

# Mail & Deploy

## Reference Manual

Version 2.1.1

Build #15

# TABLE OF CONTENTS

<b>Introduction .....</b>	<b>5</b>
General Introduction .....	6
Terminology .....	6
Architecture .....	7
Server .....	7
Client .....	7
Contact Us .....	8
<b>Server .....</b>	<b>9</b>
Introduction .....	10
Components .....	10
Requirements .....	11
Installation .....	12
Updates .....	13
Configuration .....	14
Control Panel .....	14
<b>Client .....</b>	<b>17</b>
Introduction .....	18
Requirements .....	18
Installation .....	18
Updates .....	18
Starting .....	19
Users and User Groups .....	20
Introduction .....	20
Users .....	20
User Groups .....	23
Datasources .....	24
Introduction .....	24
Datasource Elements .....	24
Datasource Types .....	25
Reports .....	27
Introduction .....	27
General .....	27
Report Types .....	30
Preview .....	43
Hub Authorizations .....	44
Replacing Datasources .....	44
Tasks .....	46
Introduction .....	46
Attributes .....	46
Hub Authorizations .....	50
Exception Handling .....	50
Server Status .....	54
Executions .....	54
Help .....	55
Additional Settings .....	56

Styles.....	56
Calendars.....	56
E-Mail Servers.....	56
Hub Settings.....	57
Active Directory Connection Settings .....	58
Production Repository.....	58
Global Variables.....	59
<b>Hub.....</b>	<b>60</b>
Introduction .....	61
Configuration.....	62
Hub Authorizations .....	62
Usage.....	65
Introduction .....	65
General .....	65
Reports .....	65
Tasks .....	69
<b>API.....</b>	<b>70</b>
Introduction .....	71
Configuration.....	72
Usage.....	73
Introduction .....	73
Reference .....	74
<b>Appendix.....</b>	<b>79</b>
Actions.....	80
Introduction .....	80
Action Types .....	80
Expressions .....	99
Introduction .....	99
Constant Values .....	99
Dynamic Values.....	100
Calculations.....	103
Data Types.....	103
Built-In Functions .....	105
Module.....	105
Examples.....	106
Execution Calendar.....	107
Introduction .....	107
Scope .....	107
Filtering.....	107
Month View.....	108
Day View.....	109
File Type Settings .....	110
Adobe PDF Document .....	110
Table Cell Types .....	111
Introduction .....	111
Examples.....	111
Styles.....	112
Introduction .....	112
General Attributes.....	112
Style Types .....	112

Log Files .....	114
Application Logs .....	114
Task Execution Logs .....	114
Development Environment .....	115
Introduction .....	115
Configuration .....	115
Publishing .....	115
Tools .....	117
API Shell .....	117
Repository Converter .....	118
Qlik Sense Extension .....	119
Version History .....	121
Version 2.1.1 .....	121
Version 2.1.0 .....	129

Section 1

# Introduction

# General Introduction

Mail & Deploy is a tool that enables you to design reports using data and charts from QlikView and Qlik Sense and distribute them by e-mail, to the file system or by uploading them to an FTP server. This reference manual contains a full documentation about all components of Mail & Deploy and should ideally be used together with other parts of the documentation such as tutorials and videos.

This first section includes an overview about the architecture of Mail & Deploy (→ [Architecture](#)) as well as information about how to obtain support, assistance, access to knowledge bases and consulting services (→ [Contact Us](#)).

## Terminology

In order to use this documentation in the most efficient manner, the definition of some terms used in a Mail & Deploy environment is extremely helpful. These terms are used consistently throughout the documentation of Mail & Deploy (as well as in the user interface of the software itself).

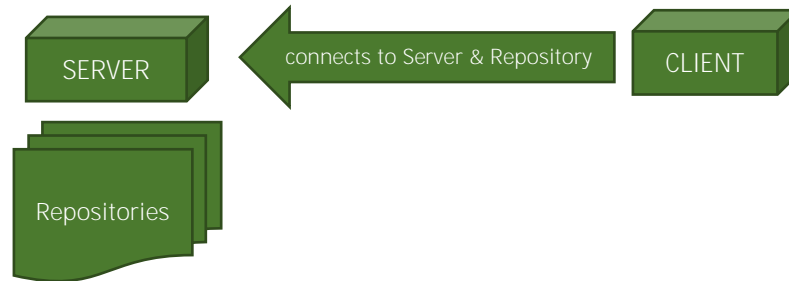
## Reports and Report Documents

In Mail & Deploy a report document is a file (e.g. a PDF file) that contains the actual report with data from QlikView, Qlik Sense etc. In other words: a report document is what you want to distribute to users.

A report, on the other hand, describes how such a report document shall be created; the report – depending on the report type - contains a template, parameters, filter instructions, report elements such as charts, tables and text; if a user requests a report document through Mail & Deploy Hub or if a task is setup to automatically create and distribute report documents on a regular basis, the report will be processed (by processing all filter instructions, reading data such as tables and charts from the datasources and inserting it into the template) and the output is a report document that will be distributed to the users.

# Architecture

Mail & Deploy consists of two main components; **server** and **client**. In a typical Mail & Deploy installation, there is one server and one or more clients.



## Server

Each Mail & Deploy environment has at least one Mail & Deploy Server component which manages repositories, executes tasks, provides an interface for Mail & Deploy Clients and hosts Mail & Deploy Hub, which is a web-interface that allows on-demand requests and subscription of reports. You can get more detailed information about the server in the section [Server](#).

## Repository

A repository is a database that contains users, user groups, data sources, reports and tasks. Mail & Deploy supports a theoretically unlimited number of repositories, you can organize your report data by splitting it up into multiple repositories, if required. Repositories are stored and organized by Mail & Deploy Server.

**Example:** If you want to distribute IT reports and Controlling reports in Mail & Deploy, it makes sense to create one repository for each category and allows IT users to access the IT repository only and the Controlling users to access the Controlling repository only.

## Client

Mail & Deploy Clients are used to connect to a Mail & Deploy Server, open a repository and edit data in that repository by managing users and datasources, designing reports and setting up tasks. The number of clients is not limited – you can install as many as you want in your Mail & Deploy environment. Clients communicate with Mail & Deploy Server via the network using a port that you can configure (→ [Server Configuration](#)). You can get more detailed information about clients in section [Client](#).

# Contact Us

You can contact us by navigating to <http://www.mail-and-deploy.com> as well as by e-mail using [info@mail-and-deploy.com](mailto:info@mail-and-deploy.com) or [support@mail-and-deploy.com](mailto:support@mail-and-deploy.com).



Section 2  
Server

# Introduction

Each Mail & Deploy environment requires at least one Mail & Deploy Server. The server is a component that manages Repositories, den web-server for Mail & Deploy Hub and Mail & Deploy API and it executes regular tasks or on-demand requests. Mail & Deploy Server runs as a local windows service named *Mail & Deploy Server*; if that service is not running, Mail & Deploy cannot be accessed by clients, the hub or API consumers. The configuration of the service can be done in Mail & Deploy Server Control Panel (→ [Control Panel](#)).

## Components

The service described above that hosts Mail & Deploy Server contains the following components:

### Main Server

The main server is allows Mail & Deploy Clients to connect to the server and open a repository. The main server will automatically be started when the service is started.

### Execution Engine

The execution engine handles the execution of tasks (→ [Tasks](#)) and on-demand requests submitted through the hub (→ [Hub](#)) or the API (→ [API](#)). The parallel executions of more than one jobs is handled by the execution engine. You can manage the execution engine using Mail & Deploy Server Control Panel (→ [Control Panel](#)).

### Hub

The hub is an optional, integrated HTTP web server which provides users the possibility to logon to a web interface and request reports and execute tasks on-demand (→ [Hub](#)).

### API

Mail & Deploy API is an optional, integrated HTTP web server that allows 3<sup>rd</sup> party developers to access Mail & Deploy functionality (→ [API](#)).

# Requirements

The following requirements must be fulfilled in order to install and use Mail & Deploy Server:

- An environment capable of executing 64-bit applications.
- Microsoft .NET Framework 4.7
- 500 MB of free disk space
- Service Account

The service account is used to run Mail & Deploy Server. The minimum requirement is, that the service account has local administration rights. Optional requirements – depending on which features of Mail & Deploy you want to use – may be:

- If you want to use QlikView documents with Mail & Deploy, the service account needs to have access to a licensed local QlikView Desktop Client. The best way to ensure this is to login as the service account on the computer you want to install Mail & Deploy Server to and start QlikView Desktop Client; if that results in only a Personal Edition to be opened, you need to ensure a proper license before you can continue.
- If you want to use QlikSense apps with Mail & Deploy, the service account needs to have full access to those Qlik Sense apps. The service account needs to be an account explicitly available for Mail & Deploy; otherwise, license issues might occur when Mail & Deploy connects to a QlikSense app.
- If you want to read data from or save data to network drives, the service account needs to have read and/or write permissions on those network drives.

- SMTP Server (*optional*)

If you want to distribute reports by sending e-mails through Mail & Deploy Server, the computer that runs Mail & Deploy Server needs to have access to an SMTP server (i.e. there needs to be an SMTP server and it must not be blocked by a firewall). Mail & Deploy can either use windows authentication (e.g. the service account will be authenticated) or custom username and password for authentication.

- QlikView Desktop Client (*optional*)

If you want to use QlikView documents with Mail & Deploy, a licensed QlikView Desktop client needs to be installed on the computer that runs Mail & Deploy Server and the service account needs to have access to it (see above for service account requirements). Mail & Deploy Server needs to be able to access these QlikView documents through the file system.

- Qlik Sense (*optional*)

If you want to use Qlik Sense apps with Mail & Deploy, a Qlik Sense Server must be in reach of the computer on which Mail & Deploy Server is installed. Mail & Deploy Server does not have to be installed on the Qlik Sense server; a working network connection is sufficient, as long as necessary ports are not blocked by a firewall. However, installation directly on the computer on which Qlik Sense is installed is also possible. The service account needs to have full access to the Qlik Sense apps to be used (see above for service account requirements).



Please note, that detailed information about the number of CPUs and RAM required cannot be given here, because that heavily depends on the complexity and size of QlikView documents and Qlik Sense apps involved, the number of executions to be performed at a given time etc.

# Installation

The installation of Mail & Deploy Server can be started by executing *Mail & Deploy Setup.exe* under a user account with local administration rights and then selecting *Install Mail & Deploy Server*. The installation wizard guides you through the installation and requires you to enter the following data:

<b>Path</b>	The full path to the directory to which to install Mail & Deploy Server.
<b>Default Language</b>	The default language for this installation of Mail & Deploy Server. This is the language in which Mail & Deploy Server Control Panel (→ <a href="#">Control Panel</a> ) will be displayed.
<b>Service Credentials</b>	The windows account name and password of the service account that will run the local service that contains Mail & Deploy Server. The windows account name must be entered in <i>domain\accountname</i> format.

The required license can be installed by copying the *license.lic* file to the *Data\Licensing* sub-folder of the root directory of Mail & Deploy Server.

# Updates

New versions of Mail & Deploy are published to all customers with a valid and current maintenance contract. There are two ways of updating Mail & Deploy Server:

**Manual Update.** When a new version is available, you will be notified via e-mail; this e-mail contains a download link to a file named *Mail & Deploy Update.mdpkg*. This file can then be installed through Mail & Deploy Server Control Panel (→ [Update](#)). The download also contains a file named *Version History.pdf* which will provide you information about all the changes when updating to the new version, particularly if there are any breaking changes that might require manual intervention after the update has been installed.

**Automatic Update.** If the computer that runs Mail & Deploy Server has access to the internet, you can use the auto-update feature of Mail & Deploy Server Control Panel (→ [Update](#)) to automatically download and install the update.

The updates – whether it's a manual or an automatic update – also contain the new version for Mail & Deploy Client, which will be installed automatically whenever Mail & Deploy Client is started and connects to Mail & Deploy Server (→ [Updates](#)).

# Configuration

## Control Panel



The server can be configured using **Mail & Deploy Server Control Panel**. You can start the control panel by double-clicking the file *Mail & Deploy Server Control Panel.exe* in the root directory of your Mail & Deploy Server installation. The user interface is divided into sections described in detail below.

### Overview

In this section an overview about the current status of your Mail & Deploy environment is displayed.




#### Server Service Status

The current status of the local service that runs Mail & Deploy Server is displayed and can be controlled using the following buttons:


-  Click this button to start the service; it may take a few seconds until the service is actually running.
-  Click this button to stop the service; it may take a few seconds until the service is actually stopped.

#### Execution Engine Status

The current status of the execution engine (→ [Execution Engine](#)) is displayed and can be controlled using the following buttons:

-  Click this button to start the execution engine; it may take a few seconds until the execution engine is actually running.
-  Click this button to pause the execution engine; it may take a few seconds until the execution engine is actually paused.
-  Click this button to stop the execution engine; it may take a while until the execution engine is actually stopped.

### Executions



The list shows all executions which are currently processed by Mail & Deploy Server. You can abort an execution by clicking on the item in the list and then clicking the  button.



Please note that it may take a few seconds until the execution is actually aborted and disappears from the list of current executions.

## Repositories

In this section a list of all repositories is displayed. You can manage repositories using the buttons next to the repository list.

-  Click this button to create a new, empty repository. A new dialog will show up that allows you to enter the name of the repository as well as the windows account name of a user than will be granted administration access to the repository (→ [Users and User Groups](#)). The windows account name must be entered in *domain\accountname* format.
-  Click this button to import a repository from a file. A new dialog will show up that allows you to locate the repository file to import. It is important to note, that only repositories which are compatible with the installed version of Mail & Deploy should be imported; otherwise, errors or other unwanted effects may occur.



Click this button to edit the name of the repository. A new dialog will show up that allows you to enter the new name of the repository.




Click this button to create a copy of the currently selected repository. A new dialog will show up that allows you to enter the name of the copied repository.




Click this button to delete the currently selected repository.



Please note, that repositories can only be created, edited, copied and deleted when the server service is currently stopped.

By clicking the  button you can open the execution calendar (→ [Execution Calendar](#)) which will include data of all available repositories.

## Settings

In this section the settings of Mail & Deploy Server can be configured. In order to submit changed settings you need to click the  button.

### General

This section contains the following settings:

<b>Default Language</b>	This is the default language of Mail & Deploy which controls the user interface language of Mail & Deploy Server Control Panel and Mail & Deploy Hub (before a user has logged on to a repository and his personal language preference will be used).
<b>Maximum Concurrent Executions</b>	This is the maximum number of parallel executions performed by the execution engine; this number is limited by your license.
<b>Maximum Processors</b>	This is the maximum number of processors which limit the consumption of system resources by some components of Mail & Deploy. Currently, this setting only controls how many processors an instance of QlikView used by Mail & Deploy is allowed to use.
<b>Automatically Start Execution Engine with Server</b>	If this setting is enabled, the execution engine (→ <a href="#">Execution Engine</a> ) will automatically be started, when Mail & Deploy Server is started.

### Network

This section contains the following settings:

<b>Main Server Port</b>	This is the port used for the communication between Mail & Deploy Clients and the server. If clients are used to connect to the server from other computers, you must ensure that the firewall has this port open.
<b>Enable Hub</b>	This determines whether Mail & Deploy Hub (→ <a href="#">Hub</a> ) is enabled; if not, the HTTP web server that hosts the hub is not started and users cannot connect to the hub website.
<b>Hub Port</b>	This is the port on which the HTTP web server of Mail & Deploy Hub (→ <a href="#">Hub</a> ) is started.
<b>Hub Relative URL</b>	This is the relative URL on which the HTTP web server of Mail & Deploy Hub (→ <a href="#">Hub</a> ) is started.

## Service

This section contains the following settings:

<b>Service User Credentials</b>	The username and password of the windows account that Mail & Deploy Server Service shall run under.
---------------------------------	---

## Update

This section can be used to install updates for Mail & Deploy Server (→ [Updates](#)).

### Automatic Update

If the computer that runs Mail & Deploy Server Control Panel has a working internet connection, you can click *Check for updates* to connect to our update server and check whether there is a new version available. If that is the case, you can view the version history to see which changes the new version will bring and if there are any manual steps required due to breaking changes from a previous version. If you decide to install the update, the installer will automatically be downloaded and started.

### Manual Update

If you have received an update package file named *Mail & Deploy Update.mdpkg*, you can click *Install a manually downloaded update* and select that file. The installer will automatically be started.



### Section 3

## Client

# Introduction

Mail & Deploy Client is an application that is required to log on to a repository and design reports, setup distributions, configure tasks administrate users and user groups, set authorizations for users for Mail & Deploy Hub etc.

## Requirements

The following requirements must be fulfilled in order to install and use Mail & Deploy Client:

- An environment capable of executing 64-bit applications.
- Microsoft .NET Framework 4.7
- 500 MB of free disk space
- Mail & Deploy Server Connection  
The computer needs to have a connection to the computer that runs Mail & Deploy Server and the firewall must not block the port used for the communication between Mail & Deploy Server and Mail & Deploy Client; this port, however, can be configured in Mail & Deploy Server Control Panel (→ [Control Panel](#)).
- Microsoft Office (*optional*)  
If you want to design reports based on Microsoft Office formats (Microsoft Excel, Microsoft PowerPoint, Microsoft Word) then the respective Microsoft Office application needs to be installed on the computer and licensed.
- PDF Viewer (*optional*)  
If you want to design reports that will be created in PDF format, a PDF viewer needs to be installed in order to preview these reports during the design process.

## Installation

The installation of Mail & Deploy Client can be started by executing *Mail & Deploy Setup.exe* or *Mail & Deploy Client Setup.exe* under a user account with local administration rights and then selecting *Install Mail & Deploy Client*. The installation wizard guides you through the installation and requires you to enter the following data:

<b>Path</b>	The full path to the directory to which to install Mail & Deploy Client.
<b>Default Language</b>	The default language for this installation of Mail & Deploy Client. This is the language in which Mail & Deploy Client will be displayed until you have logged in to a repository; at that time the language set in the user () will be used.
<b>Mail &amp; Deploy Server</b>	The host name / IP address of the computer that runs Mail & Deploy Server and the port on which Mail & Deploy Server runs.

## Updates

Everytime Mail & Deploy Client is started, it connects to the configured Mail & Deploy Server and checks if a new version of Mail & Deploy Client is available; if that is the case, the update will automatically be downloaded and installed and Mail & Deploy Client will be restarted.



Please note that a local administrator account is required to install the update; if the user that runs Mail & Deploy Client does not have these rights, you will have to enter credentials of a windows account that does have local administrator rights.

## Starting

Mail & Deploy Client can be started by double-clicking *Mail & Deploy Client.exe* located in the installation directory you provided (→ [Installation](#)).

## Connecting to a Repository

When you start Mail & Deploy the *Connect to Repository* dialog comes up. It lists all repositories of the currently configured Mail & Deploy Server (→ [Server Connection Settings](#)) to which the windows user, with which you are starting Mail & Deploy Client, has access to. The following criteria apply:

- The repository needs to contain a user (→ [Users](#)) that has windows credentials (→ [Credentials](#)) and whose username matches the windows username (in *domain\accountname* format) of user starting Mail & Deploy Client.
- The user needs have a role of at least *Designer*.

The following buttons are available:



Click this button to refresh the list of available repositories.



Click this button to edit the server connection settings (→ [Server Connection Settings](#)).

## Server Connection Settings

These settings enable you to specify the host / IP address of the computer on which Mail & Deploy Server is installed as well as the port on which the main server runs (→ [Network](#)). These settings are initialized during the installation of Mail & Deploy Client (→ [Installation](#)).

## Main Window

After connecting to a repository, you will be redirected to the main window of Mail & Deploy Client. It consists of:

**Top Menu.** This menu allows you to connect to another repository (→ [Connecting to a Repository](#)) and to configure additional settings (→ [Additional Settings](#)).

**Left Menu.** This menu allows you to manage users and user groups (→ [Users and User Groups](#)), datasources (→ [Datasources](#)), reports (→ [Reports](#)) and tasks (→ [Tasks](#)), review the server status (→ [Server Status](#)) and access help (→ [Help](#)).

**Status Bar.** The status bar at the bottom contains – from left to right – the current status of Mail & Deploy Client, the username with which you are authenticated at the repository, the server host and the name of the repository to which you are currently connected and the version of Mail & Deploy Client.

# Users and User Groups

## Introduction

Users and User Groups have two usages in Mail & Deploy:

- Users and User Groups control who has access to Mail & Deploy Client and Mail & Deploy Hub (→ [Hub](#)) in order to create and design reports respectively to request reports and task executions on-demand.
- Users and User groups can be used as e-mail recipients, regardless of whether they have access to Mail & Deploy Client and Mail & Deploy Hub (→ [Hub](#)); in other words: you can add users and user groups with no access rights to Mail & Deploy at all in which case they function as e-mail addresses to which reports can be distributed.

You can access users and user groups by clicking *Users* in the left menu of Mail & Deploy Client.

## Managing Users and User Groups

You can manage users and user groups in three ways:

- Users and User Groups can be managed manually, by adding them to the repository by manually entering their data. You can do this by using the following buttons:



Click this button to create a new user or user group.



Click this button to edit the currently selected user or user group.



Click this button to create a copy of the currently selected user or user group.



Click this button to delete the currently selected user or user group.

- Users and User Groups can be synchronized with an active directory or a local user and user group directory (→ [Synchronize Users and User Groups with Active Directory Action](#)) by using a task (→ [Tasks](#)).
- Users and User Groups can be synchronized with a CSV file (→ [Synchronize Users and User Groups with File Action](#)) by using a task (→ [Tasks](#)).

Users and user groups can only be managed if the users with which you are connected to the repository has a role of at least *Administrator*.

## Users

A user is an entity that usually represents a single person to which report documents shall be distributed and/or who is allowed to design reports and setup tasks and perform administrative tasks in Mail & Deploy.

## General Attributes

Each user has the following general attributes:

<b>Name</b>	The name of the user; this is only used within Mail & Deploy to identify the user.
<b>Role</b>	The role (→ <a href="#">Roles</a> ) of the user which determines what the user can do within Mail & Deploy.

<b>Language</b>	The language in which user interfaces shall be displayed to the user.
<b>Time Zone</b>	The time zone to which the user's location belongs.
<b>Culture</b>	The culture associated with the user.
<b>E-Mail Address</b>	The e-mail address of the user. This attribute is only required if e-mails shall be sent to the user.

## Roles

Every user has a role that determines the rights the user has on the repository. All roles include all access rights of lower roles.

### None

A user with role *None* does not have any access rights in Mail & Deploy. The user therefore functions merely as an e-mail address that can be used to distribute reports to (→ [Distribute by E-Mail Action](#)).

### Hub Consumer

A user with role *Hub Consumer* can log on to the repository through Mail & Deploy Hub (→ [Hub](#)). Access rights to individual reports and tasks are managed through hub authorizations (→ [Hub Authorizations](#)). Users with this role can not connect to the repository through Mail & Deploy Client.

### API Consumer

A user with role *API Consumer* can access the repository through Mail & Deploy API (→ [API](#)). Users with this role can not connect to the repository through Mail & Deploy Client.

### Designer

A user with role *Designer* can access the repository through Mail & Deploy Client. Users with this role can not do certain tasks which are exclusive for administrators (→ [Administrator](#)).

### Administrator

A user with role *Administrator* have full rights on the repository. They have the following exclusive rights:

- Manage users and user groups (→ [Users and User Groups](#)).
- Manage e-mail servers (→ [E-Mail Servers](#)).
- Change hub settings (→ [Hub Settings](#)).
- Change active directory connection settings (→ [Active Directory Connection Settings](#)).
- Abort task executions (→ [Server Status](#)).

## Credentials

Every user can optionally have credentials that are used to authenticate the user in the repository. Credentials are required, if:

- The user wants to use Mail & Deploy Client to log on to the repository. In this case, the user is required to have credentials of type *Windows*.
- The user wants to login to Mail & Deploy Hub (→ [Hub](#)). In this case, any type of credentials is sufficient.

- The user is used to authenticate when using Mail & Deploy API (→ [API](#)). In this case, any type of credentials is sufficient.

## Types

There are two types of credentials:

**Custom.** This type of credentials consists of a custom username and password.

**Windows.** This type of credentials represents a windows account name; the credentials are connected to the windows account name with that name and the password will be the same as the one for that windows account.



Please note that if you want users to be able to use Mail & Deploy Client to connect to the repository, they are required to have credentials of type *Windows*. Custom credentials authentication through Mail & Deploy Client is currently not supported.

## Custom Properties

Each user can have custom properties which can be used for various things in Mail & Deploy:

- Custom properties can be used to define the range of values from which a user may select the value of a report parameter (→ [Report Parameters](#)) or task parameter (→ [Task Parameters](#)) when requesting a report document or the execution of a task through Mail & Deploy hub.
- When assigning e-mail addresses dynamically to a report document (→ [Dynamically Assign E-Mail Addresses to Report Document Action](#)), custom properties may be used to identify the users whose e-mail addresses shall be assigned to a report document.
- When assigning users dynamically to a report document (→ [Dynamically Assign Users to Report Document Action](#)), custom properties may be used to identify the users to assign to a report document.

Each custom property has the following attributes:

<b>Name</b>	The name of the custom property; this is only used within Mail & Deploy to identify the custom property.
<b>Values</b>	A list of expressions (→ <a href="#">Expressions</a> ) which evaluate to the values of the custom property.

Custom properties can be managed using the following buttons:



Click this button to create a new custom property.



Click this button to edit the currently selected custom property.



Click this button the create a copy of the currently selected custom property.



Click this button to delete the currently selected custom property.

In addition to managing custom properties manually, you can also assign custom properties when synchronizing with an active directory (→ [Synchronize Users and User Groups with Active Directory Action](#)) or by importing them from a file (→ [Import User Custom Properties From File Action](#)).

## User Groups



You can organize users in Mail & Deploy in user groups which can then be used to assign certain authorizations and distributions.

### General Attributes

Each user group has the following general attributes:

<b>Name</b>	The name of the user group; this is only used within Mail & Deploy to identify the user group.
-------------	--

### Users

This a list of all users who are members of the group. You can manage members of the group by using the  button to add and by using the  button to remove members.

# Datasources

## Introduction

In Mail & Deploy, datasources provide data, tables and charts that can be used in reports (→ [Reports](#)). Before a datasource, such as a QlikView document or a Qlik Sense app, can be used, it needs to be added to Mail & Deploy.

You can access datasources by clicking *Datasources* in the left menu of Mail & Deploy Client.

## Managing Datasources

Datasources can be managed using the following buttons:



Click this button to create a new datasource; a context menu will open that allows you to select which type of datasource you want to create.



Click this button to edit the currently selected datasource.



Click this button to delete the currently selected datasource.



Click this button to refresh the cache of datasource elements for the currently selected datasource (→ [Datasource Elements](#)). This process might take a little while, depending on the complexity of the datasource.

## Datasource Elements

Each datasource contains different types of elements which are stored in an internal cache by Mail & Deploy, so they can be accessed rapidly during report design and when setting up tasks. This internal cache needs to be rebuilt, if the structure – not the data – of the datasource changes (→ [Managing Datasources](#)).

**Example:** If a new field, variable, bookmark, embedded report or object has been added to your datasource (e.g. a QlikView document), then rebuilding the cache of datasource elements is required, if you want to use the newly added elements in Mail & Deploy. If the datasource has been reloaded, the cache does not have to be rebuilt, because no structural changes have taken place in the datasource.

The datasource element cache of a datasource can automatically be rebuilt on a regular basis (→ [Rebuild Datasource Cache Action](#)) in a task (→ [Tasks](#)).

## Bookmarks

This type of datasource element represents a bookmark that stores a state of the datasource. Bookmarks can be used to restore the state that they store (→ [Activate Datasource Bookmark Action](#)).

## Embedded Reports

This type of datasource element represents a report that is embedded in the datasource, such as a report designed with QlikView Desktop Client or a story in a Qlik Sense app. Embedded reports can be used as reports (→ [Datasource Embedded Reports](#)).

## Fields

This type of datasource element represents a single field that contains one or more values in the datasource. Fields can be used in various ways, such as in expressions (→ [Expressions](#)), to apply a filter (→ [Filter Datasource Field Action](#)), to perform a cycle over the values of a field (→ [Cycle Action](#)) etc.



## Objects

This type of datasource element represents various types of objects (such as a chart or a table) in the datasource. Objects can be included in reports (→ [Reports](#)), especially by using them a report elements (→ [Report Elements](#)).

## Variables

This type of datasource element represents a variable of a datasource that contains a value. Variables can be used in expressions (→ [Expressions](#)).

# Datasource Types

## QlikView Document

This type of datasource represents a QlikView document. The following attributes can be set:

<b>Name</b>	The name of the datasource; this is only used within Mail & Deploy to identify the datasource.
<b>QVW Path</b>	<p>The full path to the QlikView document. It's important to note that the QlikView document will be accessed by Mail &amp; Deploy Server and therefore the following requirements have to be met:</p> <ul style="list-style-type: none"> <li>➤ The user that runs Mail &amp; Deploy Server needs to have sufficient user rights to read and write to the directory.</li> <li>➤ The file path needs to be reachable from the computer on which Mail &amp; Deploy Server runs; this might require to use UNC file paths in some cases. Please note that mapped network drive letters cannot be used, even if the user that runs Mail &amp; Deploy Server has a mapped network drive set up.</li> </ul>
<b>Section Access</b>	If the QlikView document has section access enabled, you have to activate section access in Mail & Deploy and provide a username and password that will be used to authenticate against the QlikView document (→ <a href="#">Section Access</a> ).
<b>Click Button before Cache Refresh</b>	This is an optional attribute; it allows you to specify the QlikView Object ID of a button that you want Mail & Deploy to click, before rebuilding the datasource element cache (→ <a href="#">Datasource Elements</a> ). This can, for instance, be used to run a macro that makes all hidden datasource objects (such as charts, tables etc.) visible, so that Mail & Deploy can find them when rebuilding the datasource element cache. You can leave the attribute empty, if you don't want Mail & Deploy to perform a click on a button before rebuilding the datasource element cache.

