as a Resource.
• With the option **Create script on remote host** you can run specified Shell commands as a script on a remote host, instead of executing them per command. If selected, RES ONE Automation will create a script file in the user's home directory on the remote host.

• With the option **Set parameter with standard output** you can place the standard output into a parameter, for use in another Task in the Module (for example, a Query). The standard output parameter can contain a maximum of 4MB.

• With the option **Run script as a superuser**, which can be used in combination with the option **Create script on remote host**, you can run scripts with sudo. The sudo command is a program for certain Linux and Unix operating systems that allows a permitted administrator to run commands with the security privileges of another user (normally the root). Before you select this option, please make sure the sudoers configuration file permits access to execute the sudo command to the relevant user account.

• In the field **Timeout command execution after**, you can specify a maximum number of 9999 minutes (about 166 hours and 40 minutes). You can use parameters, functions and variables.
  - If the use of parameters, functions and/or variables in the field **Timeout command execution after** results in non-numeric values (text) when the Task is executed, the timeout will use a fall back value of 1 minute.
  - In certain situations, Agents can continue to execute the remaining commands in the script when the timeout expired. You can prevent situations like these by selecting **Terminate process when timeout expires**.

**Commands tab**

• Scripts can be typed in directly. You can use RES ONE Automation functions, RES ONE Automation parameters and environment Variables in Secure Shell commands. These Variables, functions and parameters will be parsed when the Task is executed.

• Use the **File extension of script** field to specify the file extension that RES ONE Automation should use. This extension is used to save the script as a script file in the specified format when the Task is executed. RES ONE Automation will save this file in the temporary folder on the Agent that executes the Task.

• Use the **Open in editor** button to open an external editor that is associated with the specified file extension in the **File extension of script** field. This makes it easier to create complex scripts and troubleshoot existing ones. After creating or editing the script, it will be copied from the external editor to the **Script** tab. For more information on how to configure file associations, see [http://msdn.microsoft.com/en-us/library/windows/desktop/cc144175%28v=vs.85%29.aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/cc144175%28v=vs.85%29.aspx).

  • The **Commands** tab has a limit of 64KB. If you use an external editor to edit a script, RES ONE Automation will disregard any characters that exceed this limit.

• When this Task is executed, any output is shown in the detailed Job results per Agent, on a separate tab **Console Output**. This output can also be viewed using Notepad.

---

**Tip**

If you place a script on the remote host to which the Agent that executes this Task should connect, you can use a Secure Shell command to call this script.
Service Properties (Manage)

Use the Task Manage Service Parameters to manage the local service properties of Agents. You can change: the account that the service uses to log on, the startup method for the service, the service state, and the service recovery options.

Warning

Changing the default service parameters can prevent important services in Microsoft Windows from running correctly. Use extreme caution when changing the Startup Type and Log On As settings of services that are configured to start automatically.

Configuration

• For Change Service Account: if specifying an account, you can use:
  • user name including a domain name (for example demo\administrator)
  • user name without a domain (for example administrator), in order to use a local account on the target computer.
  • user name including \localhost (for example localhost\administrator), in order to use a local account on the target computer.
  • user name including computer name (for example [computer name]\administrator), in order to use a local account on the target computer.

• For Change Service Account: if the provided account does not have the rights to log on as a service, these can be added by selecting the option Add user right "Log on as a service".

• For Change Service Startup Type the following options are available:
  • Manual: the specified service can be started by an administrator, a related service, a system device driver or a user action.
  • Automatic
  • Automatic (Delayed Start): this type is available for the operating systems Microsoft Windows Vista and higher, and Microsoft Windows Server 2008 and higher. Otherwise, the Startup Type will fall back to Automatic. Not all services support the Startup Type Automatic (Delayed Start), for example, the Print Spooler service.
  • Disabled

Note

If the provided account does not have the rights to log on as a service and you did not select the option Add user right "Log on as a service", a Task to change a service account will fail with an error "Failed to start service ‘....’ (1069 - The service did not start due to a logon failure.)."

Tip

It can be useful to combine this Task with the Task Query Service Properties, to obtain an overview of the current services plus their settings on Agents. For example, this allows you to check whether Microsoft Alert Services is enabled on all computers at startup and to take action, based on the query results.

Configuring RES ONE Workspace to use Windows Authentication using a Task “Manage Service Properties”

Suppose you want to configure the RES ONE Workspace Agent Service to connect to a RES ONE Workspace Datastore on a Microsoft SQL Server using Windows Authentication. You can do this manually, by changing the settings for EACH RES ONE Workspace Agent. You can also do this with a
Chapter 5: Library

Task Manage Service Properties, as long as the computer on which a RES ONE Workspace Agent is installed is also a RES ONE Automation Agent.

1. Configure a Task Manage Service Properties.
2. Click the Settings tab and select Change Service Account.
3. Click  in the Service name field. This opens the Select Service window.
4. Select the service to which the changes should apply (RES) and click OK.
5. Select This account and enter the account and its credentials that should be used to run the RES ONE Workspace Agent Service.
6. Click OK and execute the Task on all computers on which Windows Authentication should be used.

Service Properties (Query)

Use the Task Query Service Properties to gather information about all services that run on Agents, including name, state, start mode, recovery options and description. This allows you to compare services plus their settings on multiple Agents.

For example, you can check whether the service Telnet is running on a set of computers. Based on the Query results, you can configure a Task Manage Service Parameters to change the service state of this service to Start Service.

**Note**

If you schedule a recurring Job with this Task and view the results of this Job, the Job History tab in the detailed results window will display changes as they occur when comparing the query results with previous query results. Depending on the configuration of the query, this allows you to track changes in installed applications, printer drivers, services, shares, etc.

Share (Add, Remove, Query)

- Use the Tasks Add Share and Remove Share to share existing folders or printers in your environment, or to remove these shares.
- Use the Task Query Share to search for file shares and printer shares on Agents.

**Notes**

- Most settings require Microsoft Windows XP SP2 or higher.
- When a printer is shared with Agents running different Microsoft Windows versions, additional printer drivers may need to be installed on these Agents.
- If you schedule a recurring Job with this Task and view the results of this Job, the History tab in the detailed results window will display changes as they occur when comparing the query results with previous query results. Depending on the configuration of the query, this allows you to track changes in installed applications, printer drivers, services, shares, etc.

**Tip**

- Use a Task Perform File Operations to create a new folder that can subsequently be shared using the Task Add Share.
- It can be useful to combine the Task Add Share with the Task Query Share, to check for existing shares and their details and to take action based on the query results.
SNMP trap (Send)

Use the Task **Send SNMP trap** to send SNMP traps, which can be received by third-party SNMP managers, such as HP OpenView or CA Unicenter.

This Task is especially useful when combined with other Tasks in a Module. For example, you can use this Task to send a message when another Task fails.

This Task is closely related to the global setting **Send SNMP trap**:

- With the global setting **Send SNMP trap**, you can specify whether RES ONE Automation sends an SNMP trap for every Job that is executed.
- With the Task **Send SNMP trap**, you can send an SNMP trap only for the Job in which the Task is included.

**Configuration**

- RES ONE Automation uses its own mechanism when sending SNMP traps and therefore does not require the installation of SNMP agent software on the computers that run RES ONE Automation.
- The SNMP protocol version and the community string (case-sensitive) that you specify in the Task needs to match the settings of the SNMP manager.
- You can send the SNMP traps to one or more specific IP (IPv4 or IPv6) addresses or broadcast the trap on the local network.
  - When sending a request to an IPv6 destination, the SNMP Version should be set to SNMPv2c.
- To receive the SNMP traps correctly in an SNMP manager, import or load the files `resoneautomation.mib` into this application, depending on the SNMP protocol version that you use. You can find the MIB files in the installation folder of the Console.

TCP/IP Properties (Query)

Use the Task **Query TCP/IP Properties** to check the TCP/IP properties of Agents. For the Agent+, IPv4 and IPv6 addresses are supported. For the legacy Agent, only IPv4 is supported.

Combined with the Task **Apply Registry Settings**, you can use this Task to discover and remedy incorrect DNS Server settings, based on the query results.

**Note**

If you schedule a recurring Job with this Task and view the results of this Job, the **Job History** tab in the detailed results window will display changes as they occur when comparing the query results with previous query results. Depending on the configuration of the query, this allows you to track changes in installed applications, printer drivers, services, shares, etc.
Unattended Installation (Perform)

Use the Task **Perform Unattended Installation** to run unattended installations automatically. This Task is particularly useful for installing software that has not been packaged as an MSI file and has to be installed with a number of command line switches.

**Warning**

The unattended installation of non-MSI packages often requires the use of specific switches and parameters. Make sure that this information is available.

**Configuration**

- In the **Parameters** field on the **Settings** tab, enter the relevant command line switches that are required to perform an unattended installation. See the documentation of the software that you want to install.

- If necessary, enter the credentials that are required to perform the installation in the **Username** and **Password** field. If you do not specify credentials, the installation will be performed by the default system account.

- Optionally, click **...** in the **Optional grab log file** field to grab the log file that is created by the unattended installation on the Agent. This file will then be imported in RES ONE Automation.

**Tip**

It can be useful to combine this Task with the Task **Query Installed Programs**, to obtain an overview of all applications that have been installed on an Agent and to perform an unattended installation of an application based on the query results.
Unix/Linux Command (Execute)

Use the Task **Execute Unix/Linux Command** to run commands on Agents running on Linux or Unix Operating Systems. This allows you to schedule any action that can be performed by a command line. For example, you can use this Task to execute scripts.

**Configuration**

- Scripts that you run on the Agent for Unix/Linux using the Task **Execute Unix/Linux Command** cannot include reboot and shutdown commands. Use the Tasks **Reboot Unix/Linux Computer** or **Shutdown Unix/Linux Computer** instead. See Unix/Linux Computer (Reboot, Shutdown) (see page 145).

**Settings tab**

- Use the option **Insert resource link** from the context menu in the Command line box to specify a Resource. When inserting a Resource link, the GUID of the selected Resource will be used in the command line. The Resource will be added to the command line box as $Workspace{<resource GUID>}. When the Task is executed, the Resource will be downloaded to the Agent`s cache automatically and will be called from that location.
- The option **Set parameter with exit code** allows you to place an exit code into a parameter, for use in another Task in the Module. For example, suppose you have configured a Module with this Task and another Task that installs certain software. By using the option **Set parameter with exit code** in the first Task, you can use this parameter in a condition in the Task that installs software, to determine whether it should be executed.
- The option **Set parameter with standard output** allows you to place the standard output into a parameter, for use in another Task in the Module (for example, a Query).
- In the field **Timeout command execution after**, you can specify a maximum number of 9999 minutes (about 166 hours and 40 minutes). You can use parameters, functions and variables.
  - When 0 timeout is detected (when 0 was entered or when parameter, functions, variables translate to non-numeric values) the fallback will be the maximum timeout period (9999 minutes).
  - In certain situations, Agents can continue to execute the remaining commands in the script when the timeout expired. You can prevent situations like these by selecting the option **Terminate process when timeout expires**.

**Script tab**

- Scripts can be typed in directly.
- Use the **File extension of script** field to specify the file extension that RES ONE Automation should use. This extension is used to save the script as a script file in the specified format when the Task is executed. RES ONE Automation will save this file in the temporary folder on the Agent that executes the Task.
- Use the **Open in editor** button to open an external editor that is associated with the specified file extension in the **File extension of script** field. This makes it easier to create complex scripts and troubleshoot existing ones. After creating or editing the script, it will be copied from the external editor to the **Script** tab. If no associated editor can be found, the script will be opened in Notepad. For more information on how to configure file associations, see [http://msdn.microsoft.com/en-us/library/windows/desktop/cc144175(v=vs.85).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/cc144175%28v=vs.85%29.aspx).
  - The **Scripts** tab has a limit of 64KB. If you use an external editor to edit a script, RES ONE Automation will disregard any characters that exceed this limit.
- Log and error files are included in the detailed Job results.
Chapter 5: Library

Unix/Linux Computer (Reboot, Shutdown)

Use the Tasks Reboot Unix/Linux Computer and Shutdown Unix/Linux Computer to reboot or shut down Agents automatically. A restart is often required after the installation of new software. These Tasks can also be useful after the execution of a Task Execute Unix/Linux Command. See Unix/Linux Command (Execute) (see page 144).

Configuration

- In the Tasks Reboot Unix/Linux Computer, the option Reboot computer after shutdown is enabled by default.
- In the Tasks Shutdown Unix/Linux Computer, the option Reboot computer after shutdown is disabled by default.
- With the option Wait for Agent to come back online, you can specify whether a Job should wait after a reboot with continuing with the next Task in the Job until the Agent has come back online. This option is enabled by default. It can be useful to disable the option in scenarios in which the computer on which the Agent is running switches to a different OS after a reboot and a different Agent comes online. This prevents RES ONE Automation from waiting until the original Agent comes back online.

Unix/Linux Parameters (Query)

Use the Task Query Unix/Linux Parameters to view the values that Module parameters had when a Job was executed on one or more Unix/Linux Agents. This Task is particularly useful in a Module where the value of one or more parameters is set depending on conditions or evaluators.

The results of this Task are presented per Agent and can be sorted per parameter. This Task is also available for Windows (see page 130) and Mac OS X (see page 124) Agents.

Configuration

- If no parameters are shown, this means that no Module parameters are available (yet). When no Module parameters are available, the Evaluators tab is grayed out.
- By default, all Module parameters are reported in this Query. To exclude specific Module parameters from the Query results, clear Show all parameters in Query summary and then clear specific check boxes.

Note
You can only use this Task for Module parameters. If the Module is included in a Project with Project parameters, the Project parameters will overrule the values of the Module parameters.

Unix/Linux Resource (Download)

Use the Task Download Unix/Linux Resource to download Resources to Agents running on Linux or Unix Operating Systems, for example to pre-cache large files or to send a specific version of an MSD file to Agents on which it is needed.

Configuration

When you select the option Enable logging, the detailed Job history will include a log.

Note
It is only possible to download Resources that are stored in the Datastore. Resources stored on a fileshare or RES ONE Automation Resource Packages are not supported.
User Management (Create, Delete, Lock, Unlock, Change Password, Query)

Use the Unix/Linux User Management Tasks to manage user accounts on Agents running on Linux or Unix Operating Systems. You can create, change, delete, lock or unlock user accounts; and you can query all user accounts that exist on the target Agents.

Web Service (Invoke)

Use the Task Invoke Web Service to communicate with a remote web service and, depending on the web service, use it to perform specific actions. The Task may be scheduled on any Agent, as long as the Agent is able to reach the web service.

Configuration

- Use the WSDL field to specify the WSDL file (also files with extensions other than .WSDL are allowed).
  - To specify a WSDL that is stored on a file share, click . If the share requires authentication, you will be prompted for credentials.
  - To access a WSDL from a URL, specify the URL in the WSDL field and specify the required security credentials in the Security context field.
- Use the Security context field to specify any required security credentials. These may contain parameters and/or variables. Security credentials are only required:
  - When accessing a WSDL file from URL.
  - When clicking the Test button.
  - When the Task is performed by the Agent.
- After the WSDL file has been loaded successfully, the credentials will be remembered and all other fields will be cleared (Web Service Host, Method and Description).
- When clicking the Cancel button while the WSDL file is loading, the credentials will be remembered and all other fields will be cleared.
- After browsing to the URL for the WSDL contract or opening a WSDL file the methods available are automatically enumerated from the WSDL. When a method is selected, the Description field (if available) and the Parameter(s) form are filled automatically. It is possible to use RES ONE Automation parameters as Value in the Parameter(s) form.
- Use the Test button to check if the Web Service is available and the parameters are set correctly.
- Use Set parameter with web results to place the response of the Web Service request into a parameter, for use in another Task in the Module (for example, a Query).
  - Use Return as to select the format of the value that will be placed into the parameter:
    - XML: This is the default setting. The parameter will be filled with the response message received from the Web Service host.
    - Table: The response message received from the Web Service host will be analyzed and a comma delimited file will be generated excluding a header row. The parameter will be filled with the contents of the comma delimited file.
    - Table with headers: The response message received from the Web Service host will be analyzed and a comma delimited file will be generated including a header row. The parameter will be filled with the contents of the comma delimited file.

Notes

- If one of the Return as options Table or Table with headers is selected, the response message will be analyzed as follows:
  - The contents of the Soap:Body node will be extracted.
  - The rows of the comma delimited file will be determined by the children of the Soap:Body node.
  - The columns of the comma delimited file will be determined by the children of each row. The header of the column (if required when selecting the option Table with headers) will be the name of the node of the first row.
Chapter 5: Library

Known Limitations

The Invoke Web Service Task only supports single name spaces in SOAP messages. All contents in the SOAP message need to be in that name space. SOAP- and WSDL-based Web Services using multiple name spaces are not supported.

Windows Firewall Settings (Manage, Query)

Use the Task Manage Windows Firewall Settings to manage Windows Firewall settings on Agents. By default, Windows XP SP2 enables Windows Firewall on all network connections. However, this can impair certain types of communications. This Task allows you to modify the appropriate settings for Windows Firewall on Agents, so that communications are not impaired in your RES ONE Automation environment.

Use the Task Query Windows Firewall Settings to retrieve windows firewall settings.

Both the Manage and the Query Tasks support IPv4 and IPv6 addresses for the Agent+. For the legacy Agent, only IPv4 is supported.

Notes

- You can only change the Windows Firewall settings if the target computer runs Microsoft Windows XP SP 2 or higher.
- Before you add any exceptions to the Windows Firewall, carefully consider whether these exceptions are needed at all. Every exception that you add exposes your computer environment to attacks.

Tip

It can be useful to combine this Task with a Task Query Windows Firewall Settings, to take action based on the query results.
Chapter 5: Library

Windows Installer Package (Install, Repair, Patch, Remove)

Use the Tasks Windows Installer Package to install, repair, patch or remove MSI packages on Agents. These Tasks automatically convert the options that you select into a command line, and execute this command line.

Configuration tab

- On the Parameters tab, add public properties for the deployment of the MSI package. Note that these parameters are not related to those specified on the tab Module Parameters. Use the Category list to select:
  - General properties
  - Microsoft App-V Client/SoftGrid Client properties (Microsoft SoftGrid Client 3.1 or later)
  - RES ONE Workspace properties (RES PowerFuse Series 8 or later)
  - RES ONE Workspace Relay Server properties (RES Workspace Manager 2012 Relay Server or later)
  - RES ONE Identity Director properties (RES IT Store 2014 or later)
- At Transforms list on the Parameters tab, you can also select MST files that are contained in a RES ONE Automation Resource Package.
- If enabled, the Log file will be shown in the detailed Job results.
- You can store your configured Logging settings as default for all new Tasks Invoke Windows Installer.

Notes

- Patching MSI packages requires Microsoft Windows Installer 3.1 or higher.
- You can select multiple patches at once.

Tip

- It can be useful to base the execution of this Task on the results of the Task Query Installed Programs, with which you can obtain an overview of all applications installed on an Agent. For example, this allows you to decide on which Agents updates need to be installed, and on which Agents this is not necessary.
- You can install RES ONE Workspace and RES ONE Identity Director using the Task Install Windows Installer Package.
Installing RES ONE Workspace using a Task “Install Windows Installer Package”

You can install RES ONE Workspace or the RES ONE Workspace Relay Server using the RES ONE Automation Task Install Windows Installer Package. This is especially useful if you need to install RES ONE Workspace on several computers at once. This Task supports installations of RES ONE Workspace from RES PowerFuse Series 8 or higher and RES ONE Workspace Relay Server from RES Workspace Manager 2012 or higher.

The installation of RES ONE Workspace or the RES ONE Workspace Relay Server is exactly the same as any other MSI installation using the Task Install Windows Installer Package.

- To install RES ONE Workspace, click the Parameters tab, select the category RES PowerFuse series 8 or later and configure the relevant public properties.
- To install the RES ONE Workspace Relay Server, click the Parameters tab, select the category RES Workspace Manager 2012 Relay Server or later and configure the relevant public properties.

These properties correspond to the properties that are available when perform an unattended installation of RES ONE Workspace or RES ONE Workspace Relay Server via a command line. Please refer to the RES ONE Workspace documentation for more information about the installation of RES ONE Workspace and the Relay Server and the available public properties and their values.
Windows PowerShell Script (Execute)

Use the Task **Execute Windows PowerShell Script** to execute Microsoft Windows PowerShell scripts on Agents. This effectively allows you to create your own, custom Tasks in RES ONE Automation. Microsoft Windows PowerShell is a command line shell and task-based scripting technology for the automation of a wide range of system administration tasks.

Using RES ONE Automation to handle your Windows PowerShell scripts makes it easy to execute and schedule scripts on many computers (Agents). The same version of the script is executed in the same manner across the entire set of Agents, ensuring consistent results. If the script is run on several Agents, RES ONE Automation presents the outputs in a single Job result.

**Prerequisites**

- Microsoft Windows PowerShell must be installed on the target computer. It is available as a free download on the Microsoft Download Center.
- The Windows PowerShell script is executed with the user profile of the user name provided at Security Context for this Task. The account that is used to execute Windows PowerShell scripts needs to have "Log on locally" rights.

**Configuration**

**Settings tab**

- By default, the PowerShell scripts that you can run using this Task need to be digitally signed. Select **Override execution policy for this Task** to temporarily lower the PowerShell execution policy to "Unrestricted" and use unsigned PowerShell scripts. After execution of the Task, the PowerShell security will be reverted to the previous security level. This can be useful for example when running the VI Toolkit cmdlet.
- If you use PowerShell snap-ins, for example to manage your Microsoft Exchange Server 2007 servers, store the console file that identifies the snap-ins (.psc1) as a Resource. Refer to this Resource in the Task, at **Use Windows PowerShell Console File** under **Advanced Settings**.
- In the optional field **Grab log file**, you can select a text file to be collected from the target machine. This file will be shown as an extra tab in the Job results named **Grabbed Log**.
- Depending on the amount of information presented in the output, columns may be very far apart, which makes for awkward reading; or too narrow, so that the information is truncated. Under **Advanced Settings**, adjust the total width of the Windows PowerShell Console output to obtain wider or narrower columns.
- In the field **Set width of Windows PowerShell console output at**, you can specify the width of the Windows PowerShell console (default is 16386 Bytes). This is useful to prevent data from becoming unreadable. You can use parameters, functions and variables.
- In the field **Timeout Windows PowerShell execution after**; you can specify a maximum number of 9999 minutes (about 166 hours and 40 minutes). You can use parameters, functions and variables.

In certain situations, Agents can continue to execute the remaining commands in the script when the timeout expired. You can prevent situations like these by selecting **Terminate Windows PowerShell when timeout expires**. This exits any active process when the timeout expires.

- If the use of parameters, functions and/or variables in the fields **Set width of Windows PowerShell console output at** and **Timeout command execution after** result in non-numeric values (text) when the Task is executed, the width and timeout will use a fall back value:
  - **Set width of Windows PowerShell console output at**: Fall back value is 16386 Bytes (16KB).
  - **Timeout command execution after**: Fall back value is 1 minute.

- The option **Set parameter with exit code** allows you place an exit code into a parameter, for use in another Task in the Module. For example, suppose you have configured a Module with a Task **Execute Command** and a Task that installs certain software. By using the option **Set
**parameter with exit code** in the Task **Execute Command**, you can use this parameter in a condition in the Task that installs software, to determine whether this Task should be executed.

- The option **Set parameter with standard output** allows you to place the standard output into a parameter, for use in another Task in the Module (for example, a Query).

**Script tab**

- Scripts can be typed in directly. You can use RES ONE Automation functions, RES ONE Automation parameters and environment Variables in Secure Shell commands. These Variables, functions and parameters will be parsed when the Task is executed.
  - To prevent code injection when using parameters from an untrusted source, use the `Get-ResParam` Cmdlet to safely pass the data from these parameters.
    - Example for the parameter 'MyPassword':
      - For trusted sources, use: `$[MyPassword]`
      - For untrusted sources, use: `Get-ResParam -Name MyPassword`
- Use the **File extension of script** field to specify the file extension that RES ONE Automation should use. This extension is used to save the script as a script file in the specified format when the Task is executed. RES ONE Automation will save this file in the temporary folder on the Agent that executes the Task.
- Use the **Open in editor** button to open an external editor that is associated with the specified file extension in the **File extension of script** field. This makes it easier to create complex scripts and troubleshoot existing ones. After creating or editing the script, it will be copied from the external editor to the **Script** tab. If no associated editor can be found, the script will be opened in Notepad. For more information on how to configure file associations, see [http://msdn.microsoft.com/en-us/library/windows/desktop/cc144175(v=vs.85).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/cc144175(v=vs.85).aspx).
  - The **Script** tab has a limit of 64KB. If you use an external editor to edit a script, RES ONE Automation will disregard any characters that exceed this limit.
- Any script output is shown in the detailed results per Agent, on a separate tab **Console Output**. The default output format is **format-table**.
- When executing this Task, log files will be generated and included in the detailed Job results. Before the Agent uploads these output log, error log or grabbed log files to the Dispatcher and make them available, the Agent will first compress the file if it has a size less than 100 MB.
  - In the following situations, the compression of that file will not take place. As a result, the file will not be uploaded:
    - The file size cannot be determined
    - The file is empty
    - The file is larger than 100 MB
  - The Job results will report if a file could not be uploaded because the file size could not be determined, or if the file size was larger than 100 MB. Empty files will not be reported. If a file cannot be uploaded, the outcome of the Task is not affected: the Task will still complete successfully, but the log file(s) that were not uploaded are not available in the Console.

---

**Note**

To find out more about signing scripts, open Windows PowerShell, type the following, and press ENTER: `Get-Help About_Signing`.  

---

Copyright © RES Software Development B.V. All rights reserved.
**PowerShell Group Policies (GPOs)**

In the following scenarios, execution of the Task **Execute Windows PowerShell Script** will lead to different results when GPOs are active on the machine on which it is executed:

**Scenario 1**

A user on this system opens a PowerShell Console and specifies a PowerShell command.

Result: This command executes normally.

**Scenario 2**

The user creates a PowerShell script containing the PowerShell command and tries to execute the script.

Result: An error message is shown, because the user is not allowed to execute scripts. This is enforced by the GPO.

**Explanation**

In the Task **Execute Windows PowerShell Script**, the source of the script may be either one of the following:

- A PowerShell script that was loaded as a Resource at an earlier stage.
- Script contents specified on the **Script** tab.

The Agent will execute the PowerShell commands (from either of these sources) as explained in scenario 1. As a result, the commands are executed normally, even when an active GPO prohibits the execution of PowerShell scripts.

Of course, if the script contains an instruction to call another script, scenario 2 is performed and an error message will be shown (enforced by the GPO).

**Windows Shell (Set, Query)**

- Use the Task **Set Windows Shell** to change the desktop shell of Agents. This Task is very useful for Agents running RES ONE Workspace and/or RES Subscriber for VDX. This Task also applies to computers running RES Subscriber.
- Use the Task **Query Windows Shell** to view which shell is active on Agents. Microsoft Windows, RES ONE Workspace, RES Subscriber for VDX or a custom shell. This Task also applies to computers running RES Subscriber.

**Configuration**

In a Task **Set Windows Shell**, if you set **RES Subscriber** as shell, the RES Subscriber for VDX Menu will completely replace the local desktop. This applies to the computer, regardless of the user who logs on.

**Notes**

- It can be useful to combine the Task **Set Windows Shell** with the Task **Query Windows Shell**, to check which shells are currently in use in your environment and to change the shells, based on the query results.
- You can only run this Task on workstations.
- When changing the shell, the application and/or Resources to which the shell refers need to be installed on the Agents.
- When changing the shell to **RES Subscriber**, **RES Subscriber** or **RES Subscriber for VDX** must be installed and configured on each local computer where virtual desktop applications need to be available. For more information about **RES Subscriber for VDX**, please refer to the Getting Started with RES Subscriber for VDX, available at [http://success.res.com](http://success.res.com) at the Downloads section.
5.2 Projects

If you find yourself repeatedly scheduling the same sequence of Modules, you can combine Modules into a Project.

You could also combine multiple Modules in one Job when scheduling it, without creating a Project. However, if you frequently combine the same Modules into a Job, it is more convenient to create a Project once that consists of those multiple Modules. By combining several Modules into one Project, the Modules in the Project can be scheduled as one Job instead of as a series of separate Jobs. A Project is therefore ideal for combinations of Modules that you need to perform often.

5.2.1 Managing Projects

At Library > Projects:

• On the Folders tab of the Projects node, you can create a folder structure for all your Projects and place the Projects that you have created in the designated folders. Grouping Projects in folders can be useful in large environments and multi-tenant RES ONE Automation sites to easily locate any Projects you have created and also to create folders for various needs.
  • You can add, edit or delete folders by using the three folder buttons on the lower right side of the screen.
  • If you delete a folder, its contents will be deleted as well.
• The Projects tab shows exactly the same list of Projects, but alphabetically listed and without the folder structure.
• To create a duplicate of a Project, select it and click Duplicate in the command bar. Creating duplicates of a Project is useful when you need to create a new Project that only slightly differentiates from an already existing Project. By changing the values and settings of a duplicated Project, you can customize it to your needs. This saves configuration time.
• When saving changes to the Module, the Version Control window will be opened. This window makes it possible to configure the versioning (see page 272) of changes to the Module. The availability of this window depends on the global setting Use silent versioning (see page 48).
• After modifying a Module, a notification window will open if the Module is used in a scheduled Job. Provided you have sufficient permissions to the Scheduling node, you can use this window to open the scheduled Job and reload the relevant Tasks, if necessary.

When configuring a Project:

• Use the Properties tab to specify a name and an optional description for the Module.
• Use the Modules tab to manage the Modules in the Project.
• Use the Current Resulting Tasks tab to view the combination of all Tasks of all Modules in the Project.
• Use the Project Parameters tab to manage the Module parameters (on page 257) in the Project.
• Use the Usage tab to view in which Run Book(s) and/or Team(s) the Project is used.
• Use the Job History tab to view details of Jobs in which the Project was used and to reschedule these Jobs if necessary.
• Use the Versioning tab to view a list of changes made to the Project. This includes all known versions of the Project.
• Use the Permissions tab to view the level of access that Console users have to the Module.
5.2.2 Project Properties tab

At Library > Projects, you can configure and manage Projects. By combining several Modules into one Project, the Modules in the Project can be scheduled as one Job instead of a series of separate Jobs. When configuring Projects, use the Properties tab to specify a name and an optional description for the Project.

Configuration
- The Project name is the name that is displayed on the Folders/Projects tab of the Projects node. When creating a Project, it is therefore wise to choose a name that reflects its contents.
- The Description that you can optionally specify is only shown on the Projects tab. In the Description field, you can use hyperlinks. Console users can directly browse to these hyperlinks from the Console. The following links are supported:
  - http://
  - https://
  - file://

5.2.3 Project Modules tab

At Library > Projects, you can configure and manage Projects. By combining several Modules into one Project, the Modules in the Project can be scheduled as one Job instead of a series of separate Jobs. When configuring Projects, use the Modules tab to manage the Modules in the Project.

Modules are containers for the Tasks that you can execute on Agents. Modules hold information about how these Tasks should be executed, such as Task settings, order of execution of Tasks, parameters and conditions.

Configuration
- When adding a new Module, you can use the Instant Search field when selecting Modules. To find a specific Module, simply start typing part of its name. As you type, the list will be filtered to show only items that contain the search term. Alternatively, you can just browse the tree of available Modules.
- When you disable a Module in a Project, the Module will only be disabled for the Project and not “globally”.
- To create a duplicate of a Module, select it and click Duplicate. Creating a duplicate of a Module is useful when you need to configure a new Module that only slightly differentiates from an already existing Module in the Project. By changing the values and settings of a duplicated Module, you can customize it to your needs. This saves configuration time.
- You can add multiple Modules to a Project.
- To create sets of Modules, right-click the Modules tab and click Add Separator. Separators are a visual aid to enhance readability in Projects that contain many Modules; they do not affect the execution order of Modules, but make it possible to configure visible sections of Modules that, for example, are related to each other. In addition, it is possible to add a descriptive text to a set of Modules.
  - The name of a separator can contain a maximum of 255 characters.
  - The description of a separator can contain a maximum of 4000 characters. The use of hyperlinks is supported.
- To change the order of execution of Modules in the Project, select a Module and use the arrow buttons. Likewise, use the arrow buttons to move any separators up or down the list of Modules.
- To set conditions (on page 233) on a Module, select the Module and click Conditions. Module conditions determine whether a Module in a Project should be executed, skipped or failed.
5.2.4 Project Current Resulting Tasks tab

At Library > Projects, you can configure and manage Projects. By combining several Modules into one Project, the Modules in the Project can be scheduled as one Job instead of a series of separate Jobs. When configuring Projects, use the Current Resulting Tasks tab to view the combination of all Tasks of all Modules in the Project.

Configuration

• To view the configuration of a specific Task, select it and click View. This will show read-only version of the Task configuration window. It is not possible to make changes to these settings from the Project window.

5.2.5 Project Parameters tab

At Library > Projects, you can configure and manage Projects. By combining several Modules into one Project, the Modules in the Project can be scheduled as one Job instead of a series of separate Jobs. When configuring Projects, use the Project Parameters tab to manage the Module parameters in the Project.

When you combine multiple Modules into a Project, you can link any Module parameters in these Modules to corresponding Project parameters. With a linked Project parameter, input is requested only once. When linking a Project parameter to a Module parameter, you can specify which action should be performed at the input moment. This action determines which value should be used by the parameters. See Project parameters (on page 257).

Configuration

Project Parameters > Parameters tab

• Click AutoCreate to create Project parameters automatically. During creation, you will also be asked if you also want to link the Project parameters to matching Module parameters automatically.
• Use the arrow buttons to change the order in which Project parameters are presented at input moment.
• Click Cleanup to delete Project parameters that are not linked to any Module parameter in the Project.
Project Parameters > Links tab

- To link a Project parameter to one or more Module parameters manually, select a Module parameter and drag it to a Project parameter. After this, click Action to specify which action should apply to the link.

<table>
<thead>
<tr>
<th>Link action</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>The Project parameter is not linked to any underlying Module parameters. As a result, each Module parameter keeps its own value.</td>
</tr>
<tr>
<td>Set initial value</td>
<td>Use this action to set the value of the Project parameter as initial value for the Module parameter that is linked to it. This is the default action when auto-linking parameters. For example, this is useful if multiple Module parameters from different Modules in the Project share the same value (e.g. a specific user name or password). By specifying this value in the Project parameter, the linked Module parameters will all get the same value at the input moment.</td>
</tr>
<tr>
<td>Get final value</td>
<td>Use this action to get the value of the Project parameter from the Module parameter that is linked to it after its Module has been executed. This is useful if you want to use this value in the Module parameter of the next Module in the Project before it is executed (through a Set initial value link-action).</td>
</tr>
<tr>
<td>Both</td>
<td>Use this action to set the value of the Project parameter as initial value for the Module parameter that is linked to it. After execution of its Module, the Project parameter will then get its value again from the Module parameter that is linked to it. You can then use this parameter value in the next Module of the Project.</td>
</tr>
</tbody>
</table>

- Alternatively, you can specify which action should apply to the link by editing the properties of the Project parameter (on the Links tab of the Add Project Parameter window).
- To link all unlinked Module parameters to matching Project parameters automatically, click AutoLink. This automatically links existing Project parameters to matching Module parameters of the same name and type. If no matching Project parameters exist yet, you will be asked if you also want to create these automatically. AutoLinked Project parameters use the default link action Set initial value.
- Linked Module parameters are shown below the Project parameter to which they are linked; unlinked Module parameters are listed separately.
- To unlink a linked Module parameter, right-click the linked Module parameter and click Delete. This will only delete the link; not the Module parameter itself.
- Alternatively, you can delete links by dragging it to Not Linked at the top of the list.

### 5.2.6 Project Usage tab

At Library > Projects, you can configure and manage Projects. By combining several Modules into one Project, the Modules in the Project can be scheduled as one Job instead of a series of separate Jobs. When configuring Projects, use the Usage tab to view in which Run Book(s) the Project is used.

Besides the Run Books in which the Project is used, the Usage tab also shows in which Team(s) it is used if Teams use the Project for Automatic Job Scheduling. See Teams (on page 205).

**Configuration**

- To jump to a Run Book or Team, select it and click Open.
5.2.7 Project Job History tab

At Library > Projects, you can configure and manage Projects. By combining several Modules into one Project, the Modules in the Project can be scheduled as one Job instead of a series of separate Jobs. When configuring Projects, use the Job History tab to view details of Jobs in which the Project was used and to reschedule these Jobs if necessary.

Configuration

- To reschedule (part of) a Job, right-click the relevant Job and click Reschedule Job.

5.2.8 Project Versioning tab

At Library > Projects, you can configure and manage Projects. By combining several Modules into one Project, the Modules in the Project can be scheduled as one Job instead of a series of separate Jobs. When configuring Projects, use the Versioning tab to view a list of changes made to the Project, including all known versions of the Project.

Configuration

- The Current version field specifies the latest saved version number of the Project. It is possible to change this version number manually by clicking ..., regardless of the global setting Use automatic versioning (see page 48).
- The Version column specifies all versions of the Project.
- The Comment column specifies the comment that Console users made about individual versions.
- Use the drop-down field to customize the number of Audit Trail results that are shown per page.
- Use the navigation buttons to browse through the various Audit Trail pages.

5.2.9 Project Permissions tab

At Library > Projects, you can configure and manage Projects. By combining several Modules into one Project, the Modules in the Project can be scheduled as one Job instead of a series of separate Jobs. When configuring Projects, use the Permissions tab to view the level of access that Console users have to the Project.

Configuration

- Use the option Do not allow this Project to be scheduled programmatically to allow the Project from being invoked via the RES ONE Automation Console only. When selected, it will no longer be possible to invoke the Project:
  - During unattended installations of an Agent (by applying properties to its MSI file).
  - Via the Prepare for Image functionality.
  - Via RES ONE Workspace.
5.3 Run Books

When you schedule a Job in RES ONE Automation, all Tasks that are part of the Job will be performed by all specified Agents. With Run Books, you can create a chain of Jobs that makes it unnecessary to schedule these Jobs individually. Because each Job in the chain can be performed by a different (set of) Agent(s), a Run Book can perform all steps of a specific maintenance Task. And because you can let each step depend on the results of the previous step, you can create predictable results: with Run Books, maintenance Tasks can only perform in one way, based on best practices. It is as if you are saying: “In this particular scenario, the following Job should be performed on this computer, the next Job on that computer, the next Job on these computers, then the next Job on this computer, etc.”

Because provisioning tasks are rarely performed on a single computer, Run Books are very useful in complex maintenance tasks in which, for example, a computer needs to be set up for a newly hired employee, an ERP server needs to be backed up, or all servers in a Citrix server farm need to reboot sequentially.

For example, when you need to set up a computer for a newly hired employee, a whole sequence of Tasks is needed on different computers: on the Active Directory server, on the file server, on the local computers, etc. You can create Modules and Projects for the necessary sequence of actions, and can combine these in a Run Book. This ensures that the Jobs on the different Agents will be executed in turn, with each Job waiting until the previous one has finished.

With RES ONE Automation's Run Book technology

- you can chain any combination of Jobs into a single Run Book. This makes it unnecessary to schedule these Jobs individually and makes it possible to perform these Jobs sequentially.
- each Job can be performed by a different Agent, which is very useful when a certain change to your environment requires that a certain Task is performed by one Agent and another Task by another Agent.
- you only need to configure once, after which Run Books can be used repeatedly. Because you can use Run Book parameters in Run Books, you can make Run Books generic. For example, if you configure a parameter for the user name, you can use the relevant user name in all the underlying Modules and parameters each time you schedule the Run Book. Not only does this ensure that all necessary steps are performed, but also that these steps can be repeated when necessary - whether next week or next year.

5.3.1 Scenarios

Scenario 1

When you need to set up a computer for a newly-hired employee, a whole sequence of Tasks is needed on different computers: on the Active Directory server, on the file server, on the local computers, etc. You can create Modules and Projects for the necessary sequence of actions, and can combine these in a Run Book. This ensures that the Jobs on the different Agents will be executed in turn, each Job waiting until the previous one has finished.
Scenario 2

When you need to update the servers in a Citrix server farm with the latest security update, all servers need to be rebooted after the update has been installed. Of course, you could schedule this reboot for each Citrix server individually: Servers 1-5 should reboot at 12:00, servers 6-10 should reboot at 16:00, servers 11-15 should reboot at 20:00, and so on. However, this could lead to a situation in which none of the servers are online: if the servers that were rebooted at 12:00 fail to come online again, the reboot scheduled at 16:00 will still take place. Just like the other scheduled reboots. You can prevent this situation by using a Run Book. For example, you can create a Run Book Job 1 that installs the security update on all servers, a Run Book Job 2 that reboots servers 1-5, a Run Book Job 3 that reboots servers 6-10, and so on. When the Run Book is executed, all Run Book Jobs will wait for the previous Run Book Job to finish before they will be started. Servers 6-10 will not be rebooted until servers 1-5 are back online again, and so on.

5.3.2 Managing Run Books

At Library > Run Books:

- On the Folders tab of the Run Books node, you can create a folder structure for all your Run Books and place the Run Books that you have created in the designated folders. Grouping Run Books in folders can be useful in large environments and multi-tenant RES ONE Automation sites to easily locate any Run Books you have created and also to create folders for various needs.
  - You can drag-and-drop existing Run Books into folders.
  - You can add, edit or delete folders by using the three folder buttons on the lower right side of the screen.
  - If you delete a folder, its contents will be deleted as well.
- The Run Books tab shows exactly the same list of Run Books, but alphabetically listed and without the folder structure.
- To create a duplicate of a Run Book, select it and click Duplicate in the command bar. Creating duplicates of a Run Book is useful when you need to create a new Run Book that only slightly differentiates from an already existing Run Book. By changing the values and settings of a duplicated Run Book, you can customize it to your needs. This saves configuration time.
- To create a RES ONE Identity Director service based on a Run Book, right-click it and click Publish in RES ONE Identity Director from the context menu. This opens the RES ONE Identity Director Service Wizard. See Creating RES ONE Identity Director services based on Run Books (on page 167).
- When saving changes to the Run Book, the Version Control window will be opened. This window makes it possible to configure the versioning (see page 272) of changes to the Run Book. The availability of this window depends on the global setting Use silent versioning.
- After modifying a Run Book, a notification window will open if the Run Book is used in a scheduled Job. Provided you have sufficient permissions to the Scheduling node, you can use this window to open the scheduled Job and reload the relevant Tasks, if necessary.
- You can schedule a Run Book as if it were a normal Job.

When configuring a Run Book:

- Use the Properties tab to specify a name and an optional description for the Run Book.
- Use the Jobs tab to manage the Jobs in the Run Book.
- Use the Run Book Parameters tab to manage the Job parameters in the Run Book.
- Use the Job History tab to view details of Jobs in which the Run Book was used and to reschedule these Jobs if necessary.
- Use the Versioning tab to view a list of changes made to the Run Book, including all known versions of the Run Book.
- Use the Permissions tab to view the level of access that Console users have to the Run Book.
5.3.3 Run Book Properties tab

At Library > Run Books, you can configure and manage Run Books. With Run Books, you can create a chain of Jobs that makes it unnecessary to schedule these Jobs individually. When configuring Run Books, use the Properties tab to specify a name and an optional description for the Run Book.

Configuration

- The Run Book name is the name that is displayed on the Folders/Run Books tab of the Run Books node. When creating a Run Book, it is therefore wise to choose a name that reflects its contents.
- The Description that you can optionally specify is only shown on the Run Books tab. In the Description field, you can use hyperlinks. Console users can directly browse to these hyperlinks from the Console. The following links are supported:
  - http://
  - https://
  - file://
- You can disable a Run Book. If a Run Book is disabled, you can no longer directly schedule it.

5.3.4 Run Book Jobs tab

When using Run Books, it is important to understand the difference between Job scheduling for Modules and Projects and Job scheduling for Run Books.

Job Scheduling for Modules and Projects

A Module is a container for one or more Tasks that you can execute on Agents. A Module usually contains a sequence of Tasks that are related to each other (for instance, a Task Install Windows Installer Package and a Task Reboot).

A Project is a combination of Modules. If you frequently use the same combination of Modules, it can be more convenient to combine these Modules in a Project.

When you schedule a Job at a specific time, you assign one or more Agents to execute one or more Modules and/or Projects. The Agents will execute the Tasks in these Modules and/or Projects consecutively.

In effect, you are saying: “the following Job should be performed by the following Agents at the following moment.” This way of Job scheduling is sufficient to perform relatively simple maintenance tasks, such as installing an application or a Windows security update.

Job Scheduling for Run Books

A Run Book consists of a chain of Jobs that are executed in turn, possibly on different Agents.

