Altova FlowForce Server 2017
7.11.0 10.11 01.00 00.1101 2011
User and Reference Manual
Oser and Neterence Manual

# Altova FlowForce Server 2017 User & Reference Manual

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Published: 2017

© 2017 Altova GmbH

# **Table of Contents**

1	Altova FlowForce Server 2017	3
1.1.	What's New	4
2	FlowForce Server Overview	8
2.1.	Basic Concepts	9
	Security Concepts	
	How It Works	
	Web Administration Interface	
3	Installation	20
3.1	System Requirements	21
3.2	Linux	22
	3.2.1Installing FlowForce Server	23
	3.2.2Migrating to FlowForce Server 2017	25
	3.2.3Viewing Currently Installed Products	26
	3.2.4Uninstalling FlowForce Server	27
	3.2.5Installation Directory	28
3.3	Mac OS X	29
	3.3.1Installing FlowForce Server	30
	3.3.2Migrating to FlowForce Server 2017	32
	3.3.3Uninstalling FlowForce Server	33
	3.3.4Installation Directory	
3.4	Windows	35
	3.4.1Installing FlowForce Server	36
	3.4.2Windows Service Configuration	
	3.4.3Network Configuration (Windows Only)	
	3.4.4Migrating Data Manually	42
	3.4.5Installation Folder	
3.5	Migrating Altova Servers	44
4	Licensing	50

4.1 <i>P</i>	Altova LicenseServer	51
4	.1.1Network Information	53
4	.1.2Installation (Windows)	54
4	.1.3Installation (Linux)	56
4	.1.4Installation (Mac OS X)	58
4	.1.5Altova ServiceController	59
4	.1.6How to Assign Licenses	60
	Start LicenseServer	61
	Open LicenseServer's Config Page (Windows)	62
	Open LicenseServer's Config Page (Linux)	65
	Open LicenseServer's Config Page (Mac OS X)	
	Upload Licenses to LicenseServer	
	Register Product/s	72
	Register Altova Desktop Products	72
	Register FlowForce Server	
	Register MapForce Server	
	Register RaptorXML(+XBRL) Server	
	Register Style Vision Server	82
	Assign Licenses to Registered Products	
4	.1.7Configuration Page Reference	
	License Pool	
	Client Management	
	Client Monitoring	
	Settings	
	Messages, Log Out	
4	.1.8Password Reset	109
5 (	Configuring the Server	112
5.1F	lowForce Server Application Data	113
5.20	Opening the Setup Page	115
5.3I	Defining the Network Settings	117
	Configuration File Reference	
	etting the Default Time Zone	
	etting Mail Parameters	
	Changing the Windows Active Directory Settings	
	starting and Stopping Services (Linux)	
	tarting and Stopping Services (Mac OS X)	
5.10S	tarting and Stopping Services (Windows)	130
5 11 I	ocalizing FlowForce Server	

5.12	2Setting Environment Variables	
6	Managing User Access	134
6.1.	Users and Roles	135
	6.1.1Creating Users	136
	6.1.2Creating Roles	137
	6.1.3 Importing Domain Users and Roles	138
	6.1.4Default Users and Roles	140
	6.1.5Renaming Users and Roles	141
	6.1.6Assigning Roles to Users	142
	6.1.7Assigning Roles to Other Roles	143
	6.1.8Resetting the Root Password	144
6.2.	Privileges	145
	6.2.1How Privileges Work	146
	6.2.2Viewing Privilege Reports	150
6.3.	Permissions and Containers	152
	6.3.1How Permissions Work	153
	6.3.2Understanding Containers	156
	6.3.3 Creating, Renaming, and Moving Containers	158
	6.3.4Viewing Container Permissions	161
	6.3.5 Setting Container Permissions	162
	6.3.6Restricting Access to the /public Container	164
6.4.	Password Policies	166
	6.4.1How Password Policies Work	167
	6.4.2Creating and Assigning Password Policies	168
7	Working with Jobs	170
7.1.	Credentials	171
	7.1.1 How Credentials Work	172
	7.1.2 Defining Credentials	173
7.2.	Creating Jobs	174
7.3.	Duplicating jobs	175
	Managing Input Parameters	
	Managing Steps	
	Processing Steps Sequentially	
	Processing Steps Conditionally	
	Handling Step Errors	
	Declaring the Return Type of a Job	
,.,.	vermany use recents ripe or a boo	100

7.10Handling Data Types in Steps	189
7.11Managing Triggers	192
7.11.1Timer Triggers	194
7.11.2File System Triggers	196
7.11.3HTTP Triggers	198
7.12Exposing Jobs as Web Services	200
7.13Caching Job Results	202
7.14Defining Queue Settings	205
7.15Viewing the Job Log	206
7.16Stopping Jobs	207
7.17Importing and Exporting Jobs	208
8 The FlowForce Expression Language	216
8.1Creating a "Hello, World!" Job	217
8.2Computing an Expression	219
8.3The Expression Language Rules	220
8.4Embedding Expressions in String Fields	
8.5Calling Expression Functions	225
8.6FlowForce Data Types	226
8.7Operators	228
8.8Example: Check if a Path Exists	230
9 MapForce/StyleVision Server Integration	234
9.1Accessing the Mapping/Transformation Result	236
9.2Example: Generate Multiple PDF's from Multiple XML's	
10 Command Line Interface	250
10.1help	252
10.2 assignlicense	
10.3compactdb	
10.4createdb	
10.5debug	
10.6exportresourcestrings	
10.7foreground	
10.8initdb	
10.9install	

10.10.licenseserver	261
10.11. migratedb	
10.12.repair	
10.13.resetpassword	
10.14.setdeflang (sdl)	
10.15.start	
10.16.uninstall	
10.17upgradedb	
10.18.verifylicense	
11 Built-in Functions	272
11.1/system/abort	273
11.2/system/compute	274
11.3/system/compute-string	275
11.4/system/filesystem	277
11.5/system/ftp	281
11.6/system/mail	
11.7/system/maintenance	
11.8/system/shell	
11.9/RaptorXML	
11.10./RaptorXMLXBRL	
12 Expression Functions	296
12.1Step Result Functions	
12.2Stream Functions	
12.3File System Functions	301
12.4File Path Functions	
12.5List Functions	305
12.6String Functions	
12.7Boolean Functions	
12.8Runtime Information Functions	
13 Job Configuration Examples	314
13.1Creating a Job from a MapForce Mapping	315
13.2 Using a Joh as Step of Another Joh	

13.3Creating a Directory Polling Job	
13.4Adding Error Handling to a Job	
13.5Exposing a Job as a Web Service	
13.6Caching Job Results	
13.7Creating a Job from a StyleVision Transformation	
13.8Using RaptorXML to Validate a Document	
13.9Validate an XML Document with Error Logging	
13.10.Using RaptorXML to Pass Key/Value Parameter Pairs	
14 Glossary	352
14.1C	
14.2E	
14.3F	
14.4I	
14.5J	
14.6P	
14.7R	
14.8S	
14.9T	
14.10.U	
14.11.W	

# Index

# **Chapter 1**

**Altova FlowForce Server 2017** 

# 1 Altova FlowForce Server 2017

FlowForce Server is a cross-platform software solution used to automate tasks on Windows, Linux, and Mac OS X servers and workstations through a Web interface.

FlowForce Server integrates with other Altova server products (MapForce Server, StyleVision Server, and both flavours of Raptor XML Server) and automates their functions, such as executing a MapForce mapping, executing a StyleVision transformation, or validating an XML or XBRL file. With FlowForce Sever you can also create and automate common server tasks, such as sending emails, managing files on the local system or network, managing files through a File Transfer Protocol (FTP), running shell scripts, and others.



The FlowForce Server documentation is organized into the following main sections:

- FlowForce Server Overview
- Installation
- Licensing
- Configuring the Server
- Managing User Access
- Working with Jobs
- The FlowForce Expression Language
- MapForce Server and StyleVision Server Integration
- Command Line Interface
- Built-in Functions
- Expression Functions
- Job Configuration Examples
- Glossary

Last updated: 21 April 2017

# 1.1 What's New

#### FlowForce Server 2017r3

- FlowForce Server integrates not only with MapForce Server, but also with a new edition of
  it, "MapForce Server Accelerator Edition". Both MapForce Server and MapForce Server
  Accelerator Edition integrate seamlessly with FlowForce; either of them (though not both)
  can run simultaneously alongside FlowForce. Starting with Release 2017r3, during
  FlowForce Server installation on Windows and OS X, you are prompted to optionally
  install MapForce Server Accelerator Edition (not MapForce Server, as in previous
  releases).
- Both FlowForce Server and FlowForce Web Sever can accept not only plain
   (unencrypted) HTTP but also HTTPS (SSL-encrypted) connections. The HTTPS
   connectivity settings can be defined either from the Setup page (see <u>Defining the Network Settings</u>) or by means of configuration files (see <u>Configuration File Reference</u>). On
   Windows, the connectivity settings can also be defined during installation, see <u>Network Configuration</u> (Windows Only).
- During FlowForce Server installation on Windows, the Windows services corresponding to FlowForce Server and to FlowForce Web Server can be configured to run as a specific operating system user. The start-up manner of the service (automatic, manual) can also be set up, see Windows Service Configuration.
- A new system maintenance command is available at the command line: <u>compactdb</u>.

### FlowForce Server 2017

- Improved behaviour at system shutdown
- Internal updates and optimizations

# FlowForce Server 2016 R3

- Steps, triggers, and parameters of a job can be duplicated using the new **Duplicate** ( ) button. Also, after you delete a step, trigger, or parameter, an **Undo** ( ) button is available (see Managing Input Parameters, Managing Steps, and Managing Triggers).
- Jobs called by other jobs display parameter descriptions as tooltips, if a description is available. For information about adding parameter descriptions, see <u>Adding Input</u> Parameters.

### FlowForce Server 2016

- Existing jobs can be saved with a new name, acting as templates for new jobs (see <u>Duplicating jobs</u>).
- Enhancements to job management: you can select for deletion or export not only specific
  jobs within a container, but also whole containers, including sub-containers (see
  Importing and Exporting Jobs ).
- It is now possible to perform a root password reset from the command line interface (see Resetting the Root Password).
- Performance improvements: timeouts of the Web interface due to intensive job runs are now significantly reduced.
- The built-in T/system/mail/send function supports SSL/TLS connections to the SMTP

server (see Setting Mail Parameters ).

#### FlowForce Server 2015 R4

- The FlowForce Server installer for Mac OS X is now available as a disk image (.dmg) file (see Installing FlowForce Server).
- Greater flexibility to handle the outcome of failed jobs. When the job execution fails, a
  new "Abort on error" parameter determines whether the job should be aborted, or the
  function should return false. This parameter applies to functions under <a href="/>/system/filesystem"/system/filesystem"/system/filesystem</a>, /system/ftp, /system/mail, and /system/shell.
- Performance improvements: large jobs are loaded significantly faster in the job configuration page.

# FlowForce Server 2015 R3

You can now view from the Web administration interface the date and time when an
object (such as a job or function) was created or modified, and the user name who
modified it (see <u>Understanding Containers</u>).

### FlowForce Server 2015

- Windows Active Directory integration
- Configure the complexity of users passwords
- Reports on system privileges
- Job import and export
- Stop running jobs

# FlowForce Server 2014 R2

- Enhancements to FlowForce caching
- Ability to rename users and roles
- Ability to rename containers/objects
- New Step expression functions
- <u>List parameters</u> for RaptorXML (key/value parameter pairs)

# FlowForce Server 2014

- New server maintenance functions
- Ability to cache job results and use them in other jobs
- Extended RaptorXML support

# FlowForce Server 2013 R2

- Integration with RaptorXML Server edition
- Job flow control allowing the execution of job steps based on conditions
- Ability to repeat execution steps any number of times
- Definition of step variables allowing the results of one step to be used in following job

steps

 An expanded set of <u>built-in steps</u> allowing mail notifications, FTP server interaction, and the ability to compute expressions

Last updated: 2017-04-21

# **Chapter 2**

FlowForce Server Overview

# 2 FlowForce Server Overview

This section provides an overview of FlowForce Server concepts, functions, and its browser-based interface, namely:

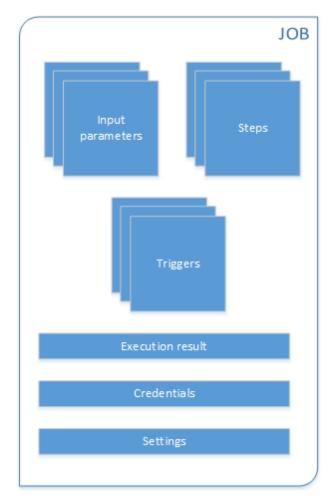
- Basic Concepts
- Security Concepts
- How It Works
- Web Administration Interface

FlowForce Server Overview Basic Concepts 9

# 2.1 Basic Concepts

#### **Jobs**

A job is a core concept in FlowForce Server. It represents a task or a sequence of tasks to be executed by the server. Jobs can be as simple as one-step tasks such sending an email. However, you can also create jobs that perform multiple actions and pass the result (for example, a file) as parameter to another job. A job consists of <u>input parameters</u>, <u>steps</u>, <u>triggers</u>, and other settings.



Structure of a FlowForce job

# **Input parameters**

In the context of a FlowForce Server <u>job</u>, input parameters represent some arbitrary information supplied to the job based on which the job will execute in a particular way. Job input parameters are similar to function arguments in a programming language. Input parameters can be of various types (for example, file or directory references, text, numbers, Boolean values, and others). For some job types, FlowForce Server creates the input parameters automatically.

# **Triggers**

When creating a job, you must specify the conditions (or criteria) that will trigger the job, known as triggers. FlowForce Server monitors any defined triggers and executes the job whenever the

trigger condition is met.

# Steps

Steps define what the FlowForce Server job must actually do (for example, delete a file, execute a MapForce mapping, or send an email). In its simplest form, a step is just an operation with either failed or successful outcome, and it requires a <u>function</u> to be executed (see <u>Execution step</u>). However, there are also steps that provide means to execute other steps conditionally or in a loop (see <u>Choose step</u>, <u>Error/success Handling step</u>, and <u>For-Each step</u>). You can create as many steps as required within the same job, and you can set the order in which they must take place.

#### **Functions**

In the context of a <u>job</u>, a function is an instruction understood by FlowForce Server that performs some operation on the target file system. A function can be one of the following:

- A built-in FlowForce function (see Built-in Functions)
- A StyleVision transformation
- A MapForce mapping
- The execution step of a job

Most functions have input parameters. Any mandatory input parameters must be supplied by the caller in order for the step to execute successfully.

#### **Execution result**

In FlowForce Server, a step's execution result defines what is returned after the step is executed (for example, a file, or some text). When working with jobs, you can explicitly declare a step's execution result to be of a specific data type (such as String or Boolean), or be discarded. You typically need to declare the data type of the execution result if you intend to use it in other jobs, or if you want to cache the result.

# **Credentials**

A credential record is the combination of user name and password associated with a user account on the operating system where the FlowForce Server job runs. When you define a <u>job</u> in FlowForce Server, you must supply the credentials with which the job must be executed. Note that if the user account associated with the supplied credentials does not have sufficient rights on the operating system, the job cannot execute successfully.

# **Settings**

When creating a job, you can configure the following optional settings:

- Make the job available as a Web service (see also Exposing Jobs as Web Services)
- Limit the number of instances running in parallel for the same job (see also <u>Defining</u> <u>Queue Settings</u>).
- Cache the result returned by the job (see also Caching Job Results).

FlowForce Server Overview Security Concepts 11

# 2.2 Security Concepts

FlowForce Server uses a role-based user access control mechanism configurable according to the needs and structure of your organization or business model. For example, you can organize and package jobs and credentials into special data containers that require access rights in order to be viewed or modified. Only users with corresponding access rights would then be able to access data inside the container.

# **Containers**

As the name of the term implies, a container is data packaged together. In FlowForce Server, containers can be roughly compared to folders on an operating system. Containers can contain any of the following: <a href="mailto:jobs">jobs</a>, <a href="mailto:credentials">credentials</a>, <a href="functions">functions</a>, and other containers. By setting <a href="permissions">permissions</a> on containers, you can control who can view or access the data inside them. Organizing data into containers and setting up the relevant permissions for each container is a good security practice.

### Users

Users are persons who log on to FlowForce Server to configure jobs, deploy MapForce or StyleVision transformations, or manage the FlowForce Server. The actions available to users in FlowForce Server depend on the following:

- a) Their assigned permissions or privileges
- b) The permissions and privileges assigned to any roles that users are members of.

# **Roles**

Roles are named sets of <u>privileges</u> that help enforce security based on the business need. The typical role-based security involves at least two roles: an administrator and a standard user. Each role is defined by the privileges granted to that role. For example, administrators can change their own password and that of other users, whereas standard users can change only their own password. You can assign roles to users and revoke roles from users as necessary.

# **Privileges**

Privileges define what users can do in FlowForce Server (for example, set own password, read users and roles, stop any job, and so on). Privileges are different from <u>permissions</u> in the sense that permissions control user access to <u>containers</u>, whereas privileges are effective globally across FlowForce Server. The following simple rule might help you distinguish quickly between privileges and permissions: privileges are global, permissions are local.

Like permissions, privileges can be assigned both to individual users and to <u>roles</u>. Therefore, when users log on to FlowForce Server, their set of effective privileges is determined by:

- a) the privileges they have been assigned directly
- b) the privileges assigned to any roles that the user is member of.

#### **Permissions**

Permissions control user access to <u>containers</u>. Like privileges, <u>permissions</u> can be granted both to users and to <u>roles</u>. Therefore, if a user is a part of a role, any permissions granted to the role will automatically apply to the user as well.

By default, permissions set on a container are inherited from the parent container. For example, let's assume that container A has a child container B. Users who have permission to access

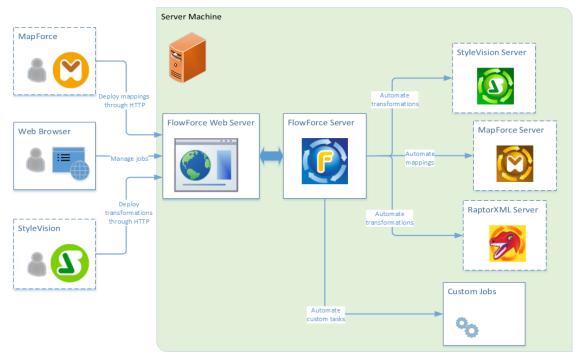
container A will have by default permission to access container B as well. However, an administrator can redefine the permissions of any user or role at every level of the container hierarchy.

# **Password policies**

FlowForce Servers uses password policies to help administrators manage the complexity of user passwords. A password policy is a set of minimum requirements that a user password must meet in order to be valid (for example, at least *N* characters long).

# 2.3 How It Works

The typical FlowForce Server installation is illustrated in the following diagram. Components that are optionally licensed are represented with dashed borders.



Typical FlowForce Server installation

As the diagram shows, the typical FlowForce Server installation consists of several server products (or, in technical terms, services) that are installed on the same server machine as FlowForce Server. The FlowForce Server solution itself is composed of two services: FlowForce Web Server and FlowForce Server. These two run as separate services and can be configured, started, or stopped separately. The manner in which these two services are managed depends on the operating system family (Linux, Windows, Mac OS), as further described in this documentation.

FlowForce Web Server accepts and validates requests from clients, and passes them to FlowForce Server. FlowForce Server is the core of the FlowForce Server solution and runs as a background service without a graphical user interface. FlowForce Server continuously checks for trigger conditions, starts and monitors job execution, and writes detailed logs. In addition to this, FlowForce Server listens to requests for jobs that were exposed as Web services (it can be configured to accept HTTP requests both from the local machine and from remote clients, see Defining the Network Settings).

FlowForce Web Server, on the other hand, handles requests to the Web administration interface where you define or monitor jobs, or manage various FlowForce settings. FlowForce Web Server accepts HTTP (or HTTPS) connections from the following types of clients:

Web browser	The Web browser is used to configure FlowForce Server jobs
	and other settings (for an overview, see Web Administration

	Interface).
MapForce Enterprise or Professional Edition	MapForce is a data mapping desktop application where you visually design the mappings that transform your data or convert it from one format to another.
	Once the mappings are created and tested in MapForce, you can deploy them to FlowForce Server, in order to convert them into flexibly configurable jobs. For example, you can configure the mapping jobs to run at a specific time daily, or whenever a file is added to a monitored directory.
	In order to run jobs created from MapForce mappings, FlowForce Server calls MapForce Server (or MapForce Server Accelerator Edition), whose role is to actually execute the mappings and produce the resulting output files.
	Both MapForce Server and MapForce Server Accelerator Edition integrate seamlessly with FlowForce; however, only one of them can be installed at the same time alongside FlowForce. By default, when installing FlowForce, you will be prompted to optionally install the MapForce Server Accelerator Edition.
StyleVision Enterprise or Professional Edition	StyleVision is a desktop application used to design reports and forms based on XML, SQL database, and XBRL inputs.
	Once a stylesheet has been tested and debugged, it can be deployed to FlowForce Server. The deployed files are then available for use in any transformation job on the server.
	To execute jobs created from deployed StyleVision transformations, FlowForce Server calls StyleVision Server, whose role is to execute the transformation and produce the resulting output files.

For further information about each product, refer to the Altova documentation page (<a href="https://www.altova.com/documentation.html">https://www.altova.com/documentation.html</a>).

# RaptorXML Server

14

Altova RaptorXML Server (also called RaptorXML for short) is Altova's third-generation, super-fast XML processor, optimized for the latest standards and parallel computing environments. Designed to be highly cross-platform capable, the engine takes advantage of today's ubiquitous multi-core computers to deliver lightning-fast processing of XML. RaptorXML is available in two editions: (i) RaptorXML Server and (ii) RaptorXML+XBRL Server. The RaptorXML+XBRL Server edition includes support for validating and processing XBRL (eXtensible Business Reporting Language) documents, in addition to XML.

When RaptorXML Server is installed on the same server as FlowForce Server, its functions become available as built-in FlowForce Server functions. This means that you can create jobs that validate or check the well-formedness of XML documents, or transform XSLT and XQuery documents. For the list of functions, see <a href="RaptorXML Server Functions">RaptorXML Server Functions</a>.

#### 2.4 **Web Administration Interface**

The FlowForce Server Web administration interface is the front end of FlowForce Server from where you administer the server and configure jobs. You can access the Web administration interface from a Web browser at the configured address and port.

The following sections describe the menu items available in the Web administration interface.

- Home
- Configuration
- Log
- Administration
- Help

Note: Access to resources and actions available from the Web administration interface is driven by a user access control mechanism. This means that you can access and modify configuration data as long as your assigned permissions allow it. Similarly, you can perform actions (and see the corresponding menu items) if you have been granted the corresponding privilege.

#### **Home**

Displays the list of currently running or aborted jobs and active triggers.

To refresh the list of jobs or triggers, click **Reload Grid** ( ).



# Configuration

Displays the currently defined FlowForce containers, jobs, credentials, and functions. To view the contents and further information about any object, click the corresponding record.

The following containers are available by default:

- /public
- /system
- /RaptorXML (if you have licensed RaptorXML Server)

For further information about containers, see Understanding Containers.

From the Configuration page, you can also manage containers, jobs, credentials, and functions, and set permissions on containers, provided you have the relevant access rights.

### Log

Displays the server log. The displayed log entries include both server-related and job-related messages and depend on your access rights.

You can narrow the results by time, job name and the minimum severity (info, warning, error).

You can sort the results in ascending or descending order by any of the following criteria:

15

Date	Specifies the date when the entry was logged.		
Severity	Specifies the severity of the logged entry. Valid values:     Info     Warning     Error		
Module	Specifies the internal FlowForce Server module from where the logged entry originates.		
User	Specifies the name of the user associated with the logged entry.		
InstanceID	Specifies the system instance ID of the logged entry (if applicable).		
Message	Specifies the message text associated with the logged entry.		

To refresh the server log, click **Refresh** ( ).



To export the contents of the log, click **Export**. All records that match the filter criteria will be exported as a sa a zipped JSON (JavaScript Object Notation) file. When you click Export, you may either be prompted to save the zip archive to a local directory, or it will be downloaded to your browser's default download directory. The maximum number of records that can be exported is 100,000 (this limitation is imposed by performance reasons). Note that, for the duration of the export, the Web interface remains unresponsive to any other HTTP requests.

# Administration

From the Administration area, you can perform actions to related server configuration and user management. The Administration area consists of the following menu items:

Allows you to create, delete, and maintain users, including changing their privileges, roles, and password policy. For further information, see Managing User Access.

## Roles

Allows you to create, delete, and maintain access control roles. For further information, see Users and Roles.

# Password policies

Allows you to create, delete, and maintain password complexity policies. For further information, see Password Policies.

# Reports

Allows you to view reports on currently assigned user privileges. For further information, see Viewing Privilege Reports.

# Settings

Allows you to define the default time zone, mail server, and Windows Active Directory settings. For further information, see Configuring the Server.

# Help

Opens the FlowForce Server documentation in a separate browser tab or window.

# Chapter 3

Installation

# 3 Installation

This section includes the FlowForce Server installation procedures applicable to  $\underline{\text{Linux}}$ ,  $\underline{\text{Mac OS X}}$ , and  $\underline{\text{Windows}}$  platforms.

Installation System Requirements 21

# 3.1 System Requirements

# **Platform support**

- ▼ Linux
  - · CentOS 6 or newer
  - · RedHat 6 or newer
  - Debian 7 or newer
  - Ubuntu 12.04 or newer

The following libraries are required as a prerequisite to install and run the application. If the packages below are not already available on your Linux machine, run the command yum (or apt-get if applicable) to install them.

Required by	CentOS, RedHat	Debian	Ubuntu
FlowForce Server	libidn, krb5-libs	libidn11, libgssapi- krb5-2	libidn11, libgssapi- krb5-2
LicenseServer	krb5-libs	libgssapi-krb5-2	libgssapi-krb5-2

# ▼ Mac OS X

Mac OS X 10.10, 10.11, 10.12 or newer

# ▼ Windows

Windows Vista, Windows 7/8/10

# ▼ Windows Server

Windows Server 2008 R2 or newer

# **Browser support**

The Web administration interface can be accessed with the latest versions of Microsoft Internet Explorer, Mozilla Firefox, and Google Chrome.

# 3.2 Linux

This section includes installation tasks applicable to FlowForce Server running on Linux, as follows:

- Installing FlowForce Server
- Migrating to FlowForce Server 2017
- Viewing Currently Installed Products
- Uninstalling FlowForce Server
- Installation Directory

# 3.2.1 Installing FlowForce Server

# **Prerequisites**

 Ensure that the latest version of <u>Altova LicenseServer</u> is installed and running either on the local machine or network.

- Perform installation either as **root** user or as a user with **sudo** privileges.
- If you are installing FlowForce Server with other Altova server products, it is recommended to install FlowForce Server first.

#### To install FlowForce Server

- 1. Download the installation package from the Altova website to a local directory.
- Change to the directory where you downloaded the FlowForce Server installation package and install it.

To change directory, run the cd command. For example, if you downloaded the package to the directory /home/User/MyAltova, run the following command:

```
cd /home/User/MyAltova
```

To install FlowForce Server, run the following command:

#### Debian

```
sudo dpkg --install flowforceserver-2017-debian.deb
```

# Ubuntu

```
sudo dpkg --install flowforceserver-2017-ubuntu.deb
```

## CentOS, RedHat

```
sudo rpm -ivh flowforceserver-2017-1.x86_64.rpm
```

If you have installed other Altova server products before installing FlowForce Server, copy
the .tool file from /etc directory of the corresponding product to the FlowForce Server /
tools directory.

The .tool file contains the path to the executable of each Altova server product. To copy the tool file, run the cp command. For example, if you are copying the .tool file from the MapForce Server installation path, run:

```
cp /opt/Altova/MapForceServer2017/etc/*.tool /opt/Altova/
FlowForceServer2017/tools
```

4. Start the FlowForce Web Server service.

```
[CentOS 6]: sudo initctl start flowforcewebserver
[CentOS 7]: sudo systemctl start flowforcewebserver
[Debian 7]: sudo /etc/init.d/flowforcewebserver start
```

```
[Debian 8]: sudo systemctl start flowforcewebserver
[RedHat]: sudo initctl start flowforcewebserver
[Ubuntu <=14]: sudo initctl start flowforcewebserver
[Ubuntu 15]: sudo systemctl start flowforcewebserver</pre>
```

On first run, if there are no custom configuration files, FlowForce Web Server starts on a random port and creates a setup page.

Make sure that the random port address on which FlowForce Web Server was started is not blocked by your firewall.

To open the setup page in a Web browser, run (remember that this action requires **root** privileges):

```
file:///var/opt/Altova/FlowForceServer2017/flowforceweb.html
```

If your Linux machine does not have a Web browser, extract the URL to the setup page from the FlowForce Web server log file, using the following command:

```
sudo grep running /var/opt/Altova/FlowForceServer2017/data/ffweb.log
```

Note that, on Debian 8, Ubuntu 15 and CentOS 7, the log is written to the system log. Therefore, on such systems you can extract the URL to the setup page from the system log:

#### CentOS 7

```
sudo grep running /var/log/messages
```

## Debian 8, Ubuntu 15

```
sudo grep running /var/log/syslog
```

The output is similar to:

```
FlowForceWeb running on http://127.0.0.1:34597/setup?key=52239315203
```

To access the setup page from another machine on the network, type this link into the address bar of your browser and replace "127.0.0.1" with the host name of your server machine.

- 5. Register FlowForce Server with Altova LicenseServer (see Register FlowForce Server).
- Configure the network address and port of the Web administration interface (see <u>Setting</u> the <u>Network Address and Port</u>).
- Log on to the Web administration interface (by default, http://localhost:8082, unless you changed the address and port in the previous step) and change the default FlowForce Server root password. The default login name and password is root and root, respectively.

# 3.2.2 Migrating to FlowForce Server 2017

# **Prerequisites**

- FlowForce Server 2017 must be installed (see Installing FlowForce Server).
- Perform data migration either as root user or as a user with sudo privileges.

# To migrate data to FlowForce Server 2017

1. Stop the FlowForce Web Server service.

```
[CentOS 6]: sudo initctl stop flowforcewebserver
[CentOS 7]: sudo systemctl stop flowforcewebserver
[Debian 7]: sudo /etc/init.d/flowforcewebserver stop
[Debian 8]: sudo systemctl stop flowforcewebserver
[RedHat]: sudo initctl stop flowforcewebserver
[Ubuntu <=14]: sudo initctl stop flowforcewebserver
[Ubuntu 15]: sudo systemctl stop flowforcewebserver
```

2. Remove or rename the data directory created during installation.

### Debian, Ubuntu

```
sudo rm -rf /var/opt/Altova/FlowForceServer2017/data
```

# CentOS, RedHat

```
sudo rm -rf /var/opt/Altova/FlowForceServer2017/data
```

3. Migrate existing data by running the migratedb command available in the command-line interface of FlowForce Server.

```
sudo /opt/Altova/FlowForceServer2017/bin/flowforceserver migratedb
--olddatadir=/var/opt/Altova/FlowForceServer2015/data --datadir=/var/opt/
Altova/FlowForceServer2017/data
```

Start the FlowForce Web Server service.

```
[CentOS 6]: sudo initctl start flowforcewebserver
[CentOS 7]: sudo systemctl start flowforcewebserver
[Debian 7]: sudo /etc/init.d/flowforcewebserver start
[Debian 8]: sudo systemctl start flowforcewebserver
[RedHat]: sudo initctl start flowforcewebserver
[Ubuntu <=14]: sudo initctl start flowforcewebserver
[Ubuntu 15]: sudo systemctl start flowforcewebserver</pre>
```

# 3.2.3 Viewing Currently Installed Products

To check if an Altova server product is installed

• Run the following command:

# Debian, Ubuntu

```
dpkg --list | grep [product name]
```

Replace [product name] with one of the following: flowforceserver, stylevisionserver, mapforceserver, licenseserver.

# CentOS, RedHat

```
rpm -qa | grep server
```

# 3.2.4 Uninstalling FlowForce Server

# To uninstall FlowForce Server:

• Run the following command:

# Debian, Ubuntu

sudo dpkg --remove flowforceserver

# CentOS, RedHat

sudo rpm -e flowforceserver

If you need to uninstall other packages, use the same command as above and replace "flowforceserver" with the package name of the package you want to remove. For example:

# Debian, Ubuntu

sudo dpkg --remove licenseserver

# CentOS, RedHat

sudo rpm -e licenseserver

# 3.2.5 Installation Directory

On Linux, the location of the FlowForce Server installation directory is as follows:

/opt/Altova/FlowForceServer2017/bin

# 3.3 Mac OS X

This section includes installation tasks applicable to FlowForce Server running on Mac OS X, as follows:

- Installing FlowForce Server
- Migrating to FlowForce Server 2017
- Uninstalling FlowForce Server
- Installation Folder

# 3.3.1 Installing FlowForce Server

## **Prerequisites**

 Ensure that <u>Altova LicenseServer</u> is installed and running either on the local machine or network.

- Perform installation as a user with administrative (root) privileges.
- If you are installing FlowForce Server with other Altova server products, it is recommended to install FlowForce Server first.
- The Mac machine must be configured so that its name resolves to an IP address (that is, you must be able to successfully ping the host name from the Terminal using the command ping <hostname>).

#### To install FlowForce Server:

- Download the disk image (.dmg) file from the Altova website ( <a href="http://www.altova.com/download.html">http://www.altova.com/download.html</a>) and click to open it. This causes the installer to appear as a new drive on your computer.
- Double-click FlowForceServer2017.pkg, and go through the successive steps of the
  installation wizard. These are self-explanatory and include one step in which you have to
  agree to the license agreement before being able to proceed. When you complete the
  wizard, the FlowForce Server icon becomes available in Applications. The package is
  installed in the directory /usr/local/Altova/FlowForceServer2017/.
- 3. If you have not installed Altova License Server already, or if you want to upgrade to the latest version, double-click the corresponding package (.pkg) file and follow the on-screen instructions. Altova LicenseServer is required to run any Altova server application, including FlowForce Server (see Licensing).
- 4. In **Applications**, double-click the FlowForce Server icon. This opens the FlowForce Server configuration page in the browser.

If the name of the Mac machine cannot resolve to an IP address (see <a href="Prerequisites">Prerequisites</a>), the browser opens a page with the following message: "FlowForceWeb does not appear to be available at <a href="http://<hostname>:<port>">http://<hostname>:<port>">http://<hostname>:<port>">lease restart it and reload this page." If you see this message, do the following:

- a. Click the link mentioned in the message.
- b. In the browser's address bar, replace <hostname> with either localhost or the IP address of your Mac.
- 5. From the configuration page, register FlowForce Server with Altova LicenseServer (see Register FlowForce Server).
- 6. Configure the network address and port of the Web administration interface (see <u>Setting</u> the Network Address and Port).
- Log on to the Web administration interface (by default, http://localhost:8082, unless you changed the address and port in the previous step) and change the default FlowForce Server root password. The default login name and password is root and root, respectively.

#### To eject the installer drive after installation:

• Right-click it and select "Eject".

Note:

If you have already installed other Altova server products before installing FlowForce Server, copy the .tool file from /etc directory of the corresponding product to the FlowForce Server /tools directory. The .tool file contains the path to the executable of each Altova server product. To copy the tool file, run the cp command in the terminal. For example, if you are copying the .tool file from the MapForce Server installation path, run:

cp /usr/local/Altova/MapForceServer2017/etc/\*.tool /usr/local/Altova/
FlowForceServer2017/tools

# 3.3.2 Migrating to FlowForce Server 2017

## **Prerequisites**

- FlowForce Server 2017 must be installed (see Installing FlowForce Server).
- Perform data migration either as a user with administrative (root) privileges.

## To migrate data to FlowForce Server 2017:

1. Stop the FlowForce Server service.

sudo launchctl unload /Library/LaunchDaemons/com.altova.FlowForceServer.plist

2. Stop the FlowForce Web Server service.

sudo launchctl unload /Library/LaunchDaemons/
com.altova.FlowForceWebServer.plist

3. Remove or rename the data directory that was created during installation. This will delete any objects in the new version that were created after installation prior to performing this manual upgrade:

sudo rm -rf /var/Altova/FlowForceServer2017/data

4. Run the migratedb command:

sudo /usr/local/Altova/FlowForceServer2017/bin/FlowForceServer
migratedb
--olddatadir=/var/Altova/FlowForceServer2015/data --datadir=/var/
Altova/FlowForceServer2017/data

5. Start the FlowForce Server service.

 $\verb|sudo| launchctl| load| / \verb|Library/LaunchDaemons/com.altova.FlowForceServer.plist|$ 

6. Start the FlowForce Web Server service.

sudo launchctl load /Library/LaunchDaemons/
com.altova.FlowForceWebServer.plist

# 3.3.3 Uninstalling FlowForce Server

To uninstall FlowForce Server:

1. Stop the FlowForce Server service.

sudo launchctl unload /Library/LaunchDaemons/com.altova.FlowForceServer.plist

2. Stop the FlowForce Web Server service.

sudo launchctl unload /Library/LaunchDaemons/
com.altova.FlowForceWebServer.plist

To check whether a service has been stopped, open the Activity Monitor terminal and make sure that the service is not in the list.

- 3. In **Applications**, right-click the FlowForce Server icon and select **Move to Trash**. Note that your FlowForce Server configuration data is stored separately and is not removed at this stage (see next step).
- 4. Optionally, if you want to remove the FlowForce Server configuration data as well, run the following command:

sudo rm -rf /usr/local/Altova/FlowForceServer2017/

**Warning:** When you take this action, the FlowForce Server configuration data is deleted permanently.

# 3.3.4 Installation Directory

On Mac OS X, the location of the FlowForce Server installation directory is as follows:

/usr/local/Altova/FlowForceServer2017

# 3.4 Windows

This section includes installation tasks applicable to FlowForce Server running on Windows, as follows:

- Installing FlowForce Server
- Migrating Data Manually
- Installation Folder

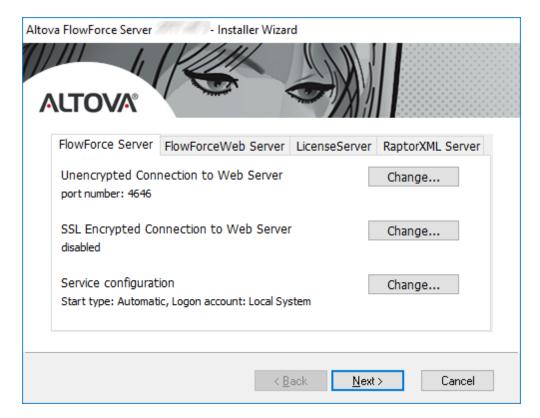
# 3.4.1 Installing FlowForce Server

# **Prerequisites**

Perform installation as a user with administrative privileges.

#### To install FlowForce Server:

- 1. Download the installation file from the Altova website (<a href="http://www.altova.com/download-trial-server.html">http://www.altova.com/download-trial-server.html</a>) and run it.
- 2. Optionally, select the installation language from the box in the lower left area of the wizard, and click **Next**. The language you select here also determines the language of the FlowForce Server user interface in the Web browser. If necessary, you can change the language later from the command line (see Command Line Interface).
- 3. Do one of the following:
  - a. If you haven't installed Altova LicenseServer yet, leave the default settings as is. The wizard will install the latest version of Altova LicenseServer on the computer where you are running the wizard.
  - b. If you haven't installed Altova LicenseServer yet and want to install Altova LicenseServer on another computer, click to clear the Install Altova LicenseServer on this machine check box, and then choose Register Later. In this case, you will need to install Altova LicenseServer and register FlowForce Server separately.
  - c. If Altova LicenseServer has already been installed on your computer but it has a lower version than the one indicated by the installation wizard, leave the default settings as is. In this case, the installation wizard will automatically upgrade your LicenseServer version with the one indicated on the dialog box. Note that the existing registration and licensing information will be preserved after the upgrade.
  - d. If Altova LicenseServer has already been installed on your computer or network, and if it has the same version as the one indicated by the wizard, do the following:
    - i. Click to clear the Install Altova LicenseServer on this machine check box.
    - ii. Under **Register this product with**, choose the Altova LicenseServer instance on which you want to register FlowForce Server, or choose **Register later**.
- 4. Click Next.
- 5. Optionally, select any additional Altova server products that you want to install.
- 6. Optionally, define the network settings and Windows service configuration applicable to FlowForce Server and FlowForce Web Server. For more information, see <a href="Network Configuration">Network Configuration (Windows Only)</a> and <a href="Windows Service Configuration">Windows Service Configuration</a>. From the same dialog box, you can optionally define the network and Windows service settings for other Altova products that are being installed at the same time with FlowForce, by clicking their respective tab. If you want to define these settings after installation, click Next.



7. Follow the wizard instructions to complete the installation. When the installation completes, the Altova ServiceController icon ( ) is added in the system notification area.

#### **Post-installation tasks**

- 1. If you haven't done so already during installation, register FlowForce Server with Altova LicenseServer (see Register FlowForce Server).
- 2. If you haven't done so already during installation, configure the network settings (see Defining the Network Settings).
- 3. Log on to the Web administration interface (by default, <a href="http://localhost:8082">http://localhost:8082</a>, unless you changed the address and port) and change the default FlowForce Server root password. The default login name and password is root and root, respectively.