## Section Access


Mail & Deploy supports the datasources in the form of QlikView documents which have section access enabled. There is, however, a requirement that there needs to be a custom credentials section access user defined in the QlikView document's section access table by using the *UserID* and *Password* fields in QlikView. This user needs to have sufficient rights in the QlikView document to perform everything you want Mail & Deploy to do in that QlikView document; a recommended practice would be to set this user up as an administrator user by setting *Access* in the section access table to *Admin*.

## Qlik Sense Server Application

This type of datasource represents a Qlik Sense app located on a Qlik Sense Server. The following attributes can be set:

<b>Name</b>	The name of the datasource; this is only used within Mail & Deploy to identify the datasource.
-------------	--

<b>Host</b>	The host name or IP address of the Qlik Sense Server.
<b>Port</b>	The port of the Qlik Sense Server.
<b>Virtual Proxy Path</b>	The path to the virtual proxy to connect to. Please note, that the proxy that Mail & Deploy connects to must not have extended security enabled in the associated Qlik Sense QMC.
<b>Protocol Type</b>	The protocol type to use to connect to the Qlik Sense Server. Which protocol type is required depends on how the Qlik Sense Server is configured in your environment.
<b>Application Name</b>	The name of the Qlik Sense app that shall be represented by the datasource.
<b>Set Initial Filter before Rebuild Cache</b>	When enabled, you can specify a field name and an expression (→ <a href="#">Expressions</a> ). Before rebuilding the cache of this datasource, Mail & Deploy will filter the field with given name to the value(s) returned by the expression. This significantly reduces the time required to rebuild the cache (or prevents it from timing out), especially for large/complex Qlik Sense apps.

Once you have entered a host and protocol type (and optionally a virtual proxy path), you can use the  button to refresh the list of apps that you have access to on the configured Qlik Sense server.




Please note that Mail & Deploy can only communicate with Qlik Sense through Windows Authentication. Therefore, the default proxy of Qlik Sense must be set to that authentication type, or you need to create a virtual proxy that uses windows authentication and use the path to that virtual proxy in Mail & Deploy.

## Qlik Sense Desktop Application

This type of datasource represents a Qlik Sense app located on a local Qlik Sense Desktop installation. The following attributes can be set:

<b>Name</b>	The name of the datasource; this is only used within Mail & Deploy to identify the datasource.
<b>Application Name</b>	The name of the Qlik Sense app that shall be represented by the datasource.
<b>Set Initial Filter before Rebuild Cache</b>	When enabled, you can specify a field name and an expression (→ <a href="#">Expressions</a> ). Before rebuilding the cache of this datasource, Mail & Deploy will filter the field with given name to the value(s) returned by the expression. This significantly reduces the time required to rebuild the cache (or prevents it from timing out), especially for large/complex Qlik Sense apps.

You can use the  button to refresh the list of apps that are available on your local Qlik Sense Desktop installation.



Please note that Mail & Deploy can only communicate with Qlik Sense Desktop if the Qlik Sense Desktop Client is started and you have logged in; otherwise, every communication attempt by Mail & Deploy will fail. Therefore, the use of Qlik Sense Desktop apps as datasources is only recommended for testing purposes and local report design development, not for actual report document creation and distribution in tasks.

# Reports

## Introduction

In Mail & Deploy, a report contains instructions about how report documents (→ [Reports and Report Documents](#)) are created. Report documents can be created based on templates (→ [Template Based Reports](#)) or using reports embedded in a datasource (→ [Datasource Embedded Reports](#)).

You can access reports by clicking *Reports* in the left menu of Mail & Deploy Client.

## Managing Reports

Reports can be managed using the following buttons:



Click this button to create a new report; a context menu will open that allows you to select which type of report you want to create.



Click this button to edit the currently selected report.



Click this button to create a copy of the currently selected report.



Click this button to delete the currently selected report.

## General

### General Attributes

Each report has the following general attributes:

<b>Name</b>	The name of the report; this is only used within Mail & Deploy to identify the report.
<b>Description</b>	The description of the report; the description is displayed in Mail & Deploy Hub (→ <a href="#">Hub</a> ) and can be requested by consumers of Mail & Deploy API (→ <a href="#">API</a> ).
<b>Condition</b>	An optional expression (→ <a href="#">Expressions</a> ) that evaluates to either <i>True</i> or <i>False</i> . If the condition is not fulfilled, the report document will not be created.
<b>Preview File Type</b>	The file type in which preview report documents of the report shall be created.
<b>Default Culture</b>	This is the default culture to apply when creating the report document (→ <a href="#">Culture Awareness</a> ). This option is not available for all report types.
<b>Allow Custom Culture</b>	<p>This setting determines if a custom culture may be supplied when requesting a report document based on the report through the Mail &amp; Deploy Hub (→ <a href="#">Hub</a>) or Mail &amp; Deploy API (→ <a href="#">API</a>).</p> <p><i>True</i>. In this case, users can provide a custom culture when requesting the report.</p> <p><i>False</i>. In this case, users can not provide a custom culture when requesting the report; the default culture will be used.</p> <p>This option is not available for all report types.</p>

## Report Parameters

By using report parameters, data can be passed on to the report by whatever initiates the execution of the task (e.g. a user who requests the creation of a report document based on the report through Mail & Deploy Hub or in a task by using a [Create Report Document Action](#)), which will then be used to influence the creation of the report document (e.g. by filtering a datasource). Each report parameter has the following attributes:

<b>Name</b>	The name of the report parameter; this is only used within Mail & Deploy to identify the report parameter.
<b>Caption</b>	This is the caption of the report parameter which will be used in Mail & Deploy Hub (→ <a href="#">Hub</a> ) when users are required to enter the value of the report parameter.
<b>Values</b>	A list of expression (→ <a href="#">Expressions</a> ) which will be used as the default values of the report parameter.

You can access the values of report parameters in expressions (→ [Expressions](#)) within the context of the report (→ [Report Parameter](#)).

### Hub Value Range Provider

These settings allow you to define, how values can be passed to the report parameter when requesting the report through the hub (→ [Hub](#)). The following types are available:

**None.** This means that there is no value range provided to the user; therefore, the user can enter free text to be passed as the value of the parameter.

**Custom Value Range.** This means that users can select one value from a pre-defined list of values (→ [Expressions](#)) to be passed as the value of the parameter. When you select this type, an expression list will appear that allows you to add expressions whose values will be presented to the user when requesting a report document.

**User Custom Property Value Range.** This type allows you to present the user a drop-down of values to choose from; the available values will be taken from a custom property of that user (→ [Custom Properties](#)). With this option you can achieve user-specific value ranges.

**User Windows Account Name.** This type will set the value of the parameter to the windows account name (in *domain\accountname* format) of the user that performs the report document request through the hub. This could – for example – be used to then filter a QlikView Document datasource's section access table (→ [Filter Datasource Field Action](#)) to the value of the parameter, so that the requested report document only contains data which the requesting user is allowed to see.

## Preparation Actions

This is a list of actions (→ [Actions](#)) which will be executed before the report document will be created. These actions can – for instance – be used to filter datasource fields or activate bookmarks in order to prepare the datasource to be in exactly the state required for the report document to contain the necessary data.

## Tags

This is a list of key-value-pairs which will be evaluated during the creation of a report document, after the preparation actions (→ [Preparation Actions](#)) have been executed. The expression (→ [Expressions](#)) that makes up the value of the tag will therefore be evaluated when the datasources involved are in a defined state. When creating a report document from the report during the execution of a task (→ [Tasks](#)) the values of these tags can be accessed and used in an expression (→ [Expressions](#)). Tags can be managed using the following buttons:



Click this button to create a new tag.



Click this button to edit the currently selected tag.



Click this button to create a copy of the currently selected tag.




Click this button to delete the currently selected tag.

Each tag has the following attributes:

<b>Name</b>	The name of the tag; this is only used within Mail & Deploy to identify the tag.
<b>Value</b>	An expression (→ <a href="#">Expressions</a> ) that evaluates to the value of the tag.

**Example:** Let's suppose there is a report that creates a report document which contains monthly statistics for a given sales region. Such a report would have a report parameter (→ [Report Parameters](#)) that can be set to the sales region to create the report document for and a preparation action (→ [Preparation Actions](#)) that filters the underlying datasource to the sales region you want to create the report document for. If such a report document would be created in the context of a task (→ [Tasks](#)) and shall then be saved to the file system (→ [Save Report Document to File System Action](#)), you would probably want to use the name of the sales region as a part of your file name or directory path; however, you cannot simply use the value of the datasource field that contains the name of the sales region in the expression used when saving the report document, because at that stage the underlying datasource might no longer be filtered to the sales region that the report document has been created for; therefore, you should add a new tag that contains the name of the sales region, because tags are evaluated at a stage where the underlying datasource is still filtered to whatever the preparation actions of the report dictate. When saving the report to the file system, you can access the tag in the expression (→ [Expressions](#)) that defines the file path to save the report document to.

## Culture Awareness

A culture is a set of rules that specify how numbers and dates are formatted ( [Windows Regional Settings](#)). Generally speaking, the culture defines in what format numbers and dates are being presented to users when they are displayed in the user interface or included in a report. However, these culture settings only apply when converting the original datatype (e.g. a date or a decimal) to a textual representation; therefore, not all report types can be culture aware. Culture awareness is handled in different report types as shown below:

**HTML Reports.** All data that is displayed in HTML reports is represented as text; therefore HTML reports do support culture awareness. Whenever a value of another data type (e.g. a date) is written to the HTML report, it is converted to text first. This conversion process uses the culture of the report.

**Microsoft Excel Reports.** In Microsoft Excel, culture awareness is handled by Excel automatically; the values are written to the Excel Workbook in their original data type (thus, no conversion that would require culture awareness is required) and each cell can have a format string that defines how dates and numbers shall be formatted. When a Microsoft Excel Workbook is opened on a computer, the culture of the user (as set in the operation system) will be used to determine how numbers and dates will be displayed to the user. Therefore, Mail & Deploy does not apply culture-specific settings to Microsoft Excel Reports and the option to set a culture is not enabled.

**Microsoft PowerPoint Reports.** All data in Microsoft PowerPoint Presentations is represented as text; therefore PowerPoint reports do support culture awareness. Whenever a value of another data type (e.g. a date) is written to the PowerPoint report, it is converted to text first. This conversion process uses the culture of the report.

**Microsoft Word Reports.** All data in Microsoft Word Documents is represented as text; therefore Word reports do support culture awareness. Whenever a value of another datatype (e.g. a date) is written to the Word report, it is converted to text first. This conversion process uses the culture of the report.

**QlikView Document Embedded Reports.** When creating a report document from a QlikView Document Embedded Report, the QlikView Desktop Client will do all the rendering and conversions required for the generation of the output Adobe PDF document. Therefore, Mail & Deploy does not apply culture-specific settings to QlikView Document Embedded Reports and the option to set a culture is not enabled.

**QlikSense Application Story Report.** When creating a report document from a Qlik Sense Application Story, the slides of that story will be imported as images, not as actual text; therefore no value conversion

to a textual representation takes place; therefore Mail & Deploy does not apply culture-specific settings to Qlik Sense Application Story Reports and the option to set a culture is not enabled.

However, even in report types with enabled culture awareness, you have to be careful as to when culture settings made in Mail & Deploy are applied to that report. Culture specific formatting is only applied, if:

- The report type supports culture awareness.
- The value written to the report is in its original datatype.

**Example:** Let's suppose you want to create a Microsoft Word report based on a QlikView Document that contains a date field. If you want to write the value of that date field into the report document, you can use a text element (→ [Text Elements](#)) that contains an expression (→ [Expressions](#)) that directly references the field of the QlikView Document that contains the date value (→ [Datasource Field Value](#)). If that is the case, when evaluating the content of the text element, a value of datatype date will be returned by the expression and needs to be converted to a text value in order to be inserted in the Microsoft Word document; therefore, culture settings made in Mail & Deploy will be applied. If, however, the expression you use as the content of the text element does not contain a direct reference to the field but rather a datasource expression (→ [Datasource Expression](#)) which automatically applies a date format to the value of the field that contains the date, then this expression will already return a formatted textual representation of the date; therefore, culture settings made in Mail & Deploy will not be applied, because Mail & Deploy treats this value as a text, not as a date.

In reports with enabled culture awareness, the culture will also be applied when converting values in expressions (→ [Formatting and Conversions](#)) except for conversions that are being done inside a datasource expression (→ [Datasource Expression](#)).

## Report Types

### CSV Reports

This type of report exports the tabular data of a single datasource object (→ [Objects](#)) into a CSV file. In addition the the general attributes (→ [General Attributes](#)), CSV reports have the following attributes:

<b>Datasource Object</b>	The datasource object (→ <a href="#">Objects</a> ) from which to export tabular data into a CSV file.
<b>Row Separator</b>	The row separator to use.
<b>Column Separator</b>	The column separator to use.
<b>Encoding</b>	The encoding to be used for the CSV file.

### Datasource Embedded Reports

This type of report is based on reports embedded in a datasource (→ [Embedded Reports](#)).

### Qlik Sense Application Story Report

This type of report create reports documents based on a story embedded in a Qlik Sense app. Mail & Deploy can create Adobe PDF documents or Microsoft PowerPoint Presentations from a story. In addition to the general attributes (→ [General Attributes](#)), Qlik Sense Application Story Reports have the following attributes:

<b>Datasource</b>	The datasource (→ <a href="#">Datasources</a> ) that contains the embedded report from which to create a report document.
<b>Story Name</b>	An expression (→ <a href="#">Expressions</a> ) that evaluates to the name of the embedded report (→ <a href="#">Embedded Reports</a> ) from which to create a report document. This name is equal to the name of the story in the Qlik Sense app.

## QlikView Document Embedded Report

This type of report creates report documents based on a report embedded in a QlikView document. These reports can be designed using the report designer of QlikView Desktop Client. Mail & Deploy can then create Adobe PDF documents from that report. In addition to the general attributes (→ [General Attributes](#)), QlikView Document Embedded Reports have the following attributes:

<b>Datasource</b>	The datasource (→ <a href="#">Datasources</a> ) that contains the embedded report from which to create a report document.
<b>Report Name</b>	An expression (→ <a href="#">Expressions</a> ) that evaluates to the name of the embedded report (→ <a href="#">Embedded Reports</a> ) from which to create a report document.

## Template Based Reports

This type of report is based on one or more templates; a template is a file in the native file format of the report type; templates of Microsoft Word reports are Microsoft Word Documents, templates of Microsoft Excel reports are Microsoft Excel Workbooks etc. Generally speaking, a report template is nothing more than a file in the native format of the report type that contains references to report elements (→ [Report Elements](#)) wherever you want Mail & Deploy to insert dynamic parts – e.g. a chart or a table read from a datasource – into the report document.

### Templates

#### General

Every template based report supports multiple templates. This is particularly useful if you have some reports which are very similar (e.g. they are based on a similar template and require similar report elements), but are still slightly different. When the creation of a report document is requested, Mail & Deploy will – by using the *Report Template Name* attribute (→ [Expressions](#)) – automatically determine which template is to be used. If there is only one template for a given report, the *Report Template Name* attribute can be empty, because in such a scenario Mail & Deploy will automatically use the one report template available.

**Example:** Let's suppose you want to create reports for all your sales managers; generally, the reports for all sales managers are identical in their structure (e.g. they all contain the same charts and tables; only the data is filtered to a different sales manager). However, you have sales managers in three different countries: Germany, the US and France and you want to send each sales manager a report in their native language. The data model of the datasource (e.g. a QlikView Document) that you base your report on has a field "LanguageCode" which contains the two-letter code of the language for each sales manager (e.g. "de" for German, "en" for English and "fr" for French). In such a scenario, the multi-template feature of Mail & Deploy is particularly useful, because you can create three templates – each in a different language – and name them "Template\_de", "Template\_en" and "Template\_fr". However, you won't have to create the required report elements (→ [Report Elements](#)) three times, because – as previously stated – the charts and tables will be the same for each language. Therefore you can reference the same report elements in all your templates. But how does Mail & Deploy know which template to use for each sales manager? The answer lies in the "Report Template Name" attribute; if you set this to an expression (→ [Expressions](#)) that returns the applicable template name by combining the constant "Template\_" (because all template names begin with that phrase) and then the value of the "LanguageCode" field of the datasource, the correct template will be chosen. Of course, this feature is not only limited to multi-language templates; you could also create different templates for private or corporate report recipients, or various currencies etc.

The Report Template Name attribute can not only return a single value but also an array of values. Mail & Deploy will then merge all specified templates into a single template on which basis the report is created.

**Example:** If you specify a report template name expression of `New String() {"TemplateName1", "TemplateName2"}` then Mail & Deploy will merge the templates TemplateName1 and TemplateName2 (in the order specified) to a single template and will then process that template and create a report document.

Templates can be managed using the following buttons:



Click this button to create a new template. A new dialog will pop up which allows you to assign a name to the newly created template.



Click this button to import a template from the file system. A new dialog will pop up which allows you to browse to the file you want to import and to assign a name to the imported template.



Click this button to edit the name of the currently selected template.



Click this button to create a copy of the currently selected template.



Click this button to delete the currently selected template.



Click this button to enter the design mode (→ [Design Mode](#)) for the currently selected template.

## Design Mode

The design mode is used to design a report template. When you click the button to enter the design mode, Mail & Deploy will open the editor the type of report template (e.g. for a Microsoft Excel template, Microsoft Excel will be launched) and re-arrange the windows so your template editor is at the right side of the screen, while Mail & Deploy is on the left side of the screen.

Once the report template editor (e.g. Microsoft Excel) has launched, you can start designing the report template by using all features of the report template editor. If you want to place report elements (→ [Report Elements](#)) in the report templates, you can simply drag and drop the report element to the location of the report template (e.g. a Microsoft Excel cell) where you want the report template to be placed.

If you want to undo all changes made since you launched the design mode, you can simply close the report template editor without saving changes. Otherwise, simply save the changes in the report template editor and close it.



Please note you can only enter the design mode of report templates if you have the required design software (e.g. Microsoft Excel if you want to design Microsoft Excel report templates) installed and licensed on the computer that runs Mail & Deploy Client (→ [Client](#)).

## Report Elements

All the parts of a template based report which are inserted dynamically during report execution by Mail & Deploy into the report template are represented by a report element.

### General Attributes

Each report element has the following general attributes:

<b>Code</b>	The code that is used to reference the report element in a report template.
<b>Name</b>	The name of the report element; this is only used within Mail & Deploy to identify the report element.
<b>Condition</b>	An optional expression (→ <a href="#">Expressions</a> ) that evaluates to either <i>True</i> or <i>False</i> . What happens if a condition is not fulfilled ( <i>False</i> ) is determined by the <i>Condition Not Fulfilled Behaviour</i> attribute.
<b>Condition Not Fulfilled Behaviour</b>	This attribute defines what shall happen if the condition is not fulfilled (i.e. returns <i>False</i> ).

*Skip Report Element.* If you select this behaviour, the report element will be skipped and not included in the final report document.



	<p><i>Skip Report Element And Delete Row.</i> If you select this behavior, the report element will be skipped and the row containing the report element will be deleted. This option is only available in Microsoft Excel Reports.</p> <p><i>Skip Report Element And Delete Worksheet.</i> If you select this behavior, the report element will be skipped and the worksheet containing the report element will be deleted. This option is only available in Microsoft Excel Reports.</p> <p><i>Skip Report Element And Delete Slide.</i> If you select this behavior, the report element will be skipped and the slide containing the report element will be deleted. This option is only available in Microsoft PowerPoint Reports.</p> <p><i>Cancel Report Document Creation.</i> If you select this behaviour, the report document creation will be cancelled and no report document will be created.</p>
Filter Check Behaviour	<p>This attribute defines what shall happen if a filter check (→ <a href="#">Filter Datasource Field Action</a>) of the report element fails.</p> <p><i>Skip Report Element.</i> If you select this behaviour, the report element will be skipped and not included in the final report document.</p> <p><i>Cancel Report Document Creation.</i> If you select this behaviour, the report document creation will be cancelled and no report document will be created.</p>

## Preparation Actions

This is a list of actions (→ [Actions](#)) which will be executed before the report document will be created. These actions can – for instance – be used to filter datasource fields or activate bookmarks in order to prepare the datasource to be in exactly the state required for the report document to contain the necessary data.

## Text Elements

A text element can be used to insert a text portion into the report template during the creation of the report document. The content of the text portion can be dynamically defined (→ [Expressions](#)). Text elements have – in addition to the general attributes (→ [General Attributes](#)) which all report elements have - the following attributes:

Content	An expression (→ <a href="#">Expressions</a> ) that evaluates to the content of the text element (i.e. the text portion that will be inserted into the report document).
---------	--

## Chart Elements

A chart element can be used to insert the graphical representation of a chart into the report template during the creation of the report document. Chart elements have – in addition to the general attributes (→ [General Attributes](#)) which all report elements have - the following attributes:

Datasource Object	The datasource object (→ <a href="#">Objects</a> ) from which you want to get the graphical representation.
Dimensions	<p>The dimensions are used to control the size in pixels of the graphical representation.</p> <p><i>Input Width and Input Height.</i> These are the width and height in pixels in which the chart shall be rendered before scaling is applied.</p>

*Scale.* The scale to apply to the rendered chart.

*Crop.* These settings can be used to cut off the rendered chart by the amount of pixels given.

*Output Width and Output Height.* These are read-only values which indicate the size in pixels of the final representation of the chart. The chart will be inserted into the report document in the size given.

## Sheet Elements

A sheet element can be used to insert the graphical representation of a sheet (e.g. a Qlik Sense sheet) into the report template during the creation of the report document. Sheet elements have – in addition to the general attributes (→ [General Attributes](#)) which all report elements have - the following attributes:

<b>Datasource Object</b>	The datasource object (→ <a href="#">Objects</a> ) from which you want to get the graphical representation. This is the sheet object of the datasource.
<b>Dimensions</b>	<p>The dimensions are used to control the size in pixels of the graphical representation.</p> <p><i>Input Width and Input Height.</i> These are the width and height in pixels in which the chart shall be rendered before scaling is applied.</p> <p><i>Scale.</i> The scale to apply to the rendered sheet.</p> <p><i>Crop.</i> These settings can be used to cut off the rendered sheet by the amount of pixels given.</p> <p><i>Output Width and Output Height.</i> These are read-only values which indicate the size in pixels of the final representation of the sheet. The chart will be inserted into the report document in the size given.</p>

## Image Elements

An image element can be used to load an image file from a URI (e.g. a local file path, a URL or a UNC path) and add it into the report template during the creation of the report document. Image elements have – in addition to the general attributes (→ [General Attributes](#)) which all report elements have - the following attributes:

<b>Uri</b>	<p>An expression (→ <a href="#">Expressions</a>) that evaluates to the URI of the image to insert into the report document. Please note, that the following conditions need to be fulfilled:</p> <ul style="list-style-type: none"> <li>➤ The file needs to be accessible for the user that runs Mail &amp; Deploy Server. The user needs at least the rights to read the file. To ensure that the preview (→ <a href="#">Preview</a>) also works, the user, that runs Mail &amp; Deploy Client needs to have read access to the file as well.</li> <li>➤ The URI needs to be reachable by Mail &amp; Deploy Server; in most cases this means that you cannot use local file paths and need to use UNC paths instead. Please note, that mapped network drives cannot be used. To ensure that the preview (→ <a href="#">Preview</a>) also works, the computer that runs Mail &amp; Deploy Client needs to be able to access the URI as well.</li> </ul>
------------	---

## Table Elements

A table element can be used to insert a tabular representation of data into the report template during the creation of the report document. Table elements have – in addition to the general attributes (→ [General Attributes](#)) which all report elements have – the following attributes:

<b>Datasource Object</b>	The datasource object (→ <a href="#">Objects</a> ) from which you want to get the tabular data.
<b>Included Cell Types</b>	Determines which cell types (→ <a href="#">Table Cell Styles</a> ) of the source table shall be included in the table that is written to the report document.
<b>Direction</b>	The direction in which the output table shall be written to the report document. Please note, that this option is not available for pivot tables.

## Column Specifications

By specifying column specifications, you can control which columns of the source table (e.g. the table in the datasource) will be included in the report and in which order. The order of the column specifications controls the order of the columns in the table that will be written to the report document. Column Specifications have the following attributes.

<b>Type</b>	<p>The type of column specifications, which can be one of the following values:</p> <p><i>All Source Columns.</i> When you use this type, the column specification will include all columns of the source table and apply its settings to them.</p> <p><i>Named Source Columns.</i> When you use this type, you can select a column of the source table by its name, if the datasource object (→ <a href="#">Objects</a>) of the table report element supports named columns.</p> <p><i>Source Column by Number.</i> When you use this type, you can specify a column number which addresses the column at that position in the source table.</p> <p><i>Source Column Range.</i> When you use this type, you can specify a start and end column number which addresses the range from the start column number to the end column number in the source table.</p>
<b>Width</b>	<p>Specifies the method by which the width of all columns addressed by the column specification shall be determined when writing the table to the report document.</p> <p><i>Do Not Set.</i> The width of the columns are not set by Mail &amp; Deploy at all.</p> <p><i>Automatic.</i> The width of the columns shall be automatically determined by the size of the content of all cells belonging to each column.</p> <p><i>Custom.</i> If you select this option, you can specify an expression (→ <a href="#">Expressions</a>) that evaluates to the custom column width in pixels.</p>
<b>Cell Type Settings</b>	<p>These settings allow you to control whether each cell of a given cell type (→ <a href="#">Table Cell Types</a>) has a custom content; if activated, you can specify an expression (→ <a href="#">Expressions</a>) that evaluates to the value of each cell of that cell type. You can also set how each cell type of each column of the column specification shall be styled (→ <a href="#">Table Cell Styles</a>) when writing the table to the report document. For each cell type you have the following options:</p>

*None.* If you choose this option, Mail & Deploy will not set any style for the given cell type; thus, whatever style is set in the report template will be kept.

*Keep Original.* If you choose this option, Mail & Deploy will keep the style of the original table in the datasource (→ [Datasources](#)). Please note, that not all style elements may be kept, because this also depends on the type of datasource.

*Specific Style.* If you choose this option, you can select a specific datatable cell style (→ [Table Cell Styles](#)) to apply to the cell type.

## Pivot Settings

If the datasource object (→ [Objects](#)) that is the source of the table report element is a pivot table, the following additional attributes can be set:

<b>Indent Rows</b>	If this attribute is enabled, all columns with left header cells (→ <a href="#">Table Cell Types</a> ) will be combined to one single column and all left header cells will receive an indent depending on the level they belong to.
<b>Merge Vertically</b>	If this attribute is enabled, consecutive dimension header cells in vertical direction of the pivot table that contain the same value will be merged. Also, you can define which vertical alignment the content of the merged cells shall have.
<b>Merge Horizontally</b>	If this attribute is enabled, consecutive dimension header cells in horizontal direction of the pivot table that contain the same value will be merged. Also, you can define which horizontal alignment the content of the merged cells shall have.

## Raw Table Elements


A raw table element can be used to insert a tabular representation of raw data. The advantage of the raw table element in comparison to the table element (→ [Table Elements](#)) is, that the raw table element is much faster; on the other hand, the following limitations apply to the raw table element:

- Raw tables and their cells have no styles
- The raw table contains the complete source table and the cell types to include (→ [Table Cell Styles](#)) and the order of exported columns cannot be changed.

Raw table elements have – in addition to the general attributes (→ [General Attributes](#)) which all report elements have – the following attributes:


<b>Datasource Object</b>	The datasource object (→ <a href="#">Objects</a> ) from which you want to get the tabular data.
<b>Exclude Columns</b>	A list of column numbers (the first column equals 1) to exclude from the table.

## trueChart Image Elements

This type of report element can be used to insert the graphical representation of a trueChart, a Qlik Sense extension available from our partner HighCoordination ( [Website](#)). trueChart image elements have – in addition to the general attributes (→ [General Attributes](#)) which all report elements have – the following attributes:

<b>Datasource Object</b>	The datasource object (→ <a href="#">Objects</a> ) from which you want to get the graphical representation. This represents the trueChart object.
<b>Input Dimensions</b>	These settings control the input dimensions of the trueChart object.
<b>Extraction Settings</b>	This control the type of extraction of the graphical representation of the trueChart.


*Extract the full chart.* If you choose this setting, the complete trueChart will be exported. You can specify additional settings to control how many pixels shall be removed from the top, bottom, left and right side of the extracted graphic. The chart will be placed into the report element with the output size.

*Extract a grid of sub-charts.* If you choose this setting, you can define a grid size by using the  button. Then you can click on any of the grid cells and assign the ID of the sub-chart you want to place on that position as well as additional settings to control how many pixels shall be removed from the top, bottom, left and right side of the extracted sub-chart graphic. The actual graphic that will be inserted into the report template will be constructed from all grid cells; its output size, however, cannot be determined at design time.



Please note, that any trueChart Menu that is required by the trueChart object you want to export needs to be initialized (→ [Initialize Datasource Object Action](#)) before the trueChart can be processed.

### trueChart Table Elements

This type of report element can be used insert the tabular representation of a trueChart, a Qlik Sense extension available from our partner HighCoordination ( [Website](#)). trueChart table elements have – in addition to the general attributes (→ [General Attributes](#)) which all report elements have - the following attributes:

<b>Datasource Object</b>	The datasource object (→ <a href="#">Objects</a> ) from which you want to get the tabular representation. Only trueChart datasource objects may be selected.
<b>Sub-Chart ID</b>	The ID of the sub-chart of the trueChart. Only the table of the sub-chart will be inserted into the report template.