Run Books make it possible to schedule Jobs as a chain of Jobs, where each Job in a Run Book can be executed by a different Agent. It becomes unnecessary to schedule these Jobs individually. In effect, you are saying: “The first Job should be performed by this Agent, the next Job by that Team, the next Job by these Agents, then the next Job on this Agent, etc.”

Configuration

- To create a duplicate of a Run Book Job, select it and click Duplicate. Creating a duplicate of a Job is useful when certain Jobs should perform the same action (for example, reboot a server). This not only saves configuration time, but also ensures that all duplicated Run Book Jobs contain the same content. By changing the values and settings of a duplicated Run Book Job, you can customize it to your needs (for example to specify a different server for each reboot action).
- You can add multiple Jobs to a Run Book.
• To create sets of Jobs, right-click the Jobs tab and click Add Separator. Separators are a visual aid to enhance readability in Run Books that contain many Jobs; they do not affect the execution order of Jobs, but make it possible to configure visible sections of Jobs that, for example, are related to each other. In addition, it is possible to add a descriptive text to a set of Jobs.
  • The name of a separator can contain a maximum of 255 characters.
  • The description of a separator can contain a maximum of 4000 characters. The use of hyperlinks is supported.
• To change the order of execution of Jobs in the Run Book, select a Job and use the arrow buttons. Likewise, use the arrow buttons to move any separators up or down the list of Modules.
• To set conditions (on page 233) on a Job, select the Job and click Conditions. Job conditions determine whether a Job in a Run Book should be executed, skipped or failed.

When configuring a Run Book Job:
• Use the What section on the Properties tab of the Add Run Book Job window to specify which Modules or Project should be executed.
  • On the Resulting Tasks tab, you can view which Tasks result from the selection of Modules and Projects. It is not possible to make changes to the Tasks from this tab; only in the original Module or Project. If you do so, an already scheduled Job will NOT include these changes, unless you click Reload Resulting Tasks.
• Use the Who field to specify which Team(s) or Agent(s) should perform the Job.
  • When selecting Agents that should perform the Job, you can filter on Agents that are member of a Team or on Agents that executed a specific Project or Module.
  • You can filter the results further by searching on all Agents, only licensed and online Agents, offline Agents, or no Agents and include Teams in the search results.
  • As an alternative to selecting Agents manually, you can add multiple Agents at once by copying them from a list of Agents separated by semi-colons (";"), comma’s (","), or tabs and pasting them directly into the Who section.
  • On the Resulting Agents tab you can view all Agents that will perform the Tasks. Any Teams selected in the Who field are automatically resolved into Agents. If a Team changed after the Run Book Job was created and before it is executed, click Reload Resulting Agents to refresh the list of Agents.
• Ask during Run Book scheduling: select this option to leave the choice of which Team(s) or Agent(s) should perform the Job to the moment when the Run Book is scheduled. Any Agent(s) or Team(s) that you specify at that moment will then be used for all Run Book Jobs for which no Agents or Teams were specified (for example, if your Run Book contains two Run Book Jobs that use this option, you can only specify Agent(s) or Team(s) once). This option is useful, for example if a Run Book Job contains a Task to install a specific application or update, but it is not yet known which Agent should install the application or update.
• Run on one of the specified Agents: select this option to execute the Run Book Job on a random Agent. This is useful if it does not matter which Agent executes the Job, as long as it is executed. For example, for Exchange Tasks and some Active Directory Tasks, it does not matter by which Agents they are executed. When you schedule a Run Book Job on a set of Agents or a Team, RES ONE Automation will randomly select one of these Agents to executed the Job. The order in which RES ONE Automation selects an Agent that should execute the Job is as follows:
  1. Any online Agent that is not busy executing another Job.
  2. Any online Agent.
  3. Any offline Agent.
- **Use Run Book Parameter:** select this option to parameterize the **Who** field with the parameter $\{\text{RunBookWho}\}$ (availability of this option depends on the global setting **Run Book Who** (see page 35)). When selected, the parameter $\{\text{RunBookWho}\}$ will specify which Team(s) or Agent(s) should perform the Job. It may, for example, contain a number of Agents or users, separated by a semi-colon. If also **Enhanced security** is enabled, you are only allowed to schedule the Run Book with the parameter $\{\text{RunBookWho}\}$ when using a login with **RES ONE Automation Authentication**. This functionality is particularly useful when using the Run Book in a **RES ONE Identity Director** service: it allows for a scenario in which the Agents that should execute the Run Book Job can be selected from a list during the delivery of the **RES ONE Identity Director** service. When you have created or selected a Run Book parameter, also specify how it should be used:
  - **to select Agents on computer name:** select this option to execute the Run Book Job by Agents running on the computers that match the provided computer name(s). Alternatively, you can also provide Team names. Separate multiple entries with a semi-colon (;).
  - **to select Agents on active Console user:** select this option to execute the Run Book Job by online Agents running on computers on which Console users are logged on whose names match the provided user names. Because Console users can be logged on at more than one machine, the Run Book may be executed by multiple Agents. The input format of the parameter is a user name (for example `demo\jsmith` or `root`). In **RES ONE Identity Director**, this allows for scenarios in which a Console user can request a service that provides him with something on the computer on which he is currently logged on.
  - **to select Agents on last Console user:** select this option to execute the Run Book Job by Agents running on computers on which the last Console user logged on. Because Console users can be logged on at more than one machine, the Run Book may be executed by multiple Agents.
  - **to select most recent Agent on last Console user:** select this option to execute the Run Book Job by the Agent running on the most recent computer on which the last Console user logged on.
  - **Create multiple Jobs and divide Agents equally amongst them:** select this option to automatically create a specific number of duplicates of a Run Book Job and divide the relevant Agents amongst those duplicates. For example, instead of creating one Run Book Job to be performed on 50 Agents at the same time, you can automatically split the Run Book Job into 10 Run Book Jobs that are each performed on 5 Agents. Because Run Book Jobs are executed in turn rather than at the same time, this is ideal for reboot schedules of server farms, for example. When you select this option and click **OK**, you will be asked how many Jobs you want to create. The available Agents will be divided amongst the specified number of Jobs.
  - **Schedule offline Agents for next boot:** select this option to postpone the execution of the Job for any offline Agents. During Job scheduling, **RES ONE Automation** will automatically determine whether Agents are online or offline and create **Immediately** Jobs for any online Agents and **After next boot** Jobs for any offline Agents. When **one** of the online Agents gets online, the **After next boot** Job will be executed and remains active until **all** specified Agents have completed the Job. If you do not select this option, the Job will fail on all offline Agents, depending on the **Launch timeout** settings (at **Setup > Global Settings**).
  - **or when Agent is resumed after hibernate, sleep or standby:** select this option when scheduling Jobs on Agents running on laptops: When a laptop goes into hibernation, sleep or standby, the Agent running on the laptop goes offline. However, when the laptop is resumed and the Agent comes back online again, any Jobs that are scheduled for next boot on offline Agents will not be executed, because the laptop has not booted. As many users rarely turn off their laptop, but put it in hibernation, sleep or standby instead, this can cause a backlog in Jobs that need to be executed on these Agents. By selecting the option **or when Agent is resumed after hibernate, sleep or standby**, Jobs scheduled on these Agents will also be executed.
• **Use Wake-On-LAN when Job starts**: select this option to boot offline Agents automatically when the Job starts, by broadcasting a “wake-up” packet to the Agent. This option forces the Agent to use UTC time. By default, RES ONE Automation uses a global broadcast address, 255.255.255.255 and UDP port 3163 for Wake-on-LAN. If necessary, you can set up different Wake-on-LAN settings at Setup > Global Settings. This option forces the Agent to use UTC time without any further restrictions. This option cannot be used in combination with option Schedule offline Agents for next boot.

• **The Error control options** allow you to determine what should happen to the Run Book if the Run Book Job fails:
  - to fail the entire Run Book, select **Stop Run Book on error**.
  - to proceed to the next Job in the Run Book, select **Continue Run Book on error**.

• **Maximum Job duration**: use this field to configure a timeout that specifies the maximum Job duration of the Job. This can be useful to prevent situations in which Run Books cannot be completed because an Agent that executes a Run Book Job goes offline during execution. Because the Run Book Job remains active, a succeeding Job in the Run Book will not start. If a timeout has been specified, the Run Book Job will be canceled when the timeout expires. A succeeding Run Book Job can then be executed.
  - **Maximum Job duration** is disabled by default. When enabled, a default value of 10 minutes will be shown.
  - If the Run Book Job is not completed within the specified timeout, it will change the status of any active Agent to the status **Timed out**. Subsequent Run Book Jobs will continue as normal. The end result of the Run Book will be **Completed with errors**.

---

**Notes**

• When you select **Run on one of the specified Agents**, selecting the options **Schedule offline Agents for next boot** and **Use Wake-on-LAN when job starts** only make sense if the selected Agents are offline.

• The option **Create multiple Jobs and divide Agents equally amongst them** is only available when configuring a new Run Book Job. It is not available when editing an existing Run Book Job.

• The following applies to the option **Use Run Book parameter**:
  - You can use the option in combination with the option **Run on one of specified Agents**.
  - The option does not allow for the use of parameter values from a CSV file. See Using parameter values from a CSV file (on page 61).
  - The options based on Console users do not apply to Agents running on terminal servers.
  - If the option **to select Agents on computer name** is selected and a combination of Agents and Teams is provided, RES ONE Automation will schedule the Job on Agents only if the first name that is provided matches an Agent; it will schedule the Job on Teams only if the first name that is provided matches a Team. If an Agent is a member of multiple Teams, the Job will be scheduled once on this Agent.
  - In combination with **Enhanced security** on the Run Book Who and scheduled to a (Team with a) list of Agents, the Run Book Job will only be executed on Agents the user has access to (login with RES ONE Automation Authentication). The other Agents will be skipped. If no Agents are allowed to execute the Job, the Job will be canceled and an error message will be shown. Please note that the permissions of the user are being checked during scheduling and not at execution of the Run Book. So if, for instance, the Run Book was already scheduled and over time the access rights of the user have been revoked, the first Job will be executed and the consecutive Jobs will be aborted.

• In RES ONE Identity Director, using a Run Book with **Use Run Book parameter** set to **Select Agents on computer name**, may lead to unexpected results in the following situation:
  - If this Run Book is used in a RES ONE Identity Director service,
  - And the Run Book parameter is linked to a service attribute,
  - And the service attribute is based on a People Identifier,
  - And the People Identifier contains multiple values,
  - Then the first People Identifier value that was added, will be used as the Run Book parameter value. This does not necessarily have to be the value that appears first in the list. For example, if a user is identified with a Windows user account and the person identifier has the values ‘demo\jsmith’ and ‘resdemo\jsmith’.
  - This is a known limitation.
5.3.5 Run Book Parameters tab

At Library > Run Books, you can configure and manage Run Books. With Run Books, you can create a chain of Jobs that makes it unnecessary to schedule these Jobs individually. When configuring Run Books, use the Run Book Parameters tab to manage the Job parameters in the Run Book. See Parameters (on page 257).

Like Project parameters, a Run Book parameter is an umbrella parameter for a set of Module parameters and/or Project parameters that occur in one Run Book. With a linked Run Book parameter, input is requested only once. In Run Books, Module parameters and Project parameters are known as Job parameters. When linking a Run Book parameter to a Job parameter, you can specify which action should be performed by the link at the input moment to determine the value of a parameter.

Configuration

Run Book Parameters > Parameters tab

- Click AutoCreate to create Run Book parameters automatically. During creation, you will also be asked if you also want to link the Run Book parameters to matching Job parameters automatically.
- Use the arrow buttons to change the order in which Run Book parameters are presented at input moment.
- Click Delete to delete Run Book parameters.
- Click Cleanup to delete Run Book parameters that are not linked to any Job parameters in the Run Book.

Project Parameters > Links tab

- To link a Run Book parameter to one or more Job parameters manually, select a Job parameter and drag it to a Run Book parameter. After this, click Action to specify which action (see page 264) should apply to the link.
- Alternatively, you can link a Run Book parameter to one or more Job parameters manually by editing the properties of the Run Book parameter (on the Links tab of the Add Run Book Parameter window).
- To link all unlinked Job parameters to matching Run Book parameters automatically, click AutoLink. This automatically links existing Run Book parameters to matching Job parameters of the same name and type. If no matching Run Book parameters exist yet, you will be asked if you also want to create these automatically. AutoLinked Run Book parameters use the default link action Set initial value.
- Linked Job parameters are shown below the Run Book parameter to which they are linked; unlinked Job parameters are listed separately.
- To unlink a linked Job parameter: Right-click the linked Job parameter and click Delete. This will only delete the link; not the Job parameter itself.
- Alternatively, you can delete links by dragging it to Not Linked at the top of the list.
Notes

- In RES ONE Identity Director, using a Run Book with **Use Run Book parameter** set to **Select Agents on computer name**, may lead to unexpected results in the following situation:
  - If this Run Book is used in a RES ONE Identity Director service,
  - And the Run Book parameter is linked to a service attribute,
  - And the service attribute is based on a People Identifier,
  - And the People Identifier contains multiple values,
  - Then the first People Identifier value that was added, will be used as the Run Book parameter value. This does not necessarily have to be the value that appears first in the list. For example, if a user is identified with a Windows user account and the person identifier has the values 'demo\jsmith' and 'resdemo\jsmith'. This is a known limitation.
- The use of global Variables in Run Book parameters, Run Book conditions and Run Book evaluators is not supported.

5.3.6 Run Book Job History tab

At **Library > Run Books**, you can configure and manage Run Books. With Run Books, you can create a chain of Jobs that makes it unnecessary to schedule these Jobs individually. When configuring Run Books, use the **Job History** tab to view details of Jobs in which the Run Book was used and to reschedule these Jobs if necessary.

**Configuration**

- To reschedule (part of) a Job, right-click the relevant Job and click **Reschedule Job**.

5.3.7 Run Book Versioning tab

At **Library > Run Books**, you can configure and manage Run Books. With Run Books, you can create a chain of Jobs that makes it unnecessary to schedule these Jobs individually. When configuring Run Books, use the **Versioning** tab to view a list of changes made to the Run Book, including all known versions of the Run Book.

**Configuration**

- The **Current version** field specifies the latest saved version number of the Run Book. It is possible to change this version number manually by clicking , regardless of the global setting **Use automatic versioning** (see page 48).
- The **Version** column specifies all versions of the Run Book.
- The **Comment** column specifies the comment that Console users made about individual versions.
- Use the drop-down field to customize the number of Audit Trail results that are shown per page.
- Use the navigation buttons to browse through the various Audit Trail pages.
5.3.8 Run Book Permissions tab

At Library > Run Books, you can configure and manage Run Books. With Run Books, you can create a chain of Jobs that makes it unnecessary to schedule these Jobs individually. When configuring Run Books, use the Permissions tab to view the level of access that Console users have to the Run Book.

Configuration

- Use the option Do not allow this Project to be scheduled programmatically to allow the Run Book from being invoked via the RES ONE Automation Console only. When selected, it will no longer be possible to invoke the Run Book:
  - During unattended installations of an Agent (by applying properties to its MSI file).
  - Via RES ONE Workspace.

5.3.9 Creating RES ONE Identity Director services based on Run Books

At Library > Run Books, you can create new services in RES ONE Identity Director (or RES ONE Service Store) based on Run Books and service templates using the RES ONE Identity Director Service Wizard. This wizard is available if you have configured and enabled RES ONE Identity Director Integration (at Setup > RES ONE Identity Director) AND your administrative role has Modify access to the Run Books node.

Configuration

- To create a new RES ONE Identity Director service based on an existing Run Book, right-click the Run Book and click Publish in RES ONE Identity Director from the context menu. This opens the RES ONE Identity Director Service Wizard. Follow the steps to configure a new service.
- Template and Properties step:
  - Select the RES ONE Identity Director service template on which the new service will be based. You can configure new service templates in the RES ONE Identity Director Management Portal.
  - The name of the new service must be unique in the RES ONE Identity Director environment.
- Category step:
  - Select the category in the service catalog in RES ONE Identity Director in which the new service will be placed.
- Qualification step:
  - Select the qualification criteria of the new service:
    - Use from template: The qualification criteria of the service template will be applied to the new service.
    - Everyone qualifies: Every person in the RES ONE Identity Director environment will qualify for the new service.
    - Base on organizational context: Select an Organizational Context item as qualification criterion. People must be classified in this item to qualify for the service.
- Summary step:
  - Review all configured settings for the new service.
  - When you click Create, all configured settings will be validated and a new service will be created.
When the new service is created in RES ONE Identity Director:

- Its delivery workflow contains as first action a **Provide Information** action, to ask for input for the Run Book parameters. As last action, it contains an action **Invoke Run Book** with the selected Run Book.
- Service attributes will be automatically created and linked to the Run Book parameters (if any).
- An exception handler will be added with an **End Workflow** action set to “Failed to deliver”.
- The service will be disabled by default, which allows for fine-tuning of the service before making it available in the RES ONE Identity Director environment.

**Note**

All communication of the RES ONE Identity Director Wizard to the Catalog Services of RES ONE Identity Director is SSL-encrypted, using port 8081.
5.4 Resources

Resources are files that are used in the execution of Tasks, such as MSI files, setup files, patches, hotfixes, etc. When you configure a Task, Resources can be used to install applications, security updates, etc. For example, if you create a Task Perform Unattended Installation to install an application, you first need to store the executable of the application so that RES ONE Automation can access it.

5.4.1 Managing Resources

At Library > Resources:

- On the Folders tab of the Resources node, you can create a folder structure (see page 170) for all your Resources and place the Resources that you have created in the designated folders.
  - You can drag-and-drop existing Resources into these folders. This allows you to easily locate any Projects you have created and also to create folders for various needs.
  - You can add, edit or delete folders by using the tree folder buttons on the lower right side of the screen.
  - If you delete a folder, its contents will be deleted as well.
- The Resources tab shows exactly the same list of Resources, but alphabetically listed and without the folder structure.
- To save an existing Resource to an external location (for example, a share or a USB drive), select it in the list and click Save Resource in the command bar. This functionality is only available for Resources that are stored in the Datastore and Resource packages (Resources stored on a file share are already stored on an external location).
- To scan for unused Resources, right-click either the Folders tab or the Resources tab and choose Scan for unused Resources. Because Resources in the Datastore can take up a lot of space, you can scan for Resources that are not referred to by any existing Tasks. The resulting list of unused Resources indicates whether a Resource has ever been used in a previous Job. You can safely delete all Resources that have never been used. Resources that have been used in past Jobs can also be deleted, but keep in mind that these Resources are required if you want to reapply the Job history of an Agent or when using Snapshot Intelligence. See Reapplying the Job History of Agents and Teams (on page 68) and Snapshot Intelligence (on page 197).
- If you leave the used Resources in the Agent cache after a Job has been executed, it takes up some disk space on the Agents. By default, a cached Resource will automatically be deleted from an Agent after two consecutive days of idle time.
- It is also possible to convert existing Resources from Stored in Datastore to Located on files share, and the other way round, for example to free up disk space on a file share. To convert a Resource, right-click it and click Convert in the context menu. Resources cannot be converted to or from the RES ONE Automation Resource Package format. The conversion and its settings are only applied to Resources to which they are applicable. For example, special actions are only set for Resources of the relevant storage method and file type.
- When saving changes to the Resource, the Version Control window will be opened. This window makes it possible to configure the versioning (see page 272) of changes to the Resource. The availability of this window depends on the global setting Use silent versioning.

When configuring a Resource:

- Use the Resource tab to select the file(s) that should be stored as a Resource.
- Use the Properties tab to view the properties of the selected file. This tab is only available for Resources that are stored in the Datastore or that are located on a file share.
- Use the Contents tab to configure the contents of the Resource package. This tab is only available for Resources that are stored as RES ONE Automation Resource Packages.
- Use the Usage tab to view in which Modules, Projects, and/or Run Books the Resource is used.
• Use the **Versioning** tab to view a list of changes made to the Run Book, including all known versions of the Run Book.

• Use the **Trusts** tab to define trusted Agents and Teams for the Resource. This tab is only shown if **Trusts Security** (see page 269) has been enabled.

• Use the **Permissions** tab to view the level of access that Console users have to the Resource.

### 5.4.2 Resource Folders

By grouping Resources in folders (At Library > Resources) you can:

• configure permissions on groups of Resources. See **Administrative Roles** ("Security" on page 213).

• configure trusted Agents and Teams for a Resource folder. See **Trusts** (see page 269).

#### Configuration

• When configuring a Resource folder, use the **Trusts** tab to define trusted Agents and Teams.

• Use the search bar to search for specific objects.

• You can use the following settings when configuring Trusts:

  • **Do not trust**: If this setting is selected, the selected item is not trusted. For example, if you are configuring Trusts on a specific Agent and specify that the Module Delete RES ONE Automation Results is not trusted by the Agent, the Module will fail when used in a Job that is executed by the Agent.

  • **Trust**: If this setting is selected, the selected item is trusted. This is the default setting. Depending on whether a full Trust exists with the item, it is possible to use the item in a Job. For example, if you are configuring Trusts on a specific Module and specify that Agent ONE is trusted by the Module, the Agent is allowed to use the Module in a Job.

  • **Inherit trust**: If no particular settings are selected (blank check box), the item will inherit the Trusts from the above lying item. Trusts that are inherited from an above lying item are grayed out and show the type and name of this item, except when this is the top-level folder: The Trusts of each top-level folder use the default setting (Trust). These Trusts are shown in black and it is not possible to change their settings.

• All items that are shown on the various **Trusts** tabs are subject to the access permissions of the administrative role(s) of the Console user.

• Select **Show only items that trust this** ... to filter on items that already trust the Agent, Team, Team folder, Module, Module folder, Resource or Resource folder that you are editing. When Trusts are enabled, Agents can only execute Jobs with Modules and Resources when a **full Trust** exists between them (i.e. the Module or Resource is trusted by the Agent AND the Agent is trusted by the Module or Resource). By using a filter, it becomes easier to create full Trusts.

  • Filtering the **Trusts** tab may take some time, especially in large environments with many Agents, Modules and Resources. When selecting the option, a popup window is therefore shown, showing the progress. If necessary, you can cancel the filtering.

  • If an item does not trust the item that you are editing, it will not be shown. Folders will be shown with an overlay icon.

  • This option requires Microsoft .NET Framework 4.0 (Full version) or higher to be installed on the machine that runs the Console.

---

**Note**

Availability of the **Trusts** tab depends on the global setting **Trusts Security**. See **Trusts** (see page 269).
5.4.3 Resource tab

At Library > Resources, you can manage Resources. Resources can contain any file that is needed to perform a certain Task, such as MSI files, setup files, patches, hotfixes, etc. When adding or editing Resources, use the Resource tab to select the file(s) that should be stored as a Resource.

A Resource can be one of the following types, each with their own advantages and disadvantages:

- **Stored in Datastore**: This type of Resource has the advantage that it is stored in the Datastore: as a result, it is always available and additional credentials are not necessary. However, only single files can be stored in this manner. Multiple-file installations must be stored on a file share or as a RES ONE Automation Resource Package.

- **Located on fileshare**: This type of Resource has the advantage that it is stored on a file share, and therefore bypasses Dispatchers when it is used in a Job. As a result, the Resource might be slightly more responsive. Although this is the fastest storage method, it can severely affect network traffic: Agents will download the Resource directly from the file share; it is not downloaded and cached by Dispatchers first. This storage method requires credentials to access the share each time the Task is run and the availability of the Resource also depends on the availability of the share. Multiple-file installations are possible. Choose this method if the Resource is too large to be stored in the Datastore or cached on Dispatchers (for example, OpenOffice or Microsoft Office 2010).

- **Located at URL**: Similar to Resources stored at a fileshare, this type of Resource makes it possible to refer to Resources that are stored at a certain Internet location. However, it can severely affect network traffic: Agents will download the Resource directly from the URL: it is not downloaded and cached by the Dispatchers first. The availability of the Resource depends on the availability of the Internet site. The functionality also supports URL redirection. This storage method can be used for Agents using Windows, Apple Mac OS X and Linux (only Red Hat, Suse, Ubuntu). For all Agents, except the Windows Agents, applies that a LibCurl library must be installed.

  - Use the Name field to specify a name for the Resource. The filename of the Resource is used as a back-up. When the filename cannot be retrieved from the URL header, it will use the filename from the Resource.
  
  - In the Download location field, URLs must start with "http://" or "https://" (not "file://" or ftp://).  

- **RES ONE Automation Resource Package**: This type of Resource is stored in the Datastore and can contain files and folders. The package is treated as a single Resource by RES ONE Automation and its contents are available for browsing and selecting. Therefore, if you set up a Task requiring a single file that is included in a RES ONE Automation Resource Package, you can expand the package and browse the contents to select the file. RES ONE Automation Resource Packages are typically used for software that is not contained in a single setup file.

When configuring a Resource:

- The Edit button allows you to edit any file type, as long as an editor is associated with it. This option is available for Resources stored in the Datastore or on a file share.

- You can disable a Resource. If a Resource is disabled, you can no longer add it to a Task. However, it is still enabled in existing Tasks.
5.4.4 Resource Properties tab

At Library > Resources, you can manage Resources. Resources can contain any file that is needed to perform a certain Task, such as MSI files, setup files, patches, hotfixes, etc. When configuring Resources, use the Properties tab to view the properties of the selected file. This tab is only available for Resources that are stored in the Datastore or that are located on a file share.

Configuration

If a Resource is Stored in the RES ONE Automation Datastore, the special action Parse environment variables, parameters and functions is available.

Parsing environment variables, parameters and functions

Certain file types can contain environment variables, parameters and functions. If these files are Resources stored in the Datastore, RES ONE Automation can replace the environment variables, parameters and functions in the contents of those files with the actual values that apply on the Agent when the Task that uses that file is executed. This option is available for all file types, as long as the file type can be edited with an associated editor. See Parameters (on page 257) and Functions (on page 246).

For example, an application may need an INI file to be located on all computers running the application. Each instance of the INI file must contain the actual computer name of the computer on which it is located. To set up this example:

1. Use the environment variable %computername% in the INI file.
2. Store the file as a Resource in the Datastore and select Parse variables, parameters and functions in contents of this file.
3. Use the Task Perform File Operations to place the INI file on the relevant Agents.

When the Task is run, each Agent that receives the file replaces the environment variable with the value that applies on that Agent - in this case, the computer name. As a result, each Agent holds an INI file that refers to its computer name.

For certain files, it can be desirable that only RES ONE Automation environment variables are replaced with actual values, not Microsoft Windows environment variables. With the option Skip parsing of environment variables, you can prevent this situation.

Example

The Resource example.ini in the Datastore contains:

```
[Test]
Path=%PROGRAMFILES%
Password=$[PASSWORD]
Guid=@[GUID]
Date=@[DATETIME(YYYYMMDD)]
```
After the Resource has been downloaded for use in a Task on a computer, it can contain:

[Test]
Path=C:\Program Files
Password=NeverGuess
Guid={86A1D38D-0627-493F-BCF1-8CC468BDB82E7}
Date=20090228

- In order to parse parameters in the contents of a file, those parameters must also be available as RES ONE Automation parameters on the Module, Project or Run Book that uses the Resource. For example, if the file contains the parameter $[USERNAME]$, then a Module parameter of the same name must exist.
- The option Parse variables, parameters and functions in contents of this file will replace these items with actual values, when the Resource is used in a Job.
  - When a Resource is downloaded, parameters and functions can be parsed.
  - When the Job is executed on a machine running Unix, Linux or Apple Mac OS X, environment variables will not be replaced. As a result, the option Skip parsing of environment variables also does not work on these machines.

5.4.5 Resource Contents tab

At Library > Resources, you can manage Resources. Resources can contain any file that is needed to perform a certain Task (see page 85), such as MSI files, setup files, patches, hotfixes, etc. When configuring RES ONE Automation Resource Packages, use the Contents tab to configure the contents of the Resource package. This tab is only available for Resources that are stored as RES ONE Automation Resource Packages.

Configuration
- Click Change to change the contents of the Resource Package.
- Use drag and drop to add files and folder directly from the Microsoft Windows Explorer. On computers running Microsoft Windows Vista or higher, this functionality requires User Account Control to be disabled or lowered to Notify me only when programs try to make changes to my computer (do not dim my desktop).
- If necessary, you can also rebuild the folder structure of a software package manually.
- Click Extract to extract a file in the Resource package to an external folder.

5.4.6 Resource Usage tab

At Library > Resources, you can manage Resources. Resources can contain any file that is needed to perform a certain Task (see page 85), such as MSI files, setup files, patches, hotfixes, etc. When configuring Resources, use the Usage tab to view in which Modules, Projects, and/or Run Books the Resource is used.

Configuration
- To jump to a Modules, Projects or Run Books, select it and click Open.
5.4.7 Resource Versioning tab

At Library > Resources, you can manage Resources. Resources can contain any file that is needed to perform a certain Task, such as MSI files, setup files, patches, hotfixes, etc. Use the Versioning tab to view a list of changes made to the Run Book, including all known versions of the Run Book.

Configuration
- The Current version field specifies the latest saved version number of the Run Book. It is possible to change this version number manually by clicking , regardless of the global setting Use automatic versioning (see page 48).
- The Version column specifies all versions of the Run Book.
- The Comment column specifies the comment that Console users made about individual versions.
- Use the drop-down field to customize the number of Audit Trail results that are shown per page.
- Use the navigation buttons to browse through the various Audit Trail pages.

5.4.8 Resource Trusts tab

Use the Trusts tab to define trusted Agents and Teams for a specific Resource. A Resource can have explicit Trusts or inherit its Trusts from its Resource folder. See Trusts (see page 269).

- Use the search bar to search for specific objects.
- You can use the following settings when configuring Trusts:
  - Do not trust: If this setting is selected, the selected item is not trusted. For example, if you are configuring Trusts on a specific Agent and specify that the Module Delete RES ONE Automation Results is not trusted by the Agent, the Module will fail when used in a Job that is executed by the Agent.
  - Trust: If this setting is selected, the selected item is trusted. This is the default setting. Depending on whether a full Trust exists with the item, it is possible to use the item in a Job. For example, if you are configuring Trusts on a specific Module and specify that Agent ONE is trusted by the Module, the Agent is allowed to use the Module in a Job.
  - Inherit trust: If no particular settings are selected (blank check box), the item will inherit the Trusts from the above lying item. Trusts that are inherited from an above lying item are grayed out and show the type and name of this item, except when this is the top-level folder: The Trusts of each top-level folder use the default setting (Trust). These Trusts are shown in black and it is not possible to change their settings.
  - All items that are shown on the various Trusts tabs are subject to the access permissions of the administrative role(s) of the Console user.
• Select Show only items that trust this ... to filter on items that already trust the Agent, Team, Team folder, Module, Module folder, Resource or Resource folder that you are editing. When Trusts are enabled, Agents can only execute Jobs with Modules and Resources when a full Trust exists between them (i.e. the Module or Resource is trusted by the Agent AND the Agent is trusted by the Module or Resource). By using a filter, it becomes easier to create full Trusts.

• Filtering the Trusts tab may take some time, especially in large environments with many Agents, Modules and Resources. When selecting the option, a popup window is therefore shown, showing the progress. If necessary, you can cancel the filtering.

• If an item does not trust the item that you are editing, it will not be shown. Folders will be shown with an overlay icon.

• This option requires Microsoft .NET Framework 4.0 (Full version) or higher to be installed on the machine that runs the Console.

---

Note

Availability of the Trusts tab depends on the global setting Trusts Security. See Trusts (see page 269).

---

5.4.9 Resource Permissions tab

At Library > Resources, you can manage Resources. Resources can contain any file that is needed to perform a certain Task (see page 85), such as MSI files, setup files, patches, hotfixes, etc. When configuring Resources, use the Permissions tab to view the level of access that Console users have to the Resource.
5.5 Variables

When configuring a Task, the values of many fields can optionally be replaced with Variables. Similar to parameters, Variables function as placeholders for customer-specific values, such as, for example, server names, passwords or credentials. Different from parameters however, Variables are resolved at the moment of Job execution, rather than at the moment of Job scheduling.

The global values of Variables can be set up at Library > Variables. These values are inherited by default by all Teams and Agents, but exceptions for individual Teams and Agents can be set up on the Team’s Team Variables tab and the Agent’s Agent Variables tab. Team Variables and Agent Variables allow “owners” of a specific Team or Agent to override the global Variables by specifying a custom-value. This makes Variables especially useful for use in multi-tenant RES ONE Automation sites, because it makes it easier to manage different settings for different customers and assigned projects, and to exchange Modules, Projects and Run Books with other branches of an organization.

Example

You can configure Variables to specify a Domain, Security Context and Domain controller for usage in a Task Manage Active Directory Computer. This creates a generic Task that can be used by all customers in your multi-tenant site. “Owners” of a specific Team or Agent at the customer site can override the global values of these Variables by specifying a custom-value that only applies to their site. When the Task is executed, the Task automatically gets the correct values that applies to the individual customer site. Depending on the hierarchy of these Variables, these can be the global values, Team-specific values or Agent-specific values.

Configuration

- Variables can be divided in categories. The category General is a default category and cannot be edited or removed.
- The name of a Variable or category must be unique in your RES ONE Automation environment.
- When configuring Variables, you can use the following types:
  - Credentials: this is a combination of domain\user name and password. Use this in a user name field for which there is also a password field, for example in the Security Context of a Task. The matching password field will be disabled, as this type of Variable will fill both. The password is always masked.
  - Password: this is a password only. Use this in password fields that are not part of a set of credentials, for example for the password for a new local user. The value of this type of Variable is always masked. Its input must be entered twice for confirmation.
  - Text: this is a text string of a single line. Use this in any field that needs to contain text, for example file name, Organizational Unit name, etc.
- The global values of Variables apply to all Teams and Agents, but can be overridden by Team-specific or Agent-specific values.
  - Agents can use Agent-specific values, inherit these values from their primary Team, or inherit them from the global values.
  - Teams can use Team-specific values, or inherit them from the global values. Nested Teams do not inherit their settings from the Team in which they are nested.
- Variables can only be added and deleted at Library > Variables; not from the Variables tab of a Team or Agent.
  - When a Variable is deleted, its value will no longer be resolved in the Tasks in which it is used, but becomes an actual value. For example, if you have used a Variable "^[CustomerName]" in a field in a Task, and delete this Variable without changing its value in this Task, the actual value of the field becomes "^[CustomerName]."
  - To create and insert a Variable in a field of a Task automatically, right-click the relevant field in the Task, click Insert Variables and select the Variable. Variables are divided by category in the context menu.
• Certain Tasks require the use of Variables, as placeholders for information that is needed for a successful execution of these Tasks. If the required Variables do not yet exist in your RES ONE Automation environment, a section will be available on the Settings tab of the Task, which contains fields to provide values for these Variables.

• When saving the configuration of the Task, the required Variables will be created automatically at Library > Variables. This makes it easier to configure these Tasks.

---

**Notes**

- When creating and importing a Building Block (see page 225) of Modules containing one or more Tasks in which Variables are used, the Variables and related categories will be saved and/or restored.
- When exporting Job results, all Agent Variables are included in the XML files, which can for example be used for Asset Management (CMDB, Content Management Database).
- Custom Properties from previous versions of RES ONE Automation are automatically migrated to Variables. Because names of categories in Variables must be unique, any categories in Custom Properties with the same name will be merged into one category in Variables and any existing Custom Properties objects in these categories will be categorized in this category.
- It is not possible to base the value of a Variable on another Variable, a parameter or a function. These values cannot be resolved.
- The use of global Variables in Run Book parameters, Run Book conditions and Run Book evaluators is not supported.
Chapter 6: Topology

The Topology of a RES ONE Automation environment consists of:

- Consoles (on page 178)
- Dispatchers (on page 180)
- Agents (on page 193)
- Teams (on page 205)

6.1 Consoles

The Console is the application from which RES ONE Automation is configured and used, from setting up the RES ONE Automation environment to the execution of Tasks. A RES ONE Automation environment can contain several Consoles. Each Console can connect to different RES ONE Automation sites, but not simultaneously.

Configuration

At Topology > Consoles:

- You can deploy additional Consoles to create multiple points of administration of your RES ONE Automation environment. Before you deploy new Consoles, ensure that the target computers meet the prerequisites. Please refer to the Getting Started with RES ONE Automation.
- You can also deploy Consoles manually (see page 221) or unattended (see page 297).
- If Consoles are identified by MAC address or by Domain name/NetBIOS name, this information is shown in the Console Properties, in the field Identified by. This field is not shown if components are identified by their GUID. You can change the way in which components are identified in the Global Settings (on page 35).

Notes

- Consoles will automatically restart if they fail.
- It is not possible to create a new Datastore from an additional Console (from a computer on which only a Console has been installed, not a full installation of RES ONE Automation).
- Enabling or disabling the Auto-Refresh option to automatically update items affects ALL nodes in which this functionality is available (all Jobs and Topology nodes).
## Additional Console logging

The installation of RES ONE Automation also writes important information into the Windows registry of the computer at HKEY_LOCAL_MACHINE level:

<table>
<thead>
<tr>
<th>Registry Key</th>
<th>String Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>HKEY_LOCAL_MACHINE\SOFTWARE\RES\AutomationManager\Preferences (32-bit)</td>
<td>WUID</td>
</tr>
<tr>
<td>HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\RES\AutomationManager\Preferences (64-bit)</td>
<td></td>
</tr>
<tr>
<td>HKEY_LOCAL_MACHINE\SOFTWARE\RES\AutomationManager\WMC (32-bit)</td>
<td>DBServer</td>
</tr>
<tr>
<td>HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\RES\AutomationManager\WMC (64-bit)</td>
<td></td>
</tr>
<tr>
<td>HKEY_LOCAL_MACHINE\SOFTWARE\RES\AutomationManager\WMC (32-bit)</td>
<td>DBType</td>
</tr>
<tr>
<td>HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\RES\AutomationManager\WMC (64-bit)</td>
<td></td>
</tr>
<tr>
<td>HKEY_LOCAL_MACHINE\SOFTWARE\RES\AutomationManager\WMC (32-bit)</td>
<td>DBName</td>
</tr>
<tr>
<td>HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\RES\AutomationManager\WMC (64-bit)</td>
<td></td>
</tr>
<tr>
<td>HKEY_LOCAL_MACHINE\SOFTWARE\RES\AutomationManager\WMC (32-bit)</td>
<td>DBUser</td>
</tr>
<tr>
<td>HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\RES\AutomationManager\WMC (64-bit)</td>
<td></td>
</tr>
<tr>
<td>HKEY_LOCAL_MACHINE\SOFTWARE\RES\AutomationManager\WMC (32-bit)</td>
<td>DBPassword</td>
</tr>
</tbody>
</table>

### Notes

- The Console logs all updates and errors by default. These files are not created during installation, but at the first time an event happens. The names of the log file are wmc.log and wmc_update.log. The log files will be created in the installation directory of RES ONE Automation.
- When closing the Console, its position is stored. As a result, the next time the Console is opened, it will open at the position it had before it was closed. This increases usability, as it becomes easier to navigate the Console. For example, if a specific Module was selected in the node Library > Modules before the Console was closed, the next time the Console will open at this Module. If the node Jobs > Scheduling was selected, the Console will open at this node. Any positions in the nodes Administration > Audit Trail and Jobs > Job Results are not stored. The position of the Console is stored as a FIPS-compliant encrypted file for each individual user on the machine on which the Console is running, at:
  - %userprofile%\Local Settings\Application Data\RES\AM\cs_<guid of environment>.bin (Microsoft Windows XP and Windows Server 2003)
  - %userprofile%\AppData\Local\RES\AM\cs_<guid of environment>.bin (Windows Vista, Windows 7, Windows Server 2008 and higher).
6.2 Dispatchers

Dispatchers communicate information from the Datastore to Agents. Dispatchers contact the Datastore at regular intervals. If new Jobs are available for Agents to which it connects, the Dispatcher will download all necessary data from the Datastore and store it. The Agents will use this data to execute the Jobs.

- A RES ONE Automation environment can contain several Dispatchers. The size of your RES ONE Automation environment determines the number of Dispatcher needed: in test situations, a number of 9000 concurrent connections was reached. However, the actual maximum number of concurrent connections depends on the type of Jobs that are executed and on the hardware properties of the machine on which the Dispatcher is running.
- **Master caching** (on page 181) makes it possible for Dispatchers to share a common Resource cache on a specific environment: by assigning a master caching Dispatcher, only this Dispatcher is allowed to download Resources from the Datastore.

**Configuration**

At Topology > Dispatchers:

- You can deploy a Dispatcher from the Console. Before you deploy new Dispatchers, ensure that the target computers meet the prerequisites. Please refer to the Getting Started with RES ONE Automation. You can also deploy Dispatchers manually ("Install Dispatchers manually" on page 221) or unattended (see page 298).
- The column Connections (poll|push) shows the number of currently connected Agents or Agents+, and the column Peak (poll|push) shows the maximum number of Agents or Agents+ that have connected concurrently to each Dispatcher. The first value relates to the poll communication between Dispatcher and Agent or Agent+ using port 3163. The second value relates to the push communication between Dispatcher and Agent+ using port 3162 (or configured differently). The statistics are updated approximately every thirty seconds if the Dispatcher is active. Every time the Dispatcher service is restarted, the counter is reset.
- The column Roles shows which Dispatcher role is assigned:
  - **Housekeeping** role (Housekeeping tasks of Dispatcher (see page 188))
  - **Licensing** role (Reporting to RES ONE License Server (on page 189))
  - **Telemetry** role (Telemetry tasks of Dispatcher (on page 190))
- If Dispatchers are identified by MAC address or by Domain name/NetBIOS name, this information will be shown in the Dispatcher Properties, in the field Identified by. This field is not shown if components are identified by their GUID. You can change the way in which components are identified in the Global Settings.
- The availability of a Web API (on page 183) provides full support for building and deploying RESTful and SOAP web services. This allows you to integrate Job scheduling in external systems.

**Notes**

- Dispatcher services will automatically restart if they fail.
- If a Dispatcher is offline, a warning sign will be shown in the Dispatchers node.
- The Dispatcher logs all events in the Event log, which can be viewed with for example the Event Viewer that is part of the Microsoft Windows Operating System.
- If the bandwidth in your RES ONE Automation environment is limited, you can set up master caching. By assigning a master caching Dispatcher, only this Dispatcher is allowed to download Resources from the Datastore. Other Dispatchers will download Resources from the master caching Dispatcher. This reduces bandwidth consumption between the Dispatchers and the Datastore.
- Enabling or disabling the Auto-Refresh option to automatically update items affects ALL nodes in which this functionality is available (all Jobs and Topology nodes).
6.2.1 Dispatcher settings

New Dispatchers that are deployed from the Console always take on the Global Settings (Setup > Global Settings). If necessary, you can edit these settings for individual Dispatchers after their deployment.

You can change the settings of Dispatchers in different ways:

- Open the properties of an individual Dispatcher and click the Settings tab.
- Right-click a Dispatcher and click Settings Overview. This opens the Settings Overview window, which shows an overview of the settings of all Dispatchers. To change these settings, select one or more Dispatchers and click Change settings of selected Dispatchers.
- Select the relevant Dispatchers, right-click this selection and click Change settings of selected Dispatchers. This opens the Change settings for multiple Dispatchers window.
  - To change a setting, select it and click Edit.
  - After changing the setting, its check box will be selected. To ignore a changed setting, clear this check box.

Except for the settings Housekeeping, Licensing and Exclude from Dispatcher list, you can choose for each setting whether to use Dispatcher specific settings or global settings.

You can assign the Housekeeping role to one or more Dispatchers so you have more control on which Dispatcher the housekeeping tasks (see page 188) are performed. Only one Dispatcher will be responsible for running the housekeeping tasks at one time. For performance reasons it is advised to choose a Dispatcher near the Datastore.

You can assign the Licensing role to one or more Dispatchers so you will have control over which Dispatcher will report the license peak usage to the RES ONE License Server (see page 189).

6.2.2 Master caching

Resources are files that are used in the execution of RES ONE Automation Tasks and can be stored in the Datastore. For example, a Task Perform Unattended Installation may use an application executable that is stored as a Resource in the Datastore. Depending on the global settings in your RES ONE Automation environment, each Dispatcher caches this Resource, so that it can be downloaded by Agents to execute a certain Task. This can be a problem if the bandwidth in your RES ONE Automation environment is limited: if many Dispatchers connect to the Datastore simultaneously to download the Resource to their cache, the download speed will decrease and the maximum bandwidth capacity may be reached, especially if the Resource is large.