# 3.4.2 Windows Service Configuration

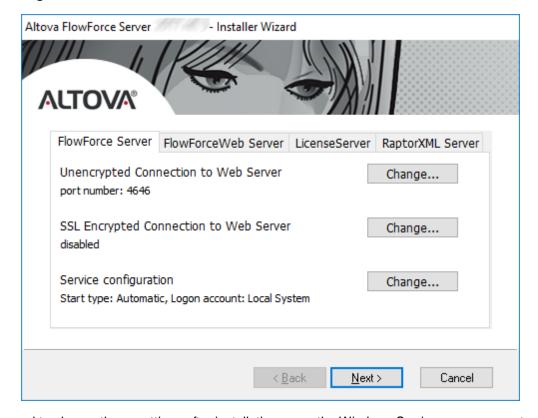
During the installation of FlowForce Server on a Windows operating system, it is possible to configure the following settings:

- How should FlowForce Server start up as a Windows service: automatic, on demand, delayed automatic, or disabled.
- The operating system user account to be used by FlowForce Server as a Windows service: Local System, Local Service, Network Service, or Other User.
- If FlowForce Server is configured to run as Other User, you can set the username and
  password of this user, similar to how this is done in the Windows Services management
  console.

**Note:** The settings listed above can also be configured for FlowForce Web Server.

The settings have the same meaning as when you change the service properties from the Windows Services management console.

To define the settings above during installation, click the relevant tab (**FlowForce Server** or **FlowForce Web Server**) on the installation dialog box, and, next to **Service configuration**, click **Change**.

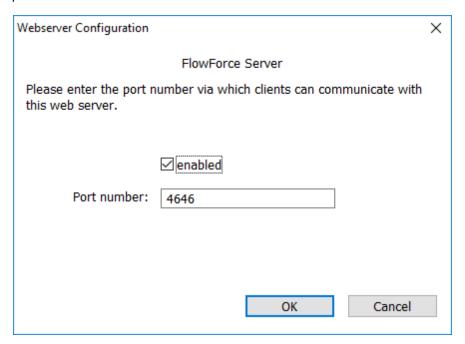


If you need to change these settings after installation, open the Windows Services management console, and then change the properties of the required service from there. To open the Windows Services management console, open a command line window and type Services.msc.

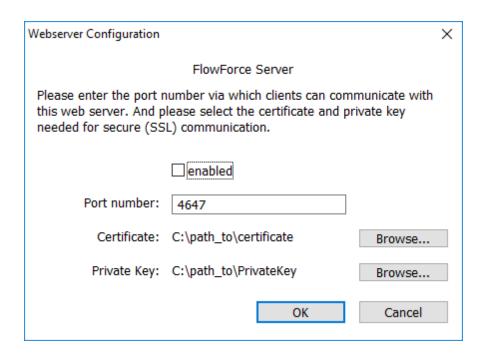
# 3.4.3 Network Configuration (Windows Only)

During the installation of FlowForce Server on a Windows operating system, it is possible to configure the following settings:

 Whether unencrypted connections to FlowForce Server are allowed, and, if yes, on which port.

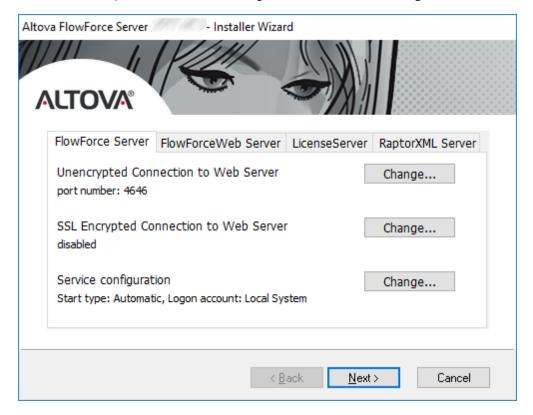


Whether secure (SSL-encrypted) connections to FlowForce Server are allowed, and, if
yes, on which port. By default, secure connections are disabled. Note that, in order to
enable SSL-encrypted connections, the paths to a certificate and a private key file must
also be specified. The host (bind address) can be specified from the Setup page after
installation (see <u>Defining the Network Settings</u>).



**Note:** The settings listed above can also be configured for FlowForce Web Server.

To define all of the settings above during installation, click the relevant tab (**FlowForce Server** or **FlowForce Web Server**) on the installation dialog box, and then click **Change**.



If you prefer to configure the settings above at a later time, click Next. In this case, FlowForce

Server will be installed with the default options as shown under each respective tab of the dialog box. After installation, you can change the settings as described in Defining the Network Settings.

# 3.4.4 Migrating Data Manually

On Windows, you do not typically need to migrate configuration data manually. When you install a new major version of FlowForce Server and a previous major version is already installed, the installation wizard prompts you to migrate the configuration data.

Should you need to migrate configuration data manually, follow the instructions below.

#### To migrate configuration data manually:

- 1. Ensure that Altova ServiceController ( ) is running in the system notification area. Otherwise, start the Altova ServiceController.
- 2. Stop the FlowForce Server service and the FlowForce Web Server service.
- 3. Delete the FlowForce Server data folder installed by the 2017 installation wizard. The path to the data folder depends on your Windows version (see <a href="HowForce Server Stores Configuration Data">How FlowForce Server Stores Configuration Data</a>).
- 4. At the command prompt, run the FlowForce executable with the migratedb command. Make sure to specify the correct path to the FlowForce Server .exe file and the configuration data folders applicable to your Windows version (see <a href="HowForce Server Stores Configuration Data">How FlowForce Server Stores Configuration Data</a>). For example, for FlowForce Server 32-bit running on 64-bit Windows 8, the command is:

```
"C:\Program Files(x86)\Altova\FlowForceServer2017\bin \FlowForceServer.exe" migratedb --datadir=C:\ProgramData\Altova\FlowForceServer2017\data -- olddatadir=C:\ProgramData\Altova\FlowForceServer2015\data
```

5. Start the FlowForce Server Web and the FlowForce Server services.

# 3.4.5 Installation Folder

On Windows, the default installation path of FlowForce Server is as follows.

Windows Vista, Windows 7, 8, 10	C:\Program Files\Altova\
32 bit Version on 64-bit OS	C:\Program Files (x86)\Altova\

# 3.5 Migrating Altova Servers

This topic provides instructions for migrating Altova servers from one server machine to another (including across any of the various supported platforms). These instructions use the term "old server" or "old machine" to refer to the machine from which you are migrating data, and the term "new server" or "new machine" to refer to the machine to which you are migrating data.

#### Altova LicenseServer

To migrate Altova LicenseServer from one server machine to another with the least possible downtime, follow the steps below:

- 1. Install Altova LicenseServer on the new machine.
- 2. Stop the Altova LicenseServer service on the new machine and copy the licenseserver.db database file from the old machine to the new machine. Make sure to back up the licenseserver.db on the new server in case you want to revert to it for any reason. The licenseserver.db file can be found in the LicenseServer application directory and its path will vary depending on the operating system:

Windows Server 2008, Windows Server 2012, Windows Vista, Windows 7, Windows 8, Windows 10	C:\ProgramData\Altova\LicenseServer
Linux	/var/opt/Altova/LicenseServer
Mac OS X	/var/Altova/LicenseServer

- 3. Stop the Altova LicenseServer on the old machine and start the Altova LicenseServer service on the new machine (for step-by-step instructions, see <a href="Start LicenseServer">Start LicenseServer</a>). This ensures that the two instances of LicenseServer are not running simultaneously.
- 4. Register all Altova server products with the new Altova LicenseServer. If the number of licenses to be migrated is small, you can do this in the standard way, as described in <u>Register Products</u>. If you have a large number of licenses to migrate, it is possible to automate registration only by custom scripts or code, as follows:
  - Write a command-line script which calls the "register" command for each server product that is under Altova LicenseServer control.
  - Write a program that can open the \*.licsvr file of each Altova server product
    programmatically and modify the "host" attribute so that it points to the new Altova
    LicenseServer. You can find the .licsvr file in the application directory of the respective
    server product (the path varies by operating system, as shown in the table above).

#### Altova FlowForce Server

Preparing for migration:

- 1. Ensure you have administrative privileges both on the old and the new server machine.
- 2. It is recommended that all jobs should use globally (not locally) stored credentials. If you are using locally stored (inline) credentials in jobs, all such jobs will have to be edited on the new server machine to match the user credentials linked to that operating system. By contrast, if you are using globally stored (standalone) credentials, you will only need to

- edit the standalone credentials on the new server machine. For more information, see How Credentials Work.
- 3. If your FlowForce Server uses Windows Active Directory integration, then, for a smooth migration, ensure that the new server machine has the same operating system users and configuration as the old machine.
- 4. Delete any FlowForce files (jobs, MapForce mappings, StyleVision transformations) that are not in use. In addition to this, perform a general FlowForce Server cleanup and maintenance by running the built-in functions "archive-log", "cleanup-files", and "truncate-log". The size of the internal FlowForce database can be reduced by running the compactdb command. This will help you significantly reduce the time required to copy the configuration data from the old server to the new server (as described in the steps below).

### Performing the migration:

On the new server machine, install FlowForce Server. To minimize server downtime during
migration, at this time you may also want to register FlowForce Server with Altova
LicenseServer, without transferring yet the license to the new server machine; this will be
done in a subsequent step. For step-by-step instructions on how to install FlowForce
Server, see Installation.

If you are migrating MapForce Server, StyleVision Server, and RaptorXML(XBRL) Server as well, it is recommended to install these products as part of the FlowForce Server installation.

2. On the old server machine, ensure that there are no running jobs, and stop both the FlowForce Server and the FlowForce Web Server services. For instructions on how to start/stop services on each operating system, see:

Linux: Starting and Stopping Services (Linux)
Mac OS X: Starting and Stopping Services (Mac OS X)
Windows: Starting and Stopping Services (Windows)

3. On the old server, locate the data directory of FlowForce Server (see FlowForce Server Application Data), and then copy it to the new server machine, with a temporary name, for example "C:\transfer\migrated\_data". The data directory includes any deployed jobs, MapForce mappings and StyleVision transformations, as well as FlowForce users, system functions, settings, and logs (for a complete list, see FlowForce Server Application Data). The duration of this operation will depend on the amount of you FlowForce configuration data.

The copying time will be greatly reduced provided that you cleaned up the FlowForce Server logs in advance (as explained above, see "Preparing for migration").

- 4. On the new server machine, stop both the FlowForce Server and FlowForce Web Server services, and then rename the data directory to data\_backup or a similar name. This will serve as a backup of the initial FlowForce Server configuration on the new server, in case you will want to revert to it for any reason.
- 5. From the command-line interface, change the current directory to the FlowForce program directory, and then run the "migratedb" command, for example:

FlowForceServer migratedb --datadir=C:\ProgramData\Altova \FlowForceServer2017\data --olddatadir=C:\transfer\migrated\_data

- On UNIX systems, use lowercase flowforceserver to call the executable, and adjust the paths accordingly (see <u>FlowForce Server Application Data</u>). You may also need to prepend the "./" characters to flowforceserver, to indicate the current directory.
- If the source **data** directory is of a major version of FlowForce Server that is older than the FlowForce Server version to which you are migrating, the configuration data will be upgraded automatically to the new FlowForce Server version. The same result can be achieved by running manually the "upgradedb" command from the command-line interface.
- 6. On the new server machine, start (in this order) the FlowForce Server and the FlowForce Web Server services.
- 7. If you haven't done that already, register FlowForce Server with Altova LicenseServer. For step-by-step instructions, see Register FlowForce Server.
- 8. From the Altova LicenseServer web interface, deactivate your FlowForce Server license from the old server and re-assign it to the new server machine (see <u>Assign Licenses to Registered Products</u>).

## Altova MobileTogether Server

Use the Backup and Restore functionality, as described in the MobileTogether Server documentation.

## **Altova MapForce Server**

For MapForce Server, migration to a new machine consists of re-assigning the license from the old machine to the new machine, as follows:

- 1. Install MapForce Server on the new machine. If it has already been installed as part of FlowForce Server installation, ignore this step.
- 2. On the new machine, register MapForce Server with Altova LicenseServer.
- 3. On the old machine, make sure no clients are using the server (for example, no mappings are running).
- Open the Altova LicenseServer administration page, and deactivate the license from the old MapForce Server machine and re-assign it to the new machine (see <u>Assign Licenses</u> to <u>Registered Products</u>).

#### Altova RaptorXML(XBRL) Server

For RaptorXML(XBRL) Server, migration to a new machine consists of re-assigning the license from the old machine to the new machine, as follows:

1. Install RaptorXML(XBRL) Server on the new machine. If it has already been installed as part of FlowForce Server installation, ignore this step.

- 2. On the new machine, register RaptorXML(XBRL) Server with Altova LicenseServer.
- 3. On the old machine, make sure no clients are using the server (for example, no transformations are running).
- 4. Open the Altova LicenseServer administration page, and deactivate the license from the old RaptorXML(XBRL) Server machine and re-assign it to the new machine (see <a href="Assign Licenses">Assign Licenses</a> to Registered Products ).

## **Altova StyleVision Server**

For StyleVision Server, migration to a new machine consists of re-assigning the license from the old machine to the new machine, as follows:

- 1. Install StyleVision Server on the new machine. If it has already been installed as part of FlowForce Server installation, ignore this step.
- 2. On the new machine, register StyleVision Server with Altova LicenseServer.
- 3. On the old machine, make sure no clients are using the server (for example, no transformations are running).
- Open the Altova LicenseServer administration page, and deactivate the license from the old StyleVision Server machine and re-assign it to the new machine (see <u>Assign</u> <u>Licenses to Registered Products</u>).

# Chapter 4

Licensing

# 4 Licensing

This section provides instructions on how to license FlowForce Server. FlowForce Server is licensed in the same way as other Altova server products, through <u>Altova LicenseServer</u>.

## 4.1 Altova LicenseServer

**Altova LicenseServer** (hereafter also called **LicenseServer** for short) provides a central location for the management of licenses for Altova products. Altova applications running in a network can have licenses assigned to them from the LicenseServer, thus giving administrators the flexibility to manage and monitor licenses.

Current version: 2.3

# **Licensing process with Altova LicenseServer**

To assign a license to an Altova product via Altova LicenseServer, you need to do the following:

- 1. Start LicenseServer
- 2. Open the <u>LicenseServer Configuration page</u>, which is the Web UI of LicenseServer, on Windows, Linux, or Mac OS X.
- 3. <u>Upload the Altova product license/s</u> you have received from Altova to the license pool of LicenseServer. Do this in the License Pool tab of the Configuration page.
- 4. Register the Altova product/s with LicenseServer.
- 5. In the <u>Client Management</u> tab of the Configuration page, <u>assign licenses</u> to the Altova product/s.

Licenses can thereafter be conveniently monitored and managed centrally with LicenseServer. See the Configuration Page Reference for available functionality.

**Note:** The LicenseServer Configuration page does not support SSL.

#### LicenseServer versions and their compatibility with Altova products

New versions of Altova products can only be licensed with the version of LicenseServer that is the latest at the time of the Altova product's release. However, older versions of Altova products will work with newer versions of LicenseServer.

So, if you are installing a new version of an Altova product and if your current LicenseServer version is not the latest, de-install this older LicenseServer version and install the latest version that is available on the Altova website. All registration and licensing information held in your older version of LicenseServer will be saved at the time of de-installation to a database on your machine, and will be imported automatically into the newer version of LicenseServer. When you install a newer version of LicenseServer, the older version will be de-installed before the newer version is installed.

The version number of the currently installed LicenseServer is given at the bottom of the LicenseServer configuration page (all tabs).

Current version: 2.3

#### **About this documentation**

This documentation is organized into the following parts:

• Introductory information about: <u>network requirements</u>; installation on <u>Windows</u>, <u>Linux</u>, and <u>Mac OS X</u>; and <u>Altova ServiceController</u>.

- How to Assign Licenses, which describes in a step-by-step way how to assign licenses with Altova LicenseServer.
- <u>Configuration Page Reference</u>: A description of the administrator's interface with LicenseServer.

Last updated: 21 April 2017

#### 4.1.1 Network Information

Altova LicenseServer must be installed on a server machine that is accessible by all clients running Altova products that require a license. Any firewall on both the client and server must allow the flow of network traffic to and from the LicenseServer that is necessary for the LicenseServer to operate correctly.

On the LicenseServer machine, **port 35355** is used to distribute licenses, and therefore it must be open for network traffic with client machines.

The following are the default networking parameters and requirements of LicenseServer:

For LicenseServer license distribution:
 Either one or both of
 IPv4 TCP connection on port 35355
 IPv6 TCP connection on port 35355

For administrative tasks, the LicenseServer is accessed by a web interface that uses port 8088. The port used can be configured to suit your requirements.

## Connection to the Master Licensing Server at altova.com

Altova LicenseServer needs to be able to communicate with the Master Licensing Server at altova.com to validate and authenticate license-related data and to ensure continuous compliance with the Altova license agreements. This communication occurs over HTTPS using port 443. If Altova LicenseServer, after making the initial verification with the altova.com Master Licensing Server, is unable to again connect with altova.com for a duration of more than 5 days (= 120 hours), then Altova LicenseServer will no longer permit the usage of any Altova software products connected to that Altova LicenseServer.

Any such loss of connection with the altova.com master servers will be logged in the Messages tab of the Configuration page of Altova LicenseServer. In addition, the administrator can configure the Altova LicenseServer to automatically send an alert email when the connection to altova.com is lost. Alert Mail settings are available in the Settings tab of the Configuration page.

# 4.1.2 Installation (Windows)

Altova LicenseServer can be installed on Windows systems in one of two ways:

- As an independent installation.
- As part of an Altova server product installation. (Altova server products are: Altova
  FlowForce Server, Altova MapForce Server, Altova StyleVision Server, Altova
  RaptorXML(+XBRL), and Altova MobileTogether Server.) If LicenseServer is not installed
  on your system at the time an Altova server product is installed, the option to install
  LicenseServer is selected by default during installation setup. If LicenseServer is already
  installed, the option to install it is deselected by default. You can change the default
  option if you like.

For information about assigning licenses via LicenseServer, see the section <u>How to Assign</u> <u>Licenses</u>.

## System requirements

▼ Windows

Windows Vista, Windows 7/8/10

▼ Windows Server

Windows Server 2008 R2 or newer

# ▼ LicenseServer versions and their compatibility with Altova products

New versions of Altova products can only be licensed with the version of LicenseServer that is the latest at the time of the Altova product's release. However, older versions of Altova products will work with newer versions of LicenseServer.

So, if you are installing a new version of an Altova product and if your current LicenseServer version is not the latest, de-install this older LicenseServer version and install the latest version that is available on the Altova website. All registration and licensing information held in your older version of LicenseServer will be saved at the time of de-installation to a database on your machine, and will be imported automatically into the newer version of LicenseServer. When you install a newer version of LicenseServer, the older version will be de-installed before the newer version is installed.

The version number of the currently installed LicenseServer is given at the bottom of the LicenseServer configuration page (all tabs).

Current version: 2.3

The version number of the LicenseServer that is appropriate for any particular version of a server product is displayed during the installation of that version of the server product. You can choose to install this version of LicenseServer along with the server product, or you can install the newer

version of LicenseServer separately. In both cases, the installer will automatically de-install the previous version and install the new version.

# 4.1.3 Installation (Linux)

Altova LicenseServer can be installed on Linux systems (Debian, Ubuntu, CentOS, RedHat).

#### **System requirements**

- ▼ Linux
  - CentOS 6 or newer
  - RedHat 6 or newer
  - Debian 7 or newer
  - Ubuntu 12.04 or newer

The following libraries are required as a prerequisite to install and run the application. If the packages below are not already available on your Linux machine, run the command yum (or apt-get if applicable) to install them.

Required by	CentOS, RedHat	Debian	Ubuntu
FlowForce Server	libidn, krb5-libs	libidn11, libgssapi- krb5-2	libidn11, libgssapi- krb5-2
LicenseServer	krb5-libs	libgssapi-krb5-2	libgssapi-krb5-2

## **Uninstalling old versions of LicenseServer**

On the Linux command line interface (CLI), you can check whether LicenseServer is installed with the following command:

```
[Debian, Ubuntu]: dpkg --list | grep Altova [CentOS, RedHat]: rpm -qa | grep server
```

If LicenseServer is not installed, go ahead with the installation as documented in the next steps. If LicenseServer is installed and you wish to install a newer version of it, uninstall the old version with the command:

```
[Debian, Ubuntu]: sudo dpkg --remove licenseserver [CentOS, RedHat]: sudo rpm -e licenseserver
```

#### **Installing Altova LicenseServer**

On Linux systems, LicenseServer must be installed independently of other Altova server products. It is not included as part of the installation packages of Altova server products. Download Altova LicenseServer from the <u>Altova website</u> and copy the package to any directory on the Linux system.

Distribution	Installer extension
Debian	.deb
Ubuntu	.deb
CentOS	.rpm
RedHat	.rpm

In a terminal window, switch to the directory where you have copied the Linux package. For example, if you copied it to a user directory called MyAltova (that is located, say, in the /home/User directory), then switch to this directory as follows:

```
cd /home/User/MyAltova
```

Install LicenseServer with the following command:

```
[Debian]: sudo dpkg --install licenseserver-2.3-debian.deb

[Ubuntu]: sudo dpkg --install licenseserver-2.3-ubuntu.deb

[CentOS]: sudo rpm -ivh licenseserver-2.3-1.x86_64.rpm

[RedHat]: sudo rpm -ivh licenseserver-2.3-1.x86_64.rpm
```

The LicenseServer package will be installed in:

```
/opt/Altova/LicenseServer
```

For information about how to proceed with assigning licenses, see the section <u>How to Assign Licenses</u>.

#### ▼ LicenseServer versions and their compatibility with Altova products

New versions of Altova products can only be licensed with the version of LicenseServer that is the latest at the time of the Altova product's release. However, older versions of Altova products will work with newer versions of LicenseServer.

So, if you are installing a new version of an Altova product and if your current LicenseServer version is not the latest, de-install this older LicenseServer version and install the latest version that is available on the Altova website. All registration and licensing information held in your older version of LicenseServer will be saved at the time of de-installation to a database on your machine, and will be imported automatically into the newer version of LicenseServer. When you install a newer version of LicenseServer, the older version will be de-installed before the newer version is installed.

The version number of the currently installed LicenseServer is given at the bottom of the <u>LicenseServer configuration page</u> (all tabs).

Current version: 2.3

# 4.1.4 Installation (Mac OS X)

Altova LicenseServer can be installed on Mac OS X systems (version 10.8 or higher). Since you might need to uninstall a previous version, uninstalling is described first.

#### **System requirements**

▼ Mac OS X

Mac OS X 10.10, 10.11, 10.12 or newer

#### **Uninstalling old versions of LicenseServer**

Before uninstalling LicenseServer, stop the service with the following command:

sudo launchctl unload /Library/LaunchDaemons/com.altova.LicenseServer.plist

To check whether the service has been stopped, open the Activity Monitor terminal and make sure that LicenseServer is not in the list.

In **Applications**, right-click the LicenseServer icon and select **Move to Trash**. The application will be moved to Trash. You will, however, still need to remove the application from the usr folder. Do this with the command:

sudo rm -rf /usr/local/Altova/LicenseServer

#### **Installing Altova LicenseServer**

Open the download page <a href="http://www.altova.com/download.html">http://www.altova.com/download.html</a> and locate the Altova LicenseServer installer under "Server Software Products" for Mac. After downloading the disk image (.dmg) file, click to open it. This mounts a new virtual drive on your computer. On the virtual drive, double-click the package (.pkg) file, and follow the on-screen instructions. You will need to accept the license agreement for installation to proceed.

The LicenseServer package will be installed in the folder:

/usr/local/Altova/LicenseServer

To eject the virtual drive after installation, right-click it, and select **Eject**.

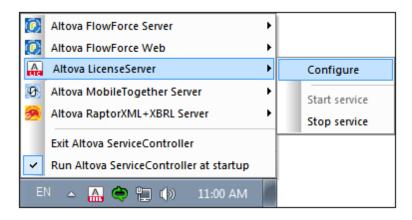
## 4.1.5 Altova ServiceController

Altova ServiceController (ServiceController for short) is an application for conveniently starting, stopping and configuring Altova services **on Windows systems**.

ServiceController is installed with Altova LicenseServer and with <u>Altova server products that are installed as services</u> (FlowForce Server, RaptorXML(+XBRL) Server, and Mobile Together Server). It can be started by clicking **Start | Altova LicenseServer | Altova ServiceController**. (This command is also available in the **Start** menu folders of <u>Altova server products that are installed as services</u> (FlowForce Server, RaptorXML(+XBRL) Server, and Mobile Together Server).) After ServiceController has been started, it can be accessed via the system tray (*screenshot below*).



To specify that ServiceController starts automatically on logging in to the system, click the **ServiceController** icon in the system tray to display the **ServiceController** menu (*screenshot below*), and then toggle on the command **Run Altova ServiceController at Startup**. (This command is toggled on by default.) To exit ServiceController, click the **ServiceController** icon in the system tray and, in the menu that appears (*see screenshot below*), click **Exit Altova ServiceController**.



#### Starting and stopping Altova services

Each installed Altova service component will have an entry in the ServiceController menu (see screenshot above). An Altova service can be started or stopped via a command in its ServiceController sub-menu. Additionally, important administration tasks of individual services can be accessed via the ServiceController menu. In the screenshot above, for example, Altova LicenseServer service has a sub-menu in which you can choose to access LicenseServer's Configuration page via the **Configure** command.

# 4.1.6 How to Assign Licenses

To assign a license to an Altova product via Altova LicenseServer, do the following:

- 1. Start LicenseServer
- 2. Open the <u>LicenseServer Configuration page</u>, which is the administrator's interface with LicenseServer, on Windows, Linux, or Mac OS X.
- 3. <u>Upload the license/s</u> you have received from Altova to the license pool of your Altova LicenseServer. Do this in the <u>License Pool</u> tab of the LicenseServer Configuration page.
- 4. Register the Altova product with LicenseServer. Depending on the product's type, the method of registering it with LicenseServer will be different: respectively, via the product's Software Activation dialog, Web UI, or command line. See the documentation of your Altova product for additional information.
- 5. In the <u>Client Management</u> tab of the <u>LicenseServer Configuration page</u>, <u>assign a license</u> to the Altova product.

#### Note on cores and licenses

The licensing of Altova server products is based on the number of processor cores available on the product machine. For example, a dual-core processor has two cores, a quad-core processor four cores, a hexa-core processor six cores, and so on. The number of cores licensed for a product must be greater than or equal to the number of cores available on that server machine, whether the server is a physical or virtual machine. For example, if a server has eight cores (an octa-core processor), you must purchase at least one 8-core license. You can also combine licenses to achieve the core count. So, two 4-core licenses can also be used for an octa-core server instead of one 8-core license.

If you are using a computer server with a large number of CPU cores but only have a low volume to process, you may also create a virtual machine that is allocated a smaller number of cores, and purchase a license for that number. Such a deployment, of course, will have less processing speed than if all available cores on the server were utilized.

**Note:** Each Altova server product license can be used for only one client machine—the machine on which the Altova server product is installed—at a time, even if the license has unused licensing capacity. For example, if a 10-core license is used for a client machine that has 6 CPU cores, then the remaining 4 cores of licensing capacity cannot be used simultaneously for another client machine.

#### MobileTogether Server licenses

MobileTogether Server licenses are based on the number of CPU cores on the MobileTogether Server machine. Core licenses allow an unlimited number of MobileTogether Client devices to connect to the server. However, if you check the *Limit to single thread execution* check box, then only one mobile device will be able to connect to the MobileTogether Server at any time. This is useful for evaluation and small-scale testing. Note that, if, in this case, a second device connects to MobileTogether Sever, then it will take over the license. The first device will not be able to connect any more and will receive an error message to this effect.

#### Start LicenseServer

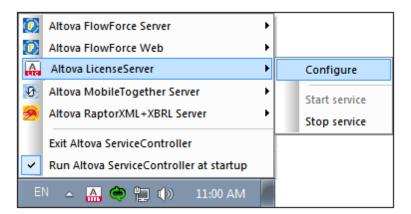
#### This section:

- How to start LicenseServer on Windows systems
- How to start LicenseServer on Linux systems
- How to start LicenseServer on Mac OS X systems
- Note about Connection to altova.com

#### Windows systems

You can start LicenseServer via the Altova ServiceController, which is available in the system tray.

First, click Start | All Programs | Altova LicenseServer | Altova ServiceController to start Altova ServiceController and display its icon in the system tray (see screenshot below). If you select the Run Altova ServiceController at Startup option, Altova ServiceController will start up on system start and its icon will be available in the system tray from then onwards.



To start LicenseServer, click the Altova ServiceController icon in the system tray, hover over **Altova LicenseServer** in the menu that pops up (see screenshot below), and then select **Start Service** from the LicenseServer submenu. If LicenseServer is already running, the *Start Service* option will be disabled.

#### **Linux systems**

To start LicenseServer as a service on Linux systems, run the following command in a terminal window.

```
[Debian 7]: sudo /etc/init.d/licenseserver start [Debian 8]: sudo systemctl start licenseserver [Ubuntu <=14]: sudo initctl start licenseserver [Ubuntu 15]: sudo systemctl start licenseserver [CentOS 6]: sudo initctl start licenseserver [CentOS 7]: sudo systemctl start licenseserver [RedHat]: sudo initctl start licenseserver
```

(If you need to stop LicenseServer, replace start with stop in the above command.)

#### Mac OS X systems

To start LicenseServer as a service on Mac OS X systems, run the following command in a terminal window:

sudo launchctl load /Library/LaunchDaemons/com.altova.LicenseServer.plist

If at any time you need to stop LicenseServer, use:

sudo launchctl unload /Library/LaunchDaemons/com.altova.LicenseServer.plist

## **Connection to the Master Licensing Server at altova.com**

Altova LicenseServer needs to be able to communicate with the Master Licensing Server at altova.com to validate and authenticate license-related data and to ensure continuous compliance with the Altova license agreements. This communication occurs over HTTPS using port 443. If Altova LicenseServer, after making the initial verification with the altova.com Master Licensing Server, is unable to again connect with altova.com for a duration of more than 5 days (= 120 hours), then Altova LicenseServer will no longer permit the usage of any Altova software products connected to that Altova LicenseServer.

Any such loss of connection with the altova.com master servers will be logged in the Messages tab of the Configuration page of Altova LicenseServer. In addition, the administrator can configure the Altova LicenseServer to automatically send an alert email when the connection to altova.com is lost. Alert Mail settings are available in the Settings tab of the Configuration page.

#### Open LicenseServer's Config Page (Windows)

#### This section:

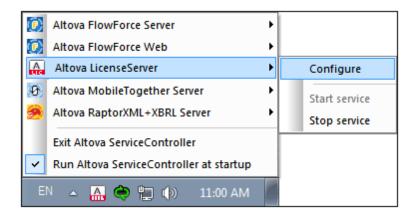
- Opening the Configuration page if LicenseServer is on the same machine
- Opening the Configuration page if LicenseServer is on another machine
- Logging in with the initial password
- Setting a fixed port for the Configuration page

Opening the Configuration page if LicenseServer is on the same machine
On Windows systems, if LicenseServer is on the same machine, you can open the Configuration page of LicenseServer in one of two ways:

• Click Start | All Programs | Altova LicenseServer | LicenseServer Configuration

Page. The Configuration page opens in a new tab of your Internet browser.

Click the Altova ServiceController icon in the system tray, mouse over Altova
 LicenseServer in the menu that pops up (see screenshot below), and then select
 Configure from the LicenseServer submenu.



The <u>Configuration page</u> opens in a new browser window, and its login mask is displayed (screenshot below).

## Opening the Configuration page if LicenseServer is on another machine

To open the LicenseServer <u>Configuration page</u> from some other Windows machine on the local network (than that on which LicenseServer is installed), enter the URL of the LicenseServer <u>Configuration page</u> in the address bar of a browser and press **Enter**. By default, the URL of the Configuration page will be:

http://<serverIPAddressOrName>:8088/

The URL is present in the HTML code of the Configuration page itself, which is named webul.html and is located at:

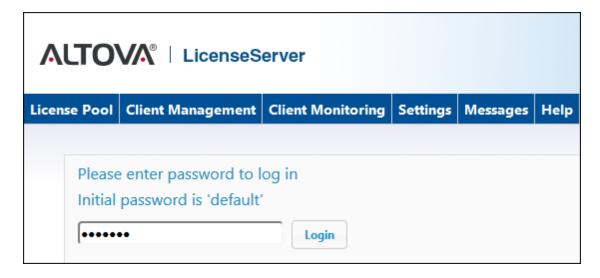
C:/ProgramData/Altova/LicenseServer/WebUI.html

If you have <u>set the URL of the Configuration page</u> to be generated dynamically (in the Settings tab of the Configuration page), then a new URL is generated each time LicenseServer is started. You will need to check the current version of webul.html to find out the current URL of the Configuration page.

The dynamically generated URL in webur.html will have a form something like: http://127.0.0.1:55541/optionally-an-additional-string, and it is located in the function checkIfServiceRunning() in a script near the end of the <head> element. While the port number in the URL is dynamically assigned, the IP address part identifies the server on which LicenseServer has been installed. If you wish to access the LicenseServer Configuration page from another machine, make sure that the IP address part of the URL has the correct IP address or name of the server on which LicenseServer has been installed. For example, the URL could be something like: http://someServer:55541.

#### Logging in with the initial password

After going through the steps above, the <u>Configuration page</u> is opened with the login mask displayed (*screenshot below*). You can log in with the initial password of <code>default</code>. After you have logged in, you can change your password in the <u>Settings</u> tab.



#### Setting a fixed or dynamic port for the Configuration page

The port of the Configuration page (Web UI)—and consequently its address—can be specified in the <u>Settings page</u>. By default the port is 8088. You can set any other port you want for the LicenseServer <u>Configuration page</u> (see screenshot below). Alternatively, you allow the port to be selected dynamically each time LicenseServer starts up. In this case, you will need to find out the URL of the Configuration page from the file WebuI.html (see <u>Open LicenseServer Config Page</u> (<u>Windows</u>), <u>Open LicenseServer Config Page</u> (<u>Linux</u>), and <u>Open LicenseServer's Config Page</u> (<u>Mac OS X</u>).



The advantage of a fixed port is that the page URL is known in advance and therefore can be accessed easily. If the port is assigned dynamically, the port part of the URL will have to be looked up in the file <code>WebUI.html</code> each time LicenseServer is started.

# Open LicenseServer's Config Page (Linux)

### This section:

- Opening the Configuration page for the first time with the returned URL
- URL of the LicenseServer Configuration page
- Logging in with the initial password
- Setting a fixed port for the Configuration page

## Opening the Configuration page for the first time with the returned URL

On Linux systems, when you register your Altova server product with LicenseServer via the CLI, the URL of the LicenseServer Configuration page is returned. On opening this URL in a browser, you are prompted to read and accept the license agreement. After accepting the license agreement, the Configuration page's login mask is displayed (*screenshot below*).

**Note:** Altova desktop products are available for Windows only.

# **URL** of the LicenseServer Configuration page

To open the LicenseServer Configuration page at any time, enter its URL in the address bar of a browser and press **Enter**. By default, the URL of the Configuration page will be:

http://<serverIPAddressOrName>:8088/

The URL is present in the HTML code of the Configuration page itself, which is named webul.html and is located at:

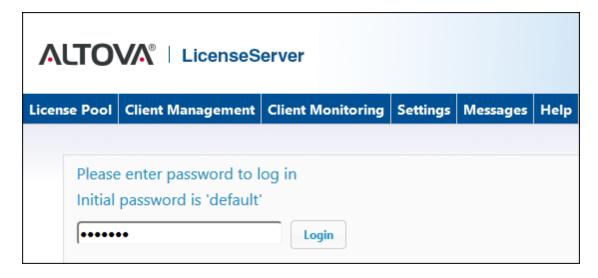
#### /var/opt/Altova/LicenseServer/webUI.html

If you have <u>set the URL of the Configuration page</u> to be generated dynamically (in the Settings tab of the Configuration page), then a new URL is generated each time LicenseServer is started. You will need to check the current version of webul.html to find out the current URL of the Configuration page.

The dynamically generated URL in webul.html will have a form something like: http://l27.0.0.1:55541, and it is located in the function checkIfServiceRunning() in a script near the end of the <head> element. While the port number in the URL is dynamically assigned, the IP address part identifies the server on which LicenseServer has been installed. If you wish to access the LicenseServer Configuration page from another machine, make sure that the IP address part of the URL has the correct IP address or name of the server on which LicenseServer has been installed. For example, the URL could be something like: http://myserver:55541.

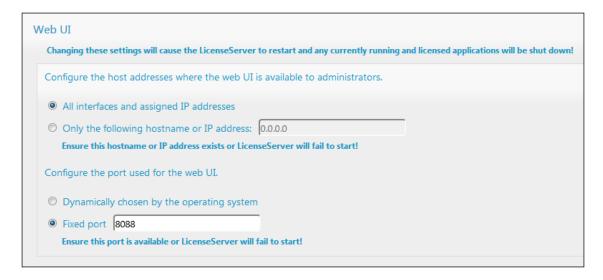
## Logging in with the initial password

After going through the steps above, the <u>Configuration page</u> is opened with the login mask displayed (*screenshot below*). You can log in with the initial password of <code>default</code>. After you have logged in, you can change your password in the <u>Settings</u> tab.



#### Setting a fixed or dynamic port for the Configuration page

The port of the Configuration page (Web UI)—and consequently its address—can be specified in the <u>Settings page</u>. By default the port is 8088. You can set any other port you want for the LicenseServer <u>Configuration page</u> (see screenshot below). Alternatively, you allow the port to be selected dynamically each time LicenseServer starts up. In this case, you will need to find out the URL of the Configuration page from the file <code>WebUI.html</code> (see <u>Open LicenseServer Config Page</u> (Windows), <u>Open LicenseServer Config Page</u> (Linux), and <u>Open LicenseServer's Config Page</u> (Mac OS X)).



The advantage of a fixed port is that the page URL is known in advance and therefore can be accessed easily. If the port is assigned dynamically, the port part of the URL will have to be looked up in the file <code>WebUI.html</code> each time LicenseServer is started.

## Open LicenseServer's Config Page (Mac OS X)

### This section:

- Opening the Configuration page for the first time with the returned URL
- URL of the LicenseServer Configuration page
- Logging in with the initial password
- Setting a fixed port for the Configuration page

### Opening the Configuration page for the first time with the returned URL

On Mac OS X systems, when you register your Altova server product with LicenseServer via the CLI, the URL of the LicenseServer Configuration page is returned. On opening this URL in a browser, you are prompted to read and accept the license agreement. After accepting the license agreement, the Configuration page's login mask is displayed (*screenshot below*).

**Note:** Altova desktop products are available for Windows only.

# **URL** of the LicenseServer Configuration page

To open the LicenseServer Configuration page at any time, enter its URL in the address bar of a browser and press **Enter**. By default, the URL of the Configuration page will be:

http://<serverIPAddressOrName>:8088/

The URL is present in the HTML code of the Configuration page itself, which is named webul.html and is located at:

#### /var/Altova/LicenseServer/webUI.html

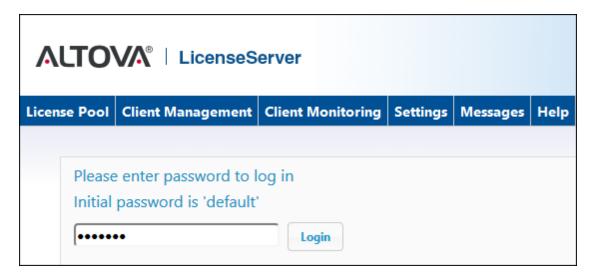
If you have <u>set the URL of the Configuration page</u> to be generated dynamically (in the Settings tab of the Configuration page), then a new URL is generated each time LicenseServer is started. You will need to check the current version of webul.html to find out the current URL of the Configuration page.

The dynamically generated URL in webur.html will have a form something like: http://l27.0.0.1:55541, and it is located in the function checkIfServiceRunning() in a script near the end of the <head> element. While the port number in the URL is dynamically assigned, the IP address part identifies the server on which LicenseServer has been installed. If you wish to access the LicenseServer Configuration page from another machine, make sure that the IP address part of the URL has the correct IP address or name of the server on which LicenseServer has been installed. For example, the URL could be something like: http://myserver:55541.

Note: The Configuration page can also be accessed directly via the Finder | Applications |
Altova License Server icon.

## Logging in with the initial password

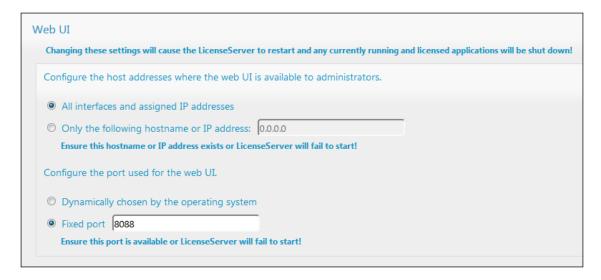
After going through the steps above, the <u>Configuration page</u> is opened with the login mask displayed (*screenshot below*). You can log in with the initial password of default. After you have logged in, you can change your password in the <u>Settings</u> tab.



