

Peregrine

# AssetCenter



## Desktop Administration

© Copyright 2005 Peregrine Systems, Inc.  
All Rights Reserved.

Information contained in this document is proprietary to Peregrine Systems, Incorporated, and may be used or disclosed only with written permission from Peregrine Systems, Inc. This manual, or any part thereof, may not be reproduced without the prior written permission of Peregrine Systems, Inc. This document refers to numerous products by their trade names. In most, if not all, cases these designations are claimed as Trademarks or Registered Trademarks by their respective companies.

Peregrine Systems® and AssetCenter® are trademarks of Peregrine Systems, Inc. or its subsidiaries.

This document and the related software described in this manual are supplied under license or nondisclosure agreement and may be used or copied only in accordance with the terms of the agreement. The information in this document is subject to change without notice and does not represent a commitment on the part of Peregrine Systems, Inc. Contact Peregrine Systems, Inc., Customer Support to verify the date of the latest version of this document.

The names of companies and individuals used in the sample database and in examples in the manuals are fictitious and are intended to illustrate the use of the software. Any resemblance to actual companies or individuals, whether past or present, is purely coincidental.

If you need technical support for this product, or would like to request documentation for a product for which you are licensed, contact Peregrine Systems, Inc. Customer Support by email at [support@peregrine.com](mailto:support@peregrine.com).

If you have comments or suggestions about this documentation, contact Peregrine Systems, Inc. Technical Publications by email at [doc\\_comments@peregrine.com](mailto:doc_comments@peregrine.com).

This edition applies to version 4.4 of the licensed program

AssetCenter

Peregrine Systems, Inc.  
3611 Valley Centre Drive San Diego, CA 92130  
858.481.5000  
Fax 858.481.1751  
[www.peregrine.com](http://www.peregrine.com)



# Table of Contents

I. Foreword . . . . .	7
Introduction (Desktop Administration) . . . . .	9
Who is Desktop Administration intended for? . . . . .	10
What does Desktop Administration do? . . . . .	11
Chapter 1. General principles . . . . .	13
Main concepts used . . . . .	13
Presentation of the applications . . . . .	14
Architecture of the module . . . . .	15
Automation - Detailed architecture . . . . .	17
Encryption and security . . . . .	18
Chapter 2. Setting up Desktop Administration . . . . .	21
Installation . . . . .	21
Configuration . . . . .	27
II. Automation . . . . .	31
Chapter 3. Creating deployment workflows . . . . .	33
Creating a deployment workflow . . . . .	33
Chapter 4. Using the deployment server . . . . .	37

Modifying the references in the AssetCenter database . . . . .	37
Modifying the private key . . . . .	38
Configuring the <b>Desktop Administration</b> service . . . . .	38
Modifying the server configuration tool options . . . . .	38
Restarting the <b>Desktop Administration</b> service automatically . . . . .	39
Modifying the parameters of a deployment server . . . . .	39
Obtaining information about how the <b>Desktop Administration</b> service functions . . . . .	40
Chapter 5. Examples of deployment workflows . . . . .	41
<b>IDD scan</b> workflow . . . . .	41
<b>PDI scan</b> workflow . . . . .	41
<b>WMI scan</b> workflow . . . . .	42
<b>InstallShield silent installation</b> workflow . . . . .	43
<b>Example of temporary agent</b> workflow . . . . .	43
Chapter 6. Examples of deployment wizards . . . . .	45
Computer deployment wizard . . . . .	46
Service deployment wizard . . . . .	46
Location deployment wizard . . . . .	47
NT-domain computer import wizard . . . . .	47
NT-domain user import wizard . . . . .	47
Chapter 7. Glossary (Desktop Administration) . . . . .	49
Deployment workflow . . . . .	49
Workflow schema . . . . .	50
Deployment instances . . . . .	50
Deployment server . . . . .	50
Deployment target . . . . .	51
Workflow activity . . . . .	51
Workflow event . . . . .	51
Workflow transition . . . . .	51
Agent . . . . .	51
Depot . . . . .	52
Broadcast signal . . . . .	52
Chapter 8. References (Desktop Administration) . . . . .	53
Deployment activities . . . . .	53



# List of Figures

1. Desktop Administration - Structure of the module . . . . .	10
1.1. Desktop Administration module - Architecture . . . . .	16
1.2. Presentation of the architecture . . . . .	17





**PART** | **Foreword**





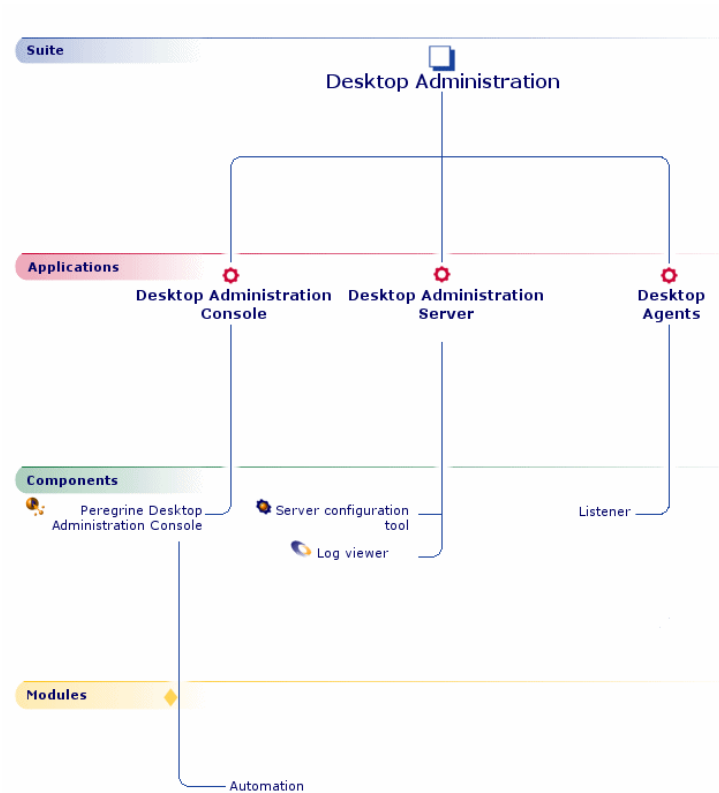


**PEREGRINE**

# Introduction (Desktop Administration)

The structure of the Desktop Administration module is illustrated in the following diagram.

Figure 1. Desktop Administration - Structure of the module



## Who is Desktop Administration intended for?

Desktop Administration is mainly intended for those in charge of WAN or LAN networked IT-portfolio maintenance and who also want to operate on remote computers.

It is generally implemented by the following persons:

- System administrators
- Network administrators
- IT managers

- IT maintenance technicians
- 

 **Warning:**

This guide does not explain the technical knowledge related to the domains mentioned above; it is assumed that you are already familiar with it.