Master caching makes it possible for Dispatchers to share a common Resource cache on a specific environment: by assigning a master caching Dispatcher, only this Dispatcher is allowed to download Resources from the Datastore. After downloading the Resources from the Datastore, the master caching Dispatcher stores them in its cache, after which other Dispatchers can download the Resources from the master caching Dispatcher. If you “redirect” the other Dispatchers to this master caching Dispatcher, they can only download Resources from the master caching Dispatcher, because they can no longer contact the Datastore directly. This reduces bandwidth consumption between the Dispatchers and the Datastore, and so improves the reliability of the connection with the Datastore and increases the overall speed at which Resources are downloaded.
Master caching is illustrated in the following figure:

Set up master caching Dispatcher

When setting up master caching, RES recommends to make a master caching Dispatcher exclusively available to Dispatchers: this prevents Agents from being able to contact the master caching Dispatcher. Keep in mind though that this may require additional Dispatchers, because any Agent will be unable to contact the master caching Dispatcher.

Configure different port

By default, RES ONE Automation uses port 3163 for communication between Agents and Dispatchers. Via the registry settings, you can specify a different port on the master caching Dispatcher (for example, 3164). This makes the master caching Dispatcher unavailable for Agents. It may still be reached by other Dispatchers, provided you have specified this on these Dispatchers.

Redirect the other Dispatchers

Via the registry settings, you can also specify on a Dispatcher to which master caching Dispatcher it should redirect to download Resources. If you specify multiple master caching Dispatchers, the next master caching Dispatcher in the list will be used if a certain master caching Dispatcher is offline or for other reasons unavailable.

See also Registry settings for Master caching Dispatcher (on page 305).
6.2.3 Web API

The main function of a Dispatcher is to communicate information from the Datastore to the Agents. The Dispatcher also offers the availability of a Web API. This allows system developers to build and develop their own RESTful and SOAP web services and to integrate Job scheduling in external systems. Communication with the Web API is secured with the regular login accounts of your RES ONE Automation environment.

This chapter provides basic information about how to get started with the Dispatcher Web API. To use the Web API, you should have a basic familiarity with software development and RESTful and SOAP Web services.

Authentication

- Web API authentication will only be applied if there are one or more Logins defined in RES ONE Automation (at Administration > Security).
- Only accounts using RES ONE Automation Authentication, and that have a password defined, can be used to access the Web API.

Configuration

The Dispatcher Web API is disabled by default. You can enable it by enabling the global setting WebAPI state (at Setup > Global Settings > Dispatcher WebAPI section) or the setting on Dispatcher-level (Topology > Dispatchers > Settings > Dispatcher WebAPI section). When using the Dispatcher Web API in your RES ONE Automation environment, it is also possible to secure communication using HTTPS (see page 187). It is not necessary to manually configure these settings in the Dispatcher configuration file. The settings in the Dispatcher configuration file will be ignored when deploying the Dispatcher.

To access the Web API, use the following base address in your browser as a basis for Job scheduling in an external system or client:

- Enabled: http://[host name]/Dispatcher/SchedulingService/
- SSL enabled: https://[host name]/Dispatcher/SchedulingService/

The host name specifies the name or IP address of the computer that hosts the Dispatcher. If you are hosting the web services on a different address (URL) and/or port number, specify an alternative base address and/or custom port number in the Dispatcher WebAPI setting.

- When prompted for a username and password, use the credentials of an existing login account in your RES ONE Automation environment. This needs to be an account using RES ONE Automation Authentication, and that has a password defined. If your RES ONE Automation environment does not contain any login accounts, you can leave the fields empty.

- The Help for the Web API is available at: http://[host name]/Dispatcher/SchedulingService/help. This page gives an overview of all available resources and actions. All actions correspond to the functionality in the Jobs nodes of the Console.

<table>
<thead>
<tr>
<th>Schedule a new Job</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>URL</strong></td>
</tr>
<tr>
<td><strong>Method</strong></td>
</tr>
<tr>
<td><strong>Parameter:</strong></td>
</tr>
<tr>
<td>-</td>
</tr>
</tbody>
</table>

To schedule a new Job, send an XML message with the new Job settings. See the Web API Help for the message format.
## Retrieve a list of Jobs

**URL**

```
http://[host name]/Dispatcher/SchedulingService/jobs?stage={STAGE}&page={PAGE}&pagesize={PAGESIZE}&runbookjobs={INCLUDERUNBOOKJOBS}&recurringjobs={INCLUDERECURRINGJOBS}
```

**Method**

GET

**Parameter:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>STAGE</td>
<td>Specifies which Jobs should be shown. Values: All, Scheduled, Active, History.</td>
</tr>
<tr>
<td>PAGE</td>
<td>Specifies which page of the overview of Jobs should be shown. Values: 1, 2, 3, etc.</td>
</tr>
<tr>
<td>PAGESIZE</td>
<td>Specifies how many Jobs should be shown on the page. Values: 10, 20, ..., 100.</td>
</tr>
<tr>
<td>INCLUDERUNBOOKJOBS</td>
<td>Specifies whether to include Run Book Jobs in the overview. Values: true, false.</td>
</tr>
<tr>
<td>INCLUDERECURRINGJOBS</td>
<td>Specifies whether to include recurring Jobs in the overview. Values: true, false.</td>
</tr>
</tbody>
</table>

## Retrieve Job details

**URL**

```
http://[host name]/Dispatcher/SchedulingService/jobs/{JOBID}
```

**Method**

GET

**Parameter:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOBID</td>
<td>Specifies the Job GUID (for example, {7E9C07B8-C5E6-4F60-A01B-F096F98E5919}).</td>
</tr>
</tbody>
</table>

When retrieving the Job details, also the parameters of the specific Modules and Projects are shown.

## Abort a Job

**URL**

```
http://[host name]/Dispatcher/SchedulingService/jobs/{JOBID}
```

**Method**

PUT

**Parameter:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOBID</td>
<td>Specifies the Job GUID (for example, {7E9C07B8-C5E6-4F60-A01B-F096F98E5919}).</td>
</tr>
</tbody>
</table>

To abort a Job, send an XML message with the new Job settings. This sends the Job status Aborting to the Job. See the Web API Help for the message format.
### Retrieve Job results

<table>
<thead>
<tr>
<th>URL</th>
<th>http://[host name]/Dispatcher/SchedulingService/jobs/{JOBID}/results?overview={INCLUDEOVERVIEW}&amp;detailed={INCLUDEDETAILED}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>GET</td>
</tr>
<tr>
<td>Parameter:</td>
<td>Explanation:</td>
</tr>
<tr>
<td>• JOBID</td>
<td>Specifies the Job GUID (for example, {7E9C07B8-C5E6-4F60-A01B-F096F98E5919}).</td>
</tr>
<tr>
<td>• INCLUDE OVERVIEW</td>
<td>Specifies whether to include an overview of the results of the Job. Values: true, false.</td>
</tr>
<tr>
<td>• INCLUDE DETAILED</td>
<td>Specifies whether to include a detailed overview of the results of the Job. Values: true, false.</td>
</tr>
</tbody>
</table>

**Note**

When exporting the Job results, the document will be wrapped in a soap document. This Job results document is base64-encoded and added to the soap document. As a result, the document needs to be decoded with a base64 decoder first before it can be read.

### Report on Team composition

<table>
<thead>
<tr>
<th>URL</th>
<th>http://[host name]/Dispatcher/SchedulingService/report/teams</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>GET</td>
</tr>
<tr>
<td>Parameter:</td>
<td>Explanation:</td>
</tr>
<tr>
<td>-</td>
<td>The report will only show the resulting Agents from Teams the user has Read or Modify access to.</td>
</tr>
</tbody>
</table>

### Report on Job usage

<table>
<thead>
<tr>
<th>URL</th>
<th>http://[host name]/Dispatcher/SchedulingService/report/jobusage?start={START}&amp;stop={STOP}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>GET</td>
</tr>
<tr>
<td>Parameter:</td>
<td>Explanation:</td>
</tr>
<tr>
<td>• START</td>
<td>Specifies the date time when the Job was started.</td>
</tr>
<tr>
<td>• STOP</td>
<td>Specifies date time when the Job was finished.</td>
</tr>
</tbody>
</table>

When specifying the date time, use the format yyyyMMddHHmmss:

- yyyy: Year (e.g. 2013)
- MM: Month (e.g. 03)
- dd: Day (e.g. 27)
- HH: Hour (e.g. 19 - use 24h notation)
- mm: Minute (e.g. 59)
- ss: Seconds (e.g. 30)
### Retrieve Job details of a Run Book Job

**URL**  
http://[host name]/Dispatcher/SchedulingService/runbookjobs/{JOBID}

**Method**  
GET

**Parameter:**  
Explanation:

- **JOBID**  
  - Specifies the Job GUID (for example, `{7E9C07B8-C5E6-4F60-A01B-F096F98E5919}`).

### Retrieve a list of input parameters for a Module, Project or Run Book

**URL**  
http://[host name]/Dispatcher/SchedulingService/what/{WHATTYPE}s/{WHATID}/inputparameters

**Method**  
GET

**Parameter:**  
Explanation:

- **WHATYPE**  
  - Specifies what was executed. Values: RunBook, Project, Module.
- **WHATID**  
  - Specifies the GUID of the Run Book, Project, or Module.

### Retrieve a list of available Modules

**URL**  
http://[host name]/Dispatcher/SchedulingService/what/modules

**Method**  
GET

**Parameter:**  
Explanation:

- -

### Retrieve a list of available Projects

**URL**  
http://[host name]/Dispatcher/SchedulingService/what/projects

**Method**  
GET

**Parameter:**  
Explanation:

- -

### Retrieve a list of available Run Books

**URL**  
http://[host name]/Dispatcher/SchedulingService/what/runbooks

**Method**  
GET

**Parameter:**  
Explanation:

- -
## Retrieve a list of available Agents/Teams

<table>
<thead>
<tr>
<th>URL</th>
<th>http://[host name]/Dispatcher/SchedulingService/who</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>GET</td>
</tr>
<tr>
<td>Parameter:</td>
<td>Explanation:</td>
</tr>
<tr>
<td>• INCLUDEAGENTS</td>
<td>Specifies whether to include Agents in the selection of Agents/Teams. Values: true, false.</td>
</tr>
<tr>
<td>• INCLUDETEAMS</td>
<td>Specifies whether to include Agents in the selection of Agents/Teams. Values: true, false.</td>
</tr>
<tr>
<td>• ONLINEONLY</td>
<td>Specifies whether to show online Agents only. Values: true, false.</td>
</tr>
<tr>
<td>• LICENSEDONLY</td>
<td>Specifies whether to show licensed Agents only. Values: true, false.</td>
</tr>
<tr>
<td>• SEARCHFOR</td>
<td>Specifies the Agents to search for: [Agent name].</td>
</tr>
</tbody>
</table>

### Notes
- In releases prior to RES Automation Manager 2014, the Web API could be enabled on individual Dispatchers by setting the registry value HKLM\SOFTWARE\RES\AutomationManager\Dispatcher\WebAPI. This registry value has become obsolete. Web APIs that were enabled with the registry value in previous releases will be disabled in RES Automation Manager 2014 and higher. They can be enabled again with the global setting WebAPI state.
- When scheduling a Run Book to an Agent from RES ONE Identity Director, it is possible to use the Agent GUID and the Agent name as a unique identifier. The Agent GUID is only supported when communicating via the Web API and the command line. The RES ONE Automation Console does not accept the Agent GUID.
Using HTTPS

When using the Dispatcher Web API in your RES ONE Automation environment, it is possible to secure communication using the HTTPS protocol. If the Datastore is configured to use AES-256 encryption, HTTPS is used by default.

By default, the Dispatcher Web API uses an SSL certificate to communicate over HTTPS. This certificate is untrusted, which means that a security exception needs to be configured, to bypass any security warnings. The SSL certificate can be viewed in the list of certificates by searching on "certificate issuer: "RES AM - DISPATCHER SERVICE"" or "certificate issued by "RES AM - DISPATCHER SERVICE".

It is also possible to use an SSL certificate that has been issued by an official Certification Authority (CA). The SSL certificate must be installed on the Dispatcher, in a certificate store in LocalComputer (not CurrentUser).

Configuration

To configure the Dispatcher Web API to use HTTPS, select the option SSL enabled for the global or Dispatcher-specific setting WebAPI state (see page 42) and specify the Port number. The default port number for SSL is 443. It is possible to specify a custom port number. It is not necessary to manually configure these settings in the Dispatcher configuration file. The settings in the Dispatcher configuration file will be ignored when deploying the Dispatcher.

When using an SSL certificate that was issued by a CA, the SSL certificate thumbprint and the SSL certificate store that contains the certificate are also required.

Notes

It is not possible to use the HTTPS protocol to secure communication of the Dispatcher Web API for Dispatchers running on Microsoft Windows Server 2003 and Windows XP machines.

Known limitations

If a Dispatcher with the WebAPI enabled is installed on an IIS webserver, and both the Dispatcher and IIS use HTTPS over port 443, the existing site bindings in IIS may be overwritten by the Dispatcher.
To prevent this, please configure the WebAPI to use a different port.
6.2.4 Housekeeping tasks of Dispatcher

At Topology > Dispatchers, you can assign the Housekeeping role to one or more Dispatchers, so you will have more control on which Dispatcher the housekeeping tasks will be executed.

Housekeeping tasks include:

- Setting components (Consoles, Dispatchers and Agents) offline when there has not been any activity for at least five minutes.
- Cancel Jobs that have been scheduled to Agents that have been removed.
- Timeout of Jobs that have been running longer than the defined timeout period.

Only one Dispatcher will be responsible for running the housekeeping tasks at one time. For performance reasons, it is advised to choose a Dispatcher near the Datastore.

Configuration

At Topology > Dispatchers, on the Settings tab of the Dispatcher’s properties, you enable the setting Housekeeping to assign the Housekeeping role to the selected Dispatcher. To enable the setting for multiple Dispatchers, you can also use the option Change settings of selected Dispatchers from the context menu or settings overview. The same procedure applies to disable the Housekeeping role.

Execution of housekeeping tasks

The Dispatcher will check every 60 seconds if it needs to take over the housekeeping tasks. Depending on the configuration of the Dispatchers, several scenarios are possible:

- One or more Dispatchers have been assigned the Housekeeping role and (some of them) are online: The first available Dispatcher with the Housekeeping role will run or take over the housekeeping tasks.
- All Dispatchers that have been assigned the Housekeeping role are offline: The first available Dispatcher will run or take over the housekeeping tasks.
- No Dispatchers have been assigned the Housekeeping role: The first available Dispatcher will run or take over the housekeeping tasks.

Overview Housekeeping role

Depending on the scenario a different value will be shown in the column Roles in the Dispatchers overview. That way you can see which of the Dispatchers are currently performing the housekeeping tasks. The values for this column are:

<table>
<thead>
<tr>
<th>Dispatcher role</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;empty&gt;</td>
<td>The Dispatcher is not assigned any role and is not performing housekeeping tasks.</td>
</tr>
<tr>
<td>Housekeeping</td>
<td>The Dispatcher is assigned the Housekeeping role, but currently not performing housekeeping tasks.</td>
</tr>
<tr>
<td>Housekeeping*</td>
<td>The Dispatcher is assigned the Housekeeping role and is performing housekeeping tasks.</td>
</tr>
<tr>
<td>(Housekeeping)*</td>
<td>The Dispatcher is not assigned the Housekeeping role, but is performing housekeeping tasks.</td>
</tr>
</tbody>
</table>

Please note that only one Dispatcher will show the * at one time.
6.2.5  Reporting to RES ONE License Server

Note
This feature becomes available when the RES ONE License Server is released.

At **Topology > Dispatchers**, you can assign the **Licensing** role to one or more Dispatchers, so you will have control over which Dispatcher will communicate with the RES ONE License Server. Only one Dispatcher will be responsible for collection and reporting the peak usage of Datacenter Server and Desktop ASP and Corporate licenses at one time.

**Configuration**

At **Topology > Dispatchers**, on the **Settings** tab of the Dispatcher's properties, you enable the setting **Licensing** to assign the **Licensing** role to the selected Dispatcher. To enable the setting for multiple Dispatchers, you can also use the option **Change settings of selected Dispatchers** from the context menu or settings overview. The same procedure applies to disable the **Licensing** role.

**Execution of collecting and/or reporting task**

The Dispatcher will check every 60 seconds if it needs to take over the collecting and/or reporting task. Depending on the configuration of the Dispatchers, several scenarios are possible:

- One or more Dispatchers have been assigned the **Licensing** role and (some of them) are online: The first available Dispatcher with the Licensing role will run or take over the collecting and/or reporting task.
- All Dispatchers that have been assigned the **Licensing** role are offline: The first available Dispatcher will run or take over the **collecting** task. The reporting task cannot be done. A warning message will be shown when the Console user logs in.
- No Dispatchers have been assigned the **Licensing** role: The first available Dispatcher will run or take over the **collecting** task. The reporting task cannot be done. A warning message will be shown when the Console user logs in.

See **License usage collection and reporting** (on page 29) for more information on calculation, timing and sequence of tasks.

**Overview Licensing role**

Depending on the scenario a different value will be shown in the column **Roles** in the Dispatchers overview. That way you can see which of the Dispatchers are currently performing the reporting tasks. The values for this column are:

<table>
<thead>
<tr>
<th>Dispatcher role</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;empty&gt;</td>
<td>The Dispatcher is not assigned any role and is not performing any tasks.</td>
</tr>
<tr>
<td>Licensing</td>
<td>The Dispatcher is assigned the Licensing role, but currently not performing the collecting and/or reporting task.</td>
</tr>
<tr>
<td>Licensing*</td>
<td>The Dispatcher is assigned the Licensing role and is performing collecting and/or reporting task.</td>
</tr>
<tr>
<td>(Licensing)*</td>
<td>The Dispatcher is not assigned the Licensing role, but is performing the collecting task.</td>
</tr>
</tbody>
</table>

Please note that only one Dispatcher will show the * at one time.
6.2.6 Telemetry tasks of Dispatcher

At Topology > Dispatchers, you can assign the Telemetry role to one or more Dispatchers, so you will have control on which Dispatcher the telemetry tasks will be executed. Only one Dispatcher will be responsible for running the telemetry tasks at one time. For performance reasons, it is advised to choose a Dispatcher near the Datastore. The Dispatcher needs Internet access to communicate with the telemetry web server.

Telemetry tasks include:

- Collect and report back on:
  - How many Agents and Agents+ are deployed
  - Which Tasks are executed and how many times

Please refer to the RES ONE Automation Administration Guide for a complete list of collected data and an illustration of the data provided to RES.

Important: The Telemetry functionality enables software and hardware information related to the RES product and/or the environment in which the product is utilized, to be collected and provided to the RES Research and Development team. This information may contain aggregate or statistical information about the use of the RES product and the type of system(s) and related database(s) in which the RES product is deployed. This data contains no information that could reasonably be used by RES to directly identify a person. Installing this version of RES ONE Automation that includes the Telemetry functionality means that you allow RES to receive and use this information for research, analytics and improvement of its products. The information provided to RES is considered non-confidential.

Configuration

At Topology > Dispatchers, on the Settings tab of the Dispatcher’s properties, you enable the setting Telemetry to assign the Telemetry role to the selected Dispatcher. To enable the setting for multiple Dispatchers, you can also use the option Change settings of selected Dispatchers from the context menu or settings overview. The same procedure applies to disable the Telemetry role.

Perform telemetry tasks

When the Dispatcher starts, it tries to contact the telemetry web server to check for an Internet connection. If the connection has been established successfully, the Dispatcher will check every 5 minutes if it needs to run or take over the telemetry task. If no connection could be established, the Dispatcher will wait until the Dispatcher restarts and try again. Depending on the configuration of the Dispatchers, several scenarios are possible:

- One or more Dispatchers have been assigned the Telemetry role and (some of them) are online: The first available Dispatcher with the Telemetry role will run or take over the telemetry task.
- All Dispatchers that have been assigned the Telemetry role are offline: The first available Dispatcher will run or take over the reporting task.
- No Dispatchers have been assigned the Telemetry role: The first available Dispatcher will run or take over the telemetry task.

Once every 30 days, one Dispatcher will do the collection just before sending the data (if a successful Internet connection could be made). The telemetry action is logged in the Audit Trail.

Overview Telemetry role

Depending on the scenario a different value will be shown in the column Roles in the Dispatchers overview. That way you can see which of the Dispatchers are currently performing the telemetry tasks. The values for this column are:
### Dispatcher role

<table>
<thead>
<tr>
<th>Dispatcher role</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;empty&gt;</td>
<td>The Dispatcher is not assigned any role and is not performing telemetry tasks.</td>
</tr>
<tr>
<td>Telemetry</td>
<td>The Dispatcher is assigned the Telemetry role, but currently not performing telemetry tasks.</td>
</tr>
<tr>
<td>Telemetry*</td>
<td>The Dispatcher is assigned the Telemetry role and is performing telemetry tasks.</td>
</tr>
<tr>
<td>(Telemetry)*</td>
<td>The Dispatcher is not assigned the Telemetry role, but is performing telemetry tasks.</td>
</tr>
</tbody>
</table>

Please note that only one Dispatcher will show the * at one time.
6.3 Agents

An Agent is a service running on a computer, and can execute Tasks on networked Microsoft Windows desktops and servers. This can be on the same computer on which the Agent runs, but also on another computer (for example, when executing a Task Execute Secure Shell (SSH) Commands or a Task Create Exchange Mailbox.

Prerequisite

For the Agent to function properly, 8.3 naming convention must be used. By default, Microsoft Windows has 8.3 naming convention enabled, which means that when an application is installed that uses a long path or filename, Microsoft Windows will create an 8.3 name for it. However, it is possible to disable 8.3 names. This means that Microsoft Windows will not create an 8.3 name for long path or file names.

Configuration

At Topology > Agents:

- You can deploy an Agent+ (on page 202) from the Console. Before you deploy new Agents+, ensure that the target computers meet the prerequisites. Please refer to the Getting Started with RES ONE Automation. You can also deploy Agents+ manually (see page 222), unattended (see page 295) or embedded (see page 196).
- If you deploy an Agent+ on the same computer where a legacy Agent is installed, the Agent will be converted to the Agent+. The Agent+ will automatically adopt the Agent settings, and the legacy Agent will be removed from the system. Please note that if you used the default Agent resource folder for caching Resources and you installed the Agent on a custom location, the cached Resources will be removed after the convert.
- To deploy the legacy Agent, use the MSI file and install it manually. It is not possible to install the legacy Agent, if the Agent+ is already installed on the same machine.
- Agents for Mac OS X (see page 204) and Agents for Unix/Linux (see page 203) cannot be deployed from the Console, but need to be installed separately.
- If Agents are identified by MAC address or by Domain name/NetBIOS name, this information will be shown in the Agent Properties, in the field Identified by. This field is not shown if components are identified by their GUID. You can change the way in which components are identified in the Global Settings.
- If an Agent is offline because the computer on which it is installed is turned off, you can use wake-on-LAN to turn on the computer and bring the Agent online. To use wake-on-LAN, right-click the Agent and choose Wake-on-LAN. Please note that this feature only works if the Agent and your network infrastructure support wake-on-LAN.
- If you want to delete obsolete Agents that have the same name as an online Agent, please use incorrect credentials if you want to keep this Agent: during deletion RES ONE Automation will use the Agent name to resolve the machine on which the obsolete Agents are installed, including the one with the online Agent. As a result, the online Agent will also be deleted.
- If you repair an installed Agent or Agent+, the current Agent settings will be used. The component is replaced like for like (either the Agent, or the Agent+). A legacy Agent will not be converted to an Agent+.
The File version for the Agent+ follows the Microsoft standard: <Major>.<Minor>.<Days>.<Seconds>. For example, File Version 10.0.6205.36370 for the Agent+. The File version for the legacy Agent follows the standard: <Major>.<SR>.<incremental number>. For example, File Version 7.05.0577 for the legacy Agent.

Notes
- Agent services will automatically restart if they fail.
- A deployed Agent will only get online if it can connect to a Dispatcher.
- Enabling or disabling the Auto-Refresh option to automatically update items affects ALL nodes in which this functionality is available (all Jobs and Topology nodes).

6.3.1 Agent settings

New Agents that are deployed from the Console always take on the Global Settings (Setup > Global Settings). If necessary, you can edit these settings for individual Agents after their deployment.

You can change the settings of Agents in different ways:
- Open the properties of an individual Agent and click the Agent Settings tab.
- Right-click an Agent and click Settings Overview. This opens the Settings Overview window, which shows an overview of the settings of all Agents. To change these settings, select one or more Agents and click Change settings of selected Agents.
- Select the relevant Agents, right-click this selection and click Change settings of selected Agents. This opens the Change settings for multiple Agents window.
  - To change a setting, select it and click Edit.
  - After changing the setting, its check box will be selected. To ignore a changed setting, clear this check box.

Except for the setting Agent resource cache folder, you can choose for each setting whether to use Agent specific settings or inherited settings. Inherited Agent settings are inherited from:

- the primary Team (if available). A primary Team is a Team whose settings can be inherited by each Agent that is a member of this Team and for which the Team has been set as its primary Team. You can set Teams as primary Teams on the Teams tab and at Topology > Teams. If an Agent is a member of a primary Team, this will be shown in the Primary Team field. See Teams (on page 205).
- the global settings, if the Agent is not a member of a primary Team.

Notes
- When you repair an Agent or Agent+, RES ONE Automation uses the current Agent settings by default.
- Agents that use a Team as their primary Team not only inherit its Trusts (Security settings), but also its Team settings (Topology settings). It is therefore recommended to consider the consequences before assigning an Agent to a different primary Team, as this may have a severe impact on the performance of the Agent.

Best practice: Create a Team folder, configure Trusts on this folder and add the two primary Teams to the Team folder. In this way, it is not necessary to change Team settings or Trusts for each primary Team: the Agent will still inherit the correct settings from its primary Team. See Trusts (see page 269).
6.3.2 Agent Variables

When configuring a Task, the values of many fields can optionally be replaced with Variables. Similar to parameters, Variables function as placeholders for customer-specific values, such as, for example, server names, passwords or credentials. Different from parameters however, Variables are resolved at the moment of Job execution, rather than at the moment of Job scheduling.

The global values of Variables can be configured at Library > Variables. These values are inherited by default by all Agents, but exceptions for individual Agents can be set up on the Agent's Agent Variables tab. See Variables (on page 176).

- It is not possible to edit categories from the Agent Variables tab.
- When editing an Agent Variable, only the value can be set.
- Like the settings on the Agent Settings tab, you can choose for each Agent Variable whether to use Agent-specific values or inherited values. Inherited Agent values are inherited from:
  - the primary Team (if available). A primary Team is a Team whose settings can be inherited by each Agent that is a member of this Team and for which the Team has been set as its primary Team. You can set Teams as primary Teams on the Teams tab and at Topology > Teams. Check the Teams tab to see whether an Agent is a member of a primary Team.
  - the global Variables (on page 176), if the Agent is not a member of a primary Team.

6.3.3 Agent Trusts

Use the Trusts tab to define trusted Modules and Resources for the Agent. This tab is divided in the tabs Trusted Modules and Trusted Resources and allows you to define trusted Modules and Resources for specific Agents. An Agent can have explicit Trusts or inherit its Trusts from its primary Team. See Trusts (see page 269).

- Use the search bar to search for specific objects.
- You can use the following settings when configuring Trusts:
  - Do not trust: If this setting is selected, the selected item is not trusted. For example, if you are configuring Trusts on a specific Agent and specify that the Module Delete RES ONE Automation Results is not trusted by the Agent, the Module will fail when used in a Job that is executed by the Agent.
  - Trust: If this setting is selected, the selected item is trusted. This is the default setting. Depending on whether a full Trust exists with the item, it is possible to use the item in a Job. For example, if you are configuring Trusts on a specific Module and specify that Agent ONE is trusted by the Module, the Agent is allowed to use the Module in a Job.
  - Inherit trust: If no particular settings are selected (blank check box), the item will inherit the Trusts from the above lying item. Trusts that are inherited from an above lying item are grayed out and show the type and name of this item, except when this is the top-level folder: The Trusts of each top-level folder use the default setting (Trust). These Trusts are shown in black and it is not possible to change their settings.
  - All items that are shown on the various Trusts tabs are subject to the access permissions of the administrative role(s) of the Console user.
• Select Show only items that trust this ... to filter on items that already trust the Agent, Team, Team folder, Module, Module folder, Resource or Resource folder that you are editing. When Trusts are enabled, Agents can only execute Jobs with Modules and Resources when a full Trust exists between them (i.e. the Module or Resource is trusted by the Agent AND the Agent is trusted by the Module or Resource). By using a filter, it becomes easier to create full Trusts.

• Filtering the Trusts tab may take some time, especially in large environments with many Agents, Modules and Resources. When selecting the option, a popup window is therefore shown, showing the progress. If necessary, you can cancel the filtering.

• If an item does not trust the item that you are editing, it will not be shown. Folders will be shown with an overlay icon.

• This option requires Microsoft .NET Framework 4.0 (Full version) or higher to be installed on the machine that runs the Console.

6.3.4 Open a remote Console on an Agent

The Remote Console functionality allows you to open a remote Console on an online Agent. This is useful, for example, if a user phones you about a problem and you want to start a remote session on the user’s computer to find the cause of this problem.

Prerequisites

• The Agent must be online.
• The Agent must be running Microsoft Windows Server 2003, Windows XP or later.
• Your administrative role must at least have Read permissions to the Agent.
• Microsoft Terminal Services Client must be installed on the computer that runs the Console.
• The Agent and the Console must be able to access the same Dispatcher.

Configuration

To open a remote Console, right-click the relevant Agent and click Remote Console. This sets up a temporary tunnel between the Console and the Agent via a Dispatcher. Both the Agent and the Console will initiate an outgoing IP connection to the Dispatcher, after which RES ONE Automation will start an RDP-session to set up the remote Console.

Notes

• Consider adding additional firewall rules to disallow incoming RDP-traffic on the Agent. In this situation, only RES ONE Automation will be allowed to use Remote Consoles. This allows you to rely on the Audit Trail and to enforce compliancy.
• The Audit Trail reflects the duration of the Remote Console session.
• RDP takes care of encrypting the data that is sent.
6.3.5 Install Agents embedded using Prepare for Image

Agents can also be installed embedded. This is useful if you use images to roll out new computers in your network.

Images are commonly used to install new computers in a network. By including an Agent in the image, you can deploy new Agents that are identical copies of the original Agent when a new computer is installed using the image. After installation, the Agent will start automatically and register itself in the Datastore with its new computer name. Optionally, you can specify Projects for the Agent in the image, which will then be executed by the new Agent when it connects to a Dispatcher. This functionality is also available for RES ONE Automation Agents for Unix/Linux.

When you create an image, RES recommends to keep it as "clean" as possible. It should only contain the Operating System, an Agent, and other necessary system components.

1. Install the Agent on the computer that you want to use to create an image. Prepare this Agent for imaging by using the Prepare for Image wizard. You can find this wizard in the lower-left corner of the Properties tab of the new Agent.
2. Click Next to continue the wizard.
3. Optionally, you can select the Projects that need to be invoked when the new Agent starts and connects to a Dispatcher. You can select multiple Projects.
4. Click Next to continue the wizard.
5. Click Finish to complete the wizard and to prepare the Agent for imaging.
6. The Agent will now be prepared for imaging. It will be removed from the Agent list in the Console. If you install a new computer by using the new image, the Agent will start automatically and it will register itself in the Datastore with its new computer name. After that, it will execute any pre-selected Project.

Warning
If you prepare an Agent for imaging, it will no longer be available in the Console.

6.3.6 Install Agents embedded using Prepared4Embedded

A drawback of the Prepare for Image method, as described in the chapter Installing Agents embedded using Prepare for Image (see page 196), is that the Prepare for Image wizard automatically preconfigures the Agent in the image to connect to a specific RES ONE Automation site (by using the GUID of the RES ONE Automation site.) Instead of relying on the identifier of a RES ONE Automation site, you can also use the RES ONE Automation site:

To do this, set the following registry entry:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>HKEY_LOCAL_computer\SOFTWARE\RES\AutomationManager\Agent (32-bit)</td>
<td>Name of RES ONE Automation environment</td>
</tr>
<tr>
<td>HKEY_LOCAL_computer\SOFTWARE\Wow6432Node\RES\AutomationManager\Agent (64-bit)</td>
<td>Name of RES ONE Automation environment</td>
</tr>
</tbody>
</table>

When you now install a new Agent by using the image, the new Agent will start automatically and starts looking for a RES ONE Automation environment that matches this name. When it finds the RES ONE Automation environment, the Agent will register itself in the Datastore and execute any pre-selected Project.
6.3.7 Snapshot Intelligence

Snapshot technology is available in various forms and from various vendors, such as VMware and Symantec Ghost (because traditional disk images can also be seen as snapshots).

A snapshot records the state of a computer at a moment in time. Any software changes that occur on that computer afterwards can be undone simply by restoring the snapshot. This brings the computer back to the recorded moment, before the undesirable changes were made. Then you might try out a different combination of changes to see whether that works better. For example, you can re-do all the changes except that one software patch that caused all those problems.

Snapshots therefore offer many advantages in various scenarios, such as backup and data recovery, change management and development and test cycles. In all these situations, it can be useful or even crucial to jump to a previous state and start with a clean slate. System administrators can take snapshots at regular intervals to create multiple points to which the computer can be returned.

Reverting to a snapshot, however, can also bring some significant disadvantages.

Firstly, change management systems are generally not aware of snapshots, or only at specific intervals. If a change is removed when a snapshot is reverted, that obsolete change will remain in a change management system for a while, depending on the system's update interval. In some cases, obsolete records remain in the system permanently. There is an obvious danger, for example if the records show that crucial updates are present on the computer, when in fact they are not.

Secondly, the snapshot moment may not be the actual moment to which the computer needs to return. This moment may be earlier, and considerably so if snapshots are only taken occasionally. In such cases, changes will have been removed which should have been kept. These changes will have to be implemented again, at considerable cost in time and effort.

RES ONE Automation overcomes these disadvantages with Snapshot Intelligence. With Snapshot Intelligence, you:

- will always have the Agent's history automatically matched to their actual state: When snapshots are used as a backup and recovery mechanism, reverting to a snapshot returns the Agent to the state it was in when the snapshot was taken. Changes that were made after the snapshot moment were lost. This made it necessary to perform additional Jobs to get the Agent back to the state it was in immediately prior to the problem. When a snapshot is reverted, this often results in inconsistencies between the actual computer state and any tracked change history in a database. Snapshot Intelligence allows RES ONE Automation to detect when a snapshot has been applied on an Agent, and automatically changes that Agent's change history so that it once again matches the Agent's actual state.

- can automatically reapply the last change path: You can configure RES ONE Automation to reapply the last change path automatically after a snapshot has been applied. This brings the computer back to its most recent state, rather than the state it had when you last created a snapshot. This effectively closes the time gap between the snapshot moment and the desired state to which the Agent needs to return.

- can manually select a change path to be reapplied: RES ONE Automation remembers all change paths that have been implemented on an Agent, even after they have become obsolete. When an Agent has reverted to a snapshot, you can view previous change paths and reapply a change path of your choice. You can also save a change path as a new Module. These options make it possible to create a selection of target states, all based on the same snapshot as a starting point.
Manually choose additional Jobs to execute after reverting to snapshot

When you use snapshots as a backup and recovery mechanism, reverting to a snapshot returns an Agent to the state it was in when the snapshot was taken. Any changes that you made after the snapshot moment are lost. This makes it necessary to perform additional Jobs, to get the Agent back to the state it was in immediately prior to the snapshot moment.

Snapshot Intelligence makes it possible to reapply a set of Jobs that have become obsolete. This change path can be selected manually (on the Agent’s Snapshot Intelligence tab), so that you can determine exactly which Jobs to reapply. This option also effectively allows you to switch between different Agent states, based on one snapshot starting point.

Automatically execute additional Jobs after reverting to snapshot

RES ONE Automation can be configured to automatically reapply the last change path, so that each Agent is brought back to its most recent state automatically. The Jobs effectively form an appendix to the snapshot. This is particularly useful in environments using disk virtualization (such as Provisioning Server), as it reduces the frequency with which the snapshots themselves need to be updated.

Configuration

- RES ONE Automation always automatically matches Job and Agent histories to the actual state of the Agents. No configuration is required.
- The Global Setting When use snapshot is detected determines what RES ONE Automation will do when an Agent reverts to a snapshot. You can make exceptions to this setting for specific Teams and Agents.
  - Enabled Snapshot processing: To manage snapshots and be able to revert the Agent to the same state as it had before the snapshot was taken, RES ONE Automation logs the change path of the Agent on its Snapshot Intelligence tab when the use of a snapshot is detected. The Job history of the Agent is cleaned up accordingly. However, in environments in which it is not necessary to manage snapshots, for example in environments using disk virtualization (such as Citrix Provisioning Server), in which a snapshot of an Agent is created each time a user logs on, this behavior may not always be desirable. As of RES ONE Automation version 10.2, Snapshot Intelligence is disabled by default when creating a new Datastore.
  - When use of snapshot is detected: do nothing: RES ONE Automation will simply clean up the histories, but will undertake no further action. This is the default setting. Before executing any new Jobs on the Agent, you can manually select a change path to reapply (on the Agent’s Snapshot Intelligence tab). The selected change path can be any change path from the Agent’s past, not necessarily the last one.
  - When use of snapshot is detected: Reapply last change path: RES ONE Automation will always reapply the last change path on an Agents on which a snapshot is detected. This brings the computer back to its most recent state, rather than the state it had when you last created a snapshot. This effectively closes the time gap between the snapshot moment and the desired state to which the Agent needs to return. This option is essential when applying changes to a standard image combined with disk virtualization. Normally, these changes are lost after rebooting the computer (that is, the standard image will be reapplied by the use of disk virtualization).
  - Skip shutdown and reboot Tasks: these types of Tasks are excluded from any change path. This is necessary for Agents that boot from snapshot (read-only virtual disk), because otherwise they would end up in a loop. This option should be cleared on regular Agents on which software is installed, for example, because that will require a reboot.
  - All change paths that can be chosen from the current point in time are shown on the Change Tree tab on the Agent’s Snapshot Intelligence tab. Change paths that do not follow on from the snapshot moment are not available for selection.
  - Manually selected change paths can be edited before being reapplied, so that specific Tasks or types of Tasks can be excluded.
Chapter 6: Topology

- On the Change Tree tab, click **Toggle Change Path** to select the beginning and end of the change path to be reapplied.
  - To choose the starting point for the change path, click on the first Job that you want to reapply.
  - Click **Toggle Change Path** to see the range of all possible change paths from that starting point (the area is marked light blue).
  - Select the last Job that you want to reapply.
  - Click **Toggle Change Path** once again to mark the selected change path bright green.
  - To change the end point of the change path, select the relevant Job and then click **Toggle Change Path** again.
  - To clear the selection and start again, click **Clear Change Path**.
  - Click **Last Change Path** to automatically select the most recent change path.
  - Reapply the selected change path by clicking **Reapply Change Path**.

- In the Job results for a specific Job, clear **Show Agents that are marked obsolete for this Job by Snapshot Intelligence** to hide Agents for which this Job has become obsolete because the Agent has reverted to a snapshot. This option is also available for query result overviews per Task.

**Notes**

- Change paths consist of RES ONE Automation Jobs. Changes implemented outside of RES ONE Automation cannot be detected, nor reapplied.
- There are three locations where you can see when snapshots were detected: on the Agent’s Audit Trail tab, at Administration > Audit Trail (in the Description column), and on the Agent’s Snapshot Intelligence tab under Used Snapshots.
- The Snapshot Intelligence tab appears in an Agent’s Properties window when that Agent first reverts to a snapshot. After that, the tab remains visible.
- Because of database changes, the Job History will become unavailable if you downgrade from RES ONE Automation 2015 to a lower version. As a result, Snapshot Intelligence change paths that were made in RES ONE Automation 2015 will become useless.

**Tip**

- When a change path has been selected to be reapplied, and the Tasks within that path have been selected or cleared, the selected Tasks can be saved as a Module for use at any time and on any Agent.
- On the Change Tree tab, expand or collapse the entire change tree by right-clicking the column header Description and choosing **Collapse all** or **Expand all**.

**Deleting the Job History of an Agent**

If many snapshots of an Agent are made, for example by using a provisioning server, the Job history of such Agents can become very large and may contain many obsolete entries. In such situations, it can be desirable to clean up the Job history of the Agent.

**Configuration**

To delete the Job history of an Agent, click **Delete Job History** on the History tab of the Agent Properties window, which is available when viewing the properties of an Agent. You can also delete the Job history of an Agent at Topology > Agents (right-click the Agent, select **Delete Job History**).

**Warning**

This functionality will delete the entire Job history of the Agent and is irreversible.
Manually removing Agents from the Datastore and Console

If you want to remove obsolete Agents that are no longer in use, you need to remove these Agents from the Datastore manually.

Configuration
1. At Topology > Agents, select the relevant Agent and click Remove in the command bar. The Remove Components window opens.
2. Click Remove now.
3. When prompted for the credentials to connect to the target computer, enter an incorrect password so that the connection fails. RES ONE Automation will attempt to remove the component, but because it cannot connect to the computer, the attempt will fail.
4. When prompted whether to remove the selected agent from the Datastore only, click Yes. The offline Agent will be removed from the Datastore and the Console.
6.3.8 Agent+

The Agent+ uses the latest Microsoft tools. This has enhanced the legacy Agents with the following capabilities:

- Improved scalability
- Native 64-bit and 32-bit installer to maximize the hardware and operating systems possibilities
- Use of push communication to proactively notify changes to the Agent+
- The Agent+ logs events in the Event log, which can be viewed with e.g. the Event Viewer that is part of the Microsoft Windows Operating System (see Administration Guide for complete list of Event logs (on page 309))

With the Agent+, you can perform the same desktop automation Tasks on machines running the legacy Agent. There are some differences between the Agent and the Agent+ to keep in mind:

Prerequisites

The Agent+ requires the installation of Microsoft .NET Framework 4.5.2 (Full version) and Microsoft Windows 7 or Microsoft Windows Server 2008 R2 and higher on the machine. The following operating systems are not supported for the Agent+: Microsoft Windows XP, Microsoft Windows Vista, Microsoft Windows 2003 Server and Microsoft Windows 2008 Server.

Installation

You can deploy the Agent+ directly from the Console. RES ONE Automation will automatically use the correct MSI file (x64 or x86), depending on the operating system of the target computer. It is also possible to install the Agent+ manually by saving the component as an MSI file. It is not possible to install the legacy Agent, if the Agent+ is already installed on the same machine.

Communication between Agent+ and Dispatcher

Besides the communication between Agent and Dispatcher (see page 6) (using port number 3163), the Agent+ also uses an additional communication method called push notifications. Via push notification messages, the Agent+ is informed when there are any changes for it to process, such as changes to the Agent+ properties, or Jobs to be performed. By default, the Agent+ uses port number 3162 to connect to a Dispatcher for receiving push notification messages. With the global setting Dispatcher push communication, you can specify a different port for this push communication between the Dispatchers (server) and Agents+ (clients).

IPv6 support

Besides IPv4, also IPv6 as Internet Protocol is supported for the communication to and from the Agent+. On the legacy Agents, only IPv4 addresses are supported.
Chapter 6: Topology

Tasks “Exchange Mailbox”

The Agent+ supports Microsoft Exchange Servers 2010, 2013 and 2016. If you run the Tasks Create, Manage, Move, Export or Disable Exchange Mailbox on Microsoft Exchange Servers 2003 or 2007 using the Agent+, the Tasks will fail. On the legacy Agents, these Tasks will still run.

Notes

- On the Agent+, List or Multi-select list parameters linked to Credentials parameters are not supported.
- On the Agent+, leading and/or trailing spaces in parameter values are not removed at Job scheduling. On the legacy non-Windows Agents, the same applies. On the legacy Windows Agents, these leading and/or trailing spaces will be removed.
- On the Agent+, the abort function does not behave as expected for the following Tasks:
  - Message Box (Show)
  - Secure Shell (SSH) Commands (Execute)
  - Windows PowerShell Script (Execute)
  - RES ONE Automation Results (Delete, Export)
  - Microsoft Exchange Mailbox (Create, Manage, Move, Export, Disable, Query)
  - Citrix Workflow (Invoke)
  - Citrix Published Applications (Query)
  - Parameters (Query)
  - RES ONE Automation Dispatchers (Discover)
  - RES ONE Automation Team Membership (Change)
  - E-mail (Send)

6.3.9 RES ONE Automation Agents for Unix/Linux

RES ONE Automation is not only capable of automating IT tasks on machines running Microsoft Windows, but can also do this on machines running Unix and Linux. For this purpose, the RES ONE Automation Agent for Unix/Linux is available.