## Setting a fixed or dynamic port for the Configuration page

The port of the Configuration page (Web UI)—and consequently its address—can be specified in the <u>Settings page</u>. By default the port is 8088. You can set any other port you want for the LicenseServer Configuration page (see screenshot below). Alternatively, you allow the port to be

selected dynamically each time LicenseServer starts up. In this case, you will need to find out the URL of the Configuration page from the file WebUI.html (see Open LicenseServer Config Page (Windows), Open LicenseServer Config Page (Linux), and Open LicenseServer's Config Page (Mac OS X)).



The advantage of a fixed port is that the page URL is known in advance and therefore can be accessed easily. If the port is assigned dynamically, the port part of the URL will have to be looked up in the file <code>WebUI.html</code> each time LicenseServer is started.

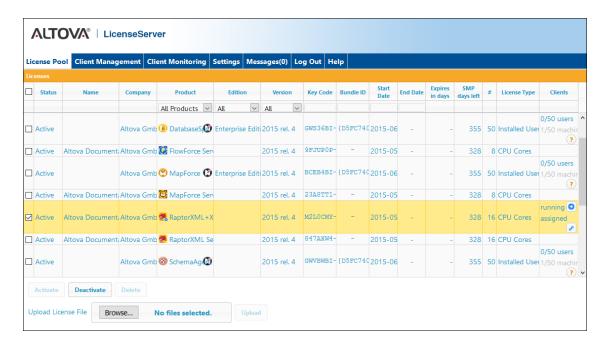
### Upload Licenses to LicenseServer

# This section:

- Uploading a license file to the license pool of LicenseServer
- License status
- Activating the licenses you wish to use
- Next steps

### Uploading a license file to the license pool of LicenseServer

After you have obtained a license file from Altova, you must upload it to the Altova LicenseServer. Each license file can contain one or more licenses and depends on your purchase. When you upload a license file, all the licenses in it will be uploaded to the license pool of LicenseServer and can be assigned to an Altova product that has been registered with that LicenseServer. All the uploaded licenses, from one or more license files and for all Altova products, are collected in a license pool on the LicenseServer. The license pool is displayed in the License Pool tab of the LicenseServer Configuration page (screenshot below).



License files are uploaded to the LicenseServer using the *Upload License File* function of the License Pool tab (*see screenshot above*). Click the **Browse** button and select the license file you want. The license file will appear in the *Upload License File* text field and the **Upload** button will be enabled. Click the **Upload** button to upload the license file. All the licenses in the file are uploaded and displayed in the License Pool tab. The screenshot above shows multiple licenses, uploaded from multiple license files.

### License status

License status values are as follows:

- Activating: When a license is uploaded to the license pool of LicenseServer, the server
  will transmit license-related data to the altova.com master licensing server to validate,
  authenticate, and activate the license that was supplied. This is necessary to ensure
  compliance with the Altova license agreements. During this initial activation and
  authentication transaction—which typically lasts between 30 seconds and a couple of
  minutes, depending on your Internet connection, speed, and overall network traffic—the
  status of the license will be indicated as Activating....
- Failed Verification: If a connection with the altova.com master licensing server cannot be
  made, then the status of the license in the pool will be shown as Failed Verification. If
  this happens, check your Internet connection and firewall rules to ensure that
  LicenseServer is able to communicate with the altova.com master licensing server.
- Active: Once the license has been authenticated and activated, the status will change to Active.
- *Inactive:* If a license has been verified, but is present on another LicenseServer on the network, the status will be shown as *Inactive*. An *Inactive* status also results when a license is manually deactivated in the license pool by the administrator.

Pending: If the start date of a license is a date in the future, then the license is shown as
 Pending. The status will change to Active on the start date. A Pending license can be
 assigned to a product and ensures that the licensing of a product continues uninterrupted
 before an existing license expires. (A product is allowed to have two active licenses
 assigned to it at a given time.)

• Blocked: A license is shown in the license pool as Blocked if there was a problem authenticating the license and the altova.com master licensing server has not granted permission to the LicenseServer to use this license. This could be the result of a license agreement violation, over-usage of a license, or other compliance issues. Should you see a license showing up as Blocked, please contact Altova Support with your license information and any other relevant data.

These statuses are summarized in the table below:

Status	Meaning
Activating	On upload, license information is sent to altova.com for verification. Refresh the browser to view the updated status. Verification and activation can take a few minutes.
Failed Verification	A connection to altova.com could not be made. After establishing a connection, either restart the service or activate the license (with the <b>Activate</b> button).
Active	Verification was successful, the license is active.
Inactive	Verification was successful, but the license is also on another LicenseServer on the network. Licenses can be made inactive with the <b>Deactivate</b> button.
Pending	A <i>Pending</i> license has a start date in the future, and it turns <i>Active</i> on that start date. It can be assigned to a product and ensures the automatic renewal of a product's licensing before an existing license expires.
Blocked	Verification was not successful. License is invalid and is blocked. Contact Altova Support.

**Note:** After a license has been sent to altova.com for verification, the browser must be refreshed to see the updated status. Verification and activation can take a few minutes.

**Note:** If a connection to altova.com could not be made, the status will be *Failed Verification*. After establishing a connection, either restart the service or try activating the license with the **Activate** button.

**Note:** When a license is given a status of *Inactive* or *Blocked*, a message explaining the status is also added to the Messages log.

Only an active or pending license can be assigned to a product installation. An inactive license can be activated or deleted from the license pool. If a license is deleted from the license pool, it can be uploaded again to the pool by uploading the license file containing it. When a license file is updated, only those licenses in it that are not already in the pool will be uploaded to the pool. To

activate, deactivate, or delete a license, select it and then click the **Activate**, **Deactivate**, or **Delete** button, respectively.

# Activate the license/s you wish to use

Before you can assign a license to an Altova product, it must be active. So do ensure it is active. If it is inactive, select it and click **Activate**.

# **Next Steps**

After you have uploaded the license file to the LicenseServer and checked that the license you want is active, do the following:

- 1. Register the Altova server product (<u>FlowForce Server</u>, <u>MapForce Server</u>, <u>StyleVision</u> <u>Server</u>) with LicenseServer. (If you have already done this prior to uploading the license file, you can now start assigning licenses.)
- 2. <u>Assign a license</u> to your Altova product that has been registered with the LicenseServer.

# Register Product/s

Before you can <u>assign a license</u> to an Altova product, you must register the product installation with LicenseServer. The registration is done from the Altova product, and the process differs according to the type of product.

- Desktop products: Registration is via the the product's Software Activation dialog.
- Server products that have Web Uls: Registration of FlowForce Server and MobileTogether Server is via the Web Ul's Setup tab or the product's CLI.
- Server products that have no Web UI: Registration of MapForceServer, RaptorXML(+XBRL) Server, and StyleVisionServer is via the CLI of these products. You will need the server name or IP Address of the machine on which LicenseServer is installed to carry out the registration.

This section describes how to register different Altova products:

- Register Altova Desktop Products
- Register FlowForce Server
- Register MapForce Server
- Register MobileTogether Server
- Register RaptorXML(+XBRL) Server
- Register StyleVision Server

Register Altova Desktop Products

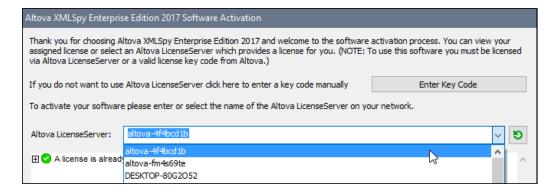
To register an Altova Desktop product with an Altova LicenseServer, do the following:

 Select the menu command Help | Software Activation to go to the product's Software Activation dialog. You can activate your software either (i) via Altova LicenseServer, or (ii) by entering your product's key code details. In this documentation we describe only licensing via Altova LicenseServer.

2. To license your Altova product via LicenseServer, click **Use Altova LicenseServer** (located at the bottom of the dialog; *see screenshot below*)



3. This switches the dialog to LicenseServer activation mode (*screenshot below*). In the *Altova LicenseServer* combo box, select a LicenseServer from the dropdown list.



When the connection to the selected LicenseServer is made, the product is immediately registered with the selected LicenseServer, and, in the <u>Client Management tab</u>, the product is displayed in that client machine's product list.

### Unregistering a desktop product

To unregister a desktop product, go to the <u>Client Management tab</u> of LicenseServer and, in the right-hand *Product licensing* pane, click the product's **Unregister Product** button.

Register FlowForce Server

# This section:

- Methods of registering FlowForce Server with LicenseServer
- Accessing the FlowForce Server Setup page (Windows)
- Accessing the FlowForce Server Setup page (Linux)
- Registering FlowForce Server via the Setup page
- Registering FlowForce Server via the FlowForce CLI (Windows)
- Registering FlowForce Server via the FlowForce CLI (Linux)
- Next steps

### **Methods of registering FlowForce Server**

FlowForce Server can be registered with LicenseServer using any of the following methods:

Via the FlowForce Server Setup page

- Via the FlowForce CLI (Windows)
- Via the FlowForce CLI (Linux)

# Accessing the FlowForce Server Setup page (Windows)

The FlowForce Server Setup page can be accessed in one of the following ways:

- Via the Start menu: Start | Altova FlowForce Server 2017 | FlowForce Server Setup Page
- Via <u>Altova ServiceController</u>: Click the ServiceController icon in the system tray. In the menu that pops up, select *Altova FlowForce Web | Setup*.

This pops up the FlowForce Server Setup page (screenshot above).

# Accessing the FlowForce Server Setup page (Linux)

After you have installed FlowForce Server on Linux (see the FlowForce Server user documentation for information about how to do this), start FlowForce Web Server as a service with the following command:

sudo /etc/init.d/flowforcewebserver start

A message containing the URL of the FlowForce Server Setup appears in the terminal window:

FlowForceWeb running on http://127.0.1.1:3459/setup?key=52239315203

Enter the URL in the address field of a browser and hit **Enter** to access the FlowForce Server Setup page (*screenshot below*).

# Registering FlowForce Server via the Setup page

In the Setup page (*screenshot below*)—how to access it is described above—the LicenseServer field specifies the Altova LicenseServer to be used for registration.



The LicenseServer can be specified in one of two ways.

You can search for Altova LicenseServers that are currently available on the network—that is, those that are currently running. Do this by clicking the Search for Altova LicenseServers button (highlighted yellow in the screenshot below).



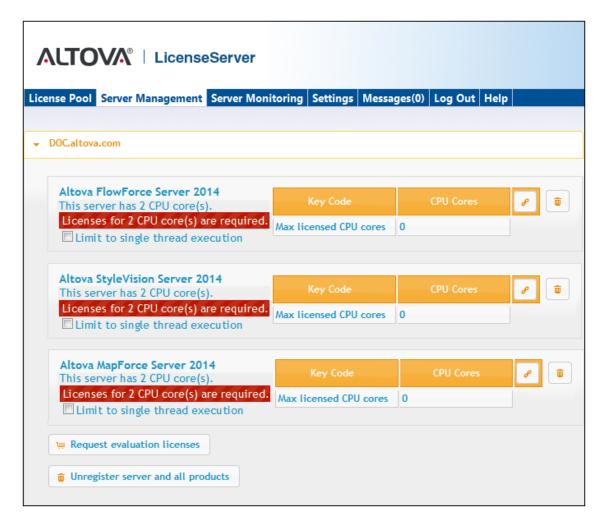
The search returns a list of available Altova LicenseServers on the network. One LicenseServer will be selected (*screenshot below*) and the others will be available in the dropdown list of the combo box. Select the LicenseServer on which your FlowForce license is stored.



Alternatively, you can enter the address of the LicenseServer in the LicenseServer field. If
the currently running LicenseServers are available as a dropdown list, you must click the
Manually Enter Address button to be able to enter an address in the LicenseServer
field.

After you have specified the LicenseServer, click **Register with LicenseServer**. The Altova server application will be registered with the specified LicenseServer, and that LicenseServer's Configuration page will open in a browser with its Client Management tab active (screenshot below).

**Note:** You may need to allow pop-ups in order for the LicenseServer Configuration page to be displayed.



In the screenshot below, three Altova products have been registered with the Altova LicenseServer at DOC.altova.com. How to assign licenses is described in the next section, <u>Assign Licenses to Registered Products</u>.

# Registering FlowForce Server via the FlowForce CLI (Windows)

On Windows machines, FlowForce Server can also be registered with an Altova LicenseServer on your network via the command line (CLI) by using the licenseserver command:

FlowForceServer licenseserver Server-Or-IP-Address

For example, if LicenseServer is running on http://localhost:8088, then register FlowForce Server with:

FlowForceServer licenseserver localhost

If FlowForce Server was installed with other Altova server products as sub-packages, registering

FlowForce Server will automatically also register the Altova server products. After successfully registering FlowForce Server, you can go to LicenseServer and assign a license to FlowForce Server. How to do this is described in the section Assign Licenses to Registered Products.

## Registering FlowForce Server via the FlowForce CLI (Linux)

On Linux machines, FlowForce Server can be registered with LicenseServer by using the <code>licenseserver</code> command of the FlowForce Server CLI. Note that FlowForce Server must be started with root rights.

sudo /opt/Altova/FlowForceServer2017/bin/flowforceserver licenseserver
localhost

In the command above, localhost is the name of the server on which LicenseServer is installed. Notice also that the location of the FlowForce Server executable is:

/opt/Altova/FlowForceServer2017/bin

After successfully registering FlowForce Server, you can go to LicenseServer and assign a license to FlowForce Server. How to do this is described in the section <u>Assign Licenses to Registered Products.</u>

### **Next Steps**

After you have registered your Altova product with LicenseServer, do the following:

- 1. If you have not already uploaded your license file/s to the LicenseServer (see previous section, <u>Upload the license/s</u>), upload the license file now and check that the license you want is active. If you have already done this, carry on to the next step, <u>Assign Licenses</u>.
- 2. <u>Assign a license</u> to your Altova product that has been registered with the LicenseServer.

# Register MapForce Server

## This section:

- Registering MapForce Server from FlowForce Server (Windows)
- Registering a standalone MapForce Server (Windows)
- Registering MapForce Server (Linux)
- Next steps

MapForce Server can be installed as part of the FlowForce Server package, or as a standalone server product. In either case, it must be registered with Altova LicenseServer. After it has been registered with LicenseServer, you can <u>assign a license</u> to it from LicenseServer. On Windows systems, you can install MapForce Server as part of the FlowForce installation. If MapForce

Server is installed as part of the FlowForce Server package, it will automatically be registered with LicenseServer when FlowForce is registered with LicenseServer. On Linux systems, the two products must be installed separately. If MapForce Server is installed after FlowForce Server, then it will be registered automatically when FlowForce Server is registered. But if MapForce Server is installed before FlowForce Server, you will have to register both products separately.

## Registering MapForce Server from FlowForce Server (Windows)

MapForce Server is packaged with FlowForce Server, so when FlowForce Server is registered with an Altova LicenseServer on your network, MapForce Server will automatically also be registered with LicenseServer. How to register FlowForce Server is described in the FlowForce Server documentation and in the section, Register FlowForce Server with LicenseServer.

After the registration, you can go to LicenseServer and assign a MapForce Server license to MapForce Server. How to do this is described in the section, <u>Assign Licenses to Registered Products</u>.

# Registering a standalone MapForce Server (Windows)

If you have installed MapForce Server as a standalone package, you must register it with an Altova LicenseServer on your network and then license it from the Altova LicenseServer. You can register MapForce Server via its command line interface (CLI) by using the licenseserver command:

MapForceServer licenseserver Server-Or-IP-Address

For example, if LicenseServer is running on http://localhost:8088, then register MapForce Server with:

MapForceServer licenseserver localhost

After successfully registering MapForce Server, you can go to LicenseServer and assign a license to MapForce Server. How to do this is described in the section, <u>Assign Licenses to Registered Products</u>.

# Registering MapForce Server (Linux)

On Linux machines, MapForce Server can be registered with LicenseServer by using the licenseserver command of the MapForce Server CLI. Note that MapForce Server must be started with root rights.

sudo /opt/Altova/MapForceServer2017/bin/mapforceserver licenseserver localhost

In the command above, localhost is the name of the server on which LicenseServer is installed.

Notice also that the location of the MapForce Server executable is:

/opt/Altova/MapForceServer2017/bin

After successfully registering MapForce Server, you can go to LicenseServer and assign a license to MapForce Server. How to do this is described in the section <u>Assign Licenses to Registered</u> Products.

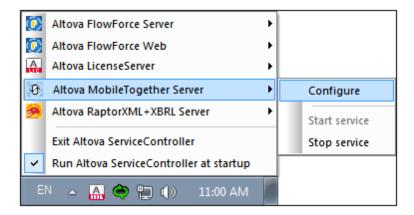
### **Next Steps**

After you have registered your Altova product with LicenseServer, do the following:

- 1. If you have not already uploaded your license file/s to the LicenseServer (see previous section, <u>Upload the license/s</u>), upload the license file now and check that the license you want is active. If you have already done this, carry on to the next step, <u>Assign Licenses</u>.
- 2. Assign a license to your Altova product that has been registered with the LicenseServer.

Register MobileTogether Server

To start MobileTogether Server, click the **ServiceController** icon in the system tray, hover over **Altova MobileTogether Server** in the menu that pops up (see screenshot below), and then select **Start Service** from the MobileTogether Server submenu. If MobileTogether Server is already running, the *Start Service* option will be disabled.



Register MobileTogether Server via:

- The Settings tab of the MobileTogether Server Web UI: (i) Start MobileTogether Server via ServiceController (see previous point); (ii) Enter your password to access the Configuration page; (iii) Select the Settings tab; (iv) Go to the LicenseServer pane at the bottom of the page, select the LicenseServer name or address, and click Register with LicenseServer.
- its CLI, using the licenseserver command:

  MobileTogetherServer licenseserver [options] ServerName-Or-IP-Address

  For example, if localhost is the name of the server on which LicenseServer is installed:

#### MobileTogetherServer licenseserver localhost

After successful registration, go to the <u>Client Management tab of LicenseServer's configuration</u> <u>page</u> to assign a license to MobileTogether Server.

Register RaptorXML(+XBRL) Server

#### This section:

- Registering RaptorXML(+XBRL) Server (Windows)
- Registering RaptorXML(+XBRL) Server (Linux)
- Next steps

RaptorXML(+XBRL) Server must be installed on the server machine or network to which LicenseServer is connected and then be started as a service. It must then be registered with LicenseServer. Only after registration can a <u>license be assigned</u> to it from LicenseServer. This section describes how to register RaptorXML(+XBRL) Server with LicenseServer.

# Registering RaptorXML(+XBRL) Server (Windows)

You can register RaptorXML(+XBRL) Server via its command line interface (CLI) by using the licenseserver command:

RaptorXML Server: RaptorXML licenseserver Server-Or-IP-Address
RaptorXML+XBRL RaptorXMLXBRL licenseserver Server-Or-IP-Address

Server:

For example, if LicenseServer is running on http://localhost:8088, then register RaptorXML(+XBRL) Server with:

RaptorXML Server: RaptorXML licenseserver localhost
RaptorXML+XBRL RaptorXMLXBRL licenseserver localhost

Server:

After successfully registering RaptorXML(+XBRL) Server, you can go to LicenseServer and assign a license to RaptorXML(+XBRL) Server. How to do this is described in the section <u>Assign</u> <u>Licenses to Registered Products</u>.

# Registering RaptorXML(+XBRL) Server (Linux)

On Linux machines, RaptorXML(+XBRL) Server can be registered with LicenseServer by using the licenseserver command of the RaptorXML(+XBRL) Server CLI. Note that RaptorXML(+XBRL) Server must be started with root rights.

sudo /opt/Altova/RaptorXMLServer2017/bin/raptorxmlserver licenseserver
localhost

sudo /opt/Altova/RaptorXMLXBRLServer2017/bin/raptorxmlxbrlserver licenseserver
localhost

In the command above, localhost is the name of the server on which LicenseServer is installed. Notice also that the location of the RaptorXML(+XBRL) Server executable is:

```
/opt/Altova/RaptorXMLServer2017/bin
/opt/Altova/RaptorXMLXBRLServer2017/bin
```

After successfully registering RaptorXML(+XBRL) Server, you can go to LicenseServer and assign a license to RaptorXML(+XBRL) Server. How to do this is described in the section <u>Assign</u> <u>Licenses to Registered Products</u>.

## **Next Steps**

After you have registered your Altova product with LicenseServer, do the following:

- 1. If you have not already uploaded your license file/s to the LicenseServer (see previous section, <u>Upload the license/s</u>), upload the license file now and check that the license you want is active. If you have already done this, carry on to the next step, <u>Assign Licenses</u>.
- 2. Assign a license to your Altova product that has been registered with the LicenseServer.

#### Register StyleVision Server

## This section:

- Registering StyleVision Server from FlowForce Server (Windows)
- Registering a standalone StyleVision Server (Windows)
- Registering StyleVision Server (Linux)
- Next steps

StyleVision Server can be installed as part of the FlowForce Server package or as a standalone server product. In either case, it must be registered with Altova LicenseServer. Only after it has been registered with LicenseServer can a <u>license be assigned</u> to it from LicenseServer. On Windows systems, if StyleVision Server was installed as part of the FlowForce Server package, it will automatically be registered when FlowForce is registered. On Linux systems, only if StyleVision Server is installed after FlowForce Server will it be registered automatically when FlowForce Server is registered subsequently.

# Registering StyleVision Server from FlowForce (Windows)

StyleVision Server is packaged with FlowForce Server, so when FlowForce Server is registered

with an Altova LicenseServer on your network, StyleVision Server will automatically also be registered with LicenseServer. How to register FlowForce Server is described in the FlowForce Server documentation and in the section, Register FlowForce Server with LicenseServer.

After the registration, you can go to LicenseServer and assign a StyleVision Server license to StyleVision Server. How to do this is described in the section <u>Assign Licenses to Registered Products</u>.

# Registering a standalone StyleVision Server (Windows)

If you have installed StyleVision Server as a standalone package on Windows, you must register it with an Altova LicenseServer on your network and then license it from the Altova LicenseServer. You can register StyleVision Server via its command line interface (CLI) by using the licenseserver command:

StyleVisionServer licenseserver Server-Or-IP-Address

For example, if LicenseServer is running on http://localhost:8088, then register StyleVision Server with:

StyleVisionServer licenseserver localhost

After successfully registering StyleVision Server, you can go to LicenseServer and assign a license to StyleVision Server. How to do this is described in the section <u>Assign Licenses to Registered Products</u>.

# Registering StyleVision Server (Linux)

On Linux machines, StyleVision Server can be registered with LicenseServer by using the <code>licenseserver</code> command of the StyleVision Server CLI. Note that StyleVision Server must be started with root rights.

sudo /opt/Altova/StyleVisionServer2017/bin/stylevisionserver licenseserver
localhost

In the command above, localhost is the name of the server on which LicenseServer is installed. Notice also that the location of the StyleVision Server executable is:

/opt/Altova/StyleVisionServer2017/bin

After successfully registering StyleVision Server, you can go to LicenseServer and assign a license to StyleVision Server. How to do this is described in the section <u>Assign Licenses to Registered Products</u>.

# **Next Steps**

After you have registered your Altova product with LicenseServer, do the following:

1. If you have not already uploaded your license file/s to the LicenseServer (see previous section, <u>Upload the license/s</u>), upload the license file now and check that the license you want is active. If you have already done this, carry on to the next step, <u>Assign Licenses</u>.

2. <u>Assign a license</u> to your Altova product that has been registered with the LicenseServer.

### **Assign Licenses to Registered Products**

## This section:

- Before assigning a license
- The Client Management tab
- Icons in the Client Management tab
- Note on cores and licenses
- Assigning a license
- Unregistering products from LicenseServer

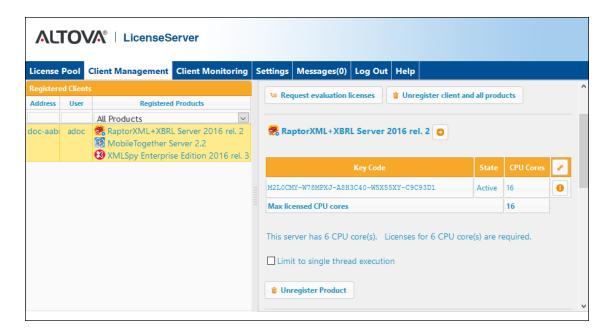
# Before assigning a license

Before you assign a license to an Altova product, make sure that:

- The relevant license has been uploaded to the <u>license pool of LicenseServer</u> and that the license is active.
- Your Altova product has been registered with LicenseServer.

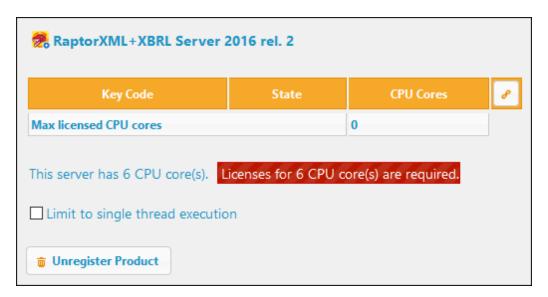
# The Client Management tab

Licenses are assigned in the <u>Client Management tab</u> of the LicenseServer Configuration page (*screenshot below*). The screenshot shows, in the left pane, that there is one client machine, which has three Altova products that have been registered with LicenseServer.



Note the following points about the Client Management tab:

- In the left pane, each product is listed under the name of its client machine. In the screenshot above, one client machine is listed. This client machine has three Altova products registered with LicenseServer. If an Altova product on a different client machine is registered with this LicenseServer, then that client machine, with its registered products, will also be listed in the left pane.
- When you select a client machine in the left pane, the licensing details of that machine's registered products are displayed in the right-hand pane. Here, you can edit the license assignments of each product.
- Each registered Altova product on a client machine has its own Key Code entry, which
  takes the key code of a license. A registered product is assigned a license by clicking
  its Edit Assigned Licenses button (see icon list below) and selecting the required
  license from those available for that product in the license pool. This procedure is
  explained in more detail below.
- Server products also have a line stating how many CPU cores need to be licensed to run
  that product on that client. If the number of licensed cores is less than the number
  required, then the information is marked in red (see screenshot below). (The number of
  CPU cores that need to be licensed is the number of CPU cores on that client and is
  obtained from the client machine by LicenseServer.)



- If multiple versions of a single product (for example, StyleVision Server 2013 and StyleVision Server 2014) have been installed on one machine and if each of these installations has been registered with a single LicenseServer, then the multiple registrations are consolidated in a single registration in the Client Management tab and displayed as a single registration. When a license is assigned to this single registration, all the installations indicated by that registration will be licensed. However, multiple instances of only one installation can be run simultaneously on the client machine. For example, multiple instances of StyleVision Server 2013 or multiple instances of StyleVision Server 2014 can be run simultaneously, but not one instance of StyleVision Server 2013 and one instance of StyleVision Server 2014. Note that newly installed versions must be registered for them to run.
- New versions of Altova server products can only be licensed with the version of LicenseServer that is the latest at the time of the product's release. Older Altova server products will work with newer versions of LicenseServer. So, if you are installing a new version of an Altova server product and if your current LicenseServer version is not the latest, de-install the older version of LicenseServer and install the latest version. All registration and licensing information held in your older version of LicenseServer will be saved, at the time of de-installation, to a database on the client machine, and will be imported automatically into the newer version. (The version number of the LicenseServer that is appropriate for any particular version of a server product is displayed during the installation of that server product. You can choose to install this version along with the server product. The version of the currently installed LicenseServer is given at the bottom of the LicenseServer configuration page.)

# **Icons in the Client Management tab**



*Edit Assigned Licenses*. Available with each product listing. Pops up the <u>Edit Assigned Licenses dialog</u>, in which new licenses can be assigned and already assigned licenses can be edited.



Show Licenses. Appears with each license listing. Switches to the License Pool tab and

highlights the selected license. Details of the license can be read here.



Unregister Product. The selected product will be unregistered from LicenseServer.

#### Note on cores and licenses

The licensing of Altova server products is based on the number of processor cores available on the product machine. For example, a dual-core processor has two cores, a quad-core processor four cores, a hexa-core processor six cores, and so on. The number of cores licensed for a product must be greater than or equal to the number of cores available on that server machine, whether the server is a physical or virtual machine. For example, if a server has eight cores (an octa-core processor), you must purchase at least one 8-core license. You can also combine licenses to achieve the core count. So, two 4-core licenses can also be used for an octa-core server instead of one 8-core license.

If you are using a computer server with a large number of CPU cores but only have a low volume to process, you may also create a virtual machine that is allocated a smaller number of cores, and purchase a license for that number. Such a deployment, of course, will have less processing speed than if all available cores on the server were utilized.

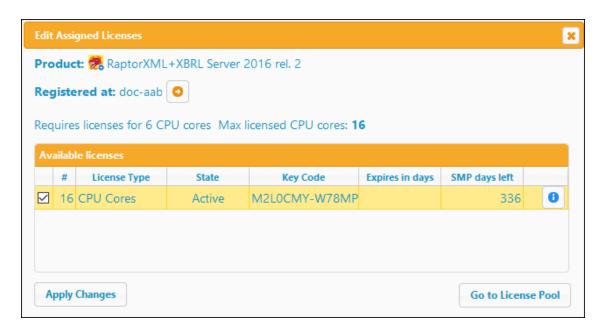
**Note:** Each Altova server product license can be used for only one client machine—the machine on which the Altova server product is installed—at a time, even if the license has unused licensing capacity. For example, if a 10-core license is used for a client machine that has 6 CPU cores, then the remaining 4 cores of licensing capacity cannot be used simultaneously for another client machine.

#### MobileTogether Server licenses

MobileTogether Server licenses are based on the number of CPU cores on the MobileTogether Server machine. Core licenses allow an unlimited number of MobileTogether Client devices to connect to the server. However, if you check the *Limit to single thread execution* check box, then only one mobile device will be able to connect to the MobileTogether Server at any time. This is useful for evaluation and small-scale testing. Note that, if, in this case, a second device connects to MobileTogether Sever, then it will take over the license. The first device will not be able to connect any more and will receive an error message to this effect.

## **Assigning a license**

To assign a license to a registered product, click the **Edit Assigned Licenses** button of that product. This pops up the Manage Licenses dialog (*screenshot below*).



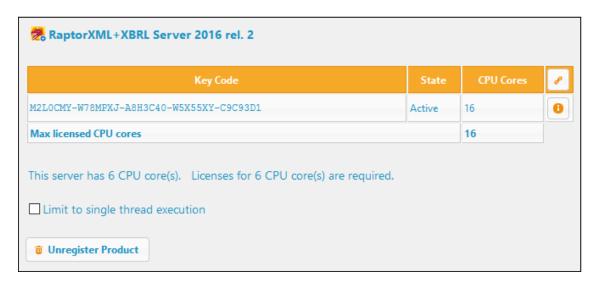
Note the following points about the licenses displayed in the Manage Licenses dialog:

- The product to be licensed is listed at the top left of the dialog. In the screenshot above the product is Altova RaptorXML+XBRL Server.
- The machine on which the server is installed (doc-aab in the screenshot above) is listed
  next
- The dialog displays all the currently active licenses for that product that are in the license
  pool. In our screenshot, there is one currently active RaptorXML+XBRL Server license in
  the license pool. (LicenseServer will automatically detect from each license in the pool
  the product for which it has been issued.)
- The license type can be either *Cores* (all Altova server products, including MobileTogether Server) or *Users* (MobileTogether Server only). The license type is shown in the *License Type* column. The license in the screenshot above has been licensed for 16 CPU cores.
- You need to know the number of processor cores on the server on which the Altova server product has been installed. If the machine has a dual-core processor, you need a two-core (the CPU Cores count) license. How many cores the registered server product requires is listed below the machine name. The license you assign to this server must be valid for a sufficient number of cores. Note that you can combine licenses to reach the required number of cores. So, if the machine's processor is octa-core (eight-core), you can combine two 4-core licenses.
- The Edit Assigned Licenses dialog will list only currently active licenses for that product.
   Licenses for other Altova products will not be listed.
- Licenses that have been assigned already—for example, to another installation of the product on the network—will have their check boxes checked. So only unchecked licenses may be selected.
- The CPU Cores (or # for MobileTogether Server users) column indicates for how many CPU cores (or MobileTogether Clients) a license is valid.
- If you wish to make modifications to the license pool—for example, to upload, activate, deactivate, or delete a license—click the **Go to License Pool** button.

Select the license you wish to assign. The license's check box will be checked. Also, the total number of CPU cores licensed for that product on that client is listed near the top left of the dialog

as Max licensed CPU cores (see screenshot above). You can select more licenses if you wish to increase the number of licensed CPU cores for that product on that client. The Max licensed CPU cores in this case will be the sum of the CPU cores of all the selected licenses.

After selecting the license/s, click **Apply Changes**. The license/s will be assigned to that product and displayed in the Client Management tab (see screenshot below). The screenshot shows that a 16-CPU-core license for Altova RaptorXML+XBRL has been assigned to the client machine.



### **Unregistering products**

Each Altova product registered with LicenseServer is listed in the right-hand (*Product licensing*) pane under its client machine name and has an **Unregister Product** button at the bottom of its entry (see screenshot above). Click this button to unregister the product from LicenseServer. If a license was assigned to the product, the assignment will be terminated when the product is unregistered. To unregister all products, click the **Unregister client and all products** button at the top of the right-hand (*Product licensing*) pane (see first screenshot in this section).

To re-register a product with LicenseServer, do the following:

- Server products: Go to the Settings page in the server product's Web UI. If the server product has no Web UI, open a command prompt window and use the product's CLI to register it. The steps for each server product. are described here: Register FlowForce Server, Register MapForce Server, Register MobileTogether Server, Register StyleVision Server, and Register RaptorXML(+XBRL) Server.
- Desktop products: Open the product's <u>Software Activation dialog</u> (Help | Software Activation), switch to Activation via LicenseServer mode, and, in the *Altova LicenseServer* field, select the LicenseServer you want to register the product with. The product will be registered, and it will appear in LicenseServer's Client Management tab, in that client's registered-product list.

# 4.1.7 Configuration Page Reference

The LicenseServer Configuration page is the administrator's interface with LicenseServer (Web UI). It allows the management of LicenseServer and the licensing of Altova products that have been registered with LicenseServer. The LicenseServer Configuration page is viewed in a web browser. How to open the Configuration page is described in the sections, Open LicenseServer Config Page (Windows), Open LicenseServer Config Page (Linux) and Open LicenseServer's Config Page (Mac OS X).

This section is a user reference for the Configuration page and is organized by the tabs of the Configuration page:

- License Pool
- Client Management
- Client Monitoring
- Settings
- Messages, Log Out

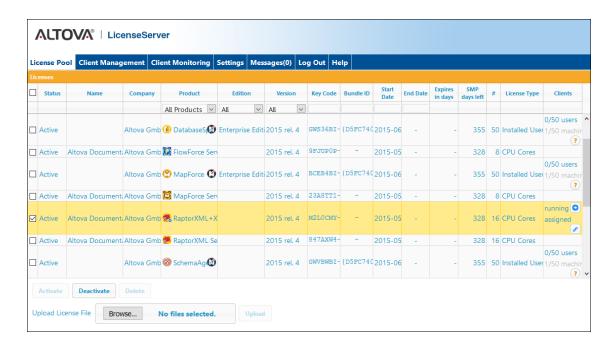
For a step-by-step guide of how to assign licenses with LicenseServer, see the section  $\underline{\text{How to}}$  Assign Licenses.

#### License Pool

### This section:

- Uploading a license
- License status
- Activating, de-activating, and deleting a license
- Icons in the License Pool tab
- License information
- Note on desktop product licenses
- Note on cores and licenses

The **License Pool** tab displays information about the licenses that are currently available on LicenseServer (see screenshot below). When a license file is uploaded to LicenseServer with the **Upload** button on this page, all the licenses contained in that license file are placed in the license pool on LicenseServer. The License Pool page therefore provides an overview of all Altova product licenses that are currently available on LicenseServer, together with the details of each of these licenses. On this page, in addition to being able to upload licenses to the license pool, you can activate, deactivate, and delete selected licenses.



## Uploading a license

To upload a license file (a .altova\_licenses file which you receive from Altova GmbH for your Altova product), click the **Browse** button, browse for the license file, and select it. On clicking **Upload**, all the licenses contained in the license file are placed in the license pool and displayed on the License Pool page (*screenshot above*).

## **License status**

License status values are as follows:

- Activating: When a license is uploaded to the license pool of LicenseServer, the server
  will transmit license-related data to the altova.com master licensing server to validate,
  authenticate, and activate the license that was supplied. This is necessary to ensure
  compliance with the Altova license agreements. During this initial activation and
  authentication transaction—which typically lasts between 30 seconds and a couple of
  minutes, depending on your Internet connection, speed, and overall network traffic—the
  status of the license will be indicated as Activating....
- Failed Verification: If a connection with the altova.com master licensing server cannot be made, then the status of the license in the pool will be shown as Failed Verification. If this happens, check your Internet connection and firewall rules to ensure that LicenseServer is able to communicate with the altova.com master licensing server.
- Active: Once the license has been authenticated and activated, the status will change to Active.
- Inactive: If a license has been verified, but is present on another LicenseServer on the

network, the status will be shown as *Inactive*. An *Inactive* status also results when a license is manually deactivated in the license pool by the administrator.

- Pending: If the start date of a license is a date in the future, then the license is shown as
   Pending. The status will change to Active on the start date. A Pending license can be
   assigned to a product and ensures that the licensing of a product continues uninterrupted
   before an existing license expires. (A product is allowed to have two active licenses
   assigned to it at a given time.)
- Blocked: A license is shown in the license pool as Blocked if there was a problem
  authenticating the license and the altova.com master licensing server has not granted
  permission to the LicenseServer to use this license. This could be the result of a license
  agreement violation, over-usage of a license, or other compliance issues. Should you see
  a license showing up as Blocked, please contact Altova Support with your license
  information and any other relevant data.

These statuses are summarized in the table below:

Status	Meaning
Activating	On upload, license information is sent to altova.com for verification. Refresh the browser to view the updated status. Verification and activation can take a few minutes.
Failed Verification	A connection to altova.com could not be made. After establishing a connection, either restart the service or activate the license (with the <b>Activate</b> button).
Active	Verification was successful, the license is active.
Inactive	Verification was successful, but the license is also on another LicenseServer on the network. Licenses can be made inactive with the <b>Deactivate</b> button.
Pending	A <i>Pending</i> license has a start date in the future, and it turns <i>Active</i> on that start date. It can be assigned to a product and ensures the automatic renewal of a product's licensing before an existing license expires.
Blocked	Verification was not successful. License is invalid and is blocked. Contact Altova Support.

**Note:** After a license has been sent to altova.com for verification, the browser must be refreshed to see the updated status. Verification and activation can take a few minutes.

**Note:** If a connection to altova.com could not be made, the status will be *Failed Verification*. After establishing a connection, either restart the service or try activating the license with the **Activate** button.

**Note:** When a license is given a status of *Inactive* or *Blocked*, a message explaining the status is also added to the Messages log.

Only an active or pending license can be assigned to a product installation. An inactive license can be activated or deleted from the license pool. If a license is deleted from the license pool, it

can be uploaded again to the pool by uploading the license file containing it. When a license file is updated, only those licenses in it that are not already in the pool will be uploaded to the pool. To activate, deactivate, or delete a license, select it and then click the **Activate**, **Deactivate**, or **Delete** button, respectively.

## Connection to the Master Licensing Server at altova.com

Altova LicenseServer needs to be able to communicate with the Master Licensing Server at altova.com to validate and authenticate license-related data and to ensure continuous compliance with the Altova license agreements. This communication occurs over HTTPS using port 443. If Altova LicenseServer, after making the initial verification with the altova.com Master Licensing Server, is unable to again connect with altova.com for a duration of more than 5 days (= 120 hours), then Altova LicenseServer will no longer permit the usage of any Altova software products connected to that Altova LicenseServer.

Any such loss of connection with the altova.com master servers will be logged in the Messages tab of the Configuration page of Altova LicenseServer. In addition, the administrator can configure the Altova LicenseServer to automatically send an alert email when the connection to altova.com is lost. Alert Mail settings are available in the Settings tab of the Configuration page.

## Activating, deactivating, and deleting a license

An active license can be deactivated by selecting the license and clicking **Deactivate**. An inactive license can be activated (**Activate** button) or deleted (**Delete** button). When a license is deleted it is removed from the license pool. A deleted license can be added again to the license pool by uploading the license file containing it. If a license file is re-uploaded, only licenses that are not already in the license pool will be added to the license pool; licenses that are already in the pool will not be re-added.

# **Icons in the License Pool tab**

- Altova MissionKit logo. Appears next to individual Altova desktop product names when the desktop product license is part of a MissionKit license. See Note on desktop product licenses below.
- Show Assigned Client. Appears in the Clients column of a license that has been assigned. Goes to the Client Management tab, in which the licenses of a client's registered products can be managed.
- Show Running Client. Appears in the Clients column of a license that has been assigned to software that is currently running. Goes to the Client Monitoring tab of the client machine running that software. Here, information about the selected client and its registered software is displayed.
- Show Info. Appears in the Clients column of a license that has not been assigned. Displays information about the license, such as the user count and whether the license is part of a license bundle.