Please note, that any trueChart Menu that is required by the trueChart object you want to export needs to be initialized (→ [Initialize Datasource Object Action](#)) before the trueChart can be processed.

### Repeater Elements

A repeater element can be used to repeat a certain part of the report – depending on the report type – for each value of a given set of levels.

#### Levels

Repeaters have one or more levels, each level representing a set of values to repeat over. Each level has the following attributes:

<b>Type</b>	The type of level.
	<i>Table Column.</i> This setting can be used to repeat over a set of distinct values read from a column from the table of a datasource object.

	<p><i>Datasource Field.</i> This setting can be used to repeat over a set of distinct values read from a datasource field.</p> <p><i>Expression.</i> This setting can be used to repeat over a set of distinct values read from an expression.</p>
<b>Datasource Field</b>	The datasource field (→ <a href="#">Fields</a> ) from which to query values to repeat. This setting is only applicable if the type of level is set to <i>Datasource Field</i> .
<b>Field Value Selector</b>	<p>The field value selector that determines which values of the datasource field shall be used for the repeater. This setting is only applicable if the type of level is set to <i>Datasource Field</i>.</p> <p><i>All Values.</i> If you use this selector, all values of the datasource field will be used for the repeater, regardless of the current filter state of the datasource.</p> <p><i>Applicable Values.</i> If you use this selector, only those values of the datasource field will be used for the repeater which are compatible to the current filter state of the datasource.</p> <p><i>Filtered Values.</i> If you use this selector, only those values of the datasource field will be used for the repeater to which the datasource field is explicitly filtered.</p>
<b>Sort Order</b>	<p>This setting determines in which order the values will be processed by the repeater. This setting is only applicable if the type of level is set to <i>Datasource Field</i>.</p> <p><i>None.</i> No specific sort order will be used and the values will be used in the exact order in which they have been read from the datasource.</p> <p><i>Ascending.</i> The values of the datasource field will be processed by the repeater in ascending order.</p> <p><i>Descending.</i> The values of the datasource field will be processed by the repeater in descending order.</p>
<b>Datasource Object</b>	The datasource object from which a table column is used for the repetition. This setting is only applicable if the type of level is set to <i>Table Column</i> .
<b>Column Number</b>	The number of the column of the table of a datasource object from which the values to repeat are read. This setting is only applicable if the type of level is set to <i>Table Column</i> .
<b>Fields to Filter</b>	A list of datasource fields to filter before the report elements (→ <a href="#">Report Elements</a> ) contained in the repeated section of the report will be evaluated.

**Example:** Let's suppose you have a Microsoft Excel Report (→ [Microsoft Excel Reports](#)) and you want to repeat a worksheet for each value of the field "Year" in the datasource. In that case, you would create a repeater with a level that queries all values of the field "Year" and place it in the report template; in the list of fields to filter, you would also add the field "Year", because you want each repeated section to be filtered to the year of the repeated worksheet. The reason why you need to explicitly specify the list of fields to filter (instead of Mail & Deploy automatically filtering that repeated field) is, that you could have report elements from various datasources in the repeated section and therefore you might need to filter fields different from the field the repeated values are taken from.

## Settings

If the repeater is used in the context of a Microsoft Excel or Microsoft PowerPoint report, an additional setting *Number Of Repeated Items* may be set. This setting controls how many worksheets (in the case

of a Microsoft Excel report) or slides (in the case of a Microsoft PowerPoint report) shall be repeated per repeated value.

**Example:** Let's suppose you have a Microsoft Excel Report (→ [Microsoft Excel Reports](#)) and you want to repeat a worksheet and the next worksheet for each repeated value. In that case you have to place the repeater element on the first worksheet to be repeated and set "Number Of Repeated Items" to 2.

## HTML Reports

This report type is based on templates in HTML format; report documents based on an HTML report may be created in the following file formats:

- HTML Document
- Adobe PDF Document
- Adobe PDF/A Document

The following tables gives information about report elements (→ [Report Elements](#)) specifically applying when being used in HTML reports:

<a href="#">Text Elements</a>	Text elements are referenced in HTML templates by using the following format:  %%report_element_code%%  They can be placed anywhere in the HTML template.
<a href="#">Chart Elements</a> <a href="#">Image Elements</a> <a href="#">trueChart Image Elements</a>	These elements are referenced in HTML templates by using the following format:  %%report_element_code%%  They can be placed anywhere in the HTML template.
<a href="#">Table Elements</a> <a href="#">Raw Table Element</a> <a href="#">trueChart Table Elements</a>	Table elements are referenced in HTML templates by using the following format:  %%report_element_code%%  They can be placed anywhere in the HTML template.
<a href="#">Repeater Elements</a>	Repeater elements are referenced in the Microsoft Excel template by using the following format:  {{report_element_code}} ... {{/report_element_code}}  The part of the HTML document that lies between these two tags will be repeated. You can place as many repeater element as you like into an HTML template and you can also nest repeater elements.

HTML reports have – in addition to the general attributes (→ [General Attributes](#)), that every report type has – the following attributes:

<b>Page Layout</b>	The page layout to use when exporting the HTML document as Adobe PDF Document or Adobe PDF/A Document.
<b>Page Orientation</b>	The page orientation to use when exporting the HTML document as Adobe PDF Document or Adobe PDF/A Document.

## Microsoft Excel Reports

This report type is based on templates in Microsoft Excel Workbook format; report documents based on a Microsoft Excel report may be created in the following file formats:

- Microsoft Excel Workbook
- Adobe PDF Document
- Adobe PDF/A Document

The following tables gives information about report elements (→ [Report Elements](#)) specifically applying when being used in Microsoft Excel Reports:

<p><a href="#">Text Elements</a></p>	<p>Text elements are referenced in Microsoft Excel templates by using the following format:</p> <p style="text-align: center;">%%report_element_code%%</p> <p>They can be used in any cell as well as the title of a worksheet. However, text elements cannot be a sub-text of a cell; the complete cell must consist of the output of a text element.</p>
<p><a href="#">Chart Elements</a>  <a href="#">Sheet Elements</a>  <a href="#">Image Elements</a>  <a href="#">trueChart Image Elements</a></p>	<p>These elements are referenced in Microsoft Excel templates by using the following format:</p> <p style="text-align: center;">%%report_element_code%%</p> <p>They can be used in any cell. However, these elements cannot be a sub-text of a cell; the complete cell must consist of the output of such an element.</p>
<p><a href="#">Table Elements</a>  <a href="#">Raw Table Element</a>  <a href="#">trueChart Table Elements</a></p>	<p>Table elements are referenced in Microsoft Excel templates by using the following format:</p> <p style="text-align: center;">%%report_element_code%%</p> <p>They can be used in any cell. However, table elements cannot be a sub-text of a cell; the complete cell must consist of the reference to a table element. Table elements have the following additional attributes when used in Microsoft Excel reports:</p> <p><i>Insertion Mode.</i> This mode determines how the table will be written to the Microsoft Excel worksheet.</p> <p><i>Overwrite.</i> If you set the insertion mode to this value, the left upper cell of the table will be written to the cell that contains the reference to the report element. All other cells of the table will then be exported to the Microsoft Excel worksheet, overwriting any content that may be contained in the overwritten cells.</p> <p><i>Insert Rows.</i> If you set the insertion mode to this value, starting from row two of the table, a full row will be inserted into the Microsoft Excel worksheet, thus moving all content below that row down.</p> <p><i>Insert Cells.</i> If you set the insertion mode to this value, starting from row two of the table, a number of cells (which equal the number of columns to be written to the Microsoft Excel worksheet) will be inserted, thus moving all content within those columns below that row down.</p>



*Copy Rows.* If you set the insertion mode to this value, the row that contains the report element reference will be copied for each row of the exported table. This allows you to add excel formulas in cells right of the exported table which will then be copied (with their references updated).

*Delete Row Below.* If you activate this option, the row immediately below the last exported row of the table, will be deleted from the Microsoft Excel worksheet after the table has been exported.

### Repeater Elements

Repeater elements can be used in two different ways in Microsoft Excel.

**Sheet Repeater.** If you want to repeat a sheet, you can place the following reference to the report element to any location of the Microsoft Excel worksheet that you want to be repeated:

```
{{report_element_code}}
```

You can only place one sheet repeater per Microsoft Excel worksheet.

**Row Repeater.** If you want to repeat a certain number of rows of a worksheet, you can put the following report element references to the worksheet:

```
{{report_element_code}}
{/report_element_code}}
```

Every row in-between these references will be repeated and the two rows that contain the start and end tag will be deleted afterwards. You can have multiple row repeaters per Microsoft Excel worksheet and it is also possible to nest them.

## Microsoft PowerPoint Reports

This report type is based on templates in Microsoft PowerPoint Presentation format; report documents based on a Microsoft PowerPoint report may be created in the following file formats:

- Microsoft PowerPoint Presentation
- Adobe PDF Document
- Adobe PDF/A Document

The following tables gives information about report elements (→ [Report Elements](#)) specifically applying when being used in Microsoft PowerPoint Reports:

### Text Elements

Text elements are referenced in Microsoft PowerPoint templates by using the following format:

```
%%report_element_code%%
```

They can be used in any text-box and can also be a sub-text of the text-box.

### Chart Elements

### Sheet Elements

### Image Elements

### trueChart Image Elements

These elements are referenced in Microsoft PowerPoint templates by using the following format:

```
%%report_element_code%%
```

	They can be used in any text-box and the associated text-box must not have any other content than the reference to the element.
<a href="#">Table Elements</a> <a href="#">Raw Table Element</a> <a href="#">trueChart Table Elements</a>	<p>Table elements are referenced in Microsoft PowerPoint templates by using the following format:</p> <p style="text-align: center;">%%report_element_code%%</p> <p>They can be used in any text-box and the associated text-box must not have any other content than the reference to the table element.</p> <p>Alternatively, table elements and raw table elements may also be used to feed a native PowerPoint chart with data by putting the reference to the element into the alternative text of the chart. The PowerPoint chart needs to have at least one category and at least one series; Mail &amp; Deploy will automatically add new categories and series if the data contains more than one row and column.</p>
<a href="#">Repeater Elements</a>	<p>Repeater elements are referenced in Microsoft PowerPoint templates by using the following format:</p> <p style="text-align: center;">{{report_element_code}}</p> <p>The expression can be placed in any text-box in the Microsoft PowerPoint template and the associated text-box must not have any other content than the reference to the repeater element. The text-box can be placed anywhere on the slide to be repeated and each slide can only have one reference to a repeater element.</p>

## Microsoft Word Reports

This report type is based on templates in Microsoft Word Document format; report documents based on a Microsoft Word report may be created in the following file formats:

- Microsoft Word Document
- Adobe PDF Document
- Adobe PDF/A Document

The following tables gives information about report elements (→ [Report Elements](#)) specifically applying when being used in Microsoft Word Reports:

<a href="#">Text Elements</a>	<p>Text elements are referenced in Microsoft Word templates by using the following format:</p> <p style="text-align: center;">%%report_element_code%%</p> <p>They can be placed anywhere in the Microsoft Word document, including headers and footers.</p>
<a href="#">Chart Elements</a> <a href="#">Sheet Elements</a> <a href="#">Image Elements</a> <a href="#">trueChart Image Elements</a>	<p>Chart elements are referenced in Microsoft Word templates by using the following format:</p> <p style="text-align: center;">%%report_element_code%%</p> <p>They can be placed anywhere in the Microsoft Word document, including headers and footers.</p>
<a href="#">Table Elements</a> <a href="#">trueChart Table Elements</a>	<p>Table elements are referenced in Microsoft Word templates by using the following format:</p>

---

```
%%report_element_code%%
```

They can be placed anywhere in the Microsoft Word document, including headers and footers. These elements have the following additional attributes when used in Microsoft Word reports:

*Repeat Header Row.* This settings determines whether top header cells (→ [Table Cell Types](#)) are being repeated on page break.

*Keep Rows On Same Page.* This setting determines whether rows of the exported table shall be kept within the same page, if possible.

*Fit To Available Width.* If this setting is activated, the table will be fitted to the available width (e.g. the page width).

*Remove Empty Paragraph Below.* If this setting is activated, an empty paragraph that Microsoft Word creates below a table, will be removed (e.g. to prevent unwanted page breaks triggered by the empty paragraph).

*Write Into Existing Table.* If this setting is activated, you can write the reference to the table element (see above) in the top-left corner of an existing Microsoft Word table in the template. That table needs to have the same number of columns as the table you want to export. Instead of creating a new table, Mail & Deploy will fill the existing table; if the existing table does not have enough rows, the last row of the table will be cloned until the required number of rows is reached. This allows you to set column widths and styles within the Word template rather than having to format everything using styles and/or column width expressions.

---

### Raw Table Element

Raw table elements are referenced in Microsoft Word templates by using the following format:

```
%%report_element_code%%
```

They can be placed anywhere in the Microsoft Word document, including headers and footers.

---

### Repeater Elements

Repeater elements are referenced in the Microsoft Word template by using the following format:


```
{{report_element_code}}
...
{{/report_element_code}}
```

The part of the Microsoft Word document that lies between these two tags will be repeated. You can place as many repeater element as you like into a Microsoft Word template and you can also nest repeater elements.

## Fields

Mail & Deploy also supports fields, such as a table of contents, page numbers etc. in the Microsoft Word document and will automatically update them as the content of the document changes throughout the creation of the report document.

## Preview

During the design process of a report, you very often will want to know what the report looks like if it were created in the development stage that it currently is in. You can launch such a preview using the  button in the lower left corner of the report window. A new window will pop up where you can see

the log file during the creation of the resulting report document. If the creation of the report document finishes successfully, the log file window will automatically close and the preview will be launched in the associated application on your computer (e.g. PDF files will be opened in a PDF viewer, Microsoft Excel files will be opened in Microsoft Excel etc.). Before creating a new preview, you will have to close the previously generated preview. If the creation of the report document fails, the log file will not automatically close and you can see why the creation of the preview has failed.

However, there are a few things to consider when creating a preview:

- The preview will be created using the user (→ [Users](#)) with which you are currently authenticated in Mail & Deploy. That means, that all resources required for the creation of the preview (e.g. the connection to the datasource) need to be available to that user.

**Example:** If you preview a report that is based on a Qlik Sense app and a QlikView Document, the user that launches the preview needs to have the required access to the Qlik Sense app and the QlikView Document.

- The preview will be created locally by Mail & Deploy Client, not by Mail & Deploy Server (as any other report document creation and task execution would). Therefore, all resources required for the creation of the preview need to be available on the computer that runs Mail & Deploy Client.


**Example:** If you preview a report that is based on a QlikView Document, the computer that runs Mail & Deploy Client (and the user, that runs Mail & Deploy Client) requires a licensed QlikView Desktop Client installed and access to the associated QlikView Document via the file system.

- The application to view the resulting preview report document needs to be installed on the computer that runs Mail & Deploy Client and available to the user running Mail & Deploy Client.

**Example:** If you preview a Microsoft Excel report, Microsoft Excel (or any other viewer for Microsoft Excel Workbooks) must be installed on the computer and available to the user that runs the preview. If you preview a report that creates an Adobe PDF Document, a PDF viewer application is required.

- Report Parameters (→ [Report Parameters](#)) will be set to their default values when creating the preview.
- The default culture (→ [Culture Awareness](#)), if applicable, will be used when creating the preview.
- The preview file type will be used when creating the preview.


## Hub Authorizations

If you want users to be able to request the report through Mail & Deploy Hub (→ [Hub](#)), you can manage hub authorizations (→ [Report Authorizations](#)) by clicking the  button in the lower left corner of the report window. This button is only available for reports that have been saved at least once.

## Replacing Datasources

Sometimes during the design process of a report, it can be very helpful to replace all references of a report to a datasource with references to another datasource.

**Example:** If you work in an environment where you have access to a Qlik Sense Test Server and a Qlik Sense Production Server, then it might be helpful, if you could easily change all references of your report to the datasource on the test server with references to the datasource on the production server or vice versa. This way, you can test the report for both servers without having to manually duplicate the report and change datasource references by hand.

You can replace references to a datasource by clicking the  button; you can then select the datasource whose references shall be replaced as well as the datasource by which references shall be replaced.

# Tasks

## Introduction

In Mail & Deploy, a task is a collection of actions (→ [Actions](#)) that can be executed automatically on a regular basis (→ [Execution Plans](#)), by users requesting the execution (→ [Hub](#)) or by a 3<sup>rd</sup> party application requesting the execution (→ [API](#)).

You can access tasks by clicking *Tasks* in the left menu of Mail & Deploy Client. Each task has one of the following images by which it is categorized:



Tasks with this image do not have any planned execution in the future based on an execution plan (→ [Execution Plans](#)).



Tasks with this image do have at least one planned execution in the future based on an execution plan (→ [Execution Plans](#)).

## Managing Tasks

Tasks can be managed using the following buttons:



Click this button to create a new task.



Click this button to edit the currently selected task.



Click this button to create a copy of the currently selected task.



Click this button to delete the currently selected task.



Click this button to execute the task.

## Attributes

Each task has the following attributes:

<b>Name</b>	The name of the task; this is only used within Mail & Deploy to identify the task.
<b>Description</b>	The description of the task; the description is displayed in Mail & Deploy Hub (→ <a href="#">Hub</a> ) and can be requested by consumers of Mail & Deploy API (→ <a href="#">API</a> ).
<b>Category</b>	The category of the task; this is used in the list of tasks to categorize tasks in groups.
<b>Exception Behaviour</b>	Determines what Mail & Deploy shall do if the execution of one of the actions of the task causes an exception (→ <a href="#">Exception Handling</a> ).

*Do Not Handle.* If you select this behaviour, any exception caught by the task will lead to the task failing.

*Cancel Task.* If you select this behaviour, any exception caught by the task will lead to the task to be cancelled. Technically, this means, that the task did not fail, because cancelling a task is a controlled means of ending a task.

*Continue With Next Action.* If you select this behaviour, any exception caught by the task will lead to the next action to be executed.

## Task Parameters

By using task parameters, data can be passed on to the task by whatever initiates the execution of the task (e.g. a user who requests the execution of the task through Mail & Deploy Hub). Each report parameter has the following attributes:

<b>Name</b>	The name of the task parameter; this is only used within Mail & Deploy to identify the task parameter.
<b>Caption</b>	This is the caption of the task parameter which will be used in Mail & Deploy Hub (→ <a href="#">Hub</a> ) when users are required to enter the value of the task parameter.
<b>Values</b>	A list of expression (→ <a href="#">Expressions</a> ) which will be used as the default values of the task parameter.




You can access the values of task parameters in expressions (→ [Expressions](#)) within the context of the task (→ [Task Parameter](#)).

## Actions

This is a list of actions (→ [Actions](#)) that make up the task; whenever the task is being executed, these actions will be executed in sequential order.

## Execution Plans

By using execution plans you can control when the task shall be executed automatically by Mail & Deploy Server. You can manage execution plans by using the following buttons:

	Click this button to create a new execution plan.
	Click this button to edit the currently selected execution plan.
	Click this button to delete the currently selected execution plan.

The following attributes can be set:

<b>Time Zone</b>	The time zone to apply to the execution plan; all times in the execution plan are always local times according to this time zone.
<b>Start Date</b>	The date-time where you want the next execution of the task to take place (→ <a href="#">Execution Sequence</a> ).
<b>End Date</b>	The end date is optional; if enabled, you can specify until when the execution plan shall be valid. After the end date, the task will no longer be executed on the basis of this execution plan.
<b>Maximum Delay</b>	The maximum delay to apply; if the scheduled next execution of the task according to the execution plan lies further in the past than the maximum delay, the task execution will be skipped and the next one in the sequence will be scheduled, until the first scheduled execution is found that is either in the future or in the past, but within the maximum delay (→ <a href="#">Execution Sequence</a> ).
<b>Frequency</b>	The frequency determines how often the task is due to be executed.

*Daily.* If you choose this frequency, the task will be executed on a daily basis.

*Weekdays.* If you choose this frequency, the task will be executed on the specified weekdays.

*Weekly.* If you choose this frequency, the task will be executed on a weekly basis.

*Monthly.* If you choose this frequency, the task will be executed on a monthly basis (this does not equal a 30-day period, but rather to a full month; e.g. April 3<sup>rd</sup> plus 1 month will be May 3<sup>rd</sup>). You can optionally choose to stick the execution to the end of each month.

*Quarterly.* If you choose this frequency, the task will be executed on a quarterly basis, which equals a 3-month period.

*Yearly.* If you choose this frequency, the task will be executed on a yearly basis (this does not equal a 365-day period, but rather to a full year; e.g. April 3<sup>rd</sup> 2018 plus 1 year will be April 3<sup>rd</sup> 2019; leap years will automatically be recognized by Mail & Deploy).

*Custom.* If you choose this frequency, the task will be executed in intervals as specified. However, you can limit the task execution to specific weekdays.

*Calendar.* If you choose this frequency, you can select a calendar (→ [Calendars](#)) that contains the date-times on which to perform an execution. If you enable *Custom Time* then you can specify a custom time which will override the ones contained in the calendar; only the dates of the calendar combined with your custom time will be used then.

## Execution Sequence

The logic by which the executions based on execution plans are calculated by Mail & Deploy works as follows: the start date of an execution plan needs to be set to the next date-time at which you want the task to be executed on the basis of the execution plan.

Once the task is executed by the execution engine of Mail & Deploy Server (), the date-time for the next execution will be calculated by adding the frequency to the execution date.

**Example:** Let's suppose you want the task to be executed every Monday at 9:00 AM. To achieve this, you need to set the start date of the execution plan to 9:00 AM of the following Monday and the frequency to "Weekly". Monday, 9:00 AM the task will be executed and the next execution date-time will be calculated by adding the frequency (= 1 Week) to the previous planned execution date (which in this case equals the following Monday, 9:00 AM).

If, for whatever reason – for instance, if Mail & Deploy did not have enough execution slots or Mail & Deploy Server was not running at the time of the planned execution – the execution of a task could not be performed at the planned execution date-time, the *Maximum Delay* attribute will be applied. If a free execution slot becomes available or Mail & Deploy Server is running again, then all task executions which could not be completed at the planned execution date-time will only be caught up, if these planned execution date-times are not longer than *Maximum Delay* in the past.

**Example:** Let's suppose you have set up a task that runs daily at 9:00 AM and the "Maximum Delay" attribute is set to 2 days. The last execution of that task has taken place on June 1<sup>st</sup> and then the server that runs Mail & Deploy Server is shut down for maintenance reasons. On June 4<sup>th</sup>, 10:00 AM, the server is started again. We can see, that the executions of that task on June 2<sup>nd</sup>, June 3<sup>rd</sup> and June 4<sup>th</sup> have been processed, because the server was down at that time. Mail & Deploy no identifies the earliest of these executions date-times which lie within the "Maximum Delay" attribute. June 2<sup>nd</sup> 9:00 AM is the first execution date-time to be evaluated, because it was the first one that has not been processed: Since the server was started again on June 4<sup>th</sup> 10:00 AM, June 2<sup>nd</sup> 9:00 AM lies 2 days and 1 hour in the past; this exceed the maximum delay, which was set to 2 days. Therefore, this execution is skipped. Then, the next scheduled execution date-time (June 3<sup>rd</sup> 9:00 AM) will be evaluated; this lies within the maximum delay of 2 days and therefore, the task execution will be initiated once Mail & Deploy server is up and running again.



The same applies to the scheduled execution on June 4<sup>th</sup> 9:00 AM. The next task execution will then be scheduled for June 5<sup>th</sup> 9:00 AM.

## Parallelization

By using parallelization, a single task can be split into multiple jobs that will be executed simultaneously (→ [Implementation](#)). The following attributes can be set:

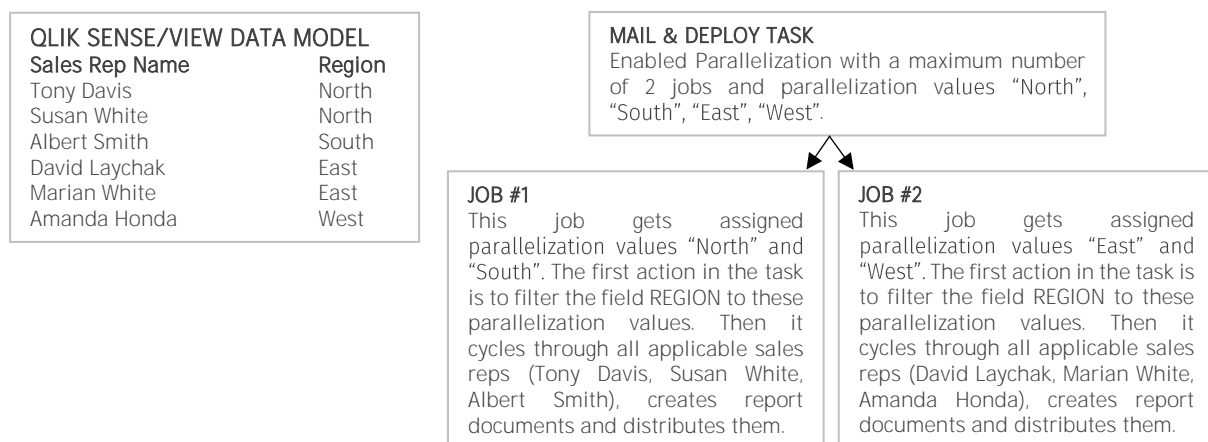
<b>Enable Parallelization</b>	This setting controls whether parallelization shall be activated for this task.
<b>Maximum Number of Jobs</b>	This is the maximum number of parallel jobs that the task shall be split into.
<b>Parallelization Values</b>	This is a list of expressions (→ <a href="#">Expressions</a> ) that will be assigned to the parallel jobs.

## Implementation

If a task is configured to support parallelization, a maximum number of jobs and a list of parallelization values is supplied. Mail & Deploy – when executing the task – assigns these values to individual jobs and then executes these jobs.

**Example:** Let's suppose you have a Qlik Sense App that contains 4 Sales Regions: North, East, South, and West. Each sales region has sales representatives assigned and you want to create one report for each sales representative for each region. If you do not use parallelization, you could create a task that cycles (→ [Cycle Action](#)) through all sales representatives, creates a report document and distributes it. However, this means, that these report documents will be created sequentially, which could take a rather long time. Another approach would be to use parallelization: if you enable parallelization for that task, you can specify the maximum number of jobs to use; if we set this number to be 2, the task will be split into two jobs. In the list of parallelization values, you can then supply expressions which will be assigned to each job. We can add our sales regions here: North, East, South, and West, so we have 4 parallelization values. When Mail & Deploy executes the task, it will assign these 4 values to the 2 jobs into which the task will be split: job number 1 will get North and East, job number 2 will get South and West. In the action list of the task, you can then filter the sales region field of your Qlik Sense app to the values of the parallelization parameter by filtering to a dynamic value in an expression (→ [Parallelization Parameter](#)). After that filter action, you can – just as you would without parallelization – add your cycle that cycles through all currently applicable sales representatives, creates a report document and distributes it.

A more detailed explanation about what's going on in the background: by enabling parallelization for a maximum number of 2 jobs, you "force" Mail & Deploy to split the task into 2 jobs (which means to parallel executions). The list of parallelization values is used to supply parameters to these individual jobs. When the task is executed by Mail & Deploy, it first creates two jobs. Then it goes through the list of parallelization values and assigns the values you provided for each job (so that each job gets its share of the complete list of parallelization values). Then the individual jobs will be executed. If you then use a filter action in your task, you can access the list of parallelization values that has been assigned to the job and then filter the Qlik Sense data model accordingly. Here's a flow chart of what happens:



If – in the above example – you would set the maximum number of jobs to 4, then each job would get exactly one region assigned (and would therefore only process sales representatives of that region). Mail & Deploy assigns these parallelization values automatically and it does not have to be the case that every job has the same number of parallelization values applied; for instance, you could set the maximum number of jobs to 3; Mail & Deploy will then assign two regions to one job and one regions for each of the other two jobs.

## E-Mail Alert Settings

By using the e-mail alert feature of Mail & Deploy, users can automatically be notified by e-mail if a task execution does not successfully complete. The following attributes can be set:

Trigger	<p>This setting controls which conditions need to be fulfilled (→ <a href="#">Exception Handling</a>) in order to send the notification e-mail to the specified users.</p> <p><i>None.</i> If you set this trigger, a notification e-mail is never sent.</p> <p><i>Task Execution Result Contains Any Exception.</i> If you set this trigger, a notification e-mail will be sent if the log file (→ <a href="#">Task Execution Logs</a>) of the task execution contains any exception.</p> <p><i>Task Execution Result Contains Any Warning.</i> If you set this trigger, a notification e-mail will be sent if the log file (→ <a href="#">Task Execution Logs</a>) of the task execution contains any exception.</p> <p><i>Task Execution Failed.</i> If you set this trigger, a notification e-mail will be sent if the execution of the task has failed.</p>
E-Mail Server	The e-mail server to use to send the e-mail (→ <a href="#">E-Mail Servers</a> ).
Sender E-Mail Address	The sender e-mail address to use to send the e-mail.
Reply-To E-Mail Address	The reply-to e-mail address to use.

## Recipients

This is a list of users and/or user groups (→ [Users and User Groups](#)) to send the notification e-mail to, if the conditions defined by the attribute *Trigger* are fulfilled. The recipient list is only visible if the attribute *Trigger* is not set to *None*. Recipients can be managed using the following buttons:



Click this button to add a new recipient; a context menu will open that allows you to select which type of recipient you want to create.



Click this button to remove the currently selected recipient.