With the RES ONE Automation Agents for Unix/Linux you can perform the following Tasks:

- Unix/Linux Command (Execute) (see page 144)
- Unix/Linux Computer (Reboot, Shutdown, Uptime) (see page 145)
- Unix/Linux Resource (Download) (see page 146)
- User Management (Create, Delete, Lock, Unlock, Change Password, Query) (see page 147)

These Tasks are grouped in the Task library Unix/Linux in the Console. Please refer to the Getting Started with RES ONE Automation Agents for Unix and Linux for more information on installation and upgrading.
6.3.10 Agents for Apple Mac OS X

RES ONE Automation is not only capable of automating IT tasks on machines running Microsoft Windows, but can also do this on machines running Apple Mac OS X. For this purpose, the RES ONE Automation Agent for Apple Mac OS X is available.

With the RES ONE Automation Agents for Apple Mac OS X you can perform the following Tasks:

- Mac OS X Automator (Invoke) (on page 122)
- Mac OS X Command (Execute) (on page 122)
- Mac OS X Computer (Reboot, Shutdown, Inventory, Uptime) (on page 124)
- Mac OS X Disk Space (Query) (on page 125)
- Mac OS X Installed Programs (Query) (on page 125)
- Mac OS X Resource (Download) (on page 124)
- Mac OS X Software Package (Install, Remove) (on page 126)

These Tasks are grouped in the Task library Mac OS X in the Console. Please refer to the Getting Started with RES ONE Automation Agents for Mac OS X for more information on installation and upgrading.

6.3.11 Bare Metal OS deployment using WDS and RES ONE Automation

When deploying new workstations or servers, Bare Metal OS deployment usually is the first task that is performed. This task can easily be carried out by combining Microsoft Windows Deployment Services (WDS) and RES ONE Automation.

For more information about Bare Metal OS deployment using WDS and RES ONE Automation, see Bare Metal OS Deployment Using WDS and RES ONE Automation (on page 274).
6.4 Teams

If certain Jobs are often scheduled on the same group of Agents, it can be useful to group these Agents in a Team. Teams can be based on any criterion. For example, you may choose to group Agents in Teams based on location, machine type, Operating System and so on. This makes it much easier to execute Jobs uniformly on the correct Agents.

For example, if you create a Team for all Agents that run Microsoft Windows Vista, it becomes much easier to execute Vista maintenance Tasks on all of these Agents: you just schedule a Job with the Team, instead of on all individual Agents.

At Topology > Teams:

• Use the Folders tab to group Teams into folders. Grouping Teams in folders can be useful in large environments and multi-tenant RES ONE Automation sites. You can use Team folders for example to:
  • set up permissions on groups of Teams. See Administrative Roles ("Security" on page 213).
  • set up trusted Modules and Resources for a Team folder. See Trusts (see page 269) and Team Folders (on page 205).

• Use the Teams tab to view general information about Teams:
  • The Members column shows the number of unique Agents that are member of each Team. The member count of a Team is based on the number of unique Agents: because Teams can be nested in other Teams, RES ONE Automation only counts each individual Agent once, irrespective of whether it is also a member of a nested Team. The Members column only shows the number of members for enabled Teams.
  • The Primary Teams column shows which Teams are set as a primary Team for one or more Agents (values are Yes/No).

Notes

• If Agents in a Team are offline because the computers on which they are installed are turned off, you can use wake-on-LAN to turn on the computers and bring the Agent online. You can do this by selecting the Team, right-click it and choose Wake-on-LAN. Please note that this feature only works if the Agent and your network infrastructure support wake-on-LAN.
• If you disable a Team, the Team can no longer be selected in a Job. However, the Agents can be selected.

6.4.1 Team Folders

By grouping Teams in folders (At Topology > Teams) you can:

• configure permissions on groups of Teams. See Administrative Roles ("Security" on page 213).
• configure trusted Modules and Resources for a Team folder. See Trusts (see page 269).

Configuration

• When adding or editing a Team folder, use the Trusts tab to define trusted Modules and Resources.
  • Use the Trusted Modules tab to define trusted Modules.
  • Use the Trusted Resources tab to define trusted Resources.
  • Use the search bar to search for specific objects.
• You can use the following settings when configuring Trusts:
  • **Do not trust**: If this setting is selected, the selected item is not trusted. For example, if you are configuring Trusts on a specific Agent and specify that the Module *Delete RES ONE Automation Results* is not trusted by the Agent, the Module will fail when used in a Job that is executed by the Agent.
  • **Trust**: If this setting is selected, the selected item is trusted. This is the default setting. Depending on whether a full Trust exists with the item, it is possible to use the item in a Job. For example, if you are configuring Trusts on a specific Module and specify that Agent *ONE* is trusted by the Module, the Agent is allowed to use the Module in a Job.
  • **Inherit trust**: If no particular settings are selected (blank check box), the item will inherit the Trusts from the above lying item. Trusts that are inherited from an above lying item are grayed out and show the type and name of this item, except when this is the top-level folder: The Trusts of each top-level folder use the default setting (*Trust*). These Trusts are shown in black and it is not possible to change their settings.

• All items that are shown on the various **Trusts** tabs are subject to the access permissions of the administrative role(s) of the Console user.

• Select **Show only items that trust this** ... to filter on items that already trust the Agent, Team, Team folder, Module, Module folder, Resource or Resource folder that you are editing. When Trusts are enabled, Agents can only execute Jobs with Modules and Resources when a **full Trust** exists between them (i.e. the Module or Resource is trusted by the Agent AND the Agent is trusted by the Module or Resource). By using a filter, it becomes easier to create full Trusts.

• Filtering the **Trusts** tab may take some time, especially in large environments with many Agents, Modules and Resources. When selecting the option, a popup window is therefore shown, showing the progress. If necessary, you can cancel the filtering.

• If an item does not trust the item that you are editing, it will not be shown. Folders will be shown with an overlay icon.

• This option requires Microsoft .NET Framework 4.0 (Full version) or higher to be installed on the machine that runs the Console.

---

**Note**

Availability of the **Trusts** tab depends on the global setting **Trusts Security**. See **Trusts** (see page 269).
6.4.2 Team settings

New Teams that are created always take on the Global Settings (Setup > Global Settings). If necessary, you can edit these settings for individual Teams. By configuring Team settings, you can configure settings for a Team that deviate from the global settings. This makes it much easier to change settings on groups of Agents. The settings of a Team only apply to an Agent if the Team is set as primary Team for that Agent.

Primary Teams are Teams whose settings can be inherited by each Agent that is a member of the Team. An Agent can only inherit settings from its primary Team: if an Agent is also a member of other Teams, these settings will be ignored.

For each Agent, you can choose whether to use inherited settings or Agent specific settings. Inherited Agent settings are inherited from:

- the global settings, if the Agent is not a member of a primary Team.
- the primary Team, if the Agent is a member of this Team.

Configuration

You can change the settings of Teams in different ways:

- Open the properties of an individual Team and click the Team Settings tab.
- Right-click a Team and click Settings Overview. This opens the Settings Overview window, which shows an overview of the settings of all Teams. The settings overview does not resolve nested Teams to show the settings of the underlying Agents.
  - To change these settings, select one or more Teams and click Change settings of selected Teams. It is not possible to change the settings from nested Teams.
  - Select the relevant Teams, right-click this selection and click Change settings of selected Teams. This opens the Change settings for multiple Teams window.
    - To change a setting, select it and click Edit.
    - After changing the setting, its check box will be selected. To ignore a changed setting, clear this check box.

To set a Team as a primary Team for an Agent, do one of the following:

- open the Agent Properties of a specific Agent, select a Team on the Teams tab and click Set as primary Team.
- open the Team Properties of a specific Team, select one or more Agents on the Members tab and click Set current Team as primary. You can automatically set Teams as primary Teams for new members.

6.4.3 Team Variables

When configuring a Task, the values of many fields can optionally be replaced with Variables. Similar to parameters, Variables function as placeholders for customer-specific values, such as, for example, server names, passwords or credentials. Different from parameters however, Variables are resolved at the moment of Job execution, rather than at the moment of Job scheduling.

The global values of Variables can be set up at Library > Variables. These values are inherited by default by all Teams, but exceptions for individual Teams on the Team’s Team Variables tab. See Variables (on page 176).

- It is not possible to edit categories from the Team Variables tab.
- When editing a Team Variable, only the value can be set.
- When editing a Team Variable, select Inherit from global to apply the value of the Global Variable to the Team. Clear the option to specify Team-specific values that override the global values. Nested Teams do not inherit their settings from the Team in which they are nested.
6.4.4 Team members

A Team is formed by its Team members. These are Agents, but can also be other Teams. These so-called "nested Teams" are useful if you regularly execute Jobs on the same group of Teams. For example, the Team Microsoft Windows XP laptops and the Team Microsoft Windows Vista laptops can both be member of the Team Laptops.

You can add members to a Team on its Members tab. Membership of a Team can be explicit (that is, Agents or Teams that have been added manually) or rule-based. Explicit members take precedence over rule-based members.

Configuration

- In the overview of all Teams, the Members column shows the number of unique Agents that are member of each Team. The member count of a Team is based on the number of unique Agents: because Teams can be nested in other Teams, RES ONE Automation only counts each individual Agent once, irrespective of whether it is also a member of a nested Team.
- Suppose Team 1 is embedded in Team 2, which is embedded in Team 3, which is embedded in Team 1 again. The order in which the Job will be executed will be ascending according to Team names.
- When viewing the properties of a Team, in the New/Edit Team window, the Members column of the Members tab shows the number of unique Agents that are member of each nested Team (if any); the Member count field shows the total number of unique Agents that are member of the Team. These numbers are not updated during any Team member changes, but first (re)calculated when the changes to the Team are saved.

Notes

- The Members column only shows the number of members for enabled Teams.
- Agents that use a Team as their primary Team not only inherit its Trusts (Security settings), but also its Team settings (Topology settings). It is therefore recommended to consider the consequences before assigning an Agent to a different primary Team, as this may have a severe impact on the performance of the Agent.

  Best practice: Create a Team folder, configure Trusts on this folder and add the two primary Teams to the Team folder. In this way, it is not necessary to change Team settings or Trusts for each primary Team: the Agent will still inherit the correct settings from its primary Team. See Trusts (see page 269).
- On the Properties tab, select Do not allow add to team outside console to prevent Agents from being added to a Team from outside the Console, for example through an MSI property or a registry value.
Team rules

By setting up Team rules, you can add Agents as members of a Team automatically. Team rules are based on the specification of the computer on which the Agent runs. They are evaluated by the Dispatcher each time the Agent comes online or when network IP address changes are made. If an Agent meets the rules of a Team, it will automatically be added to the Team. If it no longer meets the rules of a Team, it will be removed from the Team.

The Rules button shows the number of configured Team rules in brackets.

You can base Team rules on:

- Computer Function
- First Accessed Dispatcher
- Fully Qualified Domain Name
- IP Address (IPv4 (incl. subnets) or IPv6 (Agent+ only))
- MAC Address
- Operating System (Bit Version, Suite, Type, Version)
- Processor Architecture

Team rules based on IP address and MAC address

- You can enter multiple IP addresses or MAC addresses. Separate multiple entries with a semicolon (;).
- Each IP address must be validated against all Team rules based on IP address.
  - For AND rules: one of the Agent’s IP addresses must apply to ALL IP address rules.
  - For OR rules: one of the Agent's IP addresses must apply to at least 1 IP address rule.
- MAC addresses must be specified in this hexadecimal format: XX:XX:XX:XX:XX:XX. For example: 00:0C:29:23:51:A3
- You can use pattern matching. For example, suppose you want to configure a Team for any Agent whose IP address starts with a specific set of numbers. Instead of specifying the IP addresses of each individual Agent, you can use a pattern. This not only applies the rule to all existing Agents whose IP address match the specified pattern, but also to new Agents:

<table>
<thead>
<tr>
<th>IP address</th>
<th>Pattern</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting with 10.1.7</td>
<td>10.1.7.*</td>
<td>Applies to any Agent whose IP address starts with 10.1.7</td>
</tr>
<tr>
<td>Starting with 10.1</td>
<td>10.1.?.*</td>
<td>Applies to any Agent whose IP address starts with 10.1, followed by a single digit and then any number</td>
</tr>
</tbody>
</table>

- you can verify the IP addresses of the Agents that have been added to the Team by scheduling a Task Query TCP/IP Properties.
Automatic Job scheduling

With **Automatic Job scheduling** you can schedule a Project or a single Module for automatic execution when new Agents are added to a Team or when Agents are removed from a Team. This is supported for explicit Team members, as well as rule based Team members. Jobs for new Team members are typically used to add new functionality to a computer. Jobs for removed Team members are typically used to remove old functionality from a computer.

For example, you can configure a Job **Install Dispatcher** to run automatically on Agents that are made member of the Team **Dispatchers**, and a Job **Remove Dispatcher** on Agents that are removed from that Team. Any automatic Job Scheduling applies to both directly assigned memberships and rule-based memberships, and applies to members of nested Teams as well as direct members.
6.4.5 Team Trusts

Use the Trusts tab to define trusted Modules and Resources for the Team. This tab is divided in the tabs Trusted Modules and Trusted Resources and allows you to define trusted Modules and Resources for specific Agents. Teams. A Team can have explicit Trusts or inherit its Trusts from its Team folder. See Trusts (see page 269).

- Use the search bar to search for specific objects.
- You can use the following settings when configuring Trusts:
  - **Do not trust**: If this setting is selected, the selected item is not trusted. For example, if you are configuring Trusts on a specific Agent and specify that the Module Delete RES ONE Automation Results is not trusted by the Agent, the Module will fail when used in a Job that is executed by the Agent.
  - **Trust**: If this setting is selected, the selected item is trusted. This is the default setting. Depending on whether a full Trust exists with the item, it is possible to use the item in a Job. For example, if you are configuring Trusts on a specific Module and specify that Agent ONE is trusted by the Module, the Agent is allowed to use the Module in a Job.
  - **Inherit trust**: If no particular settings are selected (blank check box), the item will inherit the Trusts from the above lying item. Trusts that are inherited from an above lying item are grayed out and show the type and name of this item, except when this is the top-level folder: The Trusts of each top-level folder use the default setting (Trust). These Trusts are shown in black and it is not possible to change their settings.
- All items that are shown on the various Trusts tabs are subject to the access permissions of the administrative role(s) of the Console user.
- Select **Show only items that trust this** … to filter on items that already trust the Agent, Team, Team folder, Module, Module folder, Resource or Resource folder that you are editing. When Trusts are enabled, Agents can only execute Jobs with Modules and Resources when a full Trust exists between them (i.e. the Module or Resource is trusted by the Agent AND the Agent is trusted by the Module or Resource). By using a filter, it becomes easier to create full Trusts.
  - Filtering the Trusts tab may take some time, especially in large environments with many Agents, Modules and Resources. When selecting the option, a popup window is therefore shown, showing the progress. If necessary, you can cancel the filtering.
  - If an item does not trust the item that you are editing, it will not be shown. Folders will be shown with an overlay icon.
  - This option requires Microsoft .NET Framework 4.0 (Full version) or higher to be installed on the machine that runs the Console.
  - When you create a duplicate of a Team for which Trusts have been defined, you will be prompted whether the Trusts should be included in the duplicate.

**Notes**

- Availability of the Trusts tab depends on the global setting Trusts Security. See Trusts (see page 269).
- Agents that use a Team as their primary Team not only inherit its Trusts (Security settings), but also its Team settings (Topology settings). It is therefore recommended to consider the consequences before assigning an Agent to a different primary Team, as this may have a severe impact on the performance of the Agent.

Best practice: Create a Team folder, configure Trusts on this folder and add the two primary Teams to the Team folder. In this way, it is not necessary to change Team settings or Trusts for each primary Team: the Agent will still inherit the correct settings from its primary Team. See Trusts (see page 269).

6.4.6 Team Permissions

Use the Permissions tab to view the level of access that Console users have to the Team.
Chapter 7: Administration

The Administration node contains all settings of your RES ONE Automation environment. It contains the following sections:

- **Information**: View general information about the Datastore.
- **Security**: Configure different levels of access to the Console.
- **Components**: View and update RES ONE Automation components.
- **Audit Trail**: View the Audit Trail.

### 7.1 Information

The Information node shows information about the Datastore.

- The **Settings** tab shows the connection settings to the current Datastore.
  - To change the connection settings, click **Edit**. The Console will jump to the node **Setup > Database**.
- The **Properties** tab shows general information about the Datastore.
- The **Size** tab shows information about the size of the Datastore. This tab is not available when using a Datastore located on an IBM DB2 database server.
7.2 Security

Setting up different levels of access to the Console can be useful to prevent unauthorized users from making (accidental) changes to your RES ONE Automation environment.

For example, by setting up different levels of access, you can allow help desk employees to schedule predefined (and tested) Run Books, but disallow them to make changes to any of the settings of these Run Books. In this way, help desk employees can perform complex tasks without you running the risk that they make changes to these tasks.

When login accounts and administrative roles have been set up, access of a user to the Console is determined by two things:

- **Login accounts** (see page 213): A login account allows RES ONE Automation to determine the identity of a user. The Console is password-protected as soon as a login account exists.
- **Administrative roles** (on page 216): An administrative role allows RES ONE Automation to determine the level of access that a legitimate user has to the various parts of the Console. A login account must be assigned to at least one administrative role in order to access the Console: login accounts that are not assigned to an administrative role are denied access to the Console.

7.2.1 Login accounts

Login accounts are used to authenticate users as legitimate Console user. Login accounts can be configured at Administration > Security, on the Logins tab.

Because you will usually assign permissions to users, instead of the other way around, the first step you need to take when setting up different levels of access to the Console is to create login accounts. A login account identifies a legitimate user of the Console.

- The **Global Password Security Policy** area reflects the password security policy as configured in the global settings (see page 48). This makes it possible to get an immediate overview which global settings are enabled, and if so, what their value is. These settings only apply to login accounts using RES ONE Automation Authentication.
- The columns give an overview of the settings of all individual logins accounts, including settings related to password security. The password security columns only apply to login accounts that use RES ONE Automation Authentication.
- To apply an account action to multiple login accounts, select the login accounts and right-click to display the context menu. In the context menu you can delete, enable or disable the selected login accounts at once. In addition, you can change the password security policy for login accounts using RES ONE Automation Authentication. If your selection includes Microsoft Windows Authentication logins and you want to change the password security policy, the login accounts using Microsoft Windows Authentication will be skipped.
When configuring a login account:

- Use the **Properties** tab to specify general properties of the login account, such as authentication settings and account actions.
  - The maximum length of the name of the Login account is 255 characters.
  - Select **RES ONE Automation Authentication** at **Account type** to configure custom authentication credentials. The user will be prompted for these credentials when he starts the Console. **RES ONE Automation Authentication** is an efficient way to provide access to a group of people with the same level of access: simply create a single login account, assign a relevant administrative role to it and provide all legitimate Console users with the relevant login information.
  - Use the **Account action** area to specify the password settings of the login account. The configured values will be shown in the corresponding columns on the **Logins** tab.
    - **Do not apply global Password Security Policy**: Select this option to not apply the configured global Password Security Policy (see page 48) for this login account.
    - **Account is locked out**: Use this option to unlock a locked out login account. When a user enters the wrong password and the number of attempts has exceeded, the account is locked out and this option is enabled and checked. After clearing (and saving) the option, the password is unlocked. It is not possible to lock accounts using this option: disable the account instead.
    - **User must change password at next login**: Use this option to prompt the user to change the password at next login. The **Change my password** (on page 215) window appears.
      - When the password is changed by the administrator, the password change date will also be reset. Except if the password is changed to the previous password.
  - Select **Microsoft Windows authentication** at **Account type** to use existing Microsoft Windows user accounts and local and global groups to authenticate access to the Console. This allows users to access the Console with their Microsoft Windows account and does not require additional authentication. If users also need access to the Console when not logged on with their Microsoft Windows account, use **RES ONE Automation Authentication** instead.
    - If a Microsoft Windows user account cannot be authenticated, the user will be prompted for the credentials of a login account using **RES ONE Automation Authentication**. It is not possible to provide Microsoft Windows credentials at that point.
    - By selecting the **Check type** of an account that applies to the login account, you can handle authentication in scenarios in which machines have the same name as a Windows domain, but are not a member of any domain:
      - **Domain**: Select this option to grant access to users who are logged on to a machine with a domain account.
      - **Local**: Select this option to grant access to users who are logged on to a machine that is not a member of a Windows domain.
      - **Both**: Select this option to grant access to users irrespective of the type of account used. This is the default option.
  - You can use login accounts using **RES ONE Automation Authentication** and **Microsoft Windows Authentication** alongside each other in the same RES ONE Automation environment.
    - The authentication method of existing login accounts cannot be changed from one method to the other.
• Use the Administrative Roles tab to assign administrative roles (on page 216) to the login account.
  • The Console is password-protected if at least one login account exists. The level of access that authorized users have to the Console once they have opened it, is determined by each user's administrative role.
  • To avoid accidental lockout (see page 304) of the Console, the first login account that you create is automatically assigned to the Full Access administrative role.
  • You can assign a login account to multiple administrative roles (see page 218).
  • Login accounts that are not assigned to an administrative role do not have access to the Console.

• Use the Permissions tab to view the permissions of the login account.

Warning
Deleting the last login account with administrative role Full Access restores Full Access for all users of the Console.

Notes
• Active Directory nested groups are not supported.
• When integrating with <RES_P or RES ONE Identity Director, only login accounts using RES ONE Automation Authentication are supported.

7.2.2 Change my password

At File > Change my password, Console users can change their password manually. The Change my password window will open automatically at login in the following scenarios:

• When a password has actually expired. When the password will expire within 5 days, the user gets a message to change the password manually.
• When an administrator has selected the option User must change password at next login for a specific account.

When the global Password Security Policy (see page 48) applies for the login account (see page 213), the password will be checked according to the configured security rules:

<table>
<thead>
<tr>
<th>Password Security Policy</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Password complexity</strong></td>
<td>When the password is changed, it must contain at least 3 of the following character types: uppercase, lowercase, numbers and symbols.</td>
</tr>
<tr>
<td></td>
<td>• You can use any of the following symbols: ~ ! @ # $ % ^ &amp; * _ + = \ ( ) { } [ ] ; : ; &quot; &lt; &gt; , . /</td>
</tr>
<tr>
<td><strong>Password history</strong></td>
<td>When the password is changed, it is not allowed to reuse a password until a specific number of password changes have been made.</td>
</tr>
<tr>
<td><strong>Minimum password length</strong></td>
<td>When the password is changed, a minimum number of characters needs to be used.</td>
</tr>
</tbody>
</table>

Also, the new password must match the confirmed password.

Note
The password complexity and minimum password length will be checked real time as you type. The password history will be checked at clicking OK.
7.2.3 Administrative Roles

When you have created one or more login accounts, you can assign them specific administrative roles.

- The access permissions (on page 216) of an administrative role determine the level of access that a Console user has to the various parts of the Console.
- The trust permissions (on page 217) of an administrative role determine the level at which a Console user is allowed to manage Trusts.

When you installed RES ONE Automation, a default administrative role was created: Full Access. This administrative role grants Modify access to all Console items and Modify permissions to Trusts on Team folders, Teams, Agents, Resource folders, Resources, Module folders and Modules. It is not possible to remove, disable, rename, or assign other permissions to this default administrative role. To avoid accidental lockout of the Console, the first login that you create is automatically assigned to this Full Access administrative role.

Access permissions

The access permissions of an administrative role determine the level of access that a Console user has to the various parts of the Console.

When you configured administrative roles, you can select the permissions for an administrative role at the Permissions tab. For each administrative role, access permissions on the various parts of the Console can be set to Deny, Read, Modify or Inherit:

- If access is set to Deny, the node, (sub)folder or object will be hidden and access to the windows and options that are associated with it will be blocked or hidden. By default, the access permissions of a new administrative role are set to Deny.
- If access is set to Read, the node, (sub)folder or object and the windows and options that are associated with it will be shown as read-only. Console users with Read access may still be able to set trusts on the item, depending on the trust permissions of their administrative role.
- If access is set to Modify, the login has full access to the node, (sub)folder or object and all windows and options that are associated with it.

In certain nodes, you can set explicit access permissions on (sub)folders and objects that deviate from the access permissions of the “parent” node. For example, this can be useful to set up administrative roles with specific access permissions on specific Agents and Teams. In large RES ONE Automation environments, this makes it easier to configure permissions to Resources, Modules, Projects or Run Books that are grouped in folders, for example to differentiate between different environments. This makes it possible to configure administrative roles that grant access to certain (sub)folders only.

- If no explicit access is set on a (sub)folder, it will inherit the access permissions of the node in which it is located.
- If no explicit access is set on an object in a (sub)folder, it will inherit the access permissions of the folder in which it is located.

Notes

- In the Jobs nodes, the permissions on a Job are determined by the permissions on the Agent(s) or Team(s) on which the Job is scheduled and by the permissions on the Module(s), Project or Run Book in the Job.
- You can only create Instant Reports and Building Blocks of items that relate to nodes and sections in the Console to which your administrative role(s) grant at least Read access. For example, a user who is denied access to the Resources node cannot create Building Blocks that include Resources.
- You can only import those Building Block items into the related nodes to which your administrative role(s) grant Modify access, including all (sub)folders. For example, if a Building Block contains Resources and Modules, but you only have Modify access to the Resources node, it will only be possible to import Resources: the Modules in the Building Block will be disabled in the Import Building Block window.
- If you only have Read access to the Console, it will not be possible to import Building Blocks and the import Building Block functionality will be disabled in the Console.
Special permissions

When you set up administrative roles, certain sections in the Console have additional functionality on which permissions can be set.

Agents

- **Allow Remote Console**: select this option to allow the Remote Console functionality in RES ONE Automation for the administrative role. This option is disabled by default for new administrative roles. You can also set the availability of the Remote Console as a Global Settings (see page 35).

- **Inherit permissions from primary team if no explicit access has been set**: select this option to assign the permissions of a primary Team to Agents. In large RES ONE Automation environments with many Agents at multiple environments, this makes it much easier to configure permissions for an administrative role to a specific set of Agents at a specific environment: by selecting the option and assigning explicit permissions to a primary Team, all Agents that belong to this Team will inherit its permissions. For example, this makes it possible to configure administrative roles that grant Read or Modify permissions to the primary Team, yet Deny permissions to the Agents node. When a user logs on with such an administrative role, only Agents that belong to the Team to which the user has permissions will become available; other Agents will not be available.

Modules

- **Limit task details when read permissions are set**: select this option to hide detailed information of Tasks when a user's administrative role only has Read permissions to the Modules node. This prevents unqualified personnel from obtaining crucial information about scripts and other Tasks (does not apply to Query Tasks). If you select the option:
  - only regular tabs of a Task (Properties tab, Module Parameters tab, Conditions tab) will be shown when viewing the Task in the Modules node or when scheduling a Job.
  - it is not possible to create Building Blocks of Modules.
  - information in an Instant Report about Tasks will be limited.

Job Scheduling

- **Override launch window**: select this option to make the option Override window and continue Job available when scheduling a Job. The option Override window and continue Job is part of the Job Scheduling setting If outside launch window and allows Console users with these permissions to override any launch windows during Job Scheduling. You can set up a launch window at Global Settings (see page 35) and set exceptions for specific Teams and Agents.

Trust permissions

Before Console users can start working with Trusts, you need to specify which Console users are allowed to do this, and on which items they are allowed to configure Trusts. Because "ownership" of customer-specific Agents, Modules and Resources often lies with customer representatives in multi-tenant sites, you may want to delegate control over Trusts to these representatives first, before you enable Trusts in your environment. This prevents situations in which unauthorized personnel can configure Trusts. You can delegate control over Trusts by assigning permissions to Trusts to specific administrative roles.

The permissions on Trusts in an administrative role define the level at which a Console user is allowed to manage Trusts. This allows for a granular delegation of control of Trusts and allows you to set up administrative roles for owners of specific Teams, Agents, Modules and Resources, with specific permissions on these items. For example, you can configure an administrative role for the owner of a specific Team that only allows him to configure Trusts for this Team. Other administrative roles may only see whether a Trust has been configured for the Team.

Permissions to Trusts can be assigned in the various Trusts columns on the Permissions tab of an administrative role, available in the Console at Administration > Security. Please note that the Trusts columns are only available when the global setting Trusts Security has been set to Enabled or to Disabled, configure only AND the Console user has logged in with administrative role Full Access. This setting is available in the Console at Setup > Global Settings.
For each administrative role, Trust permissions can be set to Deny, Read, Modify or Inherit:

- **Deny**: If this permission has been assigned, it is not possible to configure Trusts for the item. This is the default setting. This setting applies to new administrative roles and when upgrading your version to RES Automation Manager 2014 or higher.
- **Read**: If this permission has been assigned, a Console user can only view which Trusts are configured for the item. It is not possible to modify these Trusts.
- **Modify**: If this permission has been assigned, a Console user has all permissions to configure Trusts for the item.
- **Inherit**: If no particular permissions have been assigned (blank check box), the item will inherit the permissions from the above lying item, if applicable. If no above lying item exists, the default setting applies: Deny.

- Team folders and Resource folders show the name of the parent folder and the items they contain. This makes it easier to assign permissions to Trusts on these items. For consistency, Project folders and Run Book folders also show the name of the parent folder and the items they contain. It is not possible to configure Trusts on these items.

- Console users can only configure Trusts for items in the Console to which their administrative role grants access, so if they either have Read or Modify access to this item. When configuring an administrative role, you can use the Access column on the Permissions tab to assign access permissions to items:
  - If an administrative role denies access to a specific Resource, the Resource will not be shown when a Console user logs on with this administrative role: it will thus not be possible for him to configure Trusts for this Resource.
  - If an administrative role grants Read access to a specific Resource, it will not be possible for a Console user who logs on with this administrative role to modify any settings of the Resource except configure Trusts.
  - If an administrative role grants Modify access to a specific Resource, a Console user who logs on with this administrative role can modify all settings of the Resource, including configuring Trusts.
7.2.4 Assigning administrative roles to login accounts

When you have created administrative roles, you can assign them to login accounts.

1. Open the login account, select the Administrative Roles tab and click Add. The Select Administrative Roles window opens.
2. Select one or more administrative roles and click OK.

When a user logs on to the Console, all assigned administrative roles will be applied to the login account: if a login account has several administrative roles, with different levels of access to a specific node, folder or object, the user gets the highest of the assigned access levels. Modify takes precedence over Read and Deny, and Read takes precedence over Deny.

Example 1

Suppose 3 administrative roles have been assigned to a login account:

<table>
<thead>
<tr>
<th>Administrative role 1</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Topology node</td>
<td>Read</td>
<td></td>
</tr>
<tr>
<td>Library node</td>
<td>Modify</td>
<td></td>
</tr>
<tr>
<td>Jobs node</td>
<td>Deny</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Administrative role 2</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Topology node</td>
<td>Deny</td>
<td></td>
</tr>
<tr>
<td>Library node</td>
<td>Read</td>
<td></td>
</tr>
<tr>
<td>Jobs node</td>
<td>Read</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Administrative role 3</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Topology node</td>
<td>Deny</td>
<td></td>
</tr>
<tr>
<td>Library node</td>
<td>Modify</td>
<td></td>
</tr>
<tr>
<td>Jobs node</td>
<td>Modify</td>
<td></td>
</tr>
</tbody>
</table>

Result

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Topology node</td>
<td>Read</td>
</tr>
<tr>
<td>Library node</td>
<td>Modify</td>
</tr>
<tr>
<td>Jobs node</td>
<td>Modify</td>
</tr>
</tbody>
</table>

Example 2

You may have configured a Module that contains a Task Create Local User, which uses parameters for the logon name, full name and password of the user. If you then create an administrative role with Read access to the Topology nodes and the Library nodes, and Modify access to the Jobs nodes, any user with the administrative role will be able to schedule the Module in a Job and provide values for all parameters, but cannot edit the Module itself.
Example 3

You may have configured a Module that contains a Task Change Service Parameters that restarts a print spooler. If you then create an administrative role with Read access to this Module, and Modify access to the Jobs nodes, any user with the administrative role will only be able to schedule the Module in a Job, but no other Modules.

7.2.5 Secure delegation

Administrative roles allow you to delegate control over the configuration and execution of Tasks, so that users with a specific administrative role can schedule a Job, but cannot change the configuration of the Task itself.

Example 1

You may have configured a Module that contains a Task Create Local User, which uses parameters for the logon name, full name and password of the user. If you then create an administrative role with Read access to the Topology nodes and the Library nodes, and Modify access to the Jobs nodes, any user with the administrative role will be able to schedule the Module in a Job and provide values for all parameters, but cannot edit the Module itself.

Example 2

You may have configured a Module that contains a Task Change Service Parameters that restarts a print spooler. If you then create an administrative role with Read access to this Module, and Modify access to the Jobs nodes, any user with the administrative role will only be able to schedule the Module in a Job, but no other Modules.
Chapter 7: Administration

7.3 Components

The Components node allows you to save components (Agents or Agents+, Dispatchers, Consoles) as standalone MSI files that can be deployed manually or unattended.

7.3.1 Saving components as MSI files

At Administration > Components, you can save the RES ONE Automation components (Consoles, Dispatchers, Agents or Agents+) as standalone MSI files to a location of your choice, for example a share or a USB stick. You can use these files to deploy components manually or unattended.

- Deploy a component manually if, for example, a firewall makes it impossible to deploy it using the Console.
- Deploy a component unattended if, for example, you want to deploy it on many computers at once using a command line (see page 287). When you deploy a component unattended, it is not necessary to preconfigure it. Unattended deployments can be scripted, for RIS/WDS or BareMetal installations.

Install Consoles manually

If it is not possible to deploy Consoles directly from the Console, you can also install Consoles manually. This may be necessary if it is not possible to connect to the target computers, for example due to a firewall. This option is typically used to distribute components manually to other users who can install the component.

1. Open the Console and go to Administration > Components.
2. Right-click the file RES-ONE-Automation-Console-\{x.x.x.x\}.msi and choose Save component as.
3. Choose whether to preconfigure the MSI file. Preconfigured Consoles will automatically connect to a RES ONE Automation site with the specified settings after their installation.
4. Click OK and save the MSI file to the desired location.
5. At the desired location, double-click the MSI file and follow the prompts.
6. After installation, the new Console becomes visible in the Console.

Install Dispatchers manually

If it is not possible to deploy Dispatchers directly from the Console, you can also install Dispatchers manually. This may be necessary if it is not possible to connect to the target computers, for example due to a firewall. This option is typically used to distribute components manually to other users who can install the component.

1. Open the Console and go to Administration > Components.
2. Right-click the relevant installation file and choose Save component as:
   - RES-ONE-Automation-Dispatcher+(x64)-\{x.x.x.x\}.msi for 64-bit Microsoft Windows versions
   - RES-ONE-Automation-Dispatcher+(x86)-\{x.x.x.x\}.msi for 32-bit Microsoft Windows versions
3. Choose whether to preconfigure the MSI file. Preconfigured Dispatchers will automatically connect to a Datastore with the specified settings after their installation.
4. Click OK and save the MSI file to the desired location.
5. At the desired location, double-click the MSI file and follow the prompts.
6. After installation, the new Dispatcher will become visible in the Console.
Chapter 7: Administration

Note
It is also possible to install Dispatcher unattended. See Installing Dispatchers (see page 298).

Install Agents manually

If it is not possible to deploy Agents directly from the Console, you can also install them manually. This may be necessary if it is not possible to connect to the target computers, for example due to a firewall. This option is typically used to distribute components manually to other users who can install the component. For the legacy Agent, it is the only way to install the component.

1. Open the Console and go to Administration > Components.
2. Right-click the relevant installation file and choose Save component as:
   - RES-ONE-Automation-Agent-x.x.x.x.msi for Agents
   - RES-ONE-Automation-Agent+(x64)-x-x-x-x.msi for Agents+ on 64-bit Microsoft Windows versions
   - RES-ONE-Automation-Agent+(x86)-x-x-x-x.msi for Agents+ on 32-bit Microsoft Windows versions
3. Choose whether to preconfigure the MSI file. Preconfigured Agents will automatically connect with these settings to a RES ONE Automation environment when it is deployed. The following options are available (see also Global Settings (see page 35)):
   - Dispatcher discovery: Specifies the Agents’ search method for Dispatchers.
     - Use above configuration for first time deployment: Specify settings that are only applied during deployment. When the Agent has been deployed and connected to a Dispatcher with these initial settings, the Agent will from then on use the settings that are inherited from the global settings or the primary Team. This makes it possible to gain more control over the Agent deployment process. The option Use above configuration for first time deployment is only available when the option Use global setting has been selected.
   - Dispatcher locations: Specifies where Agents should search for Dispatchers.
   - Dispatcher recovery: Specifies which action Agents should take if no Dispatchers can be contacted.
   - Add to specified Teams: Adds the Agent automatically to a specific Team.
4. Click OK and save the MSI file to the desired location.
5. At the desired location, double-click the MSI file and follow the prompts.
6. Click Next to continue the installation.
7. Accept the license agreement and click Next.
8. Select the installation directory and click Next.
9. Click Install to start the actual installation process.
10. A window that contains all available RES ONE Automation environments shown when the installation has completed. This may take a few moments. If you want an Agent to connect to specific Dispatchers, you can enter the Dispatcher names in this window.
11. Click OK.
12. Select the appropriate RES ONE Automation environment and click OK.
13. Click Finish to complete the installation. After installation, the new Agent will become visible in the Console.

Note
Since version RES Automation Manager 2014, some additional security features have been implemented regarding the communication between the Agent and the Dispatcher. An Agent that is offline because the machine was down and then removed from the RES ONE Automation environment, will not be able reconnect to that environment. Also an Agent that is connected to environment ‘A’ cannot be moved to environment ‘B’. In both cases the Agent should be reinstalled.
7.3.2 Upgrading your RES ONE Automation environment

You can keep your version of RES ONE Automation up to date by installing RES ONE Automation Upgrade Packs or RES ONE Automation Service Packs. This requires Solution Assurance. Solution Assurance is a service that you can buy from RES Software, on the basis of a yearly subscription. If you subscribe to Solution Assurance, you are entitled to technical support, product upgrades, and access to the Online Knowledge Base.

When you buy RES ONE Automation, Solution Assurance is automatically included for one year. After that first year, your Solution Assurance subscription is automatically renewed, unless you cancel it. For more information about Solution Assurance, please contact your reseller.

RES ONE Automation Upgrade Packs and RES ONE Automation Service Packs are available at http://success.res.com, depending on the latest release of RES ONE Automation currently available.

- RES ONE Automation Upgrade Packs are available at the support portal after a major software release.
- RES ONE Automation Service Packs are available at the support portal after a service release. Upon their release, they replace the RES ONE Automation Upgrade Pack.

Please refer to the RES ONE Automation Upgrade Guide for more information.
7.4 Audit Trail

The Audit Trail allows you to view a complete log of all actions of RES ONE Automation. Because it also shows which account took the action, the audit trail provides you with a very useful tool for examining any alterations or actions that are performed in, or by RES ONE Automation.

An Audit Trail is available on three levels:

- **Global**: All actions.
- **Category**: A category of objects (all Consoles/Dispatchers/Agents, etc.).
- **Object**: A specific object (Console/Dispatcher/Agent, etc.).

**Global level**

At global level, you can see a complete log of all actions performed in or by RES ONE Automation. This log contains information about changes to any object in the environment, RES ONE Automation upgrades, etcetera.

- To view the complete Audit Trail, go to Administration > Audit Trail.
- Use the drop-down field to customize the number of Audit Trail results that are shown per page.
- To refresh the Audit Trail list, press F5.
- To clear the Audit Trail completely, right-click the list and click Clear Audit Trail. You can password-protect this option with the option Protect clear of audits in the Global Settings.

**Category level**

At category level, you can see a complete log of actions that apply to a specific category.

- To view the Audit Trail of a specific category, right-click the section and click Audit Trail.

**Object level**

At object level, you can see a complete log of changes that apply to a specific object.

- To view the Audit Trail of a specific object, open the item and select the Audit Trail tab.

### Notes

- When the number of Audit Trail results exceeds 32,000, these records will be truncated.
- The Audit Trail is only updated automatically while on page 1. When browsing to a different page, the Audit Trail is not updated with new results, until page 1 is reached again.
- When closing the Console, your selection is saved.
- When the Audit Trail contains a large number of results, (that is 20,000+) the loading of Audit Trail results is slow on Datastores based on Microsoft SQL Server.
Chapter 8: General functionality

- Use **Building Blocks** (on page 225) to export or import objects in an environment’s Library (Modules, Projects, Run Books and/or Resources), for backup purposes and to transport objects from one RES ONE Automation environment to another.

- Use the **Command Bar** (on page 232) to access a button bar that is visible throughout the Console, below the menu bar. It contains the most frequently used functions and features.

- Use **Conditions** (on page 233) to specify, based on expressions, whether a Task, Module or Run Book Job should be executed, skipped or failed.

- Use **Evaluators** (on page 237) in queries to let the execution of succeeding Tasks depend on the results of a Query.

- Use **Functions** (on page 246) to generate (partial) input for text-based properties of Tasks. This makes it possible to create information in various fields in Tasks automatically.

- Use **Parameters** (on page 257) to configure placeholders for values in various fields in Tasks, such as text, file paths, credentials, etc. This makes it possible to create generic Modules, Projects and Run Books that can be customized to each situation when required.

- Use **pattern matching, wildcards and operators** (on page 267) to define sets of objects, rather than a specific one in Tasks, Team rules, Conditions, and Evaluators.

- Use **Trusts** (see page 269) to configure Trust relationships between Agents and Modules and between Agents and Resources. These relationships determine whether or not an Agent can execute a Job with these Modules and Resources.

When closing the Console, its position is stored. As a result, the next time the Console is opened, it will open at the position it had before it was closed. This increases usability, as it becomes easier to navigate the Console. For example, if a specific Module was selected in the node **Library > Modules** before the Console was closed, the next time the Console will open at this Module. If the node **Jobs > Scheduling** was selected, the Console will open at this node. Any positions in the nodes **Administration > Audit Trail** and **Jobs > Job Results** are not stored.

The position of the Console is stored as a FIPS-compliant encrypted file for each individual user on the machine on which the Console is running, at:

- `%userprofile%\Local Settings\Application Data\RES\AM\cs_<guid of environment>.bin` (Microsoft Windows XP and Windows Server 2003)
- `%userprofile%\AppData\Local\RES\AM\cs_<guid of environment>.bin` (Windows Vista, Windows 7, Windows Server 2008 and higher).
8.1 Building Blocks

A Building Block stores items of a RES ONE Automation environment (Modules, Projects, Run Books, Resources, and Teams). A Building Block can be imported back into its original environment to recreate these items as a backup solution. A Building Block can also be imported into a different environment in order to copy the settings. This makes Building Blocks useful for change management, and data exchange.

A Building Block is an XML file that can be edited manually. This makes it possible to use a Building Block as a template: before importing a Building Block into another site, you can replace site-specific information such as server names with the information applicable in the target site.

AES-256 encryption ("Encryption using AES-256" on page 21)

- In an AES-256 encrypted RES ONE Automation environment, you can only create Building Blocks using AES-256 encryption. You are able to import legacy Building Blocks in an AES-256 encrypted environment.
- Although RES does not recommend it, it is possible to import Building Blocks created from an AES-256 encrypted Datastore into RES ONE Automation version 10 or lower. However, the following needs to be considered: since the legacy Datastore is not aware of AES-256 encryption, you will need to re-enter all password values used in Modules, Projects and Run Books.
- When creating an AES-256 encrypted Building Block, you need to generate a passkey. It is not possible to fill in a password manually or leave the Building Block password blank. The generated passkey can be copied to clipboard for future reference. When importing the Building Block, this passkey needs to be provided again.
- In these Building Blocks, all credentials and passwords (and parameters or Variables based on this type) used in Modules, Projects, Run Books and Resources are encrypted and decrypted with AES-256. You can identify the used encryption method by the m=aes256 label in the XML file.

8.1.1 Creating Building Blocks

The Building Blocks feature allows you to select which Module, Project, Run Book or Resource in your RES ONE Automation site should be included in a Building Block. You can only create Building Blocks of nodes and items to which your administrative role has access. For example, if your administrative role does not have access to the Resources node, you cannot create Building Blocks that include Resources.

You can create Building Blocks that include:

- only Modules, Projects, Run Books, Resources or Teams
- specific Modules, Projects, Run Books, Resources and/or Teams
- a specific Module, Project, Run Book, Resource or Team
- empty folders to recreate a complete folder structure (applicable to RES ONE Automation version 7.4.2.2 and later).