### License information

The following license information is displayed:

• Status: Can be one of the following values: Activating | Failed Verification | Active | Inactive | Blocked. See License status above.

- *Name, Company:* The name and company of the licensee. This information was submitted at the time of purchase.
- *Product, Edition, Version:* The version and edition of the licensed products. At the top of each column is a combo box to filter licenses by category.
- Key Code, Bundle ID: The license key to unlock the product. All products in a single
   Altova MissionKit bundle have the same Bundle ID. Non-bundled products have no Bundle
   ID.
- Start Date, End Date: Together they give the validity period of the license. Permanent licenses have no end date.
- Expires in days, SMP (days left): The number of days left before the license expires.
   Each licensed purchase comes with a Support & Maintenance Package, which is valid for a certain number of days. The SMP column notes how many SMP days are still left.
- #, License Type: The number of users or CPU cores that are allowed for the license is listed in the # column. Whether it is users or cores that the license regulates is given in the License Type column. In the case of Altova's MobileTogether Server product, licenses are also assigned on the basis of the number of MobileTogether client devices that connect to MobileTogether Server. In the case of all other Altova server products, licenses are assigned on the basis of CPU cores only (see note below). In the case of Altova desktop products, licenses are assigned on the basis of users. See Note on desktop product licenses below.
- Clients: This column has entries only for MobileTogether Server licenses and desktop product licenses. It does not have entries for server product licenses. For MobileTogether Server device licenses, this column displays whether the license is assigned. For Desktop products, the column displays the machine count and user count as described below.

# Desktop products: machine count and user count

- The *machine count* indicates the number of machines that are currently licensed to run the software with a given license. For example, 7/10 machines means that the license can be used for software instances on 10 machines, and is currently being used for software instances on 7 machines. Click the **Show Assigned Client** button to go to the Client Management tab and see the details of the client machine's license/s.
- The *user count* indicates the number of users that are currently using the license out of the total number of allowed users. Only licensed software installations that are currently running are counted. For example, 3/10 users means that the license is currently being used by 3 users out of a total of 10 allowed users. If a licensed software installation is currently running, you can click the **Show Running Client** button to open the Client Monitoring tab and see details of Altova products that are running on client machines on the network.
- The *user count* and *machine count* together tell you the current licensing capacity and usage of a given license. For example, if the machine count of an <u>installed user license</u> is 7/10 and if the user count is 3/10, then we know the following: (i) The product software is allowed to be licensed on 10 machines; (ii) The software has been licensed on 7

machines; (iii) 3 of the 7 licensed software installations are currently running.

# Unassigning a license

To unassign a license from a software installation on a machine, go to the <u>Client Management</u> tab. Select the machine and then the software to be unassigned. Then click the **Edit Assigned Licenses** button, deselect the license, and click **Apply Changes**.

### Note on desktop product licenses

There are three types of desktop user licenses:

- Installed User. Licenses are purchased for an exact number of computers on which you
  would like to install the software. For example, if you purchase a 10-Installed-User
  license, you may install and use the software on up to 10 computers. On each licensed
  machine you may start as many simultaneous software instances as you wish. The
  license, for each "installed user" allows the software product to be used on that machine.
- Concurrent User: This license allows you to install the software on up to that number of computers that is equal to 10 times the allowed number of concurrent users. All the installations must be on the same physical network. At any given time, the software can be used by only the allowed number of concurrent users. For example, if you purchase a 20-Concurrent-User license, then the software may be installed on up to 200 computers on the same physical network and used on up to 20 computers at any given time. If you would like to use Concurrent User licenses on different physical networks, you must purchase a separate license for each network. Note that you cannot use one Concurrent User license to span multiple networks.
- Named User: For each Named-User license you may install the software on up to 5 computers, but only the user named in the license may use the software. With this license, a user can work on different computers provided that only one instance of the software is used at any given time.

## Note about Altova MissionKit licenses

Altova MissionKit is a suite of Altova desktop products. An Altova MissionKit license comprises individual licenses for each of the desktop products in the MissionKit suite. Each of these individual product licenses has a different and unique key code, but the same MissionKit Bundle ID. If you upload an Altova MissionKit license to the license pool, then the individual licenses of each product comprising MissionKit is listed in the License Pool (with the Altova MissionKit logo next to it). If you assign one of these product licenses to a particular user, then all the other products of that MissionKit bundle are also assigned to that user. As a result, no other product in that particular MissionKit bundle can be assigned to another user.

## License check-outs

You can check out a license from the license pool for a period of up to 30 days so that the license is stored on the product machine. This enables you to work offline, which is useful, for example, if you wish to work in an environment where there is no access to your Altova LicenseServer (such as when your Altova product is installed on a laptop and you are traveling). While the license is checked out, LicenseServer displays the license as being in use, and the license cannot be used by any other machine. The license automatically reverts to the checked-in state when the check-out period ends. Alternatively, a checked-out license can be checked in at any time via the **Check in** button of the Software Activation dialog. To check out a license from the license pool to your product machine, go to the **Help** menu of your Altova desktop product and select **Software** 

Activation. Refer to the user manual of your Altova product for more more information.

#### Note on cores and licenses

The licensing of Altova server products is based on the number of processor cores available on the product machine. For example, a dual-core processor has two cores, a quad-core processor four cores, a hexa-core processor six cores, and so on. The number of cores licensed for a product must be greater than or equal to the number of cores available on that server machine, whether the server is a physical or virtual machine. For example, if a server has eight cores (an octa-core processor), you must purchase at least one 8-core license. You can also combine licenses to achieve the core count. So, two 4-core licenses can also be used for an octa-core server instead of one 8-core license.

If you are using a computer server with a large number of CPU cores but only have a low volume to process, you may also create a virtual machine that is allocated a smaller number of cores, and purchase a license for that number. Such a deployment, of course, will have less processing speed than if all available cores on the server were utilized.

**Note:** Each Altova server product license can be used for only one client machine—the machine on which the Altova server product is installed—at a time, even if the license has unused licensing capacity. For example, if a 10-core license is used for a client machine that has 6 CPU cores, then the remaining 4 cores of licensing capacity cannot be used simultaneously for another client machine.

### MobileTogether Server licenses

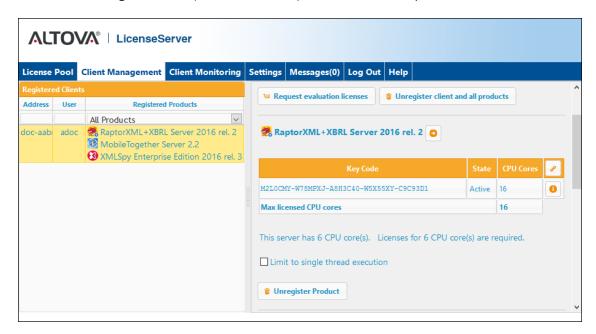
MobileTogether Server licenses are based on the number of CPU cores on the MobileTogether Server machine. Core licenses allow an unlimited number of MobileTogether Client devices to connect to the server. However, if you check the *Limit to single thread execution* check box, then only one mobile device will be able to connect to the MobileTogether Server at any time. This is useful for evaluation and small-scale testing. Note that, if, in this case, a second device connects to MobileTogether Sever, then it will take over the license. The first device will not be able to connect any more and will receive an error message to this effect.

### **Client Management**

#### This section:

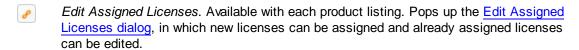
- Icons in the Client Management tab
- Managing licenses in the Product Licensing pane
- Assigning licenses
- Single thread execution
- One client machine under different names
- Requesting an evaluation license
- Unregistering products

The Client Management tab (screenshot below), is divided into two panes:



- Registered clients: The left-side pane displays a tabular listing of machines on the network that have at least one Altova product which is registered with LicenseServer. Such machines are called registered clients. Each registered client is listed in the left pane with all its registered products. How to register products with LicenseServer is described in Register Product/s. The display in this pane can be filtered by selecting or entering a filter at the top of one of the pane's columns.
- Product licensing: This is the right-side pane. When a registered client is selected in the
  left (Registered Clients) pane, then the licensing details of that client's registered
  products are displayed in the right pane. Here you can manage the licensing of each
  registered product (described below).

### **Icons in the Client Management tab**



- Show Licenses. Appears with each license listing. Switches to the <u>License Pool tab</u> and highlights the selected license. Details of the license can be read here.
- Unregister Product. The selected product (on the selected client machine) will be unregistered from LicenseServer. See <u>Unregistering products</u> below. To unregister a client and all its products, click **Unregister client and all products** at the top of the pane.

# Managing licenses in the Product Licensing pane

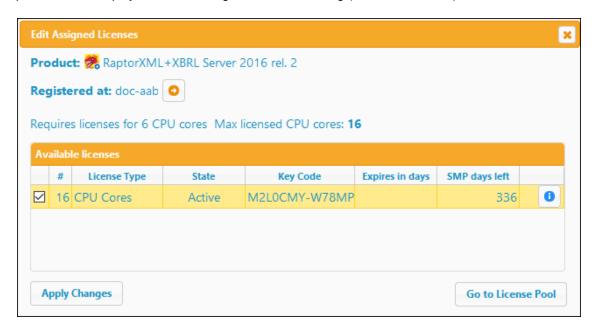
In the right-side *Product licensing* pane, you can do the following:

Assign, unassign, change a product's license: Click the product's Edit Assigned
 Licenses button to do this. See <u>Assigning licenses</u> below. Note that each server product
 has a line stating how many CPU cores need to be licensed to run that product on that
 client. If the number of licensed cores is less than the number required, then the
 information is marked in red. (The number of CPU cores that need to be licensed is the
 number of CPU cores on that client and is obtained from the client machine by
 LicenseServer.)

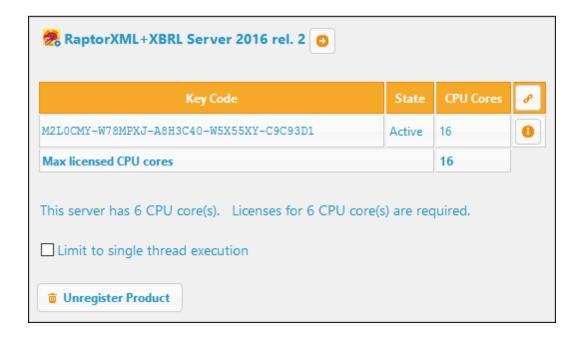
- Set up a single-core, server-product license to use only one core of a client: See <u>Single</u> thread execution below.
- Unregister a product from LicenseServer: Use a product's Unregister Product button for this. See Unregistering products below.

## **Assigning licenses**

To assign a license to a registered product, click the **Edit Assigned Licenses** button of that product. This displays the Edit Assigned Licenses dialog (*screenshot below*).



Select the license/s you wish to assign, then click **Apply Changes**. The license/s will be assigned to that product and displayed in the *Product licensing* tab of the Client Management tab (see screenshot below).



## Single thread execution

If an Altova server-product license for only one core is available in the license pool, a machine with multiple cores can be assigned this one-core license. In such a case, the machine will run that product on a single core. Processing will therefore be slower, because multi-threading (which is possible on multiple cores) will not be available. The product will be executed in single thread mode on that machine.

To assign a single-core license to a multiple-core machine, select the *Limit to single thread execution* check box for that product.

In the case of **MobileTogether Server (MTS)**, if single-thread execution is selected for an MTS core license, then only one mobile device will be able to connect to the MobileTogether Server at any time. Note that, if, in this case, a second device connects to MobileTogether Sever, then it will take over the license. The first device will not be able to connect any more and will receive an error message to this effect.

## One client machine under different names

If a client machine is registered more than once with LicenseServer, it might appear in the Client Management tab under multiple names, that is, with multiple entries. This could happen, for example, if a machine is re-registered with the host name given in a different form.

To ensure that additional licenses are not redundantly assigned to the same machine under its different names, you should unregister redundant client machine entries by clicking the **Unregister client and all products** button at the top of the right-hand (*Product licensing*) pane. Also, if the same license is assigned multiple times to the same machine under its different names, licensing conflicts could arise. So, to avoid these two situations (redundant licensing and multiple assignments of a single license), it is recommended that redundant entries of a single

client machine be unregistered.

Given below are forms a machine name might take in the Client Management tab:

• Host name with domain name (the fully qualified domain name, FQDN), such as: "win80-x64\_1.my.domain.com" or "Doc3.my.domain.com". This happens when the host name of the machine (with or without the domain information) is passed as the argument of the licenseserver CLI command that is used to register the server product with LicenseServer. For example: <altravaserverProduct> licenseserver Doc3. This produces an FQDN such as: Doc3.my.domain.com.

An FQDN is also produced when localhost is supplied on Windows 7 and 10 systems as the host name.

- Host name without domain name. For example: "win80-x64\_1" or "Doc3". This happens on Windows 8 systems when localhost is given as the machine name.
- *localhost*. In some cases, localhost is also displayed as a machine name.

**Note:** If, during installation of an Altova server product on Windows machines, the machine is automatically registered with LicenseServer, localhost is used by the installer as the machine name.

#### Requesting an evaluation license

You can obtain a 30-day free evaluation license for each of a client's installed Altova products that have been registered with LicenseServer. Click the **Request Evaluation Licenses** button at the top of the right-hand (*Product licensing*) pane. A dialog appears that contains a list of the Altova products (on that client machine) which have been registered with LicenseServer. Make sure that the products for which you want an evaluation license are checked, then fill in the registration fields, and send the request. You will receive an e-mail from Altova containing the 30-day evaluation license/s. For server products, the number of cores for which the license is valid will be exactly the number required by the product at the time the request is sent. Save the license/s to disk and upload them to the license pool.

### **Unregistering products**

Each Altova product registered with LicenseServer is listed in the right-hand (*Product licensing*) pane under its client machine name and has an **Unregister Product** button at the bottom of its entry. Click this button to unregister the product from LicenseServer. If a license was assigned to the product, the assignment will be terminated when the product is unregistered. To unregister all products, click the **Unregister client and all products** button at the top of the right-hand (*Product licensing*) pane (see first screenshot in this section).

To re-register a product with LicenseServer, do the following:

Server products: Go to the Settings page in the server product's Web UI. If the server
product has no Web UI, open a command prompt window and use the product's CLI to
register it. The steps for each server product. are described here: Register FlowForce

<u>Server, Register MapForce Server, Register MobileTogether Server, Register StyleVision</u> Server, and Register RaptorXML(+XBRL) Server.

Desktop products: Open the product's <u>Software Activation dialog</u> (Help | Software Activation), switch to Activation via LicenseServer mode, and, in the *Altova LicenseServer* field, select the LicenseServer you want to register the product with. The product will be registered, and it will appear in LicenseServer's Client Management tab, in that client's registered-product list.

For more information, see the section, Assigning Licenses to Registered Products.

### **Client Monitoring**

The **Client Monitoring** tab provides an overview of the selected client machine. The tab displays the following:

#### Checked-out Clients

End users of an Altova desktop product (not server product), such as XMLSpy or MapForce, can check out a license that is registered with LicenseServer. This would typically be done in situations where it is foreseen that the end user machine will be offline for a certain period of time. The license can be checked out from LicenseServer for the period during which the machine will be offline. For this period, the end user can continue using the Altova desktop product without making contact with LicenseServer. The licenses and users that are currently checked out, together with the check-out period, are listed under this heading.

Note: License check-out is carried out by the end user via the Software Activation dialog (Help | Software Activation) of the Altova desktop product.

#### Running clients

A list of the Altova products that are currently running on that client. If multiple instances of a product are running then each of these instances is listed.



**Note:** <u>Failover LicenseServers</u> work with client applications that are v2015rel3 or later (in the case of Altova MobileTogether Server, version 1.5 or later); older clients are flagged.

**Note:** If a license for a desktop product is checked out, then it will be displayed in a separate *Checked-out Clients* section of the Client Monitoring tab. The license will be indicated as running, and will not be able for use to any other machine on the network.

### Icons in the Client Monitoring tab



*Show License(s).* Available for each product instance. Goes to the <u>License Pool</u> tab, and highlights the license of the selected product instance.



*Manage Client*. Available for each product instance. Goes to the <u>Client Management</u> tab, and highlights the client of the selected product instance.

### Settings

#### This section:

- Failover LicenseServer settings
- Network settings
- Alert Mail settings
- Miscellaneous settings

The **Settings** tab is described below. You can set the following:

- A time period to wait till LicenseServer shuts down. A shutdown would be implemented typically for server maintenance. The time to shutdown can be used to save work on clients running Altova desktop products. The shutdown time you select is the maximum time to shutdown. LicenseServer will shut down earlier, as soon as LicenseServer is no longer connected to any client running a desktop product. The countdown to shutdown starts when you click Shutdown. To cancel shutdown, click Abort Shutdown. To enable clients to run during a LicenseServer shutdown, configure a Failover LicenseServer.
- A second LicenseServer can be configured to take over from the Primary LicenseServer if the Primary LicenseServer becomes unavailable. This second LicenseServer is called the Failover LicenseServer. How to specify its settings are described here.
- The password for logging in to LicenseServer. Enter the desired password and click Change Password.
- Test connectivity to Altova by clicking **Test Connection to Altova**. Note that you must save new settings (by clicking the **Save** button at the bottom of the pane) before testing the connection. The **Test Connection to Altova** button is disabled while the test is in progress, and becomes enabled again when the test has been completed.
- Network settings for the web-based configuration page (Web UI), for the proxy server (if any) that is used to connect to the Internet, and for License Service availability. These settings are described in <a href="Network settings">Network settings</a> below.
- Email server settings and the alert mail recipient to contact in the event of a significant LicenseServer occurrence. These settings are described in Alert Mail settings below.
- After you change a setting, click Save at the bottom of the pane. Note that a changed setting will not take effect till it is saved.

#### Failover LicenseServer settings

A second LicenseServer can be configured to take over from the Primary LicenseServer if the Primary LicenseServer becomes unavailable. This second LicenseServer is called the **Failover LicenseServer**.

### Failover LicenseServer Settings

To reduce the risk of an unavailable LicenseServer you can configure a second LicenseServer as a backup or "Failover LicenseServer".

In the event that the Primary LicenseServer becomes unavailable a Failover LicenseServer can take over.

### LicenseServer Mode

- Primary LicenseServer
- Failover LicenseServer

Please note: The Failover LicenseServer periodically synchronizes all licenses, registered clients and license assignments from the Primary LicenseServer. Whenever a Failover LicenseServer takes over from a Primary LicenseServer any changes to these items made on the Failover LicenseServer during this period will be lost as soon as the Primary LicenseServer regains control. Other settings such as Proxy Server and Mail settings are independently set in each server and are not synchronized.

This is a Failover LicenseServer for the LicenseServer at kubu6.altova.com

Last seen 2/5/2015, 11:56:04 AM

To set up a LicenseServer as the Failover LicenseServer, do the following:

- 1. Install LicenseServer as described in the Installation section.
- 2. Set the LicenseServer's mode to Failover LicenseServer by selecting the corresponding radio button (see screenshot above). (By default the LicenseServer Mode is set to Primary LicenseServer.)
- 3. In the Find Primary LicenseServer dialog that appears (screenshot below), enter the Primary LicenseServer you want to back up with this Failover LicenseServer. You can do this in one of two ways: (i) Click Search for LicenseServers and then select, in the combo box, the LicenseServer you want to backup from the list of found LicenseServers; (ii) Click Manually Enter Address, and enter the address of the LicenseServer you want to backup. After entering the Primary LicenseServer, click Connect to Primary LicenseServer.



4. A confirmation dialog appears, asking you to confirm that you wish to set the current LicenseServer as the Failover LicenseServer (of the Primary LicenseServer you have just selected). Click **Yes** if you wish to confirm. Note that going ahead with the confirmation will remove any installed licenses and registered clients on the current LicenseServer.

Once a Failover LicenseServer has been configured, both the Primary LicenseServer and Failover LicenseServer will have notifications about their respective modes at the top of the Configuration Page. In the two screenshots below, the Failover LicenseServer is shown first, then the Primary LicenseServer.



### Note the following points:

- After a Failover LicenseServer has been configured, it periodically synchronizes all licenses, registered clients, and license agreements from the Primary. If the Primary becomes unavailable, then the Failover takes over the role of LicenseServer. Once the Primary becomes available again, the Primary retakes control from the Failover. Any license-related modifications made on the Failover in this period will be lost when the Primary regains control.
- The Failover LicenseServer will provide licenses only to clients that are of version 2015 rel 3 or later (in the case of Altova MobileTogether Server, version 1.5 or later). Older clients are flagged as such in the <u>Client Monitoring tab</u> of the Primary LicenseServer (screenshot below). We recommend that you upgrade your client applications to version 2015 rel 3 or later if you wish to use the Failover LicenseServer feature (in the case of Altova MobileTogether Server, version 1.5 or later).

#### License checkout

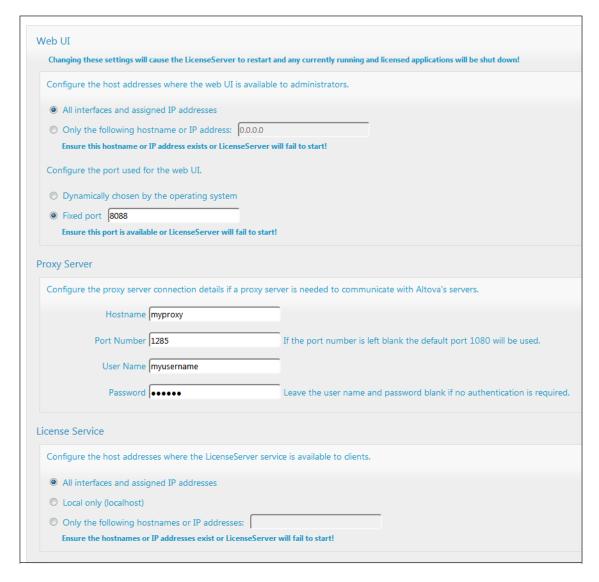
You can select whether to allow desktop licenses to be checked out from the license pool to the machine on which the Altova desktop product is installed. If you allow this, then a client that has

acquired a license from your LicenseServer can check out this license and remain unmonitored for any period up to the maximum period you specify in this setting. The maximum allowed period for check outs is 30 days. (This enables a client to continue using the desktop product even when access to LicenseServer is not possible (say, when traveling) or not wanted.) The client can do the check out via the **Help | Software Activation** command of the Altova desktop application.

After a license has been checked out, it is moved to the *Checked-out Clients* section of the <u>Client Monitoring</u> tab. It will be considered to be in use, and it will not be available for use by any other client on teh network.

### **Network settings**

Administrators can specify network access points to the LicenseServer configuration page and to LicenseServer.



Web UI: Allowed IP addresses can vary from: all interfaces and IP addresses on that

machine to a fixed address. Ports can be either dynamically calculated or fixed. This allows a wide range of allowed IP-Address:Port settings. The default port setting is **8088**.

- Proxy Server (available from v1.3 onwards): If a proxy server is being used to connect to the Internet, the details of the proxy server must be entered in the Proxy Server pane (see screenshot above). Otherwise, the proxy server settings can be left blank. To configure LicenseServer to use a proxy server, enter the proxy server's host name, and, if required, a port number. If the proxy server does not need authentication, then the User Name and Password fields can be left blank.
- License Service: The machine on which License Server is installed can be connected to multiple networks via one or more network interfaces. On each network, the License Server machine is known by a host name and an IP address. The License Service setting lets you configure on which networks the license service is available. The localhost option allows the service on the local machine only. If you list hostnames and/or IP addresses, use a comma-separated list without any spaces (for example: hostname1, IPAddress1, hostname2). The port number of the service is fixed at 35355.

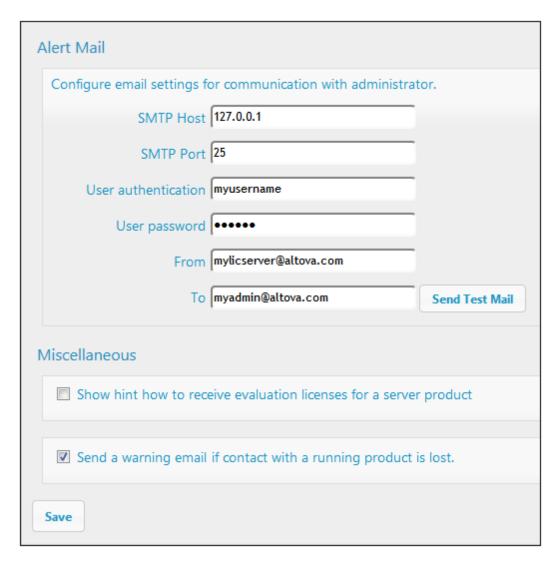
The default settings allow unrestricted access to LicenseServer and its configuration page from within the networks to which LicenseServer is connected. If you wish to restrict access to either LicenseServer or its configuration page, enter the appropriate settings and click **Save**.

Run a connectivity test (see above) to check that the settings are correct.

### **Alert Mail settings**

Altova LicenseServer needs to be connected to the altova.com server. If the connection is broken for more than 24\*5 hours (5 days), LicenseServer will not allow licenses. As a result, work sessions with Altova products licensed by LicenseServer could be disrupted.

In order to alert the administrator that a connection is broken, an alert mail can be sent to an email address. The Alert Mail pane (see screenshot below) is where you enter settings for sending alert mails to an administrator's email address.



*SMTP Host* and *SMTP Port* are the access details of the email server from which the email alert will be sent. *User Authentication* and *User Password* are the user's credentials for accessing the email server. The *From* field takes the address of the email account from which the email will be sent. The *To* field takes the recipient's email address.

Click **Save** when done. After saving the Alert Mail settings, email alerts will be sent to the address specified whenever a significant event occurs, such as when connection to altova.com is lost. Note that such events are also recorded in the Messages tab, and can be looked up there.

### Miscellaneous settings

### Show hints for receiving and deploying evaluation licenses

Checking this box (see screenshot above) displays, at the top of the configuration page, brief instructions about how to evaluate and deploy evaluation licenses.

Send a warning email if contact with a running product is lost

A warning message is sent from the *From* address to the *To* address if a connection with a product that is licensed and running is lost.

### Messages, Log Out

The **Messages** tab displays all messages relevant to licenses in the license pool of the LicenseServer. Each message has a **Delete** button that allows you to delete that particular message.

The **Log Out** tab serves as the Log Out button. Clicking the tab logs you out immediately and then displays the Login mask.

### 4.1.8 Password Reset

If you forget your LicenseServer password, you can use the passwordreset command from the CLI to reset the password to default.

- 1. Open a command line window.
- 2. Change to the directory where the LicenseServer application package or executable is installed.
- 3. Enter the command: licenseserver passwordreset

  This resets the LicenseServer administrator password to default.
- 4. You can now log in to the Administrator Web UI with the password default.

# Chapter 5

**Configuring the Server** 

## 5 Configuring the Server

This section includes FlowForce Server configuration instructions.

You can change the FlowForce Server settings either from the Web administration interface or by editing the FlowForce Server .ini files. Some server maintenance or configuration tasks are also available from the command line interface (see <a href="Command Line Interface">Command Line Interface</a>).

After editing the .ini files manually, you might need to restart the FlowForce services for the settings to take effect. If you need to restart services manually, see:

- Starting and Stopping Services (Linux)
- Starting and Stopping Services (Mac OS X)
- Starting and Stopping Services (Windows)

### 5.1 FlowForce Server Application Data

This topic describes the contents of the FlowForce Server application data directory. This information can be useful during manual data migration to a major FlowForce Server version or if you want to change some of the FlowForce Server configuration settings by editing .ini files.

The application data directory stores data generated by both FlowForce Server and its users, such as jobs, triggers, system functions, server logs, and other files.

The application data directory also contains several .ini style configuration files. Administrators can edit the .ini configuration files with a text editor, as an alternative to changing settings from the Web Administration Interface or from the Command Line Interface.

The path to the application data directory depends on the operating system and platform and is as follows.

Linux	/var/opt/Altova/FlowForceServer2017
Mac OS X	/var/Altova/FlowForceServer2017
Windows	C:\ProgramData\Altova\FlowForceServer2017

The following table lists the main files and folders in the application data directory.

flowforceserver.ini	Stores global configuration settings of FlowForce Server (currently, the language used in server logs and in error messages).  Note: Do not confuse this file with the flowforceserver.ini file stored in the data directory (described below).
flowforceweb.ini	Stores global configuration settings of FlowForce Web Server (currently, the language of the Web administration interface).  Note: Do not confuse this file with the flowforceweb.ini file stored in the data directory (described below).
flowforceweb.html	The setup page is used to register FlowForce Server with Altova LicenseServer and configure the default Web administration interface address and port (see also Opening the Setup Page). This page is regenerated when you restart the FlowForce Web Server service.
data/cache.db	This database file stores data related to the caching feature of FlowForce (see <u>Caching Job Results</u> ).
data/state.db	This database file stores the volatile (that is, not configured) state of FlowForce.
data/ffweb.log data/flowforce.log	These files store the log of messages from FlowForce Web Server and FlowForce Server, respectively. This is

	,
	applicable to Windows, Mac OS X, as well as Debian 7, Ubuntu 14, and CentOS 6 (or earlier) systems.
	On Debian 8, Ubuntu 15 and CentOS 7 or later, the log is written to the system log ( /var/log/syslog ).
data/flowforce.db	This database file stores the FlowForce Server object system, user data, active jobs, roles, and others.
data/flowforcelog.db	This database file stores the FlowForce Server logs.
data/flowforce.ini	This configuration file defines the port and listening interfaces of FlowForce Server. See also Configuration File Reference.
data/flowforceweb.ini	This configuration file defines the port and listening interfaces of FlowForce Web Server. See also Configuration File Reference.
data/files	Stores files associated with deployed functions.
data/logs	Contains captured output from job execution steps.
data/tmp	Stores temporary files.
data/tools	When other Altova server products (such as MapForce Server or StyleVision Server) are installed alongside FlowForce Server, this directory stores .tool files which enable FlowForce Server to work with these products.  If this directory is empty, you can copy manually any tool files from the installation directory of the respective product to this directory.
	You may need to edit a .tool file in order to set environment variables that may be required to execute MapForce mappings or StyleVision transformations (see Setting Environment Variables).

### 5.2 Opening the Setup Page

The FlowForce Server setup page allows you to specify various network settings, including the host and port on which FlowForce Server and FlowForce Web Server should listen. The setup page is regenerated every time you start the FlowForce Web Server service.

### **Prerequisites**

- The FlowForce Web Server service must be started if it is not running already (see how to start it on Linux, Mac OS X, and Windows).
- Make sure the random port address on which FlowForce Web Server was started is not blocked by your firewall.

### To open the setup page (Linux):

Do one of the following:

- If the Linux machine has a GUI browser, run the **flowforceweb.html** file (note this action requires root privileges):
  - file:///var/opt/Altova/FlowForceServer2017/flowforceweb.html
- Open the URL of the setup page in the browser. You can find the default URL in the flowforceweb.html file located in the application's <u>configuration folder</u> (the URL has the format http://127.0.0.1:34597/setup?key=52239315203).

#### To open the setup page (Mac OS X):

Do one of the following:

- In **Applications**, double-click the FlowForce Server icon (note this action requires administrative privileges). The FlowForce Server setup page opens in the Web browser.
- Open the URL of the setup page in the browser. You can find the default URL in the flowforceweb.html file located in the application's <u>configuration folder</u> (the URL has the format http://127.0.0.1:34597/setup?key=52239315203).

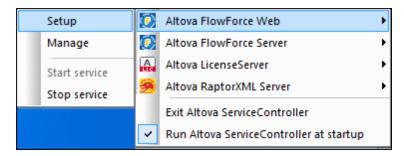
If the name of the Mac machine cannot resolve to an IP address (see <a href="Prerequisites">Prerequisites</a>), the browser opens a page with the following message: "FlowForceWeb does not appear to be available at <a href="http://<hostname>:<port>">http://<hostname>:<port>">http://<hostname>:<port>">http://<hostname>:<port>">http://<hostname>:<port>">http://<hostname>:<port>">http://<hostname>:<port>">http://<hostname>:<port>">http://<hostname>:<port>">http://<hostname>:<port>">http://<hostname>:<port>">http://<hostname>:<port>">http://<hostname>:<port>">http://<hostname>:<port>">http://<hostname>:<port>">http://<hostname>:<port>">http://<hostname>:<port>">http://<hostname>:<port>">http://<hostname>:<port>">http://<hostname>:<port>">http://<hostname>:<port>">http://<hostname>:<port>">http://<hostname>:<port>">http://<hostname>:<port>">http://<hostname>:<port>">http://<hostname>:<port>">http://<hostname>:<port>">http://<hostname>:<port>">http://<hostname>:<port>">http://<hostname>:<port>">http://<hostname>:<port>">http://<hostname>:<port>">http://<hostname>:<port>">http://<hostname>:<port>">http://<hostname>:<port>">http://<hostname>:<port>">http://<hostname>:<port>">http://<hostname>:<port>">http://</port>">http://<hostname>:<port>">http://</port>">http://</port>">http://</port>">http://</port>">http://</port>">http://</port>">http://</port>">http://</port>">http://</port>">http://</port>">http://</port>">http://</port>">http://</port>">http://</port>">http://</port>">http://</port>">http://</port>">http://</port>">http://</port>">http://</port>">http://</port>">http://</port>">http://</port>">http://</port>">http://</port>">http://</port>">http://</port>">http://</port>">http://</port>">http://</port>">http://</port>">http://</port>">http://</port>">http://</port>">http://</port>">http://</port>">http://</port>">http:///</port>">http:///</port>">http:///</port>">http:///</port>">http:///</port>">http:///</port>">http:///</port>">http:///</port>">http:///</port>">http:///</port>">http:///</port>">http:///</port

- a. Click the link mentioned in the message.
- b. In the browser's address bar, replace <hostname> with either localhost or the IP address of your Mac.

### To open the setup page (Windows):

Do one of the following:

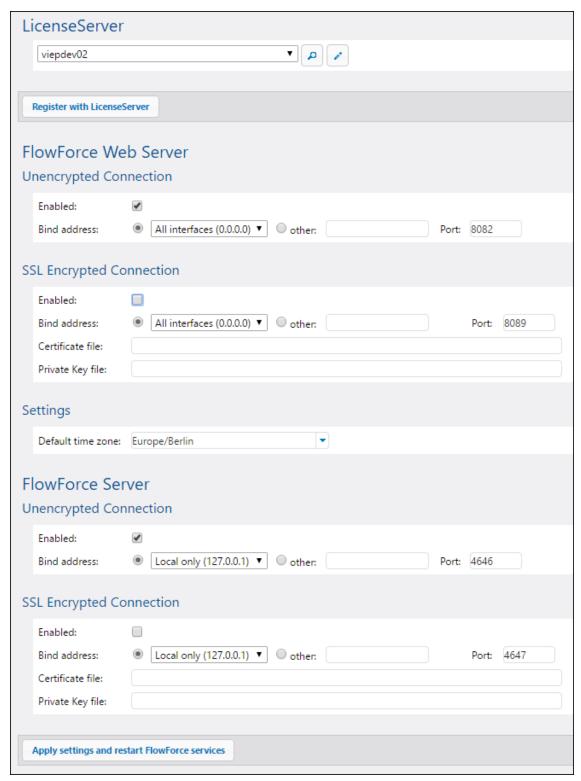
• Click the ServiceController icon ( ) in the system notification area, and then select Altova FlowForce Web > Setup.



Open the URL of the setup page in the browser. You can find the default URL in the flowforceweb.html file located in the application's <u>configuration folder</u> (the URL has the format <a href="http://127.0.0.1:34597/setup?key=52239315203">http://127.0.0.1:34597/setup?key=52239315203</a>).

### 5.3 Defining the Network Settings

You can specify the host and port on which FlowForce Server and FlowForce Web Server should listen, as well as other network-related settings, from the Setup page. The Setup page can be opened in various ways, depending on the operating system (see <a href="Opening the Setup Page">Opening the Setup Page</a>). Alternatively, most of these settings can also be defined by means of configuration files (see <a href="Configuration File Reference">Configuration File Reference</a>). The settings defined in the Setup page will be preserved when you install a new minor version of FlowForce Server. If you install a major version, the settings will be preserved only if you opted to migrate data from the previous major version during installation.



Setup page

The settings you can configure are as follows. (To understand the difference between "FlowForce

Web Server" and "FlowForce Server", see How It Works.)

#### LicenseServer

FlowForce Server must be registered with LicenseServer (see <u>Altova LicenseServer</u>). If you haven't specified a LicenseServer host during installation, enter here the address or host name of the machine where Altova LicenseServer runs. This can be either the address of the local machine (if LicenseServer is installed locally), or a network address.

#### FlowForce Web Server

This group of settings enables you to set the host (bind address) and port for the FlowForce Web Server, and specify whether it accepts unencrypted connections, SSL-encrypted connections, or both.

By default, plain HTTP (unencrypted) connections to FlowForce Web Server are enabled, unless you modified this during installation\* or from configuration files, see <a href="Configuration File Reference">Configuration</a>. On Windows, the Web administration interface is available by default on all network interfaces on port 8082 . On Linux and Mac OS, the port number is chosen randomly during installation. To specify a custom address other than "Local only" or "All interfaces", enter it in the "Other" text box.

\* Modifying the network settings during FlowForce Server installation is supported on Windows only.

To enable SSL-encrypted connections, first select the **Enabled** check box under "SSL Encrypted Connection", and then select the applicable interface where FlowForce Server should listen for SSL-encrypted connections ("Local only", "All interfaces", or a custom address). Secondly, next to **Port**, specify a port which is not already in use. Thirdly, enter the absolute path to your organization's certificate file and the certificate's private key in the corresponding text boxes. These must be provided by your organization's IT or similar department. Note that the certificate must be in PEM format. The file extension of PEM certificates is usually .pem but it can also be .cert, .cer, or .crt. The private key should not be password-protected, because of the complexities arising from having to enter this password from the FlowForce user interface. For security reasons, the certificate and the private key must be stored in a directory which is not readable by unauthorized users.

If your organization uses a self-signed digital certificate, as is often the case on internal company networks, each client operating system (or the browser, if applicable) must be configured to trust the certificate. Otherwise, the browser will display warnings such as "Your connection is not secure" or "There is a problem with this website's security certificate" when users access FlowForce Server through HTTPS (even though the connection itself will be encrypted). For information about adding a certificate to the trusted certificate store on Windows, see <a href="https://technet.microsoft.com/en-us/library/cc754841(v=ws.11).aspx">https://technet.microsoft.com/en-us/library/cc754841(v=ws.11).aspx</a>. For other operating systems, see:

- Trusting Server Certificates on Linux
- Trusting Server Certificates on Mac

Note the following:

- After the certificate has been added to the trusted certificates store, the browser will still display security warnings if the Common Name (CN) of the certificate does not correspond to the domain name or IP address where FlowForce Server runs.
- After the certificate has been added to the trusted certificates store, the browser will still display security warnings if it uses a certificate store other than that of the operating system (for example, Mozilla Firefox 52.0.1 on Windows 10). In this case, both your personal certificate (issued by some authority, for example, your company) and the certificate of the authority you want to trust (for example, your company) must be imported into the browser.

When you click **Apply settings and restart FlowForce services**, you will be redirected to the "https" (not the "http") URL, if you have enabled SSL encrypted connections. If the SSL settings are not configured correctly, you may no longer be able to access the Setup page. In this case, either use the unencrypted URL (if this is enabled), or change the SSL settings manually (see Configuration File Reference).

#### FlowForce Server

The default setting for FlowForce Server accepts only requests from the same machine (127.0.0.1) on port 4646, through an unencrypted connection. If you intend to start jobs as Web services via HTTP from remote machines, select "All interfaces (0.0.0.0)" from the **Bind address** combo box.

Enabling FlowForce Server to accept SSL-encrypted connections entails the same steps and requirements as those outlined above for FlowForce Web Server.

The unencrypted connection for FlowForce Server must not be disabled. An unsecured HTTP connection is required so that FlowForce Server can communicate with FlowForce Web Server.

### 5.4 Configuration File Reference

The network settings of both FlowForce Server and FlowForce Web Server can be configured either from the Setup page (see <u>Defining the Network Settings</u>) or by editing .ini configuration files, as described below.