---

## What does Desktop Administration do?

Desktop Administration is an enterprise-class remote control tool.

AssetCenter integrates the administration functionalities of Desktop Administration via the Automation functions. The availability of this module will depend on the license you have acquired from Peregrine Systems, Inc.

Desktop Administration enables you to perform the following tasks:

- Configure remote computers.
- Propagate data to and retrieve data from remote computers.
  - Distribute and deploy software.
  - Verify the implementation of internal security rules on all the computers in your IT portfolio.
  - Prevent the propagation of a virus by stopping the infected computers and servers.
  - And so on.

More generically, this module enables you execute - on a regular or occasional basis - a series of basic tasks on a set of defined computers. The triggering and running of these tasks is determined by a workflow called a deployment workflow.





# 1 General principles

CHAPTER

---

## Main concepts used

The Desktop Administration module uses the following notions:

- [Deployment workflow](#) [page 49]
- [Workflow schema](#) [page 50]
- [Deployment instances](#) [page 50]
- [Deployment server](#) [page 50]
- [Deployment target](#) [page 51]
- [Workflow activity](#) [page 51]
- [Workflow event](#) [page 51]
- [Workflow transition](#) [page 51]
- [Agent](#) [page 51]
- [Depot](#) [page 52]
- [Broadcast signal](#) [page 52]

# Presentation of the applications

Desktop Administration is comprised of several integrated applications, which can be installed directly from the installation CD-ROM. The following sections give a quick description of each.

## Automation module of AssetCenter

This module is installed with AssetCenter and enables you to use the automation (deployment, etc.) capabilities of Desktop Administration.



The availability of this module depends on the license you have acquired from Peregrine Systems, Inc. If you wish to restrict the functionality available through the graphical interface (for example, to fit the needs of individual users), you may enable or disable the available modules using the **File/ Activate modules** menu item.

## Desktop Administration Server

This application contains two components.

### Server Configuration Tool

This component enables you to configure the Desktop Administration server. Using this component you can:

- Start and stop the server.
- Declare the connection information to the database.

### Log Viewer

The Log Viewer enables you to view the log files produced by different Peregrine Systems, Inc., applications. These files have the **.log** extension.

# Desktop Agents

This application contains the **Listener** which is a mandatory component.

---

 **Important:**

In the corresponding documentation, this application is referred to generically as the **Agent**.

---

The **Listener** is the part of the agent that is installed as a service on the remote computer. This service enables the computer to be controlled by managers at any given time. In Windows 2000, XP and Server 2003, a manager can instantly deploy this service on computers using the QuickDeploy! function or a mass deployment workflow.

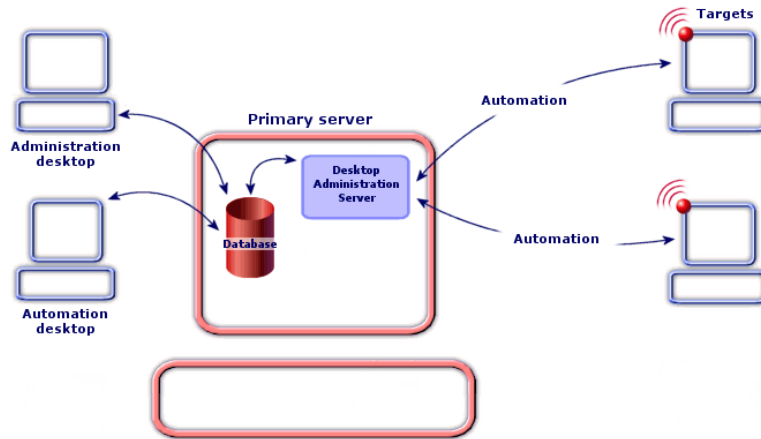
The **Listener** service can be used by other Peregrine Systems products in order to:

- Automatically scan remote computers.
  - Integrate electronic software distribution tools.
- 

## Architecture of the module

The following diagram outlines the architecture of the module.

Figure 1.1. Desktop Administration module - Architecture



As this diagram shows, there are three main areas to the architecture, which are described later on in this manual:

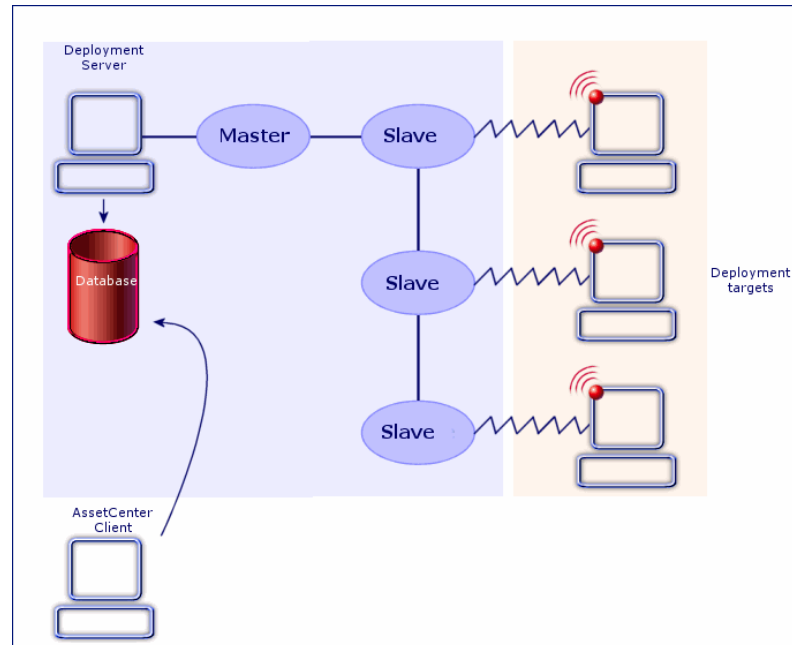
- The AssetCenter client installed on **machines** dedicated in whole or in part to the following tasks: Database administration, automation. It enables you to create deployment workflow schemes, declare the deployment server(s) used and deployment instances. All of these objects are stored in the AssetCenter database.
- The **main server**, where the Desktop Administration server (for automation) is installed. The Desktop Administration server executes the deployment workflow instances on a list of computers stored in the AssetCenter database. It must therefore have access to the AssetCenter database. This access is accomplished via the API layer of AssetCenter.
- The deployment targets are computers on which the deployment is carried out. These computers must be declared in the AssetCenter database, in order for the administration console and the deployment server to be able to access them.



## Automation - Detailed architecture

The following diagram presents a detailed view of the architecture on which the AssetCenter automation functions are based.

Figure 1.2. Presentation of the architecture



When executing a workflow instance, the server triggers a process that controls the activity in its entirety. This process, called the master process, starts (during the deployment, but not simultaneously) a sub-process, also called a slave process, for each target computer.

You can have multiple deployment servers.