When a Building Block is created of Modules, Projects, Run Books and/or Teams, they are stored in a regular XML file (*.xml). Variables and related categories that are used in Modules, Projects and Run Books are also exported to the XML file. You can choose to save the Building Block with Resources. The Resources are then exported to an encrypted XBB file (*.xbb). When importing such a Building Block, the Resources are automatically included.
How to create

Node-specific

To create a Building Block of all Modules, Projects, Run Books, Resources or Teams:

1. Select the node and click in the command bar, and choose Create Building Blocks.
   - Alternatively, right-click the Modules, Projects, Run Books, Resources or Teams node and choose Building Blocks > Create Building Blocks from the context menu.
2. Specify a location, file name and type in the file save dialog.

Selected items

To create a Building Block of selected items:

1. Click in the command bar, and choose Select items for Building Blocks. This will expand the Console tree with check boxes. The Modules, Resources, Projects and Teams nodes will be selected by default.
   - Alternatively, right-click a Module, Project, Run Book, Resource or Team, and select Building Blocks > Select items for Building Blocks from the context menu.
2. Click in the command bar, and choose Create Building Blocks of selected items.
3. Specify a location, file name and type in the file save dialog.
4. To exit the selection modus, click in the command bar and clear the option Select items for Building Blocks.

Single item

To create a Building Block of a single item:

1. Select the item and click in the command bar, and choose Create Building Block "<item>".
   - Alternatively, right-click the item in the Modules, Projects, Run Books, Resources, or Teams node and select Building Blocks > Create Building Block "<item>" from the context menu.
2. Specify a location, file name and type in the file save dialog.

Building Blocks without Resources

When specifying the storage location of a Building Block, you can also save Building Blocks without Resources. Any Resources to which the items in the Building Block refer will not be saved as an XBB file. Upon import of such a Building Block in the target environment, it is then possible to add new Resources for the relevant items. This is useful when distributing Building Blocks to other RES ONE Automation environments: because it minimizes large Building Blocks, it becomes easier to distribute them. In addition, it makes it possible to comply to security policies of organizations in which the use of external Resources is not allowed.
Building Blocks with versioning information

Building Blocks of Modules, Projects, Run Books and Resources include versioning (see page 272) information. When importing a Building Block in an environment, these version numbers and version comments will be included.

- Objects that do not contain version numbers and version comments will be referred to as version number “1.0.0” and version comment “Initial version” in the Building Block.
- The version in the Building Block will be used as the new version, irrespective of the version in the target environment: Existing Modules, Projects, Run Books and Resources will be overwritten with the version in the Building Block.
- When importing Building Blocks without versioning information, the version number of the object will be set to “1.0.0” and the comment to “Imported via Building Block”.

Excluded

- Modules and Resources can contain Trusts on Teams and Agents. These are not included in a Building Block.
- Teams can contain fixed Agents, nested Teams and Job scheduling options. These are not included in a Building Block.
- The (encrypted) values for the passwords in the Variables of the type Password and Credentials are not included in a Building Block.

8.1.2 Importing Building Blocks

When you import a previously prepared Building Block, you can select which items should be imported in your RES ONE Automation environment. This allows you to import only those items that are relevant at that specific moment.

To import a Building Block, do the following:

1. Click in the command bar and choose Import Building Blocks (or in the Action menu > Building Blocks > Import Building Blocks).
   - Alternatively, select a specific node and right-click > Building Blocks > Import Building Blocks.
2. Select the Building Block and click Open. This opens the Import Building Block window.
3. Click Start Import. This opens the Import Building Block settings window.
   - To create a new folder in which the Building Block items will be imported, select Place in new root subfolder and specify a folder. Creating a folder in which Building Block items are imported makes it easier to retrieve any item that you imported. This is for example very useful when importing the Building Block in DTAP environments or at customer sites.
     - When specifying a subfolder in the root, this folder will be created in the root folder of the item in the Building Block: if it only contains Modules, it will be created in the Modules node; if it contains Modules, Projects and Run Books the folder will be created in the Modules, Projects and Run Books nodes.
     - When selecting Place in new root subfolder, the options Import Modules, Projects and Run Books with and Import Resources with will be unavailable: the Building Block items will always be imported using new GUIDs.
The receiving environment may contain previous versions of the imported items. These are identified by their Global Unique Identifier (GUID).

- To overwrite existing items, import items with the original GUIDs. When selecting the option original GUID’s, a confirmation message is shown for each item that is imported whether the existing item should be overwritten (Yes/No). You can apply this choice to all items that are imported and skip further messages from that point on by selecting Yes to All/No to All. This is very useful when importing large Building Blocks that contain many items: selecting one of the options drastically reduces the required configuration. The options Yes to All and No to All function independently of the option Do not ask input for any Parameters.

- To create new items, import items with new GUIDs. This may create duplicate items.

- To overrule parameter input settings when importing Building Blocks, select Do not ask input for any Parameters. When selecting this option, RES ONE Automation will not ask for parameter input, despite the input setting of individual parameters. This is very useful when importing large Building Blocks that contain many parameters, and whose default values do not need to be changed: selecting the option drastically reduces the required configuration.

- If you import a Building Block of items that refer to Resources that have been excluded when the Building Block was created, and the Resource is also not available in the target environment, a message will now be shown that the Resource is missing. When clicking OK, the New Resource window will then open, which makes it possible to create a new Resource for the relevant items.

Rules for Teams and Team Variables

- Teams will always be imported into an environment with their original GUIDs and folder structure, irrespective of the Import Building Block settings. This means that existing Teams in the target environment will be updated with rules, properties and Team settings from the Building Block. Any explicit members and automatic Job scheduling settings will remain untouched.

- If you import a Building Block of items that include Team Variables:
  - and the Team Variables already exist, the Team Variables will be overwritten with the value in the Building Block.
  - and the Team Variables do not yet exist, the global Variables will be created with an empty value, after which the Team Variables will get the value in the Building Block.

Rules for global Variables and their categories

- If you import a Building Block of items that includes Variables:
  - and the Variable does not yet exist in the environment in which the Building Block is imported (based on name), a new Variable of the same type will be created with the same value as in the originating environment. Team Variables and Agent Variables will inherit this value from the global Variable.
  - and the Variable already exists in the environment in which the Building Block is imported (based on name, irrespective of type), no Variable or category will be created or changed, nor will the value of the Variable be overwritten.
  - and the non-existing Variable is categorized in a category that does not yet exist in the receiving environment (based on GUID and then on name), a new category will be created and the Variables will be categorized in this new category.
  - and the non-existing Variable is categorized in a category that already exists in the receiving environment (based on GUID and then on name), the Variables will be categorized in the existing category.
  - If you import a Building Block into RES ONE Automation 2015 SR5 or earlier, the category information will not be retained and the Variable will be placed under the General category.
Rules for administrative roles and Building Blocks

When accessing the Console and administrative roles apply, this may impact the ability to import Building Blocks: You can only import those Building Block items into the related nodes to which your administrative role(s) grant Modify access, including all (sub)folders. When importing a Building Block, the Import Building Block window will show which Building Block settings you can import. Building Block items that relate to nodes to which your administrative role(s) do not grant access will be disabled.

For example, if a Building Block contains Resources and Modules, but you only have Modify access to Resources, it will now be possible to import Resources: the Modules in the Building Block will then be disabled in the modules disabled the Import Building Block window.

If you only have Read access to the Console, it is not possible to import Building Blocks and the import Building Block functionality will be disabled in the Console.

Tip

You may want to change the default values of Module, Project and Run Book parameters when importing items into another RES ONE Automation environment. For example, the receiving environment may need a different default domain, or different credentials. To set RES ONE Automation to prompt for new default parameter values when importing a Building Block, set the parameters’ Input settings to When importing Building Block. This allows you to create generic Modules, Projects and Run Books that can be customized when you import them as a Building Block.
8.1.3 The structure of Building Blocks

Because of its XML format, the contents of a Building Block can be edited manually using any text editor or XML-authoring solution.

Example 1

If you have installed a new file server to replace the existing one, all references in your RES ONE Automation site to the file server need to be changed.

Instead of editing each setting individually, do the following:

1. Create a Building Block of the Library settings (Resources, Modules, Projects, Run Books).
2. Open the Building Block XML file and replace all references to the old file server with references to the new one.
3. Import the edited Building Block back into the RES ONE Automation site. When prompted, select the original GUIDs, to overwrite the existing Resources, Modules, Projects and Run Books.

Example 2

If the Security Context password to execute a Task has changed in your RES ONE Automation site, all instances of this password in your RES ONE Automation site need to be changed:

1. Create a Building Block of the Library settings (Resources, Modules, Projects, Run Books) for backup purposes.
2. In the Console, change the Context Password in a Task.
3. Create another Building Block of the Library settings, with the changed Context Password.
4. Open the Building Block XML file in a text editor.
5. Use the Search functionality to search for and copy the changed Context Password.
6. Use the Search and Replace functionality to replace the other Context Passwords with the changed Context Password.
7. Save the changed Building Block.
8. Import the Building Block back into the RES ONE Automation site. When prompted, select the original GUIDs, to overwrite the existing Resources, Modules, Projects and Run Books.
9. Check your changes to see whether the actions succeeded. If necessary, import the backup Building Block to revert to your original settings.

⚠️ Warning

Editing a Building Block may lead to unexpected results. Before you modify a Building Block, make sure you have created a backup.
8.2 Command Bar

The command bar features an easy accessible button bar that is visible throughout the Console, below the menu bar. It contains the most frequently used functions and features:

Navigation

- Clicking will take you to the top node of the Console.
- Clicking and allows you to retrace your steps back and forth through the Console, similar to back/forward buttons in an Internet browser. You can retrace to up to 20 steps.

Advanced functions

The command bar replaces the buttons formerly used in feature windows (such as New, Edit, Delete, etc.). Depending which node is selected, different advanced functions are available in the command bar.

Common actions and support

In the command bar you will find the general buttons for Instant Reports, Building Blocks, Audit Trail, Video Tutorials and Help. Certain buttons, such as Instant Reports and Building Blocks, contain a drop-down menu with additional options.

You can use shortcut keys to perform actions from the command bar. For example:

- <Alt> + N: Create a new object
- <Alt> + E: Edit an existing object
- <Ctrl> + T: Audit Trail
- <Ctrl> + H: Home
- <Ctrl> + <: Back
- <Ctrl> + >:: Forward
- <Ctrl> + F: Search
8.3 Conditions

When configuring Tasks, Modules, Projects or Run Books, you can optionally set a condition. A condition determines to which settings a Task, Module or Run Book Job in a Job must comply before it is executed. A condition contains one or more expressions that determine whether the condition can be satisfied and defines what action should be taken based on this. This makes it possible to create intelligent Modules, Projects and Run Books.

Navigation
- At Library > Run Books, click the Jobs tab, click the Condition button.
- At Library > Projects, click the Modules tab, click the Condition button.
- At Library > Modules, click the Tasks tab, click the Condition button.

Configuration
- Task conditions determine whether a Task should be executed, skipped or failed and can be set when configuring Tasks and Modules.
- Module conditions determine whether a Module should be executed or skipped and can be set when configuring Projects.
- Job conditions determine whether a Run Book Job should be executed or skipped and can be set when configuring Run Books. When configuring Job conditions, only the expression types Date Time, Parameter and Status of Previously Executed Job are available.

Notes
- When Projects or Run Books are executed, all conditions are evaluated, including Task and Module conditions.
- The details of a Job result show information about (satisfied) conditions as they existed during execution of the Job.
8.3.1 Settings

After adding a condition, click Add and select the expression type. You can add as many expressions as needed.

Condition expressions can be based on:

- Computer Function
- Date Time
- Environment Variable
- File/Folder Operations (see page 235)
  - File Version
  - File Exists
  - Folder Exists
- Fully Qualified Domain Name
- Operating System
  - Bit Version
  - Suite
  - Type
  - Version
- Parameter (see page 235)
- Processor Architecture
- Registry Setting (see page 235)
- Status of previously executed Task/Job

See Pattern matching, wildcards and operators (on page 267) for an overview of all operators.

After configuring the expressions, you need to define whether all expressions in the condition need to be satisfied, or at least one. You also need to define which action should be carried out if the expression has or has not been met:

- **Fail this Task** (only available if you configure a Task): The Task will show a Failed status in the Job history.
- **Skip this Task/Module/Job**: Skips the Task, Module, or Job and continues with the next.
- **Skip this Task/Module/Job and all remaining Tasks/Modules/Jobs in this Module/Project/Run Book**: Skips this Task, Module or Job, and all remaining Tasks, Modules, or Jobs in the Module, Project, or Run Book.
- **Execute this Task/Module/Job**: Executes the Task, Module, or Job.
- **Execute this Task/Module/Job, but skip all remaining Tasks/Modules/Jobs**: Executes the Task, Module, or Job and skips all remaining Tasks, Modules, or Jobs.
- **Set parameter**: Sets a specific parameter when the condition is met. For example, if you configure a Change Service Parameters Task to restart a certain service on a number of Agents, this service can have a different name, depending on the operating system of the Agent. By letting the condition result specify the correct parameter value (for example, the Windows Firewall services SharedAccess or mpssvc), the Task will restart the correct service on each Agent.

To check the validity of the condition on your computer click Validate.
Example

For example, a Module could contain the following:

- Task Export Exchange Mailbox
- Task Disable Exchange Mailbox
- Task Send E-mail

If the Task Disable Exchange Mailbox fails, the administrator should receive a warning by e-mail. To achieve this, you can set the following condition on the Conditions tab of the Task Send E-mail:

- Expression: Status of previously executed Task = Failed.
- If this condition is FALSE then: skip this Task
- If this condition is TRUE then: execute this Task

Notes

- In Run Books, it is not possible to use parameter values from a CSV file in conditions based on parameters whose values are based on other parameters. See Using parameter values from a CSV file (on page 61).
- The use of global Variables in Run Book parameters, Run Book conditions and Run Book evaluators is not supported.

8.3.2 Conditions based on existence of file or folder

When basing the value of a condition expression on the existence or version of a file or folder:

- You cannot use functions in the Operand field.
- You can use parameters and global Variables in the Operand field.
- It is possible to use environment variables in the Operand field: right-click and select Replace folder names with their associated environment string.

8.3.3 Conditions based on parameters

When basing the value of a condition expression on a parameter, the value of this parameter can be based on another parameter or a function. Except for password parameters based on another parameter.

Using parameter values to skip or execute Modules, Projects, and Jobs

One of the expressions that you can use in a condition is Parameter:

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

Another example is a Module that contains two Tasks to create a mailbox for a user on a Microsoft Exchange server: one Task creates a mailbox on Exchange server A, and the other creates a mailbox on Exchange server B. By setting a condition based on a parameter with as value the surname of the user, the mailboxes will be created on Exchange server A or B, depending on the surname of the user that schedules the Module in a Job: 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.)
8.3.4 Conditions based on registry settings

When the value of a condition expression is based on a registry setting:

- You can specify registry keys with a backslash (\) at the end of a setting (as in HKEY_LOCAL_MACHINE\SOFTWARE\RES\AutomationManager\WMC\).
- You can specify registry values without a backslash (as in HKEY_LOCAL_MACHINE\SOFTWARE\RES\AutomationManager\WMC\DBServer).
- You can also check whether a registry setting actually exists, by using the operator Exist. This allows you to create Tasks that are only executed if a specific registry setting exists in the registry of the Agent. This can be useful for example, if a Task should only be executed if a specific version of an application has been installed, but this information is only stored as a registry key.

Example

Adobe Acrobat Reader stores its version number (for example, 10.0) in the registry as HKEY_CURRENT_USER\SOFTWARE\Adobe\Acrobat Reader\10.0\.

If a certain Task should only be executed if Adobe Acrobat Reader 10.0 has been installed, you can set a condition for this Task based on the registry key HKEY_CURRENT_USER\SOFTWARE\Adobe\Acrobat Reader\10.0\. If you use the operator Exist with the value True, RES ONE Automation will check the existence of this key in the registry of the Agents that perform the Task. If the condition returns True, this means that the registry key exists and thus that Adobe Acrobat Reader 10.0 has been installed.
8.4 Evaluators

For a number of Tasks that query objects, it is possible to configure Evaluators. Evaluators are very similar to conditions, but where a condition determines to what settings a Task must comply before it is executed, an evaluator does this afterwards: An evaluator contains one or more expressions that determine whether the evaluator can be satisfied based on the query results and defines what action should be taken based on this.

Different from conditions, evaluators do not contain a fixed set of available criteria on which you can base expressions. The list of available criteria depends on the nature of the Task in which the evaluator is used. This makes it possible to let the execution of succeeding Tasks depend on the results of a query. An evaluator is optional.

**Scenario: Install applications only on Agents with sufficient disk space** (on page 239)

**Scenario: Add Active Directory users to Distribution Lists** (on page 241)

You can also use evaluators to check Agent on compliancy. For example, you can configure an evaluator

- for a Task **Query Service Properties** to check whether a certain service is running on an Agent.
- for a Task **Query Disk Space** to check the minimum disk space on an Agent.
- for a Task **Query Installed programs** to check whether a certain application version has been installed.

**Navigation**

Add or edit one of the following Tasks and click the **Evaluators** tab:

- Query Active Directory Computer
- Query Active Directory Group
- Query Active Directory Object
- Query Active Directory Organizational Unit (OU)
- Query Active Directory User
- Query Certificate
- Query Computer Inventory
- Query Computer Properties
- Query Disk Space
- Query Environment Variables
- Query Event Logs
- Query Files
- Query Installed Programs
- Query Local Group
- Query Parameters
- Query Printer
- Query Printer Driver
- Query Service Properties
- Query TCP/IP Properties
8.4.1 Settings

To add an evaluator to a Task, select its Evaluator tab when setting up the Task. After this, click Add and select the evaluator type. Because each of the mentioned Tasks queries on different properties, the types of evaluator that you can select depend on the Task in which they are used. After selecting the evaluator type, the Add/Edit Expression window opens. You can add as many expressions as needed.

See Pattern matching, wildcards and operators (on page 267) for an overview of all operators.

After configuring the expressions, you need to define how many expressions in the evaluator need to be satisfied. You also need to define which action should be taken if the evaluator does or does not return data:

- **Continue**: The next Task in the Job will be executed.
- **Fail this query**: The Task will show a Failed status at Job History.
- **Set Parameter**: Optionally, you can set a parameter as part of the action of an evaluator. This allows you to use a value that is returned by an evaluator in a succeeding Task, for example as part of a condition on this Task.

<table>
<thead>
<tr>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>When viewing the detailed Job results of these Tasks, use the option Only show data returned by evaluator to filter the results of the query.</td>
</tr>
<tr>
<td>Because an evaluator can specify that a query should fail, the Tasks that can be configured with an evaluator also contain error control settings. The default value of error control for these Tasks is Continue Job on error, whereas the default value of other Tasks is Stop Job on error.</td>
</tr>
<tr>
<td>The use of global Variables in Run Book parameters, Run Book conditions and Run Book evaluators is not supported.</td>
</tr>
</tbody>
</table>
8.4.2 Scenario: Install applications only on Agents with sufficient disk space

Suppose you need to install an application on one or more Agents, but you are not sure whether all Agents have sufficient disk space. To handle this situation, do the following:
1. Create a Module containing a Task **Query Disk Space** and a Task **Install Windows Installer Package**.

2. In the Task **Query Disk Space**, configure an evaluator that defines that the Task should fail if the target disk on the Agent has less than 100 MB free disk space available:

   ![Query Disk Space](image1)

3. In the **Install Windows Installer Package**, specify the Windows Installer Package and configure a condition that defines that the Task should only be executed if the previous Task completed:

   ![Install Windows Installer Package](image2)
4. When you execute the Module in a Job, the Task **Install Windows Installer Package** will only be executed on Agents whose target disk has at least 100 MB free disk space available. On Agents with less disk space, the Task will fail immediately.

**8.4.3 Scenario: Add Active Directory users to Distribution Lists**

Suppose you want to add a user in Active Directory to one or more Distribution Lists, based on user logon name and group membership. However, you are not sure that the user actually exists in Active Directory. Also, you want make any Task generic, so it can also be used for different users. To handle this situation, do the following:

1. At Library > Modules, create two Modules, a Module **Query User Properties** and a Module **Add User to DL_Group**.
2. The Module **Query User Properties** contains two Tasks **Query Active Directory User** and a Task **Message Box**.
3. In the first Task **Query Active Directory User**, click the **Settings** tab and select Filter by OU and Include child organizational units.
4. On the **User Properties** tab, add the following AD user properties: Folder, Name, Department, Member of and User logon name (pre-Windows 2000).
5. On the **Module Parameters** tab, click **AutoCreate > All**.
6. In addition to the automatically-created parameters, also create the parameters **$[Department]**, **$[UserLogonName]** and **$[MemberOf]**:

![Query Active Directory User](image)
7. Configure an evaluator that defines that the Task should fail if the specified user does not exist in Active Directory. If the user exists, the parameter $\text{[Department]}$ should get the same value as the Active Directory user property Department:

![Evaluator Configuration](image)

8. In the second Task Query Active Directory User, click the Settings tab and select Filter by OU and Include child organizational units.

9. In the Domain field, specify the parameter $\text{[Domain]}$.

10. In the Security context field, specify the parameter $\text{[SecurityContext]}$.

11. In the Domain controller field, specify the parameter $\text{[DomainController]}$.

12. In the Filter by OU field, specify the parameter $\text{[FilterOU]}$.

13. On the User Properties tab, add the following AD user properties: Folder, Name, Department, Member of and User logon name (pre-Windows 2000).
14. Configure an evaluator that defines that the Task should fail if the specified user does not exist in Active Directory. If the user exists, the parameter $[MemberOf]$ should get the same value as the Active Directory user property Member Of:

15. In the Task Message Box, create a message with caption Query Active Directory Directory User (UserLogonName): $[UserLogonName]$ and message Department: $[Department]$, Member Of: $[MemberOf]$.


17. On the Settings tab, specify the parameter $[Domain]$.


19. In the Domain controller field, specify the parameter $[DomainController]$.

20. Select Single User and specify the parameter $[UserLogonName]$ in the User logon name field.

21. On the Member of tab, add the group DL_$[Department]$ with as action Add to group:

22. At Library > Run Books, create a Run Book with three Run Book Jobs.

23. In the first Run Book Job, on the Properties tab, select the Module Query User Properties in the What field.


25. In the second Run Book Job, on the Properties tab, select the Module Add User to DL_Group in the What field.


27. In the third Run Book Job, on the Properties tab, select the Module Query User Properties in the What field.
28. Select **Use Run Book Parameter** and select the parameter $[$RunBookWho$].

29. On the **Run Book Parameters** tab of the Run Book, click **AutoCreate** and specify values that apply to your RES ONE Automation environment for the parameters $[$FilterOU$], $[$Domain$], $[$DomainController$] and $[$SecurityContext$]. For example:
30. On the Links tab, AutoLink the Module parameters to the Run Book parameters with their default action **Set initial value**.

31. Change the parameter link action for the parameter $[\text{Department}]$ for the first link to **Get initial value**:

32. After configuring the Run Book, schedule a Job with it. On the **Job Parameters** tab, specify the user logon name of the user whose settings you want to change and specify which Agent should execute the Task.

When the Run Book is executed, a message box shows the logon name of the user, which department the user belongs to, and to which Distribution Lists the user has been added:
8.5 Functions

Besides parameters, you can also use functions to build generic Modules, Projects and Run Books. Functions can generate (partial) input for text-based properties of Tasks and for parameter values. This makes it possible to create information in fields automatically.

Examples

• A log file `c:\log\log.txt` is generated daily. You want to store each log file rather than overwriting it each day. This can be achieved by renaming each new log file to show the creation date, using the function `@[DATETIME]`:

  In the Task Perform File Operations, type `c:\log\log.txt` in the field Source path, and `c:\log\@[DATETIME(DDMMYY)]log.txt` in the field Destination path.

  Each time this Task is executed, the file `c:\log\log.txt` is renamed to include the date at that moment. If the Task runs on 25 September 2015, the file becomes `092515log.txt`, the next day `092615log.txt`, etc.

• When creating new Active Directory users, you want to generate a User Logon Name that has the first three letters of a user’s first name followed by the full surname. You fill the first name and last name using parameters. This can be achieved using the function `@[SUBSTRING]`:

  In the Task Create Active Directory User, fill the fields First name and Last name with the parameters `[firstname] and `[lastname].

  In the field User logon name, type: `@[SUBSTRING([firstname],1,3)]` `[lastname]` (Here the SUBSTRING function takes the first three characters of the `[firstname] parameter’s value).

  Each time this Task is executed, the User Logon Name is generated out of the values provided for the parameters for first name and last name. For Amanda Cavendish, for example, resulting User Logon Name is AmaCavendish. For John Smith, it is JohSmith.

Configuration

Set up Tasks using a function

• Open a Module at Library > Module, open a Task in the Module, click the Settings tab, right-click a text field, select Insert Functions and click the relevant function. This functionality may not be available, depending on the type of Task.

Set up parameters using a function

• Add or edit any type of parameter, click the Properties tab, right-click a field that specifies the value of the parameter, select Insert Functions and click the relevant function. The name of this field may differ, depending on the type of Task. This functionality may not be available, depending on the type of Task.
### 8.5.1 Overview

Functions are inserted by right-clicking in a text field when configuring or editing a Task in a Module. The selected function is inserted into the field as a pattern for the necessary format. If relevant, change the pattern to the actual function format. For example, RES ONE Automation will provide the pattern `@@[DATETIME(<format>)]`, but you need to fill out the desired format for the information.

<table>
<thead>
<tr>
<th>Function</th>
<th>Returns</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>@[CALC(&lt;num1&gt;,&lt;{+,-,*,/}&gt;,&lt;num2&gt;)]</code></td>
<td>Returns a calculated value.</td>
<td>If <code>@[GET-MEMORY]</code> returns &quot;2145456128&quot;, then <code>@[CALC([GET-MEMORY],*,2)]</code> returns &quot;4290912256&quot;</td>
</tr>
<tr>
<td><code>@[DATETIME(&lt;format&gt;)]</code></td>
<td>the current date and/or time in the specified format.</td>
<td>On September 25, 2015 at 17:04:23:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>for Agents:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <code>@[DATETIME(YYYYMMDDHHNNSS)]</code> returns &quot;20150925170423&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <code>@[DATETIME(YYYY-MM-DD HH:NN)]</code> returns &quot;2015-09-25 17:04&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <code>@[DATETIME(HH:NN)]</code> returns &quot;17:04&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <code>@[DATETIME(SS)]</code> returns &quot;23&quot;</td>
</tr>
<tr>
<td><code>@[FILEVERSION(&lt;filename&gt;)]</code></td>
<td>the file version of the specified file.</td>
<td>&quot;1.00.0001&quot;</td>
</tr>
<tr>
<td><code>@[FINDINFILE(&lt;filename&gt;,&lt;search&gt;,&lt;return offset&gt;,&lt;return length&gt;)]</code></td>
<td>a string from the specified section in a specific file.</td>
<td><code>@[FINDINFILE(C:\\WINDOWS\system32\drivers\etc\hosts,localhost,1,5)]</code> returns &quot;ocalh&quot;</td>
</tr>
<tr>
<td><code>@[GET-ASSETTAG]</code></td>
<td>any asset tag of the producer of the computer of the Agent.</td>
<td></td>
</tr>
<tr>
<td><code>@[GET-COMPUTERFUNCTION]</code></td>
<td>the type of software the computer of the Agent is running.</td>
<td>RES ONE Automation can recognize the following computer functions:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• &quot;Microsoft SQL Server&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• &quot;Primary domain controller&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• &quot;Backup domain controller&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• &quot;Running the Timesource service&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• &quot;Apple File Protocol server&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• &quot;Novell server&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• &quot;LAN Manager 2.x domain Member&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• &quot;Sharing Print queue&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• &quot;Running dial-in service&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• &quot;Xenix server&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• &quot;Microsoft File and Print for Netware&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• &quot;Running a Browser service as backup&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• &quot;Running the master Browser service&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• &quot;Running the domain master Browser&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• &quot;Terminal Server&quot;</td>
</tr>
<tr>
<td>Function</td>
<td>Returns</td>
<td>Example</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------------------------------------------------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>@GET-COMPUTERNAME</td>
<td>the computer name of the Agent.</td>
<td>&quot;NYHSQL01&quot;</td>
</tr>
<tr>
<td>@GET-FQDN</td>
<td>the fully qualified domain name of the Agent.</td>
<td>&quot;NYHSQL01.D-ENERGY.res.nl&quot;</td>
</tr>
<tr>
<td>@GET-IPV4</td>
<td>the IP address of the Agent that executes the Task. If an Agent has multiple network cards, RES ONE Automation will return the first IP address of the Agent as defined by its Network Connections settings.</td>
<td>&quot;172.16.41.20&quot;</td>
</tr>
<tr>
<td>@GET-IPV4ALL</td>
<td>all IP addresses of the Agent that executes the Task. If the Agent has multiple network cards, multiple entries will be returned, if the Agent only has one network card, only one IP address will be returned.</td>
<td>&quot;172.16.41.20;192.168.140.1;192.168.186.1&quot;</td>
</tr>
<tr>
<td>@GET-IPV6</td>
<td>the IPv6 address of the Agent that executes the Task. If an Agent has multiple network cards, RES ONE Automation will return the first IP address of the Agent as defined by its Network Connections settings.</td>
<td>&quot;2001:4016:1018:246:b4c8:d5e3:ac83:5855&quot;</td>
</tr>
<tr>
<td>@GET-IPV6ALL</td>
<td>all IPv6 addresses of the Agent that executes the Task, excluding link-local addresses. If the Agent has multiple network cards, multiple entries will be returned, if the Agent only has one network card, only one IP address will be returned.</td>
<td>&quot;2001:4016:1082:354:d1d6:ace4:8d8a:a2c3;2001:4016:1018:246:b4c8:d5e3:ac83:5855&quot;</td>
</tr>
<tr>
<td>@GET-MEMORY</td>
<td>the amount of memory installed on the Agent.</td>
<td>&quot;2145456128&quot;</td>
</tr>
<tr>
<td>@GET-OS-BIT</td>
<td>the bit version of the operating system of the Agent.</td>
<td>&quot;32&quot;</td>
</tr>
<tr>
<td>Function</td>
<td>Returns</td>
<td>Example</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>@[GET-OS-SUITES]</td>
<td>the product suites available on the operating system of the Agent.</td>
<td>RES ONE Automation can recognize the following product suites:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• &quot;BackOffice components installed&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• &quot;Windows Home Edition&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• &quot;Windows Server Compute Cluster Edition&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• &quot;Windows Server Datacenter Edition&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• &quot;Windows Server Enterprise Edition&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• &quot;Windows Server Web Edition&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• &quot;Windows Small Business Server&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• &quot;Windows Storage Server&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• &quot;Windows XP Embedded&quot;</td>
</tr>
<tr>
<td>@[GET-OS-TYPE]</td>
<td>the computer type of the Agent.</td>
<td>&quot;WORKSTATION&quot;</td>
</tr>
<tr>
<td>@[GET-OS-VERSION]</td>
<td>the version number of the operating system of the Agent.</td>
<td>If the operating system of the Agent is Microsoft Windows XP:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>@[GET-OS-VERSION] returns for example &quot;5.1.200&quot;</td>
</tr>
<tr>
<td>@[GUID]</td>
<td>an alpha-numeric string that is unique in the entire RES ONE Automation environment, enclosed in curly brackets.</td>
<td>{9C15453A-DDDE-4EB1-BC4D-0E6AE3BBF17E}</td>
</tr>
<tr>
<td>@[LOWER(&lt;value&gt;)]</td>
<td>sets all characters in a value in lower case. This function is typically used in combination with parameters and/or other functions, and is useful for example when provisioning user accounts, when it is necessary to control lower-case characters as part of best practice naming conventions for FQDN domain names, Login names, etc.</td>
<td>If you create a parameter $[FQDN server] and the FQDN of a server is &quot;SRV01.demo.com&quot;, @[LOWER($[FQDN server])] returns &quot;srv01.demo.com&quot;.</td>
</tr>
<tr>
<td>@[MASTERJOBGUID]</td>
<td>the unique identifier of the master (Module, Project or Run Book) Job that is running.</td>
<td>If you export the job results based on the Master Job GUID, for example: {7DF4C174-3C1C-46DD-8454-C333CD8613F2}</td>
</tr>
<tr>
<td>@[MAX(&lt;num1&gt;,&lt;num2&gt;)]</td>
<td>the maximum value of two specified values.</td>
<td>If @[GET-MEMORY] returns &quot;2145456128&quot;, then @[MAX(4000000000,@[GET-MEMORY])] returns &quot;4000000000&quot;</td>
</tr>
<tr>
<td>@[MIN(&lt;num1&gt;,&lt;num2&gt;)]</td>
<td>the minimum value of two specified values.</td>
<td>If @[GET-MEMORY] returns &quot;2145456128&quot;, then @[MIN(4000000000,@[GET-MEMORY])] returns &quot;2145456128&quot;</td>
</tr>
<tr>
<td>Function</td>
<td>Returns</td>
<td>Example</td>
</tr>
<tr>
<td>----------</td>
<td>---------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| `@RANDOM(<min>,<max>)` | a random decimal number between the set minimum and maximum.  
  - minimum allowed value: 0  
  - maximum allowed value: 2147483647 | `@RANDOM(1,100)` returns a number from 1 up to and including 100 |
| `@RANDOM(<min>,<max>,(<returnhex>))` | a hexadecimal value for a random decimal number between the set minimum and maximum.  
  `<returnhex>` is an optional value. Please remove the brackets when using it in a function. | `@RANDOM(1,100,1)` returns a hexadecimal value for a number from 1 up to and including 100 |
| `@REGISTRY(<registry value>)` | the value of the specified registry value name. | If your RES ONE Automation environment uses an MSSQL database:  
  `@REGISTRY(HKEY_LOCAL_MACHINE\SOFTWARE\RES\AutomationManager\WMC\DBType)` returns “MSSQL” |
| `@REPLACE(<value>,<find>,<replace>)` | replaces all specified characters in the given expression with other characters. | On the E-mail Addresses tab of the Task Manage Active Directory User, you can base the user’s e-mail address on a variable, such as %d for the display name. Use the REPLACE function to replace blanks in the display name with another character.  
  For example, with the display name Dan van Zandt, `@REPLACE(%d@d-energy.com, ,)` returns DanVanZandt@d-energy.com. The blanks have been deleted. |
| `@REPLACE-DIACRITICS(<value>)` | convert characters containing diacritics to an ASCII equivalent. For example, if you use data from an external HR system to create new users in Active Directory, but you do not want user names to contain diacritics, you can use this function to prevent this. | |
| `@SUBSTRING(<value>,<start>)` | all the characters from the given string, starting from the given position. | `@SUBSTRING(Robert,2)` returns “obert” (= the 2nd character and all following characters of the string “Robert”) |
# Chapter 8: General functionality

<table>
<thead>
<tr>
<th>Function</th>
<th>Returns</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>@SUBSTRING(&lt;value&gt;,&lt;start&gt;,(&lt;length&gt;))</td>
<td>all the characters from the given string, starting from the given position, with a maximum number of characters returned. (&lt;length&gt;) is an optional value. Please remove the brackets when using it in a function.</td>
<td>@SUBSTRING(Robert,2,3) returns &quot;obe&quot; (= the 2nd character and subsequent characters up to a maximum of 3 characters)</td>
</tr>
<tr>
<td>@UPPER(&lt;value&gt;)</td>
<td>sets all characters in a value in upper case. This function is typically used in combination with parameters and/or other functions, and is useful for example when provisioning user accounts, when it is necessary to control upper-case characters as part of best practice naming conventions for FQDN domain names, Login names, etc.</td>
<td>If you create a parameter $[FQDN server] and the FQDN of a server is &quot;srv01.DEMO.COM&quot;, @UPPER($[FQDN server]) returns &quot;SRV01.DEMO.COM&quot;.</td>
</tr>
</tbody>
</table>
Diacritics and their Conversion

By using the function @[REPLACE-DIACRITICS(<value>), you can convert any character containing diacritics to an ASCII equivalent. For example, if you use data from an external HR system to create new users in Active Directory, but you do not want user names to contain diacritics, you can use this function to prevent this.

The table below gives an overview of the most common characters containing diacritics and how they are converted using the function @[REPLACE-DIACRITICS]:

<table>
<thead>
<tr>
<th>Diacritic</th>
<th>Standard</th>
</tr>
</thead>
<tbody>
<tr>
<td>à</td>
<td>a</td>
</tr>
<tr>
<td>á</td>
<td>a</td>
</tr>
<tr>
<td>â</td>
<td>a</td>
</tr>
<tr>
<td>ã</td>
<td>a</td>
</tr>
<tr>
<td>ä</td>
<td>ae</td>
</tr>
<tr>
<td>Â</td>
<td>Ae</td>
</tr>
<tr>
<td>â</td>
<td>aa</td>
</tr>
<tr>
<td>ä</td>
<td>a</td>
</tr>
<tr>
<td>â</td>
<td>a</td>
</tr>
<tr>
<td>ã</td>
<td>a</td>
</tr>
<tr>
<td>æ</td>
<td>ae</td>
</tr>
<tr>
<td>ç</td>
<td>c</td>
</tr>
<tr>
<td>č</td>
<td>c</td>
</tr>
<tr>
<td>ć</td>
<td>c</td>
</tr>
<tr>
<td>ĉ</td>
<td>c</td>
</tr>
<tr>
<td>ċ</td>
<td>c</td>
</tr>
<tr>
<td>d’</td>
<td>d</td>
</tr>
<tr>
<td>d</td>
<td>d</td>
</tr>
<tr>
<td>ë</td>
<td>e</td>
</tr>
<tr>
<td>é</td>
<td>e</td>
</tr>
<tr>
<td>è</td>
<td>e</td>
</tr>
<tr>
<td>ë</td>
<td>e</td>
</tr>
<tr>
<td>ė</td>
<td>e</td>
</tr>
<tr>
<td>ē</td>
<td>e</td>
</tr>
<tr>
<td>é</td>
<td>e</td>
</tr>
<tr>
<td>ê</td>
<td>e</td>
</tr>
<tr>
<td>ĝ</td>
<td>g</td>
</tr>
<tr>
<td>ģ</td>
<td>g</td>
</tr>
<tr>
<td>ģ</td>
<td>g</td>
</tr>
</tbody>
</table>
Chapter 8: General functionality

Diacritic

Standard

ģ

g

ĥ

h

ħ

h

ì

i

í

i

î

i

ï

i

ĩ

i

ī

i

ĭ

i

ı

i

į

i

ĳ

ij

ĵ

j

ķ

k

ĸ

k

ĺ

l

ļ

l

ľ

l

ŀ

l

ł

l

ñ

n

ń

n

ņ

n

ň

n

ŋ

n

ò

o

ó

o

ô

o

ö

oe

Ö

Oe

õ

o

ø

oe

ō

o

ŏ

o

ő

o

œ

oe
Copyright © RES Software Development B.V. All rights reserved.

253


<table>
<thead>
<tr>
<th>Diacritic</th>
<th>Standard</th>
</tr>
</thead>
<tbody>
<tr>
<td>ř</td>
<td>r</td>
</tr>
<tr>
<td>ř</td>
<td>r</td>
</tr>
<tr>
<td>ř</td>
<td>r</td>
</tr>
<tr>
<td>š</td>
<td>s</td>
</tr>
<tr>
<td>š</td>
<td>s</td>
</tr>
<tr>
<td>š</td>
<td>s</td>
</tr>
<tr>
<td>š</td>
<td>s</td>
</tr>
<tr>
<td>Š</td>
<td>ss</td>
</tr>
<tr>
<td>ť</td>
<td>t</td>
</tr>
<tr>
<td>ť'</td>
<td>t</td>
</tr>
<tr>
<td>ţ</td>
<td>t</td>
</tr>
<tr>
<td>þ</td>
<td>th</td>
</tr>
<tr>
<td>ū</td>
<td>u</td>
</tr>
<tr>
<td>ū</td>
<td>u</td>
</tr>
<tr>
<td>ū</td>
<td>u</td>
</tr>
<tr>
<td>ū</td>
<td>ue</td>
</tr>
<tr>
<td>Ü</td>
<td>Ue</td>
</tr>
<tr>
<td>ū</td>
<td>u</td>
</tr>
<tr>
<td>ū</td>
<td>u</td>
</tr>
<tr>
<td>ū</td>
<td>u</td>
</tr>
<tr>
<td>ū</td>
<td>u</td>
</tr>
<tr>
<td>ū</td>
<td>u</td>
</tr>
<tr>
<td>ū</td>
<td>u</td>
</tr>
<tr>
<td>ū</td>
<td>u</td>
</tr>
<tr>
<td>ū</td>
<td>u</td>
</tr>
<tr>
<td>ą</td>
<td>u</td>
</tr>
<tr>
<td>ź</td>
<td>z</td>
</tr>
<tr>
<td>ź</td>
<td>z</td>
</tr>
<tr>
<td>ź</td>
<td>z</td>
</tr>
</tbody>
</table>
8.6 Instant Reports

With Instant Reports, you can create detailed reports of your RES ONE Automation environment. Instant Reports include all settings of the documented item(s), and can include additional information about the current RES ONE Automation version.

8.6.1 Where to find Instant Reports in the Console

<table>
<thead>
<tr>
<th>What</th>
<th>Where</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a detailed Instant Report of all items in your RES ONE Automation environment</td>
<td>Select the highest level in the Console tree and click 😊 in the command bar. -or- Right-click the highest level in the Console tree and choose Instant Reports &gt; Create Instant Report in the context menu.</td>
</tr>
<tr>
<td>Create a detailed Instant Report of all items in the Topology, Library or Jobs node</td>
<td>Select the highest level of the node and click 😊 in the command bar. -or- Right-click the highest level of the node and choose Instant Reports &gt; Create Instant Report in the context menu.</td>
</tr>
<tr>
<td>Create a detailed Instant Report of a set of selected items</td>
<td>Click the arrow next to 😊 in the command bar and choose Select items for Instant Report. Check boxes will appear in front of all items in the Console tree. After setting the selection, click the arrow next to 😊 in the command bar again and choose Create Instant Report of selected items. To exit the selection modus, click the arrow next to 😊 in the command bar and clear the option Select items for Instant Report. -or- Right-click anywhere in the active window and choose Instant Reports &gt; Select items for Instant Report in the context menu.</td>
</tr>
<tr>
<td>Create an Instant Report of a single item</td>
<td>Edit the item and click 😊.</td>
</tr>
<tr>
<td>Create an Instant Report with only an overview of all items in a specific node (for example the Modules node)</td>
<td>Select the specific node and click 😊 in the command bar. This type of Instant Report does not contain detailed information. -or- Right-click the specific node and choose Instant Reports &gt; Create Instant Report in the context menu.</td>
</tr>
</tbody>
</table>

8.6.2 Instant Report format

Instant Reports can be stored as a PDF file including a Table of Contents. Alternatively, they can be printed directly from the window View Instant Report.
8.6.3 Customizing the contents and presentation

The following customization choices are available for Instant Reports:

- Whether or not to include a cover page.
- Whether or not to include a title on the cover page. The title is editable.
- Whether or not to include a Table of Contents.
- Whether or not to include empty pages.

To change these settings, right-click the active window, point to Instant Reports and click Select items for Instant Report. At the top of the tree, the option Instant Report Settings becomes available.
8.7 Parameters

In any RES ONE Automation 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 have to provide a different user name. Of course, you could edit the original Module each time it is used. Next time you have to create a local user, you do the same again. However, it is more efficient to use parameters in the Task. Parameters function as placeholders for the values in various fields, such as text, file paths, credentials, etc. The actual values of these fields can be provided when they are needed: when the Task is scheduled or when it is imported as part of a Building Block. Optionally, you can provide default values for parameters when you create them and choose to overrule these default values at the moment of use.

Because you can also use parameters in Projects and in Run Books and link them to underlying parameters, this makes it possible to create generic Modules, Projects and Run Books that can be customized to each situation when required.

Generic Modules and Projects are also very useful for system integration purposes. If you create Building Blocks to transport Modules and Projects from one RES ONE Automation environment to another, the configuration of some Tasks may need to be changed to the new environment. This can be achieved efficiently using parameters.

For example, if you create a Building Block of a Task that joins a computer to a domain, many values need to be changed for the receiving environment. This can be set up efficiently by replacing the values in the Tasks with parameters. Then, each time the Building Block is imported in a different environment, RES ONE Automation will prompt for new values for the parameters.

Parameters have a number of uses:

- Allow parts of a 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 administrative 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 values from which to choose, making it impossible to enter other data in that field. As a result, you can control the kind of information that is used in a Task.
- 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 Modules, Projects or Run Books should be executed. 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.
Parameters can be renamed. For example, this makes it easier to correct earlier mistakes or changes in naming conventions. When renaming parameters, linked parameters may become unlinked. This may lead to unexpected results when executing Modules, Projects or Run Books in which they are used:

- If you rename a Module parameter that is linked to a Project parameter, the link will be broken.
- If you rename a Project parameter that is linked to a Module parameter, the link will remain.
- If you rename a Project parameter that is linked to a Run Book parameter, the link will be broken.
- If you rename a Run Book parameter that is linked to a Project parameter, the link will remain.
- The new name of the parameter will also be reflected when used in other parameters, conditions and evaluators.