There are two .ini files, one for FlowForce Server (**flowforceserver.ini**), and another one for FlowForce Web Server (**flowforceweb.ini**). The .ini configuration files can be found at the following path:

Linux	/var/opt/Altova/FlowForceServer2017/data/flowforceserver.ini /var/opt/Altova/FlowForceServer2017/data/flowforceweb.ini
Mac OS X	/var/Altova/FlowForceServer2017/data/flowforceserver.ini /var/Altova/FlowForceServer2017/data/flowforceweb.ini
Windows	C:\ProgramData\Altova\FlowForceServer2017\data\flowforceserver.ini C:\ProgramData\Altova\FlowForceServer2017\data\flowforceweb.ini

In the directory above, you can also can find two sample .ini files that contain comments and can be used as a template:

- flowforce.ini.template
- flowforceweb.ini.template

After editing the .ini files, remember to restart the corresponding service (FlowForce Server or FlowForce Web Server). For more information, see:

- Starting and Stopping Services (Linux)
- Starting and Stopping Services (Mac OS X)
- Starting and Stopping Services (Windows)

A sample **flowforceserver.ini** file looks as follows:

```
[Listen]
host=127.0.0.1
port=4646
active=1

[ListenSSL]
active=1
SSL=1
host=0.0.0.0
port=4647

[SSL]
certificate=/path/to/certificate.crt
private_key=/path/to/private_key.key
```

A sample **flowforceweb.ini** file looks as follows:

```
[Listen]
host=0.0.0.0
port=8082
active=1

[ListenSSL]
active=1
SSL=1
host=0.0.0.0
port=8083

[SSL]
certificate=path/to/certificate.crt
private_key=path/to/private_key.key

[FlowForce]
host=127.0.0.1
port=4646
```

The .ini files are organized into sections, as described below. Differences between both files are mentioned below where applicable.

### [Listen]

A [Listen] section defines the HTTP connection settings. It is possible to define multiple [Listen] sections. Each [Listen] section must begin with "Listen", for example [ListenSSL].

active	(Optional) Activates or deactivates this [Listen] section. Valid values:	
	<b>0</b> disabled	
	1 enabled	
	For example, active=1 means that HTTP connections are enabled.	
ssl	(Optional) Enables SSL support for this [Listen] section. Valid values:	
	0 disabled	
	1 enabled	
	To enable SSL support, set ssl=1 and also create a [SSL] section, as shown below.	
host	Specifies the host address of FlowForce (Web) Server, for example, 127.0.0.1. This can be an IPv4 or IPv6 address. Use	

	0.0.0.0 to listen on all interfaces. For local access only, use 127.0.0.1.
port	Specifies the port on which FlowForce (Web) Server will listen.

**Note:** FlowForce Server requires at all times one active HTTP interface through which it can communicate with FlowForce Web Server.

### [SSL]

This section defines the SSL/HTTPS connection settings.

certificate	Specifies the absolute path to the certificate file in PEM format.
private_key	Specifies the absolute path to the private key file.
certificate_chai	(optional) The path to the certificate chain file.

### [FlowForce]

This section is applicable only for FlowForce Web Server (the **flowforceweb.ini** file). It defines the connection details between FlowForce Web Server and FlowForce Server. Note that SSL connections are not supported between FlowForce Web Server and FlowForce Server.

host	Specifies the host address of FlowForce Server.
port	Specifies the port on which FlowForce Server will listen.

### [FlowForceWeb]

This section is applicable only for FlowForce Web Server (the **flowforceweb.ini** file).

timezone	Specifies the default time zone of FlowForce Web Server, for
	example timezone=Europe/Berlin

### 5.5 Setting the Default Time Zone

Whenever you create jobs that use time-based triggers, you must specify the applicable time zone. For convenience, you can configure globally what time zone should be selected by default in the job configuration page.

### To set the default time zone:

- 1. Click Administration.
- 2. Click Settings.
- 3. Under Input format, select the default time zone.
- 4. Click Save.

### 5.6 Setting Mail Parameters

If you are creating jobs that send emails (that is, if a job uses the FlowForce Server built-in <a href="mail/send">System/mail/send</a> function), you need to configure the SMTP address and port of the mail server, as well as the SMTP credentials.

FlowForce will first attempt to establish a connection encrypted over TLS or SSL. If the encrypted connection fails, FlowForce attempts to start communication without encryption, and then might elevate connection to encrypted if the SMTP server would explicitly require it. Otherwise, the SMTP connection remains in plain text.

### To change the mail settings:

- 1. Click Administration.
- Click Settings.
- Under SMTP Server and SMTP port, enter the name and port of the mail server, respectively. Standard SMTP servers accept connections on port 25. SMTP servers that require connection to be encrypted over SSL/TLS protocol accept connections on other ports, typically 465 or 587.
- 4. If your SMTP server requires authentication, click the plus icon next to **User** authentication and enter the username and password.
- 5. Optionally, enter a RFC2822-compliant mailbox address value in the **Default Sender** field. The value entered here is used as the default **From** parameter of the <a href="mail/send">mail/send</a> function.

### 5.7 Changing the Windows Active Directory Settings

If FlowForce Server is running on a Windows-based operating system, you can optionally integrate it with Windows Active Directory. From the FlowForce Server perspective, integration with Windows Active Directory means the following:

- Users can log on to FlowForce Server with their Windows user name and password.
- Administrators can either allow existing domain users to log on to FlowForce Server with their Windows credentials (that is, an implicit user import takes place), or they can explicitly import domain users and groups into FlowForce Server (see <a href="Importing Domain Users and Roles">Importing Domain Users and Roles</a>). In either case, the imported Windows accounts are visible in the user administration pages of FlowForce Server. This enables administrators to add or revoke privileges and permissions to Windows group or user accounts, in the same way as for the built-in FlowForce Server accounts (see <a href="How Privileges Work">How Privileges Work</a> and <a href="How Permissions Work">How Permissions</a> Work). Administrators can also assign FlowForce Server roles to Windows group or user accounts (see <a href="Assigning Roles to Users">Assigning Roles to Users</a>).
- Administrators cannot rename or change the password of Windows users imported into FlowForce Server.
- Administrators cannot rename or change the membership of Windows groups imported into FlowForce Server.
- Administrators can delete imported Windows accounts from FlowForce Server. This does
  not remove the accounts from the Windows domain and does not change in any way their
  associated Windows privileges.
- If the imported Windows accounts have FlowForce Server privileges and permissions assigned to them, they are displayed in privilege reports (see Viewing Privilege Reports).

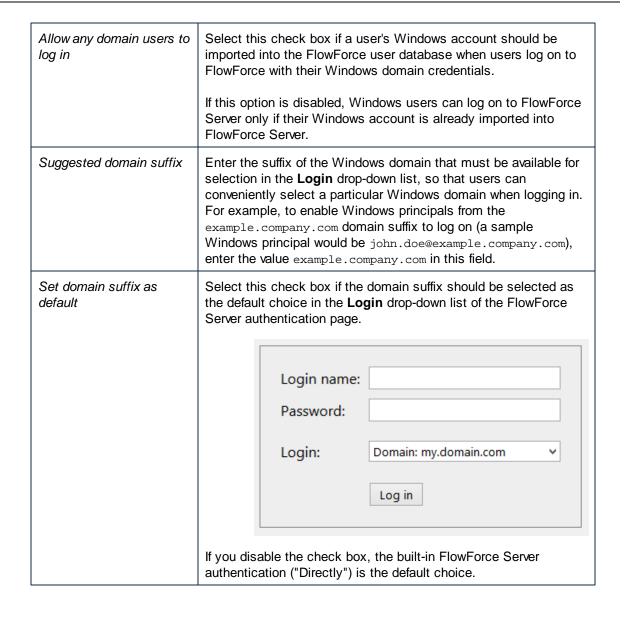
### To change the Windows Active Directory settings:

- Click Administration, and then click Settings.
- 2. Under Active Directory login, set the options as required, and then click Save.



The available settings are as follows:

Enable	Select this check box to enable users to log on to FlowForce Server with their Windows user name and password. This adds an additional drop-down list to the FlowForce Server login page, called <b>Login</b> . The drop-down list enables users to select the authentication option and contains the following items:  • Directly. This is the default FlowForce Server authentication option.  • [A specific Windows domain], as configured in the Suggested domain suffix option below.
--------	--



#### See also

• Importing Domain Users and Roles

### 5.8 Starting and Stopping Services (Linux)

The FlowForce Server solution consists of two services:

- 1. flowforcewebserver
- 2. flowforceserver

Run the commands below to start or stop the flowforcewebserver service. If you need to start or stop the flowforceserver service, replace flowforcewebserver with flowforceserver in the commands below.

#### To start the FlowForce Web Server service:

### To stop the FlowForce Web Server service:

```
[CentOS 6]: sudo initctl stop flowforcewebserver
[CentOS 7]: sudo systemctl stop flowforcewebserver
[Debian 7]: sudo /etc/init.d/flowforcewebserver stop
[Debian 8]: sudo systemctl stop flowforcewebserver
[RedHat]: sudo initctl stop flowforcewebserver
[Ubuntu <=14]: sudo initctl stop flowforcewebserver
[Ubuntu 15]: sudo systemctl stop flowforcewebserver
```

To check if a service is running, run the following command (replace servicename with either flowforcewebserver Or flowforceserver).

```
sudo service servicename status
```

### 5.9 Starting and Stopping Services (Mac OS X)

### To start the FlowForce Server service:

sudo launchctl load /Library/LaunchDaemons/com.altova.FlowForceServer.plist

### To start the FlowForce Web Server service:

sudo launchctl unload /Library/LaunchDaemons/
com.altova.FlowForceWebServer.plist

### To stop the FlowForce Server service:

sudo launchctl unload /Library/LaunchDaemons/com.altova.FlowForceServer.plist

### To stop the FlowForce Web Server service:

sudo launchctl unload /Library/LaunchDaemons/
com.altova.FlowForceWebServer.plist

### 5.10 Starting and Stopping Services (Windows)

By default, the FlowForce Server services are automatically started when Windows starts. Follow the instructions below if you need to manage services manually.

### To start the FlowForce Server service:

Click the ServiceController icon ( ) in the system notification area, and then select **Altova**FlowForce Server > Start service.

### To start the FlowForce Web Server service:

Click the ServiceController icon ( ) in the system notification area, and then select **Altova**FlowForce Web > Start service.

### To stop the FlowForce Server service:

Click the ServiceController icon ( ) in the system notification area, and then select **Altova**FlowForce Server > Stop service.

### To stop the FlowForce Web Server service:

Click the ServiceController icon ( ) in the system notification area, and then select **Altova**FlowForce Web > Stop service.

You can also start or stop the FlowForce Server services using the Microsoft Management Console (found under Control Panel > Administrative Tools > Services).

### 5.11 Localizing FlowForce Server

FlowForce Server is delivered with support for the following languages: English, French, German, Spanish, and Japanese. So you do not need to create a localized version of these languages. To set any of these languages as the default language, use FlowForce Server's <a href="mailto:setdeflang">setdeflang</a> command.

#### To create a localized version of FlowForce Server:

- 1. Generate an XML file containing the resource strings by using the exportresourcestrings command.
- 2. Translate the resource strings into the target language. The resource strings are the contents of the <string> elements in the XML file. Do not translate variables in curly brackets, such as {option} or {product}.
- 3. Contact Altova Support to generate a localized DLL file from your translated XML file.
- 4. After you receive your localized DLL file from Altova Support, save the DLL in the <INSTALLATION FOLDER>\FlowForceServer2017\bin folder. Your DLL file will have a name of the form FlowForceServer2017\_lc.dll. The \_lc part of the name contains the language code. For example, in FlowForceServer2017\_de.dll, the de part is the language code for German (Deutsch).
- 5. Run the <u>setdeflang</u> command to set your localized DLL file as the FlowForce Server app to use. Use the language code that is part of the DLL name as the argument of the setdeflang command.

### 5.12 Setting Environment Variables

When MapForce Server mappings or StyleVision Server stylesheets run under FlowForce Server management (as jobs), they may require environment variables to be set (for example, the CLASSPATH, in order to specify the location of the JDBC drivers when connecting to a database).

To set environment variables required by MapForce Server mappings or StyleVision Server transformations, edit the .tool file of the respective Altova server product. The location of the .tool file is as follows:

Linux	/var/opt/Altova/FlowForceServer2017/tools/
Mac OS X	/var/Altova/FlowForceServer2017/tools/
Windows	C:\ProgramData\Altova\FlowForceServer2017\data\tools\

The .tool file in the directory above takes precedence over the .tool file existing in other directories. If the .tool file does not exist in the directory above, copy it from the installation directory of FlowForce Server or that of the respective Altova server product.

You can add the required environment variables under the [Environment] section within the .tool file. For example, a .tool file which sets the CLASSPATH variable looks as follows:

```
[Environment]
CLASSPATH=.:/usr/local/jdbc/oracle/ojdbc6.jar
```

Sample .tool file (Linux)

Environment variables set in the .tool file override environment variables defined by other means.

For information about executing shell commands or scripts as FlowForce Server jobs, see the / system/shell/commandline function.

# **Chapter 6**

**Managing User Access** 

## 6 Managing User Access

This section includes procedures and concepts applicable to user access management in FlowForce Server.

### 6.1 Users and Roles

This section includes the following topics:

- Creating Users
- Creating Roles
- Importing Domain Users and Roles
- Default Users and Roles
- Renaming Users and Roles
- Assigning Roles to Users
- Assigning Roles to Other Roles

### 6.1.1 Creating Users

Users are persons who log on to FlowForce Server to configure jobs, deploy MapForce or StyleVision transformations, or manage the FlowForce Server. The actions available to users in FlowForce Server depend on the following:

- a) Their assigned permissions or privileges
- b) The permissions and privileges assigned to any roles that users are members of.

#### To add a FlowForce Server user:

- 1. Click Administration, and then click Users.
- 2. Click Create User.
- 3. Fill in the required fields.

User name	<ul> <li>Enter the name of the user. The following restrictions apply:</li> <li>It must not be empty</li> <li>It must not begin with or end with spaces</li> <li>The allowed characters are letters, digits, underscore (_), dash (-), and full stop (.)</li> </ul>
Password	Enter the user's password.
Re-type password	Re-type the user's password.
Change password on next login	If you select this check box, the user will be prompted to change password on next login.

- 4. Optionally, grant the required privileges to the user (for the description of available privileges, see <u>Privileges</u>). Note that you can grant privileges to users either directly from this page, or by assigning to them a role which already has some privileges. To simplify user maintenance, it is recommended to use the latter approach (see <u>Adding Roles</u> and <u>Assigning Roles to Users</u>).
- 5. Click Save.

### 6.1.2 Creating Roles

Roles are named sets of <u>privileges</u> that help enforce security based on the business need. The typical role-based security involves at least two roles: an administrator and a standard user. Each role is defined by the privileges granted to that role. For example, administrators can change their own password and that of other users, whereas standard users can change only their own password. You can assign roles to users and revoke roles from users as necessary.

#### To add a FlowForce Server role:

- 1. Click Administration, and then click Roles.
- 2. Click Create Role.
- 3. Enter the role name (for example, "Administrator").
- 4. Under **Privileges**, select the privileges that must be assigned to the role (for the description of available privileges, see **Privileges**).
- 5. Click Save.

### 6.1.3 Importing Domain Users and Roles

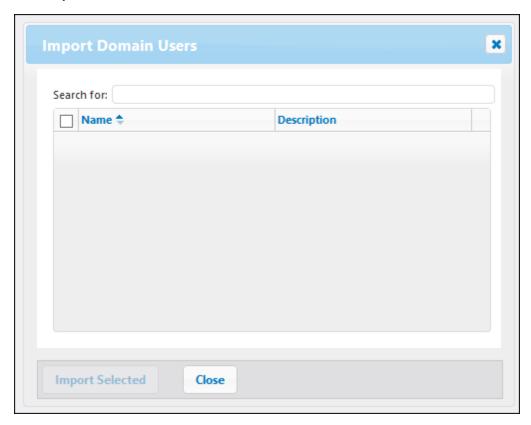
In addition to creating FlowForce Server users, you can import domain user accounts and groups from the Windows Active Directory.

When the **Allow any domain users to log in** setting is enabled in the <u>Windows Active Directory settings</u>, Windows domain users are able to log on to FlowForce Server even if you have not explicitly imported their accounts into the FlowForce Server database. To ensure that domain users log on to FlowForce server only if their Windows account has been explicitly imported by an administrator, disable the **Allow any domain users to log in** setting, and import the domain users as shown below.

**Note:** The local machine accounts are not part of Active Directory, so they cannot be imported into FlowForce Server.

#### To import Windows domain user accounts into FlowForce Server:

- Under Administration, click Users.
- 2. Click Import Domain Users.



3. Start typing the name of the user account you want to import. Partial searches are valid, so, for example, if you enter a value such as "ad", the accounts "Administrators", "Admanager", and "Admin" are retrieved from Active Directory and shown on the webpage dialog. FlowForce Server uses the Ambiguous Name Resolution (ANR) search algorithm that allows you to specify complex search conditions in a single clause. For example, you can retrieve the account of a person named Jim Smith by typing "ji sm". Refer to Microsoft documentation for further information about Ambiguous Name Resolution in

Active Directory.

4. Select the check box next to records that are to be imported, and then click **Import**Selected. Be patient while records are being retrieved from the Active Directory. Waiting time increases if the domain is not local.

### To import Windows domain groups into FlowForce Server:

- 1. Under Administration, click Roles.
- 2. Click Import Domain Roles.
- 3. Follow the steps 3-4 above.

#### See also

Changing the Windows Active Directory Settings

### 6.1.4 Default Users and Roles

#### **Default Users**

The following special users are predefined in FlowForce Server.

♣ root	This user is the initial, top-level FlowForce Server administrator. By default, it has all permissions and privileges available in the system.
4 anonymous	This is a special user account for users that do not explicitly log in. Anonymous access to the FlowForce Server Administration Interface is not possible, but you can enable anonymous access for certain services exposed as Web services (see <a href="Exposing Jobs">Exposing Jobs</a> as Web Services).

The built-in users cannot be deleted, although it is possible to change their privileges.

**Note:** User A root can change any privileges and permissions, including own permissions and privileges. Take extra caution when logged in as A **root** and editing root privileges, since you may unintentionally lose your own access to the system.

#### **Default Roles**

The following special roles are predefined in FlowForce Server.

& authenticated	This role includes all users who are authenticated using an existing user name and password. Every FlowForce Server user except user <b>anonymous</b> is a member of this role. By default, this role has the <i>Set own password</i> privilege.
& all	This role includes all FlowForce Server users, including user <b>anonymous</b> . By default, this role has no privileges.

Since the roles **authenticated** or **all** are built-in, you cannot explicitly assign these roles to users or revoke them from users. The membership of the built-in roles is automatically managed by FlowForce Server. Every time when you add a new user, FlowForce Server automatically assigns to the new user both the role **authenticated** and the role **all**.

If you want to change the privileges of any of the built-in users and roles, you should carefully analyze the potential impact. To get a global view of all currently assigned privileges, use privilege reports (see Privilege Reports).

### 6.1.5 Renaming Users and Roles

#### To rename a user:

- 1. Click Administration, and then click Users.
- 2. Click the user record you want to edit.
- 3. Enter the new name in the **User name** text box, and then click **Save**.

#### Notes:

- When a user name is changed, the currently assigned user password remains unchanged.
- If you are changing your own name (provided that you have this privilege), the changed name becomes effective as soon as you click Save, and is visible in the top right area of the page.

#### To rename a role:

- 1. Click Administration, and then click Roles.
- 2. Click the record you want to edit.
- 3. Enter the new role name in the **Role name** text box, and then click **Save**.

#### Notes:

- The members of a role do not change when the role is renamed.
- The default roles &all and &authenticated cannot be changed.

### 6.1.6 Assigning Roles to Users

### To assign one or more roles to a user:

- 1. Click Administration, and then click Users.
- 2. In the list of users, click the record you want to edit.
- 3. Under **Roles available**, select the roles that must be assigned to the user, and then click **Assign**.

#### To revoke one or more roles from a user:

- 1. Click **Administration**, and then click **Users**.
- 2. In the list of users, click the record you want to edit.
- 3. Under Roles assigned to user '<user name>', select the roles that must be revoked from the user, and then click Remove.

#### To assign a role to multiple users:

- 1. Click Administration, and then click Roles.
- 2. In the list of roles, click the record you want to edit.
- 3. Under **Users/Roles available**, select the users that must be assigned the role, and then click **Assign**.

#### To revoke a role from multiple users:

- 1. Click Administration, and then click Roles.
- 2. In the list of roles, click the record you want to edit.
- 3. Under **Members of role '<role name>'**, select the users from whom the role must be revoked, and then click **Remove**.

### 6.1.7 Assigning Roles to Other Roles

You can model the hierarchy of your organization or business within FlowForce Server by assigning roles to other roles. For example, you can create a role called **Employees** and a role called **Marketing Department**. Then you can assign the role **Marketing Department** to be a member of **Employees**. This means that all privileges and permissions granted to **Employees** will be automatically inherited by users who are members of **Marketing Department**.

### To assign a role to another role:

- 1. Click Administration, and then click Roles.
- 2. In the list of roles, click the role you want to assign to another role (for example, if you want the role A Marketing Department to inherit privileges from the role Employees, click "Employees").
- 3. Under Users/Roles available, select the role to be assigned, and then click Assign.

#### See also

- How Privileges Work
- How Permissions Work

### 6.1.8 Resetting the Root Password

In the event that you forgot or lost the password of the **Proot** user account, you can reset it to the default value from the command line interface (see the command resetpassword).

To perform root password reset, it is assumed that you have access to the operating system where FlowForce is running, including FlowForce binaries and data files. This is the same kind of access required when installing FlowForce or when migrating to a new FlowForce version or server manually.

When you perform a password reset, the privileges of the **aroot** user will also be restored to the default value (that is, all the privileges will be granted).

Performing a root password reset does not affect any FlowForce users except the **Proot** user.

#### **Privileges** 6.2

This section includes the following topics:

- How Privileges Work
- Viewing Privilege Reports

### 6.2.1 How Privileges Work

Privileges define what users can do in FlowForce Server (for example, set own password, read users and roles, stop any job, and so on). Privileges are different from <u>permissions</u> in the sense that permissions control user access to <u>containers</u>, whereas privileges are effective globally across FlowForce Server. The following simple rule might help you distinguish quickly between privileges and permissions: privileges are global, permissions are local.

Like permissions, privileges can be assigned both to individual users and to <u>roles</u>. Therefore, when users log on to FlowForce Server, their set of effective privileges is determined by:

- a) the privileges they have been assigned directly
- b) the privileges assigned to any roles that the user is member of.

The following privileges are available in FlowForce Server.

Maintain global settings	This privilege grants rights to change the FlowForce Server global settings available in the Settings page—that is, the time zone and the mail server settings. This is an administrative privilege and should only be granted to FlowForce Server administrators.
Maintain users, roles and privileges	This privilege grants rights to add, edit, and delete the following data:
	<ul><li>Users</li><li>Roles</li><li>Privileges</li><li>Passwords</li></ul>
	This is an administrative privilege and should only be granted to FlowForce Server administrators. By default, only the user \$\mathbb{A}\$ root has this privilege.
Override security	Users with this privilege can change container permissions without having "write" security permission. This allows FlowForce Server administrators to regain access to resources accidentally rendered inaccessible.
	This is an administrative privilege and should only be assigned to FlowForce Server administrators. By default, only the user \$\frac{1}{2}\$ root has this privilege.
Read users and roles	By default, users can see only their own user account and any roles they are member of. When granted this privilege, users can see all existing users and roles.
	By default, only the user Aroot has this privilege.
Set own password	This privilege grants to users the right to change their own password. Users who do not have this privilege need to have their password set by a FlowForce Server administrator.
	By default, the & authenticated role, and hence every user

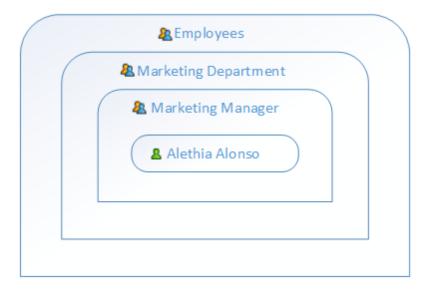
	account except 🚨 anonymous, has this privilege.
Stop any job	This privilege grants the right to stop any running FlowForce Server job, regardless of the user who created it.
View unfiltered log	By default, users can see log entries related to configurations to which they have "read" access. If granted this privilege, users can read all log entries, including those not associated with a specific configuration.  By default, only the user are root has this privilege.

#### Inheritance

You can assign privileges either directly to a user (for example, to Alethia Alonso), or to a particular role (for example, to Marketing Manager). The latter approach is recommended, because it simplifies management of privileges in the long term. For example, users may switch departments, or they might join or leave your organization. In either case, maintaining privileges for each individual user may become a counter-productive task. By assigning privileges to roles rather than users, you decrease granularity, simplify maintenance, and focus on the business need of each group or department rather than on individual users.

You can model the hierarchy of your organization or business within FlowForce Server by assigning roles to other roles. For example, you can create a role called **Employees** and a role called **Marketing Department**. Then you can assign the role **Marketing Department** to be a member of **Employees**. This means that all privileges and permissions granted to **Employees** will be automatically inherited by users who are members of **Marketing Department**.

Furthermore, you can assign the AMarketing Manager role to be a member of AMarketing Department role. In this case, the AMarketing Manager role will inherit privileges both from the AMarketing Department and from the Employees roles. When a new marketing manager joins your organization, Alethia Alonso, if she is assigned the Marketing Manager role, she will inherit all other privileges from the broader roles.



As the diagram shows, Alethia Alonso inherits permissions and privileges from the role Marketing Manager. This role, in its turn, inherits privileges from the Marketing Department, and so on.

In a newly installed FlowForce Server system, considering the <u>default users and roles</u>, the users and privileges diagram looks as follows.



As the diagram shows, every user in the system inherits the privileges defined in the all role. However, only existing users (in this case, aroot) inherit the privileges defined in the authenticated role. If you add any new users to FlowForce Server, they are automatically assigned to the all and authenticated role (and thus granted the privileges defined in those roles, if any), as follows.



#### See also

- Default Users and Roles
- Viewing Privilege Reports

### 6.2.2 Viewing Privilege Reports

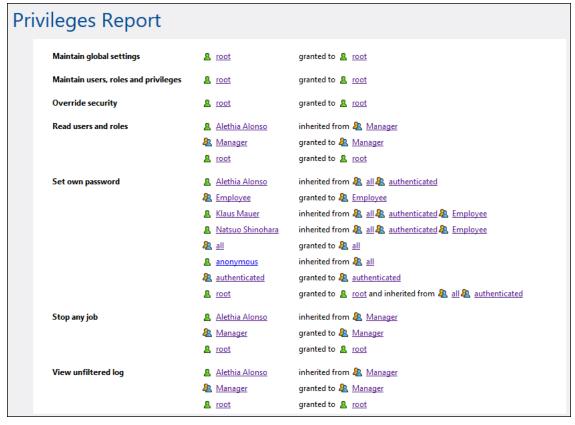
As a FlowForce Server administrator, you might find it difficult to keep track of privileges assigned to each and every role or user, especially when the number of users and roles increases. To help you get a quick overview of all privileges currently assigned to users and roles, FlowForce Server provides the following reports:

- Privileges Report
- Privileges by User Report

To view these reports, click **Administration**, and then click **Reports**.

### **Privileges Report**

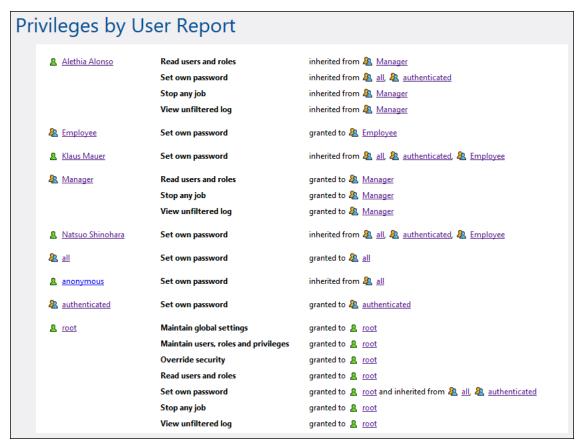
This report lists the FlowForce Server privileges. For each privilege, you can see the users who have been granted that privilege or inherited it by virtue of their roles.



Privileges Report (Sample)

#### **Privileges by User Report**

This report lists the FlowForce Server users. For each user, you can see the currently assigned privileges, and whether they have been granted or inherited.



Privileges by User Report (Sample)

### 6.3 Permissions and Containers

This section includes the following topics:

- How Permissions Work
- Understanding Containers
- Creating, Renaming, and Moving Containers
- Viewing Container Permissions
- Changing Container Permissions
- Restricting Access to the /public Container

### 6.3.1 How Permissions Work

Permissions control user access to <u>containers</u>. Like privileges, <u>permissions</u> can be granted both to users and to <u>roles</u>. Therefore, if a user is a part of a role, any permissions granted to the role will automatically apply to the user as well.

By default, permissions set on a container are inherited from the parent container. For example, let's assume that container A has a child container B. Users who have permission to access container A will have by default permission to access container B as well. However, an administrator can redefine the permissions of any user or role at every level of the container hierarchy.

FlowForce checks container permissions when users interact with containers. For example, users can view or change the contents of a container only if they have been granted the required permissions. Permissions are not evaluated upon job execution, therefore any permission changes will not apply retroactively to existing jobs.

For each FlowForce Server container, you can set the following permission types.

Container	The container per in the container.	The container permissions define what users can do with objects in the container.	
	Inherit	Provides to the user the same access rights to this container as those defined on the parent container.	
	Read	Grants the user rights to list the contents of the container.	
	Read, Write	Grants the user rights to list the contents of the container and to create or delete objects in the container. <b>Note:</b> To successfully create a new configuration object, or delete an existing one, users must be granted both the <b>Read, Write (Container)</b> permission and the <b>Read, Write (Configuration)</b> permission.	
	No access	Denies the user the right to enter the container (more specifically, the container appears to the user as disabled).	
Configuration	_	The configuration permissions define what a user can do with configuration objects (namely, jobs and credentials).	
	Inherit	Provides to the user the same configuration object-related rights as those defined on the parent container.	
	Read	Grants the user rights to view details about configuration objects within the container (such as the execution steps or triggers of a	

	T	
		job).
	Read, Write	Grants the user rights to modify any configuration object within the container (for example, edit the trigger of a job). Note: To successfully create a new configuration object, or delete an existing one, users must be granted both the Read, Write (Container) permission and the Read, Write (Configuration) permission.
	No access	Denies the user the right to view the details of any configuration objects within the container (more specifically, configuration objects appear to the user as disabled).
Service	The service permissions define access to a job exposed as a Web service, via the HTTP request interface.	
	Inherit	Provides to the user the same service—related rights as those defined on the parent container.
	Use	Grants the user rights to access the service and thus execute the job via the request interface.
		<ul> <li>Notes</li> <li>Service permission checks skip any container hierarchy checks. Therefore, if granted Use permission, users may use the service without having Read access to the container in which the corresponding job is defined.</li> <li>If you grant Use permission to user anonymous, the service becomes publicly available and does not require authentication.</li> </ul>
	No access	Denies the user the right to access the job as a Web service.
Credential	Inherit	Provides to the user the same credential—related rights as those defined on the parent container.
	Use	Grants the user rights to reuse any credentials defined in this container.
	No access	Denies the user the right to reuse credentials

		defined in this container.
Function	The function permission defines whether a user can invoke a function as an execution step in another function.	
	Inherit	Provides to the user the same function—related rights as those defined on the parent container.
	Use	Grants the user rights to call any function defined inside the container from another function.
	No access	Denies the user rights to call any function defined inside the container from another function.
Security	The security permission controls access to permissions of any child containers defined in the current container. <b>Note:</b> By default, users are permitted to read only permissions applicable to them (that is, any permissions assigned to themselves or any role they are a member of). However, users who have the <i>Read users and roles</i> privilege can read all permission entries.	
	Inherit	Provides to the user the same security—related rights as those defined on the parent container.
	Read Security	Grants the user rights to view the permissions of any child of the container.
	Read and Write Security	Grants the user rights to change the permissions of any child of the container.
	No access	Denies the user rights to view the permissions of any child of the container.

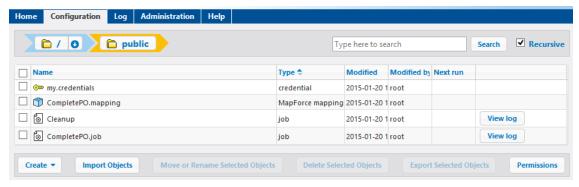
### 6.3.2 Understanding Containers

As the name of the term implies, a container is data packaged together. In FlowForce Server, containers can be roughly compared to folders on an operating system. Containers can contain any of the following: <a href="mailto:jobs">jobs</a>, <a href="mailto:credentials">credentials</a>, <a href="functions">functions</a>, and other containers. By setting <a href="permissions">permissions</a> on containers, you can control who can view or access the data inside them. Organizing data into containers and setting up the relevant permissions for each container is a good security practice.

The top-level container in FlowForce Server is the root ( / ) container. By default, the root container contains the following predefined FlowForce Server containers.

/public	The <b>/public</b> container is the default location where any FlowForce user can create jobs and credentials. It is by default empty and accessible to any FlowForce user. The <b>/ public</b> container serves as default location in the following cases:
	<ul> <li>When you deploy mappings from MapForce to FlowForce Server.</li> <li>When you deploy transformations from StyleVision to FlowForce Server.</li> </ul>
	You can, however, deploy mappings or transformations to a different container, if required.
/RaptorXML	This container is present if you licensed RaptorXML Server. It stores the validation and other functions specific to RaptorXML Server.
/system	The <b>system</b> container contains the FlowForce Server system functions. It is not recommended to make changes to this container.

You can navigate through containers from the Web administration interface, by clicking on a container to view its contents. The following screen shot shows a sample **/public** container that contains several configuration objects.



Sample FlowForce container

To go back to any container in the hierarchy, use the breadcrumb-style navigation available at the top of the page.

You can also search objects either within the current container including children objects (if the *Recursive* check box is checked) or only within the current container (if the *Recursive* check box is unchecked).

Containers contain objects such as jobs, deployed MapForce mappings or StyleVision transformations, functions, credentials. When you open a container, the following information is available about its objects:

Property	Description	
Name	Specifies the name of the object on the file system. Note that, when you create a new object, the name must not be already in use.	
Туре	Specifies the object type (such as credential, job, or function). You can also identify the object type by its accompanying icon:	
	⊚ Credential	
	Function (includes built-in functions, MapForce mappings and Sty transformations)	
	Container	
Modified	Specifies the date and time when the object was created or last modified.	
Modified by	Specifies the name of the user who modified the object.	
Next run	For jobs scheduled to run with time triggers, this column specifies the date and time of the next run, as defined in the job settings.	
View log	For jobs, this button provides quick access to the execution log of the corresponding job.	

Provided you have <u>permissions</u> to do so, you can create any number of additional containers to store your custom FlowForce server data (for example, one for each department). Alternatively, you can store data in the **/public** container, which by default is available to any authenticated user. If necessary, it is possible to restrict access to the **/public** container (see <u>Restricting</u> Access to the **/public** Container).

You can also move, rename, and delete any containers where you have the relevant permissions.

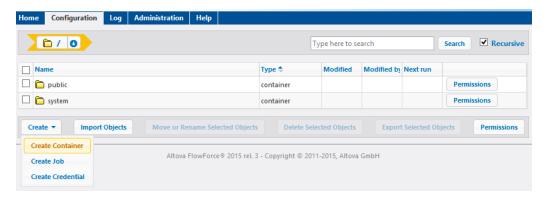
### 6.3.3 Creating, Renaming, and Moving Containers

You can create, rename and move containers if you (or any roles you are member of) have the *Container / Read, Write* permission (see also How Permissions Work).

**Note:** It is not recommended to modify the contents of the /RaptorXML and /system containers, which are provided by FlowForce Server by default.

#### To create a container:

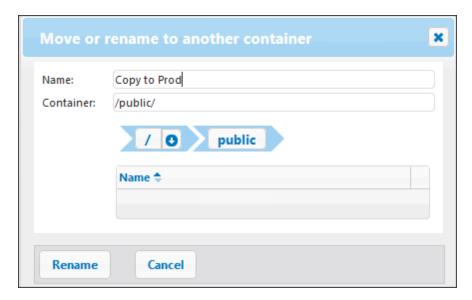
- 1. Click Configuration.
- 2. Click on an existing container under which you want to create a new container. If you want to create the container at the top level of the hierarchy, omit this step.
- 3. Click the **Create Container** button located in the lower left part of the page.



- **4.** Enter the name of the container. The following name restrictions apply:
  - It must not be empty
  - o It must not begin or end with space characters
  - It can contain letters, digits, single space, underscore (\_), dash (-), and full stop (.
     ) characters.
- 5. Click Save.

#### To rename a container:

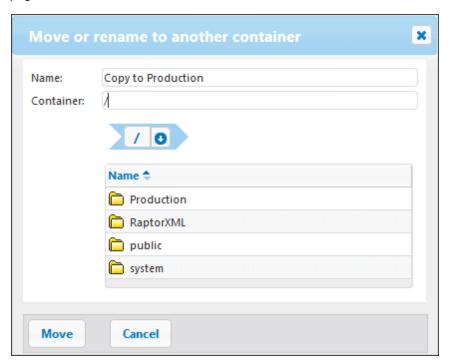
- 1. Click **Configuration**, and then navigate to the container you want to rename.
- 2. Click the **Move or Rename Selected Objects** button located in the lower left part of the page.



3. Enter the name of the container in the Name box, and then click **Rename**.

#### To move a container:

- 1. Click **Configuration**, and then navigate to the container you want to move.
- 2. Click the **Move or Rename Selected Objects** button located in the lower left part of the page.



- 3. Select the container's destination by doing one of the following:
  - Enter the path in the Container text box.
  - Use the interactive navigation controls to reach the destination container.

- 4. Optionally, set the new name of the container by typing it in the Name box.
- 5. Click Move.

### To rename or move multiple containers:

 Click the check boxes next to them, and then follow the same logic as for renaming or moving a single container.

### To select or deselect all objects in the container:

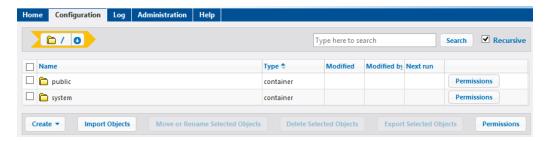
• Click the topmost check box.

### 6.3.4 Viewing Container Permissions

You can view the permissions of containers where you have the relevant permissions to do so (see also <a href="How Permissions Work">How Permissions Work</a>). By default, you can see your own permissions with respect to the container. If you are member of any role, you can also see the permissions available to roles of which you are member. If you have the privilege <a href="Read users and roles">Read users and roles</a>, you can also see the permission of other users and roles with respect to the container.

### To view the permissions of a container:

- 1. Click Configuration.
- 2. Do one of the following:
  - o Click the **Permissions** button adjacent to the container record.
  - Click the container record, and then click the click the Permissions button available in the lower right corner of the page.



The *User and Role name* column displays any users and roles whose permissions you have rights to see. The *Permissions* column displays what permission types are available to this particular user or role with respect to the container.



For the description of each permission type, see **How Permissions Work**.

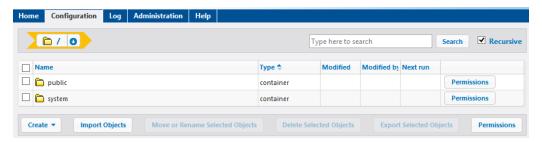
### 6.3.5 Setting Container Permissions

You can change permissions of containers where the following is true:

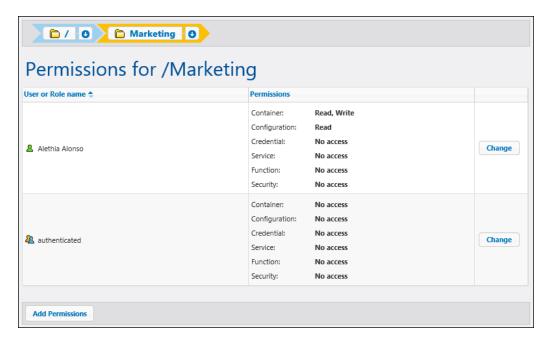
- You (or any roles you are member of) have the Security / Read and Write Security
  permission on the parent container relative to the one where you want to change
  permissions. For example, to change the permission of container "Jobs" which is a child
  of container "Marketing", you must have the permission Security / Read and Write
  Security on container "Marketing" (see How Permissions Work).
- You (or any roles you are member of) have been granted the privilege Override Security (see How Privileges Work).

### To change the permissions of a container:

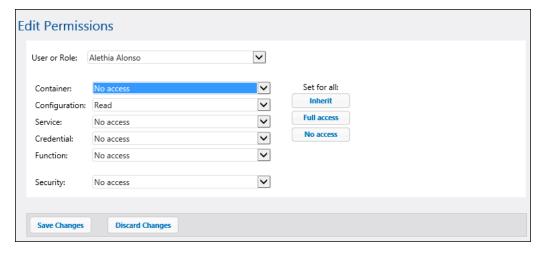
- 1. Click Configuration.
- 2. Do one of the following:
  - o Click the **Permissions** button adjacent to the container record.
  - Click the container record, and then click the click the Permissions button available in the lower right corner of the page.



- 3. Do one of the following:
  - To change the permissions of any of the listed users and roles, click the **Change** button next to the relevant user or role.
  - To add permissions for any users and roles that are not listed, click Add Permissions.