---

 Note:

For server performance reasons, the number of slaves running simultaneously on the server must be limited. This number can be defined by the user of the administration console (at the level of the record in the **Deployment servers** table (amDaServer) that corresponds to the deployment server).

---

Depending the workflow activities used, a deployment agent can be installed on these computers. Although this is not absolutely necessary, it facilitates the creation, consistency and execution of your deployment workflows.

---

 Note:

Certain functions are available without the remote agent but which require an operating system from the Windows NT family (Windows 2000, XP, and Server 2003).

---

For example, when you write a Basic script used in a workflow, the API you use will be different depending on whether you access computers with an agent or not. Consequently, if your IT portfolio contains both computers with an without an agent, you must adopt one of the following solutions.

- Write and maintain two sets for workflows for two lists of target computers,
  - Include a test for the agent in your workflows and make them behave accordingly,
  - Etc.
- 

 Note:

The agent is permanently on stand-by and each computer with an agent can broadcast its presence on the network by sending out an activity signal, also called a broadcast signal.

---

---

## Encryption and security

### Why use security keys?

With the Desktop Administration module, you can:

- Perform tasks remotely on deployment targets

- Retrieve information from the deployment targets.

It is important that such actions be perfectly controlled in order that:

- The modifications performed on the target computers match your expectations.
- The information transmitted over the network stays confidential.

This is why we use security keys.

### How do security keys work?

A security key is a text string created using an algorithm that is known only to Peregrine Systems.

The Desktop Administration module uses a double security key for your entire IT portfolio.

- **Public** key: This key is installed on each deployment target using its individual agent.
- **Private** key: This key is installed on each deployment server using the server configuration tool.

The security of the system relies on the extreme difficulty of determining the private key from the public key.

## How identification works with automation

Thus, when a Desktop Administration server addresses an order to execute on a target, the modification order is accompanied by the **private** key. This key is confronted by the target's **public** key. If the keys are compatible, the agent will execute the modification order. Otherwise, the order is rejected.

Furthermore, the information transmitted by the targets is encrypted using the **public** key and decrypted using the **private** key.

Security and confidentiality are thus totally assured.

The **public** and **private** security keys are created at the same time - and one time only - using AssetCenter.





# 2 Setting up Desktop Administration

## CHAPTER

Before using Desktop Administration you must carry out some preliminary steps, which are described in this chapter.

---

## Installation

We have already seen how Desktop Administration is made up of several elements:

- The module of the AssetCenter package.
- A main deployment or remote control server.
- The agents.

You must install these elements before doing anything else.

## Automation

Respect the following order:

- 1 Install the AssetCenter package and create the AssetCenter database.
- 2 Install the deployment server.

---

 **Note:**

Steps 1 and 2 can be performed simultaneously during the installation of AssetCenter.

---

- 3 Declare of the deployment server in the AssetCenter database.
- 4 Install the agents.

## Installing the Automation module in AssetCenter

This module is integrated in the main AssetCenter interface.

Whether or not it is activated, though, depends on the license you have received from Peregrine Systems, Inc. For more information on what is included in your license, please contact a sales representative at Peregrine Systems, Inc.

---


 **Important:**

When installing AssetCenter, make sure the **AssetCenter API** package is installed. This package is required for the module to work correctly.

---

If you are already using AssetCenter 4.4 (or higher) and you want to acquire the Automation module, please contact Peregrine Systems, Inc. for a license extension. You will receive a new license file.

To validate your new rights:

- 1 Launch AssetCenter Database Administrator and connect to your database
  - 2 Select **Action/ Edit license file**
  - 3 Click  and select the new license file
  - 4 Click **OK**
  - 5 The license file is now registered in the database. The next time your start AssetCenter you will have access to the functionality covered by the license.
- 

 **Important:**

Make sure that the Automation module is indeed activated in AssetCenter (**File/ Activate modules** menu).

---

## Creating security keys

Security keys are used to secure the data exchange between the deployment servers and the agents. Note that all the functionalities of Desktop Administration can be used without using any security key at the cost of having no security enforced during the data exchange.

To create security keys:

- 1 Start AssetCenter.
- 2 Execute the **Generate a double security key** wizard via the **Tools/ Actions** menu or directly from the function and favorites pane.
- 3 Populate the following information:

Field	Value
Identity associated with the key	<p>A value of your choice that enables you to name and recognize the key. The identity is stored in the same file storing the public and private keys. Indicate, for example, the name of your company.</p> <p><b>Note:</b></p> <p>This identity is only marginally useful at the moment since you do not have a reason to create more than one double security key.</p>
Length of the key	<p>The longer the key, the more complex the encryption algorithm, and the better the security. But the generation and control of the key will also be longer by a few seconds.</p> <p>In most cases, the value <b>1024</b> is sufficient.</p>
Private key file	Name of the file that stores the private key. The name proposed by default can be modified.
Public key file	Name of the file that stores the public key. The name proposed by default can be modified.

 **Note:**

A third file, **keypub . reg** is also created by this wizard in the same folder as the public key. It is used during the installation of an agent to declare the public key that must be used.

## Installing the deployment server

- 1 Insert the AssetCenter installation CD and select the **Install AssetCenter 4.4** item in the autorun. The installation wizard will guide you through the steps of the installation.
- 2 Select **Modify** and click on **Next**. The installation wizard displays a list of all available components.
- 3 Select the **Desktop Administration Server** package and select the **This function will be installed on the local hard drive** option of the shortcut menu.
- 4 Click on **Next** and let the wizard guide you through the rest of the installation process.

The following elements are installed:

- Desktop Administration Server
- The service associated to the server.
- The server configuration tool.

To secure the data exchange between the server and the agents a private security key must be declared to the deployment server and a public security key must be declared for each of the agents. To declare the security key in the deployment server:

- 1 Start the Deployment server configuration tool.
- 2 Select the **Server/ Select private key** menu.
- 3 Select the file that stores the private key (**keypriv.key** by default).
- 4 Click **OK**.



After the key is installed, the deployment server no longer needs the **keypriv.key** file.

---

## Installing the agents

### To install the agent on one single deployment target

The installation procedure changes whether you want to use a security key to secure the transactions or not.

To install an agent without using a security key:



- 1 Insert the AssetCenter installation CD.
- 2 Select the **Install Desktop Agents** item in the autorun. The installation wizard will guide you through the steps of the installation.

To install an agent using a security key:

- 1 Copy the contents of the **daagent** folder of the installation CD of AssetCenter in a local folder.
- 2 Copy the **keypub . reg** file, generated using the **Generate a double encryption key** wizard of AssetCenter in the same folder.
- 3 Launch setup.exe and let the wizard guide you through the installation steps. The installation detects the presence of the **keypub . reg** file and automatically declares the security key in the agent.

After the key is installed, the agent no longer needs the **keypub . reg** file.