### 8.7.1 Where to find parameters

<table>
<thead>
<tr>
<th>What</th>
<th>Where</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setting up Module parameters on Task level</td>
<td>• Open a Module at Library &gt; Module, open a Task in the Module, click the Module parameters tab, click Add</td>
</tr>
<tr>
<td></td>
<td>• or •</td>
</tr>
<tr>
<td></td>
<td>• Open a Module at Library &gt; Module, open a Task in the Module, right-click a text field on the Settings tab</td>
</tr>
<tr>
<td>Setting up Module parameters on Module level</td>
<td>• Open a Module at Library &gt; Module, click the Module parameters tab, click Add</td>
</tr>
<tr>
<td>Setting up Project parameters</td>
<td>• Open a Project at Library &gt; Project, click the Project parameters tab, click Add</td>
</tr>
<tr>
<td></td>
<td>• or •</td>
</tr>
<tr>
<td></td>
<td>• Open a Project at Library &gt; Project, click the Project parameters tab, click AutoCreate</td>
</tr>
<tr>
<td>Setting up Run Book parameters</td>
<td>• Open a Run Book at Library &gt; Run Book, click the Run Book parameters tab, click Add</td>
</tr>
<tr>
<td></td>
<td>• or •</td>
</tr>
<tr>
<td></td>
<td>• Open a Run Book at Library &gt; Run Book, click the Run Book parameters tab, click AutoCreate</td>
</tr>
</tbody>
</table>
8.7.2 Parameter types

You can use the following types of parameters:

- A **Credentials** parameter is a combination of domain\username and password. Use this in a username field for which there is also a password field, for example in the Security Context of a Task.
  - The matching password field will be disabled, as the Credentials parameter will fill both. The password is always masked.
  - You can use parameters to specify the user name. The use of parameters in passwords is not allowed, as it may lead to unpredictable results when linking in Projects and Run Books.

- A **List** parameter is a list with possible values for the parameter from which a value can be selected at the input moment of the parameter. Use this in text fields where only specific information is allowed, and where the options are known in advance. For example, in a Task Execute Database Statement, you could provide a list of Database servers.

- A **Multi-line text** parameter can be a text of several lines, containing line breaks. Use this, for example, for a message box text, Active Directory User Notes, or for the value for a multi-string registry value.
  - A Multi-line text parameter is displayed as a single line when the parameter is set.
  - When specifying the default value for this type of parameter, press ENTER to start on a new line. When providing a value at the input moment, press CTRL+ENTER to start a new line.

- A **Multi-select list** parameter is a list with possible values for the parameter from which multiple values can be selected at the input moment of the parameter. This multi selection is converted into a semicolon (;) separated string. Use this type of parameter in text fields where only specific information is allowed, where the options are known in advance, and where multiple entries are allowed. 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.

- A **Password** parameter is a password only. Use this in password fields that are not part of a set of credentials, for example for the password for a new local user.
  - The value of a Password parameter is always masked. Its input must be entered twice for confirmation.
  - You can use Variables and parameters as value. Please note the following:
    - The value of the parameter may contain only a single Variable or parameter, not multiple Variables and/or parameters (one after another) or variables and/or parameters inside Variables and/or parameters (for example “^[My password $[param]]”).
    - The value may NOT contain functions.
  - When a Password parameter value is changed in the Add/Edit Job window when the Job is scheduled, the string will be handled as a password.

- A **Text** parameter is a text string of a single line. Use this in any field that needs to contain text, for example file name, Organizational Unit name, etc.
  - The format of input for Text parameters can be restricted using an input mask (see Parameter Input Mask (on page 261)).

When scheduling a Job, you can use input for Text parameter and Password parameter values from a CSV file, instead of providing these values manually. This makes it possible to feed information into Tasks from a spreadsheet. See Using parameter values from a CSV file (on page 61).

Each parameter can have an optional default value. If a different parameter value is provided at the moment of use, the default value will be overwritten. You can set a manual value for the parameter, but you may also choose to base the value on other parameters, functions or Global Variables by right-clicking the Value field. Parameter values can have a maximum length of 512 characters.
When configuring List or Multi-select list parameters, you can specify the order in which the list values will be shown at the input moment by using the arrow buttons ↑ ↓.

When editing Module, Project, or Run Book parameters, it is possible to change the parameter type. When changing the parameter type:

- Any values will be lost.
- Linked parameters may become linked to parameters of a different type. This may lead to unexpected results when executing Modules, Projects or Run Books in which they are used. For example, a Text parameter may become linked to a credentials parameter.
- Parameters will become unlinked if the different types are not allowed to be linked. For example, Text parameters and List parameters cannot be linked.

### Notes

- The functionality to use Variables in parameters is supported for Text, Password and Credentials parameters (corresponding to the various Variables types). It is not supported for Run Book parameters.
- In previous versions of RES ONE Automation, parameter values in List and Multi-select List parameters were sorted alphabetically at the input moment. Although the specific order of parameter values will be retained when downgrading again to a previous version of RES ONE Automation, it will revert to an alphabetical order once the Task in which the parameters are used is edited again.

### 8.7.3 Input settings

For each parameter, various input settings can be configured on the parameter’s Input tab. This allows you to adjust Modules, Projects and Run Books to the specific needs of a situation before you use them.

#### Input new value

- **When importing Building Block**: Request input for a parameter when you import a Building Block that includes one or more parameters. This option is cleared by default.
  - **Show previous value**: Show the default value of the parameter, or the value that was used previously. This option is selected by default.
- **When scheduling Job**: Request input for a parameter when you schedule a Job that includes one or more parameters. This option is selected by default. When automatically creating parameters the option will be cleared by default.
  - **Show previous value**: Show the default value of the parameter, or the value that was used previously. This option is selected by default.
  - **Erase previous value when rescheduling**: Erase parameter values when a Job in which the parameter is used is rescheduled, and prompt for new values. This option increases security when used for example in Password and Credentials parameters.
Chapter 8: General functionality

Options

- **Hide parameter if parameter is not used directly**: Hide the parameter at the input moment if it is not used in any field. This option is selected by default.
- **Request input confirmation**: Request confirmation of the provided parameter value after the input moment.
- **Parameter value is required**: Make it mandatory to provide a parameter value at the input moment.
- **Use input mask**: Create a restrictive field for text entry. The values that you can enter at the input moment of the parameter are restricted to the format of the specified mask. See Input Mask (on page 261).

Tip

To change the input settings of several parameters at once, select them (on the Module parameters, Project parameters or Run Book parameters tab), right-click and choose Edit. Select the relevant input settings and click Edit.

8.7.4 Input Mask

By default, a parameter value can be any combination and number of characters. This may be undesirable, for example if the parameter specifies a phone number and you want all your phone numbers in Active Directory to follow a set pattern. In such cases, you can set an input mask.

An input mask defines which characters are allowed where in the parameter's value. Each mask character acts as a placeholder for one character of a specific type. For example, if a parameter specifies a phone number, you can set an input mask to enforce the input of exactly ten numbers. At the input moment, RES ONE Automation will translate the mask characters into underscores where the user is to provide input. If you set an input mask **1234560000**, this will be shown as **123456____** in the input field (because 0 is a mask character, whereas the other characters are not). The user will be able to enter numbers where the underscores are shown, but not other characters.

An input mask consists of two parts: the **Mask** and the **Literal**.

- The **Mask** contains mask characters. Each mask character acts as a placeholder for one character of a specific type. For example, 0 stands for a number between 0 and 9, and L stands for a lowercase or uppercase letter. Optionally, the mask may also contain regular characters that do not function as placeholders, but that are included in the parameter value.
- The **Literal** is filled automatically on the basis of the mask information. The literal shows where a user will be able to provide input by translating the mask characters into underscores. If a mask character has been translated into an underscore but should be kept as a regular character, you can toggle it back using the arrow up and down keys.
When setting input masks, you can combine regular characters with any of the mask characters from the table below:

<table>
<thead>
<tr>
<th>Mask character</th>
<th>Valid characters</th>
<th>Sample mask</th>
<th>Sample literal</th>
<th>Sample parameter value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0-9</td>
<td>Phone: (073)-000-0000</td>
<td>Phone: (073)-_<strong>-</strong>__</td>
<td>Phone: (073)-622-8800</td>
</tr>
<tr>
<td>9</td>
<td>0-9 or space (&quot; &quot;)</td>
<td>Phone: 999999999999</td>
<td>Phone: _______</td>
<td>Phone: 073 622 8800</td>
</tr>
<tr>
<td>#</td>
<td>0-9 or space (&quot; &quot;) or + or -</td>
<td>Phone: ####### ## ######</td>
<td>Phone: _______ _______</td>
<td>Phone: +0031 73 622 8800</td>
</tr>
<tr>
<td>L</td>
<td>a-Z</td>
<td>Alpha: LLLLLLL</td>
<td>Alpha: ____</td>
<td>Alpha: AbCdEfG</td>
</tr>
<tr>
<td>?</td>
<td>a-Z or space (&quot; &quot;)</td>
<td>Name: ?????????</td>
<td>Name: _____</td>
<td>Name: Dan Harris</td>
</tr>
<tr>
<td>A</td>
<td>0-9 and a-Z</td>
<td>User name: AAAAAA</td>
<td>User name: ____</td>
<td>User name: HarD1</td>
</tr>
<tr>
<td>a</td>
<td>0-9 and a-Z or space (&quot; &quot;)</td>
<td>User: aaaaaaa</td>
<td>User: _____</td>
<td>User: Harris 1</td>
</tr>
<tr>
<td>&amp;</td>
<td>All print characters</td>
<td>PrintChar: &amp; &amp; &amp; &amp; &amp; &amp;</td>
<td>PrintChar: ____</td>
<td>PrintChar: 5*X+[2n]</td>
</tr>
<tr>
<td>H</td>
<td>0-9 and A-F</td>
<td>Hex: 0xHHHHHHHH</td>
<td>Hex: 0x____</td>
<td>Hex: 0x002FD89A</td>
</tr>
<tr>
<td>X</td>
<td>0-9 and A-F and space (&quot; &quot;)</td>
<td>Hex: 0xXXXXXXX</td>
<td>Hex: 0x____</td>
<td>Hex: 0x1 FD89A</td>
</tr>
<tr>
<td>&gt;</td>
<td>Forces uppercase (A-Z)</td>
<td>Postal code: =&gt; 00000</td>
<td>Postal code: ____</td>
<td>Postal code: MI 48506</td>
</tr>
<tr>
<td>&lt;</td>
<td>Forces lowercase (a-z)</td>
<td>Title: &lt;&lt;&lt;&lt;&lt;&lt;</td>
<td>Title: ____</td>
<td>Title: manager</td>
</tr>
</tbody>
</table>

**Examples**

If you set an input mask `PHONE: 073-000000`, this will be shown as `P_ONE: _73-______` in the literal. Using the arrow up and down keys, you can toggle the relevant underscores, so that the literal shows as `PHONE: 073-_____`.

If you want to show a prompt for a telephone number that is restricted to numeric values only, but the area code should be in between parentheses, you could define your mask as `Phone No: (000) 000-0000`. This results in the following prompt: `Phone No: (___) ___-____`, in which only numeric values can be entered.

**Notes**

- You can test your input mask by clicking Show preview.
- If you set an input mask for a parameter that has a default value, the default value has to match the input mask.
8.7.5 Creating parameters automatically

When you create parameters for Modules, Projects or Run Books, you can use the AutoCreate functionality to create Module parameters, Project parameters and Run Book parameters automatically.

- On the Settings tab of a Task, you can create and insert individual Module parameters automatically in an active field, by right-clicking the field and selecting Insert Parameters > AutoCreate parameter.
- On the Module Parameters tab of a Task, you can automatically create sets of Module parameters that are inserted into the fields on the Settings tab of the Task, by clicking AutoCreate. You can create Module parameters for all fields or for empty fields only. Please note that if you choose to create parameters automatically for ALL fields in a Task, any values that already exist in those fields will be overwritten.
- On the Project Parameters tab of a Project, you can automatically create sets of Project parameters that match underlying Module parameters, by clicking AutoCreate. Each Project parameter that is created inherits all properties and values of the first matching Module parameter (that is, name, description, hint text, type, value, and input settings).
  - Optionally, newly created Project parameters can automatically be linked to the matching Module parameters at the same time. See Linking parameters (on page 264).
- On the Run Book Parameters tab of a Run Book, you can automatically create sets of Run Book parameters that match each underlying Job parameter, by clicking AutoCreate. Each Run Book parameter that is created inherits all properties and values of the first matching Job parameter (that is, name, description, hint text, type, value, and input settings).
  - Optionally, newly created Run Book parameters can automatically be linked to the matching Job parameters at the same time. See Linking parameters (on page 264).
- With the Cleanup button you can remove all unused parameters.
8.7.6 Linking parameters

When you combine multiple Modules into a Project, you can link any Module parameters in these Modules to corresponding Project parameters. Like Project parameters, a Run Book parameter is an umbrella parameter for a set of Module parameters and/or Project parameters that occur in one Run Book.

When linking parameters, you can specify which action should be performed by the link at the input moment to determine the value of a parameter. For example, this allows you to re-use the parameter values of a certain Module in a Project in another Module in the same Project. It also allows you to use the values of Module parameters as the value for Run Book parameters, which is especially useful in combination with RES ONE Identity Director.

You can choose between the following actions:

<table>
<thead>
<tr>
<th>Link action</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>None</strong></td>
<td>The parameter is not linked to any underlying parameters. As a result, each parameter keeps its own value.</td>
</tr>
<tr>
<td><strong>Set initial value</strong></td>
<td>This link action <em>sets</em> the parameter value <em>to</em> the underlying parameter to which it is linked:</td>
</tr>
<tr>
<td></td>
<td>• A Project parameter sets its value to the linked Module parameter <strong>before</strong> the Module is executed.</td>
</tr>
<tr>
<td></td>
<td>• A Run Book parameter sets its value to the linked Job parameter <strong>before</strong> the Run Book Job is executed.</td>
</tr>
<tr>
<td></td>
<td>This is the default action when auto-linking parameters and reflects the previous behavior of linked parameters.</td>
</tr>
<tr>
<td><strong>Get final value</strong></td>
<td>This link action <em>gets</em> the parameter value <em>from</em> the underlying parameter to which it is linked:</td>
</tr>
<tr>
<td></td>
<td>• A Project parameter gets its value from the linked Module parameter <strong>after</strong> the Module has been executed.</td>
</tr>
<tr>
<td></td>
<td>• A Run Book parameter gets its value from the linked Job parameter <strong>after</strong> the Run Book Job has been executed.</td>
</tr>
<tr>
<td><strong>Both</strong></td>
<td>This link action first <em>sets</em> the parameter value <em>to</em> the underlying parameter to which it is linked, and then <em>gets</em> its parameter again <em>from</em> the parameter to which it is linked:</td>
</tr>
<tr>
<td></td>
<td>• When used for a Project parameter sets its value to the linked Module parameter <strong>before</strong> the Module is executed. The Project parameter gets its value from the linked Module parameter again <strong>after</strong> the Module has been executed.</td>
</tr>
<tr>
<td></td>
<td>• When used for a Run Book parameter, the Run Book parameter sets its value to the linked Job parameter <strong>before</strong> the Run Book Job is executed. A Run Book parameter gets its value from the linked Job parameter again <strong>after</strong> the Run Book Job has been executed.</td>
</tr>
</tbody>
</table>
Example 1

Suppose you have a Run Book (or Project) with multiple Modules for the creation of an Active Directory User, including a Home Folder and a Mailbox.

- The Run Book uses Username as a Run Book parameter. For example, John Smith > smithj.
- The first Module contains some error handling in case jsmith already exits and changes the value of the parameter Username. For example, smithj02.
- The final value needs to be sent back to the Run Book (get final value) and overwrites the Run Book parameter.
- This new initial value (set initial value) will be used in the other Modules.

<table>
<thead>
<tr>
<th>RUN BOOK</th>
<th>MODULE 1 (Create User in AD)</th>
<th>MODULE 2 (Create Home Folder)</th>
<th>MODULE 3 (Create Mail Box)</th>
<th>MODULE 4 (Map Home Drive)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Username)</td>
<td>In: smithj</td>
<td>Out: smithj02</td>
<td>smithj02</td>
<td>smithj02</td>
</tr>
</tbody>
</table>

Linking Project parameters

You can link Project parameters to Module parameters when configuring Projects, in the New Project window at the tab Project Parameters > Links.

- To link all unlinked Module parameters to matching Project parameters automatically, click AutoLink. This automatically links existing Project parameters to matching Module parameters of the same name and type. AutoLinked Project parameters use the default link action Set initial value.
- When you create Project parameters and use the AutoLink functionality, you can optionally automatically create new Project parameters and link them to matching Module parameters. See Creating parameters automatically (on page 262).
- To link a Project parameter to one or more Module parameters manually, select an unlinked Module parameter and drag it to a Project parameter. After this, click Action to specify which action should apply to the link.
  - Alternatively, you can link a Project parameter to one or more Module parameters manually by editing the properties of the Project parameter. On the Links tab of the Add Project Parameter window, you can select to which Module parameter(s) the Project parameter should be linked. After linking the parameters, click Action to specify which action should apply to the link.
  - Linked Module parameters will appear below the Project parameter to which they are linked; unlinked Module parameters are listed separately.
  - To unlink a linked Module parameter, select the linked parameter and drag it to Not Linked at the top of the list.

Linking Run Book parameters

You can link Run Book parameters to Job parameters when configuring Run Books, in the New Run Book window at the tab Run Book Parameters > Links.

- To link all unlinked Job parameters to matching Run Book parameters automatically, click AutoLink. This automatically links existing Run Book parameters to matching Job parameters of the same name and type. AutoLinked Run Book parameters use the default link action Set initial value.
- When you create Run Book parameters and use the AutoLink functionality, you can optionally automatically create new Project parameters and link them to matching Module parameters.
To link a Run Book parameter to one or more Job parameters manually, select an unlinked Job parameter and drag it to a Run Book parameter. After this, click Action to specify which action should apply to the link.

Alternatively, you can link a Run Book parameter to one or more Job parameters manually by editing the properties of the Run Book parameter. On the Links tab of the Add Run Book Parameter window, you can select to which Job parameter(s) the Run Book parameter should be linked. After linking the parameters, click Action to specify which action should apply to the link.

Linked Job parameters will appear below the Run Book parameter to which they are linked; unlinked Job parameters are listed separately.

To unlink a linked Job parameter, select the linked parameter and drag it to Not Linked at the top of the list.

### Notes

- A Run Book parameter that is linked to a Module parameter with the link action Get final value gets its value from the last Agent that processes the last Module in the Run Book in which the Module parameter is used. If the Module fails on this Agent, no value is provided and the Run Book parameter keeps its default value.
- The use of global Variables in Run Book parameters, Run Book conditions and Run Book evaluators is not supported.
- Project and Run Book parameter links: linking parameters of the type List or Multi-select list containing credential parameters is not supported.

#### 8.7.7 Securing Modules, Projects and Run Books using parameters

With parameters, control over Job scheduling can be secured. For example, you can:

- use parameters to allow users with a specific administrative role to schedule Jobs and provide parameter values, but prevent them from changing Modules, Projects and Run Books.
- use parameters to protect Modules, Projects and Run Books with passwords that need to be provided during Job scheduling.

**Securing Job scheduling with an administrative role**

1. Configure the Modules, Projects or Run Books with parameters.
2. At Administration > Security, create an administrative role with read-only access to the nodes Modules, Projects, Run Books, Agents and Teams, and full access to the Job Execution node.
3. Assign this administrative role to login accounts. Users will be able to schedule Modules, Projects and Run Books, and can provide values for all parameters in the input window. However, they will not be able to edit Modules, Projects or Run Books.

**Securing Job scheduling with passwords**

1. Configure the Modules, Projects or Run Books by adding a Password parameter. It is not necessary that this parameter is actually used in a Module, Project or Run Book.
2. Create a Module, Project or Run Book condition based on the value of the Password parameter.
   - Make sure the Task, Module or Run Book Job is skipped if the person scheduling it does not provide the correct value for a password parameter.
8.8 Pattern matching, wildcards and operators

In many tasks, but also for example in team rules and conditions, wildcards and operators can be used to define a set of objects, rather than naming a specific object.

For example, you can use a pattern such as LIKE 10.1.7.* instead of a fixed list of IP addresses as a team rule. This rule will not only be applied to all existing agents whose IP address match the specified pattern: New agents are automatically added to the team if their IP addresses match the pattern provided.

Operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Stands for</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>Equals</td>
</tr>
<tr>
<td>&gt;</td>
<td>Larger/more than</td>
</tr>
<tr>
<td>&lt;</td>
<td>Smaller/less than</td>
</tr>
<tr>
<td>&lt;=</td>
<td>Equal to or smaller/less than</td>
</tr>
<tr>
<td>&gt;=</td>
<td>Equal to or larger/more than</td>
</tr>
<tr>
<td>&lt;&gt;</td>
<td>Larger/more than or smaller/less than (but not equal to)</td>
</tr>
</tbody>
</table>

LIKE Matches a given pattern. If a value matches the pattern, the operator returns True; otherwise, it returns False.

Always use LIKE if the value contains a wildcard. For example, when setting a team rule based on an IP address, set the operator LIKE with a value 10.1.7.*

LIKE ensures that the wildcard is interpreted as such. With other operators, the value is interpreted literally (So, the team rule in the above-mentioned example would never be met, because 10.1.7.* is not a valid IP addresses that can exist in any environment).

NOT LIKE Does not match the pattern. If a value does not match the pattern, the operator returns True; otherwise, it returns False.

Available wildcard characters (combinations are allowed)

<table>
<thead>
<tr>
<th>Character</th>
<th>Stands for</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>Any single character</td>
<td>ba? matches bat, bad, bag, ba1, ba2, ...</td>
</tr>
<tr>
<td>*</td>
<td>Zero or more characters</td>
<td>a* matches andromeda, aga, aa, a888a, a0a, ...</td>
</tr>
<tr>
<td>#</td>
<td>Any single digit (0-9)</td>
<td>a#b matches a1b, a2b, ..., a9b but not aga, aka, ...</td>
</tr>
<tr>
<td>!</td>
<td>Excludes a pattern</td>
<td>a[!1-8] matches aa, ab, a9, ..., but not a1, a2, ...</td>
</tr>
<tr>
<td>[charlist]</td>
<td>Any single character in charlist</td>
<td>a[A-L] matches aa, ab, ..., al but not am, an, ...</td>
</tr>
<tr>
<td>![charlist]</td>
<td>Any single character not in charlist</td>
<td>a![A-L] matches am, an, ..., az but not aa, ab, ... al</td>
</tr>
</tbody>
</table>
Pattern matching

In several windows where wildcards and operators are allowed, you can click Pattern Matching for a window where you can test whether a pattern matches what you thought it would match.

<table>
<thead>
<tr>
<th>Notes</th>
</tr>
</thead>
</table>
| • In the Tasks Apply Registry Settings and Perform File Operations, pattern matching functions somewhat differently. See the Help for those specific Tasks.  
• In the Tasks Unix/Linux and Mac OS X, pattern matching is done using regular expressions.  
• If items in your environment contain special characters in their name ([,*,/, #, etc), you can use these items in RES ONE Automation by enclosing the special characters in square brackets. For example, if the name of an item is "test" and you use this item in a filter, "test[*]" will match the item "test". Please note that if you filter on "test" in RES ONE Automation (so, without the square brackets), the special character "*" will be treated as a wildcard character: "test" will then match all items starting with "test". |
8.9 Trusts

By default, all Agents in a RES ONE Automation environment are allowed to use all Modules and all Resources when they execute Jobs. This behavior may not always be desirable in multi-tenant sites that serve multiple customers. In multi-tenant sites, "ownership" of customer-specific Agents, Modules and Resources often lies with customer representatives. You therefore may want to create a situation in which you let customer representatives determine which Modules and Resources are used by that customer's Agents.

You can tackle these situations by configuring Trust relationships ("Trusts") between these objects. By configuring Trusts, you can create relationships between Agents and Modules and between Agents and Resources. This relationship is determined, on the one hand, by the Agent(s) that are trusted by the Module and/or Resource and, on the other hand, by the Module(s) and/or Resource(s) that are trusted by the Agent. As Trusts are evaluated at the moment of Job execution, it is the combination of these Trusts that determines whether or not an Agent can execute a Job with these Modules and Resources. This is only possible when a full Trust exists (i.e. the Module or Resource is trusted by the Agent and the Agent is trusted by the Module or Resource).

Schematically:

For example:
- If a full Trust exists between an Agent and a Module, the Agent is allowed to execute the Module. If it does not exist, the Module will fail.
- If a Task in a Module uses a specific Resource, a full Trust must also exist between the Agent and the Resource. If it does not exist, the Task that uses the Resource will fail. Depending on the error control settings of the Task, this may fail the entire Module.

Trusts can be configured for Teams and Team folders, Agents, Modules and Module folders and on Resources and Resource folders.

Because you can delegate control over Trusts to the owners of Agents, Modules and Resources, this makes it possible for a customer to determine which of his Agents are authorized to use which of his Modules and Resources.
Scenarios

Trusts are ideal in the following situations:

- When owners of a customer-specific Module or Resource need to specify that it may only be used by that customer’s Teams.
- When owners of a customer-specific Team need to specify that only certain Modules and Resources may be used by that customer’s Teams.
- When only a specific Agent is allowed to use a specific Module with a specific Resource (for example, a specific AutoCad installation or an ISO with specific licensing information).

Notes

Agents that use a Team as their primary Team not only inherit its Trusts, but also its Team settings. It is therefore recommended to consider the consequences for the inherited settings and Trust permissions before assigning an Agent to a different primary Team, as this may have a severe impact on the performance of the Agent.

Best practice: Create a Team folder, configure Trusts on this folder and add the two primary Teams to the Team folder. In this way, it is not necessary to change Team settings or Trusts for each primary Team: the Agent will still inherit the correct settings from its primary Team. See Teams (on page 205).

8.9.1 Where to find Trusts in the Console

<table>
<thead>
<tr>
<th>What</th>
<th>Where</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trusts Security</td>
<td>At Setup &gt; Global Settings.</td>
</tr>
<tr>
<td>Trusts on Agents</td>
<td>At Topology &gt; Agents, when editing the properties of an Agent on the Trusts tab.</td>
</tr>
<tr>
<td>Trusts on Team folders</td>
<td>At Topology &gt; Teams, when editing the properties of a Team folder on the Trusts tab.</td>
</tr>
<tr>
<td>Trusts on Teams</td>
<td>At Topology &gt; Teams, when editing the properties of a Team on the Trusts tab.</td>
</tr>
<tr>
<td>Trusts on Resource folders</td>
<td>At Topology &gt; Resources, when editing the properties of a Resource folder on the Trusts tab.</td>
</tr>
<tr>
<td>Trusts on Resources</td>
<td>At Topology &gt; Resources, when editing the properties of a Resource on the Trusts tab.</td>
</tr>
<tr>
<td>Trusts on Module folders</td>
<td>At Topology &gt; Modules, when editing the properties of a Module folder on the Trusts tab.</td>
</tr>
<tr>
<td>Trusts on Modules</td>
<td>At Topology &gt; Modules, when editing the properties of a Module on the Trusts tab.</td>
</tr>
</tbody>
</table>
8.9.2 Configuring Trusts

• Before Console users can start working with Trusts, you need to specify:
  • The behavior of Trusts in your RES ONE Automation environment. You can do this with the global setting Trusts Security. Best practice is to set Trusts Security to Disabled, configure only first, as this setting does not enable Trusts, but allows Console users that log on with administrative role Full Access to configure Trusts and to assign administrative role permissions to them. Once you have configured all relevant Trusts, you can safely enable Trusts Security. See Global Settings (on page 45).
  • Which Console users are allowed to do this, and on which items they are allowed to configure Trusts. Because "ownership" of customer-specific Agents, Modules and Resources often lies with customer representatives in multi-tenant sites, you may want to delegate control over Trusts to these representatives first, before you enable Trusts in your environment. This prevents situations in which unauthorized personnel can configure Trusts. You can delegate control over Trusts by assigning permissions to Trusts to specific administrative roles. See Trust permissions (on page 217).
  • Once you have specified the behavior of Trusts and have delegated control on Trusts to Console users, Console users can configure Trusts and assign permissions to Trusts, depending on the permissions of their administrative role. Console users can configure Trusts for Agents, Team folders, Teams, Resource folders, Resources, Module folders and Modules by using the Trusts tab when editing these items.
    • When editing a Team folder, Team or Agent, the Trusts tab is divided in the tabs Trusted Modules and Trusted Resources. These tabs allow you to configure Trusts for Module folders, Modules, Resource folders and Resources.
    • When editing a Resource folder, Resource, Module folder or Module, the Trusts tab allows you to configure Trusts for Team folders, Teams and Agents.
  • All items that are shown on the various Trusts tabs are subject to the access permissions of the administrative role(s) of the Console user.

Trusts can be explicit or inherited:
  • A Trust on a Team folder can be explicit or, if applicable, inherited from its parent folder.
  • A Trust on a Team can be explicit or, if applicable, inherited from its Team folder.
  • A Trust on an Agent can be explicit or, if applicable, inherited from its primary Team. Agents that do not have a primary Team will be listed in the root folder of Trusted Teams and Agents (available when configuring Trusts on Resources and Modules and/or their folders).
  • A Trust on a Module folder or a Resource folder can be explicit or, if applicable, inherited from its parent folder.
A Trust on a Module or a Resource can be explicit or, if applicable, inherited from its parent folder.

At Job execution, each time an Agent uses a Module or Resource, all Trusts on the relevant Agents, Modules and Resources in the Job will be evaluated.

- If a full Trust exists between an Agent and a Module, the Agent is allowed to execute the Module. If a full Trust does not exist, the Module will fail due to a breach of Trust.
- If a Task in a Module uses a specific Resource, a full Trust must also exist between the Agent and the Resource. If a full Trust does not exist, the Task that uses the Resource will fail. Depending on the error control settings of the Task, this may fail the entire Module.

**Note**

Availability of the Trusts tab depends on the global setting Trusts Security.

## 8.10 Version Control

With Version Control, it is possible to have better auditability and change control in RES ONE Automation through versioning: Modules, Projects, Run Books and Resources are stored with a version number. This allows you to track these objects as they are changed.

- Depending on the needs of your organization, it is possible to store versions related to revisions, minor changes and major changes to Modules, Projects, Run Books and Resources. The versioning of these objects can be configured in the Version Control window, which opens when saving changes to a Module, Project, Run Book or Resource. Availability of this window depends on the global setting Use automatic versioning (see page 48).
- In the Audit Trail (on page 224), you can find versioning for Modules, Projects, Run Books and Resources on the Versioning tab of these objects.
- Versioning is also reflected in Building Blocks (on page 225).

### Configuration of the Version Control window

- The **Current version** field shows the current version number of the object. A version number consists of a major, a minor and a revision version number (for example 3.2.1).
- Various versioning options are available to specify how the version number should be updated:
  - Increase revision number
  - Increase minor version
  - Increase major version
  - Start with new version (by default selected)
- The **New version** field makes it possible to preview the new version number before it is actually applied to the object.
- Use the **Comment** field to specify a comment that applies to the new version of the object. The behavior of the **Comment** field can be managed with the global setting Version comment (see page 48).
Behavior

- When creating a new Module, Project, Run Book or Resource, only the option **Start with new version** will be available. The new version will be set to “1.0.0” and the comment to “Initial version”. The comment can still be edited.

- When editing an object, you can select which version update to apply:
  - Increase revision number
  - Increase minor version
  - Increase major version

- When duplicating an object, all four versioning options can be selected. By default, **Start with new version** will be selected.

- When creating new objects from the selection/search screen in another object, the **Version Control** window will not be shown. Instead, the version of the new object will be set to “1.0.0” and the comment to “Initial version”.

- **Building Blocks of Modules, Projects, Run Books and Resources** store the version number and version comment. When importing Building Block items in an environment:
  - The version in the Building Block will be used as the new version, irrespective of the version in the target environment: Existing Modules, Projects, Run Books and Resources will be overwritten with the version in the Building Block.
  - When importing Building Blocks without versioning information, the version number of the object will be set to “1.0.0” and the comment to “Imported via Building Block”.

---

**Note**

Saving a Module, Project, Run Book or Resource, this will be considered to be a new version, regardless of whether any actual changes have been made.
Best Practices (on page 274)

Command-line options (on page 287)

Registry settings (on page 305)

Logging (on page 309)

Flowcharts (on page 313)

RES ONE Automation Glossary (on page 314)

Compatibility Matrix (on page 316)

9.1 Best Practices

9.1.1 Bare Metal OS Deployment Using WDS and RES ONE Automation

When deploying new workstations or servers, Bare Metal OS deployment usually is the first task that is performed. This task can easily be carried out by combining Microsoft Windows Deployment Services (WDS) and RES ONE Automation.

Microsoft Windows 2008 R2 has built-in capabilities of PXE, TFTP imaging and prestaging that can be automated using RES ONE Automation. This is a highly cost-effective solution that does not need any manual intervention. Workstations and servers can be deployed directly where they need to be used, without a staging process and additional transportation.

Desired behavior

Ideally, zero-touch OS deployment using WDS and RES ONE Automation occurs according to the following scenario:

1. A new computer that is to be deployed is prestaged (enrolled), in order to determine its configuration. Prestaged computers are also called "known" computers.
2. After being prestaged, the computer is booted and the PXE server will pick up the request. The end user must not be involved in this process by having to press F12 or any other buttons that interfere with the boot process.
3. Preferably the NIC is the first in the boot-sequence.
4. WDS is configured so the computer is assigned a predefined name and joined to a domain.
5. After OS deployment, the RES ONE Automation Agent is configured either to join a RES ONE Automation Team or to execute a predefined RES ONE Automation Module that cleans up the installation or further configures the computer.
6. After deployment, the PXE configuration should be set in such a way that the end user is not tempted to press F12. Unknown computers must not be served by PXE to avoid unintentional imaging of those computers.
WDS Configuration

WDS interacts closely with DHCP, PXE, DNS and Active Directory. As all of these functions are part of the Windows Server platform, it is cost-effective to use them. DHCP can also be offered by a non-Windows server.


Active Directory configuration

1. Configure WDS to respond only to "known" computers. To make a new computer known to WDS, it must be added to Active Directory.

2. Configure the Active Directory attribute "netbootMachineFilePath" to determine the response of PXE to the request. This attribute can be edited with ADSI Edit. On servers running Microsoft Windows Server 2008 R2, ADSI Edit is installed when you install the Active Directory Domain Services (AD DS) role to make the server a domain controller. For more information about ADSI Edit, see [http://technet.microsoft.com/en-us/library/cc773354%28WS.10%29.aspx#BKMK_InstallingADSIEdit](http://technet.microsoft.com/en-us/library/cc773354%28WS.10%29.aspx#BKMK_InstallingADSIEdit).

Some values that you can use are:

- \boot\x64\PXEBoot.com (requires users to press F12 for PXE boot)
- \boot\x64\PXEBoot.n12 (no user interaction required)
- \boot\x64\abortPXE.com (boots next device, probably hard disk)

Use corresponding values for 32-bit computers.
WDS configuration
1. Make sure DNS, AD and DHCP are working correctly (not necessarily on the same server).
2. On the Start menu, click Administrative Tools > Server Manager.
3. Click Add Roles.
5. In the Server Manager, expand the server list.
6. Right-click the server that you want to manage, and click Configure Server. This starts the Windows Deployment Services Configuration Wizard.

DHCP option 60
If DHCP and PXE (part of WDS) are running on the same server, both processes must receive the DHCP request packet from the clients.
1. Open the properties of the WDS server.
2. On the DHCP tab, select Configure DHCP option 60 to indicate that this server is also a PXE Server. This redirects PXE requests to port 4011. Do NOT select this option if PXE and DHCP are running on separate servers.

PXE server initial settings
1. On the PXE Response tab, select either Respond only to known client computers or Respond to all client computers, depending on what you want to achieve. In a tightly managed environment, we recommend to select Respond only to known client computers. RES ONE Automation will take care of the rest.
2. Do NOT select Require administrative approval.

Images
WDS makes use of images in 2 stages: a pre-boot Operating System and an installed Operating System.
1. First add a pre-boot OS. On the Start menu, click Administrative Tools > Server Manager > Windows Deployment Services.
2. In the WDS administration tool, browse to the server that you are configuring, right-click the folder Boot Images and click Add Boot Image.
3. Browse to the Sources folder on the Windows 2008R2 installation DVD and select boot.wim. You can add multiple boot Operating Systems for different platforms (x86 and x64).
4. After adding the pre-boot OS, add the OS that is intended to be installed on the target computers: in the WDS administration tool, browse to the server that you are configuring, right-click the folder Install Images and click Add Install Image.
5. When prompted, browse to the Sources folder on the Windows 2008R2 installation DVD and select install.wim. You can also find this file on the Windows 7 installation DVD.
7. Add the RES ONE Automation Agent manually to the Windows 7 or Windows 2008 image using the DISM command.
8. Mount the image, add a c:\temp folder and copy the MSI file into this folder.
9. Unmount the image and do not forget to use the /commit option.
Boot menu configuration
1. In the WDS administration tool, right-click the server that you are configuring and click Properties.
2. Click the Boot tab.
3. Select Always continue the PXE boot. This allows the boot procedure to be independent from user interaction. If you allow for user interaction (by pressing F12), the success of the boot procedure depends on the user reading the instructions on the screen and acting accordingly. Because this rarely happens, it is better to bypass user interaction.

Boot loop
We assume the NIC to be the first in priority in the boot order. This can cause a boot loop: WDS deploys the image and the setup starts. During setup, the machine reboots and everything starts from the beginning. To avoid this situation, WDS can change the boot program during deployment if the following registry key is set:

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\WDSServer\Providers\W DSPXE\Providers\BINLSVC]
"ResetBootProgram"=dword:00000001

Unattend.xml
In the process, we need two unattend.txt files: one that is used in WinPE phase (WinPE.xml) and one that is associated with the image and is needed after WinPE phase (PostPE.xml).
1. Right-click the imported images and click Properties.
2. Select Allow to install in unattended mode, select the "PostPE.xml" file and click OK.
3. Leave the WinPE.xml in the WdsClientUnattend folder in the RemoteInstall install folder.

RES ONE Automation configuration
Prestage computers
When prestaging computers, the objective is to pre-populate Active Directory with details on the computer that is to be deployed. A significant detail is the identifier that must be used to differentiate between computers. This identifier can be: hardware GUID or MAC address.
1. In RES ONE Automation, create a Module with a Task Execute Command with the following command line:

   C:\windows\system32\wdsutil.exe /add-device /device:$[Workstation name] /id:$[MAC Address] /wdsclientunattend:wdsclientunattend\WinPE.xml /bootprogram:boot\x64\pxeboot.n12 /referralserver:<FQDN of your WDS server>

2. Create 2 parameters for this Module:
   - A text parameter $[Workstation name]. On the Input tab, select Parameter value is required (may not be empty).
   - A text parameter $[MAC Address] with input mask HH-HH-HH-HH-HH-HH; On the Input tab, select Parameter value is required (may not be empty).
Install the RES ONE Automation Agent

The RES ONE Automation Agent is installed in the last phase of the setup.

1. Choose one of the following:
   - Create an MSI file that puts the Agent in a preconfigured Team.
   - Install the MSI file with an option ADDTOTEAM to make sure the Agent joins a specific Team.

2. Configure the Team in such a way that a new member of the Team will execute a Module with one parameter $[NetBIOSName]$ and a Task Delete Local user. In the Logon name field, specify tempaccount. This line removes the unavoidable useraccount that you need to configure during setup. You may change the name, but then make sure you change the name in the PostPE.xml file as well.

Auto-generate computer names

In real life, people tend not to remember the last issued computer name. And they don’t need to. This can easily be automated using the following Task which can be combined with the pre-staging Task Execute Command.

1. First, create a string type registry key somewhere in the HKLM\SOFTWARE tree on the WDS server. For example: HKLM\SOFTWARE\Maverick with keyname Counter and value xxxx, where xxxx is the last issued computer name number. So if PC-2345 is your last deployed workstation, the value of Counter is 2345.

2. Create a Task Apply Registry Settings.

3. Set the value of the HKLM\SOFTWARE\Maverick\Counter to
   @\[CALC(@[REGISTRY(HKEY_LOCAL_MACHINE\SOFTWARE\Maverick\Counter)],+,1)]

4. Execute this Task on the WDS server before the pre-staging Task, to use this counter to construct the next available computer name. For example: PC-@\[REGISTRY(HKEY_LOCAL_MACHINE\SOFTWARE\Maverick\Counter)].

Re-deploying an existing machine

When the operating system on an existing machine is no longer working, you can re-deploy the machine if it has been configured in Active Directory and has been deployed with WDS in the past.

1. In RES ONE Automation, create a Module with a parameter $[Computername]$ and a Task Execute Command with the following command line:
   C:\windows\system32\wdsutil.exe /set-device /device:$[NetBIOSName] /bootprogram:boot\x64\pxeboot.n12

2. Execute this Task on the WDS server and use the target machine name as parameter. We assume the Operating System on the target machine is no longer responding, so ask the end user to restart the workstation.

3. In other scenarios you may want to create a Run Book that executes the above Task on the WDS server and a Task Reboot on the target machine.

Note

This Module does not lead to any result if the machine has not been configured in Active Directory and has not been deployed with WDS in the past.
The Unattend files

WDS has split the unattended configuration over two files:

- **WinPE.xml**: This file is only used in the Windows PE phase of the setup and determines the connection to the WDS server, the image to be used and the disk configuration. You may change the disk partitions in this file. This file uses credentials to connect to the WDS server. These credentials will not be traceable on the target system, because this XML file is destroyed during the second phase of the installation.

- **PostPE.xml**: This file is used after the reboot of the Windows PE phase. It contains a password of the local administrator account. Please note that this line will automatically be removed, so no worries about security.

The machine name is a parameter `%MACHINENAME%`, which will automatically be replaced by WDS. WDS also takes care of the credentials to add the machine to the domain. The `RunSynchronous` section of the "Specialize" phase contains the installation of the RES ONE Automation client. The MSI file is assumed to be in a `C:\temp` folder. It is therefore important to place the MSI file in this folder before deploying the image (use DISM to modify the image).