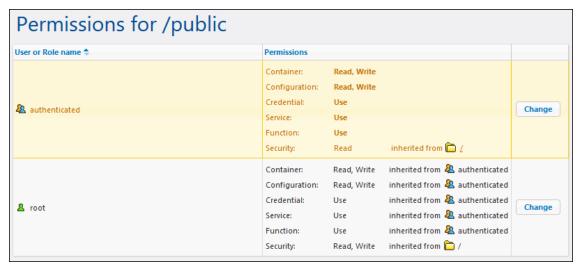
4. In the Edit Permissions section, select from the User and Role list the user or role whose permissions you want to change.



- Change each relevant group of permissions as required. For the description of each permission type, see <u>How Permissions Work</u>. If you want to modify all permission types with a single click, use the **Inherit**, **Full access**, and **No access** buttons.
- 6. Click Save Changes.

### 6.3.6 Restricting Access to the /public Container

The **/public** container (located under the top-level root container) is available by default in FlowForce Server. It acts as a location accessible to any FlowForce Server user and a location where any FlowForce Server user can store their data, without any predefined permissions. Therefore, by default, the **/public** container has the following permissions.



Default permissions of the /public container

This means that, by default, any FlowForce Server user with a user name and password (that is, any user who is member of the authenticated role) can do the following:

- Add, modify, and delete objects inside the /public container (namely, jobs, credentials, or other containers)
- Reuse any credentials available in the /public container
- Access as a Web service any job located in the /public container, provided that the job was configured to be available as a Web service
- Refer to any function available in the /public container
- Read the permissions assigned to the /public container

Note: These permissions may also be inherited by any containers that are children of the / public container. Normally, any new container inherits the permissions of the parent container; however, permissions may have been overridden by the root user, or by other users with relevant privileges.

You can restrict access to the **/public** container, if required. Note, however, that the <u>job</u> <u>configuration examples</u> included in this documentation assume the existence of the /public container.

#### To restrict access to the /public container:

- Revoke permissions on this container from the authenticated role (see <u>Setting</u> <u>Container Permissions</u>).
- 2. Create a new role and assign this role to all users who require permissions to the **/public** container (see <u>Creating Roles</u> and <u>Assigning Roles to Users</u>).

3. Assign to the new role only the required permissions (again, see <u>Setting Container</u> Permissions).

## 6.4 Password Policies

This section includes the following topics:

- How Password Policies Work
- Creating and Assigning Password Policies

### 6.4.1 How Password Policies Work

FlowForce Servers uses password policies to help administrators manage the complexity of user passwords. A password policy is a set of minimum requirements that a user password must meet in order to be valid (for example, at least *N* characters long).

The password complexity rules that you can define within a password policy are as follows:

- The total minimum length of the password (that is, the password must be at least N characters long to be valid)
- The minimum number of letters that the password must contain
- The minimum number of digits that the password must contain

You can define as many password policies as required (provided that you have the *Maintain users, roles and privilege* privilege). Once you define password policies, you can assign them to FlowForce users. A user account can have one password policy at a time.

When the user requests a password change, the system checks if the new password meets the complexity requirements defined in the user's password policy. If the password does not meet the complexity requirements defined in the password policy, the password change is denied, and the system displays a relevant message.

When an administrator changes the password of a user, FlowForce Server does not enforce the password policy. Also, if the password policy changes, any existing passwords remain unaffected. In the latter case, the password policy will be enforced when users attempt to change the existing password.

By default, FlowForce Server includes an empty password policy which does not enforce any password complexity rules. FlowForce Server implicitly assigns the default password policy to any user account that does not have a custom password policy. The default password policy cannot be changed.

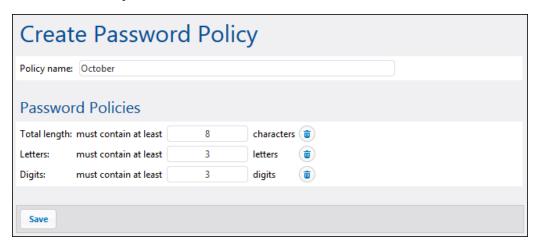
#### See also

• Creating and Assigning Password Policies

### 6.4.2 Creating and Assigning Password Policies

To create a new password policy:

- 1. Click Administration, and then click Password Policies.
- 2. Click Create Policy.



- 3. Enter the required password policy rules, and then click **Save**. The list of current users becomes available under the defined policy.
- 4. Click to select the user records that must be assigned the new policy, and then click **Assign**.



#### See also

• How Password Policies Work

# Chapter 7

Working with Jobs

# **7** Working with Jobs

This section includes procedures and concepts applicable to FlowForce Server job configuration.

Working with Jobs Credentials 171

### 7.1 Credentials

This section includes the following topics:

- How Credentials Work
- Defining Credentials

172 Working with Jobs Credentials

### 7.1.1 How Credentials Work

A credential record is the combination of user name and password associated with a user account on the operating system where the FlowForce Server job runs. When you define a job in FlowForce Server, you must supply the credentials with which the job must be executed. Note that if the user account associated with the supplied credentials does not have sufficient rights on the operating system, the job cannot execute successfully.

You can define credentials either every time when you create a new job (referred to as local credentials), or store them as a standalone object in a container (referred to as standalone credentials). In the latter case, you can refer to any standalone credentials while creating any job, provided that you have the relevant permissions. Standalone credentials are also convenient because you can update them easily when they change. By contrast, if you create local credentials every time when you define a new job, you need to manually update all jobs whenever the credentials change.

You can refer to standalone credentials not only from jobs, but also from job functions that require credentials as parameters. In particular, when you are using **ftp** functions in a job step, you can either specify credentials locally, or provide a reference to standalone credentials.

#### Notes:

- Users cannot refer to credentials stored in containers where they do not have the
  relevant permissions granted. To make credentials from a specific container
  accessible to a user or to a role, administrators must grant the *Credentials Use*permission to that user or role (see How Permissions Work).
- The credential password may be an empty string.
- As the clear text password needs to be sent to the operating system's login function, passwords are stored in a reversible encrypted form in the FlowForce Server database. The administrator should make sure to restrict access to the FlowForce Server's database file.
- File watch triggers also require credentials.

Working with Jobs Credentials 173

## 7.1.2 Defining Credentials

### **Prerequisites**

• Ensure that you have the *Container - Read, Write* and *Configuration - Read, Write* permissions granted on the container where you want to store the credentials.

• Ensure that the user account identified by the credentials has sufficient rights on the operating system. For example, if credentials are going to be referred in a job that writes to a directory, the user account must have rights to write to that directory.

#### To define credentials:

- 1. Navigate to the container where you want to store the credentials (for example, /public).
- 2. Click Create, and then Create Credential.
- 3. Enter the name of the credential as well as the operating system user name and password. To specify a user name in a Windows domain, use the form username@domain.
- 4. Click Save.

174 Working with Jobs Creating Jobs

## 7.2 Creating Jobs

This topic provides generic instructions on how to create jobs from the FlowForce Server Web administration interface. The generic instructions are meant to help you understand the structure of jobs and their available settings; however, not every step and setting listed below is applicable in real-life scenarios. Configuring jobs is a flexible process by definition and there might be more than one way to achieve the same result. To help you get an idea of various things you can do with jobs, this documentation also includes miscellaneous job configuration examples.

### **Prerequisites**

- Make sure that you have the following permissions for the <u>container</u> where you are creating the job (see <u>Viewing Container Permissions</u>)
  - o Container Read, Write
  - o Configuration Read, Write

### **Tips**

- Before creating a job, it might be a good idea to store the credentials of the operating system user account with which the job will be executed (see Defining Credentials).
- To avoid common errors when creating jobs, and especially if you intend to pass values between steps or between jobs, see The FlowForce Expression Language.

### To create a job:

- 1. Click **Configuration**, and then navigate to the container in which you want to create the job.
- 2. Click Create, and then Create Job.
- 3. Enter a job name, and, optionally, a job description.
- 4. If the job requires any values to be passed to it at runtime, create the required job input parameters (see Managing Input Parameters).
- 5. Under Execution Steps, add the steps of the job (see Managing Steps). Every job must have at least one step.
- 6. If the last step of the job returns a result, and if you intend to use the result in other jobs, select the return type under Execution Result (see <u>Declaring the return type of a job</u>).
- 7. If you want FlowForce Server to cache the returned result, specify the caching preferences (see Caching Job Results).
- 8. Under Triggers, add the trigger (or triggers) that will fire the job (see <u>Managing Triggers</u>). If the job will be exposed as a Web service, adding a trigger is not necessary.
- 9. If you want to expose the job as a Web service, click the **Make this job available via HTTP...** check box (see Exposing Jobs as Web Services).
- 10. Under Credentials, select an existing credential record or specify a local credential (see Credentials).
- 11. If the job returns a result that you want to use in other jobs or expose as a Web service, define the job's cache settings (see <u>Caching Job Results</u>).
- 12. Optionally, define the job's queue settings (see Defining Queue Settings).
- 13. Click **Save**. FlowForce Server validates the entered information and, if necessary, highlights in red any fields that require your attention.

Working with Jobs Duplicating jobs 175

# 7.3 Duplicating jobs

You can create copies of existing jobs when necessary. The copied job will have the same characteristics (execution steps, triggers, caching settings, and so on) as the original one. This will save you time, for example, when you need to quickly create a job using an existing one as a template.

### To create a copy of an existing job:

- 1. Open the existing job, and then click the **Save As** button at the bottom of the page.
- 2. Enter the name of the new job, and click Save As.



**Note:** If the credentials of the existing job are defined locally within the job, FlowForce will prompt you to enter the password again, for security reasons. If the credentials are defined as standalone credentials, this step is not necessary. For information about standalone versus local credentials, see How Credentials Work.

**Note:** If certain job components cause conflicts when the job is duplicated, FlowForce displays an error and does not duplicate the job. For example, if you attempt to duplicate a job containing a Web service, the service is already in use by the original job and cannot be duplicated. In this case, either change the URL of the Web service, or remove it completely.

Currently, the duplicated job is saved to the same container as the existing job. If you want to move it to a different container, you can do that as a two-step operation:

- 1. First, save an existing job with a new name, as described above. You can also create multiple copies of the same job without leaving the page.
- 2. Go to the parent container page, select one or more jobs, and then click **Move Selected Objects**.

## 7.4 Managing Input Parameters

In the context of a FlowForce Server <u>job</u>, input parameters represent some arbitrary information supplied to the job based on which the job will execute in a particular way. Job input parameters are similar to function arguments in a programming language. Input parameters can be of various types (for example, file or directory references, text, numbers, Boolean values, and others). For some job types, FlowForce Server creates the input parameters automatically.

#### Notes:

- FlowForce Server automatically adds an input parameter called triggerfile to jobs that use file system or HTTP triggers (see also <u>Managing Triggers</u>). The triggerfile parameter contains the name of the file that activated the trigger and must not be deleted.
- If a job is using parameters and is exposed as a Web service, FlowForce Server expects the parameters to be passed by the calling application in the request (see also <a href="Exposing Jobs as Web Services">Exposing Jobs as Web Services</a>).

An input parameter consists of the following fields.

Name	Mandatory field. Specifies the name of the input parameter.  Because you may need to refer to this parameter subsequently from any of the job's execution steps, use a name as descriptive as possible.	
Туре	Mandatory field. Specifies the data type of the input parameter, which can be one of the following:      String     String as file     String as directory     String as file or directory     Stream     Number     Boolean  You should select the data type according to the data type of the expected parameter value.	
Default	Optional field. Specifies the default value of the parameter. This value will be used if no value is specified by the job caller at runtime.	
Description	Optional field. Describes the purpose of the parameter. This description becomes available as a tooltip next to the parameter name, when you use the current job as an execution step of another job,	

Use the following buttons to manage parameters.

•	Add parameter.
---	----------------

<b>1</b>	Delete parameter
•	Duplicate parameter.
1 1	Move parameter up or down.
•	Undo a previous delete action.

178 Working with Jobs Managing Steps

# 7.5 Managing Steps

Steps define what the FlowForce Server job must actually do (for example, delete a file, execute a MapForce mapping, or send an email). In its simplest form, a step is just an operation with either failed or successful outcome, and it requires a <u>function</u> to be executed (see <u>Execution step</u>). However, there are also steps that provide means to execute other steps conditionally or in a loop (see <u>Choose step</u>, <u>Error/success Handling step</u>, and <u>For-Each step</u>). You can create as many steps as required within the same job, and you can set the order in which they must take place.

You can create the following step types.

Execution step	A <u>step</u> of type "Execution step" lets you execute a specific FlowForce function. Available functions include the <u>built-in</u> <u>functions</u> delivered with FlowForce Server, deployed MapForce mappings or StyleVision transformations, and the execution steps of other jobs.
Choose step	A step of type "Choose step" lets you define the conditions under which other job steps should be executed. Choose steps have the following structure:
	When {some expression} Execute (some step) Otherwise Execute (some other step)
	Under each "Choose step", you can nest other Choose steps (sub-conditions), for example:
	When {expression} When {expression} Execute (step) Otherwise Execute (step)
	Otherwise Execute (step)
	Any number of conditional steps can be defined.
	Within any when / Otherwise pair, FlowForce Server executes only the condition that is true; the other condition is ignored.
	For more information, see Processing Steps Conditionally.
For-Each step	A <u>step</u> of type "For-each step" lets you iterate through a sequence (for example, a list of files within a directory) and repeat an execution step any number of times. "For-Each" steps has the following structure:
	For each item in sequence {sequence expression}  Execute (step)
L	ı

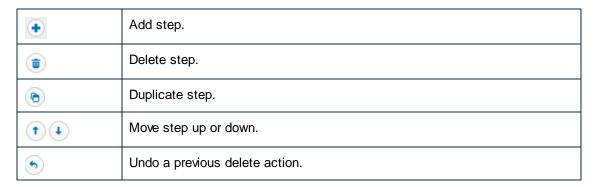
Working with Jobs Managing Steps 179

	FlowForce executes the step until it finishes looping through all items of the sequence expression.
Error/Success Handling step	When a step of a job fails, FlowForce Server aborts the job. You can use <u>steps</u> of type "Error/success handling step" to perform clean-up actions before the job finally exits (see <u>Handling Step Errors</u> ).

To add a step, click the button corresponding to the step type.



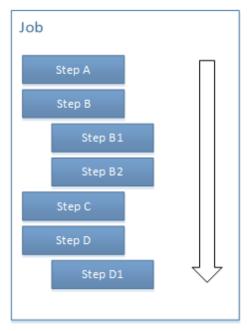
Use the following buttons to manage steps.



# 7.6 Processing Steps Sequentially

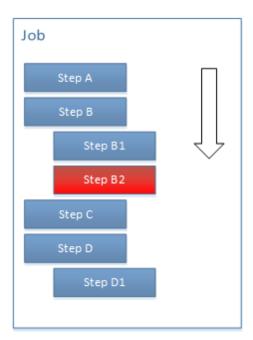
The simplest way to run a FlowForce job is to add to it a single step of type "execution step". Such a step can be used to call a FlowForce built-in function, a shell command or script, a MapForce mapping, a StyleVision transformation, a RaptorXML function, and so on.

If you add multiple execution steps to a job, FlowForce will process them sequentially, starting with the first (topmost) step down to the last step. This rule also applies to any sub-steps that a step may have. The diagram below illustrates this scenario.



Sequential processing

By default, if FlowForce encounters an error, processing stops at the step where the error was encountered, and any subsequent steps are not executed.



Sequential processing error

Sometimes, you might not want to break the execution of the whole job if a step fails. In this case, you can configure the function called by the execution step not to stop the job execution. All subsequent steps will then process normally even if a step fails. Note that this behavior can be applied only to steps that call the following functions:

- All functions under /system/filesystem,
- All functions under /system/ftp
- The system <u>\$\infty\$\system/mail/send</u> function
- The 🗊/system/shell/commandline function.

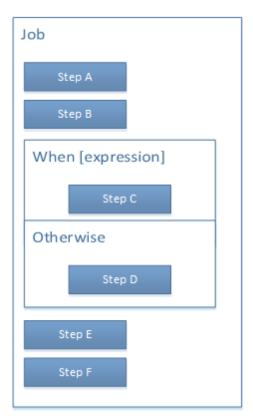
For more information, refer to the description of each function (specifically, the "Abort on error" parameter).

Sequential processing is just one of the ways to process FlowForce jobs and may not always suit your needs. For more advanced processing, see:

- Processing Steps Conditionally
- Handling Step Errors

# 7.7 Processing Steps Conditionally

To process steps depending on some condition, add a step of type "Choose". Such steps consist of two parts: a "when" part and an "otherwise" part.

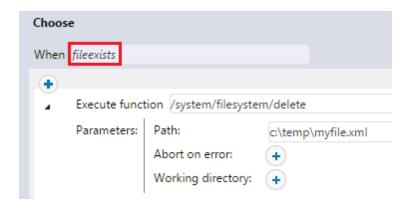


Abstract representation of a job which includes conditional processing

As shown in the image above, both the "When" part and the "Otherwise" part require an execution step. Also, the "when" part requires an expression. This expression dictates which of the two parts will be executed when the job runs. Namely, if the "When" expression evaluates to Boolean true, then Step C will be executed. Otherwise, step D will be executed. Therefore, the processing sequence of the abstract job illustrated above will be as follows (provided that there are no execution errors):

- 1. Step A
- 2. Step B
- 3. Step C or D, depending on the expression
- 4. Step E
- 5. Step F

For example, in the step illustrated below, FlowForce will execute the built-in delete function only if the expression "fileexists" evaluates to Boolean true.



Note that "fileexists" must have been declared in a previous step (or perhaps as an input parameter), for example:

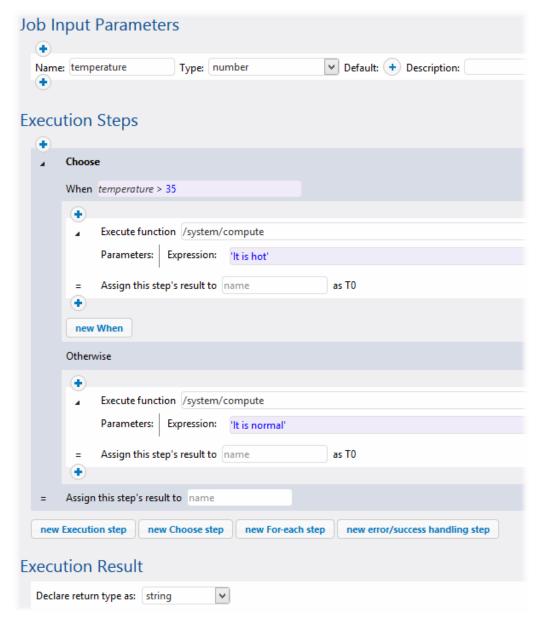
Assign this step's result to fileexists

Otherwise, FlowForce would display a syntax validation error:



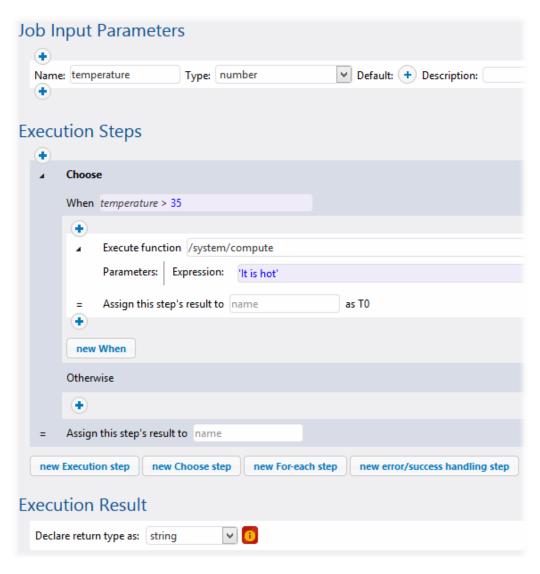
If you intend to pass the result of a "Choose" step to another step, or declare it to be of a particular type, ensure that each "When" and "Otherwise" condition produce the same return type. Otherwise, the return type of the "Choose" step is the *common* return type of all "When" and "Otherwise" branches, which might trigger validation errors.

For example, consider a job that checks the value of a numeric parameter **temperature** to tell you whether the weather is hot or normal. Such a job might look as follows:



In the job above, notice that **Declare return type as** is set to "string". If temperature is greater than a particular value (in this example, 35), a string expression is computed, namely "It is hot". Otherwise, the string expression "It is normal" is computed. So far, both the "When" and the "Otherwise" condition produce a string, so the job can be successfully saved and executed.

However, if we had left the "Otherwise" condition empty, the job would have triggered a validation error, as shown below.



The reason is that the expected return type of the job is "string" (as indicated by the **Declare return type as** setting). If the "Otherwise" branch is not defined, it computes to nothing. "Nothing" is not the same as a string; therefore, this triggers the validation error.

The jobs illustrated on this page use FlowForce expressions. For an introduction to FlowForce expressions, see The FlowForce Expression Language.

186 Working with Jobs Handling Step Errors

# 7.8 Handling Step Errors

When a step of a job fails, the job is aborted. To perform some clean-up actions before the job finally exits (such as logging or sending email notifications), you can create steps of type "error/success handling step". The error/success handling steps allow you to protect the execution of one or more steps, referred to as a "protected block". The following diagram illustrates the structure of a protected block.



Structure of a protected block

When any of the steps within the protected block fail, error handlers control what happens before the job finally exits. Error handlers can be one of the following:

- On Success (instructs the system to perform some action if all steps in the protected block were successful)
- On Error (instructs the system to perform some action if any of the steps in the protected block were not successful)
- Always (instructs the system to perform some action regardless of whether the steps in the protected block were successful)

When the protected block finishes executing, FlowForce Server executes any defined handlers based on the outcome. For example, on the diagram shown above, the protected steps are Step A and Step B, and the error handling logic is as follows:

• If A fails, then A, C and E will be executed.

Working with Jobs Handling Step Errors 187

- If B fails, then A, B, C and E will be executed.
- If A and B are successful, then A, B, D and E will be executed.

In practice, it is not necessary (although it is possible) to define all three handler types for every job that you want to handle. The most common scenario is to define only **On Error** and **Always** handlers.

### See also

Adding Error Handling to a Job (example)

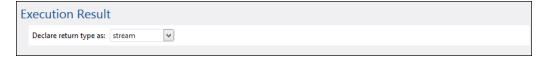
# 7.9 Declaring the Return Type of a Job

You may need to declare the data type returned by the job if you intend to use the result of the job in other jobs, or if you want to cache the result of the job. When declaring the return type, keep in mind the following:

- Declaring the return type is meaningful only for jobs that actually return a result.
- If you want to cache the result of the job, you must declare the return type.
- The return type of a job must be the same as the data type of the last step in the job. If the last step of your job returns a data type other than the one declared for the job, FlowForce Server returns an error similar to "Types string and result do not match". When type matching errors such as this one occur, use expression functions to change the data type of the last step in the job to the data type declared as the job return type (see Handling data types in steps).

### To declare the return type of the job:

- 1. Create a new job or open an existing one for editing.
- 2. Select a return type in the Execution Result section.



The available return types are as follows.

ignore/discard	This is the default return type. It instructs FlowForce Server to ignore or discard the result of the job provided there is one. Select this option if the job does not return a result, or if you do not need to process the returned result in any way.
string	Specifies that the data returned by the job represents a text value (string).
stream	Specifies that the data returned by the job represents a stream.
number	Specifies that the data returned by the job is a numeric value.
boolean	Specifies that the data returned by the job is a Boolean value.

## 7.10 Handling Data Types in Steps

When you pass the result of one step to another step or job, you will most likely need to change the data type of the step result. For example, if you create a step that lists the contents of a directory, the return type is **result**. In order to be able to process this result in a particular way (for example, to send it an email, or to write it to a file), you need to convert it from **result** to **stream** or to **string**.

You can change the return type of a step (let's call it **step 1**) as follows:

- 1. Type a value in the **Assign this step's result to** field of **step 1**. You will need to refer to this value later in order to access the result of the step.
- 2. Add a new execution step after step 1 (let's call it step 2).
- 3. Set the system/compute built-in function as execution function of step 2.
- 4. In the Expression field of step 2, type an expression which converts the value returned by step 1 into the required data type. For example, if the value returned by step 1 is a command line result that returns some output, and if you want to convert it to stream, use:

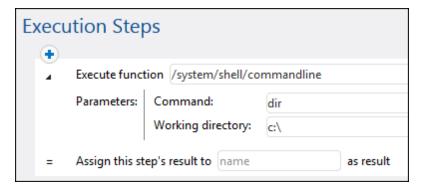
```
stdout(result)
```

Where result is the name you gave to the value returned by step 1.

The exact expression to use depends on the result you want to achieve. FlowForce Server provides various expression functions which you can combine with operators and produce meaningful expressions for use in steps. To understand the available options, see Step Result Functions.

### **Example**

Let's assume that you have created an execution step which lists the contents of a directory, as follows:



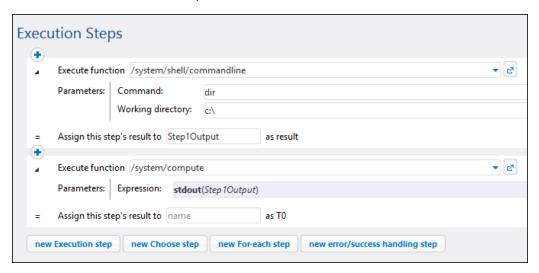
At this stage, the output of the step is a **result**. To use this result in other steps (which could, for example, write the directory contents to a file, or send an email with its contents), you need a return value of type **stream**, not **result**.

To convert the output of the directory listing step to "stream" data type:

1. Declare the result of the step 1 as **Step1Output**. To do this, type **Step1Output** in the

### Assign this step's result to field.

- 2. Add a new execution step which calls the system/compute function.
- 3. Under Parameters, enter the following expression: stdout(Step1Output), where step1output is the result of step 1.



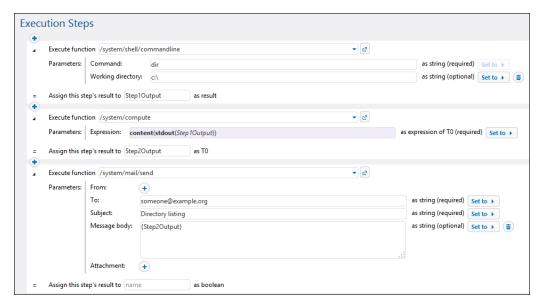
At this stage your job output data type is stream, as intended. You can now create a new execution step that will send the result of the second step to an email address.

### To send the directory listing to an email address:

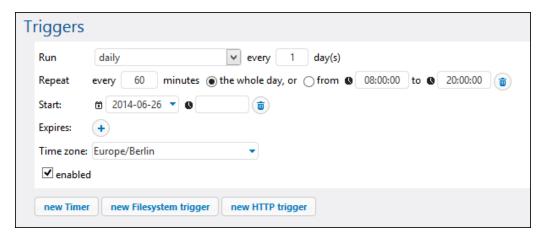
- Change the expression of step 2 from stdout(Step1Output) to content(stdout(Step1Output)). By doing this, you are converting the stream value into a string value.
- 2. Declare the result of step 2 as **Step2Output**. To do this, type **Step2Output** in the **Assign** this step's result to field.
- 3. Add a new execution step (step 3), with the following settings (note that the <u>mail server settings</u> must be configured before you can use the built-in <u>system/mail/send function</u>):

Execute function	Browse for the ①/system/mail/send function.	
From	Enter the sender's email address.	
То	Enter the destination email address.	
Subject	Enter the subject of the email, for example "Directory Listing".	
Body	Click Set to , and then select Step2Ouput.	

At this stage, the execution steps should look as follows:



4. Create a timer trigger that will run the job either once or periodically (for example, every 60 minutes).



5. Click Save.

# 7.11 Managing Triggers

When creating a job, you must specify the conditions (or criteria) that will trigger the job, known as triggers. FlowForce Server monitors any defined triggers and executes the job whenever the trigger condition is met.

You can create multiple triggers for the same job, and you can also enable or disable any of the defined triggers. Whenever any of the enabled triggers fire, FlowForce Server executes all steps of the job.

If you use triggers in jobs that have parameters, all parameters must have default values; otherwise, the job will not execute.

You can create the following types of triggers:

- Timer trigger
- File system trigger
- HTTP trigger

A timer trigger allows you to schedule jobs to start at a specific time and run for a specific time interval, with flexible recurring options (such as daily, weekly, on specific days of the week or month, and others).

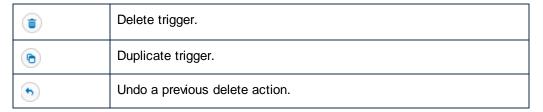
A file system *trigger* allows you to monitor a file or directory for changes such as newly added files or modified files (note that deleted files cannot be monitored). You can flexibly configure the directory polling interval (for example, every 60 seconds), and you can optionally set the start and expiry date of the trigger. You can also use wildcards to filter specific files of the directory.

An HTTP trigger allows you monitor a URI (Uniform Resource Identifier) for changes. Specifically, you can poll for changes the **Last-Modified** and **Content-MD5** HTTP header fields. You can flexibly configure the polling interval (for example, every 60 seconds), and you can optionally set the start and expiry date of the trigger.

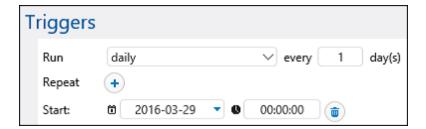
To add a trigger, click the button corresponding to the trigger type.



Use the following buttons to the right of a trigger to manage it.



Some trigger fields have the and buttons displayed next to them (for example, the start date of a timer trigger). You can use these buttons to set or clear the value of the trigger field. The value is considered set when it is visible in the page. For example, in the image below, the value of **Repeat** is not set, while the value of **Start** is set to "2016-03-29 00:00:00":



Note that, for the set trigger values to take effect, the job must be saved as well.

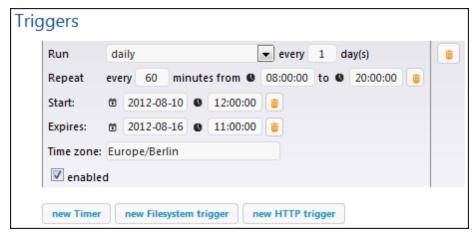
## See also

- Timer Triggers
- File System Triggers
- HTTP Triggers

## 7.11.1 Timer Triggers

A timer trigger allows you to schedule jobs to start at a specific time and run for a specific time interval, with flexible recurring options (such as daily, weekly, on specific days of the week or month, and others).

The following screen shot illustrates a sample timer trigger.



Sample Timer trigger

Timer triggers have the following structure.

Run	Defines whether the trigger should fire once or periodically every N number of days. The available options are as follows:  Once Daily On days of week On days of months On days of weeks of months	
Repeat	Defines the repeat options of the trigger. The repeat events occur on days specified in the <i>Run</i> drop-down list (see previous field).  The <i>every</i> field defines the repeat frequency, in minutes.  The <i>from</i> and <i>to</i> fields define the time range between repeat events.	
Start	Defines the trigger's starting date and time.  The Start date and time entries are mandatory if you selected Once from the <i>Run</i> drop-down list.  Clicking in the Date field opens a pop-up calendar from which you can select the start date.	
Expires	Defines the date and time when the trigger expires.	

Time zone	Defines the time zone applicable to the start and expiry date and time. The default time zone is defined in the server administration settings (see Setting the Default Time Zone).
Enabled	The <i>enabled</i> check box allows you to enable or disable the trigger. This option is useful when creating and testing new jobs.

## 7.11.2 File System Triggers

A file system *trigger* allows you to monitor a file or directory for changes such as newly added files or modified files (note that deleted files cannot be monitored). You can flexibly configure the directory polling interval (for example, every 60 seconds), and you can optionally set the start and expiry date of the trigger. You can also use wildcards to filter specific files of the directory.

The following screen shot illustrates a sample file system trigger.



Sample file system trigger

File system triggers have the following structure.

Check	Specifies how the trigger should poll the directory or file. Valid options:	
	Modified date	The trigger checks the last modification timestamp files. If any dates have changed, or a new file has b trigger fires.
	Content	This option computes and stores a hash code for the After the polling interval has passed, the hash code and compared to the stored value. If there is a differ fires. Note that this can place considerable load on If any dates have changed, or a new file has been a trigger also fires.
Polling interval	Specifies the frequency, in seconds, with which the directory will be polled.	
Wait N seconds to settle	Defines the time in seconds that the server will wait before starting the next job.	
Start	Defines the triggifield.	ger's starting date and time. This is an optional
	Clicking in the I can select the s	Date field opens a pop-up calendar from which you start date.
Expires	Defines the date and time when the trigger expires.	
Time zone	Defines the time zone applicable to the start and expiry date and time. The default time zone is defined in the server administration settings (see Setting the Default Time Zone).	
Enabled		eck box allows you to enable or disable the trigger. seful when creating and testing new jobs.

### The "triggerfile" parameter

Whenever you create a file system or HTTP trigger, FlowForce Server automatically adds a **triggerfile** input parameter to the job. When the job runs, FlowForce Server set this parameter to:

- a) the file that triggered the job (in case of file system triggers)
- b) the name of the temporary file that contains the downloaded content of the polled URI (in case of HTTP triggers).



You can pass the value of the **triggerfile** parameter as input value in any subsequent steps of the job. This way, you can user or process the triggering file as required.

By default, the **triggerfile** parameter contain the absolute path of the triggering file. To extract portions of the path, use the file path expression functions.

For an example of a job that uses the triggerfile parameter, see Creating a Directory Polling Job.

## 7.11.3 HTTP Triggers

An HTTP trigger allows you monitor a URI (Uniform Resource Identifier) for changes. Specifically, you can poll for changes the **Last-Modified** and **Content-MD5** HTTP header fields. You can flexibly configure the polling interval (for example, every 60 seconds), and you can optionally set the start and expiry date of the trigger.

The following screen shot illustrates a sample HTTP trigger.



Sample HTTP trigger

HTTP triggers have the following structure.

Check	Specifies how the	he trigger should poll the URI. Valid options:
	Modified date  Content	Instructs the system to check the Last-Modified the Last-Modified HTTP header is missing, the cheader is checked (see next option). Instructs the system to check the Content-MD5 op header. This is a 128-bit "digest" used as a messay of the MD5 header has changed after the polling into then the trigger fires. If the header is not provided by content is retrieved and hashed locally.
Polling interval	Specifies the frequency, in seconds, with which the URI will be polled.	
Wait N seconds for settle	Defines the time in seconds that the server will wait before starting the next job.	
Start	Defines the trigger's starting date and time. This is an optional field.	
	Clicking in the Date field opens a pop-up calendar from which you can select the start date.	
Expires	Defines the date and time when the trigger expires.	
Time zone	Defines the time zone applicable to the start and expiry date and time. The default time zone is defined in the server administration settings (see Setting the Default Time Zone).	
Enabled		eck box allows you to enable or disable the trigger. seful when creating and testing new jobs.

### The "triggerfile" parameter

Whenever you create a file system or HTTP trigger, FlowForce Server automatically adds a **triggerfile** input parameter to the job. When the job runs, FlowForce Server set this parameter to:

- a) the file that triggered the job (in case of file system triggers)
- b) the name of the temporary file that contains the downloaded content of the polled URI (in case of HTTP triggers).



You can pass the value of the **triggerfile** parameter as input value in any subsequent steps of the job. This way, you can user or process the triggering file as required.

By default, the **triggerfile** parameter contain the absolute path of the triggering file. To extract portions of the path, use the file path expression functions.

For an example of a job that uses the triggerfile parameter, see Creating a Directory Polling Job.

## 7.12 Exposing Jobs as Web Services

You can expose FlowForce Server jobs as Web services. Jobs exposed as Web services are primarily meant to be accessed programmatically. For testing and debugging purposes, however, you can also invoke such jobs from a browser.

### To make a job available as a Web service:

- 1. Create a new job or open an existing one for editing.
- 2. Click to select the Make this job available via HTTP... check box.



3. Type the name of the Web service in the myURL text box.

At the request URL specified, FlowForce Server starts a listener service which accepts HTTP GET and POST requests. Jobs exposed as services remain active as long as FlowForce server is running.

When the service is invoked, FlowForce Server runs the job execution steps specified and returns one of the following:

- The first result file of the last step, if the job produces a result file;
- The standard output of the last step, if no result files are produced (this might be the case when you are working with command line output).

A valid result is returned with a HTTP 200 status, with the **Content-Type** header set according to the result. The **Content-Type** header depends on the actual result. A MapForce mapping will result in text/xml if it has XML output, or text/plain for text output. Standard output of other functions is also returned as text/plain. The result is returned as the response body, without any embellishments.

Execution errors are reported as HTTP 5xx status with a generic error message. For further information, check the FlowForce Server log (see Viewing the Job Log).

### To view all currently running FlowForce Server Web services:

Open your browser and enter the following URL in the address bar.

http://[FlowForceServer][ServerPort]/service/\*

[FlowForceServer] and [ServerPort] refer to the network address and port where FlowForce Server is running. By default, FlowForce Server runs on <a href="http://localhost:4646">http://localhost:4646</a> (assuming you are accessing it from the same machine). The server name and port are as defined in the administration page (see <a href="Setting the Network Address">Setting the Network Address</a> and <a href="Port">Port</a>).

### Web service parameters

When you expose a job as Web service, all job parameters automatically become parameters for the service. If a job parameter does not have a default, it is mandatory and must be provided when invoking the service. Optional parameters take the default value if not provided in the request.

When the service is invoked, FlowForce Server verifies the parameters supplied in the request against those defined in the job. If parameter validation fails, FlowForce Server returns a 5xx HTTP status. In this case, FlowForce Server also displays an HTML parameter form, for debugging and testing purposes. You can use the HTML form to enter manually any parameters, for debugging and testing. For each parameter of type *Stream*, a Browse button becomes available in the page, and you can use it to upload the file required as parameter.

To display the testing HTML form unconditionally, supply the built-in parameter **showform** in the request (with any value).

Requests can generally be sent as both HTTP GET or HTTP POST (with **Content-Type** multipart/form-data), with the exception of parameters of type **stream**, which are only supported for HTTP POST requests.

If you use Internet Explorer to test FlowForce Server jobs exposed as Web services, disable the "Show friendly HTTP error messages" option in the **Advanced** tab.

#### Authentication

To authenticate users, FlowForce Server uses HTTP Basic authentication. User credentials are checked against the FlowForce Server user database (the same user name and password used to log on to FlowForce Server Web administration interface).

To make a Web service available without credentials, grant the *Use Service* permission to the default **anonymous** user (see also <u>How Permissions Work</u>).

You can still supply HTTP credentials when a service is available for anonymous use. The credentials are then checked against the FlowForce Server user database and the service execution is attributed to the authenticated user instead of user anonymous.

If you supply invalid credentials, the request interface returns an HTTP status of 401. If you did not supply credentials and service use has not been granted to anonymous on this service, the request interface also returns an HTTP status of 401.

If you supplied valid credentials, but the authenticated user is not granted *Use Service* permission on this service, the request interface will return an HTTP 4xx failure status . If you try accessing a service that does not exist, an HTTP 4xx failure status is returned.

#### **Queue settings**

Service execution behaves like execution via trigger, and is subject to the same queue constraints (see Defining Queue Settings).

### See also

Exposing a Job as a Web Service (example)

202 Working with Jobs Caching Job Results

# 7.13 Caching Job Results

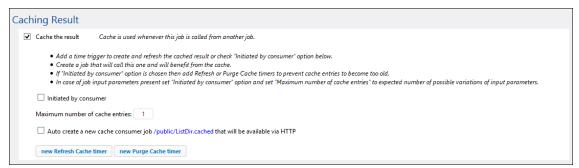
Caching reduces both the server load and the response time of jobs. It is particularly useful for jobs exposed as a Web service or for data-intensive jobs that require a long time to complete and increase the server load.

Caching the result of a job essentially means that FlowForce Server prepares and stores the job result in some internal repository (that is, the cache). If the job has parameters, the system creates a cache entry for every parameter combination, up to a limit that can be configured. When the job with cached result is called from another job (referred to as the consumer), FlowForce Server returns the cached result to the consumer (instead of executing the job again), thus reducing the response time.

The right balance between keeping the server load under control, on one hand, and not letting the cache become too old, on another hand, depends on the specifics of your environment and the jobs used. To achieve this balance, you can experiment with the caching settings provided by FlowForce Server and then decide what works best for you.

When working with cached job results, note the following:

- It is mandatory to declare the data type returned by the job whose result is cached (see Declaring the return type of a job).
- Both the job whose result is cached and the consumer job must use the same credentials. If the credentials differ, then the job executes as if no cache were defined.
- When you change the configuration of the cached job, the existing cache data is invalidated (flushed).



Caching settings

The available caching settings are as follows.

