

Getting Started with

RES ONE®

Automation

Version 10.2.0.0

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	PoSh Automation	9
Chapter 3:	Installation	10
3.1	Planning	10
3.2	Prerequisites	14
3.2.1	Hardware.....	14
3.2.2	Database	15
3.2.3	Software.....	21
3.3	Installation procedure	24
3.3.1	Install the Console.....	25
3.3.2	Connect to a Datastore	26
3.3.3	Install the Management Portal and connect to existing Datastore	27
3.3.4	Install Dispatchers.....	29
3.3.5	Install Agents.....	30
3.3.6	Install additional Consoles	31
Chapter 4:	Next steps	32
4.1	Create a Resource.....	33
4.2	Create a Module	34
4.3	Schedule a Job.....	36
4.4	View the Job results	37
Chapter 5:	Where to go from here?	40
Chapter 6:	RES ONE Automation Glossary	41
Chapter 7:	Compatibility Matrix	43
7.1	Supported Microsoft Windows operating systems	43
7.2	Supported Unix/Linux operating systems	45
7.3	Supported Directory Services	45
7.4	Supported Database systems	46
7.5	Supported Mail Servers	46
7.6	Supported Application Virtualization	47
7.7	Supported Hypervisors.....	47
7.8	Supported Provisioning Services	47
7.9	Supported Microsoft Windows PowerShell Scripting	47
7.10	Supported Citrix "Ready" products	48

7.11	Supported RES products.....	48
------	-----------------------------	----

Chapter 1: Introduction

This document guides you through the installation steps and configuration essentials to set up a RES ONE Automation sandbox environment.

More information

You can find more information about RES ONE Automation at the RES Success Center, at <http://success.res.com> > **Downloads**. Useful reading includes:

- The **Introduction to RES ONE Automation** explains the concepts behind RES ONE Automation.
- The **Release Notes** provide information on new and changed functionality.
- The **Help** provides user assistance.
 - In the Management Portal, click the question mark in the menu bar.
 - In the Setup and Sync Tool, press F1.
- The **RES ONE Automation Upgrade Guide** describes how to upgrade an existing RES ONE Automation environment.
- The **RES ONE Automation Administration Guide** provides more detailed information. This document is available at <http://help.res.com/AutomationAdminGuide10>. Alternatively, you can download a PDF version at the RES Success Center.

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.

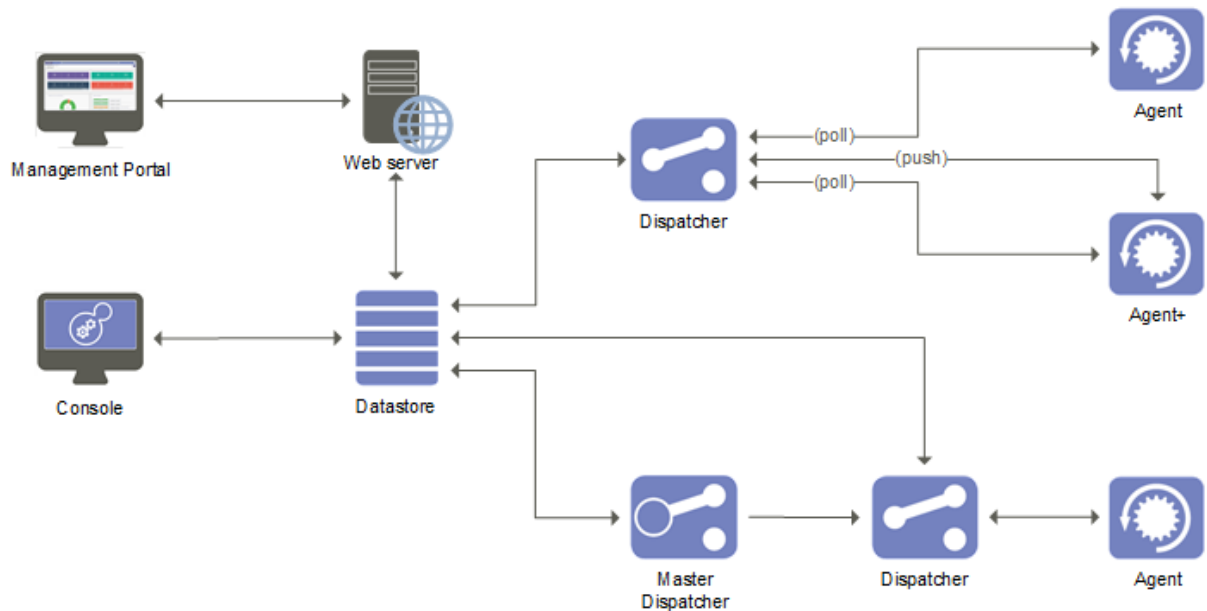
Agent

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:



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:

TCP Port	Description
443	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.
<vendor default>	<p>The following components connect to the Datastore:</p> <ul style="list-style-type: none"> • Console • Management Portal via IIS • Dispatcher <p>Microsoft SQL Server, Microsoft Azure SQL, Oracle, MySQL and IBM DB2 use different default ports. See the vendor documentation for the correct port numbers.</p>
3163	<p>Used for poll communication between Dispatchers and Agent(s) or Agent(s)+. This port is fixed and <i>cannot</i> be changed.</p> <ul style="list-style-type: none"> • 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 <i>can</i> 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

License requirements

RES ONE Automation requires licenses for each Agent. Only licensed Agents can execute Jobs.

Agent type	Number of required licenses
Windows client	0.5 (i.e. 2 Windows clients require 1 license, 3 Windows clients require 2 licenses).
Mac OS X	0.5
Windows server (including terminal servers)	6
Unix/Linux	6

License management

When you install RES ONE Automation, 75 evaluation licenses are available. These are valid for 45 days. To use RES ONE Automation after the evaluation period, you need production licenses. The production licenses need to be registered and activated before you can use them.

- You can buy production licenses at your reseller.
- If necessary, contact backoffice@res.com to extend the evaluation period.
- Evaluation licenses offer the same functionality as production licenses.
- You will receive additional RES ONE Identity Director licenses when the RES ONE Identity Director integration is enabled and validated.

Alternatively, you can run RES ONE Automation in **PoSh Automation** (on page 9) mode.

License distribution

Licenses are distributed based on the deployment date of Agents. When you remove Agents, the spare licenses are redistributed automatically.

2.4 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.

Chapter 3: Installation

3.1 Planning

Before you install RES ONE Automation components, it is important to **plan** your RES ONE Automation environment very extensively.

A RES ONE Automation environment is highly-flexible and can easily be adapted to your needs, whether your environment consists of 5 managed devices or 100,000. When you plan a RES ONE Automation environment, it is important to know when to deploy additional **Dispatchers**. No additional configuration is required to use RES ONE Automation in multiple domain environments.

In general, the following rules of thumb apply to RES ONE Automation environments:

Datastore:

- A RES ONE Automation environment usually contains one **Datastore**. This Datastore exists on a database server (Microsoft SQL Server, Microsoft Azure SQL, MySQL, IBM DB2 or Oracle). If desired, the Datastore can be made redundant by clustering it.