**WinPE.xml for Windows 7**

```xml
<?xml version="1.0" encoding="utf-8"?>

<unattend xmlns="urn:schemas-microsoft-com:unattend">
  <settings pass="windowsPE">
    <component name="Microsoft-Windows-Setup" processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35" language="neutral" versionScope="nonSxS" xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State">
      <WindowsDeploymentServices>
        <Login>
          <Credentials>
            <Domain>[Here your domain name]</Domain>
            <Password>[ Here the password]</Password>
            <Username>[Here a username]</Username>
          </Credentials>
        </Login>
        <ImageSelection>
          <WillShowUI>OnError</WillShowUI>
          <InstallImage>
            <ImageName>Windows 7 ENTERPRISE</ImageName>
            <ImageGroup>Workstations</ImageGroup>
            <Filename>Install.wim</Filename>
          </InstallImage>
        </ImageSelection>
      </WindowsDeploymentServices>
    </component>
  </settings>
</unattend>
```
<InstallTo>
  <DiskID>0</DiskID>
  <PartitionID>1</PartitionID>
</InstallTo>
</ImageSelection>
</WindowsDeploymentServices>
<DiskConfiguration>
  <WillShowUI>OnError</WillShowUI>
  <Disk>
    <DiskID>0</DiskID>
    <WillWipeDisk>true</WillWipeDisk>
    <CreatePartitions>
      <CreatePartition>
        <Order>1</Order>
        <Type>Primary</Type>
        <Extend>true</Extend>
      </CreatePartition>
    </CreatePartitions>
    <ModifyPartitions>
      <ModifyPartition>
        <Order>1</Order>
        <PartitionID>1</PartitionID>
        <Format>NTFS</Format>
        <Label>Windows</Label>
      </ModifyPartition>
    </ModifyPartitions>
  </Disk>
</DiskConfiguration>
<UserData>
  <AcceptEula>true</AcceptEula>
<ProductKey>
  <Key></Key>
</ProductKey>

</UserData>

</component>

<component name="Microsoft-Windows-International-Core-WinPE"
processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35"
language="neutral" versionScope="nonSxS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SetupUILanguage>
    <UILanguage>en-US</UILanguage>
  </SetupUILanguage>

  <InputLocale>0409:00000409</InputLocale>
  <SystemLocale>en-US</SystemLocale>
  <UILanguage>en-US</UILanguage>
  <UserLocale>en-US</UserLocale>
</component>

</settings>
</unattend>

WinPEsrv.xml for Windows 2008

<?xml version="1.0" encoding="utf-8"?>

<unattend xmlns="urn:schemas-microsoft-com:unattend">
  <settings pass="windowsPE">
    <component name="Microsoft-Windows-Setup"
processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35"
language="neutral" versionScope="nonSxS"
      <WindowsDeploymentServices>
        <Login>
          <Credentials>
            <Domain>[Here your domain name]</Domain>
            <Password>[Here the password]</Password>
            <Username>[Here a username]</Username>
          </Credentials>
        </Login>
      </WindowsDeploymentServices>
    </component>
  </settings>
</unattend>
<Credentials>
</Credentials>

</Login>

<ImageSelection>

<WillShowUI>OnError</WillShowUI>

<InstallImage>

<ImageName>Windows Server 2008 R2 SERVERSTANDARD</ImageName>

<ImageGroup>Servers</ImageGroup>

<Filename>Install.wim</Filename>

</InstallImage>

<InstallTo>

<DiskID>0</DiskID>

<PartitionID>1</PartitionID>

</InstallTo>

</ImageSelection>

</WindowsDeploymentServices>

<DiskConfiguration>

<WillShowUI>OnError</WillShowUI>

<Disk>

<DiskID>0</DiskID>

<WillWipeDisk>true</WillWipeDisk>

>CreatePartitions>

>CreatePartition>

<Order>1</Order>

>Type>Primary</Type>

<Extend>true</Extend>

</CreatePartition>

</CreatePartitions>

</ModifyPartitions>

</DiskConfiguration>
<PartitionID>1</PartitionID>
<Format>NTFS</Format>
<Label>Windows</Label>
</ModifyPartition>
</ModifyPartitions>
</Disk>
</DiskConfiguration>
<UserData>
<AcceptEula>true</AcceptEula>
<ProductKey>
<Key></Key>
</ProductKey>
</UserData>
</component>
<component name="Microsoft-Windows-International-Core-WinPE"
processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35"
language="neutral" versionScope="nonSxS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SetupUILanguage>
<UILanguage>en-US</UILanguage>
</SetupUILanguage>
<InputLocale>0409:00000409</InputLocale>
<SystemLocale>en-US</SystemLocale>
<UILanguage>en-US</UILanguage>
<UserLocale>en-US</UserLocale>
</component>
</settings>
</unattend>

PostPE.xml

<?xml version="1.0" encoding="utf-8"?>
<unattend xmlns="urn:schemas-microsoft-com:unattend"
<settings pass="specialize">
  
    <ComputerName>%MACHINENAME%</ComputerName>
    <TimeZone>W. Europe Standard Time</TimeZone>
  </component>

    <Identification>
      <UnsecureJoin>TRUE</UnsecureJoin>
      <JoinDomain>[Here your domain]l</JoinDomain>
    </Identification>
  </component>

    <RunSynchronous>
      <RunSynchronousCommand wcm:action="add">
        <Description>EnableAdmin</Description>
        <Order>1</Order>
        <Path>cmd /c net user Administrator /active:yes</Path>
      </RunSynchronousCommand>
      <RunSynchronousCommand wcm:action="add">
        <Description>Install RES-AM</Description>
        <Order>2</Order>
        <Path>msiexec /i "c:\temp\RES-AM-Agent-6.0.1.1.msi" /qn</Path>
      </RunSynchronousCommand>
    </RunSynchronous>
  </component>
</settings>
Chapter 9: Appendices

Copyright © RES Software Development B.V. All rights reserved.
Chapter 9: Appendices

</LocalAccounts>
</UserAccounts>

<OOBE>

<HideEULAPage>true</HideEULAPage>

<NetworkLocation>Work</NetworkLocation>

<ProtectYourPC>1</ProtectYourPC>

<SkipUserOOBE>true</SkipUserOOBE>

</OOBE>
</component>


<InputLocale>0409:00000409</InputLocale>

<SystemLocale>en-US</SystemLocale>

<UILanguage>en-US</UILanguage>

<UserLocale>en-US</UserLocale>

</component>

</settings>
</unattend>
9.2 Command-line options

You can perform the following Console actions from a command line:

- Adding or editing a Resource (on page 289)
- Changing Agent connection settings (on page 291)
- Changing Dispatcher connection settings (on page 291)
- Creating a Building Block for a Run Book (on page 291)
- Exporting Job Results (on page 293)
- Exporting licenses (on page 293)
- Importing Building Blocks (on page 294)
- Installing Agents (on page 295)
- Installing Consoles
- Installing Dispatchers
- Installing RES ONE Automation
- Scheduling Jobs (on page 302)
- Solving a Console lockout (on page 304)

Return Code

When you execute an action in the Console from the command line, you get a return code reflecting the status.

The execution of actions from the command line include:

- Job Scheduling
- Importing Building Blocks
- Creating Building Blocks for a Run Book
- Exporting Job Results
- Adding or Editing a Resource

The status return is given when started from a Microsoft Windows Batch/PowerShell script and when started from a RES ONE Automation Task Command Execute.
Overview of return codes:

<table>
<thead>
<tr>
<th>Code</th>
<th>Explanation</th>
<th>Applicable action</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Success</td>
<td>general</td>
</tr>
<tr>
<td>1</td>
<td>Failed</td>
<td>general</td>
</tr>
<tr>
<td>2</td>
<td>Aborted</td>
<td>general</td>
</tr>
<tr>
<td>3</td>
<td>Canceled</td>
<td>general</td>
</tr>
<tr>
<td>4</td>
<td>Completed with errors</td>
<td>general</td>
</tr>
<tr>
<td>5</td>
<td>Failed to login. This code will be returned if the global Password Security Policy (see page 48) has not been met.</td>
<td>general</td>
</tr>
<tr>
<td>6</td>
<td>Source not found or entered.</td>
<td>newresource / editresource</td>
</tr>
<tr>
<td>7</td>
<td>Folder not found or incorrect value.</td>
<td>newresource</td>
</tr>
<tr>
<td>8</td>
<td>Target type not correct.</td>
<td>newresource / editresource</td>
</tr>
<tr>
<td>9</td>
<td>The security role entered for &quot;/user&quot; has no modify access to the resources node.</td>
<td>newresource / editresource</td>
</tr>
<tr>
<td>10</td>
<td>The special parameter value was not recognized.</td>
<td>newresource</td>
</tr>
<tr>
<td>11</td>
<td>Edit Resource Failed: when saving changes to the Datastore.</td>
<td>editresource</td>
</tr>
<tr>
<td>12</td>
<td>New Resource Failed: when saving the new resource to the Datastore.</td>
<td>newresource</td>
</tr>
<tr>
<td>13</td>
<td>The GUID for the target resource to edit is invalid or not found.</td>
<td>editresource</td>
</tr>
<tr>
<td>14</td>
<td>Object type does not exist.</td>
<td>createbb</td>
</tr>
<tr>
<td>15</td>
<td>Object GUID does not exist or is invalid.</td>
<td>createbb</td>
</tr>
<tr>
<td>16</td>
<td>Invalid destination file.</td>
<td>createbb</td>
</tr>
</tbody>
</table>
9.2.1 Adding or editing a Resource

You can add a new Resource or edit an existing Resource from a command line. For example, this allows you to create an automated (daily) process to add software updates as a Resource in RES ONE Automation and to create a report on the progress. This functionality is currently only available for Resources stored in the Datastore.

New Resource via command line

You can add a new Resource from the command line using the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>/action</td>
<td>newresource</td>
<td>Specifies the action, in this case Add New Resource.</td>
</tr>
<tr>
<td>/targettype</td>
<td>&lt;type&gt;</td>
<td>Identifies the type of location where the Resource is added. Default and only</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• datastore</td>
</tr>
<tr>
<td>/source</td>
<td>&lt;path\filename&gt;</td>
<td>Identifies the path and filename of the source. Environment variables are supported. Wildcard characters are not supported.</td>
</tr>
<tr>
<td>/folderGUID</td>
<td>&lt;GUID&gt;</td>
<td>Optional: specifies the GUID of the folder where the Resource is added.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Specify resources or root to add the Resource to the root node Resources.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If the folder GUID is not a Resource folder or left empty, the action will fail.</td>
</tr>
<tr>
<td>/user</td>
<td>&lt;username&gt;</td>
<td>Optional: specifies the RES ONE Automation credentials to access the Console.</td>
</tr>
<tr>
<td>- and -</td>
<td>- and -</td>
<td>- and -</td>
</tr>
<tr>
<td>/password</td>
<td>&lt;password&gt;</td>
<td>If not given, the Microsoft Windows Authentication credentials will be used to determine if the user has access to the Console and sufficient access to perform the action.</td>
</tr>
<tr>
<td>/description</td>
<td>&lt;description&gt;</td>
<td>Optional: adds a description text.</td>
</tr>
<tr>
<td>/special</td>
<td>&lt;special&gt;</td>
<td>Optional: enables or disables the special actions. The values are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• none: no parsing is used. Default value.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• parse: Parse variables, parameters and functions in contents of this file</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• parse=noenv: Skip parsing of environment variables</td>
</tr>
<tr>
<td>/silent</td>
<td></td>
<td>Optional: no user interaction, such as message boxes, is shown when running this action.</td>
</tr>
</tbody>
</table>
Example

"C:\program files (x86)\res software\automation manager\wmc\wmc.exe"
/action=newresource /targettype=datastore /source="c:\Example\123.txt"
/folderguid={12A34B56-7890-1234-5678-1234C5D678EF} /description="New resource added via commandline" /user=jsmith /password=secret
/special=parsenoenv

Warning
When adding a Resource via the command line, it will inherit the trust data from the folder it is placed in.

Edit Resource via command line

You can edit an existing Resource from the command line using the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>/action</td>
<td>editresource</td>
<td>Specifies the action, in this case Edit Resource.</td>
</tr>
<tr>
<td>/resource</td>
<td>&lt;GUID&gt;</td>
<td>Identifies the GUID of the Resource that needs to be edited.</td>
</tr>
<tr>
<td>/targettype</td>
<td>&lt;type&gt;</td>
<td>Identifies the type of Resource that needs to be edited. Default and only value is:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• datastore</td>
</tr>
<tr>
<td>/source</td>
<td>&lt;path\filename&gt;</td>
<td>Identifies the path and filename of the source. Environment variables are supported. Wildcard characters are not supported.</td>
</tr>
<tr>
<td>/user</td>
<td>&lt;username&gt;</td>
<td>Optional: the RES ONE Automation credentials to access the Console. If not given, the Microsoft Windows Authentication credentials will be used to determine if the user has access to the Console and sufficient access to perform the action.</td>
</tr>
<tr>
<td>- and -</td>
<td>- and -</td>
<td></td>
</tr>
<tr>
<td>/password</td>
<td>&lt;password&gt;</td>
<td></td>
</tr>
<tr>
<td>/description</td>
<td>&lt;description&gt;</td>
<td>Optional: edit the descriptive text. Please notice that the current description will be replaced with the new value, also when this value is empty &quot;&quot;.</td>
</tr>
<tr>
<td>/silent</td>
<td></td>
<td>Optional: no user interaction, such as message boxes, is shown when running this action.</td>
</tr>
</tbody>
</table>

Example

"C:\program files (x86)\res software\automation manager\wmc\wmc.exe"
/action=editresource /resource={12345678-AAAA-9988-7766-123456789AAA} /targettype=datastore /source="c:\Example\123.txt" /description="Edit-update via commandline" /user=jsmith /password=secret

Warning
When editing a Resource via the command line, its attributes and trust data will stay unchanged.
### Return codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Success</td>
</tr>
<tr>
<td>1</td>
<td>Failed</td>
</tr>
<tr>
<td>2</td>
<td>Aborted</td>
</tr>
<tr>
<td>5</td>
<td>Failed to login.</td>
</tr>
<tr>
<td>6</td>
<td>Source file not found -or- Invalid Source file.</td>
</tr>
<tr>
<td>7</td>
<td>Invalid Folder GUID.</td>
</tr>
<tr>
<td>8</td>
<td>Invalid Target type.</td>
</tr>
<tr>
<td>9</td>
<td>Insufficient access. The security role entered for &quot;/user&quot; has no modify access to the resources node.</td>
</tr>
<tr>
<td>10</td>
<td>Invalid value for parameter &quot;special&quot;.</td>
</tr>
<tr>
<td>11</td>
<td>Edit Resource Failed: when saving changes to the Datastore.</td>
</tr>
<tr>
<td>12</td>
<td>New Resource Failed: when saving the new resource to the Datastore.</td>
</tr>
<tr>
<td>13</td>
<td>Invalid Resource GUID.</td>
</tr>
</tbody>
</table>

#### 9.2.2 Changing Agent connection settings

To change the RES ONE Automation environment that an Agents uses or to change its Dispatcher detection settings, an Agent can be started interactively with an interface.

To do so, use one of the following command lines:

- For Agent+: "\%PROGRAMFILES\%\RES\Automation\Agent\agent.exe" /interactive
- For Agent on 64-bit machine: "\%PROGRAMFILES\(x86\)%\RES Software\Automation Manager\Agent\agent.exe" /interactive
- For Agent on 32-bit machine: "%PROGRAMFILES\%\RES Software\Automation Manager\Agent\agent.exe" /interactive

**Note**
The configured settings for the Agent will be overwritten again if the Dispatcher detection settings in the Console are changed.

#### 9.2.3 Changing Dispatcher connection settings

To connect the Dispatcher to a different Datastore, the Dispatcher can be started interactively with an interface.

To do so, use one of the following command lines:

"\%PROGRAMFILES\%\RES\Automation\Dispatcher\Dispatcher.exe" /configdb
9.2.4 Creating a Building Block for a Run Book

You can create a Building Block for a Run Book from a command line. This allows you to create an automated process to create a Building Block of a single Run Book and create an entire environment based upon Run Books.

To do so, run the executable of the Console (\wmc.exe) from the command line and use the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>/action</td>
<td>createbb</td>
<td>Specifies the action, in this case Create a Building Block.</td>
</tr>
<tr>
<td>/objecttype</td>
<td>runbook</td>
<td>Identifies the type of the Building Block. The only value is: runbook.</td>
</tr>
<tr>
<td>/objectguid</td>
<td>&lt;GUID&gt;</td>
<td>Specifies the GUID of the objecttype, in this case a Run Book GUID.</td>
</tr>
<tr>
<td>/destination</td>
<td>&lt;path\filename.extension&gt;</td>
<td>Identifies the path where the Building Block needs to be created and the filename (including the extension like .xml) of the created Building Block.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Please note the following:</td>
</tr>
<tr>
<td></td>
<td>•</td>
<td>The destination path must be accessible for saving a file.</td>
</tr>
<tr>
<td></td>
<td>•</td>
<td>Wildcard characters are not supported.</td>
</tr>
<tr>
<td>/skipresources</td>
<td></td>
<td>Optional: use this option to skip the Resources when creating a Building Block. By default, the available Resources of a Run Book are included in the Building Block.</td>
</tr>
<tr>
<td>/user</td>
<td>&lt;username&gt;</td>
<td>Optional: specifies the RES ONE Automation credentials to access the Console.</td>
</tr>
<tr>
<td>-and-</td>
<td>&lt;password&gt;</td>
<td>If not given, the Microsoft Windows Authentication credentials will be used to determine if the user has access to the Console and sufficient access to perform the action.</td>
</tr>
<tr>
<td>/silent</td>
<td></td>
<td>Optional: no user interaction, such as message boxes, is shown when running this action.</td>
</tr>
</tbody>
</table>

Example

"C:\program files (x86)\res software\automation manager\wmc\wmc.exe"
/action=createbb /objecttype=runbook /objectguid={1234ABCD-1234-ABCD-1234-12345678ABCD} /destination="c:\temp\example.xml" /user=jsmith /password=secret

Notes

- A Resource file will be saved as XBB file if applicable. Please note that if you use .xbb as an extension in the parameter /destination, the XML file of the Run Book will overwrite the Resource file.
- When creating a Building Block via the command line in an AES-256 encrypted environment (in RES ONE Automation 10.2 or higher), a Building Block passkey will be generated automatically and stored as a separate TXT file with the same name as the XBB file, at the same location.
9.2.5 Exporting Job Results

To export the Job Results using a command line, run the wmc.exe from the command line using the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>-export2xml</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/MasterJobGUID</td>
<td>&lt;GUID&gt;</td>
<td>Export overview results</td>
</tr>
<tr>
<td></td>
<td>or</td>
<td></td>
</tr>
<tr>
<td></td>
<td>@ [MASTERJOBGUID]</td>
<td></td>
</tr>
<tr>
<td>/OverViewResult</td>
<td></td>
<td>Export overview results</td>
</tr>
<tr>
<td>/DetailedResult</td>
<td></td>
<td>Export detailed results</td>
</tr>
<tr>
<td>/ResultsPerAgent</td>
<td></td>
<td>Export results per Agent (existing files will be</td>
</tr>
<tr>
<td></td>
<td></td>
<td>overwritten)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Only valid in combination with /DetailedResult</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Output file names generated:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Export_&lt;Agent&gt;.xml</td>
</tr>
<tr>
<td>/Destination=</td>
<td>&lt;path&gt;</td>
<td>Save location</td>
</tr>
<tr>
<td>/WaitForJob</td>
<td></td>
<td>Wait for job to finish</td>
</tr>
<tr>
<td>/user=</td>
<td>&lt;user&gt;</td>
<td>The RES ONE Automation credentials to access the</td>
</tr>
<tr>
<td>-and-</td>
<td></td>
<td>Console.</td>
</tr>
<tr>
<td>/password=</td>
<td>&lt;password&gt;</td>
<td></td>
</tr>
</tbody>
</table>

Example

"C:\Program Files\RES Software\Automation Manager\WMC\WMC.exe" -export2xml /MasterJobGUID=@ [MASTERJOBGUID] /OverviewResult /DetailedResult /WaitForJob /ResultsPerAgent /Destination=C:\Users\jsmith\Documents\Export.xml /user=jsmith /password=secret

Notes

- When exporting the results of a Job, the Variable are included in the XML files for the Agents and/or Teams.
- It is not possible to export the results of multiple Jobs.
- You can only execute a command line that exports Job results on Agents on which a Console has been installed.
- The function @ [MASTERJOBGUID] cannot yet be used for Run Books.
- If the global setting Limit Job Export has been set to Enabled, the export of Jobs is only allowed for Agents to which the administrative role of Console user grants access.
9.2.6  **Exporting licenses**

To export licenses via the command line, use the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>/action</td>
<td>exportlic</td>
<td>Specifies the action, in this case Export the licenses.</td>
</tr>
<tr>
<td>/file</td>
<td>&lt;filename.extension&gt;</td>
<td>Identifies the filename of the instant report of the licenses that will be created.</td>
</tr>
<tr>
<td>/user</td>
<td>&lt;username&gt;</td>
<td>Optional: specifies the RES ONE Automation credentials to access the Console. If not given, the Microsoft Windows Authentication credentials will be used to determine if the user has access to the Console and sufficient access to perform the action.</td>
</tr>
<tr>
<td>-and-</td>
<td>&lt;password&gt;</td>
<td></td>
</tr>
<tr>
<td>/password</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/silent</td>
<td></td>
<td>Optional: no user interaction, such as message boxes, is shown when running this action.</td>
</tr>
</tbody>
</table>

**Example**

```
"C:\program files (x86)\res software\automation manager\wmc\wmc.exe"
/action=exportlic /file="filename.txt" /user=jsmith /password=secret /silent
```
9.2.7 Importing Building Blocks

To import a Building Block using a command line, run the \texttt{wmc.exe} from the command line using the parameters \texttt{/action=importbb} and \texttt{/file=<path\filename>}. If RES ONE Automation authentication is used, also provide the parameters \texttt{/user} and \texttt{/password}. The credentials must have Read/Modify permissions on the Library and the Topology nodes. Optionally, use \texttt{/silent} to suppresses any messages.

The option to create import folders for the Building Block items during the import process is not supported for unattended Building Block imports.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| \texttt{/action=importbb} \texttt{/file=<path\filename>} | Imports a single Building Block into the Console. For example:  
"C:\Program Files\RES Software\Automation Manager\WMC\WMC.exe" /action=importbb 
/file=c:\buildingblocks\basic.xml |
| \texttt{/user=<username>} \texttt{/password=<password>} | Allows you to specify credentials for the Console. The specified credentials must have Read/Modify permissions on the Library. For example:  
"C:\Program Files\RES Software\Automation Manager\WMC\WMC.exe" /action=importbb 
/file=c:\buildingblocks\basic.xml /user=jsmith 
/password=secret |
| \texttt{/BBpassword} | Specifies the Building Block password for AES-256 encryption |
| \texttt{/silent} | Suppresses any RES ONE Automation messages during import of Building Blocks. For example:  
"C:\Program Files\RES Software\Automation Manager\WMC\WMC.exe" /action=importbb 
/file=c:\buildingblocks\basic.xml /user=jsmith 
/password=secret /silent |

Example

"c:\program files\res software\automation manager\wmc\wmc.exe"  
/action=importbb /file=c:\buildingblocks\basic.xml /user=jsmith 
/password=secret /silent"
9.2.8 Installing Agents

Deploy an Agent unattended if, for example, you want to deploy it on many computers at once using a command line. When you deploy an Agent unattended, it is not necessary to preconfigure it. Unattended deployments can be scripted, for RIS or BareMetal installations.

1. Open the Console and go to Administration > Components.
2. Right-click the relevant installation file and choose Save component as:
   - RES-ONE-Automation-Agent-x.x.x.x.msi for Agents
   - RES-ONE-Automation-Agent+(x64)-x-x-x-x.msi for Agents+ on 64-bit Microsoft Windows versions
   - RES-ONE-Automation-Agent+(x86)-x-x-x-x.msi for Agents+ on 32-bit Microsoft Windows versions
3. Do not preconfigure the MSI file.
4. Save the MSI file to the desired location.
5. At the desired location, apply the following public properties to the MSI file in a command line:

<table>
<thead>
<tr>
<th>Property</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>SITELICENSE</td>
<td>Specifies the Site ID (case-sensitive). You can find this at Setup &gt; Licensing in the Console.</td>
</tr>
<tr>
<td>DISPATCHERLIST</td>
<td>This property is optional. By default, the Agent will autodetect Dispatchers. To use a fixed list of Dispatchers instead, use this property to specify the names or GUIDs of Dispatchers to use. Separate multiple entries with a semi-colon (;).</td>
</tr>
<tr>
<td>DISPATCHERAUTODETECTFIRST</td>
<td>If you are using the property DISPATCHERLIST (see above), set the property DISPATCHERAUTODETECTFIRST with the value:</td>
</tr>
<tr>
<td></td>
<td>- '1' if the Agent should try autodetecting Dispatchers before using the Dispatcher list.</td>
</tr>
<tr>
<td></td>
<td>- '0' if the Agent should only use the Dispatcher list.</td>
</tr>
<tr>
<td>DISPATCHERGETLIST</td>
<td>Specifies whether the Agent should extend its list of Dispatchers by getting lists of Dispatchers from the discovered Dispatcher. Set the value:</td>
</tr>
<tr>
<td></td>
<td>- '1' if the Agent should try to retrieve more Dispatchers through the discovered Dispatcher.</td>
</tr>
<tr>
<td></td>
<td>- '0' if the Agent should not retrieve more Dispatchers through the discovered Dispatcher.</td>
</tr>
<tr>
<td>ADDTOTEAM</td>
<td>This property is optional. To add the Agent as a member of one or more Teams, use this property to specify the names or GUIDs of the Teams. Separate multiple entries with a semi-colon (;).</td>
</tr>
<tr>
<td>INVOKEPROJECT</td>
<td>This property is optional. To run one or more Modules, Projects or Run Books on the new Agent as soon as it comes online, use this property to specify the GUIDs. You can find these GUIDs at Library &gt; Modules, Projects or Run Books. Separate multiple entries with a semi-colon (;). Separate multiple entries with a semi-colon (;). When specifying a Run Book, make sure the Who fields in the Run Book are not filled in. This ensures the Run Book Jobs are executed on the new Agent.</td>
</tr>
</tbody>
</table>
Additionally, if a rule-based Team exists at your RES ONE Automation environment and the option **Automatically set as primary team for new members** is selected for this Team, you can also apply the following public properties, to specify whether the Agent should inherit the Dispatcher detection settings of the primary Team when it is installed:

<table>
<thead>
<tr>
<th>Property</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISDEFDL</td>
<td>Specifies whether the new Agent should use the default/global Dispatcher list when it comes online.</td>
</tr>
<tr>
<td></td>
<td>• ‘1’ if the Agent should use the default/global settings.</td>
</tr>
<tr>
<td></td>
<td>• ‘0’ if the Agent should use its own settings.</td>
</tr>
<tr>
<td>ISDEFDLS</td>
<td>Specifies whether the new Agent should use the default/global Dispatcher location settings when it comes online.</td>
</tr>
<tr>
<td></td>
<td>• ‘1’ if the Agent should use the default/global settings.</td>
</tr>
<tr>
<td></td>
<td>• ‘0’ if the Agent should use its own settings.</td>
</tr>
<tr>
<td>ISDEFDR</td>
<td>Specifies whether the new Agent should use the default/global Dispatcher recovery settings when it comes online.</td>
</tr>
<tr>
<td></td>
<td>• ‘1’ if the Agent should use the default/global settings.</td>
</tr>
<tr>
<td></td>
<td>• ‘0’ if the Agent should use its own settings.</td>
</tr>
</tbody>
</table>

**Example**

The site license for your RES ONE Automation environment is 'RES-519E-04D1-FCBA-434A-80EE-99F8-FD8D-2E04'. The Agent must connect to two Dispatchers after it has started, 'qa-w7test01' and 'qa-w2k8test10'. The Agent is not allowed to use other Dispatchers. After the Agent gets online, it must be member of the Team 'Exchange Servers', and it must run a Project with the following GUID: '{301620AE-B650-4B87-BA02-05B6114E1F83}'.

To install an Agent unattended, use:

```
DISPATCHERLIST="w7test01;w2k8test10" DISPATCHERGETLIST="0"
ADDTOTEAM="Exchange Servers" INVOKEPROJECT="(301620AE-B650-4B87-BA02-05B6114E1F83)" /qn
```
9.2.9 Installing Consoles

Deploy a Console unattended if, for example, you want to deploy it on many computers at once using a command line. When you deploy a Console unattended, it is not necessary to preconfigure it. Unattended deployments can be scripted, for RIS or BareMetal installations.

1. Open the Console and go to Administration > Components.
2. Right-click the file RES-ONE-Automation-Console-[version].msi and choose Save component as.
3. Do not preconfigure the MSI file.
4. Save the MSI file to the desired location.
5. Apply the following public properties to the MSI file in the command line:

<table>
<thead>
<tr>
<th>Property</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBSERVER</td>
<td>Specifies the database server for the Datastore.</td>
</tr>
<tr>
<td>DBNAME</td>
<td>Specifies the database name for the Datastore.</td>
</tr>
<tr>
<td>DBUSER</td>
<td>Specifies the database login for the Datastore.</td>
</tr>
<tr>
<td>DBPASSWORD</td>
<td>Specifies the database login password for the Datastore.</td>
</tr>
<tr>
<td>DBTYPE</td>
<td>Specifies the database type (either MSSQL, MSSQLAZURE, DB2, ORACLE or MySQL).</td>
</tr>
<tr>
<td>DBPROTOCOLENCYPTION</td>
<td>Specifies whether protocol encryption is used to secure communication between the Datastore and the Console.</td>
</tr>
</tbody>
</table>

Example

Your SQL Server is called ‘SQLSERVER01’. The Datastore is called ‘automate’. The login and password for the Datastore is ‘automateuser’/‘Neverguess’.

To deploy a Console unattended, use: Msiexec /i "c:\RES-ONE-Automation-Console-[version].msi" DBSERVER="SQLSERVER01" DBNAME="automate" DBUSER="automateuser" DBPASSWORD="Neverguess" DBTYPE="MSSQL" /qn
9.2.10 Installing Dispatchers

Deploy a Dispatcher unattended if, for example, you want to deploy it on many computers at once using a command line. When you deploy a Dispatcher unattended, it is not necessary to preconfigure it. Unattended deployments can be scripted, for RIS or BareMetal installations.

1. Open the Console and go to Administration > Components.
2. Right-click the relevant installation file and choose Save component as:
   - RES-ONE-Automation-Dispatcher+(x64)-x.x.x.x.msi for 64-bit versions of Microsoft Windows
   - RES-ONE-Automation-Dispatcher+(x86)-x.x.x.x.msi for 32-bit versions of Microsoft Windows
3. Do not preconfigure the MSI file.
4. Click OK and save the MSI file to the desired location.
5. At the desired location, apply the following public properties to the MSI file in a command line:

<table>
<thead>
<tr>
<th>Property</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBSERVER</td>
<td>Specifies the database server for the Datastore.</td>
</tr>
<tr>
<td>DBNAME</td>
<td>Specifies the database name for the Datastore.</td>
</tr>
<tr>
<td>DBUSER</td>
<td>Specifies the database login for the Datastore.</td>
</tr>
<tr>
<td>DBPASSWORD</td>
<td>Specifies the database login password for the Datastore.</td>
</tr>
<tr>
<td>DBTYPE</td>
<td>Specifies the database type (either MSSQL, MSSQLAZURE, DB2, ORACLE or MYSQL).</td>
</tr>
<tr>
<td>DBPROTOCOL</td>
<td>Specifies whether protocol encryption is used to secure communication</td>
</tr>
<tr>
<td>ENCRYPTION</td>
<td>between the Datastore and the Dispatcher.</td>
</tr>
</tbody>
</table>

Example

Your SQL Server is called “SQLSERVER01”. The Datastore is called ‘automate’. The login and password for the Datastore is ‘automateuser’ / ‘Neverguess’. The OS version of the target machines is Microsoft Windows 7 (64-bit).

To install a Dispatcher unattended, use: Msiexec /i "c:\res-one-automation-dispatcher-(x64)-[version].msi" DBSERVER="SQLSERVER01" DBNAME="automate" DBUSER="automateuser" DBPASSWORD="Neverguess" DBTYPE=MSSQL DBPROTOCOL="No" /qn
9.2.11 Installing RES ONE Automation

Besides installing RES ONE Automation manually, you can also install RES ONE Automation unattended using a command line. This is useful if you need to install RES ONE Automation on several computers at once.

1. Installing RES ONE Automation unattended and connecting to an existing Datastore

If you want to install RES ONE Automation unattended and a Datastore already exists, you can choose to connect the Console to this Datastore as part of the installation. To do so, apply the following public properties to the RES ONE Automation Installer file:

<table>
<thead>
<tr>
<th>Property</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBCREATE</td>
<td>Specifies whether a new database should be created using the values below. When connecting to an existing Datastore, use No.</td>
</tr>
<tr>
<td>DBTYPE</td>
<td>Specifies the database type (either MSQL, MSQLAZURE, DB2, Oracle or MYSQL).</td>
</tr>
<tr>
<td>DBSERVER</td>
<td>Specifies the database server that RES ONE Automation should connect to.</td>
</tr>
<tr>
<td>DBNAME</td>
<td>Specifies the database name that RES ONE Automation should connect to.</td>
</tr>
<tr>
<td>DBUSER</td>
<td>Specifies the database user name that RES ONE Automation should use to connect to the database.</td>
</tr>
<tr>
<td>DBPASSWORD</td>
<td>Specifies the database password that RES ONE Automation should use to connect to the database.</td>
</tr>
<tr>
<td>DBPROTOCOLENCRIPTION</td>
<td>Specifies whether protocol encryption should be used when connecting to Microsoft SQL Server. Values are: Yes or No (default is No).</td>
</tr>
</tbody>
</table>

Example

Msiexec /i "c:\RES ONE Automation 10.2.0.0.msi" DBCREATE="No" DBTYPE="MSSQL" DBSERVER="SQLSERVER01" DBNAME="AutomationDB" DBUSER="AutomationDBUser" DBPASSWORD="Aut0m@t10nDBP@55w0rd" DBPROTOCOLENCRIPTION="No" /qn
2. Installing RES ONE Automation unattended and creating a new Datastore

If you want to install RES ONE Automation unattended and a Datastore does not yet exist, you can choose to create a new Datastore and connect the Console to this Datastore as part of the installation. To do so, apply the public properties mentioned above AND additionally:

<table>
<thead>
<tr>
<th>Property</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBCREATE</td>
<td>Specifies whether a new database should be created. When creating a new</td>
</tr>
<tr>
<td></td>
<td>Datastore, use Yes.</td>
</tr>
<tr>
<td>DBCREATEUSER</td>
<td>Specifies the database user name that should be used to create the new</td>
</tr>
<tr>
<td></td>
<td>database, for example 'sa'.</td>
</tr>
<tr>
<td>DBCREATEPASSWORD</td>
<td>Specifies the database password that should be used to create the new</td>
</tr>
<tr>
<td></td>
<td>database.</td>
</tr>
<tr>
<td>DBIMPORTLICENSE</td>
<td>Specifies a license file (including full path) to be imported after creating</td>
</tr>
<tr>
<td></td>
<td>a new database (optional).</td>
</tr>
<tr>
<td>DBEVALMODE</td>
<td>Specifies if, during the evaluation period, RES ONE Automation should</td>
</tr>
<tr>
<td></td>
<td>have all features available, or only the PoSh Automation features. Values</td>
</tr>
<tr>
<td></td>
<td>are FULL or POSH (default is FULL) (optional).</td>
</tr>
</tbody>
</table>

In this scenario, the following will be created:
- a new database (the value for DBNAME)
- a new database user (the value for DBUSER)
- a password for the user (the value for DBPASSWORD)

**Example**

Msiexec /i "c:\RES ONE Automation 10.2.0.0.msi" DBCREATE="Yes"
DBTYPE="MSSQL" DBSERVER="SQLSERVER01" DBNAME="AutomationDB"
DBUSER="AutomationDBUser" DBPASSWORD="Aut0m@t10nDBP@55w0rd"
DBPROTOCOLENCRYPTION="No" DBCREATEUSER="SA" DBCREATEPASSWORD="SAPassword"
DBIMPORTLICENSE="C:\Program Files\RES Software\Automation Manager\WMC\licensefile.xml" /qn

For Microsoft Azure SQL and DB2 database systems, it is not possible to create a new database that RES ONE Automation can use as its Datastore during an unattended RES ONE Automation installation.

What is possible, is:
3. Installing RES ONE Automation unattended, connect to an existing (empty) database and convert it to a RES ONE Automation Datastore

If you want to install RES ONE Automation unattended and a Datastore does not yet exist, you can also choose to connect to an existing (empty) database, convert it to a RES ONE Automation Datastore and connect the Console to this Datastore as part of the installation. To do so, apply the following public properties to the RES ONE Automation Installer file:

<table>
<thead>
<tr>
<th>Property</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBCREATE</td>
<td>Specifies (in this scenario) whether the database should be converted into a Datastore. Use Yes.</td>
</tr>
<tr>
<td>DBTYPE</td>
<td>Specifies the database type (either MSSQL, MSSQLAZURE, DB2, Oracle or MYSQL).</td>
</tr>
<tr>
<td>DBSERVER</td>
<td>Specifies the database server that RES ONE Automation should connect to.</td>
</tr>
<tr>
<td>DBNAME</td>
<td>Specifies the database name that RES ONE Automation should connect to.</td>
</tr>
<tr>
<td>DBCREATEUSER</td>
<td>Specifies the database user name (owner) that should be used to convert the database.</td>
</tr>
<tr>
<td>DBCREATEPASSWORD</td>
<td>Specifies the password for the DBCREATEUSER that RES ONE Automation should use to convert the database.</td>
</tr>
<tr>
<td>DBUSER</td>
<td>Specifies the database user name that RES ONE Automation should use to connect to the database.</td>
</tr>
<tr>
<td>DBPASSWORD</td>
<td>Specifies the password for the DBUSER that RES ONE Automation should use to connect to the database.</td>
</tr>
<tr>
<td>DBIMPORTLICENSE</td>
<td>Specifies a license file (including full path) to be imported after creating a new database (optional).</td>
</tr>
<tr>
<td>DBEVALMODE</td>
<td>Specifies if, during the evaluation period, RES ONE Automation should have all features available, or only the PoSh Automation features. Values are FULL or POSH (default is FULL) (optional).</td>
</tr>
</tbody>
</table>

During unattended installation, this method can only be used for Microsoft Azure SQL or DB2 databases. If you want to apply the same principle to other database systems, you can connect to the empty database from the Console after installation has completed.

Example

Msiexec /i "c:\RES ONE Automation 10.2.0.0.msi" DBTYPE="MSSQLAZURE"
DBSERVER="dbserver01.database.windows.net" DBNAME="AutomationDB"
DBCREATE="Yes" DBCREATEUSER="DBAdmin" DBCREATEPASSWORD="DBAdminPassword"
DBUSER="AutomationDBUser" DBPASSWORD="Aut0m@t10nDBP@55w0rd"
DBEVALMODE="POSH" /qn

Notes

- You can use all other parameters of msiexec in the command line as well.
9.2.12 Scheduling Jobs

You can also schedule Jobs unattended, for example if you want to schedule Jobs using a third-party product. You can schedule a Job unattended by running the executable of the Console (wmc.exe) from the command line using the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>/action</td>
<td>schedule</td>
<td>Specifies the action, in this case schedule. (Another option is importbb to import a Building Block unattended.)</td>
</tr>
<tr>
<td>/module</td>
<td>&lt;GUID&gt;</td>
<td>Identifies the Module, Project or Run Book.</td>
</tr>
<tr>
<td>or</td>
<td>/project</td>
<td></td>
</tr>
<tr>
<td>or</td>
<td>/runbook</td>
<td></td>
</tr>
<tr>
<td>/agent</td>
<td>&lt;GUID&gt;</td>
<td>Identifies the Agent or Team.</td>
</tr>
<tr>
<td>or</td>
<td>/team</td>
<td>This parameter can be left out for Run Books for which all target Agents or Teams have already been configured.</td>
</tr>
<tr>
<td>/user</td>
<td>&lt;user&gt;</td>
<td>Optional: the RES ONE Automation credentials to access the Console.</td>
</tr>
<tr>
<td>-and-</td>
<td>&lt;password&gt;</td>
<td></td>
</tr>
<tr>
<td>/parameterfile</td>
<td>&lt;path&gt;</td>
<td>Optional: use parameter values from a CSV file.</td>
</tr>
<tr>
<td>/scheduleon</td>
<td>job</td>
<td>Optional: applies to Run Books that are scheduled with CSV files only. Specifies whether the Run Book is to be scheduled per Run Book Job or per Task (default).</td>
</tr>
<tr>
<td>or</td>
<td>task</td>
<td></td>
</tr>
<tr>
<td>/parallel</td>
<td></td>
<td>Optional: run the Job in parallel</td>
</tr>
<tr>
<td>/silent</td>
<td></td>
<td>Optional: suppresses any messages.</td>
</tr>
<tr>
<td>/onlineagentsonly</td>
<td></td>
<td>Optional: restrict Job to online Agents in a Team only.</td>
</tr>
</tbody>
</table>

Example

"C:\program files\res software\automation manager\wmc\wmc.exe"
/action=schedule /Module={69A0BC21-41F6-46DC-B6DE-5574138EF03C}
/team={26E7EB95-35A8-41D3-AED9-FCE71571D6BD} /user=jsmith /password=secret
/silent /onlineagentsonly

Notes

- You can schedule only one Module, Project or Run Book per command line.
- You can specify only one Team or Agent per command line.
- If all the necessary Teams and/or Agents have already been configured in the Run Book, you do not need to schedule an Agent or Team for a Run Book. If the Run Book has an empty Who field anywhere, you do need to specify an Agent or Team parameter.

Tip

The GUID of a Module, Project, Run Book, Agent or Team is shown on the item's Properties tab.
9.2.13 Solving a Console lockout

An administrator can only access the Console if at least one administrative role has been assigned to him.

If a user is locked out of the Console, RES ONE Automation can add a user’s Microsoft Windows account to the Full Access administrative role. To verify legitimacy, the user must provide the correct database credentials for the relevant Datastore. Use of this method, including failed attempts, is logged in the Audit Trail.

Procedure

1. Log on to the computer with the Microsoft Windows account that needs to get access to the Console.
2. Start the Console with the parameter /lockedout (for example c:\program files\RES Software\Automation Manager\WMC\WMC.exe /lockedout).
3. In the window RES ONE Automation Management Console - Locked out, provide the correct database login and database password.
4. Click OK. If the credentials are correct, the Microsoft Windows credentials are added to the Full Access administrative role. This grants access to the Console.
9.3 Registry settings

The following Registry Settings are available to control the behavior of RES ONE Automation components:

- Frequency at which Console checks for new updates (on page 305)
- Location of Dispatcher Resource cache folder (on page 305)
- Master caching Dispatcher (on page 305)
- Tracing (on page 306)

**Warning**

Careless registry editing can cause irreversible damage! Approach this task with caution, and only after you have made a backup.

9.3.1 Frequency at which Console checks for new updates

By default, the RES ONE Automation Console checks every 5 seconds on the Datastore whether a new version is available. This can lead to performance issues when the Console is installed on many machines, for example in large environments.

To configure the frequency at which the Console service checks for new updates, set the following registry setting:

<table>
<thead>
<tr>
<th>Key:</th>
<th>HKLM\Software\RES\AutomationManager\WMC\ (32-bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>HKLM\Software\Wow6432Node\RES\AutomationManager\WMC\ (64-bits)</td>
</tr>
<tr>
<td>Type:</td>
<td>REG_DWORD</td>
</tr>
<tr>
<td>Value name:</td>
<td>WMCSVCUpdateTimer</td>
</tr>
<tr>
<td>Value data:</td>
<td>A value between 30 and 86400 seconds (1 day).</td>
</tr>
<tr>
<td></td>
<td>The frequency check will only be activated when there are no Consoles active,</td>
</tr>
<tr>
<td></td>
<td>otherwise the frequency check for new updates will not be changed and will occur</td>
</tr>
<tr>
<td></td>
<td>every 5 seconds.</td>
</tr>
</tbody>
</table>

9.3.2 Location of Dispatcher Resource cache folder

To change the location of the Dispatcher Resource cache folder, do the following:

1. Stop the Dispatcher service.
2. Add the following registry key:

<table>
<thead>
<tr>
<th>Key</th>
<th>HKEY_LOCAL_MACHINE\SOFTWARE\RES\AutomationManager\Dispatcher</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>ResourceCachePath</td>
</tr>
<tr>
<td>Type</td>
<td>REG_SZ (String Value)</td>
</tr>
<tr>
<td>Value</td>
<td>Path to the Dispatcher resource folder</td>
</tr>
</tbody>
</table>

3. Start the Dispatcher service. After the Dispatcher service has started, the new Resource folder will be filled with recorded data from the SQL server.
4. Delete the contents of the old Resource folder. The default location is: "%programfiles%\RES Software\Automation Manager\Dispatcher\Resources".
9.3.3 **Master caching Dispatcher**

**Setup**

To set up a master caching Dispatcher, add the following registry key to the Dispatcher:

<table>
<thead>
<tr>
<th>Key</th>
<th>HKEY_LOCAL_MACHINE\SOFTWARE\RES\AutomationManager\Dispatcher</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>DispatcherPort</td>
</tr>
<tr>
<td>Data Type</td>
<td>REG_DWORD (Binary)</td>
</tr>
<tr>
<td>Value</td>
<td>Port number</td>
</tr>
</tbody>
</table>

**Different port**

To specify a different port on a master caching Dispatcher, add the following registry key:

<table>
<thead>
<tr>
<th>Key</th>
<th>HKEY_LOCAL_MACHINE\SOFTWARE\RES\AutomationManager\Dispatcher</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>DispatcherPort</td>
</tr>
<tr>
<td>Type</td>
<td>REG_DWORD (Binary)</td>
</tr>
<tr>
<td>Value</td>
<td>Port number</td>
</tr>
<tr>
<td>Default</td>
<td>3163</td>
</tr>
</tbody>
</table>

**Remarks**

With the registry settings `DispatcherPort`, you can specify a different port on the master caching Dispatcher (for example, 3164). This makes the master caching Dispatcher unavailable for Agents. It may still be reached by other Dispatchers, provided you have specified this on these Dispatchers.