## Hub Authorizations

If you want users to be able to request the execution of the task through Mail & Deploy Hub (→ [Hub](#)), you can manage hub authorizations (→ [Task Authorizations](#)) by clicking the button in the lower left corner of the task window. This button is only available for tasks that have been saved at least once.

## Exception Handling

Whenever something unexpectedly goes wrong during the execution of a task, an exception is raised. When creating a task in Mail & Deploy, you can control how these exceptions will be handled.

**Example:** Let's suppose you have set up a task that distributes a report document by e-mail (→ [Distribute by E-Mail Action](#)). If the e-mail server is not reachable during the execution of the task, an exception will be raised.

In order to understand how exceptions are handled in Mail & Deploy, it's important to understand the hierarchical structure of tasks; all actions within a task are organized in a hierarchy that consists of *Action Containers* and *Actions*. These are organized in hierarchies like the following:

```

ACTION CONTAINER #1
  ACTION #1.1
  
```

```

ACTION #1.2
ACTION CONTAINER #1.3
    ACTION #1.3.1
    ACTION #1.3.2
    ACTION #1.3.3

```

Any exception that occurs during the execution of an action will be passed on to the parent action container of that action. Each action container has an *Exception Behaviour* attribute that dictates how an exception shall be handled; the available behaviours depend on the type of action container, but generally speaking there are two main ways to react to an exception:

- The action container refuses to handle the exception; in that case, that exception will be passed on to the parent action container.
- The action container handles the exception; in that case, the action container will do what the *Exception Behaviour* dictates and the exception will then be dismissed and will not affect the remainder of the execution.

**Example:** Let's have a look at the action hierarchy depicted above; if an exception occurs in Action #1.3.2 it will be passed on to Action Container #1.3, because that is the parent action container of Action #1.3.2. Let's suppose the exception behaviour of Action Container #1.3 is set to "Continue with next Action"; in that case, the action container will dismiss the exception and simply continue with the execution of Action #1.3.3. If, however, the exception behaviour of Action Container #1.3 is set to "Do Not Handle", then the exception will not be handled at all; instead, the exception will be passed on to Action Container #1, which will have an exception behaviour of its own that will control how that exception will be handled.

If an exception has reached the task itself – which means that either a direct sub-action of the task has raised the exception or that all action containers have not handled the exception –, the exception cannot be passed on to a parent action container, because the task is already the top container of the hierarchy. In that case, the exception behaviour of the task dictates whether the task execution will be regarded as successful, cancelled, or failed; the logic behind this determination is as follows:

- If no exception occurs during the execution of a task, the execution of the task will be regarded as *Successful*.
- If an exception occurs during the execution of a task and that exception is not handled by any action container within the task and the task has an exception behaviour of *Do Not Handle*, the execution of the task will be regarded as *Failed*.
- If an exception occurs during the execution of a task and that exception is not handled by any action container within the task and the task has an exception behaviour of *Cancel Task*, the execution of the task will be regarded as *Cancelled*.
- If an exception occurs during the execution of a task and that exception is either handled by an action container within the task or it's passed on to the task and the task has an exception behaviour of *Continue With Next Action*, the task will simply continue the execution and the execution of the task will be regarded as *Successful*.

## Examples

The following examples will demonstrate how exception handling works in several scenarios.

### Example 1

Let's suppose we have set up the following task:

```

TASK (Exception Behaviour=Do Not Handle)
    FILTER DATASOURCE FIELD ACTION
    CYCLE (Exception Behaviour=Do Not Handle)
        CREATE REPORT DOCUMENT ACTION
        SAVE REPORT DOCUMENT TO FILE SYSTEM ACTION
    END CYCLE

```

Let's suppose the an exception occurs when saving the report document to the file system, which is the last sub-action of the cycle action container. That exception will be passed on to the parent action container of the current action; and that is the cycle. The cycle has an exception behaviour of *Do Not Handle*, which means that the cycle does not handle the exception at all and passes it on to its parent action container. The parent action container of the cycle is the task and has an exception behaviour of *Do Not Handle*. That means that the task execution will immediately stop and the execution will be regarded as *Failed*, because an exception has occurred and has not been handled by any action container.

## Example 2

Let's suppose we have set up the following task:

```
TASK (Exception Behaviour=Do Not Handle)
  FILTER DATASOURCE FIELD ACTION
  CYCLE (Exception Behaviour=Continue With Next Action)
    CREATE REPORT DOCUMENT ACTION
    SAVE REPORT DOCUMENT TO FILE SYSTEM ACTION
  END CYCLE
```

Let's suppose the an exception occurs when creating the report document, which is the first sub-action of the cycle action container. That exception will be passed on to the parent action container of the current action; and that is the cycle. The cycle has an exception behaviour of *Continue With Next Action*, which means that the cycle will simply continue with the next action after the one that has raised the exception; in other words: the cycle will continue with the execution of the action that saves a report document to the file system. If the task then completes, its execution will be regarded as *Successful*.

## Example 3

Let's suppose we have set up the following task:

```
TASK (Exception Behaviour=Cancel Task)
  FILTER DATASOURCE FIELD ACTION
  CYCLE (Exception Behaviour=Do Not Handle)
    CREATE REPORT DOCUMENT ACTION
    SAVE REPORT DOCUMENT TO FILE SYSTEM ACTION
  END CYCLE
```

Let's suppose the an exception occurs when creating the report document, which is the first sub-action of the cycle action container. That exception will be passed on to the parent action container of the current action; and that is the cycle. The cycle has an exception behaviour of *Do Not Handle*, which means that the cycle does not handle the exception at all and passes it on to its parent action container. The parent action container of the cycle is the task and has an exception behaviour of *Cancel Task*, which means that the execution of the task will immediately be stopped and will be regarded as *Cancelled*.

## Example 4

Let's suppose we have set up the following task:

```
TASK (Exception Behaviour=Continue With Next Action)
  FILTER DATASOURCE FIELD ACTION
  CYCLE (Exception Behaviour=Do Not Handle)
    CREATE REPORT DOCUMENT ACTION
    SAVE REPORT DOCUMENT TO FILE SYSTEM ACTION
  END CYCLE
  DISTRIBUTE BY E-MAIL
```

Let's suppose the an exception occurs when creating the report document, which is the first sub-action of the cycle action container. That exception will be passed on to the parent action container


of the current action; and that is the cycle. The cycle has an exception behaviour of *Do Not Handle*, which means that the cycle does not handle the exception at all and passes it on to its parent action container. The parent action container of the cycle is the task and has an exception behaviour of *Continue With Next Action*, which means that the execution will continue with the action that distributes by e-mail, because that is the next action after the one from which the exception has been passed on (e.g. the next action after the cycle). The execution of the task then continues and if it completes, will be regarded as *Successful*.

# Server Status

This panel allows you to see the status of the Mail & Deploy Server (→ [Server](#)) to which Mail & Deploy Client is currently connected (→ [Server Connection Settings](#)).

You can access the server status by clicking *Server Status* in the left menu of Mail & Deploy Client.

## Executions

The list shows all executions which are currently processed by Mail & Deploy Server; the list is not limited to only the repository you are currently connected to, but lists the executions of all repositories. You can abort an execution by clicking on the item in the list and then clicking the  button, if the following conditions are fulfilled:

- The repository to which the selected execution belongs needs to be the same as the repository to which you are currently connected with Mail & Deploy Client.
- The user (→ [Users](#)) with which you are currently logged on to the repository in Mail & Deploy Client needs to have at least a role of *Administrator*.



Please note that it may take a few seconds until the execution is actually aborted and disappears from the list of current executions.

You can view the current log of an execution by selecting it in the list of executions.

# Help

This panel allows you to consume help resources. You can access help by clicking *Help* in the left menu of Mail & Deploy Client.

# Additional Settings

## Styles

These settings allow you to manage styles (→ [Styles](#)) which can be used to control the visual representation of report elements (→ [Report Elements](#)). You can manage styles by clicking *Tools* → *Settings* → *Manage Styles* and then using the following buttons:



Click this button to create a new style; a context menu will open that allows you to select which type of style you want to create.



Click this button to edit the currently selected style.



Click this button to delete the currently selected style.

Each style has different attributes and usages (→ [Styles](#)).

## Calendars

These settings allow you to manage calendars which are basically lists of date-times which can be used to control executions (→ [Execution Plans](#)). You can manage calendars by clicking *Tools* → *Settings* → *Calendars* and then using the following buttons:



Click this button to create a new calendar.





Click this button to edit the currently selected calendar.



Click this button to delete the currently selected calendar.

Each calendar has the following attributes:

<b>Name</b>	The name of the calendar; this is only used within Mail & Deploy to identify the calendar.
<b>Date-Times</b>	A list of date-times which make up the calendar. These cannot be manually edited in Mail & Deploy; instead, you can import them from a file by clicking the  button and then locating a file that contains one date-time per line; please note, that the date-time format must either be in ISO format (yyyy-mm-ddTHH:MM) or in a format compatible with the current regional settings of the operating system. You can export date-times of a calendar by clicking the  button.

## E-Mail Servers

These settings allow you to manage e-mail servers that can be used by Mail & Deploy to send e-mails. You can manage e-mail servers if you are a user with a role of at least *Administrator* by clicking *Tools* → *Settings* → *E-Mail Servers* and then using the following buttons:



Click this button to create a new e-mail server; a context menu will open that allows you to select which type of e-mail server you want to create.



Click this button to edit the currently selected e-mail server.




Click this button to delete the currently selected e-mail server.



## SMTP Server

The most common type of e-mail server to send e-mails is SMTP; you can configure the following settings for an SMTP server:

<b>Name</b>	The name of the SMTP server; this is only used within Mail & Deploy to identify the e-mail server.
<b>Host</b>	The IP address or host name of the SMTP server.
<b>Port</b>	The port of the SMTP server (default SMTP port is 25).
<b>Security Protocol</b>	The security protocol to use when connecting to the SMTP server.
<b>Authentication Type</b>	<p>The authentication type to use when connecting to the SMTP server.</p> <p><i>None.</i> No authentication will be performed; the SMTP allows anonymous connections.</p> <p><i>Custom Credentials.</i> A custom username and password will be used for authentication when connecting to the SMTP server.</p> <p><i>Windows Authentication.</i> The credentials of the user that runs Mail &amp; Deploy Server (→ <a href="#">Server</a>) will be used for authentication when connecting to the SMTP server.</p>
<b>Username</b>	The username to use for authentication when connecting to the SMTP server; this setting is only applicable when <i>Authentication Type</i> is set to <i>Custom Credentials</i> .
<b>Password</b>	The password to use for authentication when connecting to the SMTP server; this setting is only applicable when <i>Authentication Type</i> is set to <i>Custom Credentials</i> .

Once you have entered the SMTP server details, you can click the  button to send a test e-mail; a new dialog will show up that allows you to enter the sender e-mail address and recipient e-mail address to be used for the test e-mail.

## Hub Settings

These settings control whether the repository to which Mail & Deploy Client is currently connected is accessible through Mail & Deploy Hub (→ [Hub](#)). You can access these settings by clicking *Tools* → *Settings* → *Hub* if you are a user with a role of at least *Administrator*. A dialog will appear that allows you to configure the following settings:


<b>Enable Hub</b>	This setting determines whether the current repository is accessible through Mail & Deploy Hub.
<b>Enable E-Mail Distribution</b>	This setting determines whether users can request the distribution of report documents by e-mail through Mail & Deploy hub.
<b>E-Mail Server</b>	This is the e-mail server (→ <a href="#">E-Mail Servers</a> ) that is used by Mail & Deploy whenever an e-mail is sent as a result of an on-demand report request.
<b>Sender E-Mail Address</b>	This is the sender e-mail address that is used by Mail & Deploy whenever an e-mail is sent as a result of an on-demand report request.

<b>Reply-To E-Mail Address</b>	This is the reply-to e-mail address that is used by Mail & Deploy whenever an e-mail is sent as a result of an on-demand report request.
--------------------------------	--

## Active Directory Connection Settings


These settings control whether the repository to which Mail & Deploy is currently connected is connected to an active directory. This is required to synchronize users and user groups (→ [Users and User Groups](#)) with an active directory (→ [Synchronize Users and User Groups with Active Directory Action](#)). You can access these settings by clicking *Tools* → *Settings* → *Active Directory Connection* if you are a user with a role of at least *Administrator*. A dialog will appear that allows you to configure the following settings:

<b>Enable Active Directory Connection</b>	This setting determines whether the current repository is connected to an active directory.
<b>Active Directory Type</b>	The type of active directory to which you want to connect.  <i>Machine.</i> This is a local SAM store (e.g. the local user group and user directory of the local machine).  <i>Domain.</i> An active directory on a domain controller.
<b>Host &amp; Port</b>	The host and port of the active directory server. This setting is only applicable for the active directory type <i>Domain</i> .
<b>Container</b>	The root container of the active directory to which to connect. This setting is only applicable for the active directory type <i>Domain</i> .
<b>Credentials</b>	This is the username and password combination used to connect to the active directory. This setting is only applicable for the active directory type <i>Domain</i> .

Once you have entered the active directory connection details, you can click the  button to test the connection to the active directory.

## Production Repository

These settings allow you to set up the production repository of that repository to which Mail & Deploy is currently connected to; these settings are only required, if you want to set up a development environment (→ [Development Environment](#)) and need to be set in the test repository in order to connect it to its production repository. You can access these settings by clicking *Tools* → *Settings* → *Production Repository* if you are a user with a role of at least *Designer*. A dialog will appear that allows you to configure the following settings:

<b>Enable Production Repository</b>	This setting determines whether you want to connect the current repository to a production repository.
<b>Host &amp; Port</b>	The host and port of the Mail & Deploy Server of the production system.
<b>Credentials</b>	The username and password to be used when connecting to the production repository.
<b>Repository</b>	The repository of the Mail & Deploy Server of the production system, that is the production repository for the repository to which you are currently connected. You can use the  button after entering <i>Host</i> and <i>Port</i> in order to refresh the list of available repositories.

## Global Variables

By using global variables, you can write expressions (→ [Expressions](#)) that you can later re-use by adding global variables to your expressions (→ [Global Variables](#)). This prevents you from having to write expressions everytime you use them. You can manage global variables by clicking *Tools* → *Global Variables* if you are a user with a role of at least Designer. A dialog appears that allows you to manage global variables using the following buttons:



Click this button to create a new global variable.



Click this button to edit the currently selected global variable.



Click this button to delete the currently selected global variable.

For each global variable, the following attributes can be set:

<b>Name</b>	The name of the global variable; this name is later used to reference global variables in expressions (→ <a href="#">Global Variables</a> ).
<b>Value</b>	An expression (→ <a href="#">Expressions</a> ) that evaluates to the value of the global variable.

## Section 4 Hub

# Introduction

Mail & Deploy Hub is an integrated HTTP web server that allows users to access a website with their browsers and request report documents and task executions on-demand. The configuration of Mail & Deploy Hub can be done using Mail & Deploy Server Control Panel (→ [Control Panel](#)).

The hub is an optional component of Mail & Deploy; it is not required to enable it.

# Configuration

The configuration of Mail & Deploy Hub can be done using Mail & Deploy Server Control Panel (→ [Control Panel](#)). You can choose whether you want the hub to be enabled and on which port and relative URL path it should run. Mail & Deploy Hub – if enabled – is started whenever the server service is started.

The following conditions have to be met in order to be able to access Mail & Deploy Hub:

- Mail & Deploy Hub must be enabled (→ [Control Panel](#)).
- The repository must be configured to be accessible in Mail & Deploy Hub (→ [Hub Settings](#)).
- The user needs to be a Mail & Deploy user in the given repository with an access level of *Hub Consumer* or higher (→ [Users and User Groups](#)).
- The user needs to have the required hub authorization (→ [Hub Authorizations](#)).
- The computer from which the hub is accessed needs to have a network connection to the computer that runs Mail & Deploy Server. The port on which the HTTP server of the hub is running must not be blocked by any firewall in between.

Another requirement – of course – is that the server service is running, because it hosts the HTTP web server that is required to access the hub.



## Hub Authorizations

By default users do not have the authorization to run any on-demand report request or task execution; users (or user groups) need to be explicitly authorized to be able to submit these requests. These authorizations are called *Hub Authorizations*.

Basically, a hub authorization gives a single user or user group the right to request an on-demand report or task execution through the hub. There are two types of hub authorizations; **report authorizations** and **task authorizations**.

## Report Authorizations

This type of authorization can be used to grant one user or user group the right to request a report on-demand and – optionally – to subscribe to a report through the hub. You can manage report authorizations in two ways:

- You can edit a user (→ [Users](#)) and then click the  button; a window will be displayed that allows you to manage all hub authorizations for reports and tasks for that user.
- You can edit a report (→ [Reports](#)) and then click the  button; a window will be displayed that allows you to manage all hub authorizations for that report.

The window that will be displayed contains all current report authorizations; the following buttons can be used to manage report authorizations:



Click this button to create a new report authorization; a context menu will open that allows you to select which type of authorization you want to create.



Click this button to edit the currently selected report authorization.



Click this button to delete the currently selected report authorization.

## User Authorization

This type of report authorization can be used to grant the right to request and – optionally – subscribe to a report to a user. Each user authorization has the following attributes:

<b>Report</b>	The report to which the authorization grants access.
<b>User</b>	The user to which access to the report is granted by the authorization.
<b>Allow Subscription</b>	This setting determines whether the user is allowed to create a subscription for that report (→ <a href="#">Subscriptions</a> ).



## User Group Authorization

This type of report authorization can be used to grant the right to request and – optionally – subscribe to a report to a user group. Each user group authorization has the following attributes:




<b>Report</b>	The report to which the authorization grants access.
<b>User Group</b>	The user group to which access to the report is granted by the authorization.
<b>Allow Subscription</b>	This setting determines whether members of the user group are allowed to create a subscription for that report (→ <a href="#">Subscriptions</a> ).

## Task Authorizations

This type of authorization can be used to grant one user or user group the right to request the execution of a task through Mail & Deploy Hub. You can manage task authorizations in two ways:

- You can edit a user (→ [Users](#)) and then click the  button; a window will be displayed that allows you to manage all hub authorizations for reports and tasks for that user.
- You can edit a task (→ [Tasks](#)) and then click the  button; a window will be displayed that allows you to manage all hub authorizations for that task.

The window that will be displayed contains all current task authorizations; the following buttons can be used to manage report authorizations:

	Click this button to create a new task authorization; a context menu will open that allows you to select which type of authorization you want to create.
	Click this button to edit the currently selected task authorization.
	Click this button to delete the currently selected task authorization.

## User Authorization

This type of task authorization can be used to grant the right to request the execution of a task to a user. Each user authorization has the following attributes:

<b>Task</b>	The task to which the authorization grants access.
<b>User</b>	The user to which access to the task is granted by the authorization.

## User Group Authorization

This type of task authorization can be used to grant the right to request the execution of a task to a user group. Each user group authorization has the following attributes:

<b>Task</b>	The task to which the authorization grants access.
<b>User Group</b>	The user group to which access to the task is granted by the authorization.



# Usage

## Introduction

Mail & Deploy Hub can be accessed by using a standard web-browser such as Internet Explorer, Edge, Chrome, Firefox etc. In order to access the hub, the correct URL needs to be entered; the general form of the URL is `http://<servername>:<port>/<relativeurlpath>`. However, the following rules need to be considered:

- The port can be omitted if it's set to be the default HTTP port 80.
- The relative URL path can be omitted if it's empty.

**Example:** If you have set up the hub to use the default HTTP port and to have no relative URL path, then the hub can be reached by entering `http://<servername>`. If you have a relative URL path and the default HTTP port, you can access the hub by entering `http://<servername>/<relativeurlpath>`.

## General

### Repository Selection

If the user with which you login to the hub has access to more than one repository, a page will be displayed that allows you to select the repository with which you want to work. In order to switch to another repository, you can simply click the option *General* → *Switch Repository* in the left menu. If the user with which you login to the hub only has access to a single repository, you will automatically be redirected to the home page (→ [Home](#)).

### Home

After entering the repository, you will get to the home page; this page displays:

- The newest report documents that have been added to your report library (→ [Library](#)). The table displays the following attributes:

Created	The time at which the report document has been created in the local time according to the time zone of the currently logged on user.
Type	An icon that identifies the file type of the report document.
Name & Description	The name and description of the report document. The name is a hyperlink that you can click in order to open or download (depending on the file type) the report document.

- The most recent exceptions that occurred when executing on-demand report requests, report subscriptions or task executions that are associated with the user currently logged on (→ [History](#)).

## Reports

### Available Reports

This page displays a list of all reports to which the currently logged on user has access to (→ [Hub Authorizations](#)). For each available report, the name and description are shown in the table. Below the description, there is an option to perform a one-time request of a report document (→ [Requesting](#)

[Report Documents](#)) and – if the hub authorization of the user allows subscriptions – to create a subscription (→ [Creating Subscriptions](#)).

## Requesting Report Documents

If you want to perform a one-time request of a report, you can click the hyperlink *Request Now* which can be found immediately below the description of the report. This will open a new page where you can set the following attributes:

Parameters	If the report has any parameters (→ <a href="#">Report Parameters</a> ), they will be displayed and can be entered by the user. Depending on the report parameter settings, the user may be able to enter a free text or select a value from a given range of values.
Name	The name you want to give the created report document. This name is used to identify the report document within Mail & Deploy. The default value is automatically set to the name of the report on which the created report document will be based.
Description	The description you want to give the created report document.
File Type	The file type in which you want to create the report document. The range of available file types depends on the report type (→ <a href="#">Report Types</a> ).
Culture	This is an optional setting that depends on whether the report on which the report document will be based supports culture awareness (→ <a href="#">Culture Awareness</a> ) and allows custom cultures (→ <a href="#">General Attributes</a> ). If that is the case, you can select the culture to use for the creation of the report.
Distribute by E-Mail to...	This setting determines whether the created report document shall send by e-mail to the e-mail address of the currently logged on user. If that user does not have an e-mail address, this option is not applicable. This option is only available, if the repository supports e-mail distribution (→ <a href="#">Hub Settings</a> ).
E-Mail Addresses	You can tick this box and then enter a list of additional e-mail addresses separated by a semicolon to which you want to distribute the report document. This option is only available, if the repository supports e-mail distribution (→ <a href="#">Hub Settings</a> ).
Add to my Library	If you tick this box, the created report document will be added to the report document library of the currently logged on user (→ <a href="#">Library</a> ).
File Path	<p>If you tick this box, you can enter a file path to which you want the created report document to be saved. It's important to note that the report document will be saved to the file system by Mail &amp; Deploy Server and therefore the following requirements have to be met:</p> <ul style="list-style-type: none"> <li>➤ The user that runs Mail &amp; Deploy Server needs to have sufficient user rights to write to the directory.</li> <li>➤ The file path needs to be reachable from the computer on which Mail &amp; Deploy Server runs; this might require to use UNC file paths in some cases. Please note that mapped network drive letters cannot be used, even if the user that runs Mail &amp; Deploy Server has a mapped network drive set up.</li> </ul>



Please note you can set up any combination of distribution methods; so it's possible to not send the report document by e-mail at all while just saving it to a file system and/or adding it to the report document library. However, at least one distribution method needs to be selected in order to successfully submit the request.

Once you have set up everything to your needs, you can click *SUBMIT REQUEST* which will send the request to Mail & Deploy Server; you will automatically be re-directed to the status page (→ [Report Document Request Status](#)) of the request you have just submitted.

## Subscriptions

This page lists all subscriptions for the creation of report documents. Basically a subscription is the regular creation and distribution of a report document; in other words: it's similar to a report document request (→ [Requesting Report Documents](#)) which will be executed on a regular basis.

### Creating Subscriptions

You can create a subscription by clicking *Create Subscription* in the list of available reports (→ [Available Reports](#)).



Please note that whether a subscription for a specific report may be created depends on the authorization of the user (→ [Report Authorizations](#)).

After clicking *Create Subscription*, a new page will open which allows you to enter all the attributes as if you would request a single creation and distribution of a report document (→ [Requesting Report Documents](#)). In addition to that, the following attributes can be set:

<b>Time Zone</b>	The time zone to apply to the subscription; all times in the subscription are always local times according to this time zone.
<b>Start Date</b>	The date-time where you want the next execution of the subscription to take place (→ <a href="#">Execution Sequence</a> ).
<b>End Date</b>	The end date is optional; if enabled, you can specify until when the subscription shall be valid.
<b>Frequency</b>	The frequency determines how often the subscription is due to be executed.

*Daily.* If you choose this frequency, the subscription will be executed on a daily basis.

*Weekdays.* If you choose this frequency, the subscription will be executed on the specified weekdays.

*Weekly.* If you choose this frequency, the subscription will be executed on a weekly basis.

*Monthly.* If you choose this frequency, the subscription will be executed on a monthly basis (this does not equal a 30-day period, but rather to a full month; e.g. April 3<sup>rd</sup> plus 1 month will be May 3<sup>rd</sup>).

*Yearly.* If you choose this frequency, the subscription will be executed on a yearly basis (this does not equal a 365-day period, but rather to a full year; e.g. April 3<sup>rd</sup> 2018 plus 1 year will be April 3<sup>rd</sup> 2019; leap years will automatically be recognized by Mail & Deploy).

Once you have set everything up, you can click *SAVE SUBSCRIPTION* to save the subscription. You will then be automatically redirected to a page where you can manage your subscriptions (→ [Managing Subscriptions](#)).

## Managing Subscriptions

Subscriptions can be managed by clicking *Subscriptions* in the left menu. A table will be displayed, that contains information about all the subscriptions you currently have. You can edit a subscription by clicking *Edit Subscription* and delete a subscription by clicking *Delete Subscription*.

## Library

This page lists all report documents contained in your library. The table that will be displayed is organized by report, which means that there is one row per report (→ [Reports](#)) for which there is at least one report document in your library; the table has the following columns:

<b>Report</b>	The report, which acts as a category for your report document.
<b>Latest Report Document</b>	The latest report document in your library that has been created from the report. The name is a hyperlink that you can click in order to open or download (depending on the file type) the report document.




You can click on the name of the report and a new page will be displayed that contains all report documents (newest at the top) of the report you have clicked on. That table contains the following columns:

<b>Created</b>	The time at which the report document has been created in the local time according to the time zone of the currently logged on user.
<b>Type</b>	An icon that identifies the file type of the report document.
<b>Name &amp; Description</b>	The name and description of the report document. The name is a hyperlink that you can click in order to open or download (depending on the file type) the report document.

You can delete a report document from your library by clicking the *Delete* hyperlink in the *Name & Description* column of the table.

## History

This page displays all report document requests (→ [Requesting Report Documents](#)) and executions based on report document subscriptions (→ [Subscriptions](#)); the table contains newest entries first and has the following columns:

<b>Date</b>	The date and time at which the execution of the request has taken place; these times are local times according to the time zone of the user.
<b>Status</b>	<p>An image that describes the status of the execution:</p> <ul style="list-style-type: none"> <li> . The execution was successful.</li> <li> . The execution was cancelled.</li> <li> . The execution failed.</li> </ul>
<b>Typ</b>	<p>The type of execution.</p> <p><i>On-Demand Request.</i> The execution was due to a one-time requested report document (→ <a href="#">Requesting Report Documents</a>).</p> <p><i>Subscription.</i> The execution was due to one of your subscriptions (→ <a href="#">Subscriptions</a>).</p>

<b>Details</b>	Here you can find the name and description of the report document. By clicking <i>View Details</i> you can see the status or result of the execution.
----------------	---

## Report Document Request Status

On this page you can see the current status of a report document request. You will see the log file and – if the execution was completed successfully – a link to open or download the created report document.

## Tasks

### Available Tasks

This page lists all tasks for which you are authorized (→ [Task Authorizations](#)) to request an execution through Mail & Deploy Hub. The table contains the name and description of each task. Just below the description there is a link *Request Execution* that will bring you to the page that enables you to request the execution of that task (→ [Request Task Execution](#)).

### Request Task Execution




If you want to request the execution of a task, you can click the link *Request Execution* below the description of the task. You can then enter the following attributes:

<b>Parameters</b>	If the task has any parameters (→ <a href="#">Task Parameters</a> ), they will be displayed and can be entered by the user. Depending on the task parameter settings, the user may be able to enter a free text or select a value from a given range of values.
-------------------	---

Once you have configured everything to your needs, you can click *SUBMIT REQUEST* to actually request the execution of the task. You will then be redirected to a page that allows you to see the current status of the requested execution (→ [Task Execution Request Status](#)).

## History

This page lists all historic task execution you have requested (→ [Request Task Execution](#)); the table contains newest executions at the top and contains the following columns:

<b>Date</b>	The date and time at which the execution of the request has taken place; these times are local times according to the time zone of the user.
<b>Status</b>	<p>An image that describes the status of the execution:</p> <ul style="list-style-type: none"> <li> . The execution was successful.</li> <li> . The execution was cancelled.</li> <li> . The execution failed.</li> </ul>
<b>Details</b>	You can find the name and description of the task here. By clicking <i>View Details</i> you can view the log file of the execution.

## Task Execution Request Status

On this page you can see the status and log file of the execution of a task.

## Section 5

# API

# Introduction

The API of Mail & Deploy is an HTTP webservice that allows 3<sup>rd</sup> party vendors to directly access certain functions of Mail & Deploy. Using the API, Mail & Deploy reports (and thus QlikView and Qlik Sense) can be strongly integrated in existing applications.