Cache the result	Select this check box if you want the job results to be cached. By doing so, you are instructing any consumers of the current job to read the cached result rather than execute the job. If the current job is executed directly (not through a consumer), either because a defined trigger fired or because the job's Web service is invoked, FlowForce Server refreshes the cache (or, if the job parameters are not found in the cache, it creates a new cached entry based on the supplied parameter combination).
Initiated by consumer	When this option is enabled, any job that is calling the current job

Working with Jobs Caching Job Results 203

	(that is, the <u>consumer</u> job) will compute and populate the cache if it does not exist. Otherwise, only triggers and Web service calls will populate the cache.		
Maximum number of cache entries	This option restricts the number of cached job results per job.  When the job has parameters, you might want to set this option to the number of all possible parameter combinations (provided the combinations are finite).		
Auto create a new cache consumer job	A cache consumer job is a Web service at the HTTP address you specify. The consumer Web service acts as a convenient way to retrieve and manage the cache of the job whose result is being cached. When invoked, the consumer job attempts to use the cached result of the main job in first place. If there is no cached result and the <b>Initiated by consumer</b> option is disabled, the consumer retrieves the actual result returned by the main job. If there is no cached result and the <b>Initiated by consumer</b> option is enabled, the consumer retrieves the actual result returned by the main job and also populates the cache.		
Refresh Cache timer	A Refresh Cache timer controls how often the system should refresh the cache of the current job. All currently cached parameter combinations are refreshed.		
	Perform Refresh daily every 1 day(s) Repeat Time: 15:08:00  Time zone: Europe/Berlin enabled  new Refresh Cache timer new Purge Cache timer		
Purge Cache timer	A Purge Cache timer controls how often the system should purge the cache of the current job.		
	Perform Purge daily every 1 day(s)  Repeat +  Time: 20:00:00    Time zone: Europe/Berlin   enabled  new Refresh Cache timer new Purge Cache timer		
Save and Refresh the	Click this button to refresh the cache manually.		

204 Working with Jobs Caching Job Results

cache	
-------	--

Refresh Cache timers and Purge Cache timers can be deleted in the same way as other triggers, by clicking the **Delete** ( ) button. Also, the **Undo Delete** ( ) operation becomes available for them until you save or reload the page. The **Duplicate** button ( ) enables you to create a copy of the current trigger, with the same settings.

#### See also

• <u>Caching Job Results</u> (example)

# 7.14 Defining Queue Settings

To control usage of server resources, you can limit the number of instances running in parallel for each job. In particular, you can set the minimum time interval (in seconds) between jobs and the maximum number of job instances allowed to run simultaneously.

Queue settings		
Minimum time between runs:	0	seconds
Maximum parallel runs:	1	instances

Queue settings

Minimum time between runs	Enter the time in seconds that must pass after each of the parallel jobs starts before another one may start.
Maximum parallel runs	This option defines the number of times the same job may be executed in parallel on the server.
	Parallel processing depends on the number of server cores available to FlowForce Server. For example, if you set this value to 12 and only one server core is licensed, 12 jobs will appear as running in parallel; however, only one job will make progress at any given time.
	If you are testing parallel service execution from your browser, be aware that the browser may hold back additional requests that address the same URL. As a workaround, you can initiate parallel instances of the same job from two different browsers.

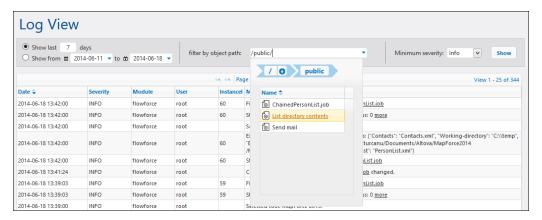
206 Working with Jobs Viewing the Job Log

# 7.15 Viewing the Job Log

You can quickly find out the details logged by FlowForce Server with respect to a particular job from the Log View. You can access the Log View either by clicking the **Log** menu item, or from various other contexts where a **View Log** button is displayed.

#### To view the job log:

1. Expand the Filter objects by path combo box, and browse for the location of the job.



- 2. Optionally, change the date range of the log and the minimum severity. (By default, the system displays the log of the last 7 days, with minimum severity "Info".)
- 3. Click Show.

### To access the log directly from the job details page:

- Click the View Log button available in the top right corner of the page. As a result, you
  are redirected to the Log View page, and the Filter objects by path combo box displays
  the name of the job.
- 2. Optionally, change the date range of the log and the minimum severity. (By default, the system displays the log of the last 7 days, with minimum severity set to "Info".)
- 3. Click Show.

Working with Jobs Stopping Jobs 207

# 7.16 Stopping Jobs

You can stop any currently running job provided that you (or any roles you are member of) have the *Stop any job* privilege (see How Privileges Work).

**Warning:** Since stopping a job may cause data corruption, it must be done only in exceptional circumstances.

When you stop a job, FlowForce Server first attempts to terminate the job gracefully. While it is not possible to predict the outcome of a stopped job, a graceful termination will normally attempt to perform the error handling tasks associated with the job (if any have been defined). If graceful termination is not possible, FlowForce terminates the job forcefully, after waiting for some time. To force FlowForce Server to terminate the job forcefully earlier, you can click the **Force stop job** button at any time.

### To stop a running job:

1. Click **Home**. Any currently running jobs are displayed in Running Jobs page section.



2. Click **Stop job**. FlowForce Server prompts you to confirm the action:



3. Click **OK**. The system attempts to stop the running job. If the job instance is stopped successfully, a relevant notification message is displayed and the job record is highlighted accordingly.



4. If, due to any reason, the job instance could not be stopped normally, click **Force stop job**.

## 7.17 Importing and Exporting Jobs

You can export jobs and other configuration objects (including deployed MapForce mappings and StyleVision transformations) from FlowForce Server as follows:

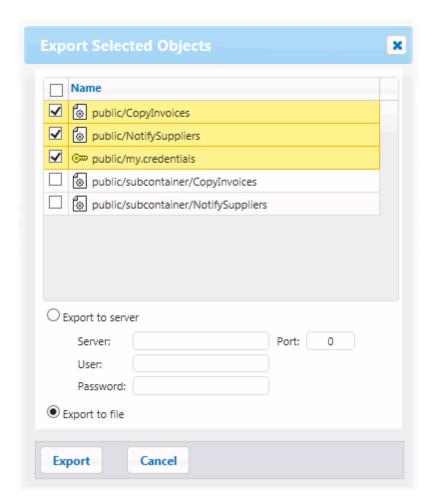
- To another running FlowForce Server instance (online export)
- To a file (offline export)

When you export objects to another running FlowForce Server instance, the exported objects become immediately available in the Web administration interface of that server.

When you export objects to a file, FlowForce Server creates a .zip archive which contains the selected objects and their dependencies. The .zip archive is named according to the date and time when the export operation took place. The naming convention is *export\_YYYYMMDDThhmmss*. For example, a file exported on 6th of August 2014 at 10:51:33 server time would be named *export\_20140806105133.zip*.

You can subsequently import the .zip archive either into the same FlowForce Server instance (provided the imported objects no longer exist at destination, or you want to overwrite them), or into another instance.

You can export either specific records within a container, or entire containers. In either case, FlowForce displays a dialog box that enables you to review the list of records before exporting them. If you selected an entire container for export, this dialog box displays all the children records of the selected container (jobs or credentials).



Before exporting objects, FlowForce informs you on a separate page about all objects that are dependent on (or are referenced by) the objects that you wish to export. This helps you see at a glance if there are missed dependencies. If you are exporting objects to a running FlowForce Server, you can also see whether each object already exists on the destination server.

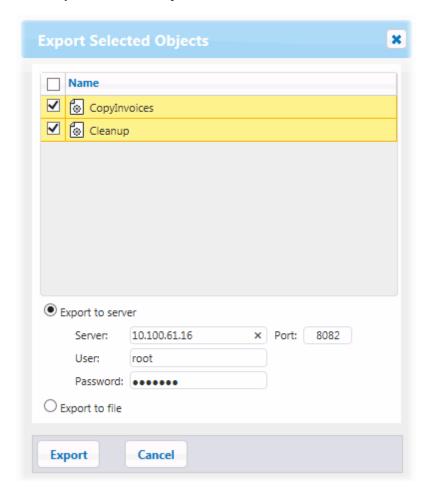
For security reasons, FlowForce Server does not export the following categories of sensitive data:

- Passwords defined locally in jobs
- Passwords available as "standalone" credential records.
- Passwords stored with system functions (such as /system/ftp functions)

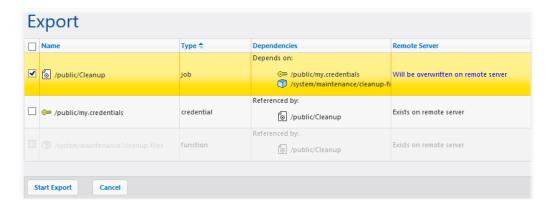
Therefore, although you can export records of type "credentials", FlowForce does not include the password in the exported record. On the other hand, when you import a credential record and you choose to overwrite it at destination, the password existing at destination is preserved.

### To export jobs to a running FlowForce Server instance:

- 1. Click **Configuration**, and select the records you want to export. You can select either specific records within a container, or the entire container.
- 2. Click Export Selected Objects.



- 3. In the dialog box, click **Export to server**, and enter the host name of the destination FlowForce Server, and the port where it runs.
- 4. Enter your user name and password on the destination FlowForce Server instance, and then click **Export**. FlowForce displays all records to be exported on a page where you can view their dependencies, or omit them from the export.



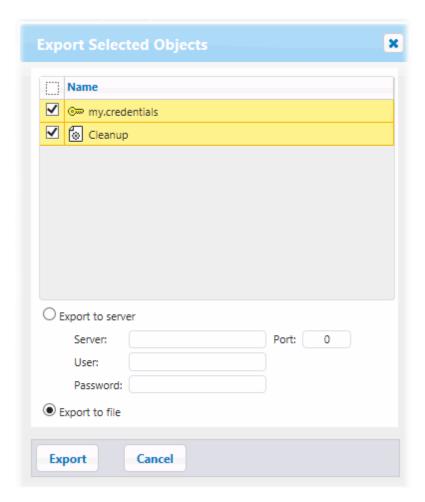
The records with a yellow background are those that are being exported. The record without a yellow background are those that you have excluded from the export, by clearing their adjacent check box. Finally, the records that are grayed out represent dependencies on built-in system functions, so you cannot take actions on them.

The "Remote Server" column indicates if the file exists at destination. If the dependencies already exist at destination, you can safely omit such records from the export. Otherwise, if you are exporting without dependencies and the dependencies do not exist at destination, such jobs will likely fail.

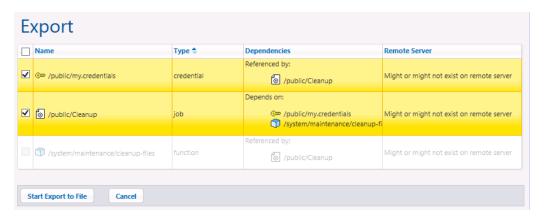
5. Click to select the objects to be exported, and then click **Start Export**.

### To export jobs to a file:

- 1. Click **Configuration**, and select the records you want to export. You can select either specific records within a container, or the entire container.
- 2. Click Export Selected Objects.



3. In the dialog box, click **Export to file**, and then click **Export**. FlowForce displays all records to be exported on a page where you can view their dependencies, or omit them from the export.



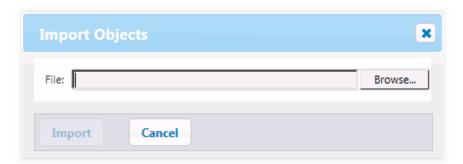
The records with a yellow background are those that are being exported. The record without a yellow background are those that you have excluded from the export, by clearing their adjacent check box. Finally, the records that are grayed out represent dependencies on built-in system functions, so you cannot take actions on them.

Because you are exporting to a file and not to a running FlowForce Server instance, it is not possible to determine whether the exported objects exist at destination. For this reason, the "Remote Server" column shows "Might or might not exist on remote server". If the dependencies will exist at destination when you import the .zip archive back into FlowForce, you can safely omit such records from the export. Otherwise, if you are exporting without dependencies and the dependencies will not exist at destination, such jobs will likely fail.

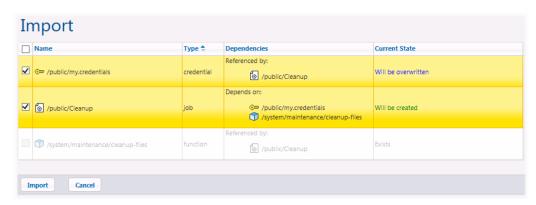
4. Click Start Export to File. Depending on your browser settings, you may either be prompted to save the .zip archive to a local directory, or the browser may save it automatically to a preconfigured destination directory.

#### To import jobs from a file:

1. Click Configuration, and then click Import Objects.



- 2. On the dialog box, click **Browse**, and select a source .zip archive that was previously exported from FlowForce Server.
- 3. Click **Import**. FlowForce Server displays the records that are about to be imported on a separate page, along with their dependencies. The "Current state" column informs about what will happen to each record after you click the **Import** button.



When you are importing credentials, the status "Will be overwritten" does not affect the currently stored password of a credential with the same name. As such, passwords are never exported for security reasons, and thus are not overwritten on import. 4. Click **Import**.

# **Chapter 8**

The FlowForce Expression Language

# **8** The FlowForce Expression Language

A FlowForce expression is a statement computed and executed by FlowForce Server when a job runs. You can think of FlowForce expressions as a basic scripting language understood by FlowForce that helps you "glue together" multiple steps within a job. FlowForce expressions are typically necessary in the following contexts:

- In parameters of built-in functions (that is, you can write or embed expressions in input fields in the job configuration page). Here are a just a few examples:
  - Change the data type of the result returned by the execution step
  - Pick a specific value from a result that returns an array of values
  - o Concatenate multiple values in order to produce a string.
- In "when" steps, to produce conditional statements. This enables you to execute the step if the expression you provide evaluates to Boolean **true**.
- In "for-each" steps. "For-each" steps enable you to loop through a sequence of items, where the sequence is defined by an expression.

This section describes the concepts that will help you build FlowForce expressions for scenarios such as the ones listed above.

# 8.1 Creating a "Hello, World!" Job

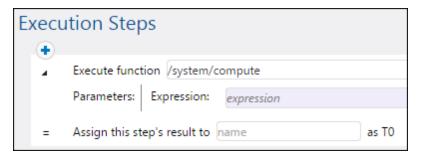
This example shows you how to create a simple job that outputs the text "Hello, World!" in the browser. The text will be created by means of a FlowForce expression. You will be able to trigger the job on demand by clicking a link in the browser (that is, the job will be exposed as a Web service).

### To create the "Hello, World!" job:

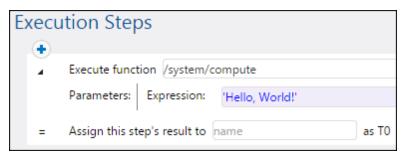
- 1. Log on to FlowForce Server and navigate to a container where you have permission to create new jobs (this example uses the default /public container).
- 1. Click Create, and then select Create job.
- 2. Enter the job title and description.



3. Add a new execution step which calls the built-in function <a>¬/system/compute</a>.



4. In the **Expression** field, enter the text 'Hello, World', enclosed within single quotes. The content of this field represents a FlowForce Server expression.



5. Declare the execution result as **string**.



6. Select the **Make this job available via HTTP...** check box and type "HelloWorld" as service name (see Exposing Jobs as Web Services).



- 7. Under "Credentials", select an existing credential record or specify a local credential (see Credentials).
- 8. Click Save.

You have now finished creating a job that computes the string value "Hello, World!" and returns it as the job result. To test the job, enter <a href="http://127.0.0.1:4646/service/HelloWorld">http://127.0.0.1:4646/service/HelloWorld</a> in the browser's address bar, provided that FlowForce Server runs at the default host and port name. If you have defined other host and port settings in the Configuration page (see <a href="Setting the Network Address">Setting the Network Address</a> and <a href="Port">Port</a>), change the address accordingly. If prompted for credentials when accessing the Web service, supply the same credentials you use to log on to FlowForce Server.

If you use Internet Explorer to test FlowForce Server jobs exposed as Web services, disable the "Show friendly HTTP error messages" option in the **Advanced** tab.

If the job executes successfully, the browser displays the output of the job:

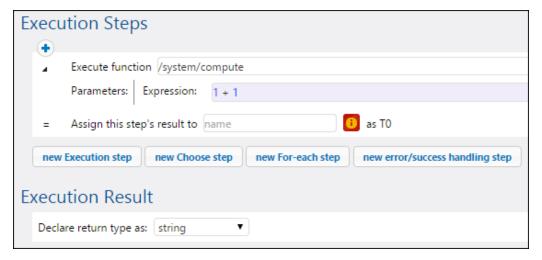
Hello, World!

If the job returns an exit code other than "0", the browser displays a "Server execution failed" message. In this case, check the log of the job in FlowForce Server to identify the error (see Viewing the Job Log).

# 8.2 Computing an Expression

A simple way to test FlowForce expressions before embedding them in jobs is to create an execution step that calls the <a href="mailto:system/compute">system/compute</a> function. For a step-by-step example, see <a href="mailto:Creating a "Hello, World!" Job.</a>

To understand this concept better, create a step that calls the <a href="#">D/system/compute</a> function and enter "1+1" in the expression field. Make sure to declare the job return type as "string", as shown below.



When you attempt to save the job, FlowForce displays a "Types string and number do not match" error. This error happens because the computed expression is a number, whereas the return type of the job is declared as a string value.

To fix the typing problem, either change the return type of the job to "number" or convert the number to a string. The example below calls the FlowForce expression function string which converts a number into a string value.



When you need to compute an expression and return the value as string, you can alternatively use the <a href="Mailto:System/compute-string">System/compute-string</a> function. In this case, note that the expression part must be delimited from the string with curly braces (see <a href="Embedding Expressions in String Fields">Embedding Expressions in String Fields</a>).

# 8.3 The Expression Language Rules

To avoid errors in FlowForce expressions, follow these rules:

- Use only allowed or declared values.
- To use a string literally, enclose it within single quotes.
- To embed an expression in a string field, enclose it within curly braces, that is, the { and } characters.
- The expression must produce a data type which is meaningful in the field where the expression was entered.

Let's now have a look at these rules in more detail.

### Rule #1: Use only allowed or declared values

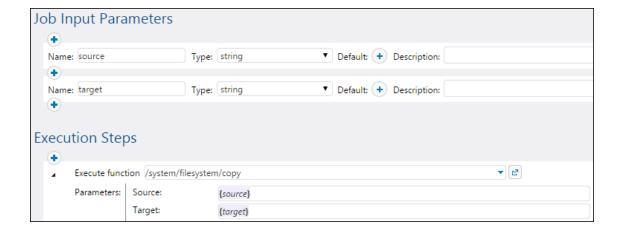
The following constructs are allowed in FlowForce expressions:

- FlowForce expression functions (for complete reference, see Expression Functions)
- FlowForce operators (see Operators )
- Numeric values
- String values
- Previously declared variables

When you type text inside a field which allows FlowForce expressions, a real-time syntax check takes place. If the syntax is not correct, FlowForce highlights in red the offending characters. Below is an example of a syntax validation error:



The error occurs because neither **source** nor **target** have been declared in the job, so FlowForce cannot interpret the expression. The problem can be fixed by declaring these values (for example, as job input parameters):



### Rule #2: Enclose strings in single quotes

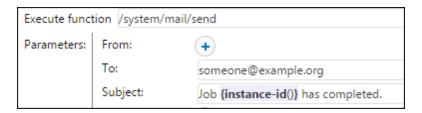
If you need to use a string literally, enclose it within single quotes. Otherwise, the expression might produce undesired results or validation will fail. Consider the following examples:

Expression	Will be evaluated as	Explanation
1+1	2	The data type of the value is numeric.
'1+1'	1+1	The data type of the value is string.
1+1==2	true	The data type of the value is Boolean.

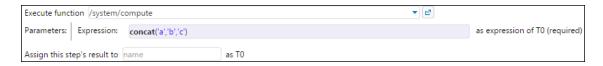
When you need to convert values from one data type to another, use the FlowForce expression functions (see also Rule #4).

### Rule #3: Use curly braces in string fields

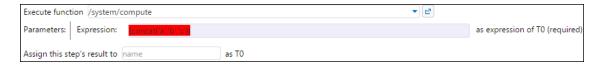
If you want to embed an expression inside a string field, enclose the expression within curly braces. In the example below, curly braces delimit the expression instance-id() (which is a FlowForce expression function) from the rest of the string.



If the entire field is of type "as expression", do not use curly braces. For example, the **Expression** parameter of the <u>system/compute</u> built-in function function has this type. Below is an example of a correct value for this field (notice no curly braces are used):



Typing curly braces inside the expression field would trigger a syntax error:



See also **Embedding Expressions in String Fields**.

## Rule #4: Use the correct data type

Finally, be aware that FlowForce performs data type checks when you save a job. An error will occur if the expression entered in a field does not match the data type expected by the field. You can see the data type expected by each field displayed on the right side of it, for example:



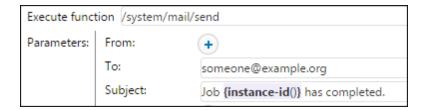
Therefore, an expression such as 1+1 is not a valid in a string field, because it is implicitly evaluated as numeric. On the other hand, the expression '1+1' is valid in a string field. Consider the following examples:

Expression	Will be evaluated as	Explanation
1/4	0.25 (as Number)	The data type of the value is numeric.
		Use this expression in a field or context which expects a numeric value; otherwise, job validation would fail.
1+1==2	true (as Boolean)	The data type of the value is Boolean.
	, ,	Use this expression in a field or context which expects a Boolean value; otherwise, job validation would fail.
'apple'	apple (as String)	The data type of the value is string.
		Use this expression in a field or context which expects a string value; otherwise, job validation would fail.
concat('1','2','3')	123 (as String)	The data type of the value is string.
		Use this expression in a field or context which expects a string value; otherwise, job validation would fail.
1+'apple'	-	This expression is not valid, and FlowForce will return an error when you attempt to save the job. Evaluation cannot take place because two different data types (string and numeric) are being compared.
{content (stdout(result))}	[] (as String)	This expression uses two nested expression functions:
		<ul> <li>The function stdout gets the standard output of a shell command, as stream.</li> <li>The function content converts the stream</li> </ul>

Expression	Will be evaluated as	Explanation
		value to a string.  Although the expression is correct, the job will validate successfully only when the following is true:
		<ul> <li>The value "result" has been previously declared.</li> <li>The value "result" actually contains the standard output of a shell command.</li> <li>The expression is embedded into a string field.</li> </ul>
		See also Calling Expression Functions.

# 8.4 Embedding Expressions in String Fields

To use a FlowForce expression in a string field, enclose the expression within curly braces, that is, the "{" and "}" characters. The expression part of a string field normally has a light purple background, which helps you distinguish the expression part from the rest of the string, for example:



In a string field, only the expression enclosed within curly braces will be treated by FlowForce as an expression. If you want FlowForce to interpret the "{" and "}" characters literally, write double braces instead of a single brace. Consider the following cases:

A string field with the following value	Will be evaluated as	Explanation
echo Hello, World!	echo Hello, World!	The string does not use any curly braces (it does not contain an embedded expression), so it is evaluated as is.
echo { <i>Hello</i> , World	-	The string cannot be evaluated. The embedded expression is not syntactically correct, so FlowForce displays a syntax error.
echo {'Hello, World!'}	echo Hello, World!	The string contains an embedded expression which is syntactically correct. However, the expression is inside a string field, so the evaluation result would be the same if you used no expression at all (see the first example above).
echo {{'Hello, World!'}}	echo {'Hello, World!'}	The string does not contain an expression, since the escape characters {{ and }} were used.

# 8.5 Calling Expression Functions

The FlowForce expression language includes a number of functions that can be used to perform basic operations (primarily, handle values returned by execution steps). You can call these functions from any context where FlowForce expressions are valid (that is, from within an expression).

FlowForce expression functions should not be confused with the FlowForce built-in functions. Built-in functions are called from FlowForce execution steps (that is, they are executed as steps), while expression functions are called from FlowForce expressions.

As a typical scenario to call expression functions, let's consider the job illustrated below, which consists of two execution steps.

The first step executes a shell command (namely, it outputs the text "Hello, World!"). Notice that the data type returned by this step is "as result". The returned value is declared as **var1**.

The second execution step calls the <a href="mailto:system/compute-string">system/compute-string</a> built-in function. We called this function in order to convert **var1** to a string. The expression itself is embedded into a string field (which is indicated by the curly braces), and it calls two nested expression functions.

- The function stdout returns the standard output of a shell command, as stream.
- The function content converts the stream value to a string.



Now that the data type conversion is complete, you can further use the string value **var2** as required by your job processing logic (for example, send it in an email).

For reference to all available expression functions, see Expression Functions.

# 8.6 FlowForce Data Types

FlowForce operates with the following data types.

### string

Represents a string value, for example: 'Hello, World!'.

#### number

Represents a numeric value, for example: -1, 0, 56, 0.45565.

#### **Boolean**

Represents a true or false value.

#### result

This is an abstract type which represents a result produced by an execution step.

An execution step may process various executable files which may be MapForce mappings, StyleVision transformation files, shell functions, and others. The result data type, therefore, stands for whatever represents the output of such files.

If the execution step runs a MapForce mapping, the output could be an XML, XBRL, text, JSON, and any other file types generated by MapForce.

If the execution step runs a StyleVision transformation, the output could be PDF, Word, HTML files, and any other output types generated by StyleVision.

To get access to the resulting value, give it some name (for example, "output"), and pass it to the {results} expression function. This will convert it to a stream, which you can further process with stream expression functions (see also <u>Calling Expression Functions</u>).

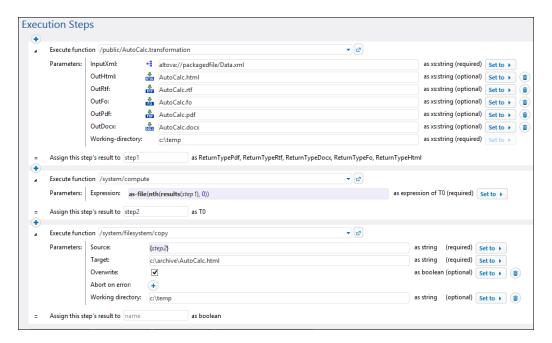
If the execution step runs a shell command, call specific step result expression functions depending on what exactly you need to output. For example, to return the standard output as a stream, use the expression {stdout(output)}. To return the standard error as a stream, use the expression {stderr(output)}. For more information, see <a href="Step Result Functions">Step Result Functions</a>.

#### results

It may be the case that a MapForce mapping or a StyleVision transformation returns multiple objects. The result produced by such steps has results as data type.

To handle such output, use the {results(output)} expression function which returns an array of streams. Then pick a particular stream from the array using the nth function.

For example, the job illustrated below was created from a StyleVision transformation file deployed to FlowForce. This job takes as input parameter an XML file and returns multiple outputs in various formats.



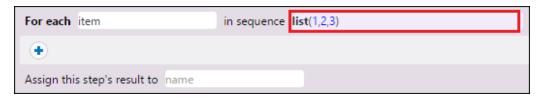
The job has three execution steps.

- 1. The first execution step performs the actual data transformation.
- 2. The second execution step calls the compute function of FlowForce to get one of the results of this transformation. Namely, the expression as-file(nth(results(output), 0)) gets the first item in the array of streams, as a file. The expression uses "0" and not "1" because the array has a zero-based index.
- 3. The third execution step copies the HTML file to the **c:\archive** directory.

#### item

Sometimes, you need to create expressions that assemble or disassemble lists (see <u>List Functions</u>). A list consists of objects of generic type item. An item has an abstract data type. You can determine the data type of item depending by looking at the type of objects that make up the list (which can be strings, numbers, or even streams). Note that a list can contain only items of the same data type.

The image below illustrates a loop where "item" is of numeric type, since the list itself consists of numeric values.



# 8.7 Operators

To build FlowForce expressions, you can use the operators listed below. Remember that you can test any expression by calling the built-in function system/compute.

Operator	Description	Example
==	Checks if a and b are equal (numerically equal for numbers, code-point equal for strings).	2 + 3 == 5 computes to <b>true</b>
		2 + 3 == 4 computes to false
!=	Checks if a and b are not equal. Note that the following three expressions are equivalent:	2 + 2!= 5 computes to <b>true</b>
	<ul> <li>a != b</li> <li>not (a == b)</li> <li>a &lt;&gt; b</li> </ul>	3 + 2!= 5 computes to false
<	Checks if ${\tt a}$ is less than ${\tt b}$ (numerically less for numbers, see below for strings).	4 < 5 computes to true
<=	Checks if a is less than or equal to b.	5 <= 5 computes to true
>	Checks if a is greater than b.	5 > 1 computes to true
>=	Checks if a is greater than or equal to b.	5 >= 5 computes to true
+	Addition.	1 + 1 computes to 2
-	Subtraction.	2 - 1 computes to 1
*	Multiplication.	3 * 2 computes to <b>6</b>
/	Division.	6/3 computes to 2

String comparisons are performed as follows:

The common prefix of the two strings are ignored (evaluated on code points)

- If both remaining strings are non-empty, their first code points are compared numerically
- Empty strings are less than non-empty strings

Use parentheses to instruct FlowForce to evaluate the expression inside first. For example:

2 + 3 \* 4 computes to **14**.

(2 + 3) \* 4 computes to **20**.

# 8.8 Example: Check if a Path Exists

This example shows you how to create a job which informs you if a path (to a file or directory) exists on the operating system. To achieve this goal, you will use a combination of built-in functions and expression functions. The job will be defined as a Web service, so that you can trigger it on demand, by accessing a URL from the browser. The job will take the path as an argument, and will return a string which informs whether the path supplied as argument exists on the operating system where FlowForce Server runs.

### To create the job:

- 1. Log on to FlowForce Server and navigate to a container where you have permission to create new jobs (in this example, we are using /public).
- 2. Click Create, and then select Create job.
- 3. Add a job name (CheckPath, in this example) and, optionally, a job description.



4. Under Job Input Parameters, click , and add the parameter **path**, as shown below.



5. Add a new execution step which calls the <a href="mailto:system/shell/commandline">system/shell/commandline</a> function, and enter the shell command which checks for the existence of the file. Make sure to declare the result of this step, as shown below (in this example, we called it **output**).

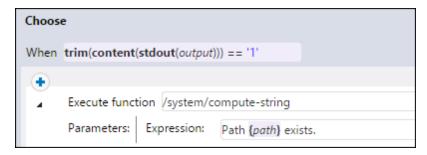


On Windows, the shell command outputs "1" when the path exists and "0" when it does not exist. If FlowForce Server runs on a Unix system, adjust the command accordingly. Notice that the command embeds the FlowForce expression {path}. This expression references the input parameter defined in the previous step.

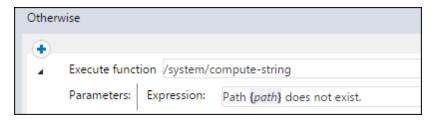
6. Under "Execution Steps", click the 📩 button, and then select **new Choose step**. Then

enter trim(content(stdout(output))) == '1' as condition expression. This expression consists of three nested functions: stdout, content, and trim. First, the stdout function gets the standard output of the result returned by the previous step. Then the content function converts the standard output to string. Finally, the trim function removes any leading or trailing spaces, carriage returns, or line feeds from the standard output. The result is then compared to "1" using the equality operator. If both values are equal, the path exists. Otherwise, the path does not exist.

7. Under the **When** clause, add an execution step as shown below. This execution step calls the <a href="mailto:system/compute-string">system/compute-string</a> function to build the string value that should be returned when the path exists. Notice that the value embeds the FlowForce expression {path}. This expression references the input parameter defined in a previous step.



8. Under the **Otherwise** clause, add an execution step as shown below. This execution step calls the <u>O/system/compute-string</u> function to build the string value that should be returned when the path does not exist. Notice that the value embeds the FlowForce expression {path}. This expression references the input parameter defined in a previous step.



9. Under Execution Result, declare the return type as **string**.

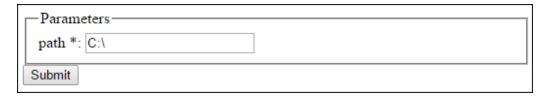


- 10. Under Service, click to select the **Make this job available via HTTP** check box, and enter **CheckPath** as name of the service (see <u>Exposing Jobs as Web Services</u>).
- 11. Under "Credentials", select an existing credential record or specify a local credential (see <u>Credentials</u> ).
- 12. Click Save.

To test the job, enter <a href="http://127.0.0.1:4646/service/CheckPath">http://127.0.0.1:4646/service/CheckPath</a> in the browser's address bar, provided that FlowForce Server runs at the default host and port name. If you have defined other host and port settings in the Configuration page (see <a href="Setting the Network Address and Port">Setting the Network Address and Port</a> ), change the address accordingly. If prompted for credentials when accessing the Web service, supply the same credentials you use to log on to FlowForce Server.

If you use Internet Explorer to test FlowForce Server jobs exposed as Web services, disable the "Show friendly HTTP error messages" option in the **Advanced** tab.

Since this job has arguments, you will be prompted to supply them when you access the Web service in the browser.



If the job executes successfully, the browser displays the output of the job, for example:

Path C:\ exists.

If the job returns an exit code other than "0", the browser displays a "Server execution failed" message. In this case, check the log of the job in FlowForce Server to identify the error (see Viewing the Job Log).

# **Chapter 9**

**MapForce/StyleVision Server Integration** 

# 9 MapForce/StyleVision Server Integration

Mapping files created with Altova MapForce and transformation files created with Altova StyleVision can be automated with the help of the following server counterpart products: MapForce Server (or MapForce Server Accelerator Edition) and StyleVision Server. MapForce Server and StyleVision Server make it possible to execute mappings and transformations in a server environment (Windows, OS X, Linux), either at the command line, or from an API call. If these products do not run alongside FlowForce, automation entails developing programs or writing scripts which call the API or invoke the command line of MapForce Server or StyleVision Server.

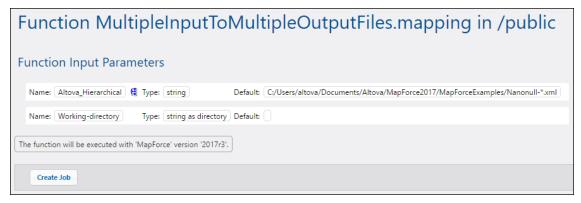
When MapForce Server and StyleVision Server run under FlowForce Server management, automation can be taken to the next level. Namely, the mappings and transformations can be deployed directly to FlowForce Server and be run as jobs (see also <a href="How It Works">How It Works</a>). This way, the mapping or transformation will benefit from all the advantages of a FlowForce Server job: scheduled or on demand execution, execution as a Web service, configuration by means of FlowForce expressions, error handling, conditional processing, email notifications, and so on. The instructions to deploy a mapping/transformation to FlowForce Server are similar in MapForce and StyleVision, and are as follows:

On the File menu of MapForce or StyleVision, click Deploy to FlowForce (Server).

For the list of prerequisites, tips, as well as reference to the deployment settings, refer to the documentation of MapForce and StyleVision, specifically:

- Deploying Mappings to FlowForce Server
- Deploying Transformations to FlowForce Server

Once deployed, the mapping or transformation appears in the FlowForce container to which you deployed it. In FlowForce, such mappings or transformations become functions, and thus can be turned into jobs. They can also be called from other jobs, and accept various inputs (typically, files) as parameters. Note that FlowForce Server does not execute such mapping or transformation functions by itself; MapForce Server or StyleVision Server (or both, depending on the case) are invoked to perform the actual execution.



FlowForce Server function automatically generated from a deployed mapping

#### To create a job from a deployed MapForce mapping or StyleVision transformation:

1. Navigate to the FlowForce container where you deployed the MapForce mapping or

- StyleVision transformation (for example, the container "/public").
- 2. Click the required mapping or transformation. By default, MapForce mappings have a naming convention *SomeFile.mapping* and StyleVision transformations have a naming convention like *SomeFile.transformation*.
- 3. Click Create Job.

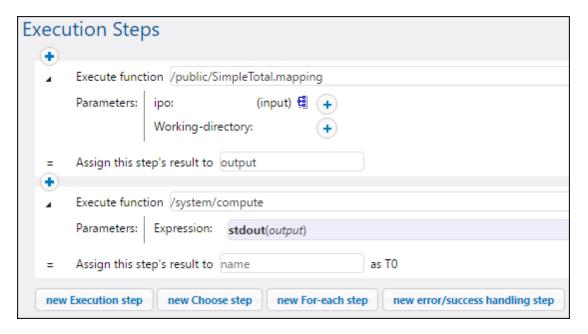
You can now configure the job according to your needs: for example, the settings in the "Trigger" section defines how and when the job should run (see <u>Managing Triggers</u>). The input parameters corresponds to the inputs defined by the original MapForce mapping (or StyleVision transformation).

The output of the <u>step</u> which executes the mapping or transformation function represents a generic FlowForce type (let's call it **result**), not an actual file or a stream. To convert **result** to actual files or streams, FlowForce expression functions must be used, see <u>Accessing the</u> Result of a Mapping or Transformation Job.

For a step-by-step example which illustrates deploying a StyleVision transformation and creating a job from it, see <u>Creating a Job from a StyleVision Transformation</u>. For a similar example for MapForce, see <u>Creating a Job from a MapForce Mapping</u>. For an example job which requires calling both MapForce Server and StyleVision Server, see <u>Example: Generating Multiple PDF</u> Files from Multiple XML Files.

# 9.1 Accessing the Mapping/Transformation Result

After a MapForce mapping or StyleVision transformation has been deployed to FlowForce Server, it becomes a FlowForce function which can be called from other execution steps. For example, in the first step of the job below, a mapping function called **SimpleTotal.mapping** is being executed.



Notice that the job consists of two steps:

- 1. Step 1 calls MapForce Server to actually run the **SimpleTotal.mapping** function. Importantly, the **Assign this step's result to** field gives a name to the mapping result (in this case, it is **output**; however, it can be any name you choose).
- 2. Step 2 calls the <a href="//system/compute">/system/compute</a> function which converts the output of the mapping to a stream.

By default, the output of a mapping or transformation function is of generic type **result**. In order for the output to become useful, **result** must be converted to whatever data type you require (for example, string, stream, file). For this purpose, the <code>/system/compute</code> built-in function is available, as well as various FlowForce expression functions. In the example above, the built-in function <code>/system/compute</code> was called to perform the required data type conversion. Namely, the <code>expression stdout(output)</code> converts the result of the previous step to a stream.

The table below lists examples of FlowForce expressions that you will likely need to process the result of a mapping or a transformation function. Remember that, in all these examples, output is the name you entered in the **Assign this step's result to** to field.

FlowForce Expression	Purpose
stdout(output)	Converts output to a stream.
as-file(stdout(output))	Converts output to a file.

FlowForce Expression	Purpose
<pre>as-file(nth(results(output), 0))</pre>	This kind of expression is required if output consists of multiple files. This happens when the mapping or transformation function was designed (in MapForce or StyleVision) to generate not just a single output, but multiple outputs. The expression converts output to a sequence of streams, picks up the first stream from the sequence, and converts it to file. For an example, see Creating a Job from a StyleVision Transformation.
<pre>as-file(nth(results(output,     "CompletePO"), 0))</pre>	Same as above, except that the file is retrieved from the sequence of streams not by its zero-based index as above, but by name (in this case, "CompletePO").

For complete reference to FlowForce expression functions that are available to handle the result of a step or job, see  $\underline{\text{Step Result Functions}}$ . For an introduction to FlowForce expressions, see  $\underline{\text{The}}$  FlowForce Expression Language.

# 9.2 Example: Generate Multiple PDF's from Multiple XML's

This example illustrates how to create a FlowForce Server job which takes as input multiple XML files and returns as output multiple PDF files. The FlowForce Server job will invoke both MapForce Server (to generate the XML output from multiple source XML files) and StyleVision Server (to convert the XML output to PDF).

This example requires a basic understanding of how MapForce mappings and StyleVision transformations work. If you are completely new to StyleVision and MapForce, it is recommended to read first the "Tutorials" chapters of MapForce and StyleVision documentation, respectively:

- Quick Start Tutorial (MapForce)
- Quick Start Tutorial (StyleVision)

### **Prerequisites**

- Required licenses: MapForce Enterprise or Professional edition, StyleVision Enterprise or Professional edition, MapForce Server or MapForce Server Accelerator Edition, StyleVision Server, FlowForce Server
- FlowForce Server is running at the configured network address and port (see <u>Defining the</u> <u>Network Settings</u>)
- Your FlowForce Server user account has permissions to one of the <u>containers</u> (by default, the **/public** container used in this example is accessible to any authenticated user).
- The mapping job created in this example generates multiple PDF files. Therefore, on the operating system where FlowForce Server runs, you must have rights to create files in some directory (this example uses the directory **C:\FlowForceWorkingDir**).

#### **Demo files used**

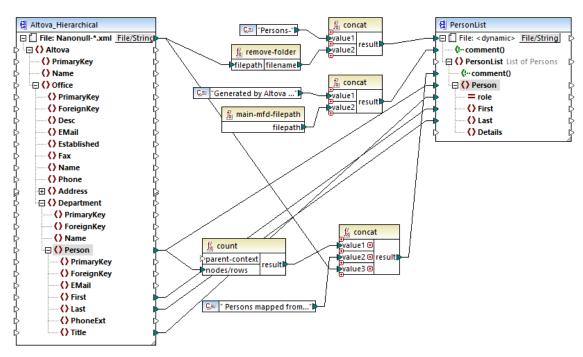
This example makes use of the following sample files, available at the following path: <Documents>\Altova\MapForce2017\MapForceExamples.