Dispatcher:

- An environment requires at least one **Dispatcher**. In a small environment, a second Dispatcher would generate unnecessary Datastore <-> Dispatcher traffic. As a rule of thumb, **one Dispatcher can handle at least 1500 concurrent Agents**.
- You can deploy a Dispatcher on any Microsoft Windows device in your network, but we recommend not to install Datastores, Dispatchers and Agents on the same device. Although a Dispatcher installation has a very small footprint, a Dispatcher **caches all Resources** and therefore needs sufficient disk space: if you would install a Dispatcher on the same device as the Datastore, Resource caching would double the amount of data on that device.
- We recommend not to use a terminal server as a Dispatcher, because of the network traffic it generates.
- The number of Dispatchers that are required also depends on the nature of the Tasks that the Agents need to execute, and on the hardware and infrastructure used: rolling out a Microsoft Office 2016 installation on 400 Agents obviously puts a higher load on a Dispatcher than installing a small Microsoft Hotfix on 10 Agents. In the **Dispatchers** node of the Console, the columns **Connections (poll | push)** and **Peak (poll | push)** show the number of current connections and the highest number of connections for each Dispatcher, separated for the poll and push communication to the Agent and Agent+. This can be a good indication of when to deploy an additional Dispatcher. You can always add other Dispatchers at a later stage.
- If a remote environment contains more than one Dispatcher, it should have a **master Dispatcher** in order to limit the amount of data traffic between the Datastore and the Dispatchers over WANs. See the Help for more information about master caching Dispatchers. The Help is available in the Console after installation, by pressing **F1**.

When you plan a RES ONE Automation environment, keep in mind that you can specify the following settings for the communication between the different components. You can specify the defaults for these settings on a global level and set exceptions per Dispatcher, Agent or Team. If necessary, you can change both global and individual settings after installation. You can find these settings in the Console at **Setup > Global Settings**.

Dispatcher detection settings

Item	Function
Dispatcher discovery	Specifies the Agents' search method for Dispatchers. Agents can discover Dispatchers by sending a multicast signal or by using a Dispatcher address list.
Dispatcher locations	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.
Dispatcher recovery	Specifies which action Agents should take if no Dispatchers can be contacted. The Agent can keep trying to discover Dispatchers or revert to a cached list of ALL known Dispatchers. This setting can be configured on global level, on Team level and on Agent level.

The settings **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.

For LAN connections, we recommend the following combination:

- **Autodetect and Only Use Discovered Dispatchers.** If you use this combination, the Agent can only discover Dispatchers in its own network segment. The Agent detects its Dispatchers by sending a multicast signal. A multicast signal cannot travel over routers. The Agent will only communicate with the Dispatchers that are detected, and randomly use these Dispatchers to retrieve information.

For WAN connections, we recommend the following combination:

- **Use Dispatcher Address List and Only Use Discovered Dispatchers.** If you use this combination, the Agent will not send a multicast signal, but only detect Dispatchers that are specified in the Dispatcher address list. Specify at least one Dispatcher in the 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 the Agent is able to communicate with these Dispatchers over IP port 3163, it will connect to these Dispatchers at random.

Connection settings

Item	Function
Bandwidth throttling	Controls the amount of data that Agents can download per second from a Dispatcher, and applies per Agent per Resource. If you configure this setting both at Agent and Dispatcher level, the bandwidth is determined by the lowest value. If you disable this setting, the maximum download speed is used, depending on your network topology, available CPU performance and number of concurrent downloads.
Resource Download throttling	Controls the amount of data that Dispatchers can download per second, per Resource when they get Resources 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 if communication between Dispatchers and Agents is secured. <ul style="list-style-type: none"> • This setting functions irrespective of database type used. • Any changes to this setting take effect immediately.

Example scenario: The Demo Company

The following scenario gives an example of how you can set up a RES ONE Automation environment and adapt it as circumstances change. The examples are based on the fictional company **The Demo Company**, which initially consisted of a single office in New York, where 50 people were employed. Over the years, The Demo Company expanded and its number of employees in New York increased from 50 to 2000. Furthermore, a satellite office in Singapore was opened with 20 employees.

Initial setup in New York:



New York
50 Agents
1 Dispatcher
SQL Server: SQL-NY
Datastore: RAMNY

- One Datastore (Microsoft SQL Server)
- One Dispatcher (only 50 Agents: one Dispatcher suffices. A 2nd Dispatcher would cause unnecessary Dispatcher <-> Datastore traffic)
- 50 Agents

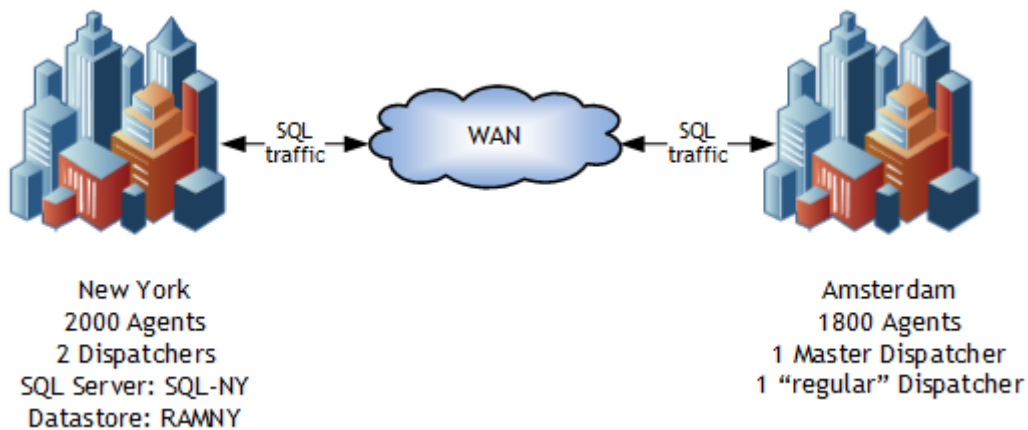
After growth in New York:



New York
2000 Agents
2 Dispatchers
SQL Server: SQL-NY
Datastore: RAMNY

- One Datastore
- Two Dispatchers (1500 Agents per Dispatcher: a reasonable number, though depending on the number of concurrent connections)
- 2000 Agents
- The additional Dispatcher and Agents have been deployed from the Console

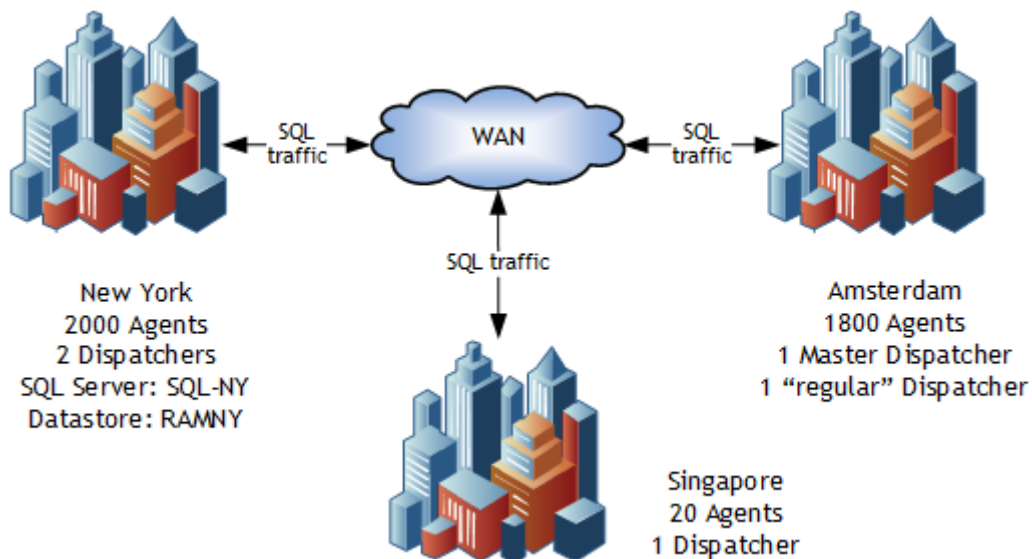
When the office in Amsterdam was opened:



The office in Amsterdam has:

- One master caching Dispatcher that connects to the Datastore RAMNY in New York.
- One Dispatcher that connects to the master caching Dispatcher (one Dispatcher suffices).
- 1800 Agents that only connect to the "regular" Dispatcher in Amsterdam, not the master caching Dispatcher (Their Dispatcher Address List only contains the "regular" Dispatcher in Amsterdam, and they do not retrieve a complete Dispatcher List).
- All Dispatchers and Agents have been deployed from the Console in Amsterdam.