**Example:** *If you are the developer of a CRM system, you can easily develop an interaction between the CRM system and Mail & Deploy: if, for instance, you have customer reports in Mail & Deploy that contain data from QlikView and/or Qlik Sense and you want your CRM users to request these reports, you can use Mail & Deploy's API to submit a request to create a report for a given Customer ID and the API will return the report document file to the caller.*

# Configuration

To use Mail & Deploy API, you have to ensure the following conditions:

- Mail & Deploy API must be enabled (→ [Control Panel](#)).
- At least one user with at least a role of *API Consumer* needs to be set up (→ [Users](#)).
- If the API is called from another computer, a network connection between that computer and the computer that runs Mail & Deploy Server needs to be established and the configured port on which the API webservice runs must not be blocked by any firewall in between.



Our recommended practice is to create one API user with a role of *API Consumer* for each 3<sup>rd</sup> party vendor that you want to grant access to; this way each 3<sup>rd</sup> party vendor gets his own credentials to authenticate and when using the execution calendar (→ [Execution Calendar](#)) you can easily identify which 3<sup>rd</sup> party vendor created requests. Also, if you want to exclude a 3<sup>rd</sup> party vendor from accessing the API, you can simply delete that user.



# Usage

## Introduction

The API is provided through an HTTP REST webservice with custom HTTP authentication (which means that simply a pair of username and password needs to be provided in the HTTP header in order to perform authentication). The URL to which the webservice listens is `http://<servername>:<port>/MailAndDeploy/Api`.

The webservice supports both POST and GET requests. POST requests need to supply their parameters in the HTTP message, GET requests may supply their parameters through URL parameters. For every request, the API will return an XML document that contains the requested data or, if unsuccessful, an error message.

## GET Requests

As previously stated, GET requests need to supply their parameters by using query string parameters. An example for a GET request might be:

```
http://<servername>:<port>/MailAndDeploy/Api/?RequestType=CreateReportDocument&RepositoryName=CrmReporting&ReportName=CustomerReport&FileType=AdobePdfDocument&Parameter_CustomerId=1
```

This would instruct Mail & Deploy to process a *CreateReportDocument* request (→ [Create Report Document Request](#)) by creating a report document from the Report *CustomerReport* in repository *CrmReporting*. The file type in which to create the report document is *AdobePdfDocument* and the ID of the customer for which to create that report is *1*.

## POST Requests

As previously stated, POST requests need to supply their parameters as part of the message header. POST requests need to be submitted to the URL `http://<servername>:<port>/MailAndDeploy/Api`.

## Responses

Every request will be answered by Mail & Deploy API with an XML document.

### Success Responses

If the request can be handled successfully by Mail & Deploy, an XML document that contains the return value for that request will be returned; the actual content of that XML document depends on the type of request; therefore, you will find detailed information about how success responses look like in the documentation of each individual request below.

### Exception Responses

The XML document that will be returned in case the request fails is formatted as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<MailAndDeployApiResponse Success="False"
  ExceptionMessage="{exception_message}"
  ExceptionStackTrace="{exception_stacktrace}" />
```

The following properties are contained in the response:

<b>Success</b>	The value <i>False</i> indicates that the request has failed.
<b>ExceptionMessage</b>	An exception message that contains information about why the request has failed.

**ExceptionStackTrace**

Contains information about the stack trace of the exception.

## Reference

This reference provides information on all request types that can be made through Mail & Deploy API.

### Create Report Document Request

This request instructs Mail & Deploy to create a report document from a report with a given file type and – optionally – parameters. The parameters for this request are:

<b>RequestType</b>	This parameter specifies the type of request and needs to be set to <i>CreateReportDocument</i> .
<b>RepositoryName</b>	The name of the repository (→ <a href="#">Repository</a> ) to which the report from which you want to create a report document belongs.
<b>ReportName</b>	The name of the report (→ <a href="#">Reports</a> ) from which you want to create a report document.
<b>FileType</b>	<p>The file type in which you want to create the report document. The range of available file types depends on the report type (→ <a href="#">Report Types</a>). The following values can be provided:</p> <p><i>AdobePdfDocument</i>. This file type represents an Adobe PDF Document.</p> <p><i>AdobePdfADocument</i>. This file type represents an Adobe PDF/A compliant document.</p> <p><i>HtmlDocument</i>. This file type represents an HTML document.</p> <p><i>MicrosoftExcelWorkbook</i>. This file type represents a Microsoft Excel Workbook.</p> <p><i>MicrosoftPowerPointPresentation</i>. This file type represents a Microsoft PowerPoint Presentation.</p> <p><i>MicrosoftWordDocument</i>. This file type represents a Microsoft Word Document.</p>
<b>Parameter_*</b>	You can supply values for report parameters (→ <a href="#">Report Parameters</a> ) if the report from which you want to create a report document supports them. Each report parameter needs to be called <i>Parameter_&lt;name_of_report_parameter&gt;</i> .

### Example

The following URL can be used as a GET request to request a report document from a report named *CustomerReport* that is contained in a repository named *CRMReports*. The file type in which the report document shall be created is an Adobe PDF Document. The report contains three report parameters (→ [Report Parameters](#)): *CustomerId* which represents the ID of the customer for which to create the report document, *Year* which defines the year for which the report document shall contain data and *Month* that defines the month for which the report document shall contain data:

```
http://<servername>:<port>/MailAndDeploy/Api/?RequestType=CreateReportDocument&RepositoryName=CRMReports&ReportName=CustomerReport&FileType=AdobePdfDocument&Parameter_CustomerId=1&Parameter_Year=2017&Parameter_Month=1
```

## Success Response

The XML document that will be returned in case the request completes successfully is formatted as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<MailAndDeployApiResponse Success="True"
  FileType="{file_type}"
  ReportDocumentContent="{base64_content}" />
```

The following properties are contained in the response:

<b>Success</b>	The value <i>True</i> indicates that the request has completed successfully.
<b>FileType</b>	<p>This indicates the file type of the returned report document. It can be one of the following values:</p> <p><i>AdobePdfDocument</i>. This file type represents an Adobe PDF Document.</p> <p><i>AdobePdfADocument</i>. This file type represents an Adobe PDF/A compliant document.</p> <p><i>HtmlDocument</i>. This file type represents an HTML document.</p> <p><i>MicrosoftExcelWorkbook</i>. This file type represents a Microsoft Excel Workbook.</p> <p><i>MicrosoftPowerPointPresentation</i>. This file type represents a Microsoft PowerPoint Workbook.</p> <p><i>MicrosoftWordDocument</i>. This file type represents a Microsoft Word Document.</p>
<b>ReportDocumentContent</b>	This is the content of the report document. It's a byte array encoded as Base64; if this Base64-String is decoded and the resulting byte array is written to a file with the correct file type (see above) the file can be opened with the associated application (e.g. Adobe PDF Reader for Adobe PDF Documents).

## Execute Task Request

This request instructs Mail & Deploy to execute a task (→ [Tasks](#)) of the given repository. The parameters for this request are:

<b>RequestType</b>	This parameter specifies the type of request and needs to be set to <i>ExecuteTask</i> .
<b>RepositoryName</b>	The name of the repository (→ <a href="#">Repository</a> ) to which the task to execute belongs.
<b>TaskName</b>	The name of the task (→ <a href="#">Tasks</a> ) you want to execute.
<b>Parameter_*</b>	You can supply values for task parameters (→ <a href="#">Task Parameters</a> ) if the task you want to execute supports them. Each task parameter needs to be called <i>Parameter_&lt;name_of_task_parameter&gt;</i> .

## Example

The following URL can be used as a GET request to request the execution of the task named *CreateStoreReports* that is contained in a repository named *CRMReports*. The task contains three task parameters (→ [Task Parameters](#)): *StoreId* which represents the IDs of the stores for which to execute

the task, *Year* which defines the year for which to execute the task and *Month* that defines the month for which to execute the task:

```
http://<servername>:<port>/MailAndDeploy/Api/?RequestType=ExecuteTask&RepositoryName=CRMRe
ports&TaskName=CreateStoreReports
&Parameter_StoreId=1|2|3&Parameter_Year=2017&Parameter_Month=1
```

### Success Response

The XML document that will be returned in case the request completes successfully is formatted as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<MailAndDeployApiResponse Success="True" />
```

The following properties are contained in the response:

<b>Success</b>	The value <i>True</i> indicates that the request has completed successfully.
----------------	--

## Get Available Reports Request

This request instructs Mail & Deploy to return all available reports (→ [Reports](#)) of the given repository. The parameters for this request are:

<b>RequestType</b>	This parameter specifies the type of request and needs to be set to <i>GetAvailableReports</i> .
<b>RepositoryName</b>	The name of the repository (→ <a href="#">Repository</a> ) for which to return all available reports.

### Example

The following URL can be used as a GET request to request all available reports of a repository named *CRMReports*:

```
http://<servername>:<port>/MailAndDeploy/Api/?RequestType=GetAvailableReports&RepositoryNam
e=CRMReports
```

### Success Response

The XML document that will be returned in case the request completes successfully is formatted as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<MailAndDeployApiResponse Success="True">
  <Report Name="{report_name}"
    Description="{report_description}"
      <SupportedFileType Value="{supported_file_type}" />
      ...
      <ReportParameter Name="{reportparameter_name}"
        Caption="{reportparameter_caption}" />
      ...
    </Report>
    ...
</MailAndDeployApiResponse>
```

The following properties are contained in the response:

<b>Success</b>	The value <i>True</i> indicates that the request has completed successfully.
<b>Report Name</b>	The name of the report.

<b>Report Description</b>	The description of the report.
<b>Supported File Types</b>	<p>A list of file types supported by that report (i.e. report documents can be created in any of these supported file types). The following values are possible:</p> <p><i>AdobePdfDocument</i>. This file type represents an Adobe PDF Document.</p> <p><i>AdobePdfADocument</i>. This file type represents an Adobe PDF/A compliant document.</p> <p><i>HtmlDocument</i>. This file type represents an HTML document.</p> <p><i>MicrosoftExcelWorkbook</i>. This file type represents a Microsoft Excel Workbook.</p> <p><i>MicrosoftPowerPointPresentation</i>. This file type represents a Microsoft PowerPoint Workbook.</p> <p><i>MicrosoftWordDocument</i>. This file type represents a Microsoft Word Document.</p>
<b>Report Parameters</b>	A list of zero or more report parameters (→ <a href="#">Report Parameters</a> ) of that report.

## Get Available Tasks Request

This request instructs Mail & Deploy to return all available tasks (→ [Tasks](#)) of the given repository. The parameters for this request are:

<b>RequestType</b>	This parameter specifies the type of request and needs to be set to <i>GetAvailableTasks</i> .
<b>RepositoryName</b>	The name of the repository (→ <a href="#">Repository</a> ) for which to return all available tasks.

## Example

The following URL can be used as a GET request to request all available reports of a repository named *CRMReports*:

*http://<servername>:<port>/MailAndDeploy/Api/?RequestType=GetAvailableTasks&RepositoryName=CRMReports*

## Success Response

The XML document that will be returned in case the request completes successfully is formatted as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<MailAndDeployApiResponse Success="True">
  <Task Name="{report_name}"
    Description="{report_description}"
      <TaskParameter Name="{taskparameter_name}"
        Caption="{taskparameter_caption}" />
      ...
    </Task>
    ...
</MailAndDeployApiResponse>
```

The following properties are contained in the response:

Success	The value <i>True</i> indicates that the request has completed successfully.
Task Name	The name of the task.
Task Description	The description of the task.
Task Parameters	A list of zero or more task parameters (→ <a href="#">Task Parameters</a> ) of that task.

Section 6









# Appendix

# Actions

## Introduction

Actions are entities widely used in Mail & Deploy; there is a variety of actions which all perform different operations such as filtering a field in a datasource, creating a report document, distributing a report document by e-mail, saving a report document to the file system etc.

These actions are managed in lists; when Mail & Deploy executes the action list, all actions will be executed sequentially (from top to bottom); this way you can control the order in which actions are executed. The following buttons can be used to manage actions in an action list:

	Click this button to create a new action; a context menu will open that allows you to select which type of action you want to create. The actions to create are organized in categories.
	Click this button to edit the currently selected action.
	Click this button to create a copy of the currently selected action.
	Click this button to delete the currently selected action.
	Click this button to move the currently selected action to the top of the list.
	Click this button to move the currently selected action one step up.
	Click this button to move the currently selected action one step down.
	Click this button to move the currently selected action to the bottom of the list.

If you want to add an action to an action container (→ [Group Action](#), → [Cycle Action](#)), you can click the top item of the action container with the right mouse and select the type of action you want to add.

You can expand and collapse an action container by clicking on its image.

## Action Types

### Activate Datasource Bookmark Action

This action can be used to activate a bookmark of a datasource (→ [Bookmarks](#)) to restore the state that the bookmark contains. The following attributes can be set:

<b>Datasource</b>	The datasource (→ <a href="#">Datasources</a> ) that contains the bookmark you want to activate.
<b>Bookmark Name</b>	An expression (→ <a href="#">Expressions</a> ) that evaluates to the name of the bookmark (→ <a href="#">Bookmarks</a> ) you want to activate.

### Preparation Actions

This is a list of actions (→ [Actions](#)) which will be executed before the bookmark button is activated.

### Activate Datasource Theme Action

This action allows you to activate a theme of a datasource to influence the presentation of subsequent chart report elements (→ [Chart Elements](#)) and sheet report elements (→ [Sheet Elements](#)). The following attributes can be set:



<b>Datasource</b>	The datasource (→ <a href="#">Datasources</a> ) in which to activate the theme.
<b>Theme Code</b>	An expression (→ <a href="#">Expressions</a> ) that evaluates to the code of the theme to activate.

Themes are currently only supported for Qlik Sense datasources. You can find out the theme code for a given theme by following these steps:

- Open the single configurator in the dev hub of your Qlik Sense installation.
- Select any app.
- Select any worksheet or chart within the app.
- In the settings, choose the theme you want to find the code for in the *General* tab.
- The theme code can be found in the resulting URL of the sheet or chart:

`...demo.qvf&sheet=jgxeqZ&theme=card&opt=nointeraction...`

## Add File To File Collection Action

This action can be used to add an external file or a report document to a file collection. The following attributes can be set:

<b>File Collection Name</b>	The name of the file collection to add the file to.
<b>Mode</b>	<p>The type of file you want to add to the file collection.</p> <p><i>Add Report Document.</i> If you choose this mode, you can select a report document that will be added to the file collection.</p> <p><i>Add External File.</i> If you choose this mode, you can enter an expression ) that evaluates to the full path of the external file to add to the file collection. Please note, that this file needs be reachable from Mail &amp; Deploy Server, because the action will be executed there. Therefore, the following requirements have to be met:</p> <ul style="list-style-type: none"> <li>➤ The user that runs Mail &amp; Deploy Server needs to have sufficient user rights to read the file.</li> <li>➤ The file path needs to be reachable from the computer on which Mail &amp; Deploy Server runs; this might require to use UNC file paths in some cases. Please note that mapped network drive letters cannot be used, even if the user that runs Mail &amp; Deploy Server has a mapped network drive set up.</li> </ul>
<b>File Name</b>	An expression (→ <a href="#">Expressions</a> ) that evaluates to the name of the file (including the file extension) that the file shall have within the file collection.

## Preparation Actions

This is a list of actions (→ [Actions](#)) which will be executed before the file is added to the file collection.

## Append To Report Document Action

This action can be used to append a report document or an external file to a report document. This action currently only works with Adobe PDF Documents. The following attributes can be set:

<b>Report Document</b>	The previously created report document to append to.
------------------------	--

**Mode**

The mode to apply.

*Append Report Document.* If you use this mode, a report document will be appended to the report document. You can then specify the report document to append.

*Append External File.* If you use this mode, an external file will be appended to the report document. You can then specify an expression (→ [Expressions](#)) that evaluates to the full path of the file to append. The file path needs to include the file type extension. It's important to note that the file will be saved to the file system by Mail & Deploy Server and therefore the following requirements have to be met:

- The user that runs Mail & Deploy Server needs to have sufficient user rights to read from the directory.
- The file path needs to be reachable from the computer on which Mail & Deploy Server runs; this might require to use UNC file paths in some cases. Please note that mapped network drive letters cannot be used, even if the user that runs Mail & Deploy Server has a mapped network drive set up.

## Click Datasource Button Action

This action can be used to click a button in a datasource (→ [Datasources](#)). This can – for instance – be used to run a macro of a QlikView Document.



Please note, that clicking a datasource button is not available for all datasource types.

The following attributes are available:

<b>Datasource</b>	The datasource (→ <a href="#">Datasources</a> ) that contains the button to click.
<b>Button ID</b>	An expression (→ <a href="#">Expressions</a> ) that evaluates to the object ID of the button to click.

## Preparation Actions

This is a list of actions (→ [Actions](#)) which will be executed before the datasource button is clicked.

## Create Report Document Action

This action can be used to create a report document from a report (→ [Reports](#)). The created report document can then be used in other actions; for instance to distribute it by e-mail (→ [Distribute by E-Mail Action](#)), to save it to the file system (→ [Save Report Document to File System Action](#)) or to add it to the hub library of users (→ [Distribute Report Document to Hub Report Document Library Action](#)). The following attributes can be set:

<b>Report</b>	The report (→ <a href="#">Reports</a> ) from which you want to create a report document.
<b>Name</b>	The name of the created report document; this name is used to reference the created report document in other actions, for instance if you want to save the report to the file system, distribute it by e-mail etc.

Please note, that there can only be one report document with a given name at any time during the execution of a task; this means that if a report document with the given name already exists, it will be overwritten by the newly created report document.

File Type	The file type in which you want to create the report document. The range of available file types depends on the report type (→ <a href="#">Report Types</a> ).
Culture	This setting allows you to specify in which culture (→ <a href="#">Culture Awareness</a> ) the report document shall be created. It can only be specified for reports where <i>Allow Custom Culture</i> is enabled (→ <a href="#">General Attributes</a> ).
Parameters	This list can be used to pass parameter values to the report; if the report on which the report document is based has parameters (→ <a href="#">Report Parameters</a> ), you can feed these parameters with a list of values (→ <a href="#">Expressions</a> ).
Settings	If the file type in which you want to create the report document supports additional settings, you can specify them on the settings tab (→ <a href="#">File Type Settings</a> ).

## Cycle Action

This action can be used to repeat a certain sub-set of actions for each value of a datasource field.

**Example:** Let's suppose you have designed a report (→ [Reports](#)) that contains data and charts about a single sales store and you want to create and distribute one report document for each sales store there is in the QlikView Document on which the report is based. In this case you can use a cycle action that cycles over each value of the field "Sales Store" in the QlikView Document, creates a report for that sales store and distributes it.

## General Attributes

You can set the following general attributes:

Datasource Field	The datasource field from which to read values to cycle over.
Field Value Selector	<p>The field value selector that determines which values of the datasource field shall be used for the cycle.</p> <p><i>All Values.</i> If you use this selector, all values of the datasource field will be used for the cycle, regardless of the current filter state of the datasource.</p> <p><i>Applicable Values.</i> If you use this selector, only those values of the datasource field will be used for the cycle which are compatible to the current filter state of the datasource.</p> <p><i>Filtered Values.</i> If you use this selector, only those values of the datasource field will be used for the cycle to which the datasource field is explicitly filtered.</p>
Sort Order	<p>This setting determines in which order the values will be processed by the cycle.</p> <p><i>None.</i> No specific sort order will be used and the values will be used in the exact order in which they have been read from the datasource.</p> <p><i>Ascending.</i> The values of the datasource field will be processed by the cycle in ascending order.</p> <p><i>Descending.</i> The values of the datasource field will be processed by the cycle in descending order.</p>
Value Name	The name of the variable with which the current cycle value can be referenced in expressions (→ <a href="#">Expressions</a> ) which are used in any of the actions of the cycle.
Exception Behaviour	Determines what Mail & Deploy shall do if the execution of one of the actions of the cycle causes an exception (→ <a href="#">Exception Handling</a> ).

*Do Not Handle.* If you select this behaviour, any exception caught by the cycle action will not be handled and passed on to the parent action container.

*Cancel Cycle.* If you select this behaviour, any exception caught by the cycle action will lead to the complete cycle to be cancelled, but no exception will be passed on to the parent action container.

*Continue With Next Action.* If you select this behaviour, any exception caught by the cycle will lead to cycle execution to be continued with the next action. No exception will be passed on to the parent action container.

*Continue With Next Value.* If you select this behaviour, any exception caught by the cycle will lead to the cycle to be continued with the next value.

<b>Actions</b>	This is a list of actions (→ <a href="#">Actions</a> ) which will be executed for each cycle value.
----------------	---

## Preparation Actions

This is a list of actions (→ [Actions](#)) which will be executed before the cycle values are evaluated and before the actions of the cycle will be executed. These actions can – for instance – be used to filter the datasource field from which the cycle values are derived so the number of cycle values is limited.

## Distribute by E-Mail Action

This action can be used to distribute report documents that have been previously created (→ [Create Report Document Action](#)) and/or external file attachments by e-mail.

## General Attributes

You can set the following general attributes:

<b>E-Mail Server</b>	The e-mail server to use to send the e-mail (→ <a href="#">E-Mail Servers</a> ).
<b>Sender E-Mail Address</b>	The sender e-mail address to use to send the e-mail.
<b>Reply-To E-Mail Address</b>	The reply-to e-mail address to use.
<b>Subject</b>	An expression (→ <a href="#">Expressions</a> ) that evaluates to the subject to be used when sending the e-mail.
<b>Body Type</b>	This settings determines how the body of the e-mail is set.

*Expression.* If you use this type, an expression field will appear where you can enter an expression (→ [Expressions](#)) that will evaluate to the body.

*HTML Report Document.* If you use this type, you can use a previously created report document (→ [Create Report Document Action](#)) of file type HTML to use as the body. In this case, the e-mail body will itself be the content of the HTML report document.

## Attachments

This list allows you to manage all attachments you want to be added to the e-mail. You can manage attachments by using the following buttons:



Click this button to create a new attachment; a context menu will open that allows you to select which type of attachment you want to create.



Click this button to edit the currently selected attachment.



Click this button to delete the currently selected attachment.

## External File

This type of attachment allows you to add an external file to the e-mail. The following attributes can be set:

<b>Attachment Name</b>	An expression (→ <a href="#">Expressions</a> ) that evaluates to the name of the attachment.
<b>File Path</b>	<p>An expression (→ <a href="#">Expressions</a>) that evaluates to the full path to the file to attach. Please note, that this file needs be reachable from Mail &amp; Deploy Server, because the action will be executed there. Therefore, the following requirements have to be met:</p> <ul style="list-style-type: none"> <li>➤ The user that runs Mail &amp; Deploy Server needs to have sufficient user rights to read the file.</li> <li>➤ The file path needs to be reachable from the computer on which Mail &amp; Deploy Server runs; this might require to use UNC file paths in some cases. Please note that mapped network drive letters cannot be used, even if the user that runs Mail &amp; Deploy Server has a mapped network drive set up.</li> </ul>

## Report Document

This type of attachment allows you to add a previously created report document () to the e-mail. The following attributes can be set:

<b>Attachment Name</b>	An expression (→ <a href="#">Expressions</a> ) that evaluates to the name of the attachment.
<b>Report Document</b>	The report document you want to attach. This needs to be a report document that has been created by a <a href="#">Create Report Document Action</a> in the same task.

## Recipients

This section lets you control who shall receive the e-mail. Please note that a combination of different recipient types (dynamic and static) is possible.

### Basic Attributes



The following basic attributes can be set:

<b>Send to Dynamically Assigned Users</b>	If you tick this option, the e-mail will be sent to e-mail addresses of users that have been dynamically assigned (→ <a href="#">Dynamically Assign Users to Report Document Action</a> ) to the report document. These dynamical e-mail recipients will receive all external file attachments but only those report document attachments to which they have been dynamically assigned.
<b>Send to Dynamically Assigned E-Mail Addresses</b>	If you tick this option, the e-mail will be sent to e-mail addresses that have been dynamically assigned (→ <a href="#">Dynamically Assign E-Mail Addresses to Report Document Action</a> ) to the report document. These dynamic e-mail recipients will receive all external file attachments but only those report document attachments to which they have been dynamically assigned.
<b>Send to Active Directory User Group(s)</b>	If you tick this option, the e-mail will be sent to all users who belong to any of the active directory user groups with the names you can specify by the expression (→ <a href="#">Expressions</a> ). This expression can return one or more names of active directory user group names.

<b>Send to E-Mail Addresses</b>	If you tick this option, the e-mail will be sent to all e-mail addresses resulting from the given expression (→ <a href="#">Expressions</a> ).
---------------------------------	--

### Static E-Mail Recipients

The list of static e-mail recipients let's you manage users and/or user groups who will receive the e-mail. These recipients can be managed using the following buttons:

	Click this button to add a new e-mail recipient; a context menu will open that allows you to select which type of e-mail recipient you want to add.
	Click this button to remove the currently selected e-mail recipient.

## Distribute Report Document to Hub Report Document Library Action

This action can be used to distribute a previously created report document (→ [Create Report Document Action](#)) to the hub report library (→ [Library](#)) of users and/or user groups.

### General Attributes

You can set the following general attributes:

<b>Report Document</b>	The report document you want to distribute. This needs to be a report document that has been created by a <a href="#">Create Report Document Action</a> in the same task.
<b>Name</b>	An expression (→ <a href="#">Expressions</a> ) that evaluates to the name the report document shall have in the library.
<b>Description</b>	An expression (→ <a href="#">Expressions</a> ) that evaluates to the description the report document shall have in the library.

### Recipients



This section lets you control to which users and/or user groups you want to distribute the report document.

The following basic attributes can be set:

<b>Distribute to Assigned Users</b>	<b>Dynamically</b>	If you tick this option, the report document will be distributed to the library of users that have been dynamically assigned (→ <a href="#">Dynamically Assign Users to Report Document Action</a> ) to the report document.
-------------------------------------	--------------------	--

### Static Recipients

The list of static recipients let's you manage users and/or user groups who will get the report document into their library. These recipients can be managed using the following buttons:

	Click this button to add a new recipient; a context menu will open that allows you to select which type of recipient you want to add.
	Click this button to remove the currently selected recipient.

## Dynamically Assign E-Mail Addresses to Report Document Action

This action can be used to dynamically assign e-mail addresses to a report document that has been previously created (→ [Create Report Document Action](#)).

**Example:** Let's suppose you have created a report document that contains data for a specific store of your retail network. In the datasource (e.g. the QlikView Document) there is a field that contains the e-mail addresses of all persons who should receive the report document. In this case, you don't want to assign static e-mail addresses when distributing the report (→ [Static E-Mail Recipients](#)); instead, you want Mail & Deploy to dynamically read these e-mail addresses from the datasource and distribute the report document to them.

The following attributes can be set:

<b>Report Document</b>	The report document you want to assign dynamic e-mail addresses to. This needs to be a report document that has been created by a <a href="#">Create Report Document Action</a> in the same task.
<b>Type</b>	<p>The type of e-mail address assignment that determines the method to be used to identify e-mail addresses to assign to the report document.</p> <p><i>Expression Evaluating to E-Mail Addresses.</i> If you use this type, you can enter a match expression (→ <a href="#">Expressions</a>). The expression needs to return zero, one, or more e-mail addresses which will be assigned to the report document. If a single value returned by the expression can contain more than one value, you can specify which character shall be used to separate these values into individual e-mail addresses.</p> <p><i>User Custom Property.</i> If you use this type, you can select a custom property (→ <a href="#">Custom Properties</a>) and a match expression (→ <a href="#">Expressions</a>). When executing the action, Mail &amp; Deploy will evaluate the expression and assign the report document to the e-mail addresses of all users (→ <a href="#">Users</a>) where the given custom property has at least one of the values that the match expression returns.</p> <p><i>User Windows Account Name.</i> If you use this type, you can enter a match expression (→ <a href="#">Expressions</a>). The expression needs to return zero, one, or more windows account names. The e-mail addresses of all users having one of the returned windows account names will be assigned to the report document.</p>

## Dynamically Assign Users to Report Document Action

This action can be used to dynamically assign users (→ [Users](#)) to a report document that has been created by a [Create Report Document Action](#).

**Example:** Let's suppose you have created a report document that contains data for a specific store of your retail network. In the datasource (e.g. the QlikView Document) there is a field that contains the windows username or human resource ID (or something similar) of all persons who should receive the report document. Let's further suppose you want to distribute the created report document to the hub report library (→ [Library](#)) of these persons using a [Distribute Report Document to Hub Report Document Library Action](#). In such a case, you don't want to add static users to distribute the report document to; instead, you want to dynamically assign the users based on the field in the QlikView Document.

The following attributes can be set:

<b>Report Document</b>	The report document you want to assign dynamic users to. This needs to be a report document that has been created by a <a href="#">Create Report Document Action</a> in the same task.
<b>Type</b>	<p>The type of user assignment that determines the method to be used to identify users to assign to the report document.</p> <p><i>User Custom Property.</i> If you use this type, you can select a custom property (→ <a href="#">Custom Properties</a>) and a match expression (→ <a href="#">Expressions</a>). When executing the action, Mail &amp; Deploy will evaluate the expression and assign the report document to all users (→ <a href="#">Users</a>) where the given custom property has at least one of the values that the match expression returns.</p> <p><i>User Windows Account Name.</i> If you use this type, you can enter a match expression (→ <a href="#">Expressions</a>). The expression needs to return zero, one, or more windows account names. All users having one of the returned windows account names will be assigned to the report document.</p>

## Filter Datasource Field Action

This action can be used to filter a field in a datasource to one or more values. The following attributes can be set:

Datasource Field	The field in the datasource (→ <a href="#">Datasources</a> ) that shall be filtered.
Values	A list of expressions (→ <a href="#">Expressions</a> ) which evaluate to the values to which the datasource field shall be filtered.
Perform Match Filter	<p>This setting determines how the list of values is used to filter the field in the datasource.</p> <p><i>True.</i> In this case, the values are filtered by applying a match filter, which means that you can use datasource specific wildcards and/or operators to perform the filters (e.g. a field that contains numbers can be filtered to the match "&gt;500" to select all values which are greater than 500). The syntax to use for the match depends on the type of datasource.</p> <p><i>False.</i> In this case, the exact values will be filtered in the datasource.</p>
Invert Filter	<p>This setting determines whether the filter shall be inverted, which works like a NOT operator.</p> <p><i>True.</i> In this case, the values are filtered and then the filter is inverted, so that all values of the datasource field, are filtered, except the ones you have provided.</p> <p><i>False.</i> In this case, the values you have provided will be filtered in the datasource field without inverting the filter.</p>
Check Type	<p>This setting determines whether the filter will be checked after all other filters have been applied in the datasource. This is particularly useful if you have more than one filter applied to the datasource; if that is the case, any filter that is applied could remove a previously applied filter if there is no applicable values.</p> <p><b>Example:</b> Let's suppose you want to apply two filters to a QlikView Document; first you filter the field "Year" to the value "2018". Thereafter you filter the field "CustomerId" to the value "123". The application of the second filter could lead to the first filter to be removed, if the customer with the ID "123" does not have any association with the year "2018". To prevent this, a filter check can be added to the filter of the field "Year"; this way, Mail &amp; Deploy will perform the filter check after all filters have been applied and would recognize, that the filter has been removed.</p> <p><i>None.</i> No filter check will be performed and the condition described above would not be recognized by Mail &amp; Deploy.</p> <p><i>At least one of the values needs to be filtered.</i> This filter check tests whether at least one of the values of the value list is filtered after all other filters have been applied.</p> <p><i>All values need to be filtered.</i> This filter check tests whether all the values of the value list are filtered after all other filters have been applied.</p> <p>The consequence of a failed filter check depends on the context:</p> <ul style="list-style-type: none"> <li>➤ If the filter is used as a preparation action of a report (→ <a href="#">Reports</a>), the creation of the report document will be cancelled.</li> <li>➤ If the filter is used as a preparation action of a report element (→ <a href="#">Report Elements</a><b>Error! Reference source not found.</b>) you can choose yourself what should happen if a filter check fails by setting the <i>Filter Check Failed Behaviour</i> attribute of the report element.</li> </ul>

## Group Action

This action can be used to group other actions. This is useful if you want to apply a specific exception behaviour to a certain group of actions (→ [Exception Handling](#)) or if you want to only execute a group of actions if a condition is fulfilled. It has the following attributes:



## General Attributes

<b>Name</b>	The name of the group action; this is only used within Mail & Deploy to identify the group action.
<b>Condition</b>	An optional expression (→ <a href="#">Expressions</a> ) that evaluates to either <i>True</i> or <i>False</i> . The actions of the group will only be executed if the condition evaluates to <i>True</i> .
<b>Exception Behaviour</b>	<p>Determines what Mail &amp; Deploy shall do if the execution of one of the actions of the group causes an exception (→ <a href="#">Exception Handling</a>).</p> <p><i>Do Not Handle.</i> If you select this behaviour, any exception caught by the group action will not be handled and passed on to the parent action container.</p> <p><i>Cancel Group.</i> If you select this behaviour, any exception caught by the group action will lead to the complete group to be cancelled, but no exception will be passed on to the parent action container.</p> <p><i>Continue With Next Action.</i> If you select this behaviour, any exception caught by the group will lead to the group execution to be continued with the next action. No exception will be passed on to the parent action container.</p>

## Actions

This is a list of actions (→ [Actions](#)) which are sub-actions of the group.

### Preparation Actions

This is a list of actions (→ [Actions](#)) which will be executed before the condition will be evaluated and the sub-actions of the group will be executed.

## Initialize Datasource Object Action

This action can be used to initialize a datasource object (→ [Objects](#)) which is required if an initialized datasource object is required for the creation of a report, e.g. when using trueChart report elements (→ [trueChart Image Elements](#) and → [trueChart Table Elements](#)). It has the following attributes:

<b>Datasource Object</b>	The datasource object (→ <a href="#">Objects</a> ) you want to initialize. Only datasource objects that support initialization may be selected.
--------------------------	---

## Import User Custom Properties From File Action

This action can be used to import custom properties of users (→ [Custom Properties](#)) from a file. It has the following attributes:

<b>File Path</b>	<p>The full path to the file you want to import from. The file path needs to include the file type extension. It's important to note that the datasource will be saved to the file system by Mail &amp; Deploy Server and therefore the following requirements have to be met:</p> <ul style="list-style-type: none"> <li>➤ The user that runs Mail &amp; Deploy Server needs to have sufficient user rights to write to the directory.</li> <li>➤ The file path needs to be reachable from the computer on which Mail &amp; Deploy Server runs; this might require to use UNC file paths in some cases. Please note that mapped network drive letters cannot be used, even if the user that runs Mail &amp; Deploy Server has a mapped network drive set up.</li> </ul>
------------------	--

## File Format

The file has the following format:

```
<username>;<custompropertyname>;<custompropertydatatype>;<custompropertyvalue>
...
```

The values of each line are as follows:

<b>Username</b>	The name of the user (→ <a href="#">Users</a> ) for which the custom property value shall be added.
<b>Custom Property Name</b>	The name of the custom property.
<b>Custom Property Data Type</b>	The data type of the value of the custom property.  <i>String.</i> If the data type of the value of the custom property shall be in textual format.  <i>DateTime.</i> If the data type of the value of the custom property shall be in date-time format.  <i>Integer.</i> If the data type of the value of the custom property shall be an integer.  <i>Decimal.</i> If the data type of the value of the custom property shall be a decimal.
<b>Custom Property Value</b>	The textual representation of the value of the custom property. This value will be converted into the actual data type according to the attribute <i>Custom Property Data Type</i> .

Each line represents one value for a custom property; thus, if you want to assign more than one value to a single custom property, you need to have one line for each value.



Please note, that before custom properties are imported from the file, all custom properties of users will be deleted. Deleted custom properties cannot be recovered, except by reverting to a backup of the repository.

## Rebuild Datasource Cache Action

This action can be used to rebuild the cache that contains the elements of a datasource (→ [Datasource Elements](#)) and has the following attributes:

<b>Datasource</b>	The datasource (→ <a href="#">Datasources</a> ) to rebuild the datasource element cache for.
-------------------	--

## Reload Datasource Action

This action can be used to reload a datasource and has the following attributes:

<b>Datasource</b>	The datasource (→ <a href="#">Datasources</a> ) to reload.
-------------------	--



Please note, that reloading a QlikView Document datasource does not save the QlikView Document; the reloaded data is only available for the duration of the task in which the reload took place. In order to save the reloaded datasource, you need to add a [Save Datasource Action](#).

## Remove Datasource Field Filter Restriction Action

This action can be used to remove field filter restrictions (such as “Always one selected value”) from a datasource field. The following attributes can be set:

<b>Datasource Field</b>	The field in the datasource (→ <a href="#">Datasources</a> ) from which filter restrictions shall be removed.
-------------------------	---



Please note, that removing field filter restrictions currently only works for QlikView Documents where Mail & Deploy will remove “Always one selected value” from that field.

## Remove All Files From File Collection Action

This action can be used to remove all files from a file collection. The following attributes can be set:

<b>File Collection</b>	The file collection from which to remove all files.
------------------------	---

## Run Application Action

This action can be used to start an application or a batch file from a given file location. Mail & Deploy will consider the application to have successfully completed if it returns exit code 0. The following attributes can be set:

<b>Application Path</b>	An expression (→ <a href="#">Expressions</a> ) that evaluates to the full path to the application or batch file to be executed.
<b>Arguments</b>	An optional expression (→ <a href="#">Expressions</a> ) that evaluates to the arguments to be passed on to the application or batch file.
<b>Number of Retries</b>	The maximum number of retries, should the application not finish with exit code 0.

## Restore Datasource Filter State Action

This action can be used to restore the filter state of a datasource (→ [Datasources](#)) that has previously been created (→ [Save Datasource Filter State Action](#)). The following attributes can be set:

<b>Datasource</b>	The datasource (→ <a href="#">Datasources</a> ) of which to restore a filter state.
<b>Name</b>	The name of the filter state to restore.

## Save Datasource Action

This action can be used to save a datasource to the location it has originally been loaded from and has the following attributes:

<b>Datasource</b>	The datasource (→ <a href="#">Datasources</a> ) to save.
-------------------	--



Please note, that saving a datasource is not available for all datasource types.

## Save Datasource Filter State Action

This action allows you to save the complete filter state of a datasource (→ [Datasources](#)) under a given name. The filter state can later be restored (→ [Restore Datasource Filter State Action](#)). The following attributes can be set:

<b>Datasource</b>	The datasource (→ <a href="#">Datasources</a> ) of which to save the filter state.
<b>Name</b>	The name of the filter state to save.

## Save Datasource to File System Action

This action can be used to save a datasource (→ [Datasources](#)) to the file system.



Please note, that saving a datasource to the file system is not available for all datasource types.

The following attributes can be set:

<b>Datasource</b>	The datasource (→ <a href="#">Datasources</a> ) to reduce and then save to the file system.
<b>File Path</b>	<p>An expression (→ <a href="#">Expressions</a>) that evaluates to the full path of the file to save the datasource to. The file path needs to include the file type extension. It's important to note that the datasource will be saved to the file system by Mail &amp; Deploy Server and therefore the following requirements have to be met:</p> <ul style="list-style-type: none"> <li>➤ The user that runs Mail &amp; Deploy Server needs to have sufficient user rights to write to the directory.</li> <li>➤ The file path needs to be reachable from the computer on which Mail &amp; Deploy Server runs; this might require to use UNC file paths in some cases. Please note that mapped network drive letters cannot be used, even if the user that runs Mail &amp; Deploy Server has a mapped network drive set up.</li> </ul>

## Preparation Actions

This is a list of actions (→ [Actions](#)) which will be executed before the datasource will be saved to the file system.

## Save Reduced Datasource to File System Action

This action can be used to save the reduced version of a datasource (→ [Datasources](#)) to the file system. The current state of the datasource is used for the reduction, so all the data that is compatible with the current filter of the datasource will be included in the reduced version.



Please note, that saving a reduced version of a datasource to the file system is not available for all datasource types.

The following attributes can be set:

<b>Datasource</b>	The datasource (→ <a href="#">Datasources</a> ) to reduce and then save to the file system.
<b>File Path</b>	<p>An expression (→ <a href="#">Expressions</a>) that evaluates to the full path of the file to save the reduced datasource to. The file path needs to include the file type extension. It's important to note that the reduced datasource will be saved to the file system by Mail &amp; Deploy Server and therefore the following requirements have to be met:</p>

- The user that runs Mail & Deploy Server needs to have sufficient user rights to write to the directory.
- The file path needs to be reachable from the computer on which Mail & Deploy Server runs; this might require to use UNC file paths in some cases. Please note that mapped network drive letters cannot be used, even if the user that runs Mail & Deploy Server has a mapped network drive set up.

## Preparation Actions

This is a list of actions (→ [Actions](#)) which will be executed before the reduced datasource will be saved to the file system. These actions can – for instance – be used to filter the datasource to control which data will be included in the reduced versions of the datasource.

## Save Report Document to File System Action

This action can be used to save a previously created report document (→ [Create Report Document Action](#)) to the file system. The following attributes can be set:

<b>Report Document</b>	The previously created report document to save to the file system.
<b>File Path</b>	<p>An Expression (→ <a href="#">Expressions</a>) that evaluates to the full path of the file to save the report document to. The file path needs to be without the file type extension, because Mail &amp; Deploy will add the extension automatically. It's important to note that the report document will be saved to the file system by Mail &amp; Deploy Server and therefore the following requirements have to be met:</p> <ul style="list-style-type: none"> <li>➤ The user that runs Mail &amp; Deploy Server needs to have sufficient user rights to write to the directory.</li> <li>➤ The file path needs to be reachable from the computer on which Mail &amp; Deploy Server runs; this might require to use UNC file paths in some cases. Please note that mapped network drive letters cannot be used, even if the user that runs Mail &amp; Deploy Server has a mapped network drive set up.</li> </ul>

## Save Report Document as Images to File System Action

This action can be used to save individual pages of a previously created report document (→ [Create Report Document Action](#)) to the file system. Each page of the report document will be saved as an individual image file; the following table indicates what a page is in regards to the file type of the created report document:

- When the report document is an *Adobe PDF Document* or an *Adobe PDF/A Document*, each page of the PDF document will be saved to an individual image file.
- When the report document is a *Microsoft Excel Workbook*, each worksheet of that workbook will be treated as a page and thus saved to an individual image file.
- When the report document is a *Microsoft PowerPoint Presentation*, each slide of that presentation will be treated as a page and thus saved to an individual image file.
- When the report document is a *Microsoft Word Document*, each page of that document will be saved to an individual image file.

All other file types cannot be converted to images. The following attributes can be set:

<b>Report Document</b>	The previously created report document to save as images to the file system.
------------------------	--

<b>Image Type</b>	<p>The file type to create the images in.</p> <p><i>BMP Image.</i> If you select this file type, bitmap images will be created.</p> <p><i>PNG Image.</i> If you select this file type, PNG images will be created.</p> <p><i>JPEG Image.</i> If you select this file type, JPEG images will be created.</p>
<b>DPI</b>	The DPI setting to apply to saved images.
<b>File Path</b>	<p>An Expression (→ <a href="#">Expressions</a>) that evaluates to the full path of the file to save the image to. The file path needs to be without the file type extension, because Mail &amp; Deploy will add the extension automatically. It's important to note that the image will be saved to the file system by Mail &amp; Deploy Server and therefore the following requirements have to be met:</p> <ul style="list-style-type: none"> <li>➤ The user that runs Mail &amp; Deploy Server needs to have sufficient user rights to write to the directory.</li> <li>➤ The file path needs to be reachable from the computer on which Mail &amp; Deploy Server runs; this might require to use UNC file paths in some cases. Please note that mapped network drive letters cannot be used, even if the user that runs Mail &amp; Deploy Server has a mapped network drive set up.</li> </ul> <p>The following expression parameters (→ <a href="#">Expression Parameter</a>) are available:</p> <p><i>PageNumber.</i> The page for which an image is currently being created.</p>

## Set Datasource Variable Value Action

This action can be used to set the value of a datasource variable (→ [Variables](#)) to a given value. The following attributes can be set:

<b>Datasource Variable</b>	The datasource variable (→ <a href="#">Variables</a> ) you want to set.
<b>Value</b>	An expression (→ <a href="#">Expressions</a> ) that evaluates to the value to set the datasource variable to.

## Set Local Variable Value Action

This action can be used to set a local variables to a given value. Local variables can be used within the currently executed task (→ [Tasks](#)) and are particularly helpful if you have an expression (→ [Expressions](#)) that you need more than once within the context of the task; in that case you do not have to rewrite them at any place you want to access them; instead, you can store the return value of the expression in a local variable and then re-use it in expressions (→ [Local Variables](#)). The following attributes can be set:

<b>Variable Name</b>	The name of the local variable; this name can be later used to reference the local variable in expressions (→ <a href="#">Local Variables</a> ).
<b>Value</b>	An expression (→ <a href="#">Expressions</a> ) that evaluates to the value to set the local variable to.

## Synchronize Users and User Groups with Active Directory Action

This action can be used to synchronize users and user groups of a repository (→ [Users and User Groups](#)) with an active directory or the local user and user group directory if the computer that runs Mail & Deploy Server is not member of a domain. The following requirements need to be met in order to synchronize users and user groups with an active directory:

- The computer that runs Mail & Deploy Server needs to be member of a domain; if that is not the case, Mail & Deploy can only synchronize with the local user and user group directory of that computer.
- The repository needs to have valid active directory connection settings (→ [Active Directory Connection Settings](#)).

A synchronization action contains a list of active directory (or local) user groups which shall be synchronized with Mail & Deploy. You can manage these groups using the following buttons:



Click this button to create new settings for the synchronization of an active directory (or local) user group.



Click this button to edit the currently selected synchronization settings.



Click this button to create a copy the currently selected synchronization settings.



Click this button to delete the currently selected synchronization settings.

For each synchronization settings, the following attributes can be defined:

<b>Active Directory User Group</b>	The active directory (or local) user group that shall be synchronized with Mail & Deploy.
<b>Synchronize as User Group</b>	<p>This setting determines whether the active directory (or local) user group shall also be added as user group in Mail &amp; Deploy.</p> <p><i>True.</i> All users of the active directory (or local) user group will be added as users in Mail &amp; Deploy and a user group will be created in Mail &amp; Deploy that contains the same users as the active directory (or local) user group does.</p> <p><i>False.</i> All users of the active directory (or local) user group will be added as users in Mail &amp; Deploy but no user group will be created in Mail &amp; Deploy.</p>
<b>Role</b>	The role to assign to users synchronized with the active directory (or local) user group.
<b>Default Language</b>	The default language to assign to users synchronized with the active directory (or local) user group. This language is only used if the user does not have the preferred language property set in the active directory (or local) user record.
<b>Time Zone</b>	The time zone to assign to users synchronized with the active directory (or local) user group.
<b>Culture</b>	The culture to assign to users synchronized with the active directory (or local) user group.
<b>Custom Properties</b>	The custom properties to assign to users synchronized with the active directory (or local) user group (→ <a href="#">Custom Properties</a> ).

## Upload Report Document to FTP Action

This action can be used to upload a previously created report document (→ [Create Report Document Action](#)) to an FTP server. The following attributes can be set:

Content Type	<i>Report Document.</i> If you select this option, you can choose a report document to upload to FTP.
	<i>Zipped File Collection.</i> If you select this option you can choose a file collection which will be zipped and optionally assigned a password and uploaded to FTP.
Host	The host name or IP address of the FTP server.
Port	The port of the FTP server.
SFTP	<i>True.</i> The SSH File Transfer Protocol (SFTP) is used for the upload.
	<i>False.</i> The File Transfer Protocol (FTP) is used for the upload.
Use Credentials	<i>True.</i> The given credentials are used to authenticate against the FTP server. In this case, you need to specify a <i>Username</i> and <i>Password</i> .
	<i>False.</i> The FTP upload will be done using anonymous access to the FTP server.
Remote File Path	An Expression (→ <a href="#">Expressions</a> ) that evaluates to the remote path to upload the file to. The remote path needs to include the file name (without the file extension) and folders need to be separated using a forward slash (/). Mail & Deploy will automatically ensure that the required directory structure exists by creating directories if they don't exist.

**Example.** If you have specified a host name of "testftpsrvr", not checked SFTP, a port of 21 and a remote path of "folder1/folder2/filename", then the file will be uploaded to "ftp://testftpsrvr:21/folder1/folder2/filename".

## Synchronize Users and User Groups with File Action

This action can be used to synchronize users and user groups of a repository (→ [Users and User Groups](#)) with a CSV file. The following attributes can be set:

File Path	<p>The full path of the file to synchronize with. The file path needs to include the file type extension. It's important to note that the file will be saved to the file system by Mail &amp; Deploy Server and therefore the following requirements have to be met:</p> <ul style="list-style-type: none"> <li>➤ The user that runs Mail &amp; Deploy Server needs to have sufficient user rights to read from the directory.</li> <li>➤ The file path needs to be reachable from the computer on which Mail &amp; Deploy Server runs; this might require to use UNC file paths in some cases. Please note that mapped network drive letters cannot be used, even if the user that runs Mail &amp; Deploy Server has a mapped network drive set up.</li> </ul>
-----------	---

### File Format

The CSV file to synchronize users and user groups with contains one entry per line.

#### User Group Entry

A user group entry has the following format, where the first value declares the line to be a user group entry:

```
usergroup;<name>
...
```

The values of each line are as follows:



Name	The name of the user group.
------	-----------------------------

## User Entry

A user entry has the following format, where the first value declares the line to be a user entry:

```
user;<id>;<name>;<role>;<language>;<timezone>;<culture>;<has_credentials>;<credentials_type>;<credentials_username>;<credentials_password>;<emailaddress>;<usergroup_name_1>;<usergroup_name_2>;...
```

The values of each line are as follows:

ID	The ID of the user. This identifies this user throughout the synchronization process.
Name	The name of the user.
Role	<p>The role (→ <a href="#">Roles</a>) of the user. This can be one of the following values:</p> <ul style="list-style-type: none"> <li>0. This value represents the role <i>None</i>.</li> <li>1. This value represents the role <i>Hub Consumer</i>.</li> <li>2. This value represents the role <i>API Consumer</i>.</li> <li>3. This value represents the role <i>Designer</i>.</li> <li>4. This value represents the role <i>Administrator</i>.</li> </ul>
Language	<p>The code of the language of the user. This can be one of the following values:</p> <ul style="list-style-type: none"> <li><i>en</i>. This value represents English.</li> <li><i>de</i>. This value represents German.</li> </ul>
Time Zone	The name of the time zone of the user. If you leave this empty, the local time zone of Mail & Deploy Server will be used. Alternatively, you can provide the name of a windows time zone (🌐 <a href="#">Windows Default Time Zones</a> ) by using the column <i>Timezone</i> of the linked table.
Culture	The ID of the culture of the user. If you leave this empty, the local culture of Mail & Deploy Server will be used. Alternatively, you can provide the ID of a windows culture (🌐 <a href="#">Locale Codes</a> ) by using the column <i>LCID Decimal</i> of the linked table.
Has Credentials	<p>Determines whether the user has credentials (→ <a href="#">Credentials</a>) or not.</p> <ul style="list-style-type: none"> <li><i>False</i>. This user does not have credentials.</li> <li><i>True</i>. This user does have credentials.</li> </ul>
Credentials Type	<p>The type of credentials of this user. This is only applicable, if the user does have credentials. Can be one of the following values:</p> <ul style="list-style-type: none"> <li><i>Custom</i>. This user has custom credentials.</li> <li><i>Windows</i>. This user has windows credentials.</li> </ul>
Credentials Username	The username of the credentials. This is only applicable if the user has credentials.

Credentials Password	The password of the credentials. This is only applicable if the user has custom credentials.
E-Mail Address	The e-mail address of the user.
User Group Names	An optional list of the names of all user groups to which the user belongs.

## Write Text File Action

This action can be used to write text to a file. The following attributes can be defined:

Mode	<p>The mode to apply when writing the file.</p> <p><i>Create New File.</i> If you use this mode, a new file will be created. Any existing file will be deleted first.</p> <p><i>Append To File.</i> If you use this mode, the content will be appended to an existing file at the given file path.</p>
File Path	<p>An expression (→ <a href="#">Expressions</a>) that evaluates to the full path of the file to write to. The file path needs to include the file type extension. It's important to note that the file will be saved to the file system by Mail &amp; Deploy Server and therefore the following requirements have to be met:</p> <ul style="list-style-type: none"><li>➤ The user that runs Mail &amp; Deploy Server needs to have sufficient user rights to write to the directory.</li><li>➤ The file path needs to be reachable from the computer on which Mail &amp; Deploy Server runs; this might require to use UNC file paths in some cases. Please note that mapped network drive letters cannot be used, even if the user that runs Mail &amp; Deploy Server has a mapped network drive set up.</li></ul>
Content	An expression (→ <a href="#">Expressions</a> ) that evaluates to the content to write.

# Expressions

## Introduction

In Mail & Deploy, expressions can be used to supply values to certain functions; expressions can either return constant values (→ [Constant Values](#)), dynamic values (→ [Dynamic Values](#)) or values calculated from constant and/or dynamic values (→ [Calculations](#)).

**Example:** Let's suppose you want to filter a datasource field (→ [Filter Datasource Field Action](#)). The list of values you want the field to be filtered to, is a list of expressions, which will be evaluated just before the field is filtered.

The language in which expressions are written in Mail & Deploy, is Visual Basic .NET; however, in most cases you won't be required to have deeper knowledge of the language. Expressions are edited in an expression editor, which provides the following buttons:



Click this button to choose a built-in value to insert.



Click this button to open a larger expression editor, which gives you more space to edit the expression.



Click this button to open an editor to edit the module (→ [Module](#)) of the current repository.

## Constant Values

If you want to use a constant, non-calculated value in an expression, you can do this by using the following syntax, which depends on the data type of the constant you want to use.

**Text.** If you want to write a text constant, the following syntax may be used:

```
"Hello World!"
```

**Number.** If you want to write a numeric constant, the following syntax may be used:

```
2018
```

**Date.** If you want to write a date constant, the following syntax may be used:

```
#2018-12-31#
```

For instance, if you simply want the expression to return the number 2018, the expression looks like this:

```
2018
```


If you want the expression to return the text *Hello World*, the expression looks like this:


```
"Hello World"
```


## Inserting Constant Values

In some cases, you don't want to type a constant value but rather pick it from a list of values which is offered by Mail & Deploy.

**Example:** Let's suppose you want to activate a bookmark with a specific name in a datasource (→ [Activate Datasource Bookmark Action](#)). You can either type the name of the bookmark as text constant, or you can choose to insert the constant value, where you will be able to pick the name of the bookmark from a list.

You can insert constant values by clicking the  button in the expression editor and then choosing a constant value type of your choice; however, not all the constant values listed below are available at all times; some of them are only available if the context in which the expression is used allows it.


**Datasource Bookmark Name.** If you choose this option, a new dialog will appear that allows you to pick one of the bookmarks (→ [Bookmarks](#)) of one of the datasources (→ [Datasources](#)). If you then click the  button, a constant value will be inserted into the expression that equals the name of the selected bookmark.

**Datasource Embedded Report Name.** If you choose this option, a new dialog will appear that allows you to pick one of the embedded reports (→ [Embedded Reports](#)) of one of the datasources (→ [Datasources](#)). If you then click the  button, a constant value will be inserted into the expression that equals the name of the selected embedded report.

**Horizontal Alignment.** If you choose this option, you can select a horizontal alignment; a constant value will be inserted into the expression that equals the selected horizontal alignment. This constant value is not always available.

**Vertical Alignment.** If you choose this option, you can select a vertical alignment; a constant value will be inserted into the expression that equals the selected vertical alignment. This constant value is not always available.

**Line Type.** If you choose this option, you can select a line type; a constant value will be inserted into the expression that equals the selected line type. This constant value is not always available.

**Printer Name.** If you choose this option, a new dialog will appear that allows you to pick one of the printers that is installed on the computer that runs Mail & Deploy Server (→ [Server](#)). If you then click the  button, a constant value will be inserted into the expression that equals the name of the selected printer.

## Dynamic Values

In expressions, dynamic values are values which are dynamically obtained by the expression at the time of its evaluation.


**Example:** Let's suppose you want to add a text report element (→ [Text Elements](#)) to a report and you want the text element to contain the current value of the datasource field "Year". In that case, you would use a dynamic value that will be evaluated by Mail & Deploy when creating the text element.

The exact syntax of dynamic values depends on the type of dynamic value.

### Report Document Tag

This dynamic value can be used to insert the value of a tag of a report document (→ [Tags](#)) into the expression. If you want to manually type in the dynamic value, you can use the following syntax:


```
[ReportDocumentTag].[<name_of_report_document>].[<name_of_tag>]
```

If you want to automatically insert the syntax for this dynamic value, you can click the  button in the expression editor, then click *Report Document Tag*, then choose the name of the report document and then click the name of the tag you want to insert.

### Datasource Field Value

This dynamic value can be used to insert the values of a datasource field (→ [Fields](#)). Only values which match the current filter state of the datasource will be added. If you want to manually type in the dynamic value, you can use the following syntax:


```
[Datasource].[<name_of_datasource>].[Field].[<name_of_field>]
```

If you want to automatically insert the syntax for this dynamic value, you can click the  button in the expression editor and then click *Datasource Field Value*. A new dialog will appear which allows you to select the datasource field whose value you want to access.

### Datasource Variable Value

This dynamic value can be used to insert the value of a datasource variable (→ [Variables](#)). If you want to manually type in the dynamic value, you can use the following syntax:


```
[Datasource].[<name_of_datasource>].[Variable].[<name_of_variable>]
```

If you want to automatically insert the syntax for this dynamic value you can click the  button in the expression editor and then click *Datasource Variable Value*. A new dialog will appear which allows you to select the datasource variable whose value you want to access.

## Datasource Expression

This dynamic value can be used to insert the return value of an expression which can be written in the syntax of the datasource used. If you want to manually type in the dynamic value, you can use the following syntax:

```
[Datasource].[<name_of_datasource>].[Expression].[<expression_code>]
```


If you want to automatically insert the syntax for this dynamic value you can click the  button in the expression editor, then click *Datasource Expression*, and then click the name of the datasource for which you want to insert an expression. The code for the dynamic value will be inserted into the expression editor and you can immediately start typing the expression code in the syntax depending on your datasource.

**Example:** Let's suppose you want to add the maximum value of the datasource field "Revenue" into the expression. The datasource used is a QlikView Document. If there is no datasource field or variable that contains the maximum value of "Revenue", you can choose to add a datasource expression and then use the syntax "[Datasource].[<datasource\_name>].[Expression].[max(Revenue)]". The value between the last brackets can be any expression syntax that is supported by the type of datasource used.

## Datasource Object Title

This dynamic value can be used to insert the title of a datasource object (→ [Objects](#)). If you want to manually type in the dynamic value, you can use the following syntax:


```
[Datasource].[<name_of_datasource>].[ObjectTitle].[<object_id>]
```

If you want to automatically insert the syntax for this dynamic value you can click the  button in the expression editor, then click *Datasource Object Title*, and then select the datasource object for which you want to insert an expression. The code will then automatically be inserted into the expression editor.

## Report Parameter

This dynamic value can be used to insert the value of a report parameter (→ [Report Parameters](#)). This type of dynamic value is only available in expressions in the context of a report. If you want to manually type in the dynamic value, you can use the following syntax:


```
[ReportParameter].[<name_of_reportparameter>]
```

If you want to automatically insert the syntax for this dynamic value you can click the  button in the expression editor, then click *Report Parameter* and then click the name of the report parameter whose value you want to insert.

## Task Parameter

This dynamic value can be used to insert the value of a task parameter (→ [Task Parameters](#)). This type of dynamic value is only available in expressions in the context of a task. If you want to manually type in the dynamic value, you can use the following syntax:


```
[TaskParameter].[<name_of_taskparameter>]
```

If you want to automatically insert the syntax for this dynamic value you can click the  button in the expression editor, then click *Task Parameter* and then click the name of the task parameter whose value you want to insert.

## Expression Parameter

This dynamic value can be used to insert the value of an expression parameter; expression parameters are additional parameters which are only available in certain contexts. If you want to manually type in the dynamic value, you can use the following syntax:


```
[<name_of_expressionparameter>]
```

If you want to automatically insert the syntax for this dynamic value you can click the  button in the expression editor, then click *Expression Parameter* and then click the name of the expression parameter whose value you want to insert.

## Parallelization Parameter

This dynamic value can be used to insert the parallelization parameter (→ [Parallelization](#)); this is only available in expressions within a task that uses parallelization. If you want to manually type in the dynamic value, you can use the following syntax:


```
[ParallelizationParameter].[Values]
```

If you want to automatically insert the syntax for this dynamic value you can click the  button in the expression editor, then click *Parallelization Parameter*.


## Cycle Value

This dynamic value can be used to insert the current value of a cycle (→ [Cycle Action](#)). This type of dynamic value is only available in expressions in the context of a cycle action. If you want to manually type in the dynamic value, you can use the following syntax:


```
[CycleValue].[<name_of_cyclevalue>]
```

If you want to automatically insert the syntax for this dynamic value you can click the  button in the expression editor, then click *Cycle Value* and then click the name of the cycle value whose value you want to insert.


## Color

This dynamic value can be used to insert a color. This type of dynamic value is only available in certain contexts. If you want to automatically insert the syntax for this dynamic value you can click the  button in the expression editor, then click *Color*; a dialog will appear in which you can configure the color to insert.

## Local Variables

This dynamic value can be used to insert the value of a local variable that has been previously set to a value (→ [Set Local Variable Value Action](#)). If you want to automatically insert the syntax for this dynamic value you can click the  button in the expression editor, then click *Local Variable*; a dialog will appear in which you can select the local variable to insert.

## Global Variables

This dynamic value can be used to insert the value of a global variable (→ [Global Variables](#)). If you want to automatically insert the syntax for this dynamic value you can click the  button in the expression editor, then click *Global Variable*; a dialog will appear in which you can select the global variable to insert.



Please note, that global variables are calculated everytime you use them in expressions. If you reference a global variable multiple times, it will be evaluated for each of these references individually at the time of the evaluation.

## Calculations

The previously mentioned constant values (→ [Constant Values](#)) and dynamic values (→ [Dynamic Values](#)) can not only be used on their own, but they can be used in calculations by using the following operators:

&	This operator can be used to concatenate two text values.
+	This operator can be used to add up two numeric values.
-	This operator can be used to subtract the numeric value right of the operator from the numeric value left of the operator.
/	This operator can be used to divide the numeric value left of the operator by the numeric value right of the operator.
*	This operator can be used to multiply two numeric values.

The syntax to concatenate the texts *Hello* and *World* is as follows:

```
"Hello" & " World"
```

You cannot only use constant values in these expressions; you could, for example, use the following expression to add value of the datasource field *NetAmount* and the value of the datasource field *VatAmount*; we assume here that these fields are both in the datasource *Invoicing*.