**Redirect to master caching Dispatcher**

To redirect a Dispatcher to a master caching Dispatcher, add the following registry keys on this Dispatcher:

<table>
<thead>
<tr>
<th>Key</th>
<th>HKEY_LOCAL_MACHINE\SOFTWARE\RES\AutomationManager\Dispatcher</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>MasterCacheDispatcherList</td>
</tr>
<tr>
<td>Data Type</td>
<td>REG_SZ</td>
</tr>
<tr>
<td>Value</td>
<td>Address of the master caching Dispatcher. Separate multiple Dispatchers using a semicolon (;)</td>
</tr>
</tbody>
</table>

**Redirect to different port**

If you have specified a different port on the master caching Dispatcher, you need to specify this port on each Dispatcher that redirects to the master caching Dispatcher.

To specify a different master caching Dispatcher port, set the following registry setting on each Dispatcher:

<table>
<thead>
<tr>
<th>Key</th>
<th>HKEY_LOCAL_MACHINE\SOFTWARE\RES\AutomationManager\Dispatcher</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>MasterCacheDispatcherPort</td>
</tr>
<tr>
<td>Data Type</td>
<td>REG_DWORD (Binary)</td>
</tr>
<tr>
<td>Value</td>
<td>Port number of the master caching Dispatcher</td>
</tr>
</tbody>
</table>

**See** Master caching **(on page 181)**.
### 9.3.4 Tracing

To create a trace file for troubleshooting RES ONE Automation, do the following:

For 32-bit machines, add the following registry values:

<table>
<thead>
<tr>
<th>Key</th>
<th>HKEY_LOCAL_MACHINE\Software\RES\AutomationManager</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Trace</td>
</tr>
<tr>
<td>Data Type</td>
<td>REG_SZ (String Value)</td>
</tr>
<tr>
<td>Value</td>
<td>yes = enabled, no = disabled</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Key</th>
<th>HKEY_LOCAL_MACHINE\Software\RES\AutomationManager</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>TraceDetailed</td>
</tr>
<tr>
<td>Data Type</td>
<td>REG_SZ (String Value)</td>
</tr>
<tr>
<td>Value</td>
<td>yes = enabled, no = disabled</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Key</th>
<th>HKEY_LOCAL_MACHINE\Software\RES\AutomationManager</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>TraceFile</td>
</tr>
<tr>
<td>Data Type</td>
<td>REG_SZ (String Value)</td>
</tr>
<tr>
<td>Value</td>
<td>&lt;path_to_logfile&gt;&lt;filename&gt;</td>
</tr>
</tbody>
</table>

For 64-bit machines, add the following registry values:

<table>
<thead>
<tr>
<th>Key</th>
<th>HKEY_LOCAL_MACHINE\Software\Wow6432Node\RES\AutomationManager</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Trace</td>
</tr>
<tr>
<td>Data Type</td>
<td>REG_SZ (String Value)</td>
</tr>
<tr>
<td>Value</td>
<td>yes = enabled, no = disabled</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Key</th>
<th>HKEY_LOCAL_MACHINE\Software\Wow6432Node\RES\AutomationManager</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>TraceDetailed</td>
</tr>
<tr>
<td>Data Type</td>
<td>REG_SZ (String Value)</td>
</tr>
<tr>
<td>Value</td>
<td>yes = enabled, no = disabled</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Key</th>
<th>HKEY_LOCAL_MACHINE\Software\Wow6432Node\RES\AutomationManager</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>TraceFile</td>
</tr>
<tr>
<td>Data Type</td>
<td>REG_SZ (String Value)</td>
</tr>
<tr>
<td>Value</td>
<td>&lt;path&gt;\filename</td>
</tr>
</tbody>
</table>

After adding the registry keys, restart the RES ONE Automation services.
Notes

- Because these settings are set on `HKEY_LOCAL_MACHINE\SOFTWARE\RES\AutomationManager` level, all RES ONE Automation services (Agent, Dispatcher, Console) installed on the computer will be traced.
- The file will always be 2 MB. Older entries will be overwritten within the file. If the trace is running for a longer period, the file should be copied on a regular basis.
- The last line in the file is not necessarily the latest entry.
9.4 Logging

9.4.1 Event logs

The following actions are available for RES ONE Automation:

<table>
<thead>
<tr>
<th>Event ID</th>
<th>Message</th>
<th>Entry type</th>
<th>Source</th>
<th>Log name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Service started successfully</td>
<td>Information</td>
<td>RESWAS, RESWDS</td>
<td>Application</td>
</tr>
<tr>
<td>0</td>
<td>Service stopped successfully</td>
<td>Information</td>
<td>RESWAS, RESWDS</td>
<td>Application</td>
</tr>
<tr>
<td>105</td>
<td>The service was started</td>
<td>Information</td>
<td>RESWCS</td>
<td>Application</td>
</tr>
<tr>
<td>107</td>
<td>The service was stopped</td>
<td>Information</td>
<td>RESWCS</td>
<td>Application</td>
</tr>
<tr>
<td>1000</td>
<td>Logged in to environment id: &lt;GUID&gt;</td>
<td>Information</td>
<td>RESWAS</td>
<td>Application</td>
</tr>
<tr>
<td>1001</td>
<td>Failed to login to environment id: &lt;GUID&gt;, failed to validate communication id</td>
<td>Warning</td>
<td>RESWAS</td>
<td>Application</td>
</tr>
<tr>
<td>1002</td>
<td>Job started: &lt;Name&gt; JobGUID: &lt;GUID&gt;</td>
<td>Information</td>
<td>RESWAS</td>
<td>Application</td>
</tr>
<tr>
<td>1003</td>
<td>Project started: &lt;Name&gt; ProjectGUID: &lt;GUID&gt;</td>
<td>Information</td>
<td>RESWAS</td>
<td>Application</td>
</tr>
<tr>
<td>1004</td>
<td>Module started: &lt;Name&gt; ModuleGUID: &lt;GUID&gt;</td>
<td>Information</td>
<td>RESWAS</td>
<td>Application</td>
</tr>
<tr>
<td>1005</td>
<td>Job: &lt;Name&gt; JobGUID: &lt;GUID&gt; ended with status: Completed</td>
<td>Information</td>
<td>RESWAS</td>
<td>Application</td>
</tr>
<tr>
<td>1006</td>
<td>Job: &lt;Name&gt; JobGUID: &lt;GUID&gt; ended with status: Failed</td>
<td>Warning</td>
<td>RESWAS</td>
<td>Application</td>
</tr>
<tr>
<td>1007</td>
<td>Job: &lt;Name&gt; JobGUID: &lt;GUID&gt; ended with status: Cancelled</td>
<td>Information</td>
<td>RESWAS</td>
<td>Application</td>
</tr>
<tr>
<td>1009</td>
<td>Agent is in Full Mode</td>
<td>Information</td>
<td>RESWAS</td>
<td>Application</td>
</tr>
<tr>
<td>1010</td>
<td>Connected to Dispatcher: &lt;GUID&gt;</td>
<td>Information</td>
<td>RESWAS</td>
<td>Application</td>
</tr>
<tr>
<td>1011</td>
<td>Got changes from dispatcher log: Agent current change guids: &lt;GUID&gt;</td>
<td>Information</td>
<td>RESWAS</td>
<td>Application</td>
</tr>
<tr>
<td></td>
<td>Response change guids: &lt;GUID&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1012</td>
<td>Update C# Agent: starting updater process: &lt;updater path&gt; with component &lt;agent msi path&gt;</td>
<td>Information</td>
<td>RESWAS</td>
<td>Application</td>
</tr>
<tr>
<td>1013</td>
<td>Discovered for environment id: &lt;GUID&gt;: the following dispatchers: &lt;GUID&gt;; &lt;GUID&gt;</td>
<td>Information</td>
<td>RESWAS</td>
<td>Application</td>
</tr>
<tr>
<td>1014</td>
<td>Agent logged off</td>
<td>Information</td>
<td>RESWAS</td>
<td>Application</td>
</tr>
<tr>
<td>1015</td>
<td>Project ended: &lt;Name&gt; ProjectGUID: &lt;GUID&gt;</td>
<td>Information</td>
<td>RESWAS</td>
<td>Application</td>
</tr>
<tr>
<td>1016</td>
<td>Module ended: &lt;Name&gt; ModuleGUID: &lt;GUID&gt;</td>
<td>Information</td>
<td>RESWAS</td>
<td>Application</td>
</tr>
<tr>
<td>1017</td>
<td>Job: &lt;Name&gt; JobGUID: &lt;GUID&gt; ended with status: Aborted</td>
<td>Warning</td>
<td>RESWAS</td>
<td>Application</td>
</tr>
<tr>
<td>2000</td>
<td>Agent is offline. Could not discover any dispatchers for environment id: &lt;GUID&gt;</td>
<td>Warning</td>
<td>RESWAS</td>
<td>Application</td>
</tr>
<tr>
<td>Event ID</td>
<td>Message</td>
<td>Entry type</td>
<td>Source</td>
<td>Log name</td>
</tr>
<tr>
<td>----------</td>
<td>-------------------------------------------------------------------------</td>
<td>------------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>2001</td>
<td>Agent has been offline for 1 hour. Could not discover any dispatchers for environment id: &lt;GUID&gt;</td>
<td>Warning</td>
<td>RESWAS</td>
<td>Application</td>
</tr>
<tr>
<td>2002</td>
<td>Agent has been offline for 2 hours. Could not discover any dispatchers for environment id: &lt;GUID&gt;</td>
<td>Warning</td>
<td>RESWAS</td>
<td>Application</td>
</tr>
<tr>
<td>2004</td>
<td>Agent has been offline for 4 hours. Could not discover any dispatchers for environment id: &lt;GUID&gt;</td>
<td>Warning</td>
<td>RESWAS</td>
<td>Application</td>
</tr>
<tr>
<td>2008</td>
<td>Agent has been offline for 8 hours. Could not discover any dispatchers for environment id: &lt;GUID&gt;</td>
<td>Warning</td>
<td>RESWAS</td>
<td>Application</td>
</tr>
<tr>
<td>2024</td>
<td>Agent has been offline for 24 hours. Could not discover any dispatchers for environment id: &lt;GUID&gt;</td>
<td>Warning</td>
<td>RESWAS</td>
<td>Application</td>
</tr>
<tr>
<td>2048</td>
<td>Agent has been offline for 48 hours. Could not discover any dispatchers for environment id: &lt;GUID&gt;</td>
<td>Warning</td>
<td>RESWAS</td>
<td>Application</td>
</tr>
<tr>
<td>3001</td>
<td>Unable to initialize RES ONE Automation Datastore! Unable to start RES ONE Automation Dispatcher+ service! Please check the connection settings.</td>
<td>Error</td>
<td>RESWDS</td>
<td>Application</td>
</tr>
<tr>
<td>3002</td>
<td>Connected to RES ONE Automation Datastore. Starting RES ONE Automation Dispatcher+ service.</td>
<td>Information</td>
<td>RESWDS</td>
<td>Application</td>
</tr>
<tr>
<td>3003</td>
<td>Unable to connect to RES ONE Automation Datastore! Not able to start RES ONE Automation Dispatcher+ service yet. Retrying to connect every 5 seconds.</td>
<td>Warning</td>
<td>RESWDS</td>
<td>Application</td>
</tr>
<tr>
<td>3004</td>
<td>Unable to download a resource from the datastore. Please check if the datastore driver for known issues about fetching blob data.</td>
<td>Error</td>
<td>RESWDS</td>
<td>Application</td>
</tr>
<tr>
<td>3005</td>
<td>Dispatcher started &lt;Assembly, File and Product version&gt;</td>
<td>Information</td>
<td>RESWDS</td>
<td>Application</td>
</tr>
<tr>
<td>3006</td>
<td>Connection to datastore has been restored. Starting Dispatcher IP listeners.</td>
<td>Information</td>
<td>RESWDS</td>
<td>Application</td>
</tr>
<tr>
<td>3007</td>
<td>Connection to the datastore has failed (Error: &lt;Error message&gt;). Stopping Dispatcher IP listeners.</td>
<td>Error</td>
<td>RESWDS</td>
<td>Application</td>
</tr>
<tr>
<td>3008</td>
<td>Started Dispatcher socket listener on port &lt;Port no&gt;</td>
<td>Information</td>
<td>RESWDS</td>
<td>Application</td>
</tr>
<tr>
<td>3009</td>
<td>Failed to start Dispatcher socket listener on port &lt;Port no&gt;. &lt;Error message&gt;</td>
<td>Error</td>
<td>RESWDS</td>
<td>Application</td>
</tr>
<tr>
<td>3010</td>
<td>Unable to update dispatcher status. Disable role service: &quot;Housekeeping&quot; or &quot;LicenseReporting&quot;</td>
<td>Information</td>
<td>RESWDS</td>
<td>Application</td>
</tr>
<tr>
<td>Event ID</td>
<td>Message</td>
<td>Entry type</td>
<td>Source</td>
<td>Log name</td>
</tr>
<tr>
<td>----------</td>
<td>-------------------------------------------------------------------------</td>
<td>------------</td>
<td>--------</td>
<td>-----------</td>
</tr>
<tr>
<td>3011</td>
<td>(&lt;Updated dispatcher role status&gt;) &lt;&quot;LicenseReporting&quot; or &quot;Garbage collector (Housekeeping)&quot;&gt; enabled is &lt;&quot;True&quot; or &quot;False&quot;&gt;</td>
<td>Information</td>
<td>RESWDS</td>
<td>Application</td>
</tr>
<tr>
<td>3012</td>
<td>Failed to start Dispatcher SchedulingService (WebAPI) on host &lt;URL&gt; port &lt;Port no&gt; (https/http): &lt;Error message&gt;</td>
<td>Error</td>
<td>RESWDS</td>
<td>Application</td>
</tr>
</tbody>
</table>
9.4.2 **Telemetry**

From RES ONE Automation 10.1, RES is monitoring the functionality that is used within the product. This allows us to prioritize the areas that we focus our development efforts on. That way, we have:

- More visibility on how the product is used amongst the different customers
- Better grip on deployment of new Agents, and used operating system versions
- More accurate information on systems used by our customers to make informed decisions about the deprecation of features or functionality
- Steady adoption rate of the Management Portal

See also: **Telemetry tasks of Dispatcher** (on page 190)

9.5 **Complete list of collected data provided to RES:**

- Site GUID
- Database Level
- Version of RES ONE Automation
- Edition (Full, Evaluation)
- AES Encrypted (Yes/No)

Dispatchers:

- Number of Dispatchers
- Version of Dispatchers
- Operating System of Dispatchers
- Operating System Version of Dispatchers
- .NET Framework Version on Dispatchers

Agents (detailed for legacy Agents, Agents+, Linux Agents and Mac OS X Agents):

- Total Number of Agents
- Number of Agents
- Version of Agents
- Operating System of Agents
- Operating System Version of Agents
- .NET Framework Version on Agents

Database:

- Database Type
- Database Version

Configuration:

- Task Name
- Number of times used
- Query Name
- Number of times used
Chapter 9: Appendices

9.6 Flowcharts

Dispatcher Discovery (on page 313)

9.6.1 Dispatcher Discovery
### 9.7 RES ONE Automation Glossary

<table>
<thead>
<tr>
<th>Object</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent</td>
<td>An Agent is a service running on a computer and can execute RES ONE Automation Tasks.</td>
</tr>
<tr>
<td>Administrative role</td>
<td>An administrative role determines the level of access that a Console user has to the various parts of the Console.</td>
</tr>
<tr>
<td>Building Block</td>
<td>A Building Block can be used to export or import objects in an environment's Library (Modules, Projects, Run Books and Resources), for backup purposes and to transport objects from one RES ONE Automation environment to another.</td>
</tr>
<tr>
<td>Condition</td>
<td>A condition determines, based on one or more expressions, whether a Task, Module or Run Book Job should be executed, skipped or failed.</td>
</tr>
<tr>
<td>Console</td>
<td>The Console is the application from which RES ONE Automation is configured and used, from setting up the RES ONE Automation environment to the execution of Tasks.</td>
</tr>
<tr>
<td>Datastore</td>
<td>The Datastore is a database that contains all the settings and Resources of your RES ONE Automation environment.</td>
</tr>
<tr>
<td>Dispatcher</td>
<td>A Dispatcher is a service running on a computer, and communicates information from the Datastore to Agents. A Dispatcher can cache Resources in order to reduce network traffic.</td>
</tr>
<tr>
<td>Evaluator</td>
<td>An evaluator can only be used in query Tasks. Based on the query results, it determines the status of the query Task after it has been executed.</td>
</tr>
<tr>
<td>Function</td>
<td>A function generates (partial) input for text-based properties of Tasks. This makes it possible to create information in various fields in Tasks automatically.</td>
</tr>
<tr>
<td>Global Settings</td>
<td>Global Settings determine the general behavior of your RES ONE Automation environment, and the default behavior of Dispatchers and Agents.</td>
</tr>
<tr>
<td>Instant Reports</td>
<td>An Instant Report is a detailed report of your RES ONE Automation environment and includes all settings of the documented item(s). It can also include additional information about the current RES ONE Automation version.</td>
</tr>
<tr>
<td>Job</td>
<td>A Job is the execution of Tasks at a specific time on specific Agents.</td>
</tr>
<tr>
<td></td>
<td>- For Modules and Projects, both the time and the Agents are set at the moment of scheduling.</td>
</tr>
<tr>
<td></td>
<td>- For Run Books, the timing is set at the moment of scheduling. The Agents may already have been set for some or all of the Modules and Projects contained in the Run Book. In that case, Agents are not set at the moment of scheduling, or only for the items not yet given a target.</td>
</tr>
<tr>
<td>Management Portal</td>
<td>The Management Portal is the web-based application for scheduling Jobs in your RES ONE Automation environment.</td>
</tr>
<tr>
<td>Module</td>
<td>A Module is a container for one or more Tasks. To execute the Tasks in a Module, the Module can be scheduled as a Job on a set of Agents at a specific time.</td>
</tr>
<tr>
<td>Parameter</td>
<td>A parameter functions as a placeholder for values in various fields, such as text, file paths, credentials, etc. The actual values of these fields can be provided when the Task is used in a Job. This makes it possible to create generic Modules, Projects and Run Books that can be customized to each situation when required.</td>
</tr>
<tr>
<td>Object</td>
<td>Function</td>
</tr>
<tr>
<td>---------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Project</td>
<td>Projects are containers for Modules. By combining several Modules into one Project, those Modules can be scheduled on a set of Agents as one Job instead of separately as a series of Jobs.</td>
</tr>
<tr>
<td>Resource</td>
<td>A Resource can be any file that is needed to perform a certain Task, such as an MSI file, a setup file, a patch, a hotfix, etc.</td>
</tr>
<tr>
<td>Run Book</td>
<td>Run Books are containers for Jobs. Run Books can contain Modules and Projects plus (optionally) the Agents on which they should be executed. To execute the Tasks in the containers in the Run Book, the Run Book is scheduled as Job for a specific time. By combining Modules and Projects and their Agents in a Run Book, a sequence of actions can be scheduled consecutively on different sets of Agents.</td>
</tr>
<tr>
<td>Task</td>
<td>A Task is a specific action that can be performed by an Agent. Tasks are part of a Module.</td>
</tr>
<tr>
<td>Team</td>
<td>A Team is a group of Agents. Teams can also contain other Teams.</td>
</tr>
<tr>
<td>Trusts</td>
<td>A Trust allows a customer representative to enforce that only Resources and Modules that have been explicitly authorized can be used on that customer’s Agents. When Trusts are enabled, Modules and Resources can only be used by Agents if a full Trust exists.</td>
</tr>
<tr>
<td>Variable</td>
<td>A Variable acts as a placeholder for a customer-specific value, such as, for example, server names, passwords or credentials. Besides a global, default value, a Variable can also have an Agent-specific and a Team-specific value. A Variable is resolved at the moment of Job execution.</td>
</tr>
</tbody>
</table>
9.8 Compatibility Matrix

Irrespective of this compatibility matrix, RES does not support versions of products for which End of Life (EOL) has been declared by their respective vendors.

9.8.1 Supported Microsoft Windows operating systems

<table>
<thead>
<tr>
<th>Windows Server version</th>
<th>Edition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2016</td>
<td>Server 2016 Datacenter</td>
</tr>
<tr>
<td></td>
<td>Server 2016 Standard</td>
</tr>
<tr>
<td></td>
<td>Server 2016 Essentials</td>
</tr>
<tr>
<td>Windows Server 2012 R2</td>
<td>Server 2012 R2 Datacenter</td>
</tr>
<tr>
<td></td>
<td>Server 2012 R2 Standard</td>
</tr>
<tr>
<td></td>
<td>Server 2012 R2 Essentials</td>
</tr>
<tr>
<td></td>
<td>Server 2012 R2 Foundation</td>
</tr>
<tr>
<td>Windows Server 2012</td>
<td>Server 2012 Datacenter</td>
</tr>
<tr>
<td></td>
<td>Server 2012 Standard</td>
</tr>
<tr>
<td></td>
<td>Server 2012 Essentials</td>
</tr>
<tr>
<td></td>
<td>Server 2012 Foundation</td>
</tr>
<tr>
<td>Windows Server 2008 R2</td>
<td>Server 2008 R2 Datacenter</td>
</tr>
<tr>
<td></td>
<td>Server 2008 R2 Enterprise</td>
</tr>
<tr>
<td></td>
<td>Server 2008 R2 Standard</td>
</tr>
<tr>
<td></td>
<td>Server 2008 R2 Web Edition</td>
</tr>
<tr>
<td>Windows Server 2008</td>
<td>(not supported by Agent+ and Dispatcher)</td>
</tr>
<tr>
<td>Windows Server 2003 R2</td>
<td>(not supported by Agent+ and Dispatcher)</td>
</tr>
<tr>
<td>Windows Server 2003</td>
<td>(not supported by Agent+ and Dispatcher)</td>
</tr>
<tr>
<td>Windows version</td>
<td>Edition</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>Windows 10</td>
<td>Windows 10 Enterprise</td>
</tr>
<tr>
<td></td>
<td>Windows 10 Enterprise LTSB</td>
</tr>
<tr>
<td></td>
<td>Windows 10 Pro</td>
</tr>
<tr>
<td></td>
<td>Windows 10 Education</td>
</tr>
<tr>
<td>Windows 8.1</td>
<td>Windows 8.1 Enterprise</td>
</tr>
<tr>
<td></td>
<td>Windows 8.1 Pro</td>
</tr>
<tr>
<td>Windows 8</td>
<td>Windows 8 Enterprise</td>
</tr>
<tr>
<td></td>
<td>Windows 8 Pro</td>
</tr>
<tr>
<td>Windows 7</td>
<td>Windows 7 Enterprise</td>
</tr>
<tr>
<td></td>
<td>Windows 7 Professional</td>
</tr>
<tr>
<td></td>
<td>Windows 7 Ultimate</td>
</tr>
<tr>
<td></td>
<td>Windows 7 Home Premium</td>
</tr>
<tr>
<td></td>
<td>Windows 7 Home Basic</td>
</tr>
<tr>
<td></td>
<td>Windows 7 Starter</td>
</tr>
<tr>
<td>Windows Vista</td>
<td>(not supported by Agent+ and Dispatcher)</td>
</tr>
<tr>
<td>Windows XP</td>
<td>(not supported by Agent+ and Dispatcher)</td>
</tr>
</tbody>
</table>
## 9.8.2 Supported Unix/Linux operating systems

<table>
<thead>
<tr>
<th>Unix/Linux</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red Hat Enterprise Linux</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>5.3 and higher</td>
</tr>
<tr>
<td></td>
<td>4.5 and higher</td>
</tr>
<tr>
<td>SUSE Linux Enterprise Server</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>11: SP1 and higher</td>
</tr>
<tr>
<td></td>
<td>10: SP3 and higher</td>
</tr>
<tr>
<td>IBM AIX</td>
<td>Version 7.1: Technical level 01 and higher (PPC)</td>
</tr>
<tr>
<td></td>
<td>Version 6.1: Technical level 7 and higher (PPC)</td>
</tr>
<tr>
<td></td>
<td>5L Version 5.3: Technical level 4 and higher (PPC)</td>
</tr>
<tr>
<td>Ubuntu</td>
<td>16.04</td>
</tr>
<tr>
<td></td>
<td>12.04</td>
</tr>
<tr>
<td>Oracle (Sun) Solaris</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>• for SPARC: update 1 and higher</td>
</tr>
<tr>
<td></td>
<td>• for Intel: update 3 and higher</td>
</tr>
<tr>
<td></td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>• for SPARC: update 6 and higher</td>
</tr>
<tr>
<td></td>
<td>• for Intel: update 11 and higher</td>
</tr>
<tr>
<td>Apple</td>
<td>Version</td>
</tr>
<tr>
<td>Apple Mac OS X</td>
<td>10.10 Yosemite</td>
</tr>
<tr>
<td></td>
<td>10.9 Mavericks</td>
</tr>
<tr>
<td></td>
<td>10.8 Mountain Lion</td>
</tr>
</tbody>
</table>

## 9.8.3 Supported Directory Services

<table>
<thead>
<tr>
<th>Directory Service</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Functional level: 2012 R2</td>
</tr>
<tr>
<td></td>
<td>Functional level: 2012</td>
</tr>
<tr>
<td></td>
<td>Functional level: 2008 R2</td>
</tr>
<tr>
<td></td>
<td>Functional level: 2008</td>
</tr>
<tr>
<td></td>
<td>Functional level: 2003</td>
</tr>
</tbody>
</table>
### 9.8.4 Supported Database systems

<table>
<thead>
<tr>
<th>Database type</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft SQL Server</td>
<td>2016</td>
</tr>
<tr>
<td></td>
<td>2014</td>
</tr>
<tr>
<td></td>
<td>2012</td>
</tr>
<tr>
<td></td>
<td>2008 R2</td>
</tr>
<tr>
<td></td>
<td>2008</td>
</tr>
<tr>
<td></td>
<td>2005</td>
</tr>
<tr>
<td>Microsoft Azure SQL</td>
<td></td>
</tr>
<tr>
<td>IBM DB2</td>
<td>10.5</td>
</tr>
<tr>
<td></td>
<td>9.8</td>
</tr>
<tr>
<td></td>
<td>9.7</td>
</tr>
<tr>
<td></td>
<td>9.5</td>
</tr>
<tr>
<td>MySQL</td>
<td>5.6</td>
</tr>
<tr>
<td></td>
<td>5.0</td>
</tr>
<tr>
<td>Oracle</td>
<td>12c</td>
</tr>
<tr>
<td></td>
<td>11.2</td>
</tr>
<tr>
<td></td>
<td>11.1</td>
</tr>
</tbody>
</table>

### 9.8.5 Supported Mail Servers

<table>
<thead>
<tr>
<th>Mail Server</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Exchange Server</td>
<td>2016</td>
</tr>
<tr>
<td></td>
<td>2013</td>
</tr>
<tr>
<td></td>
<td>2010 SP1</td>
</tr>
<tr>
<td></td>
<td>2010</td>
</tr>
<tr>
<td></td>
<td>2007 (not supported by Agent+)</td>
</tr>
<tr>
<td></td>
<td>2003 (not supported by Agent+)</td>
</tr>
</tbody>
</table>

### 9.8.6 Supported Application Virtualization

<table>
<thead>
<tr>
<th>Application Virtualization</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft App-V</td>
<td>4.6</td>
</tr>
<tr>
<td></td>
<td>4.5</td>
</tr>
</tbody>
</table>
### 9.8.7 Supported Hypervisors

<table>
<thead>
<tr>
<th>Hypervisor</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Hyper-V</td>
<td>2012 R2</td>
</tr>
<tr>
<td></td>
<td>2012</td>
</tr>
<tr>
<td></td>
<td>2008 R2</td>
</tr>
<tr>
<td>Citrix XenServer</td>
<td>6.5</td>
</tr>
<tr>
<td></td>
<td>6.2</td>
</tr>
</tbody>
</table>

### 9.8.8 Supported Provisioning Services

<table>
<thead>
<tr>
<th>Provisioning Service</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft System Center Configuration Manager</td>
<td>2016 (up to v1702)</td>
</tr>
<tr>
<td></td>
<td>2012 R2</td>
</tr>
<tr>
<td></td>
<td>2012 SP1</td>
</tr>
<tr>
<td></td>
<td>2012</td>
</tr>
<tr>
<td></td>
<td>2007</td>
</tr>
<tr>
<td>LANDesk</td>
<td>9.5 SP2</td>
</tr>
<tr>
<td></td>
<td>9.5 SP1</td>
</tr>
<tr>
<td></td>
<td>9.5</td>
</tr>
<tr>
<td>Citrix Workflow</td>
<td>2.5</td>
</tr>
</tbody>
</table>

### 9.8.9 Supported Microsoft Windows PowerShell Scripting

<table>
<thead>
<tr>
<th>Microsoft Scripting</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows PowerShell</td>
<td>1.0</td>
</tr>
<tr>
<td>Windows PowerShell</td>
<td>2.0</td>
</tr>
<tr>
<td>Windows PowerShell</td>
<td>3.0</td>
</tr>
<tr>
<td>Windows PowerShell</td>
<td>4.0</td>
</tr>
<tr>
<td>Windows PowerShell</td>
<td>5.0</td>
</tr>
</tbody>
</table>
### 9.8.10 Supported Citrix "Ready" products

<table>
<thead>
<tr>
<th>Citrix product</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>XenApp</td>
<td>7.6</td>
</tr>
<tr>
<td></td>
<td>7.5</td>
</tr>
<tr>
<td></td>
<td>6.5</td>
</tr>
<tr>
<td></td>
<td>6.0</td>
</tr>
<tr>
<td>XenDesktop</td>
<td>7.6</td>
</tr>
<tr>
<td></td>
<td>7.5</td>
</tr>
<tr>
<td></td>
<td>7.0</td>
</tr>
<tr>
<td></td>
<td>5.6</td>
</tr>
<tr>
<td></td>
<td>5.5</td>
</tr>
<tr>
<td></td>
<td>5.0</td>
</tr>
<tr>
<td>XenServer</td>
<td>6.1</td>
</tr>
<tr>
<td></td>
<td>6.0</td>
</tr>
</tbody>
</table>

### 9.8.11 Supported RES products

<table>
<thead>
<tr>
<th>RES product</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>RES Workspace Manager</td>
<td>2012 and higher</td>
</tr>
<tr>
<td>RES Workspace Manager Relay Server</td>
<td>2012 and higher, as support in Task <strong>Install Windows Installer Package</strong></td>
</tr>
<tr>
<td>RES IT Store (also known as RES ONE Service Store, RES ONE Identity Director)</td>
<td>2014 and higher</td>
</tr>
</tbody>
</table>
Chapter 10: Index

A
About RES ONE Automation • 4
Access permissions • 216
Active Directory Computer (Create, Manage, Delete, Query) • 89
Active Directory Group (Create, Manage, Delete, Query) • 90
Active Directory Object (Move, Query) • 92
Active Directory Organizational Unit (OU) (Create, Manage, Delete, Query) • 93
Active Directory User (Create, Manage, Delete, Query) • 95
Activity • 65
Adding or editing a Resource • 287, 289
Add-ons • 11, 34
Administration • 212
Administrative Roles • 213, 215, 216
Agent Cache settings • 36, 42
Agent settings • 194
Agent Trusts • 195
Agent Variables • 195
Agent+ • 135, 193, 202
Agents • 178, 193, 314
Agents for Apple Mac OS X • 135, 193, 204
Appendices • 274
Architecture • 6, 202
Assigning administrative roles to login accounts • 215, 219
Audit Trail • 224, 272
Automatic Job scheduling • 210

B
Bare Metal OS deployment using WDS and RES ONE Automation • 204
Bare Metal OS Deployment Using WDS and RES ONE Automation • 204, 274
Baseline Security (MBSA) (Query) • 99
Best Practices • 274
Building Blocks • 22, 177, 225, 226, 272

C
Certificate (Import, Delete, Query) • 100
Change my password • 214, 215
Changing Agent connection settings • 287, 291
Changing Dispatcher connection settings • 287, 291
Changing the security context password for all RES ONE Automation Tasks at once • 88
Citrix Published Applications (Query) • 100
Citrix Workflow (Invoke) • 101
Command (Execute) • 102
Command Bar • 225, 232
Command-line options • 221, 274, 287
Compatibility Matrix • 129, 274, 316
Components • 5, 221
Computer (Inventory) • 104
Computer (Reboot, Shutdown) • 105
Computer (Uptime) • 105
Computer Properties (Manage) • 106
Computer Properties (Query) • 107
Conditions • 62, 78, 82, 110, 155, 162, 225, 233
Conditions based on existence of file or folder • 234, 235
Conditions based on parameters • 234, 235
Conditions based on registry settings • 234, 236
Configure Protocol Encryption • 16, 19
Configuring RES ONE Workspace to use Windows Authentication using a Task • 141
Configuring Trusts • 271
Connection settings • 22, 35, 40
Consoles • 178
Creating a Building Block for a Run Book • 287, 292
Creating Building Blocks • 226
Creating parameters automatically • 263, 265
Creating RES ONE Identity Director services based on Run Books • 31, 160, 167
Cron expression • 54, 58
Customizing the contents and presentation • 256

D
Database • 11, 15
Database Statement (Execute, Query) • 107
Deleting the Job History of an Agent • 200
Detailed Job results • 67
Diacritics and their Conversion • 252
Disk Fragmentation (Defragment, Analyze) • 108
Disk Space (Query) • 109
Dispatcher Cache settings • 35, 41
Dispatcher Detection settings • 35, 38
Dispatcher Discovery • 313
Dispatcher Proxy settings • 28, 29, 35, 41
Dispatcher settings • 181
Dispatcher WebAPI settings • 35, 42, 188
Dispatchers • 35, 178, 180

E
E-mail (Send) • 110
Encryption using AES-256 • 15, 22, 226
Environment Variables (Query) • 111
Environment Variables (Set, Delete) • 110
Error control for Tasks • 87
Evaluators • 68, 78, 225, 237
Event logs • 111, 202, 309
Event Logs (Query, Write) • 111
Example Create Exchange Mailbox • 115
Exceptional behavior to the Job scheduling combination • 59
Exchange Mailbox (Create, Manage, Move, Export, Disable) • 95, 112
Exchange Mailbox (Query) • 116
Exporting Job Results • 287, 293
Exporting Job results immediately • 71
Exporting Job results to XML • 71, 102
Exporting licenses • 287, 294
Exporting the Job results of a previous Job using a generic command line • 71, 73
Exporting the Job results of a specific Job using a command line • 71
Exporting the Job results of an active Job using a generic command line • 71, 72

F
File Permissions, Printer Permissions, Registry Permissions, Share Permissions (Query) • 118
File Permissions, Printer Permissions, Registry Permissions, Share Permissions (Set) • 118
Files (Perform Operations) • 119
Files (Query) • 119
Flowcharts • 274, 313
Forcing an abort of a Job that is already in an aborting state • 65
Frequency at which Console checks for new updates • 305
Functions • 78, 109, 125, 172, 225, 246

G
General functionality • 225
Global Settings • 11, 35, 41, 42, 45, 50, 55, 59, 110, 163, 178, 217, 222

H
Hardware • 11, 12
Hosts (Discover) • 119
Housekeeping tasks of Dispatcher • 180, 181, 189

I
IBM DB2 • 15, 21
Importing a, RES ONE Workspace Building Block into a RES ONE Workspace environment using a Task • 104
Importing Building Blocks • 228, 287, 295
Information • 212
Input Mask • 259, 261
Input settings • 260
Install Agents embedded using Prepare for Image • 193, 197
Install Agents embedded using Prepared4Embedded • 197
Install Agents manually • 193, 222
Install Consoles manually • 221
Install database drivers using RES ONE Automation • 22
Install Dispatchers manually • 180, 221
Installation procedure • 11
Installed Programs (Query) • 120
Installing Agents • 193, 287, 296
Installing Consoles • 178, 298
Installing Dispatchers • 180, 222, 299
Installing RES ONE Automation • 300
Installing RES ONE Workspace using a Task • 150
Instant Report format • 255
Instant Reports • 225, 255
Introduction • 1

J
Job History • 66
Job Notification settings • 36, 45
Job Postpone • 120
Jobs • 52

L
LANDesk (Distribute Software) • 121
Launch window options • 55, 56
Library • 78
License server registration process • 24, 28
License usage collection and reporting • 24, 29, 190
Licensing • 8, 11, 23
Licensing process • 23, 26
Linking parameters • 165, 263, 264
Local Group (Create, Manage, Delete, Query) • 121
Local User (Create, Manage, Delete, Query) • 122
Location of Dispatcher Resource cache folder • 305
Logging • 274, 309
Login accounts • 48, 213, 215

M
Mac OS X Automator (Invoke) • 122, 204
Mac OS X Command (Execute) • 123, 124, 204
Mac OS X Computer (Reboot, Shutdown, Inventory, Uptime) • 124, 204
Mac OS X Disk Space (Query) • 125, 204
Mac OS X Installed Programs (Query) • 126, 204
Mac OS X Parameters (Query) • 124, 131, 146
Mac OS X Resource (Download) • 125, 204
Mac OS X Software Package (Install, Remove) • 126, 204
Managing Modules • 79
Managing Projects • 154
Managing Resources • 169
Managing Run Books • 160
Manually removing Agents from the Datastore and Console • 201
Master caching • 180, 181, 306
Master caching Dispatcher • 182, 305, 306
Message Box (Show) • 127
Microsoft App-V Client (Invoke) • 127
Microsoft App-V Client (Query) • 127
Microsoft Azure SQL • 15, 20
Microsoft SQL Server • 15, 16
Microsoft System Center Configuration Manager (Distribute Software) • 128
Microsoft System Center Configuration Manager (Query Client) • 129
Microsoft Update (Install) • 130
Microsoft Windows Product Key (Set, Query) • 130
Module Folders • 79, 80
Module Job History tab • 79, 83
Module Parameters tab • 79, 82
Module Permissions tab • 79, 85
Module Properties tab • 79, 81
Module Trusts • 79, 84
Module Usage tab • 79, 83
Module Versioning tab • 79, 83
Modules • 78
MySQL • 15, 21

Open a remote Console on an Agent • 45, 196
Oracle • 15, 20
Other Settings • 37, 50
Overview • 247

Parameter types • 259
Parameters • 53, 78, 79, 82, 109, 125, 154, 156, 165, 172, 225, 257
Parameters (Query) • 124, 131, 146
Parsing environment variables, parameters and functions • 172
Password Security Policy settings • 36, 48, 213, 214, 215, 288
Pattern matching, wildcards and operators • 225, 234, 238, 267
Performing Actions using Pattern Matching on Registry Key Names • 133
Permissions and prerequisites to create, manage, move or disable Exchange mailboxes • 113
Permissions and prerequisites to export Exchange mailboxes • 114
Permissions and prerequisites to query Exchange mailboxes • 117
PoSh Automation • 32
PostPE.xml • 283
PowerShell Group Policies (GPOs) • 153

Pre-2014 Licensing Model • 23, 25
Prerequisites • 12
Printer (Add, Remove, Query) • 131
Printer Driver (Add, Remove, Query) • 132
Project Current Resulting Tasks tab • 154, 156
Project Job History tab • 154, 158
Project Modules tab • 154, 155
Project Parameters tab • 154, 156
Project Permissions tab • 154, 158
Project Properties tab • 154, 155
Project Usage tab • 154, 157
Project Versioning tab • 154, 158
Projects • 78, 154

R
Reapplying the Job History of Agents and Teams • 69, 70, 169
Re-deploying an existing machine • 278
Registry settings • 274, 305
Registry Settings (Apply) • 132
Registry Settings (Query) • 134
Remote Terminal Server Logons (Change, Query) • 134
Reporting to RES ONE License Server • 28, 180, 181, 190
RES ONE Automation Agents for Unix/Linux • 135, 193, 203
RES ONE Automation Component (Deploy, Repair, Remove, Update) • 135
RES ONE Automation configuration • 277
RES ONE Automation Dispatchers (Discover) • 136
RES ONE Automation Glossary • 274, 314
RES ONE Automation Results (Delete, Export) • 71, 137
RES ONE Automation Team Membership (Change) • 138
RES ONE Identity Director • 11, 31
RES ONE Workspace (Refresh) • 135
RES Support • 3
RES Training • 2
Rescheduling a Job • 70
Resource (Download) • 139
Resource Contents tab • 169, 173
Resource Folders • 169, 170
Resource Permissions tab • 170, 175
Resource Properties tab • 169, 172
Resource tab • 169, 171
Resource Trusts tab • 170, 174
Resource Usage tab • 169, 173
Resource Versioning tab • 170, 174
Resources • 78, 169
Run Book Job History tab • 160, 166
Run Book Jobs tab • 160, 161
Run Book Parameters tab • 160, 165
Run Book Permissions tab • 160, 167
Run Book Properties tab • 160, 161
Run Book Versioning tab • 160, 166
Run Books • 45, 78, 159
S

Saving components as MSI files • 178, 221
Saving the Job results of a Query Task as a
TXT file or a CSV file • 68, 74
Scenario
  Add Active Directory users to Distribution
  Lists • 237, 241
  Install applications only on Agents with
  sufficient disk space • 237, 239
Scenarios • 159
Schedule Job in parallel • 53, 55, 63
Scheduling • 52
Scheduling a Job
  When, What, Who • 54
Scheduling in the Management Portal • 75
Scheduling Jobs • 53, 287, 303
Scheduling settings • 36, 43, 53, 54, 55, 163
Secure delegation • 220
Secure Shell (SSH) Commands (Execute) • 139
Scenarios • 159
Securing Modules, Projects and Run Books
using parameters • 266
Security • 80, 170, 205, 213
Security settings • 36, 45, 271
Service Properties (Manage) • 141
Service Properties (Query) • 142
Settings • 234, 238
Setup • 11
Share (Add, Remove, Query) • 142
Site settings • 37
Snapshot Intelligence • 47, 67, 169, 198
Snapshot Intelligence settings • 36, 47
SNMP trap (Send) • 143
Software • 11, 12
Solving a Console lockout • 215, 287, 304
Special permissions • 55, 217
Specify alternative ports for Microsoft SQL
  Server and IBM DB2 • 16, 19, 21
SQL Server authentication • 16, 17
Supported Application Virtualization • 319
Supported Citrix • 321
Supported Database systems • 319
Supported Directory Services • 318
Supported Hypervisors • 320
Supported Mail Servers • 319
Supported Microsoft Windows operating
  systems • 316
Supported Microsoft Windows PowerShell
  Scripting • 320
Supported Provisioning Services • 320
Supported RES products • 321
Supported Unix/Linux operating systems • 318

T

Task Security context • 88
Task Visibility • 11, 34
Tasks • 79, 81, 85, 173, 175
Tasks specific for Add-ons • 86
Tasks tab • 79, 81
TCP/IP Properties (Query) • 143
Team Folders • 205
Team members • 208
Team Permissions • 211
Team rules • 209
Team settings • 207
Team Trusts • 211
Team Variables • 207
Teams • 83, 157, 178, 194, 205, 270
Telemetry • 312
Telemetry tasks of Dispatcher • 180, 191, 312
The structure of Building Blocks • 231
The Unattend files • 279
Topology • 178
Tracing • 305, 307
Trust permissions • 216, 217, 271
Trusts • 46, 79, 80, 84, 170, 174, 175, 194,
  195, 196, 205, 206, 208, 211, 225, 269

U

Unattended Installation (Perform) • 144
Unix/Linux Command (Execute) • 145, 146,
  203
Unix/Linux Computer (Reboot, Shutdown) •
  145, 146, 203
Unix/Linux Parameters (Query) • 124, 131,
  146
Unix/Linux Resource (Download) • 146, 203
Upgrading your RES ONE Automation
  environment • 223
User Management (Create, Delete, Lock,
  Unlock, Change Password, Query) • 147,
  203
Using environment variables in a Task • 98
Using HTTPS • 183, 188
Using parameter values from a CSV file • 53,
  61, 164, 235, 259
Using parameter values to skip or execute
  Modules, Projects, and Jobs • 235
Using Wake-on-LAN • 54, 59
Using Windows variables in e-mail addresses •
  95, 98, 112

V

Variables • 78, 176, 195, 207
Version Control • 48, 79, 154, 160, 169, 228,
  272
Version Control settings • 36, 48, 79, 83, 154,
  158, 166, 174, 272
Video Tutorials • 9
Viewing Job results • 66

W

WDS Configuration • 275
Web API • 180, 183
Web Service (Invoke) • 147
What • 53, 60
When • 53, 54
Where to find Activity • 65
Where to find Instant Reports in the Console • 255
Where to find Job results • 66
Where to find parameters • 258
Where to find Scheduling • 52
Where to find Trusts in the Console • 270
Who • 53, 60
Windows authentication • 16, 17
Windows Firewall Settings (Manage, Query) • 148
Windows Installer Package (Install, Repair, Patch, Remove) • 149
Windows PowerShell Script (Execute) • 151
Windows Shell (Set, Query) • 153
WinPE.xml for Windows 7 • 279
WinPEsrv.xml for Windows 2008 • 281