When the satellite office in Singapore was opened:



The office in Singapore has:

- One Dispatcher. This Dispatcher was deployed from New York and connects to the existing Datastore in New York.
- 20 Agents, which use the Dispatcher in Singapore. (Their Dispatcher Address List only contains the Dispatcher in Singapore, and they do not retrieve a complete Dispatcher List.)

It is very easy to administer a RES ONE Automation environment from a single point. Because of RES ONE Automation's scalability, it is just as easy to maintain an environment with 2000+ Agents as it is to maintain one with just 50 Agents.

3.2 Prerequisites

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

3.2.1 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.

3.2.2 Database

RES ONE Automation supports the following database types:

- **Microsoft SQL Server** (on page 15)
- **Microsoft Azure SQL** (on page 18)
- **MySQL** (on page 19)
- **Oracle** (on page 18)
- **IBM DB2** (on page 19)

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 17) 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 15) by default. You can also use **Windows Authentication** (see page 16).
- You can configure Microsoft SQL Server and IBM DB2 to **use a different port** (see page 20) 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.
 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.

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.

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)

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)

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 20)

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)

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.

3.2.3 Software

See the **Compatibility Matrix** (on page 43) 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

Software installed on system	Prerequisites
RES ONE Automation	<ul style="list-style-type: none"> • 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, 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. Supported Microsoft Windows operating systems (on page 43)
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 46)
Browser	Supported browsers: <ul style="list-style-type: none"> • Microsoft Internet Explorer 10 or higher • Microsoft Edge 15 or higher • Mozilla Firefox 47 or higher • Google Chrome 52 or higher • Opera 12 or higher • Safari 9.0 (Mac only) or higher JavaScript enabled

Web server	Prerequisites
Permissions	Administrator permissions
Internet Information Services (IIS)	IIS 7.5 or higher

Microsoft Windows features installed on Web server	Necessary features for Microsoft Windows Server 2008 R2	Necessary features for Microsoft Windows Server 2012 / 2012 R2
Microsoft .NET Framework	<ul style="list-style-type: none"> Version 4.5.2 or higher To support Microsoft Internet Explorer 10, install all available Microsoft Windows updates for .NET Framework 4.5.2 first. 	<ul style="list-style-type: none"> Version 4.5.2 or higher ASP.NET 4.5
Web Server (IIS)		
Web Server	<ul style="list-style-type: none"> Common HTTP Features Static Content Default Document HTTP Errors HTTP Redirection 	<ul style="list-style-type: none"> Common HTTP Features Static Content Default Document HTTP Errors HTTP Redirection
Application Development	<ul style="list-style-type: none"> ASP.NET .NET Extensibility ISAPI Extensions ISAPI Filters 	<ul style="list-style-type: none"> ASP.NET 4.5 .NET Extensibility 4.5 ISAPI Extensions ISAPI Filters
Health and Diagnostics	<ul style="list-style-type: none"> HTTP Logging Request Monitor 	<ul style="list-style-type: none"> HTTP Logging Request Monitor
Security	<ul style="list-style-type: none"> Basic Authentication Windows Authentication Client Certificate Mapping Authentication IIS Client Certificate Mapping Authentication Request Filtering 	<ul style="list-style-type: none"> Basic Authentication Windows Authentication Client Certificate Mapping Authentication IIS Client Certificate Mapping Authentication Request Filtering
Performance	<ul style="list-style-type: none"> Static Content Compression Dynamic Content Compression 	<ul style="list-style-type: none"> Static Content Compression Dynamic Content Compression
Management Tools	<ul style="list-style-type: none"> IIS Management Console Management Service 	<ul style="list-style-type: none"> IIS Management Console

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.

3.3 Installation procedure

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 (please refer to the **Help** for more information):
 - 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.

After you install RES ONE Automation, you can also install additional RES ONE Connectors. These RES ONE Connectors contain one or more Add-ons to integrate with third party software. See the **Getting Started with RES ONE Connectors**, available at <http://success.res.com>.



Notes

- RES ONE Automation contains Third Party Software components that are not digitally signed.
- RES installation files are signed with certificates. Microsoft Windows tries to verify a certificate's validity before installing software products. This process (Certificate Revocation List (CRL)) is run to check to see if a certificate was revoked because it was expired or compromised. This process is not unique to RES and is something that happens for any product that contains a certificate and is run on Microsoft Windows (unless CRL is turned off which is not recommended by Microsoft). It is also possible under specific cases that a CRL check is done during process startup.
 - On computers without Internet access, the CRL validation may cause a delay of, for example, 20-30 seconds before an installation starts. This is by design of Microsoft Windows. To avoid this delay to occur, make sure machines can connect to the Internet. If this is not possible, implement a (manual/automatic) distribution system to keep the publisher's certificate revocation lists up to date.

3.3.1 Install the Console

The first step in configuring a RES ONE Automation environment is to install the Console. You can install the Console on any computer that can connect to the database server that hosts or will host the Datastore. It is usually installed on the administrator's workstation.

To install RES ONE Automation, use the RES ONE Automation Installer. This is an installation package that contains the MSI files for the components of RES ONE Automation, wrapped up into one executable. The RES ONE Automation Installer supports both 64-bit and 32-bit operating systems for all components of RES ONE Automation. This makes their installation easier.

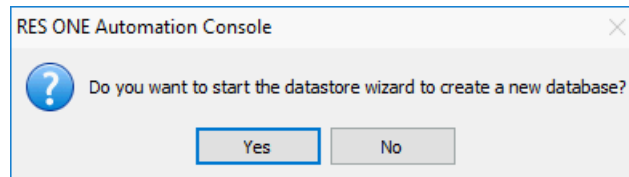
1. Log on to this computer with a valid System Administrator account. This computer must be able to connect to the database server.
2. Run the RES ONE Automation Installer on the computer on which you want to install the Console. This opens the **RES ONE Automation Installer** window.
3. When prompted, select **Select and install components**, select **RES ONE Automation** and click **Next**.
 - If you select **Extract all components** you can perform unattended installations of the components by using the MSI files.



4. Start the installation and follow the Setup Wizard prompts. By default, the Console will be installed at `C:\Program Files (x86)\RES\Automation\Console` for 64-bit devices and `C:\Program Files (x86)\RES\Automation\Console` for 32-bit devices.
5. At the end of the Setup Wizard, select **Launch RES ONE Automation**. This will launch the Console when you click **Finish**. Alternatively, you can launch the Console from the Start menu and from the desktop icon.

3.3.2 Connect to a Datastore

The next step is to create a new Datastore or connect to an existing one. After a fresh installation of RES ONE Automation (no Datastore connection is available yet), you can start the Datastore Wizard directly when you start the Console.