Whatever installation mode you choose the **Peregrine Listener 6.0.2** service is declared and automatically started.

## To install the agent on several deployment targets

If you have a large number of agents to install, we recommend performing one of the following solutions:

- Perform a silent installation.
- You can also use the **Mass Deploy** deployment workflow (provided with the line-of-business data) if the target computers use Windows 2000, XP, or Server 2003.

### Silent installation

Silent installation enables you to quickly install the agent from a DOS prompt.

To perform a silent installation:

- 1 Copy the contents of the **daagent** folder of the installation CD of AssetCenter in a local folder.
- 2 Copy the **keypub . reg** file, generated using the **Generate a double encryption key** wizard of AssetCenter in the same folder.
- 3 Execute the following command line on the computers of your NT domain:

```
setup.exe /S /V "/qn MSI_TCPPORT=<Agent listening port> MSI_BROADCAST_TARGET=<Broadcast parameter> REINSTALLMODE=vomus REINSTALL=ALL "
```

The following table details the parameters of this command line:

Switch (es) / Property	Description
/S, /qn	These parameters are used to suppress all interaction with the user and are mandatory in the case of an unattended installation.
/V	This mandatory parameter allows to send commands to the installation engine.
MSI_TCPPOINT	This property declares the listening port of the agent. The default value is 1738.
MSI_BROADCAST_TARGET	This property declares the broadcast parameters of the agent in the following format: <ul style="list-style-type: none"> <li>INET:&lt;broadcast target&gt;:&lt;broadcast port&gt;</li> <li>&gt;</li> <li>■ &lt;broadcast target&gt;: IP address of the target. This parameter is usually empty.</li> <li>■ &lt;broadcast port&gt;: UDP port used for the broadcast.</li> </ul>



#### Note:

You can append the **/L\*v debug.log** switch to the previous command line to force the creation of an installation log. If you encounter a problem during the installation we recommend that you create this log file prior to contacting Peregrine Systems support.

## Installing agents with MassDeploy workflow

AssetCenter enables you to install agents on remote computers using the **MassDeploy** workflow.

This workflow is part of the line-of-business data provided with AssetCenter.

To use this workflow, you must:

- 1 Copy the **deploy** folder installed during the Desktop Administration Server installation (by default **c:\Program Files\Peregrine\Desktop Administration Server\depot**) in the deployment server's file depot.
- 2 Copy the public security key file **keypub.reg** in the **deploy** folder of the deployment depot.
- 3 Edit the impersonation activities of the **MassDeploy** workflow by specifying the authentication information in the **Impersonation** tab (**Deployment/Deployment workflows** menu).

The scripts associated with the **MassDeploy** workflow are documented at the level of the workflow itself.



Note:

Otherwise, you can run the deployment server on an administrator account of the domain.

---

#### 4 Launch the **MassDeploy** workflow.

The workflow schema performs the following actions:

- 1 Connects to a remote computer using an **Impersonate** activity.
- 2 Stops services and the listener if they are already present on the computer.
- 3 Copies files in the **deploy** folder to the remote computer.
- 4 Installs and launches the listener as a service.
- 5 Stops and uninstalls temporary services.

---

## Configuration

### Configuring the deployment server

The deployment server must have access to the AssetCenter database. This database contains all the data needed for the deployment: target computers, deployment workflows, etc. Configuring the server essentially consists in declaring the database connection information.

Configuring the deployment server is carried out by the server configuration tool. This tool automatically detects the computer on which the server is installed.

You must declare the database to which the server must connect:

- 1 Launch the configuration tool.
- 2 If necessary, stop the service (**Service/ Stop** menu).
- 3 Select your server (**Server/ Configure the database** menu).

The **Deployment servers** table in the AssetCenter database is populated automatically.

- 4 Start the service (**Service/ Start** menu).

The information relating to the server appears in a few seconds.

The deployment server is executed as an NT service.

## Configuring AssetCenter.

To configure a deployment server:

- 1 Start AssetCenter.
- 2 Select **Deployment/ Deployment servers**.
- 3 Select the record that was automatically created when you configured the deployment server.
- 4 Complete the information listed in the table below:

Field	Description
Name	Name of the deployment server.
Computer	Computer on which the deployment server runs. This computer must be in the list of computers in the database.
Depot	<p>The path of the depot folder on the deployment server. This folder stores the files that are exchanged between the server and the deployment targets. This folder is located in the installation folder of Desktop Administration Server.</p> <p>Example: C:\Program Files\Peregrine\Desktop Administration Server\depot.</p> <p><b>Note:</b></p> <p>This folder is essential for any file-transfer operation between the server and the deployment targets. By default, the path is relative to the installation folder.</p>
Broadcast detection	<p>List of addresses used by a deployment workflow for which the <b>Start on broadcast</b> option is selected (<b>Properties</b> tab).</p> <p>These addresses are the server's listening addresses for the computers transmitting broadcast signals.</p> <p>The address uses the INET format and has the following syntax:</p> <pre>INET::<port1; inet::<port2;<="" pre=""> </port1;></pre>
Maximum number of slaves	<p>Maximum number of slave processes that the server can start at a time.</p> <p>Reduce the default value if the server's performances are insufficient.</p>
Default server	Check this box in order for the declared server to be default deployment server.

- 5 Click **Create** to validate your information.

## To configure a previously installed agent

To configure a previously installed agent you must completely reinstall the agent using the command line described in the [Installing agents with MassDeploy workflow](#) [page 26] section.





# Automation

**PART**







# 3 Creating deployment workflows

## CHAPTER

---

## Creating a deployment workflow

The first step is the creation of a workflow that details the successive steps performed during a deployment for a deployment target. This deployment workflow plays the role of a template for the instances that will be executed on the deployment targets.

Creating a deployment workflow consists of defining:

- Activities
- Activity output events that enable you to activate transitions.
- Transitions that trigger activities.

You can access the list and the detail of deployment workflows via the **Deployment/ Deployment workflows** menu. The **Activities** tab of the window that appears is divided into two panes:

- The left pane gives you a tree view of the structure of a workflow.
- The right pane gives you a graphical view of the workflow. This pane also enables you to edit the deployment workflow in a user-friendly and graphical manner.
- The starting of each workflow is defined in the **Properties** tab, which contains the **Start on broadcast** option.

This option tells the deployment server not to try starting the workflow on a computer until the computer is announced on the network by a broadcast message.

In order for this option to work, the server needs to have been configured to receive such messages, and the computers must have an installed and configured agent in order to transmit such messages.

This option enables you to, for example, reliably launch a deployment on laptop computers or computers that are shut off during the day.