```
[Datasource].[Invoicing].[Field].[NetAmount] + [Datasource].[Invoicing].[Field].[VatAmount]
```

You can also mix constant values and dynamic values in these calculations; if you want to divide the datasource field *YearlyRevenue* of datasource *Invoicing* by 12 (e.g. in order to get an average monthly revenue value), you can use the following syntax:

```
[Datasource].[Invoicing].[Field].[YearlyRevenue] / 12
```

The dynamic values used do not necessarily have to belong to the same datasource; you could also write expressions which calculate values from different datasources, combined with constants.

## Data Types

Depending on the data-type of the constant value (→ [Constant Values](#)) or dynamic value (→ [Dynamic Values](#)), you can access certain properties of the value and/or manipulate it.

### Date-Time

More information about all available functions and properties of date-times can be found in the documentation of Microsoft .NET Framework ([🌐 .NET Framework Documentation on Dates](#)). Generally speaking, the date data type can be accessed using the *Date* class literal; this class exposes a number of properties and functions.

#### Current Local Date-Time

If you want to access the current, local date-time, you can do this by using the following syntax:

```
Date.Now
```

#### Current UTC Date-Time

If you want to access the current, UTC date-time, you can do this by using the following syntax:

```
Date.UtcNow
```

## Year, Month, Day, Hour and Minute

If you just want to use a specific property (e.g. such as the number of the month) of a date, you can use this using one of the following syntaxes:

```
<Date_Value>.Year
```

```
<Date_Value>.Month
```

```
<Date_Value>.Day
```

```
<Date_Value>.Hour
```

```
<Date_Value>.Minute
```

For example, if you want to access the current year (according to the local date-time), you can use the following syntax:

```
Date.Now.Year
```

## Calculations

Visual Basic .NET also offers functionality to perform calculations on dates.

### Adding or Subtracting Days

If you want to add 5 days to the current, local date-time, you can use the following syntax:

```
Date.Now.AddDays(5)
```

You can also supply negative values in order to subtract days.

### Adding or Subtracting Months

If you want to add 5 months to the current, local date-time, you can use the following syntax:

```
Date.Now.AddMonths(5)
```

You can also supply negative values in order to subtract months.

### Creating a new Date

You can create a new date (without time) by supplying the individual numeric components for year, month, and day using the following syntax:

```
New Date(2018,12,31)
```

The values supplied do not need to be constant values. You could, for instance, get the date which equals the first day of the next month, by using the following syntax:

```
New Date(Date.Now.AddMonths(1).Year,Date.Now.AddMonths(1).Month,1)
```

### Creating a new Date-Time

You can create a new date (including time) by supplying the individual numeric components for year, month, day, hour, minute and second using the following syntax:

```
New Date(2018,12,31,18,30,15)
```

The above expression equals the 31<sup>st</sup> of December 2018, 6:30 PM and 15 seconds.

## Formatting and Conversions

You can convert a date-time into a text representation by using the *ToString* function. This conversion process uses the culture settings of the user under which the expression is being executed or another culture provided by the context, e.g. culture-aware reports (→ [Culture Awareness](#)). If you use the following syntax, the date-time value will be converted to a textual representation using the default format of the applicable culture:

```
<Date_Value>.ToString()
```



For instance, if you want to convert the current, local date-time into a textual representation using the default format of the applicable culture, you can use the following syntax:

```
Date.Now.ToString()
```

However, the *ToString* function also allows you to supply a standard format string ([Standard Date Time Format Strings Documentation](#)) or a more flexible custom format string ([Custom Date Time Format Strings Documentation](#)). For instance, if you want to convert the current, local date-time into a textual representation that contains the 2-digit day, then a dot, then the 2-digit month, then a dot, and then the 4-digit year by using a custom format string, you can use the following syntax:

```
Date.Now.ToString("dd.MM.yyyy")
```

If you want to include time-information in the textual representation (2-digit hour in 24-hour format, then ":", then 2-digit minute), then you can use the following syntax:

```
Date.Now.ToString("dd.MM.yyyy HH:mm")
```

If you want to convert the current, local date into a textual representation according to the short date pattern of the current culture, you can use the following syntax:

```
Date.Now.ToString("d")
```

A special case of conversion is the conversion from an OLE Automation Date to an actual date-time. OLE Automation Dates are used as numeric representation of a date-time.

**Example:** In Microsoft Excel, for instance, dates are internally treated as numbers; if you enter a date into a cell and then set the cell format to "Number", then you can see the numeric representation of the date. QlikView also internally treats dates as numbers; therefore, when accessing a date value of a QlikView Document in Mail & Deploy, you will get a number instead of a date.

These OLE Automation Dates can be converted into actual date-times by using the following syntax:

```
ToDate(<ole_automation_date>)
```

## Built-In Functions

Expressions also support built-in functions.

### Does Report Document Exist

This function has the following syntax:

```
DoesReportDocumentExist(<report_document_name>)
```

The function returns *True* if a report document with the given name exists in the current context (e.g. if a report document has successfully been created using a [Create Report Document Action](#)) and *False* if that is not the case. Therefore, this function can only return *True* when executed in the context of a task (→ [Tasks](#)).

## Module

You can also write your own Visual Basic .NET code in a module. A module is a set of Visual Basic .NET code that contains functions to be used from any expression within the repository. This allows you to provide additional functionality to expressions. The following module, for instance, provides to functions; one that returns the current year and month and one that returns the current year, month, and day:


```
Function GetYearMonth()  
    Return Date.Now.Year.ToString("0000") & Date.Now.Month.ToString("00")  
End Function
```

```
Function GetYearMonthDay()
    Return Date.Now.ToString("yyyyMMdd")
End Function
```

Any of these functions can be called by using the following syntax in an expression:

```
GetYearMonth()
```

Functions contained in modules may also contain parameters.

There is exactly one module per repository. You can access and edit this module by clicking *Tools* → *Module* in the main window of Mail & Deploy Client or by clicking the  button in the expression editor.

## Examples

This is a collection of expressions which are commonly used.

### Report Document Existence

Sometimes it is necessary to check whether a report document with a certain name actually exists. Let's suppose there is a task with the following action hierarchy:

```
TASK
    CREATE REPORT DOCUMENT ExampleCustomerReport FROM REPORT CustomerReport
    GROUP
        SAVE REPORT DOCUMENT TO FILE SYSTEM
        DISTRIBUTE REPORT DOCUMENT BY E-MAIL
    END GROUP
```

The above task creates a report document called *ExampleCustomerReport* (→ [Create Report Document Action](#)) based on the report (→ [Reports](#)) called *CustomerReport*. Then – embedded in a group (→ [Group Action](#)) – it saves the created report document to the file system (→ [Save Report Document to File System Action](#)) and then distributes it by e-mail (→ [Distribute by E-Mail Action](#)).

Now let's suppose that the report *CustomerReport* has a condition (→ [General Attributes](#)), which means that the report document *ExampleCustomerReport* is only created if that condition is fulfilled. Obviously, the actions in the group, which save the report document to the file system and distribute the report document by e-mail should only be executed if the report document has actually been created (e.g. if the condition of the report was fulfilled). This can easily be achieved by setting a condition on the group action that tests for the existence of the report document; the condition for the expression needs to check whether the report document exists (→ [Does Report Document Exist](#)) and has the following syntax:

```
DoesReportDocumentExist("ExampleCustomerReport")
```





If the report document *ExampleCustomerReport* does not exist, the group condition will not be fulfilled and therefore the actions to save and distribute the report document will not be executed.

# Execution Calendar

## Introduction

Keeping an overview about all executed tasks, on demand-report requests etc. is one of the key tasks for administrators of a Mail & Deploy environment. The execution calendar is a feature of Mail & Deploy that allows you to visualize previous and future executions, quickly check if any of these executions failed etc.

There are three ways to reach the execution calendar:

- In Mail & Deploy Server Control Panel (→ [Control Panel](#)) you can go to the *Repositories* tab and then click the  button. The execution calendar opens and its scope is set to all existing repositories.
- In Mail & Deploy Client (→ [Client](#)) you can click *Tasks* in the main menu and then click the  button. The execution calendar opens and its scope is only set to the repository you are currently logged on to.
- In Mail & Deploy Client (→ [Client](#)) you can click *Tasks* in the main menu, select a task, click the  button to edit the task and then click the  button to open the execution calendar. The execution calendar opens and its scope is only set to the task that you are currently editing.

## Scope

Depending on how you open the execution calendar, it has different scopes; for instance, if you open the execution calendar from Mail & Deploy Server Control Panel (→ [Control Panel](#)) the scope of the execution calendar contains the data of all repositories. If you open the execution calendar from a single task in Mail & Deploy Client (→ [Client](#)) the scope of the execution calendar is limited to this very task. The execution calendar can only display data that is within its current scope.

## Filtering

The left part of the execution calendar contains a tree-view that contains the complete scope; this tree-view can be used to filter the scope so that the items which are being displayed in the month view and the day view are limited to exactly the information you want to get. This filter tree-view contains three major categories.

## Repositories

The first top-level entries on that list are the repositories which are contained in the scope (in some cases the scope is limited to one repository only). You can select or de-select repositories by using the checkbox next to the repository's name. If you uncheck a repository the month view and the day view will not contain any executions related to that repository. Below each repository item, you can see the following folders:

**Tasks.** In this folder, all tasks (→ [Tasks](#)) of the repository are being listed. You can select or de-select tasks by using the checkbox next to the task's name. If you uncheck a task, its executions will not be displayed in the month view and day view.

**Reports.** In this folder, all reports (→ [Reports](#)) of the repository are being listed. You can select or de-select reports by using the checkbox next to the report's name. If you uncheck a report, executions of on-demand requests of that report will not be displayed in the month view and day view.

**Users.** In this folder, all users (→ [Users](#)) of the repository are being listed. You can select or de-select users by using the checkbox next to the user's name. If you uncheck a report, executions of on-demand requests of that user will not be displayed in the month view and day view.

## Execution Types

This top-level entry contains all execution types to which you can filter the month view and day view. You can select or de-select execution types by using the checkbox next to the execution type's name. If you uncheck an execution type, executions of that type will not be displayed in the month view and day view. There are the following execution types:

**Regular.** Regular executions are executions of tasks (→ [Tasks](#)) which are automatically executed by Mail & Deploy Server according to the execution plans (→ [Execution Plans](#)) you have configured for the task.

**Report Document Request.** This type of execution covers all the requests of report documents submitted through Mail & Deploy Hub (→ [Hub](#)) and through the API (→ [API](#)).

**Report Document Subscription.** This type of execution covers all executions related to subscriptions to report documents submitted through Mail & Deploy Hub (→ [Subscriptions](#)).

## States

This top-level entry contains all execution states to which you can filter the month view and day view. You can select or de-select any execution states by using the checkbox next to the execution state's name. If you uncheck an execution state, executions with that state will not be displayed in the month view and day view. There are the following execution states:

**Future.** Future executions are executions which have not yet been processed, but are scheduled for execution.

**Successful.** This execution state specifies successful executions.

**Cancelled.** This execution state specifies cancelled executions.

**Failed.** This execution state specifies failed executions.

## Month View

The center part of the execution calendar contains a month view in which individual days according to the following rules:

- If there is no execution that applies to the current filter, the day is rendered in gray.
- If there is at least one failed execution that applies to the current filter, the day will be rendered in red.
- If there is at least one cancelled execution that applies to the current filter, the day will be rendered in yellow.
- If there is at least one green execution that applies to the current filter, the day will be rendered in green.







If there are different execution results within a given day, the worst execution result dictates the color.

**Example:** If on a given day there were 3 successful executions, 2 cancelled executions and 4 failed executions, the day will be rendered in red, because there is at least one failed execution on that day.

You can click on a day in the month view to activate that day in the day view (→ [Day View](#)).

## Day View

The right part of the execution calendar contains a day view in which individual executions will be displayed for the day currently selected in the month view (→ [Month View](#)). The view can be controlled using the following buttons:


	Click this button to set the resolution of the day view to 1 minute.
	Click this button to set the resolution of the day view to 5 minutes.
	Click this button to set the resolution of the day view to 15 minutes.
	Click this button to set the resolution of the day view to 30 minutes.
	Click this button to set the resolution of the day view to 60 minutes.
	Click this button to toggle the automatic expansion functionality (→ <a href="#">Item Height</a> ).

You can double-click any of the execution items to see the execution log and other details.

## Item Height

Generally every execution item in the day view has a height that corresponds exactly to the duration of the tasks. However, depending on the resolution of the day view, this might lead to situations where an execution item would have a very small height (or even be invisible).

**Example:** If you set the resolution of the day view to 60 minute, you will have a rather coarse view of all executions on that day. Executions which had a duration of only a few seconds cannot be rendered with their height corresponding to their actual duration because that would lead them to be invisible.

Therefore there is an automatic expansion functionality that can be enabled/disabled using the  button at the top of the day view. If enabled, execution items which would be too small to be displayed in the current resolution of the day view will be resized so they are still visible. This, of course, means that these items are higher than their actual execution duration.

# File Type Settings

This section contains information about file type specific settings that can be configured when working with that file type; for instance if you create a report document with that file type (→ [Create Report Document Action](#)).

## Adobe PDF Document

Both Adobe PDF Document and Adobe PDF/A Document support the following settings:

<b>Has User Password</b>	Determines whether the created PDF document is password-protected if users want to view its content. If you tick this option, an input field that allows you to enter the password will be displayed.
<b>Has Owner Password</b>	Determines whether the created PDF document is password-protected if users want to edit its content. If you tick this option, an input field that allows you to enter the password will be displayed.

# Table Cell Types

## Introduction

When working with tables (→ [Table Elements](#)), Mail & Deploy can treat different cell types differently by setting individual column specifications (→ [Column Specifications](#)). This section explains what these different cell types mean for different datasources (→ [Datasources](#)) and types of source objects (→ [Objects](#)).

## Examples

### Straight Tables

This section explains all cell types you can find in straight tables.

Product	Value 2017	Value 2018
Totals	23.000,00 USD	21.298,88 USD
Product 1	12.000,00 USD	10.298,88 USD
Product 2	11.000,00 USD	11.000,00 USD

The following legend explains which cell type is represented by each color:

	Top Header Cells
	Total Cells
	Data Cells

### Pivot Tables

This section explains all cell types you can find in pivot tables.

Product Group	Product	2017	2018
Group A	Totals	23.000,00 USD	21.298,88 USD
	Product 1	12.000,00 USD	10.298,88 USD
	Product 2	11.000,00 USD	11.000,00 USD

The following legend explains which cell type is represented by each color:

	Super Cell
	Top Header Cells
	Left Header Cells
	Total Cells
	Please note, that pivot tables based on a QlikView Document datasource (→ <a href="#">QlikView Document</a> ) do not have total cells; in this case, total cells are treated as data cells.
	Data Cells

# Styles

## Introduction

Generally speaking, a style in Mail & Deploy is a collection of properties and settings that dictate how an item (e.g. text, a table cell etc.) is rendered in a report document.

Each style value has four different application modes.

**None.** If a style value has this application mode, the associated property will not be set in the report document and whatever is the default for the report type (→ [Report Types](#)) will be kept.

**Keep Original.** If a style value has this application mode, the associated property will be set to the value that has originally set; e.g. when a table report element (→ [Table Elements](#)) is read from a datasource, the original value of that style in the datasource will be kept.

**Constant Value.** If a style value has this application mode, the associated property will be set to a constant value.

**Expression.** If a style value has this application mode, the associated property will be set to a value evaluated from an expression (→ [Expressions](#)).

## General Attributes

Each style has the following general attributes:

<b>Name</b>	The name of the style; this is only used within Mail & Deploy to identify the style.
-------------	--

## Style Types

### Table Cell Styles

This type of style contains properties and settings to format the cell of a table and its contents. It has – in addition to the general attributes (→ [General Attributes](#)) that every style has - the following style attributes:

### Text Style Attributes

<b>Font Name</b>	The name of the font to use to render the text.
<b>Font Size</b>	The size of the font to use to render the text.
<b>Font Is Bold</b>	Determines whether the text shall be rendered bold.
<b>Font Is Italic</b>	Determines whether the text shall be rendered italic.
<b>Font Is Underlined</b>	Determines whether the text shall be rendered underline.
<b>Text Color</b>	The color in which the text shall be rendered.
<b>Value Format</b>	The format string that determines in which format the text shall be rendered. This can be any standard format string (🌐 <a href="#">Standard Date Time Format Strings Documentation</a> ) or a more flexible custom format string (🌐 <a href="#">Custom Date Time Format Strings Documentation</a> ).



<b>Wrap Text</b>	Determines whether the text may be wrapped within the table cell.
<b>Horizontal Alignment</b>	Determines how the text shall be horizontally aligned within the table cell.
<b>Vertical Alignment</b>	Determines how the text shall be vertically aligned within the table cell.

## Background Style Attributes

<b>Background Pattern</b>	The pattern in which to render the background.
<b>Background Color</b>	The color in which to render the background of the table cell.

## Border Style Attributes

For each border side (top, bottom, left, right) the following attributes are available:

<b>Border Line Type</b>	The line type in which the border shall be rendered.
<b>Border Width</b>	The width with which the border shall be rendered.
<b>Border Color</b>	The color in which the border shall be rendered.

## Padding Style Attributes

For each side (top, bottom, left, right) the following attributes are available:

<b>Padding</b>	The padding of the text within the cell.
----------------	--

# Log Files

Mail & Deploy provides two different types of log file with which you can monitor the components of Mail & Deploy.

## Application Logs

These logs contain information about the status of a component of Mail & Deploy; they do not contain information about the execution of tasks or other requests; instead they can be used to identify problems with the components of Mail & Deploy on an application level.

These logs can be found in the *Logs* sub-folder of your Mail & Deploy Server (→ [Server](#)) installation directory, where you will find a sub-folder for each component of Mail & Deploy. The log files are then organized in daily text files.

## Task Execution Logs

To get information about executed tasks, the execution calendar (→ [Execution Calendar](#)) can be used. Task execution logs have log entries in their sequential order which can be of one of the following severities:



The log entry represents an information.



The log entry represents a warning.



The log entry represents an exception.

You can move your mouse over a log entry and wait for a short period of time in order to see a tooltip that contains the complete information about this log entry (including any technical information about an exception). By selecting a log entry and pressing STRG + C you can copy the complete log entry to the clipboard.

# Development Environment

## Introduction

Mail & Deploy supports the implementation of a development environment that consists of a test system and a production system.

The test system is an installation of Mail & Deploy Server on a server that may be used to test new reports (→ [Reports](#)) and tasks (→ [Tasks](#)). The production system is an installation of Mail & Deploy Server on a server that is different from the one of the test system and contains the actual production environment.

For each repository, there is one copy in the test system and one copy in the production system. New reports and tasks or changes to reports and tasks can be first done and tested on the test system and then be published (→ [Publishing](#)) to the production system, if all tests succeed.

## Configuration


In order to configure the development environment, you need to first install Mail & Server in the test and in the production system, so that you have to instances of Mail & Deploy Server running. For each repository for which you want to use the publishing features, you need to create one copy in the test system and one copy in the production system by using the respective instances of Mail & Deploy Server Control Panel (→ [Control Panel](#)). You don't necessarily need to create new repositories when setting up your development environment; existing repositories can be used as well.

You have to connect the repository of the test system with the repository of the production system by opening Mail & Deploy Client with the test repository and specifying the connection settings of the production repository (→ [Production Repository](#)).

## Publishing

This section describes how you can publish reports and tasks from the test system to the production system.


### Reports

In order to publish a report from the test system to the production system, you need to open the list of reports in Mail & Deploy Client by clicking *Reports* in the left menu, selecting the report you want to publish and then clicking the  button. This button can only be clicked, if the repository you are currently connected to has a production repository (→ [Production Repository](#)) set up. In order to successfully publish a report, the following conditions must be met:

- All datasources (→ [Datasources](#)) which are used by the report to be published need to exist in the production repository. Datasources in the production repository need to have the same name as the ones of the test repository.

If any of these conditions are not met, you will receive a message indicating which element Mail & Deploy requires you to create in the production repository in order to publish the report.

### Tasks

In order to publish a task from the test system to the production system, you need to open the list of tasks in Mail & Deploy Client by clicking *Tasks* in the left menu, selecting the task you want to publish and then clicking the  button. This button can only be clicked, if the repository you are currently connected to has a production repository (→ [Production Repository](#)) set up. In order to successfully publish a task, the following conditions must be met:

- All users and user groups (→ [Users and User Groups](#)) which are used by the task need to exist in the production repository. Users and user groups in the production repository need to have the same name as the ones of the test repository.
- All datasources (→ [Datasources](#)) which are used by the task need to exist in the production repository. Datasources in the production repository need to have the same name as the ones of the test repository.
- All reports (→ [Reports](#)) which are required by the task need to exist in the production repository. Reports in the production repository need to have the same name as the ones of the test repository.
- All e-mail servers (→ [E-Mail Servers](#)) which are required by the task need to exist in the production repository. E-Mail Servers in the production repository need to have the same name as the ones of the test repository.
- If the task requires an active directory connection (→ [Active Directory Connection Settings](#)) then the production repository needs to have a working active directory connection set up.

If any of these conditions are not met, you will receive a message indicating which element Mail & Deploy requires you to create in the production repository in order to publish the task.



Please note that execution plans (→ [Execution Plans](#)) are not published to the production repository when publishing a task; if you publish a task that already exists in the production environment – so that the task gets updated – existing execution plans of that task will be kept.

# Tools

This section describes tools that are being shipped with Mail & Deploy.

## API Shell

### Introduction

Mail & Deploy API Shell is a commandline tool that allows you to communicate with Mail & Deploy API (→ [API](#)) through the commandline (e.g. by using a batch file).

### Requirements

Mail & Deploy API Shell requires the .NET Framework 4.7 to be installed.

### Installation

If you want to use Mail & Deploy API Shell from another computer, you can simply copy the complete API Shell directory to that computer; no further installation is required. However, the requirements (→ [Requirements](#)) need to be met on that computer.

### Parameters

The general call to the API shell looks like this:

```
"Mail & Deploy API Shell.exe" RequestFilePath={requestfilepath}
    [Silent=True/False]
    [OutputFilePath={outputfilepath}]
    [Username={username}]
    [Password={password}]
```

The names of parameters are case sensitive.

### Request File Path

This parameter is required and contains the path to an XML file that contains details about the request; the format of the XML file is as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<MailAndDeployApiRequest Host="{host}"
    Port="{port}">
    <Parameter Name="{parameter_name}" Value="{parameter_value}" />
    ...
</MailAndDeployApiRequest>
```

The meanings of these values are:

<b>Host</b>	The host that runs Mail & Deploy Server and thus Mail & Deploy API.
<b>Port</b>	The port on which Mail & Deploy API runs; this port can be configured using Mail & Deploy Server Control Panel (→ <a href="#">Control Panel</a> ).
<b>Parameters</b>	Zero, one, or more parameters for the API request, each consisting of a name and a value. These parameters correspond to the parameters required to perform a call to Mail & Deploy API (→ <a href="#">API</a> ).

## Silent

This parameter is optional and defaults to *False*. If this parameter is set to *True*, no user output is performed by the tool and no user input is allowed. If no credentials (→ [Username](#) and → [Password](#)) are provided, then a call with silent being set to *True* will fail, because Mail & Deploy API Shell would prompt the user for credentials and cannot do so because of the silent mode being active.

## Output File Path

This parameter is optional; by default there is no output file. If you supply this parameter, Mail & Deploy API Shell will write its output to the given file path.

## Username

The username that is used to authenticate against Mail & Deploy API (→ [API](#)). This needs to be the credential username of a user (→ [Users](#)) with at least a role of *API Consumer*.

## Password

The password of the user used for authentication (→ [Username](#)).

## Execution

When executed, Mail & Deploy API Shell expects at least one parameter (→ [Request File Path](#)). The request file will be read and you might be required to enter credentials if you haven't supplied them with the appropriate parameters (→ [Username](#) and → [Password](#)); if the tool prompts for credentials, you need to enter the credentials of a user (→ [Users](#)) with at least a role of *API Consumer*.

# Repository Converter

## Introduction

Repository Converter is a tool shipped with Mail & Deploy that allows you to convert repository from Mail & Deploy Version 2.0.7 to a repository of the installed version of Mail & Deploy. The tool is located in the *Tools\Repository Converter* subfolder of your Mail & Deploy Server installation and can be started by double-clicking the file *Mail & Deploy Repository Converter.exe*.



It is highly recommended that you check the converted repository before actually using it in a production environment; use at your own risk.

## Operation

Once started, you can start the conversion of a repository by clicking *Convert Repository...*; you can then browse the repository to convert. As soon as you have selected the file, the conversion process will begin and you will be able to see its progress in the log output. Any error message that occurs will be displayed in this log output. If the conversion completes successfully you will be prompted to save the new repository; once saved, the repository can be imported into the new installation of Mail & Deploy (→ [Repositories](#)).



Please note, that you must not manually save the new repository into the *Data\Repositories* directory of your Mail & Deploy Server installation. This may lead to errors and other unexpected results; it is always necessary to import repositories (→ [Repositories](#)) instead of copying the file manually.

## Limitations

Although Mail & Deploy Repository Converter is capable of converting a large part of your old repositories, it cannot cleanly convert all data. This is mainly due to breaking changes in comparison to older versions of Mail & Deploy. The following list provides an overview of all limitations:

- The datasource cache of each datasource (→ [Datasources](#)) needs to be refreshed before anything else can be done within that repository.
- CSV reports have significantly changed; in older versions, it was possible to specify which cell types shall be included in the output; when converting CSV reports to the new versions, they will be imported in such a way that all cell types are included in the output.
- When converting reports that contain table report elements based on pivot-tables, you need to manually go through these elements (→ [Table Elements](#)) and adjust their pivot settings (→ [Pivot Settings](#)), because they cannot be converted.
- In previous versions of Mail & Deploy the direction of the output table of table report elements has automatically been adjusted to be equal to the direction of the datasource table on which the report element was based; if you use such a table report element, you need to manually adjust the table direction (→ [Table Elements](#)) in the new repository.
- If you used an execution calendar on a task, you need to add the calendar (→ [Calendars](#)) and assign it to the task execution plan (→ [Execution Plans](#)) manually after the conversion.