1. Click **Yes** to start the Datastore Wizard. The Datastore Wizard helps you to create a new Datastore for RES ONE Automation.
 - If you click **No**, the Console will start, but only the node **Setup > Database** will be available. You can then choose to connect to an existing Datastore or create a new one. All other nodes will become available when you create a new Datastore or connect to an existing one.
2. In the Datastore Wizard, click **Next** and select the type of database server that will host the Datastore.
 - On Microsoft SQL Server, RES ONE Automation will create a Datastore through the Datastore Wizard.
 - On Oracle, RES ONE Automation will create a database user and table space with the names specified in the Datastore Wizard. Alternatively, you can select an existing table space for the new database user.
 - On MySQL, RES ONE Automation will create a database (or schema) and a database user with the names specified in the Datastore Wizard.
 - On IBM DB2, you cannot create a database with the Datastore Wizard: the database and database user must already exist.
3. Specify the name of the relevant database server. For Microsoft SQL Server, RES ONE Automation will automatically show a list of available SQL servers.
4. If the Datastore to which you want to connect is based on Microsoft SQL Server, specify whether this server uses **Windows Authentication** or **SQL Server authentication**.
5. Specify the user name and password of the DBA account that has access to the database.
6. If the Datastore to which you want to connect is based on Microsoft SQL Server, you can optionally select to use an **encrypted** connection to secure communication between the Datastore and the components that connect to it.
7. Click **Next** and specify a name for the Datastore. For Oracle, you can also specify an existing table space.
8. Click **Next**.
9. If necessary, change the database settings and click **Next**.
10. Create a login account that allows the Console to access the new Datastore.
11. Click **Next** and then **Finish**. After the Datastore has been created, the Console restarts automatically and connects to the new Datastore. All nodes will be available, except for the **Jobs** node.

3.3.3 *Install the Management Portal and connect to existing Datastore*

The Management Portal is the web-based application for scheduling Jobs in your RES ONE Automation environment. The Management Portal can be installed alongside the Console.

Install the Management Portal

The first step is to install the RES ONE Automation Management Portal. You need to install the Management Portal on a web server. We do not recommend installing it on a domain controller.


- The installation will create the following branch in the IIS Manager: **RES > Automation**.
- For evaluation purposes, you can also install the Management Portal on a workstation.
- Because users can access the Management Portal from any location, it is best practice to secure access to it with administrative roles. If you do not, ALL users with an Active Directory account will be able to access it.

Procedure

1. Run the RES ONE Automation Installer and choose **Select and install components > RES ONE Automation Management Portal**, or run the RES ONE Automation Management Portal Installer separately. Launch it on the computer on which you want to install the Management Portal, for example your web server. This opens the **RES ONE Automation Management Portal Setup Wizard** window.
2. Follow the steps of the Setup Wizard.
3. By default, the Management Portal will be installed at `C:\Program Files\RES\Automation`. If you install the Management Portal at a custom location, the installation path can contain a maximum of 120 characters.
4. In the **Configure Secure Binding** step, specify the settings and select a server certificate to create an SSL binding for the Management Portal.
 - The hostname must be known on your internal and external DNS servers.
 - The Management Portal is secured with SSL by default. It uses the HTTPS protocol and port 443 (or another port that you specified).
 - With the option **Select installed certificate**, you can select a server certificate from a list that is populated with computer certificates from the **Personal** and **Web Hosting Certificate Stores**.
 - For testing purposes, the option **Generate Self-Signed Certificate** is offered on Microsoft Windows 8 and higher / Microsoft Windows Server 2012 and higher. RES recommends not to use self-signed certificates in a production environment. The self-signed certificates require PowerShell 4.0 or higher.
5. In IIS, the installation creates the node **RES** and deploys the Management Portal as the web application **RES > Automation**. If you, for instance, install the Management Portal with a host name (alias) *resproducts*, the web application will be available at `https://resproducts.mydomain.com/automation`.
 - If the node **RES** already exists in IIS, this step is skipped and the actual installation begins.
6. At the end of the Setup Wizard, click **Finish**.

After the installation is finished, the RES ONE Automation Management Portal is installed and a shortcut to the URL is placed on your desktop to start the Management Portal. Alternatively, you can open the Management Portal via a browser and specify the web address including the path `"/automation"`.

When launching the Management Portal, it opens the **Setup > Environment** page to create a Datastore connection.

 **Notes**

- RES installation files are signed with certificates. Microsoft Windows tries to verify a certificate's validity before installing software products. This process (Certificate Revocation List (CRL)) is run to check to see if a certificate was revoked because it was expired or compromised. This process is not unique to RES and is something that happens for any product that contains a certificate and is run on Microsoft Windows (unless CRL is turned off which is not recommended by Microsoft). It is also possible under specific cases that a CRL check is done during process startup. A sign-in could potentially be delayed if the environment does not use a Corporate/Enterprise license and the user attempts to sign in immediately after the restart of the computer or the service, before the CRL check has been done.

On computers without Internet access, the CRL validation may cause a delay of, for example, 20-30 seconds before an installation starts. This is by design of Microsoft Windows. To avoid this delay to occur, make sure machines can connect to the Internet. If this is not possible, implement a (manual/automatic) distribution system to keep the publisher's certificate revocation lists up to date.

- If you want to use single sign-on, the IIS server must be located in the same domain as the people that use the RES ONE Automation Management Portal, or a trust must exist between multiple domains. Single sign-on requires the Windows feature Windows Authentication to be installed.

Connect the Management Portal to an existing Datastore

The next step is to connect to an existing Datastore. This Datastore stores all information related to RES ONE Automation. After a fresh installation of the RES ONE Automation Management Portal (no Datastore connection is available yet), the **Setup > Environment** page is automatically opened.

- In the Management Portal at **Setup > Environment**, specify the following information:
 - Database type:** the type of database server to which you want to connect.
 - Database server:** the name of the relevant database server.
 - Login/password:** the user name and password of the DBA account that has access to the database. When using Microsoft SQL Server and MySQL, it is possible to use a password containing special characters. The following special characters are escaped in the connection string: " ' ; { }
 - Database name:** the name for the Datastore.
- In the **Authentication** section, specify which authentication type should be used.
 - Windows:** Authentication happens via the accounts configured in the RES ONE Automation Console. Only valid login accounts using Microsoft Windows Authentication or RES ONE Automation Authentication are able to access the Management Portal. Users with a domain (domain\user or user@domain) and local account are allowed to log in.
 - Identity Broker:** Authentication is centralized via the Identity Broker. This component is aimed to 'broker' the authentication between the RES web application and the configured identity provider. See also **Getting Started with RES Identity Broker** for further information on installation and configuration.
- Click **Test connection**. If the database connection is established, a message will be shown.
- Click **Save** to connect to the Datastore using the specified configuration.


After the Datastore connection is set up, the Management Portal restarts automatically and connects to the Datastore.

3.3.4 Install Dispatchers

After you installed the Console and connected to a Datastore, you need to deploy your Dispatchers. Dispatchers handle all communication between the Datastore and the Agents. Dispatchers also cache all Resources from the Datastore, thereby lightening the load on the Datastore.

- The Datastore contains a 32-bit and a 64-bit version of the Dispatcher installation file.
- When you deploy a Dispatcher from the Console, RES ONE Automation will automatically use the correct installation file for 32-bit and 64-bit Microsoft Windows versions.
- The default installation folder of Dispatchers is `C:\Program Files\RES\Automation\Dispatcher`.
- Dispatchers can also be installed manually by saving the component as an MSI file (at **Administration > Components**) or unattended via the command line. Please refer to the **Help (F1)** or **RES ONE Automation Administration Guide**.