This section explains how to use this graphical editor to create, modify or delete the elements of workflow:

- Activities
- Events
- Transitions

## Activities

To create an activity, right-click in an empty zone of the **Activities** tab, then select an activity in one of the available categories. The table below lists the available activities according to their category:

Activity category	Available activities
Core activities	<ul style="list-style-type: none"> <li>■ Success</li> <li>■ Failure</li> <li>■ Retry</li> <li>■ Jump</li> <li>■ Empty activity</li> <li>■ Synchronization</li> <li>■ Wait</li> </ul>
File management	<ul style="list-style-type: none"> <li>■ Upload files</li> <li>■ Download files</li> <li>■ Move files</li> <li>■ Copy files</li> <li>■ Rename</li> <li>■ Delete files</li> <li>■ Create folders</li> <li>■ Delete folders</li> </ul>
Script	Script

Activity category	Available activities
Action	Action
Messaging	Messaging

The detail of an activity and its properties are automatically displayed when you select an activity.

To delete an activity:

- You can either select the activity by clicking on it with the mouse, then pressing "Delete" on the keyboard.
- Or you can select the activity, right-click and select **Delete** from the shortcut menu that appears.

 **Note:**

You can also perform all selections, creations and deletions from the tree-list view of the workflow.

The set of available activities in Desktop Administration: ► [References \(Desktop Administration\)](#) [page 53].

## Events

Events are the outputs of activities. They enable the activation of transitions, which trigger other activities. The available events vary depending on the nature of the activity:

For example:

- ◆ A **File management** type activity has **OK** and **Error** for events by default. But you can access additional events **No file**.

To add an event to an activity:

- 1 Select the concerned activity and right-click.
- 2 Select the event to add: **Error**, **OK**, **Return value**, **No file**, etc.

## Transitions

To create a transition:

- 1 Select the event that enabled the transition by clicking it with the mouse.
- 2 Hold the mouse button down and drag the event all the way to the destination activity.

To delete a transition:

- You can either select the transition by clicking on it with the mouse, then pressing "Delete" on the keyboard.
- Or you can select the transition, right-click and select **Delete** from the shortcut menu that appears.

To modify the source and/or destination of a transition:

- 1 Select the transition.
- 2 Drag the extremity you want to modify.



# 4 Using the deployment server

## CHAPTER

With section [Configuring the deployment server](#) [page 27], you will have learned how to use the server configuration tool at installation time.

This chapter tells you more about the day-to-day use of the server configuration tool.

---

## Modifying the references in the AssetCenter database

To modify the references in the AssetCenter database containing the deployment workflows and the description of the deployment targets:

- 1 Start Desktop Administration Server's Server configuration tool.
- 2 Select the **Server/ Configure the database** menu.
- 3 Modify the information relating to the database connection.
- 4 Validate the new parameters by clicking **OK**.
- 5 If the **Desktop Administration** service is already started, select the **Server/ Reload configuration** menu.

---

## Modifying the private key

- 1 Create new security keys.
  - ▶ [Creating security keys](#) [page 23]
- 2 Start Desktop Administration Server's Server configuration tool.
- 3 Select the **Server/ Modify the private key** menu.
- 4 Indicate the full path of the file that stores the new private key.
- 5 Validate the new parameters by clicking **OK**.

---

## Configuring the Desktop Administration service

When you installed Desktop Administration Server, a service was created.

To modify the parameters of the **Desktop Administration** service

- 1 Start Desktop Administration Server's Server configuration tool.
- 2 Select the **Service/Configure** menu.
- 3 Validate the new parameters by clicking **OK**.

To start, stop or restart the **Desktop Administration** service:

- 1 Start Desktop Administration Server's Server configuration tool.
- 2 Select the **Service/Stop**, **Service/Start** or **Service/Restart** menu.

---

## Modifying the server configuration tool options

- 1 Start Desktop Administration Server's Server configuration tool package.
- 2 Select the **File/Options** menu.
- 3 Modify the options.
- 4 Validate the new parameters by clicking **OK**.

For more information on option parameters, refer to the AssetCenter **Tailoring** guide, chapter **Customizing a client workstation**, section **AssetCenter interface options**.

---

## Restarting the Desktop Administration service automatically

To have the server configuration tool attempt automatically and regularly restarting the **Desktop Administration** service whenever it stops:

- 1 Start Desktop Administration Server's Server configuration tool.
- 2 Select the **File/ Watchdog mode** menu.

---

 **Note:**

When the automatic restart option is activated, a mark appears in front of the **Watchdog mode** entry of the **File** menu.

You cannot adjust the frequency of attempts to restart the service.

---

---

## Modifying the parameters of a deployment server

- 1 Start AssetCenter.
- 2 Select **Deployment/ Deployment servers**.
- 3 Select the deployment server to modify.
- 4 Update the deployment server's parameters.
- 5 Validate the new parameters by clicking **Modify**.
- 6 Start Desktop Administration Server's Server configuration tool package.
- 7 If the **Desktop Administration** service is already started, select the **Server/Refresh caches** menu.  
If the **Desktop Administration** service is not yet started, select the **Service/Restart** menu.

# Obtaining information about how the Desktop Administration service functions

- 1 Display the user interface of Desktop Administration Server's Server configuration tool package.
- 2 Select the **File/Refresh** menu.
- 3 Consult the **Server information** section in the workspace.
- 4 For more details, select the **Service/ Display log** menu.

To learn more about how the Log viewer program works, refer to the AssetCenter **Administration** guide, chapter **Consulting the log (.log) files**.





# 5 Examples of deployment workflows

## CHAPTER

This chapter describes some of the deployment workflows that are part of the **line-of-business data** installed with AssetCenter.

It also gives you some examples of deployment workflows that you can create yourself.

---

## IDD scan workflow

This workflow launches a Desktop Inventory scan on a remote computer and copies the Desktop Inventory scan files to AssetCenter.

You can then import the scan data to your database using a Connect-It workflow.

---

## PDI scan workflow

This workflow launches a Desktop Inventory scan on a remote computer and copies the Desktop Inventory scan files to AssetCenter.

This workflow differs from the **IDD scan** workflow by the type of **.xml.gz** files generated.

To work correctly, the script used by the scan executable must have the `/o` parameter. This is so it doesn't account for the default values of the scan executable, but instead accounts for those of the workflow.

For example:

```
c:\scanW32.exe /o:C:\[Computer.Name]
```

The path indicated is the one of the scan executable's folder. And each scan file generated will be saved in a file having the same name as the computer.

---

## WMI scan workflow

This workflow is an example of how using the API to access the Windows Management Instrumentation.

This workflow enables you to retrieve the information relating to the scan data of a remote computer:

- Processor
- RAM
- BIOS
- Video card
- Sound card
- Operating system
- And so on.

## Required configurations

The WMI service is installed by default for Windows XP and Windows 2000.

For Windows NT4.0, you need to have already installed the Service Pack 4, as well as WMI Core 1.5.

WMI Core 1.5 is available on the Microsoft Web site.