## Qlik Sense Extension

### Introduction

The Mail & Deploy Qlik Sense Extension can be used to offer users the capability to request reports from Mail & Deploy directly from the Qlik Sense user interface (i.e. a Qlik Sense Sheet). A zip file that contains the extension is located in the *Tools\Qlik Sense Extension* subfolder of your Mail & Deploy Server installation.

### Installation

The installation of the extension follows the standard installation steps of extensions in Qlik Sense ([🌐 Qlik Sense Custom Components Installation](#)).

### Configuration

You can use the extension by adding the custom component *Mail & Deploy* to any Qlik Sense Sheet. The following properties can be set:

<b>Host / IP-Address</b>	The host name or IP address of the server that runs Mail & Deploy Server.
<b>Port</b>	The port of Mail & Deploy API (→ <a href="#">API</a> ).
<b>Repository</b>	The name of the repository that contains the report that can be requested.
<b>Username</b>	The username of the user (→ <a href="#">Users</a> ) that is used to authenticate with Mail & Deploy Server. The user needs to have at least a role of <i>API Consumer</i> in the repository that contains the report that can be requested.
<b>Password</b>	The password of the user that is used for authentication.
<b>Datasource Name</b>	The name of the datasource (→ <a href="#">Datasources</a> ) that connects to the Qlik Sense app in which the extension is used.
<b>Report Name</b>	The name of the report (→ <a href="#">Reports</a> ) that can be requested with the extension.
<b>File Type</b>	The file type in which the report document shall be created. The range of available file types depends on the report type (→ <a href="#">Report Types</a> ). The following values can be provided:

	<i>AdobePdfDocument</i> . This file type represents an Adobe PDF Document.
	<i>AdobePdfADocument</i> . This file type represents an Adobe PDF/A compliant document.
	<i>HtmlDocument</i> . This file type represents an HTML document.
	<i>MicrosoftExcelWorkbook</i> . This file type represents a Microsoft Excel Workbook.
	<i>MicrosoftPowerPointPresentation</i> . This file type represents a Microsoft PowerPoint Presentation.
	<i>MicrosoftWordDocument</i> . This file type represents a Microsoft Word Document.
<b>Download Name</b>	The name of the download (excluding the file type extension) that will be created by the extension.
<b>Apply Current Selections</b>	Determines whether the current selections of the Qlik Sense app shall be sent to Mail & Deploy (and therefore affect the data that will be contained in the created report document).
<b>Create Report Text</b>	The text that the button of the extension shall display to indicate that clicking the button will create the report document.
<b>Please Wait Text</b>	The text that the button of the extension shall display to indicate that the report document is currently being created and the user needs to wait until that process has completed.


## Usage

The extension consists of a button that can be clicked by the user to request the report document from Mail & Deploy server. Depending on the configuration of the extension (→ [Configuration](#)), the current selections in the Qlik Sense app will be transferred to Mail & Deploy Server to influence the data contained in the report document.

Once the user has clicked the button, the request will be processed by Mail & Deploy in the background and the button will be disabled. If an error occurs, a message box will be displayed to the user; if the report document has been created successfully, it will automatically be downloaded by the web browser, using the download name provided in the configuration of the extension.



# Version History

This section lists all the changes that have been introduced in new releases of Mail & Deploy. If a change is marked with , the change is a breaking change, which means that manual intervention after updating an older version of Mail & Deploy may be required. You should then perform the update at a time where no tasks will be executed in order to have time to perform the required manual changes after the update.

## Version 2.1.1


### Build #15

With this version, the following changes have been introduced:

MDC00228	Bugfix	Fixed a bug that lead to log files of tasks in the execution calendar not being saved/displayed correctly under certain circumstances.
----------	--------	--

### Build #14

With this version, the following changes have been introduced:

MDC00224	Bugfix	Fixed a bug that lead to report elements below a table in a Microsoft Excel report not to be replaced if the table used <i>Copy Rows</i> .
MDC00225	Bugfix	Fixed a bug that lead to exceptions when rebuilding the cache (especially with older versions of Qlik Sense).
MDC00226	Change	The performance of the execution calendar and the hub have been improved.
MDC00227	 Change	In order to improve performance and repository size, the lifetime of hub report documents (report documents available to users in Mail & Deploy hub) has been reduced to 3 months. After 3 months, report documents will be deleted from the hub. If you start Mail & Deploy after applying this update, all report documents older than 3 months will be deleted from the hub.

### Build #13

With this version, the following changes have been introduced:

MDC00222	Bugfix	Fixed a bug that lead to certain Qlik Sense text colors not being read correctly from pivot tables.
MDC00223	New Feature	You can now not only upload individual report documents but also zipped file collections to FTP using the <i>Upload Report Document To FTP</i> action.

### Build #12

With this version, the following changes have been introduced:

MDC00214	New Feature	Repeater report element levels can now be used to repeat over the distinct values of an expression.
MDC00215	New Feature	Log files when being viewed in the Execution Calendar can now be exported to a text file. This is particularly useful if you want to send a log file to our support team for investigation.
MDC00216	Change	The performance of Mail & Deploy Client when navigating through the main menus (Datasources, Reports, and Tasks) has been improved.
MDC00217	New Feature	When exporting a raw table report element to native PowerPoint MSO charts, additional settings (such as the axis title and display unit) may be set.
MDC00218	New Feature	Mail & Deploy now supports the new Org and Mekko charts of Qlik Sense.
MDC00219	New Feature	You can now also reach the execution calendar from the <i>Server Logs</i> main menu of Mail & Deploy Client (previously that button was only available in the <i>Tasks</i> main menu).
MDC00220	Bugfix	A bug that lead to a memory leak leading to Mail & Deploy Server consuming more and more memory over time when exporting data from Qlik Sense apps has been fixed.
MDC00221	New Feature	Qlik Sense datasources now have additional settings to set a filter before rebuilding the datasource cache. This target the problem that if a Qlik Sense app has a complex data model and/or a lot of sheets and visualizations, rebuilding the cache may time-out or take an exceedingly long time. By setting an initial filter before rebuilding the cache that time-out can be prevented, and cache rebuilding is much faster.

## Build #11

With this version, the following changes have been introduced:

MDC00210	New Feature	Repeater report element levels can now be used to repeat over the distinct values of the column of a table from a datasource object.
MDC00211	New Feature	All action lists (preparation actions and tasks) now support to select an action and copy it to the clipboard using STRG + V. The copied action can then be pasted into an empty action list by clicking the empty action list with the left mouse button to focus is and pressing STRG + V, or by selecting an action in an action list and pressing STRG + V to insert the copied action after the selected one.
MDC00212	Bugfix	Fixed a bug that lead to exceptions when exporting completely empty Qlik Sense pivot tables.
MDC00213	New Feature	The server will now automatically upgrade all repositories to the required version upon startup. If you have to copy repositories from previous Mail & Deploy 2.1 versions you can now just stop Mail & Deploy Server, copy the repository and then start Mail & Deploy

Server. It's important to note that only upgrades (not downgrades) are possible.

## Build #10

With this version, the following changes have been introduced:

MDC00208	New Feature	Table report elements may now be used to fill the data of native PowerPoint charts by placing the reference to the report element into the alternative text of the chart. Raw table elements supports to specify additional settings for that chart (such as series and axis colors, minimum and maximum values etc.).
MDC00209	New Feature	Repeater report elements – when used in a Microsoft PowerPoint or Microsoft Excel report – now have a new attribute <i>Number Of Repeated Items</i> with which you can control how many slides/worksheets shall be repeated per repeated value.

## Build #09

With this version, the following changes have been introduced:

MDC00196	New Feature	When inserting tables into Microsoft Word Documents, you can now specify whether the table shall be written into an existing table (rather than creating a new one); if you do that, you can set column widths, formatting etc. directly in word rather than having to write (sometimes complex) expressions.
MDC00197	Bugfix	Fixed a bug that lead to scheduled tasks not being executed when certain date/time-settings in the operating systems are set.
MDC00198	Change	Templates of reports are now sorted alphabetically rather than in the order in which they have been created.
MDC00199	Bugfix	Fixed a bug that lead to Mail & Deploy crashing in some situations when a task accessing a QlikView Document datasource is aborted.
MDC00200	Bugfix	Fixed a bug that lead to range references in Excel not being updated correctly when inserting/deleting rows if the range reference is on another sheet (e.g. a chart on sheet 1 referencing data in sheet 2).
MDC00201	Bugfix	Fixed a bug that lead to <i>Invalid Window Handle</i> errors when reloading a QlikView document that uses ODBC connections.
MDC00202	Change	Fixed a problem where when a QlikView document is launched and the document cannot be accessed (e.g. due to missing Section Access), an error message will appear in the log rather than the task hanging forever.
MDC00203	Bugfix	Fixed a bug that lead to <i>Invert Filter</i> not working when being used in conjunction with <i>Perform Match Filter</i> .
MDC00204	New Feature	Objects in nested containers in QlikView Documents are now supported by Mail & Deploy.

MDC00205	Bugfix	A bug that lead to QlikView tables not being exported correctly as images has been fixed.
MDC00206	Bugfix	A bug that lead to failed Qlik Sense app reloads not being recognized correctly by Mail & Deploy (thus leading to subsequent errors) has been fixed.
MDC00207	Bugfix	The button Copy Column Specification when editing table report elements does now work as intended.

## Build #08

With this version, the following changes have been introduced:

MDC00186	New Feature	When inserting tables into Microsoft Word Documents, you can now specify whether empty paragraphs after that table (which may cause an unwanted page break in some circumstances) shall be removed.
MDC00187	New Feature	In template based reports, if you have multiple templates, you can not only specify one report template to be used (in attribute <i>Report Document Name</i> ) but also a list of templates which will then be merged to a single report template. This allows you to define individual modules as report templates and then merge them together on-the-fly.
MDC00188	Change	When using <i>Perform Match Filter</i> and a <i>Check Type</i> in a filter action, the filtering will fail if no value can be selected (because no value in the datasource matches the search string).
MDC00189	Bugfix	A bug that lead to fields not being correctly filter in some situations when using a repeater report element has been fixed.
MDC00190	New Feature	An encoding can now be specified for CSV reports that controls the encoding of the output file.
MDC00191	New Feature	Hidden fields can now be used when working with Qlik Sense datasources.
MDC00192	Bugfix	Fixed a bug where under certain circumstances (especially when using nested cycles in tasks), expression editors would not display all cycle values to be inserted into an expression.
MDC00193	New Feature	There are now more options to specify the behavior to be applied to report elements whose conditions are not fulfilled. In previous versions, it was only possible to skip the report element or to cancel the report document creation. New options include the deletion of the row or worksheet that contains the report element (for Microsoft Excel Reports) or to delete the slide that contains the report element (for Microsoft PowerPoint Reports).
MDC00194	New Feature	A new action <i>Upload Report Document To FTP</i> allows you to upload reports documents to an FTP server using FTP or SFTP.
MDC00195	New Feature	A port can now specified for Qlik Sense Server connections; this allows to connect to Qlik Sense Servers running on non-standard ports.


## Build #07

With this version, the following changes have been introduced:

MDC00181	New Feature	A new insertion mode Copy Rows has been introduced for table elements which are being exported to Microsoft Excel Reports. This allow complete template rows to be copied for each row of the inserted table so that you may use Excel formulas in cells right of the inserted table.
MDC00182	New Feature	A new option <i>Send To E-Mail Addresses</i> has been added to <i>Distribute by E-Mail</i> actions which allows you to provide an expression that returns one or more e-mail addresses; this enables you to send to dynamic e-mail addresses without dynamically assigning e-mail addresses or users to report documents beforehand.
MDC00183	Bugfix	A bug that lead to <i>A repository with the given name already exists</i> exception messages when using the API method <i>CloneRepository</i> – even though the message is factually wrong – has been fixed.
MDC00184	Bugfix	A bug that lead to Microsoft PowerPoint templates not being saved correctly under some circumstances has been fixed.
MDC00185	New Feature	A new action <i>Remove Datasource Field Filter Restriction</i> has been added which may be used to remove filter restrictions (such as “Always one selected value”) from a datasource field.

## Build #06

With this version, the following changes have been introduced:

MDC00173	Change	Exporting Qlik Sense charts now has improved performance and requires less hardware resources.
MDC00174	Bugfix	A bug that lead to parameter values of <i>Create Report Document</i> actions not being displayed correctly has been fixed.
MDC00175	 Bugfix	<p>A bug that lead to columns orders not being exported correctly when using report elements based on Qlik Sense Straight Tables has been fixed.</p> <p><b>Important:</b> If you upgrade from Version 2.1.1 Build #01 or higher this bugfix may lead to incorrect results and/or broken reports. If your reports rely on the “wrong” column order that has been exported in versions 2.1.1 Build #01 to 2.1.1 Build #05 of Mail &amp; Deploy, the new “correct” column order might break functionality in your reports. If you’re in doubt, please contact our support team to assist you in determining whether this update requires manual intervention.</p>
MDC00176	Bugfix	A bug that lead to <i>Error when determining measure location</i> exception messages and layout problems when exporting Qlik Sense tables has been fixed.
MDC00177	Bugfix	A bug that lead to <i>Object reference not set to an instance of an object</i> exceptions when executing <i>Click Datasource Button</i> Actiona has been fixed.

MDC00178	Change	When editing a report element, the list of report elements now remembers the previous selection and vertical scroll position.
MDC00179	New Feature	Column width expressions of column specifications may now access the column count ( <i>ColumnCount</i> ) and the table data matrix ( <i>TableData</i> ).
MDC00180	New Feature	Style property expressions may now access the row count ( <i>RowCount</i> ) and column count ( <i>ColumnCount</i> ).

## Build #05

With this version, the following changes have been introduced:

MDC00167	Bugfix	A bug that lead to table exports of certain QlikView objects failing has been fixed.
MDC00168	Bugfix	A bug that lead to HTML report documents not being accessible from Mail & Deploy Hub has been fixed.
MDC00169	Bugfix	A bug that lead to QlikView object images not being exported correctly when they have a shadow set, has been fixed.
MDC00170	New Feature	Text objects and current selection boxes can now be exported from QlikView datasources.
MDC00171	Bugfix	A bug that sometimes lead to Index Out Of Range-Exceptions when exporting Qlik Sense Tables has been fixed.
MDC00172	Bugfix	A bug that sometimes lead to number formats not being adopted correctly from Qlik Sense has been fixed.

## Build #04

With this version, the following changes have been introduced:

MDC00158	Bugfix	A bug that lead to padding and font settings not being applied correctly in Microsoft PowerPoint reports has been fixed.
MDC00159	Bugfix	A bug that lead to line heights not being applied correctly when exporting tables from QlikView documents has been fixed.
MDC00160	Bugfix	A bug that lead to date values not correctly interpreted when reading data from Qlik Sense straight tables has been fixed.
MDC00161	Bugfix	A bug that lead to table exports failing for some versions of QlikView has been fixed.
MDC00162	Bugfix	A bug that lead to certain cells having wrong values when exporting a QlikView pivot table has been fixed.

MDC00163	Bugfix	A bug that lead to the API Shell Tool not being able to authenticate against Mail & Deploy API Server has been fixed.
MDC00164	Bugfix	A bug that lead to empty rows occurring in some situations when exporting Qlik Sense pivot tables, has been fixed.
MDC00165	Bugfix	A bug that lead to arithmetic operation exceptions when exporting Qlik Sense Tables has been fixed.
MDC00166	Change	Session handling with Qlik Sense has been changed to avoid error messages indicating no access to Qlik Sense.


## Build #03

With this version, the following changes have been introduced:

MDC00149	Bugfix	A bug that lead to exceptions when creating a new execution plan has been fixed.
MDC00150	Bugfix	A bug that lead to German translations not being displayed correctly on some pages of Mail & Deploy Hub has been fixed.
MDC00151	Bugfix	A bug that lead to seldom and random KeyNotFound exceptions which crashed Mail & Deploy Server has been fixed.
MDC00152	Bugfix	A bug that lead to filter checks failing under some circumstances has been fixed.
MDC00153	Bugfix	A bug that lead to incorrectly formatted pivot tables when used in Microsoft PowerPoint reports, has been fixed.
MDC00154	Bugfix	A bug that lead to automatic column widths not being applied correctly to tables in Microsoft PowerPoint reports has been fixed.
MDC00155	New Feature	Styles can now be copied.
MDC00156	Bugfix	A bug that lead to horizontal alignments of cells of tables read from QlikView Documents not being read correctly has been fixed.

## Build #02


With this version, the following changes have been introduced:

MDC00139	 Change	When specifying settings for the connection a repository to its production repository, credentials to connect to the production repository have to be provided.
MDC00140	Bugfix	A bug that lead to exceptions when creating opening the datasource dialog and cancelling it without changes, has been fixed.
MDC00141	Bugfix	A bug that lead to report documents in HTML format not working correctly when requesting them with the Qlik Sense Extension, has been fixed.

MDC00142	Bugfix	A bug that lead to exceptions when trying to create <i>Initialize Datasource Object Actions</i> has been fixed.
MDC00143	New Feature	There is now an <i>Invert Filter</i> option in <i>Filter Datasource Field Actions</i> which works like a NOT operator.
MDC00144	New Feature	There is a new action <i>Activate Datasource Theme</i> that allows to specify the Qlik Sense theme to use when extracting visualizations. Please consult the manual to find out how to use this feature.
MDC00145	Change	The compression method of PDF files created from Microsoft Excel, Microsoft Word, and Microsoft PowerPoint reports has been changed to significantly reduce the size of resulting PDF files.
MDC00146	Bugfix	A bug that lead to all cells of tables extracted from QlikView Documents to be underlined, has been fixed.
MDC00147	Bugfix	A bug that lead to the italic setting of datatable cell styles not being upgraded correctly when installing a new Mail & Deploy update has been fixed.
MDC00148	Bugfix	A bug that lead to the execution engine crashing in environments with 32+ CPU cores under certain circumstances, has been fixed.

## Build #01

With this version, the following changes have been introduced:

MDC00076	New Feature	It's now possible to split a single task in multiple jobs and increase performance by executing it in multiple threads.
MDC00102	Change	The performance when viewing the server status in Mail & Deploy Client has been improved.
MDC00110	Change	The execution status page of Mail & Deploy Hub has been changed to account for parallelized task executions.
MDC00115	Change	In expressions used in data table cell styles, there is a new expression parameter <i>TableData</i> that contains the complete table as a two-dimensional matrix. To access an individual cell value of that matrix, the syntax <i>TableData(RowNumber)(ColumnNumber)</i> may be used.
MDC00116	New Feature	When using <i>Keep Original</i> in data table cell styles, text and background colors of Qlik Sense tables will be transferred to the report document.
MDC00118	Change	Performance has been increased when data table cell styles are used.
MDC00119	 Change	Performance has been increased when working with large repositories. After applying the update, you need to manually refresh the cache of all datasources.



MDC00120	New Feature	When using Keep Original in data table cell styles, table wrap settings used in Qlik Sense tables are now interpreted and correctly transferred to the report document.
MDC00124	Bugfix	A bug, that lead to an exception in certain cases when working with large tables in Microsoft Excel reports has been fixed.
MDC00128	New Feature	Repeaters in Microsoft Excel reports can now not only be used to repeat complete worksheets but also to repeat certain rows within a worksheet.
MDC00129	New Feature	When configuring a monthly execution plan, you can optionally choose to stick the execution to the end of each month.
MDC00130	Bugfix	A bug that lead to Qlik Sense datasources not being populated correctly when creating a report based on a Qlik Sense Story, has been fixed.
MDC00131	Bugfix	A bug that lead to incorrect data in pivot tables exported from QlikView Documents with indent rows has been fixed.
MDC00132	New Feature	A new action <i>Run Application</i> has been introduced which allows you to run an application or batch file from a Mail & Deploy task.
MDC00133	Bugfix	A bug that lead to slides of Qlik Sense stories not being rendered correctly, has been fixed.
MDC00134	Bugfix	A bug that lead to buttons to delete and copy active directory user group synchronization specifications not working correctly, has been fixed.
MDC00135	Bugfix	A bug that lead to exceptions when trying to import user custom properties from a CSV file, has been fixed.
MDC00136	Bugfix	A bug that lead to the distribution of report documents to hub report document libraries failing when no description is given, has been fixed.
MDC00137	Bugfix	A bug that lead to the dynamic assignment of users to report documents based on custom properties of users not assigning all users correctly, has been fixed.
MDC00138	New Feature	There is now a Qlik Sense extension that allows to directly request reports from the Qlik Sense user interface.

## Version 2.1.0

### Build #08

With this version, the following changes have been introduced:

MDC00117	Bugfix	A bug that lead to buttons not being clicked correctly in QlikView-Datasources when using a <i>Click Button in Datasource Action</i> , has been fixed.
----------	--------	--

MDC00121	Change	Some log messages have been changed to contain more information about (such as the file path to which a report document is saved etc.).
MDC00122	Bugfix	A bug, that lead to total rows read from Qlik Sense tables containing wrong values in some cases has been fixed.


## Build #07

In dieser Version wurden die folgenden Änderungen vorgenommen:

MDC00113	Bugfix	A bug that lead to (apparently) random crashes of Mail & Deploy Server combined with a <i>Database Locked</i> exception entry in the execution engine log, has been fixed.
MDC00114	Bugfix	A bug that lead to an exception when exporting a pivot table from Qlik Sense that contained at least one column dimension with enabled totals, has been fixed.


## Build #06

With this version, the following changes have been introduced:

MDC00111	New Feature	When designing a report, you can click the  button to quickly replace all references of your report to a specified datasource with a reference to another datasource.
----------	-------------	---

## Build #05

With this version, the following changes have been introduced:

MDC00027	New Feature	When setting up an active directory connection, you can click the  button to test the connection to the active directory.
MDC00072	New Feature	The concept of local variables has been introduced; these variables can be set within a task and then later re-used within the same task. This is particularly useful, if you need to use the same expression multiple times within a task; with local variables, you don't actually need to copy that expressions to all locations where you require them. Instead, you can simply set a local variable to the return value of the expression and then reference the local variable wherever you need it.
MDC00101	New Feature	The concept of global variables has been introduced; these variables can be set within a repository and then later re-used in any expression within the same repository. This is particularly useful if you need an expression in many locations within a repository; instead of copying a single expression to multiple locations, you can set a global variable to the value of that expression and then reference the global variable wherever you need it.
MDC00103	Change	Log-Files of historic task executions are now stored in compressed form in the repository which increases communication performance between Mail & Deploy Client and Mail & Deploy Server when requesting these log-files and significantly reduces the file size of repositories.

MDC00104	Change	The performance of the communication used between Mail & Deploy Client and Mail & Deploy Server has been improved. Loading times are now lower than they have been in previous versions and the user interface should generally be more responsive.
MDC00106	New Feature	Mail & Deploy now supports combo charts contained in QlikView documents.
MDC00107	Bugfix	A bug that lead to an exception when creating copies of items has been fixed.
MDC00108	Bugfix	A bug that lead to filenames of files downloaded from Mail & Deploy Hub having single quotes at the beginning and the end, has been fixed.
MDC00109	Bugfix	A bug that lead to Qlik Sense datasources not being available for selection when adding an action to activate a bookmark in a datasource has been fixed.

## Build #04

With this version, the following changes have been introduced:

MDC00066	Change	The action list editor has been completely rewritten. Improvements include a better visual indication of indented actions as well as the possibility to collapse action containers.
MDC00067	New Feature	In tasks, you can now use File Collections to which you can add external files and/or report documents. These file collections may then be distributed by e-mail either as individual attachments or as a single ZIP file with an optional password.
MDC00075	New Feature	Mail & Deploy now supports development environments where a test system and a production system are connected to each other and reports and tasks can be published from the test system to the production system.
MDC00079	New Feature	Mail & Deploy now supports the creation of report documents from Qlik Sense stories. Individual slides of stories may also be used in other reports.
MDC00080	Change	When Mail & Deploy Client updates itself after connecting to a Mail & Deploy Server, it now longer requires administration rights.
MDC00081	New Feature	In Mail & Deploy Client you can not only see all ongoing executions in the server status page; you can also click on each execution and see the current log file below the list of current executions.
MDC00082	New Feature	Mail & Deploy now supports the distribution of an e-mail directly to one or more active directory groups; the names of the active directory groups to send the e-mail to is an expression and can therefore dynamically be calculated.
MDC00083	Bugfix	A bug that lead to crop settings of trueChart Images not being applied correctly in some cases has been fixed.

MDC00084	Bugfix	A bug that lead to long waiting times after clicking the preview button for a report before the actual preview creation begins, has been fixed.
MDC00086	Bugfix	A bug that lead to crop settings of trueChart Images not being converted correctly when converting a repository with Mail & Deploy Repository Converter has been fixed.
MDC00087	Bugfix	A bug that lead to date time values used in expressions not being converted correctly when converting a repository with Mail & Deploy Repository Converter has been fixed.
MDC00088	Bugfix	A bug that lead to exceptions when creating report documents that contain tables based on Qlik Sense tables with hidden columns, has been fixed.
MDC00089	Bugfix	A bug that lead to an exception when closing the design mode of a Microsoft Word Report Template and then re-entering the design mode, has been fixed.
MDC00093	Bugfix	A bug that lead to horizontal merges of top header cells of pivot tables not working correctly has been fixed.
MDC00094	Bugfix	A bug that lead to exceptions when exporting Qlik Sense pivot tables with top dimensions, has been fixed.
MDC00095	Bugfix	A bug that lead to an exception when refreshing the cache of a Qlik Sense datasource that has a table with date columns without format string, has been fixed.
MDC00096	Bugfix	A bug that lead to variables not being set correctly in Qlik Sense apps when an action to set the variable was used, has been fixed.
MDC00097	New Feature	When exporting trueChart Tables and Tables to Microsoft Word Documents, a new option <i>Keep Rows On Same Page</i> allows you to force Microsoft Word to keep each row of a table on the same page, if possible. This can be used to avoid page breaks within rows.
MDC00098	Bugfix	A bug, that lead to problems when opening Microsoft Excel Report Templates with Microsoft Excel 2010 or below, has been fixed.
MDC00099	Bugfix	A bug, that lead to the memory consumption of Mail & Deploy constantly increasing when executing tasks that access Qlik Sense datasources, has been fixed.

## Build #03

With this version, the following changes have been introduced:


MDC00069	Change	The way you can specify columns when exporting table report elements has been changed. Column specifications can now target a single source column or a range of source columns and they you can specify expressions to customize the actual content of each cell of the exported table.
----------	--------	--

MDC00070	Bugfix	A bug that lead to time-outs when exporting charts, sheets, trueChart Images and trueChart Tables from a Qlik Sense datasource that uses a virtual proxy path has been fixed.
MDC00071	New Feature	When setting up an SMTP E-Mail Server you can now send a test e-mail to test whether the configured settings are correct.
MDC00072	Bugfix	A bug that lead to an exception when trying to distribute a report document by e-mail in a hub report document request has been fixed.
MDC00073	Bugfix	A bug that lead to variables not being read correctly from Qlik Sense datasources when refreshing the datasource cache has been fixed.

## Build #02

With this version, the following changes have been introduced:

MDC00043	Change	The context menu when adding a new action in an action list has been redesigned, so that all actions are now organized in categories.
MDC00044	New Feature	A new action has been introduced that allows to write text into a file.
MDC00045	New Feature	A new action has been introduced that allows to append other report documents or external files to a report document.
MDC00046	New Feature	New actions to save and restore the filter state of datasources have been added.
MDC00047	New Feature	The title of datasource objects can now be inserted in expressions by using a dynamic value.
MDC00048	Change	The execution calendar has been updated so that executions that started just before midnight are no longer cut-off (and in fact invisible) at the end.
MDC00049	New Feature	A module editor has been added which allows you to write Visual Basic .NET code that can be used in expressions.
MDC00050	Change	When rebuilding the datasource cache bookmarks with empty names will be ignored.
MDC00051	New Feature	A new action to save each page of a report document as an image to the file system has been added.
MDC00052	New Feature	The custom width for column specifications of table report elements is now an expression so that column widths may now be calculated.
MDC00053	Bugfix	A bug that lead to an exception when not exporting all cell types in a table report element has been fixed.
MDC00054	Bugfix	The repository converter now correctly converts parameters of an action that creates a report document.

MDC00055	Bugfix	The repository converter now correctly converts expressions that contain a reference to the current value of a cycle action.
MDC00056	New Feature	In Mail & Deploy Client, the execution of a task can now be requested by selecting the task from the list and then clicking the  button.
MDC00057	New Feature	When exporting pivot tables a new setting <i>Indent Rows</i> can now be used so that all columns with left header cells will be combined to one single column with an indent depending on the level of the left header cell.
MDC00058	Bugfix	A bug that lead to padding settings of styles when exporting table elements to HTML not being applied correctly has been fixed.
MDC00059	Change	Tasks that have at least one planned execution in the future based on an execution plan are now displayed with a different image in the task list of Mail & Deploy Client.
MDC00060	Bugfix	A bug that lead to created Adobe PDF Documents and Adobe PDF/A Documents to be protected and thus not printable has been fixed.
MDC00061	Bugfix	A bug that lead to an exception when trying to delete styles has been fixed.
MDC00062	Bugfix	A bug that lead to some cells being empty when exporting data from a pivot table has been fixed.
MDC00063	Bugfix	A bug that lead to some report elements referenced in a Microsoft Word Report not being replaced correctly during the creation of a report document has been fixed.
MDC00064	Bugfix	A bug that lead to filter checks erroneously failing when filtering numeric fields in the datasource has been fixed.
MDC00065	Change	The repository converter will now convert purely numerical expressions of old repositories to numerical expressions in the new repository instead of converting it to a textual value.
MDC00068	Bugfix	A bug that lead to date-times not to be handled correctly when reading data from or filtering fields in datasources has been fixed.