Deploy a Dispatcher

1. Open the Console and go to **Topology > Dispatchers**.
2. Click **New** in the command bar. This opens the **Push Deployment** window.
3. Select the target computer(s). You can do this by clicking  to browse your network, but you can also directly specify the host name, FQDN, or IP-address in the **Target Computer(s)** field.
4. If you select computers by browsing, select the target computer. You can make multiple selections. Alternatively, you can choose to scan all computers in a specific IP-range.
5. Click **Deploy now**.
6. Specify the **user name** and **password** for the computer(s) on which the Dispatcher is deployed. This user account requires administrative privileges, because it must have installation permissions.

You can view the deployment progression in the **Push Deployment** window. When deploying, the system will check if the correct operating system and/or framework is running on the relevant machine. If not, the installation process will stop and a message is shown.

If a deployment is successful, it shows **100%** in the column **completed successfully**. When the deployed Dispatcher starts, it will be shown in the **Dispatchers** node of the Console. 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.



Warning


The deployment of Dispatchers on 64-bit devices requires the installation of the 64-bit version of the necessary database drivers on these devices. The other components of RES ONE Automation use the 32-bit version of these database drivers. If it is not possible to use both versions on the same 64-bit device simultaneously, it is not possible to use a Console and a Dispatcher on this device that both point to the same Datastore.

3.3.5 Install Agents

When you have deployed all other RES ONE Automation components, you need to deploy the **Agents**. An Agent is a service that actually performs the scheduled Tasks in RES ONE Automation. The procedure is similar to deploying Dispatchers, with the exception of the **Agent settings**. If you deploy an Agent+ on a computer on which 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.

- When you deploy an Agent+ from the Console, RES ONE Automation will automatically use the correct installation file for 32-bit and 64-bit Microsoft Windows versions.
- The default installation folder for new installations of Agents is `C:\Program Files\RES\Automation\Agent`.
- Agents can also be installed manually by saving the component as an MSI file (at **Administration > Components**), unattended via the command line or embedded using **Prepare for Image**. Please refer to the **Help (F1)** or **RES ONE Automation Administration Guide**.

Deploy an Agent+

1. Open the Console and go to **Topology > Agents**.
2. Click **New** in the command bar. This opens the **Push Deployment** window.
3. Select the target computer(s). You can do this by clicking  to browse your network, but you can also directly specify the host name, FQDN, or IP-address in the **Target Computer(s)** field.
4. If you select computers by browsing, select the target computer. You can make multiple selections. Alternatively, you can choose to scan all computers in a specific IP-range.
5. Click the **Agent settings** tab to configure the **Dispatcher detection settings** for the Agent(s). After deployment, the Agent will use these settings to detect Dispatchers. If necessary, you can change these settings again at a later stage. If you do not set Agent-specific settings, the Agent will be deployed with the default global settings.
6. Click **Deploy now**.
7. Specify the **user name** and **password** for the computer on which the Agent is deployed. This user account requires administrative privileges, because it must have installation permissions.

You can view the deployment progression in the **Push Deployment** window. When deploying, the system will check if the correct operating system and/or framework is running on the relevant machine. If not, the installation process will stop and a message is shown.

If a deployment is successful, it shows **100%** in the column **completed successfully**. When you have deployed Agents, you can find the Agents and its current file version in the **Agents** node.



Notes


- 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 Apple Mac OS X and Agents for Unix/Linux cannot be deployed from the Console, but need to be installed separately.

3.3.6 *Install additional Consoles*

With the installation of RES ONE Automation, a Console was installed. If necessary, you can deploy additional Consoles, to create multiple points of administration of your RES ONE Automation environment. From these Consoles the same RES ONE Automation environment can be monitored and used. It is also possible to deploy Consoles to other computers and use them to set up an entirely new RES ONE Automation environment.

- Additional Consoles can also be installed manually or unattended.
- The default installation folder of Consoles is `C:\Program Files (x86)\RES\Automation\Console` for 64-bit machines and `C:\Program Files\RES\Automation\Agent` for 32-bit machines.

Install an additional Console

1. Open an existing Console and go to **Topology > Consoles**.
2. Click **New** in the command bar. This opens the **Push Deployment** window.
3. Select the target computer(s). You can do this by clicking  to browse your network, but you can also directly enter the host name, the fully-qualified domain name (FQDN), or the IP-address in the **Target Computer(s)** field.
4. If you select computers by browsing, select the computer on which you want to deploy an Agent. You can make multiple selections.
5. Alternatively, you can choose to scan all computers in a specific IP-range to narrow down the selection.
6. Click **Deploy now** to start the actual deployment. When the deployment process starts, you need to enter the **user name** and **password** for the computer on which the Console is deployed. This user account requires administrative privileges, because it must have installation permissions.

You can view the deployment progression in the **Push Deployment** window.

If a deployment is successful, it shows **100%** in the column **completed successfully**. When the deployed Console starts, it becomes visible in the **Consoles** node.



Note

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).

Chapter 4: Next steps

Now that you have set up a basic RES ONE Automation environment, you can start using it.

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, restart services on Agents or set registry keys, but also query the installed programs on Agents or query Windows event logs. RES ONE Automation contains a library full of Tasks that you can configure for a wide range of actions. Some Tasks are part of an Add-on and can be installed via **RES ONE Connectors**.

To give you a few examples, this document covers the execution of an MBSA query on an Agent. As a prerequisite, you need to install the Microsoft Baseline Security Analyzer first. This scenario consists of:


- **Create a Resource.** Resources are files that are used in the execution of Tasks, such as MSI files, setup files, patches, hotfixes, etc.
- **Create a Module.** A Module is a container for one or more Tasks. To execute the Tasks in a Module, you can schedule the Module as a Job on a set of Agents at a specific time.
- **Schedule a Job.** A Job is the execution of Tasks at a specific time on specific Agents.
- **View the Job results.** After a Job has been executed, you can view its results.

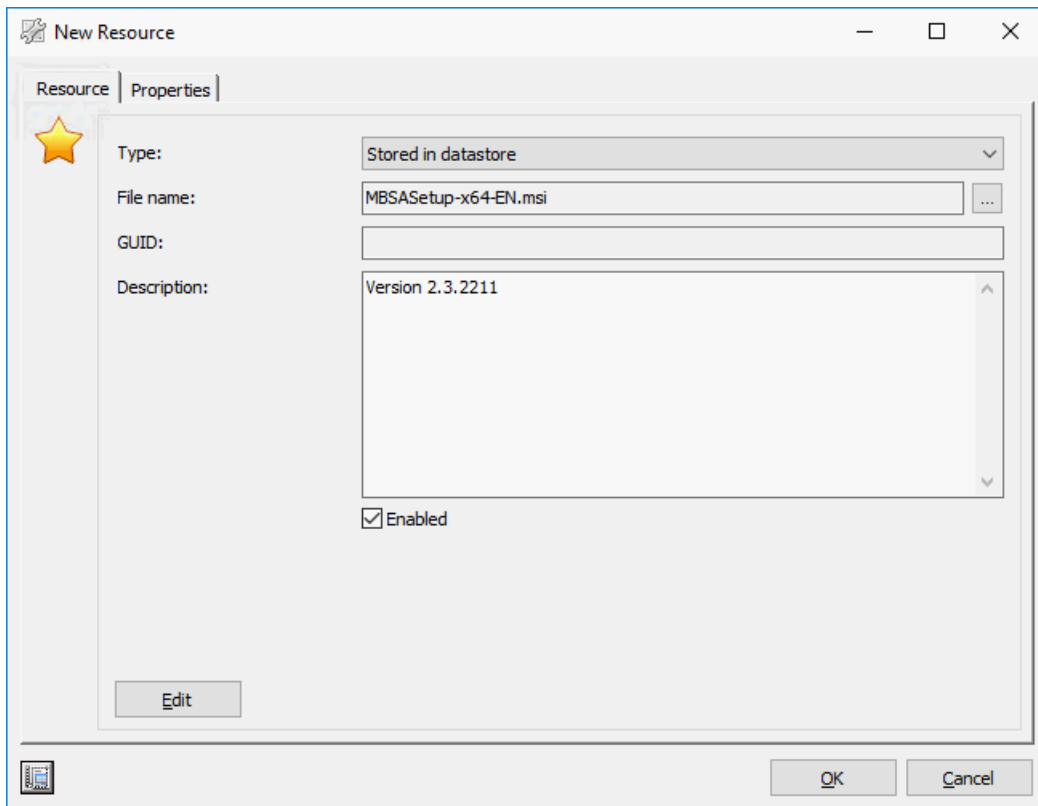
This document also provides some additional pointers for an initial tour of discovery of RES ONE Automation.