## InstallShield silent installation workflow

This workflow is a deployment example of a program based on the InstallShield installation engine.

The command parameters are defined in the **Execution** tab of the **RunSetup** activity.

The parameters in this workflow show how to use the silent installation method via the **-s** parameter and an answer file **.iss**.

The creation of this answer file is done using the **-r** parameter.

For more information about InstallShield, refer to the documentation provided with this program.

## Example of temporary agent workflow

This workflow addresses the security requirements of certain companies to not install an agent that will run permanently on a computer (server, etc.). An agent is installed temporarily to execute a task and is then uninstalled.

This workflow uses NT security and requires you to have NT administrator security privileges.

The example workflow is as follows:

- 1 Impersonation of the remote computer.
- 2 Installation of the listener on a remote computer in a temporary folder:  
Installation of the **iftlsnr.exe**, **iftagt.exe** and **iftsys.exe** files.
- 3 Creation and launching of the temporary **iftlsnr.exe** service on a listening port different from the standard port.
- 4 Addition of activities via the listener.
- 5 Stopping the listener and uninstalling the temporary folder.

The command line to launch the service is the following:

```
iftlsnr.exe -temp -listen:INET::1740 -svc
```

- **-svc**: This parameter indicates that the program is launched as a service.
- **-temp**: This parameter indicates that the .INI files or the Registry or not read from or written to.

- **-listen:INET::1740**: This parameter indicates that the program **iftlshr.exe** uses port 1740 instead of port 1738 (the default port) to avoid conflicts with an agent already installed.

To indicate that the deployment server uses a different port, the following command line is used:

```
DaSetContext "Computer.ForceConnection", "INET:" & DaContext ("Computer.Name") & ":1740"
```

- ◆ The context variable **Computer.ForceConnection** enables you to specify the port number by using the following syntax: **INET:<Name of the computer or IP address>:<Port number>**.

For more information about the syntax used, refer to the **Programmer's reference** guide.



# 6 Examples of deployment wizards

CHAPTER

Wizards are used to automate the execution of deployment workflows.

A wizard enables you to, for example, select a list of deployment targets and execute a deployment workflow on each selected target.

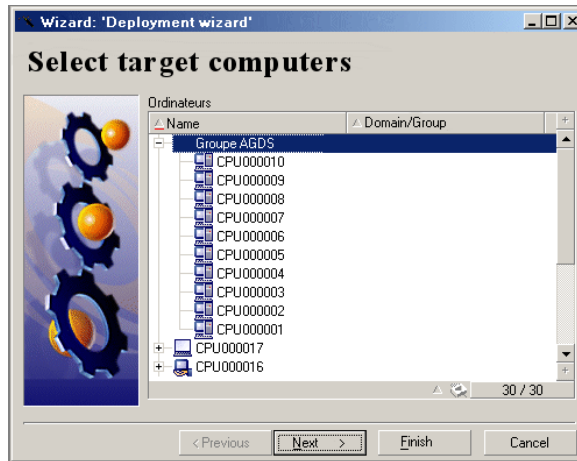
You can access AssetCenter's wizards via the **Tools/ Actions** menu.

The following wizards are provided with the AssetCenter demonstration database.

For more information about creating a wizard, refer to the **Administration** guide in the chapter **Wizards**.

## Computer deployment wizard

This wizard enables you to deploy a workflow on one or more computers recorded in the **amComputer** table.



Deployment is limited to 99 computers in multiple selection.

We recommend creating computer groups. A group can be made up of as many items as required.

When you create a computer group, the deployment takes into account the members of this groups. It is possible for this operation to be recursive because the Computers table (**amComputer**) is hierarchic.

## Service deployment wizard

This wizard enables you to deploy a workflow on one or more services recorded in the **amEmpIDept** table.

The wizard searches for the computers used by the employees of the selected departments (**amEmpIDept**).

---

## Location deployment wizard

This wizard enables you to deploy a workflow at one or more locations recorded in the **amLocation** table.

The wizard searches for the computers linked to the selected locations.

---

## NT-domain computer import wizard

This wizard enables you to:

- 1 Select an NT network.

The **Extract list from domain controller** option enables you to choose between information coming from the master explorer or those coming from the domain controller.

If you selected this option, make sure you have network administrator rights.

If you cleared this option, the information about the local computer on which you executed the wizard will not be retrieved.

- 2 Retrieve the list of computers on this network.
- 3 Create the corresponding records in the **amComputer** table.

The wizard retrieves the **Name** and **First** (First name) data depending on the customizations made to the operating system. You must therefore update the wizard script if the data to retrieve does not follow the "Name, First" order defined in the NT account manager.

---

## NT-domain user import wizard

This wizard enables you to:

- 1 Choose a Windows NT network.
- 2 Retrieve the list of users that have a login on this Windows NT network.
- 3 Create the corresponding records in the **amEmpIDept** table and populate the **First name** and **Last name** fields.

---

 Note:

The information about the local computer on which you executed the wizard will not be retrieved.

---





# 7 Glossary (Desktop Administration)

CHAPTER

---

## Deployment workflow

The deployment workflow is the formalization and/or automation of the deployment procedures.

For example, the following processes can be modeled and automated using workflow methods:

- The deployment of an application on an IT portfolio.
- The periodic execution of a hardware and software scan.
- Etc.

AssetCenter enables you to define deployment workflow schemas and manage their procedures.

---

 **Warning:**

The deployment workflows differ from the classic workflows available in AssetCenter. In principal, they are the same, but their functionality and interfaces are different from one workflow to the next.

---

---

## Workflow schema

Creating a workflow schema involves defining:

- Activities
- Activity output events that enable you to activate transitions.
- Transitions that trigger activities.

---

## Deployment instances

A deployment instance is a single occurrence of an execution of a workflow schema. It is transmitted to the deployment server to be processed and executed. You can look at a deployment workflow instance as one unit of elementary work for a deployment server.



Note:

The deployment instance is thus a copy of the deployment workflow that you initially created. You can thus modify the original deployment workflow as you wish during the execution of a deployment instance based on this workflow.

---

---

## Deployment server

The deployment server is the core of this Desktop Administration module. It executes the deployment instances on a list of the machines stored in the database.

You may use one or more deployment servers.

---

## Deployment target

A deployment target is a machine on which is deployment is performed. Targets are declared in the Computers table of the AssetCenter database.

---

## Workflow activity

A workflow activity is made up of:

- A task to perform.
- Events that trigger transitions to other activities.

The administration console proposes a series of typical activities, such as downloading, execution, etc.

---

## Workflow event

Workflow events are the outputs of activities. They enable the activation of transitions, which trigger other activities.

---

## Workflow transition

A workflow transition enables you to go from one activity to another. This is triggered by the occurrence of an event.

An event can be associated with several transitions.