4.1 Create a Resource

Resources contain files like MSI files, setup files, patches, hotfixes, etc. When you configure a Task, you can use Resources to install applications, security updates, etc. Before you can configure a Task in RES ONE Automation to install Microsoft Baseline Security Analyzer, you need to create a Resource in RES ONE Automation from the installation MSI.

This scenario is based on a Resource stored in the Datastore. You can also store Resources on a file share, as a Resource package or located at a URL.


1. Download the latest version of the Microsoft Baseline Security Analyzer from <http://www.microsoft.com/mbsa>. The MSI file is called `MBSASetup-x64-EN.msi`.
2. In the Console, go to **Library > Resources**.
3. Click **New** in the command bar and select **In datastore**. The **New Resource** window opens with **Stored in Datastore** selected as type.
4. In the **File name** field, click the **browse** button , select the file `MBSASetup-x64-EN.msi` and click **Open**.
5. Optionally, specify a short comment.
6. Click **OK**. The Resource is uploaded to the Datastore and is shown in your **Resources list**.



4.2 Create a Module

If you want to execute a specific Task, the first step is to create a **Module**. A Module consists of (a sequence of) **Task(s)**, including all settings needed to perform them. The ability to use a series of Tasks is very useful if a Task is directly related to another one.

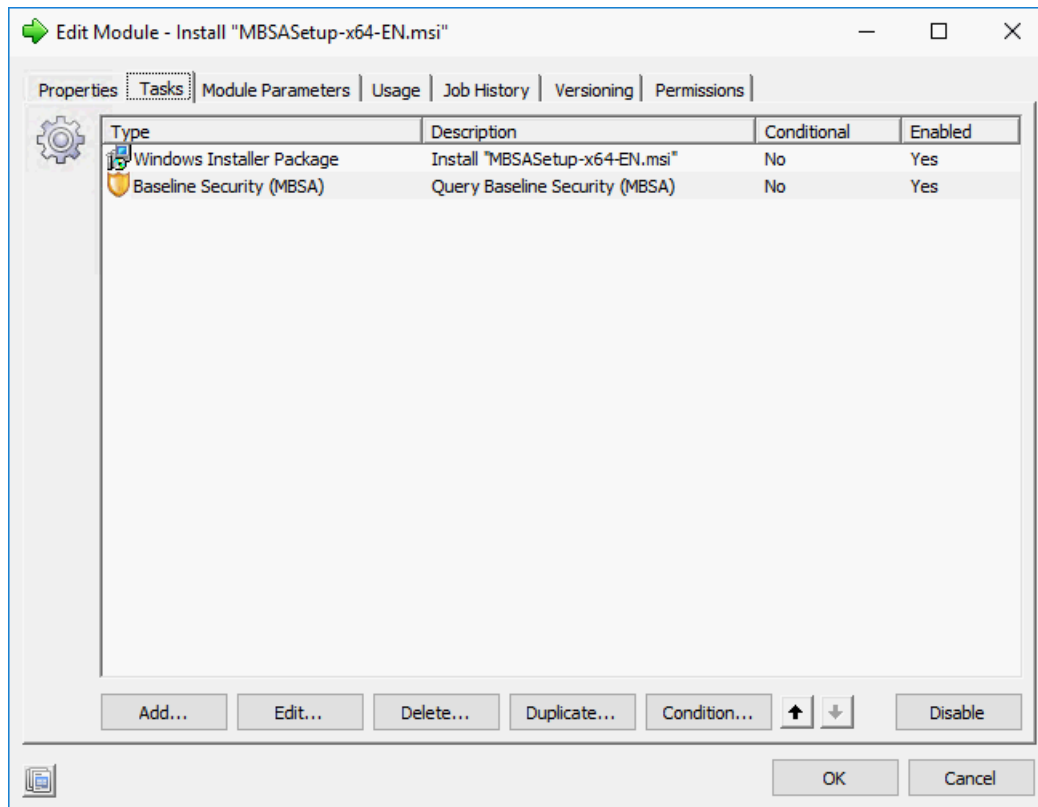
In this chapter, we will configure a Module and add a Task that can install the Microsoft Baseline Security Analyzer, and a Task that can execute an MBSA query.

1. In the Console, go to **Library > Modules**.
2. Click **New** in the command bar. The **New Module** window opens.
3. At the **Properties** tab, specify *MBSA* as Module name, optionally specify a short comment and click the **Tasks** tab.
4. Click **Add**. The **Select Task** window opens.
5. In the **Instant Search** field, specify "installer". **Windows Installer Package (Install, Repair, Patch, Remove)** is shown.
6. Click **Install**. The Task **Install Windows Installer Package (/i)** opens.
7. On the **Settings** tab, click the **browse** button  in the **Filename** field. The **Select Resource** window opens.
8. Select the Resource *MBSASetup-x86-EN.msi* and click **OK**. Click **OK** again to return to the **Add Module** window.

After configuring the Task that installs the Microsoft Baseline Security Analyzer, we can configure a Task that executes an MBSA query.

1. Click **Add**. The **Select Task** window opens again.
2. In the **Instant Search** field, type "mbsa". **Baseline Security (MBSA) (Query)** is shown.
3. Click **Query**. The Task **Query Baseline Security (MBSA)** window opens.

4. Leave the settings as they are and click **OK** to add the Task to the Module. The Module should now look like this:



5. After reviewing the Module, click **OK** to save the Module.

4.3 Schedule a Job

When you have made all preparations in the **Library** section of the Console, you can start using the available Tasks. You can take this final step at the **Jobs** node of the Console, by scheduling **Jobs**.

A Job combines Agents (**Who**), Modules and/or Projects (**What**) and time (**When**). This means that if you schedule a Job, it performs certain Tasks on certain Agents at a specific time.

To schedule a Job with the Module that we created in the previous chapter:

1. In the Console, go to **Jobs > Scheduling**.
2. Click **New**. The **New Job** window opens.
3. Click the **What** field. The **Select Run Book/Project/Module** window opens.
4. On the **Modules** tab, select the Module *MBSA* that you created in the previous chapter and click **OK**. The **New Job** window is shown again.
5. Click the **Who** field. The **Select Agents/Teams** window opens.
6. Click **Search**. The available Agent(s) in your RES ONE Automation environment are shown.
7. Select the (online) Agent or Team on which you want to execute the Module and click **OK**.

The screenshot shows the 'New Job' dialog box with the following configuration:






- Description:** Query Baseline Security (MBSA)
- When:** Immediately
- Schedule offline Agents for next boot
 - or when Agent is resumed after hibernate, sleep or standby
- Use Wake-on-LAN when job starts
- If outside launch window:** Fail the job
- Maximum job duration: 10 minutes
- Timing:** Use local time on Agent
- What:** Query Baseline Security (MBSA)
- Who:** DESKTOP02-EV
- Scheduled

8. After reviewing the Job, click **OK** to execute the Job.

4.4 View the Job results

When a Job has finished, view all basic information of the finished Job at the node **Jobs > Job History**. This includes the description of the Job, its duration and, very important: its **status**. The Job status clearly shows whether a Job finished correctly.

A finished Job can have any of the following statuses:

-  **Aborted:** The Job was aborted by the administrator during its execution.
-  **Canceled:** The Job was canceled before it was executed.
-  **Completed:** The Job has completed without any errors.
-  **Completed with errors:** The Job completed, but not all Tasks were performed correctly. On the Task's **Properties** tab, the option **Stop Job on error** was not enabled.
-  **Failed:** The Job did not finish correctly. The option **Stop Job on error** was enabled and an error occurred.

Although the basic Job results give you a basic overview of the completed Jobs, the information provided is limited and you cannot compare various Tasks and/or Agents. You therefore often need to view a Job in detail.

To view the results of the Job that we executed in the previous chapter:

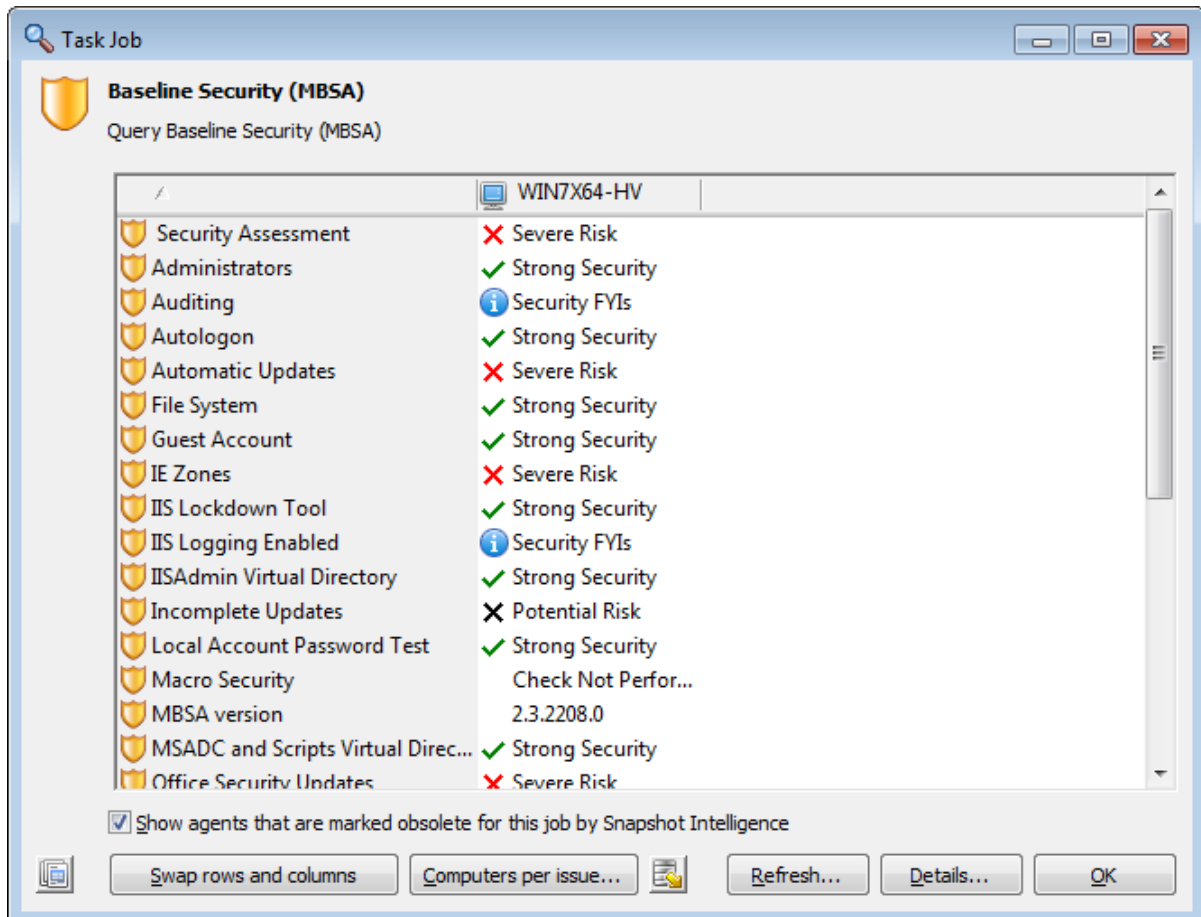
1. In the Console, go to **Jobs > Job History**.
2. Select the Job that just completed and click **View**. The **Job** window opens.

There are two different approaches to viewing the Job results details. Each approach has its specific use, advantages and disadvantages:

View the Job results per Task

When you view the Job results per Task, basic information about the results for all Agents that executed the Task is shown. If the Task performed a query, you can use this approach to compare the results of the Agents. In this way, you can easily see differences between Agents.

1. Click the **Tasks** tab. This shows all Tasks that were part of the Job.
2. Select the Task **Query Baseline Security (MBSA)** and click **Details**.



View the Job results per Agent

When you view the Job results per Agent, detailed information per Agent is shown. If the Task performed a query, use this approach to view the Task results for a specific Agent in detail. Although you cannot compare between Agents, the results are more detailed.

1. Click the **Agents** tab. This shows all Agents that performed the Tasks in the Job.
2. Select an Agent and click **Details**. This shows detailed information about all Tasks that were executed by the Agent.
3. Select the Task **Query Baseline Security (MBSA)** and click **Details**.

The screenshot shows the 'Query Baseline Security (MBSA)' window with the 'Results' tab selected. The window displays a list of security issues with their scores and detailed descriptions. Below the list is a table of system properties and their values.

Issue	Score	Result
Security Assessment	✗ Severe Risk	
Administrators	✓ Strong Security	No more than 2 Administrators were found on this com...
Auditing	ℹ Security FYIs	Neither Logon Success nor Logon Failure auditing are e...
Autologon	✓ Strong Security	Autologon is not configured on this computer.
Automatic Updates	✗ Severe Risk	The Automatic Updates feature has not been configure...
File System	✓ Strong Security	All hard drives (1) are using the NTFS file system.
Guest Account	✓ Strong Security	The Guest account is disabled on this computer.
IE Zones	✗ Severe Risk	Internet Explorer zones do not have secure settings fo...
IIS Lockdown Tool	✓ Strong Security	The IIS Lockdown tool was developed for IIS 4.0, 5.0, ...
IIS Logging Enabled	ℹ Security FYIs	All web and FTP sites are using the recommended loggi...
IISAdmin Virtual Dire...	✓ Strong Security	IISADMPWD virtual directory is not present.
Incomplete Updates	✗ Potential Risk	A previous software update installation was not comple...
Local Account Passw...	✓ Strong Security	Some user accounts (1 of 3) have blank or simple pass...