---

## Agent

The agent is a program that enables the computer on which it is installed to be controlled by the deployment server.

The **Listener** is an executable installed as a service on the computer. This service, running on permanent stand-by on the network, enables the computer to be controlled by the deployment servers.

---

## Depot

Sub-folder of the Desktop Administration Server administration folder that regroups the following folders:

- **deploy**  
This folder contains the scan executables used by the **MassDeploy** workflow.
- **idd**  
This folder contains the scan executables used by the **IDD scan** workflow.
- **pdi**  
This folder contains the scan executables used by the **PDI scan** workflow.
- **wallpaper**  
This folder is used by the **Change the wallpaper** workflow.

---

## Broadcast signal

Each agent is identified on the network by an **activity signal** called the **broadcast signal**. This signal is recognized by the deployment servers.



# 8 References (Desktop Administration)

## CHAPTER

---

## Deployment activities

This section provides an exhaustive preview of all the available deployment activities, which are classified by category.

## Core activities

This category regroups all the activities inherent in a workflow, such as its entry point (the **Start** activity created by default for all new workflows) or its different output points (**Failure** or **Success**, for example).

Most activities transmit two kinds of output messages: **OK** or **Error**. We recommend that you process these messages with a **Success** or **Failure** type activity.

### Success

End-of-workflow activity that was correctly executed.

Stops the workflow and interrupts the activities in progress.

## Failure

End-of-workflow activity that could not be executed.

Stops the workflow and interrupts the activities in progress.

## Retry

End-of-workflow activity that triggered the same workflow another time.

The workflow is restarted from the **Start** activity.

## Jump

Workflow activity that jumped to another named activity. To define the target activity, select an activity in the **To** field of the activity detail.

This activity enables you to reduce the transitions in a workflow schema and make reading it easier.

## Empty activity

Workflow activity provoking a single output message: OK.

This activity is used if there are not an identical number of workflow-event activities that must be synchronized.

For example:

- The **Send** workflow activity uses three events: OK, no file, error.
- The **Execution** workflow activity uses two events: OK, error.

To synchronize these two activities, you must create an empty activity for the **Send** activity that processes the 'OK' and 'no file' events.

## Synchronization

This activity provokes a successful output (the output event is validated) if all the input events are validated. In practice, several transitions are connected to this activity. The activity verifies all the transitions, and if it cannot verify them, it waits until it can do so. A workflow can only result from this activity after all the input transitions are verified.

## Wait

This activity pauses the deployment workflow. The duration of this pause is defined in the **Duration** field of the activity detail.

# System management

This category groups together those activities that act directly on the system.

## Impersonate

This activity enables you to perform tasks on a remote computer to:

- ◆ Perform an action in the place of the named user.

You can therefore have access to the same:

- Objects, with the same rights as the named user.
- Environment variables as the named user.
- Preference folders and Registry as the named user.

An impersonating activity does not have any immediate results. It modifies the behavior of other activities such as:

- Execution
- Edition of the Registry
- RPC/ WMI

The impersonation parameters are populated in the **Impersonate** tab of the activity detail:

- **User:** Login of the user you want to impersonate.

The **@LoggedOnUser** instruction makes it possible to use the identity of the currently logged user. If no user is logged, the activity fails.

The **@LocalSystem** instruction enables you to use the default system account (all local rights on the machine).

---

### Important:

The **@LoggedOnUser** and **@LocalSystem** instructions are only valid for the activities involving the agent. For the RPC/WMI activities, these instructions are meaningless, because authentication is performed via NT security.

---

### Note:

If a user is never physically logged on, the preferences relating to this user are not saved.

- 
- **Domain:** User domain.

---

 **Note:**

To specify a local account of the machine, you must specify the name of the machine, for example using the following syntax:

```
[Computer . Name]
```

---

- **Password:** Password associated with the user login.
  - **Test:** This operation opens a DOS box on the remote computer and verifies the validity of the populated parameters (login, domain, password).  
In order to work, this test requires a listener on the remote computer.
- 

 **Note:**

Do not perform several **impersonating** activities at the same time in the same workflow.

---

This activity does not work with Windows 9.x operating systems.

---

## Execution

This activity executes a program stored on the deployment target. All the execution parameters are stored in the **Execution** tab of the activity detail.

- **Command:** Name of the program to execute and parameters.
  - If you do not specify a path, the program to execute must be located in your operating system's **Path** environment variable.
  - If you specify a path, it must be the absolute path. Par example  
C:\temp\sample.exe
- **Path:** Current folder of the program to execute. This parameter is not mandatory.
- **Synchronous:** Select this option for the execution to be synchronous. The activity does not let the user control the computer until the program is executed.
- **Log:** Select the option that best suits you for the information to enter into the log file.
- **Visibility:** This option enables you to either force the display of graphical interface of the executed program or hide it.
  - **By default:** Only the operating system decides on the interface's display.
  - **Force display:** interface displayed.
  - **Hide:** interface hidden.



## Shutdown

This activity shuts down a deployment target. You can choose whether to restart or shut down the target by selecting the corresponding option in the **Method** field of the activity detail.

## Wake on LAN

This activity sends a wake on LAN signal to a deployment target. It wakes up the target machine.

The **Wake on LAN** activity only works in the following cases:

- The target computer has a motherboard and a network card that support this function.
- The target computer is declared in the database and the **Physical address** field is populated for the computer. This address is the computer's MAC address.

## File management

This category groups together those activities that act on files and folders.

### Upload files

This activity sends files to a deployment target.

The parameters of this activity are available in the **Files** tab of the activity detail:

- **Source folder:** Path of the files to copy on the deployment target. It is a relative path depending on the path of the file depot on the deployment server.
- **Destination folder:** Absolute path of the files copied to the deployment target.
- **Filter(s):** Set of file names (wildcard characters authorized) that are separated by a semi-colon. For example:

\*.gif; iftmsgsr.exe

If this field is not populated, all the files in the specified folder will be taken into account.

You can also select the following options:

- **Recursive:** If this option is selected, the copy is recursive.  
Copies all files and sub-folders in the current folder.

- **Replace:** If this option is selected, the copy will overwrite any files with the same name in the destination folder of the target.
- **Force:** If this option is selected, the copy will not take into account the read-only protections on the target.
- **Resume:** If this option is selected, the activity tries to resume sending files in case of a transfer interruption.

## Download files

This activity copies files from the deployment target to the deployment server. The parameters of this activity are available in the **Files** tab of the activity detail:

- **Source folder:** Path of the files to copy to the depot of the deployment server. It is an absolute path on the deployment target.
- **Destination folder:** Relative path (depending on the path of the file depot) of the files copied to the deployment server.
- **Filter(s):** Set of file names (wildcard characters authorized) that are separated by a semi-colon. For example:

```
*.gif; iftmsgr.exe
```

If this field is not populated, all the files in the specified folder will be taken into account.