Property	Value
Computer name	RESQA\WIN7X64-HV
IP address	10.1.9.7
Scan date	18-11-2013 13:13
MBSA version	2.3.2208.0
Catalog synchronization date	

Buttons at the bottom: Show these results in Microsoft Baseline Security Analyzer..., Computers per issue..., Close

System tray: WIN7X64-HV @ 18-Nov-2013 13:13:45 (local)

Chapter 5: Where to go from here?

Explore further

A few pointers to explore RES ONE Automation even further:

- Export the Job results to an XML file.
- Create an Instant Report of the Job results.
- Create a Building Block of your Modules, Projects, Run Books and/or Teams.
- Deploy an Agent on another computer in your network.
- Schedule a Job on multiple Agents and compare their results.
- View the current activity while the Job is running.
- View the history of the Agents that executed the Job.
- Quickly create a new Module for a specific Task by using the **Quick Task** functionality (available from the **Modules** node).
- Add the Module to a Project, which is a combination of Modules.
- Run this Project on a Team of Agents.
- Create a Run Book, and execute Jobs on different Agents.
- Use parameters to make your Modules, Projects and Run Books generic.
- Use evaluations and conditions in a Module, to execute or skip Tasks in the Module based on the outcome of a previous Task.

More information

- For information about all functionality of RES ONE Automation, see the **Help**, which is available after installing the RES ONE Automation Console, by pressing F1.
- For in-depth information, see the **RES ONE Automation Administration Guide**.
- For installing the RES ONE Connectors, see the **Getting Started with RES ONE Connectors**.

Chapter 6: RES ONE Automation Glossary

Object	Function
Agent	An Agent is a service running on a computer and can execute RES ONE Automation Tasks.
Administrative role	An administrative role determines the level of access that a Console user has to the various parts of the Console.
Building Block	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.
Condition	A condition determines, based on one or more expressions, whether a Task, Module or Run Book Job should be executed, skipped or failed.
Console	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.
Datastore	The Datastore is a database that contains all the settings and Resources of your RES ONE Automation environment.
Dispatcher	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.
Evaluator	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.
Function	A function generates (partial) input for text-based properties of Tasks. This makes it possible to create information in various fields in Tasks automatically.
Global Settings	Global Settings determine the general behavior of your RES ONE Automation environment, and the default behavior of Dispatchers and Agents.
Instant Reports	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.
Job	<p>A Job is the execution of Tasks at a specific time on specific Agents.</p> <ul style="list-style-type: none"> • For Modules and Projects, both the time and the Agents are set at the moment of scheduling. • 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.
Management Portal	The Management Portal is the web-based application for scheduling Jobs in your RES ONE Automation environment.
Module	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.

Object	Function
Parameter	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.
Project	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.
Resource	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.
Run Book	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.
Task	A Task is a specific action that can be performed by an Agent. Tasks are part of a Module.
Team	A Team is a group of Agents. Teams can also contain other Teams.
Trusts	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.
Variable	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.

Chapter 7: 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.

7.1 Supported Microsoft Windows operating systems

Windows Server version	Edition
Windows Server 2016	Server 2016 Datacenter
	Server 2016 Standard
	Server 2016 Essentials
Windows Server 2012 R2	Server 2012 R2 Datacenter
	Server 2012 R2 Standard
	Server 2012 R2 Essentials
	Server 2012 R2 Foundation
Windows Server 2012	Server 2012 Datacenter
	Server 2012 Standard
	Server 2012 Essentials
	Server 2012 Foundation
Windows Server 2008 R2	Server 2008 R2 Datacenter
	Server 2008 R2 Enterprise
	Server 2008 R2 Standard
	Server 2008 R2 Web Edition
Windows Server 2008	(not supported by Agent+ and Dispatcher)
Windows Server 2003 R2	(not supported by Agent+ and Dispatcher)
Windows Server 2003	(not supported by Agent+ and Dispatcher)

Windows version	Edition
Windows 10	Windows 10 Enterprise
	Windows 10 Enterprise LTSC
	Windows 10 Pro
	Windows 10 Education
Windows 8.1	Windows 8.1 Enterprise
	Windows 8.1 Pro
Windows 8	Windows 8 Enterprise
	Windows 8 Pro
Windows 7	Windows 7 Enterprise
	Windows 7 Professional
	Windows 7 Ultimate
	Windows 7 Home Premium
	Windows 7 Home Basic
	Windows 7 Starter
Windows Vista	(not supported by Agent+ and Dispatcher)
Windows XP	(not supported by Agent+ and Dispatcher)

7.2 Supported Unix/Linux operating systems

Unix/Linux	Version
Red Hat Enterprise Linux	7
	6
	5.3 and higher
	4.5 and higher
SUSE Linux Enterprise Server	12
	11: SP1 and higher
	10: SP3 and higher
IBM AIX	Version 7.1: Technical level 01 and higher (PPC)
	Version 6.1: Technical level 7 and higher (PPC)
	5L Version 5.3: Technical level 4 and higher (PPC)
Ubuntu	16.04
	12.04
Oracle (Sun) Solaris	11 <ul style="list-style-type: none"> for SPARC: update 1 and higher for Intel: update 3 and higher
	10 <ul style="list-style-type: none"> for SPARC: update 6 and higher for Intel: update 11 and higher

Apple	Version
Apple Mac OS X	10.10 Yosemite
	10.9 Mavericks
	10.8 Mountain Lion

7.3 Supported Directory Services

Directory Service	Version
Microsoft Windows Active Directory	Functional level: 2016
	Functional level: 2012 R2
	Functional level: 2012
	Functional level: 2008 R2
	Functional level: 2008
	Functional level: 2003

7.4 Supported Database systems

Database type	Version
Microsoft SQL Server	2016
	2014
	2012
	2008 R2
	2008
	2005
Microsoft Azure SQL	
IBM DB2	10.5
	9.8
	9.7
	9.5
MySQL	5.6
	5.0
Oracle	12c
	11.2
	11.1

7.5 Supported Mail Servers

Mail Server	Version
Microsoft Exchange Server	2016
	2013
	2010 SP1
	2010
	2007 (not supported by Agent+)
	2003 (not supported by Agent+)

7.6 Supported Application Virtualization

Application Virtualization	Version
Microsoft App-V	4.6
	4.5

7.7 Supported Hypervisors

Hypervisor	Version
Microsoft Hyper-V	2012 R2
	2012
	2008 R2
Citrix XenServer	6.5
	6.2

7.8 Supported Provisioning Services

Provisioning Service	Version
Microsoft System Center Configuration Manager	2016 (up to v1702)
	2012 R2
	2012 SP1
	2012
	2007
LANDesk	9.5 SP2
	9.5 SP1
	9.5
Citrix Workflow	2.5

7.9 Supported Microsoft Windows PowerShell Scripting

Microsoft Scripting	Version
Windows PowerShell	1.0
Windows PowerShell	2.0
Windows PowerShell	3.0
Windows PowerShell	4.0
Windows PowerShell	5.0

7.10 Supported Citrix "Ready" products

Citrix product	Version
XenApp	7.6
	7.5
	6.5
	6.0
XenDesktop	7.6
	7.5
	7.0
	5.6
	5.5
	5.0
XenServer	6.1
	6.0

7.11 Supported RES products

RES product	Version
RES Workspace Manager	2012 and higher
RES Workspace Manager Relay Server	2012 and higher, as support in Task Install Windows Installer Package
RES IT Store (also known as RES ONE Service Store, RES ONE Identity Director)	2014 and higher