You can also select the following options:

- **Recursive:** If this option is selected, the copy is recursive.
- **Replace:** If this option is selected, the copy will overwrite any files with the same name in the destination folder of the target.
- **Force:** If this option is selected, the copy will not take into account the read-only protections on the server.
- **Resume:** If this option is selected, the activity tries to resume receiving files in case of a transfer interruption.

## Move files

This activity moves files locally to the server or the deployment target. The way in which it works is determined by the **Perform the operation on the server** option.

The parameters of this activity are available in the Files tab of the activity detail:

- **Source folder:** Source path of the elements to move. Depending on how the function operates, it is either a path relative to the file depot (in the case of a local move to the deployment server) or an absolute path (in the case of a local move to the deployment target).

- **Destination folder:** Destination path of the elements to move. Depending on how the function operates, it is either a path relative to the file depot (in the case of a local move to the deployment server) or an absolute path (in the case of a local move to the deployment target).
- **Filter(s):** Set of file names (wildcard characters authorized) that are separated by a semi-colon. For example:

```
*.gif; iftmsgr.exe
```

If this field is not populated, all the files in the specified folder will be taken into account.

You can also select the following options:

- **Perform the operation on the server:** If this option is selected, the activity operates on the deployment server and not the target.
- **Recursive:** If this option is selected, the move is recursive.  
Moves all files and sub-folders from the current folder.
- **Replace:** If this option is selected, the move will overwrite any files with the same name in the destination folder of the target.
- **Force:** If this option is selected, the copy will not take into account the read-only protections on the target or the server.

## Copy files

This activity copies files locally on the server or the deployment target. The way in which it works is determined by the **Perform the operation on the server** option.

The parameters of this activity are available in the Files tab of the activity detail:

- **Source folder:** Source path of the elements to copy. Depending on how the function operates, it is either a path relative to the file depot (in the case of a local copy to the deployment server) or an absolute path (in the case of a local copy to the deployment target).
- **Destination folder:** Destination path of the elements to copy. Depending on how the function operates, it is either a path relative to the file depot (in the case of a local copy to the deployment server) or an absolute path (in the case of a local copy to the deployment target).
- **Filter(s):** Set of file names (wildcard characters authorized) that are separated by a semi-colon. For example:

```
*.gif; iftmsgr.exe
```

If this field is not populated, all the files in the specified folder will be taken into account.

You can also select the following options:

- **Perform the operation on the server:** If this option is selected, the activity operates on the deployment server and not the target.
- **Recursive:** If this option is selected, the copy is recursive.  
Copies all files and sub-folders in the current folder.
- **Replace:** If this option is selected, the copy will overwrite any files with the same name in the destination folder of the target.
- **Force:** If this option is selected, the copy will not take into account the read-only protections on the target or the server.

## Rename

This activity renames files locally on the server or the deployment target. The way in which it works is determined by the **Perform the operation on the server** option.

The parameters of this activity are available in the Files tab of the activity detail:

- **Current name:** Full source path of the file to rename. Depending on how the function operates, it is either a path relative to the file depot (in the case of a local renaming in the deployment server) or an absolute path (in the case of a local renaming in the deployment target).
- **New name:** This parameter contains the new file name.

You can also select the following options:

- **Perform the operation on the server:** If this option is selected, the activity operates on the deployment server and not the target.
- **Replace:** If this option is selected, the copy will overwrite any files with the same name in the destination folder of the target.
- **Force:** If this option is selected, the copy will not take into account the **read-only** option on the target or the server.

## Delete files

This activity deletes files locally on the server or the deployment target. The way in which it works is determined by the **Perform the operation on the server** option.

The parameters of this activity are available in the Files tab of the activity detail:

- **Folder:** Path of the elements to delete. Depending on how the function operates, it is either a path relative to the file depot (in the case of a local deletion from the deployment server) or an absolute path (in the case of a local deletion from the deployment target).

- **Filter(s):** Set of file names (wildcard characters authorized) that are separated by a semi-colon. For example:

```
*.gif; iftmsgr.exe
```

If this field is not populated, all the files in the specified folder will be taken into account.

You can also select the following options:

- **Perform the operation on the server:** If this option is selected, the activity operates on the deployment server and not the target.
- **Recursive:** If this option is selected, the deletion of files is recursive. Deletes all files and sub-folders from the current folder.
- **Force:** If this option is selected, the copy will not take into account the **read-only** option on the target or the server.

## Create folders

This activity creates files locally and recursively on the server or the deployment target or the server. The way in which it works is determined by the **Perform the operation on the server** option.

The parameters of this activity are available in the **Files** tab of the activity detail:

- ◆ **Folder(s):** Path of the folder to create. Depending on how the function operates, it is either a path relative to the file depot (in the case of a local creation in the deployment server) or an absolute path (in the case of a local creation in the deployment target).

You can also select the following option:

- ◆ **Perform the operation on the server:** If this option is selected, the activity operates on the deployment server and not the target.

## Delete folders

This activity deletes files locally from the server or the deployment target. The way in which it works is determined by the **Perform the operation on the server** option.

The parameters of this activity are available in the **Files** tab of the activity detail:

- ◆ **Folder(s):** Path of the folder to delete. Depending on how the function operates, it is either a path relative to the file depot (in the case of a local deletion from the deployment server) or an absolute path (in the case of a local deletion from the deployment target).

You can also select the following option:

- **Perform the operation on the server:** If this option is selected, the activity operates on the deployment server and not the target.
- **Recursive:** If this option is selected, the deletion is recursive.
- **Force:** If this option is selected, the copy will not take into account the read-only protections on the target or the server.

## Script

This activity enables you to define a script that will be executed on the deployment target. This script uses functions that are detailed in the **Programmer's reference** guide.

You can access a script editor with syntax highlighting and line numbers by pressing **F4** while in the detail of a script.

---

 **Note:**

All workflow activities are available in API form via this script-type activity. Certain other functions (API, RPC) are only available from a script.

---

## Action

This activity enables you to execute an action (in the sense given by AssetCenter) contextually on a database table. The **Action** tab regroups the parameters of this activity:

- **Action:** SQL name of the action to be executed.  
Because the action is contextual, only those tables relating to the deployment target are available.
- **Table:** Select one of the tables on which the action is executed.

## Messaging

This activity enables you to execute a messaging type action.

AssetCenter lets you manage two types of messages:

- Messages issued from AssetCenter and sent to the AssetCenter database via its internal messaging system.
- Messages created in AssetCenter and sent via an external messaging system.

The workflow's sending parameters are defined in the **Messaging** tab.

In order for a messaging-type activity to be corrected executed, you need to have already configured the messaging system according the protocols you use, as well as the deployment server.

For more information about configuring the messaging system, refer to the "Administration" guide, chapter "Messaging".