- MultipleInputToMultipleOutputFiles.mfd (the MapForce mapping file)
- **PersonListWithGrouping.sps** (the StyleVision transformation file)
- Nanonull-Branch.xml, Nanonull-HQ.xml (the source XML files)

### What the MapForce mapping does

As illustrated below, the mapping consists of a source component (**Altova\_Hierarchical**), a target one (**PersonList**), and various intermediary MapForce built-in functions used to build miscellaneous strings to be written to the output.

The mapping takes as input any XML file that begins with "Nanonull-", from the directory <Documents>\Altova\MapForce2017\MapForceExamples. This is defined in the source MapForce component settings (in MapForce, right-click the header of the Altova\_Hierarchical component illustrated below, and select Properties from the context menu). Notice that "Input File" is set to Nanonull-\*.xml, where the asterisk is a wildcard. Literally, the input is any file which begins with "Nanonull-" and has the .xml extension.



MultipleInputToMultipleOutputFiles.mfd

The target component, **PersonList**, is configured to generate file names dynamically based on the filename of the source XML file. This is defined by right-clicking the **File/String** button at the top of the component, and then selecting **Use Dynamic File Names Supplied by Mapping** menu option. The connection to the "File <dynamic>" node means that a new file will be created for every value in the source. The remove-folder function is meant to get only the filename (without the folder) from the source path. This is then passed as value to the top concat function, which builds a string like *Persons-*<*Source filename>*.

The second concat function builds a string like *Generated by Altova...* followed by the complete path to the mapping file. The result is written as a comment in the target XML file.

The third concat function uses the output of the count function to build a string that indicates how many person records have been mapped from the source. Again, the result is written as a comment in the target XML file.

Finally, the connection to the target **Person** node copies people data from the source to the target. An individual connection exists for each child element of **Person** that must be mapped.

In addition to this, the target component is configured to convert the generated output to PDF, for each XML generated file. Right-click the header of the target component, select **Properties**, and notice that the **StyleVision Power StyleSheet file** text box specifies a relative path to a StyleVision .sps stylesheet. The latter performs the actual conversion of XML to PDF (further discussed below).

To preview the output of this mapping directly in MapForce, click the **Output** tab available under the mapping pane. To preview the PDF result of the StyleVision transformation, click the **PDF** tab. You will notice that multiple XML's (or PDFs, respectively) are generated in the Output pane, for

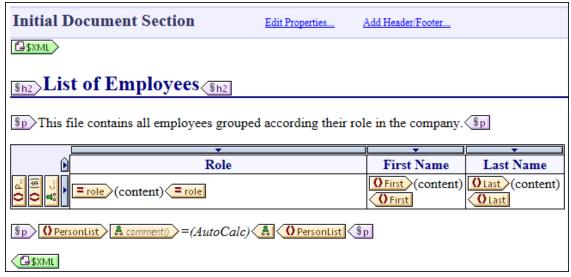
example:

```
C:\Users\
            Preview 1 of 2
                                                        \Documents\Altova\MapForce2017\MapForceExamples\Persons-Nanonull-Branch.xml
          <?xml version="1.0" encoding="UTF-8"?>
          !--Generated by Altova MapForce (http://www.altova.com/mapforce) using C:\Users\
                                                                                                              \Documents\Altova\MapForce201
 3
       <PersonList xsi:noNamespaceSchemaLocation="PersonList.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
                                                                       \Documents\Altova\MapForce2017\MapForceExamples\Nanonull-Branc
            <!--6 Persons mapped from input file C:\Users\"
            <Person role="Office Manager"> .... </Person> <Person role="Accounts Receivable"> .... </Person>
5
9
13
17
21
25
29
            <Person role="PR &amp; Marketing Manager US"> .... </Person>
            <Person role="IT Manager"> .... </Person>
<Person role="Support Engineer"> .... </Person>
            <Person role="Support Engineer"> .... </Person>
          </PersonList>
30
```

At this stage, it is recommended to save one of the two output XML files to the disk (since, by default, MapForce generates temporary files). The file will act as a sample (working XML) if you would like to open and test the StyleVision power stylesheet in StyleVision (see next section). To save an output file, first click the **Output** tab, and then, on the **Output** menu, click **Save Output File**.

### What the StyleVision transformation does

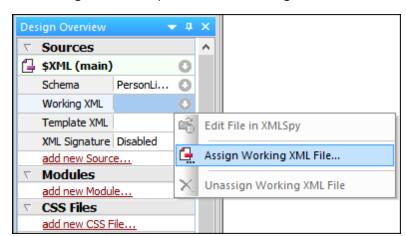
Run StyleVision and open the **PersonListWithGrouping.sps** transformation file. Recall that this file is in the same directory as the MapForce mapping discussed above, and it is referenced by the target MapForce component.



PersonListWithGrouping.sps

The StyleVision .sps stylesheet illustrated above uses a single XML as source and creates a PDF document from it. The PDF document consists of a heading ("h2"), an introductory paragraph, a table populated dynamically, and an ending paragraph. The heading and the introductory paragraph contain static text, while the table and the ending paragraph are populated from the nodes of the source XML file, as indicated by the wrapping tags.

To preview this transformation directly in StyleVision, follow the steps below:



1. In the **Design Overview** pane, next to **Working XML**, click .

2. Select **Assign Working XML File** and browse for the XML output file saved previously from MapForce (see previous section).

Importantly, the .sps stylesheet is agnostic with the respect to the actual name or origin of the source XML file; it simply processes the XML file provided as input (as long as it conforms to the specified XML schema), and creates a PDF out of it. In order to automate this stylesheet so that it generates multiple PDF files, it will need to be deployed to FlowForce Server, as shown further below.

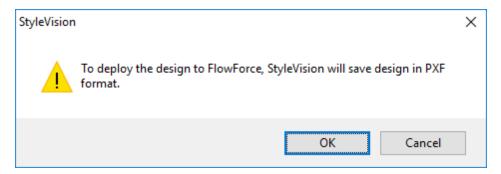
#### **Deploy the files to FlowForce Server**

So far, you have become familiar with the purpose of both the MapForce mapping and the StyleVision transformation used in this example. For more information about designing MapForce mappings and StyleVision stylesheets, refer to the documentation of these products (<a href="https://www.altova.com/documentation.html">https://www.altova.com/documentation.html</a>).

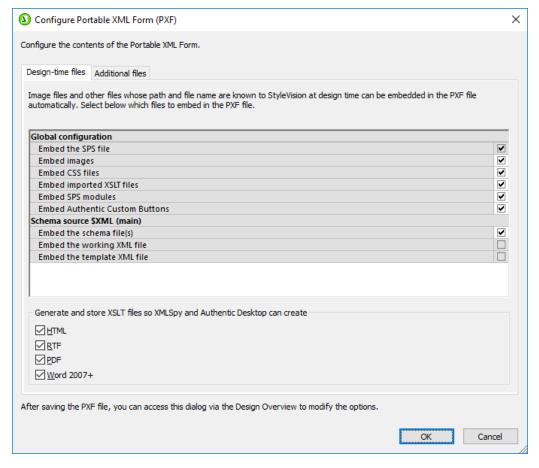
To make automation possible, both files must now be deployed to FlowForce Server. As specified in the "Prerequisites" section above, FlowForce Server must be licensed and running, and both MapForce Server and StyleVision Server must be licensed and running under FlowForce Server management. On Windows, you can use the <a href="mailto:verifylicense">verifylicense</a> command of each server product to check the status of its license. On other operating systems, the job execution will fail with an appropriate log message if the license is not found or valid.

To deploy the StyleVision stylesheet to FlowForce Server:

- 1. On the **File** menu, click **Deploy to FlowForce**.
- 2. When prompted that the design file will be saved as PXF (Portable XML Form) format, click OK.

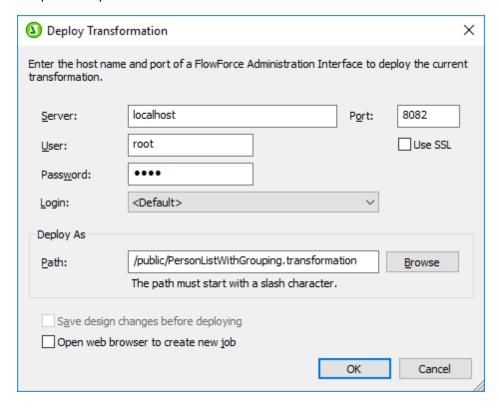


When prompted to select the desired files to be included in the deployed package, leave
the default settings as is. Although only PDF is generated in this example, including
other outputs will save you time later if you change your mind and want to generate
additional formats like HTML and RTF.



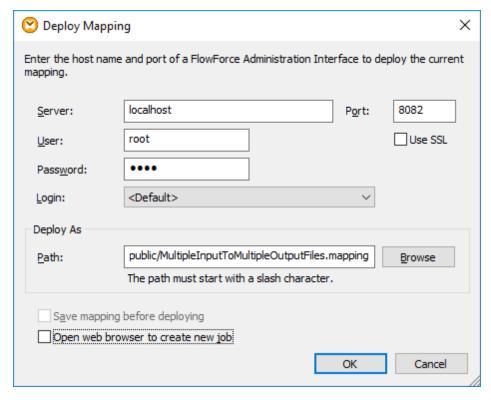
4. When prompted, fill in the connection details to FlowForce Web Server. For simplicity, in the image below, the transformation is deployed to the local machine on port 8082, through plain HTTP. It is also possible to specify a remote address and deploy the files through an SSL-encrypted connection, provided that FlowForce Web Server has been configured to accept such connections (see <u>Defining the Network Settings</u>). The user and password values are illustrated for the root FlowForce account in this case; however, any other FlowForce user account may also be used, provided that it has permissions to write data to the specified path. In this example, the **Open browser to create new job** check

box has been deliberately left unchecked, because creating and configuring the job will be a separate step discussed further below.

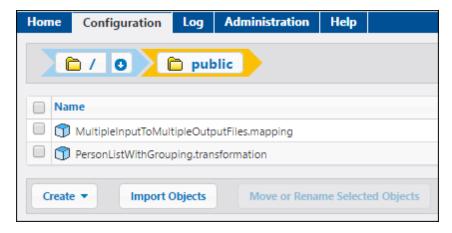


To deploy the MapForce mapping to FlowForce Server:

• On the **File** menu, click **Deploy to FlowForce Server**. Filling in the connection details illustrated below works the same way as discussed above for StyleVision.



After the files were successfully deployed, the corresponding entries will appear in the specified FlowForce container (in this case, "/public") when you log on to FlowForce Server:

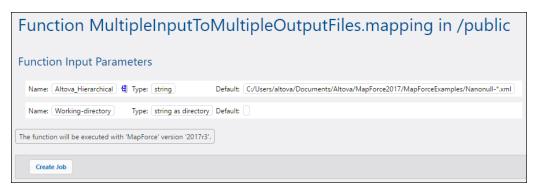


Note that the entries above are not jobs yet; they are now FlowForce functions from which actual jobs have yet to be created, as shown below.

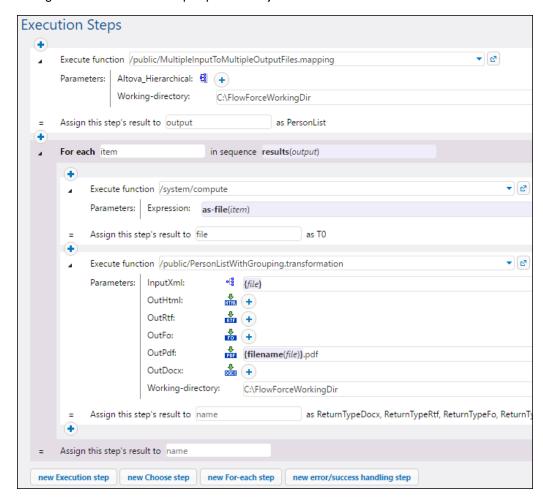
### Create and configure the FlowForce job

Now that the MapForce mapping and the StyleVision transformation have been deployed to FlowForce Server, they can be used to create the required job, as follows:

 Navigate to the FlowForce /public container, and click the function MultipleInputToMultipleOutputFiles.mapping deployed previously. Notice that the source component of the MapForce mapping discussed at the very beginning of this example has now become an input parameter to the FlowForce function. Also, it has a default value which is the path to the instance XML files processed by the mapping. This value can be overridden later if necessary. The "Working-directory" parameter was added automatically by FlowForce; its role will be clarified in the next steps.



- 2. Click Create Job.
- 3. Enter a name and optionally a description for the job you are creating.
- 4. Configure the "Execution Steps" part of the job as shown below.



5. Under "Service", select the "Make this job available via HTTP..." check box and enter the name of the Web service that will trigger the job on demand. If you prefer to run the job as

a scheduled job, or as a file system trigger, set the appropriate triggers (see Managing Triggers).

Service	
■ Make this job available via HTTP at URL http://< <i>FlowForce server</i> >/service/	GenerateMultiplePDFs

 Under "Credential", enter the username and password of the operating system user account (the job will be executed as this user). Be careful not to confuse this password with the password of the FlowForce Web administration interface (see also <u>How</u> <u>Credentials Work</u>).

C	redential			
	Run job using credential:	Select existing credential:		
		Define local credential:	User name:	altova
			Password:	Change password

7. Click Save.

To understand how the job actually works, let's have a closer look at the "Execution Steps" section of the job. The first execution step calls the mapping deployed previously. The second parameter defines the working directory. A working directory is a parameter required by <a href="execution steps">execution steps</a> if the job needs a location to unpack any input files or save output files. FlowForce Server also uses the working directory to resolve any relative paths that occur during step execution. When asked to provide a working directory, you should supply a valid path on the operating system where FlowForce Server runs. If you do not supply a working directory when creating the step, FlowForce Server uses a temporary directory. In this example, the working directory is "C: \FlowForceWorkingDir".

The output returned by the first execution step represents the data returned by the mapping. It has been explicitly named output, in order to make it possible to refer to it in a subsequent step.

The second step of the job is a "for-each" step. Notice how the "for-each" step uses a FlowForce expression <code>results(output)</code> to get access to the data returned by the first step (that is, the output returned by the mapping). Specifically, the expression calls the function <code>results()</code> which takes as argument the output returned by the previous step, see also <a href="Step Result Functions">Step Result Functions</a>. For an introduction to FlowForce expressions, see <a href="The FlowForce Expression Language">The FlowForce Expression Language</a>.

The "for-each" step consists of two smaller execution steps:

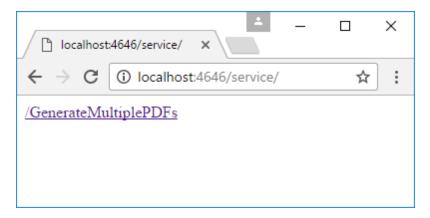
- 1. The first step calls the <u>/system/compute</u> built-in function to convert the mapping output into an actual file (generically named <u>file</u>). Importantly, the output of the <u>results(output)</u> expression is a stream, not a file. If the mapping returns multiple outputs (as in this case), the mapping output represents a sequence of streams. For this reason, a FlowForce expression function (as-file) is used to convert the current stream (the one that is being iterated) into an actual file.
- 2. The second step calls, for each stream that is being iterated, the StyleVision transformation deployed previously. Namely, with each iteration, StyleVision Server is called, an XML file is supplied as input, and a PDF file is returned as output. The

FlowForce expression {filename(file)}.pdf creates the actual PDF file name on the disk. This operation takes place in the working directory specified by the "Working-directory" parameter.

**Note:** In this example, the same working directory is used by both the step which executes the mapping and the step which runs the StyleVision transformation. In some cases, it may be necessary to specify separate working directories, in order to avoid file name collisions, and potential job execution errors.

#### Running the job

This job was created as a Web service, so you can run it on demand by clicking a link from the browser. To do this, enter the following URL in the browser address bar: http://localhost:4646/service/. Adapt the URL accordingly if you configured a different host and port for FlowForce Server, see Defining the Network Settings.



Click the Web service link to execute the job (in this case, "GenerateMultiplePDFs"). When the job finishes, the PDF files generated by the job will be available in the working directory **C**: **\FlowForceWorkingDir**. Note that, for simplicity, this job did not use error handling. The browser will display "Cannot output the job" even in case of successful execution (this is expected, since the job produces PDF files which cannot be output to the browser). If the job fails to execute for any reason, the browser will display a "Service execution failed" message. In this case, check the error log of the job to see which step has failed and why, see Viewing the Job Log.

# Chapter 10

**Command Line Interface** 

# 10 Command Line Interface

FlowForce provides a command line interface which can be used for administration purposes (such as licensing, troubleshooting, and internal database backup). The command line interface includes the following commands:

help	Displays help for the command supplied as argument.
<u>assignlicense</u>	This command is applicable to Windows platforms only. It can be used to upload and assign a license file to FlowForce Server.
compactdb	Reduces the size of FlowForce .db files if they contain deleted records.
createdb	Creates a new FlowForce database.
debug	Starts the application in debug mode.
exportresourcestrings	Exports all application resource strings to an XML file
foreground	Starts the application in foreground mode.
<u>initdb</u>	Creates or updates the FlowForce database.
<u>install</u>	Installs the application as a Windows service.
licenseserver	Registers FlowForce Server with the Altova LicenseServer on the local network.
migratedb	Migrates FlowForce Server data from a previous version to the latest version.
repair	Starts the application in repair mode.
resetpassword	Resets the password of the <b>aroot</b> user to the default value, and grants to the <b>aroot</b> user all privileges.
setdeflang sdl	Sets the default language.
start	Starts the application as a service.
<u>uninstall</u>	Uninstalls the application as a Windows service.
<u>upgradedb</u>	Upgrades the FlowForce Server database to the latest version.
verifylicense	This command is applicable to Windows platforms only. It can be used to verify whether FlowForce Server is licensed, and, optionally, whether a given license key is already assigned to FlowForce Server.

Command Line Interface 251

#### **Location of the FlowForce Server executable**

Linux	On Linux systems, the FlowForce Server executable (flowforceserver) is located by default at:  /opt/Altova/FlowForceServer2017/bin/ flowforceserver
	where the first forward slash indicates the root directory.
Mac OS X	On Mac OS X systems, the FlowForce Server executable (flowforceserver) is located by default at:
	/usr/local/Altova/FlowForceServer2017/bin
Windows	On Windows systems, the FlowForce Server executable (FlowForceServer.exe) is located by default at:
	<pre><programfilesfolder>\Altova\FlowForceServer\bin \FlowForceServer.exe</programfilesfolder></pre>

#### **General syntax**

The general command line syntax for FlowForceServer is as follows:

```
Windows
```

#### where

```
--h | --help Displays the help text.--version Displays the version of FlowForce Server.
```

The options and arguments in square brackets are optional.

**Note:** On Unix systems, when you are inside the directory of the FlowForce Server executable, you might need to prepend the current directory ("./") to the name of the executable, for example:

```
CentOS 7
./flowforceserver --version

Mac OS X
./FlowForceServer --version
```

252 Command Line Interface help

## 10.1 help

#### **Purpose**

Provides help information about the command supplied as argument.

#### **Syntax**

FlowForceServer help Command

**Note:** On Linux systems, use an all-lowercase flowforceserver to call the executable.

#### **Arguments**

The help command takes a single argument: the name of the command for which help is required. It displays the correct syntax of the command and other information relevant to the correct execution of the command.

#### **Example**

FlowForceServer help createdb

#### Using --help as option for other commands

Help information about a command is also available by using the --help option with that command. For example, using the --help option with the createdb command, as follows:

FlowForceServer createdb --help

has the same result as:

FlowForceServer help createdb

Command Line Interface assignlicense 253

# 10.2 assignlicense

#### **Purpose**

This command is applicable to Windows platforms only. It can be used to upload and assign a license file to FlowForce Server.

#### **Syntax**

FlowForceServer assignlicense [options] FILE

#### **Arguments**

FILE	Specifies the path of the license file to be uploaded.
------	--

t,test-only=true false	When set to true, the license is uploaded and validated.
	When set to false, the license is uploaded, validated, and assigned as well.
	If this option is not specified, the default value is true.

254 Command Line Interface compactdb

## 10.3 compactdb

#### **Purpose**

Reduces the size of FlowForce .db files if they contain deleted records. This command is particularly useful after running the archive-log or truncate-log system maintenance functions (see <a href="https://system/maintenance">/system/maintenance</a>).

#### **Syntax**

FlowForceServer compactdb [options]

**Note:** On Linux systems, use an all-lowercase flowforceserver to call the executable.

datadir=VALUE	VALUE is the path of the data directory which contains the .db files to be compacted. If this option is not specified, the /data directory will be used by default
	(see also FlowForce Server Application Data ).

Command Line Interface createdb 255

#### 10.4 createdb

#### **Purpose**

Creates a new database. If the database already exists then the command will fail. The default database is created at installation time, so it is usually not necessary to use this command.

#### **Syntax**

FlowForceServer createdb [options]

**Note:** On Linux systems, use an all-lowercase flowforceserver to call the executable.

datadir=VALUE	VALUE is the path of the data directory.	

256 Command Line Interface debug

# 10.5 debug

#### **Purpose**

Not for general use. This command starts FlowForce Server in debug mode (that is, not as a service). To stop this mode, press **CTRL+C**.

#### **Syntax**

FlowForceServer debug [options]

**Note:** On Linux systems, use an all-lowercase flowforceserver to call the executable.

		Ĺ
datadir=VALUE	VALUE is the path of the data directory.	l

Command Line Interface exportresourcestrings 257

# 10.6 exportresourcestrings

#### **Purpose**

Outputs an XML file containing the resource strings of FlowForce Server. It takes two arguments: (i) the language of the resource strings in the output XML file, and (ii) the path and name of the output XML file. Valid export languages (with their language codes in parentheses) are: English (en), German, (de), Spanish (es), and Japanese (ja).

#### **Syntax**

FlowForceServer exportresourcestrings Language XMLOutput

**Note:** On Linux systems, use an all-lowercase flowforceserver to call the executable.

#### **Arguments**

Language	Specifies the language of resource strings in the exported XML file Allowed languages are: en, de, es, ja	
XMLOutput	Specifies the location and name of the exported XML file.	

#### **Example**

This command creates a file called strings.xml at  $c:\$  that contains all the resource strings of the FlowForce Server application in English.

FlowForceServer exportresourcestrings en c:\Strings.xml

258 Command Line Interface foreground

# 10.7 foreground

## Purpose

Not for general use. This command starts Altova FlowForce Server in the foreground. It is used internally by the startup scripts for Linux.

Command Line Interface initdb 259

## 10.8 initdb

#### **Purpose**

Creates a new database, or updates an existing one to the latest version. The default database is created at installation time, so it is usually not necessary to use this command.

#### **Syntax**

FlowForceServer initdb [options]

**Note:** On Linux systems, use an all-lowercase flowforceserver to call the executable.

		ı
datadir=VALUE	VALUE is the path of the database directory.	

260 Command Line Interface install

## 10.9 install

#### **Purpose**

This command is executed by the FlowForce Server installer automatically and it is not available for general use. The command installs Altova FlowForce Server as a service, on Windows. This command does not apply to Linux and Mac OS X.

Command Line Interface licenseserver 261

#### 10.10 licenseserver

#### **Purpose**

Registers FlowForceServer with LicenseServer. You must have Administrator privileges (root) to register FlowForce Server with LicenseServer. For more information, see the <u>LicenseServer</u> documentation.

#### **Syntax**

```
FlowForceServer licenseserver [options] Server-Or-IP-Address
```

**Note:** On Linux systems, use an all-lowercase flowforceserver to call the executable.

#### **Example**

```
FlowForceServer licenseserver DOC.altova.com
```

The command above specifies that the machine named DOC.altova.com is the machine running Altova LicenseServer. If LicenseServer is running on the user's machine, the following commands would also be valid:

```
FlowForceServer licenseserver localhost
FlowForceServer licenseserver 127.0.0.1
```

#### **Options**

The options are listed below, in their short forms (first column) and long forms (second column), together with their descriptions. On the command line, one or two dashes can be used for both short and long forms.

j	json	Prints the result of the registration attempt as a machine-parseable JSON object.
		Form:json=true/false

262 Command Line Interface migratedb

# 10.11 migratedb

#### **Purpose**

Migrates FlowForce Server data from a previous version to the latest version.

#### **Syntax**

FlowForceServer migratedb [options]

**Note:** On Linux systems, use an all-lowercase flowforceserver to call the executable.

#### **Options**

datadir=VALUE	VALUE is the path of the database directory.
olddatadir=VALUE	VALUE is the path of the database directory

#### **Example**

 $\label{lowForceServer2017} $$ \end{align*} $$ \clin_{\end{align*} \clin_{\end{align*} \end{align*} $$ \clin_{\end{align*} \clin_{\end{align*$ 

 $--datadir=C:\ProgramData\Altova\FlowForceServer2017\data --olddatadir=C:\ProgramData\Altova\FlowForceServer2015\data$ 

Command Line Interface repair 263

# 10.12 repair

#### **Purpose**

Starts FlowForce Server with all triggers and job execution processes disabled, to enable troubleshooting.

#### **Syntax**

FlowForceServer repair [options]

**Note:** On Linux systems, use an all-lowercase flowforceserver to call the executable.

#### **Options**

datadir=VALUE	VALUE is the path of the database directory.
---------------	--

#### **Example**

 $\label{lem:programDataAltovaFlowForceServer2017} $$ \arrowvertext{ repair --datadir=C:\Pr{programDataAltovaFlowForceServer2017 \arrowvertext{ flowForceServer2017 \arrowvertext{ data}} $$$ 

264 Command Line Interface resetpassword

# 10.13 resetpassword

#### **Purpose**

Resets the password of the **aroot** user to the default value, and grants to the **aroot** user all privileges. It is recommended to stop the running instance of FlowForce Server before performing this operation (see instructions for starting or stopping services on <u>Linux</u>, <u>Mac OS X</u>, and <u>Windows</u>).

#### **Syntax**

FlowForceServer resetpassword [options]

**Note:** On Linux systems, use an all-lowercase flowforceserver to call the executable.

#### **Options**

datadir=VALUE	VALUE is the path of the database directory.	
	' '	

#### **Example**

FlowForceServer resetpassword --datadir=C:\ProgramData\Altova\FlowForceServer2017\data

Command Line Interface setdeflang (sdl) 265

## 10.14 setdeflang (sdl)

#### **Purpose**

The setdeflang command (short form is sdl) sets the default language of FlowForce Server. To change the default language, run this command for both FlowForceServer and FlowForceWebServer services (see *Syntax*).

#### **Syntax**

```
FlowForceServer setdeflang | sdl LanguageCode
FlowForceWebServer setdeflang | sdl LanguageCode
```

**Note:** On Linux systems, use an all-lowercase flowforceserver to call the executable.

The possible values of LanguageCode are as follows.

```
en English
es Spanish
de German
fr French
ja Japanese
```

#### **Example**

FlowForceServer setdeflang de

266 Command Line Interface start

## 10.15 start

#### **Purpose**

Starts FlowForce Server as a service. This command is used internally by the startup scripts or by the Windows service installation; it is not for general use.

#### **Syntax**

FlowForceServer start [options]

**Note:** On Linux systems, use an all-lowercase flowforceserver to call the executable.

datadir=VALUE	VALUE is the path of the database directory.

Command Line Interface uninstall 267

## 10.16 uninstall

#### **Purpose**

This command is executed by the FlowForce Server installer automatically and it is not available for general use. The command uninstalls Altova FlowForce Server as a service, on Windows. The command does not apply to Linux and Mac OS X

268 Command Line Interface upgradedb

# 10.17 upgradedb

#### **Purpose**

Upgrades the database to the latest version. The default database is upgraded automatically at installation time, so it is usually not necessary to run this command manually.

#### **Syntax**

FlowForceServer upgradedb [options]

**Note:** On Linux systems, use an all-lowercase flowforceserver to call the executable.

#### **Options**

datadir=VALUE	VALUE is the path of the database directory.
---------------	--

#### **Example**

Command Line Interface verifylicense 269

# 10.18 verifylicense

#### **Purpose**

This command is applicable to Windows platforms only. It can be used to verify whether FlowForce Server is licensed, and, optionally, whether a given license key is already assigned to FlowForce Server.

#### **Syntax**

FlowForceServer verifylicense [options]

1,license-key=VALUE	This option enables you to verify if a particular license key is already assigned to FlowForce Server.
	The value must be set to the license key that you wish to verify.

# Chapter 11

**Built-in Functions** 

## 11 Built-in Functions

This section provides reference information about the system functions built into FlowForce Server. The built-in functions allow you to copy or move files, create directories, execute shell commands, and perform various other actions. The FlowForce Server built-in functions are available in the **/system** container.

The following topics describe the built-in functions in groups, according to their path relative to the root container.

- /system/abort
- /system/compute
- /system/compute-string
- /system/filesystem
- /system/ftp
- /system/mail
- /system/maintenance
- /system/shell
- /RaptorXML
- /RaptorXMLXBRL

#### **Referring to Windows network paths**

When referring to a Windows network path, such as a mapped network drive, use the Universal Naming Convention (UNC) syntax. This is necessary because drive letters are not global to the system, and each logon session is assigned its own drive letters.

The UNC syntax is in the form:

\\server\sharedfolder\filepath

#### Where:

- server references the server name in the network (defined by the DNS)
- sharedfolder references a label defined by the administrator (for example, admin\$ is generally the root directory of the operating system installation).
- filepath refers to the subdirectories below the share.

Built-in Functions /system/abort 273

# 11.1 /system/abort

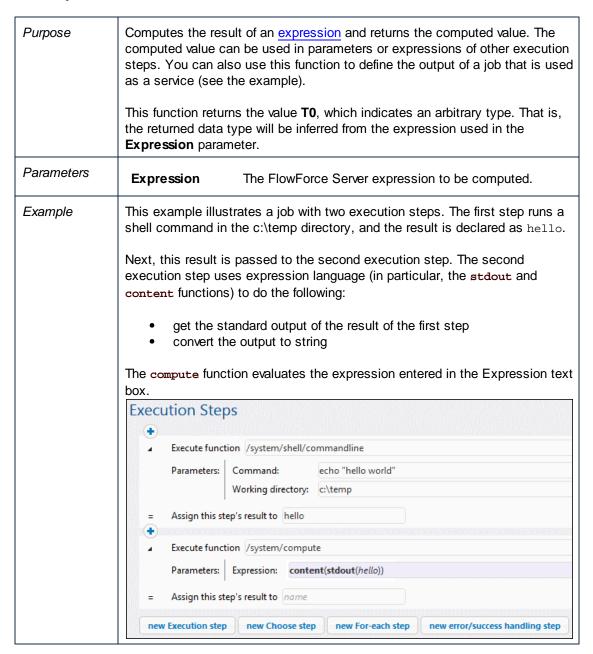
# 🗊 abort

Purpose	Aborts the execution of a job. This function is typically used inside a condition (that is, a <a href="Choose step">Choose step</a> ) to deliberately end the job when that condition is true. It is the equivalent of a <a href="throw">throw</a> or <a href="raise">raise</a> function in a programming language. This function does not return a value.		
Parameters	Message Mandatory string parameter. Specifies the message to output when aborting the job.		
Example	In the following job, the Dabort function is used to finish the job with an error if the value of a checked list exceeds 10 items. If the number of items in the list is less than or equal to 10, the job writes the text "The list has less than 10 items" to a file on the local system.    Execution Steps		
	Execute function /system/shell/commandline Parameters: Command: echo "The list has less than 10 items" > File.txt Working directory: c:\temp  = Assign this step's result to name as result  new When  Otherwise  Assign this step's result to name		

274 Built-in Functions /system/compute

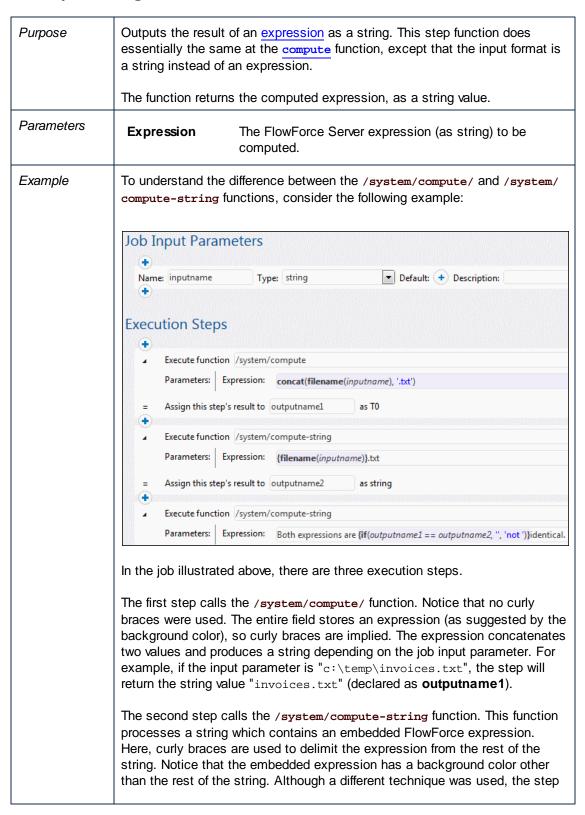
## 11.2 /system/compute

#### compute



## 11.3 /system/compute-string

#### compute-string



result (outputname2) is the same as outputname1.

Finally, the third step calls the <code>/system/compute/</code> function again, in order to compare the <code>outputname1</code> with <code>outpuname2</code>. If both values are identical, the result will be the string value "Both expression are identical". Otherwise, the result will be "Both expressions are not identical".

# 11.4 /system/filesystem

The **system/filesystem** container includes functions used to manage files and directories on the operating system where FlowForce Server runs.

All file paths in job execution steps must be paths on the operating system where FlowForce Server runs, not on your local machine.

## 🗊 сору

Purpose	Copy file(s).	
		ns Boolean TRUE if execution was successful. If the job e outcome depends on the value of the <b>Abort on error</b> ws:
	<ul> <li>If the Abort on error parameter is TRUE (default value) execution is aborted. In this case, you can still handle means of protected blocks (see <u>Handling Step Errors</u>).</li> <li>If the Abort on error parameter is FALSE, the function FALSE.</li> </ul>	
Parameters Source		The path and file name of the source file that you want to copy.
	Target	The path and file name of the destination directory. You can enter a different file name in the destination field if you want to rename it as well.
	Overwrite	When TRUE, causes the destination file to be overwritten. The default value is FALSE.
	Abort on error	This Boolean parameter determines what should be the return value of the function if the job fails. If <b>Abort on error</b> is FALSE, the function will return Boolean FALSE as well. If <b>Abort on error</b> is TRUE, the job execution is aborted. The default value is TRUE.
	Working- directory	Specifies the working directory (for example, c:\temp).

#### 1 delete

Purpose	Delete file(s).
	This function returns Boolean TRUE if execution was successful. If the job execution fails, the outcome depends on the value of the <b>Abort on error</b> parameter, as follows:

	<ul> <li>If the Abort on error parameter is TRUE (default value), the job execution is aborted. In this case, you can still handle errors by means of protected blocks (see <a href="Handling Step Errors">Handling Step Errors</a>).</li> <li>If the Abort on error parameter is FALSE, the function returns FALSE.</li> </ul>	
Parameters	Path	The path and file name of the file you want to delete.
	Abort on error	This Boolean parameter determines what should be the return value of the function if the job fails. If <b>Abort on error</b> is FALSE, the function will return Boolean FALSE as well. If <b>Abort on error</b> is TRUE, the job execution is aborted. The default value is TRUE.
	Working- directory	Specifies the working directory (for example, c:\temp).

# mkdir 🏐

Purpose	Create a directory at the specified path.		
	This function returns Boolean TRUE if execution was successful. If the job execution fails, the outcome depends on the value of the <b>Abort on error</b> parameter, as follows:  • If the <b>Abort on error</b> parameter is TRUE (default value), the job execution is aborted. In this case, you can still handle errors by means of protected blocks (see <a href="Handling Step Errors"><u>Handling Step Errors</u></a> ). • If the <b>Abort on error</b> parameter is FALSE, the function returns FALSE.		
Parameters	Path	The path of the new directory.	
	MakeParents	Clicking generates a check box. Activation allows a hierarchical path to be created in one step.	
	Abort on error	This Boolean parameter determines what should be the return value of the function if the job fails. If <b>Abort on error</b> is FALSE, the function will return Boolean FALSE as well. If <b>Abort on error</b> is TRUE, the job execution is aborted. The default value is TRUE.	
	Working- directory	Specifies the working directory (for example, c:\temp).	
Example	If Working-Directory is c:\temp, and Path is temp2\temp3, the function creates the new directory c:\temp\temp2\temp3.		

## move 🇊

Purpose	Move or rename file(s).		
	This function returns Boolean TRUE if execution was successful. If the job execution fails, the outcome depends on the value of the <b>Abort on error</b> parameter, as follows:		
	<ul> <li>If the Abort on error parameter is TRUE (default value), the job execution is aborted. In this case, you can still handle errors by means of protected blocks (see <a href="Handling Step Errors">Handling Step Errors</a>).</li> <li>If the Abort on error parameter is FALSE, the function returns FALSE.</li> </ul>		
Parameters	Source	The path and file name of the source file that you want to move.	
	Destination	The name of the destination directory. If you supply only the directory name in this field, the original file name will be retained.	
	Overwrite target	Clicking generates a check box. Activation causes destination file to be overwritten.	
	Abort on error	This Boolean parameter determines what should be the return value of the function if the job fails. If <b>Abort on error</b> is FALSE, the function will return Boolean FALSE as well. If <b>Abort on error</b> is TRUE, the job execution is aborted. The default value is TRUE.	
	Working- directory	Specifies the working directory (for example, c:\temp).	

## Tmdir

Purpose	Remove directory.		
	This function returns Boolean TRUE if execution was successful. If the job execution fails, the outcome depends on the value of the <b>Abort on error</b> parameter, as follows:		
	<ul> <li>If the Abort on error parameter is TRUE (default value), the job execution is aborted. In this case, you can still handle errors by means of protected blocks (see <a href="Handling Step Errors">Handling Step Errors</a>).</li> <li>If the Abort on error parameter is FALSE, the function returns FALSE.</li> </ul>		
Parameters	Path The name of the directory you want to delete.		
	Abort on error This Boolean parameter determines what should be the		

	return value of the function if the job fails. If <b>Abort on error</b> is FALSE, the function will return Boolean FALSE as well. If <b>Abort on error</b> is TRUE, the job execution is aborted. The default value is TRUE.
Working- directory	Specifies the working directory (for example, c:\temp).

# 11.5 /system/ftp

The  $\mbox{system/ftp}$   $\mbox{container}$  includes functions used to perform FTP operations on a remote FTP server.

## 🗊 delete

Purpose	Deletes a file from	the FTP server.
	This function returns Boolean TRUE if execution was successful. If the job execution fails, the outcome depends on the value of the <b>Abort on error</b> parameter, as follows:	
	<ul> <li>If the Abort on error parameter is TRUE (default value), the job execution is aborted. In this case, you can still handle errors by means of protected blocks (see <u>Handling Step Errors</u>).</li> <li>If the Abort on error parameter is FALSE, the function returns FALSE.</li> </ul>	
Parameters	FTP server	Address of the remote FTP server, either as a URL or IP address.
	Port	The port number used to connect to the FTP server
	Directory on host	The name of the directory, on the host, from which you want to delete a file.
	Login username	User name needed to connect to the host.
	Login password	Password needed to connect to the host.
	Use passive mode	Use passive mode if connection problems occur (for example, if routers or firewalls are set up to prevent active connections).
	Target file	The name of the file that you want delete from the server.
	Abort on error	This Boolean parameter determines what should be the return value of the function if the job fails. If <b>Abort on error</b> is FALSE, the function will return Boolean FALSE as well. If <b>Abort on error</b> is TRUE, the job execution is aborted. The default value is TRUE.
	Account	The FTP account name of the user allowed access to the files on the remote server.

# mkdir

	T	
Purpose	Creates a directory	on the FTP server.
	This function returns Boolean TRUE if execution was successful. If the job execution fails, the outcome depends on the value of the <b>Abort on error</b> parameter, as follows:	
	<ul> <li>If the Abort on error parameter is TRUE (default value), the job execution is aborted. In this case, you can still handle errors by means of protected blocks (see <a href="Handling Step Errors">Handling Step Errors</a>).</li> <li>If the Abort on error parameter is FALSE, the function returns FALSE.</li> </ul>	
Parameters	FTP server	Address of the remote FTP server, either as a URL or IP address.
	Port	The port number used to connect to the FTP server
	Directory on host	The name of the directory, on the host, from within which you want to create a new directory.
	Login username	User name needed to connect to the host.
	Login password	Password needed to connect to the host.
	Use passive mode	Use passive mode if connection problems occur (for example, if routers or firewalls are set up to prevent active connections).
	Target directory	The name of the directory that you want create on the server.
	Abort on error	This Boolean parameter determines what should be the return value of the function if the job fails. If <b>Abort on error</b> is FALSE, the function will return Boolean FALSE as well. If <b>Abort on error</b> is TRUE, the job execution is aborted. The default value is TRUE.
	Account	The FTP account name of the user allowed access to the files on the remote server.

# move 🇊

Purpose	Moves a file on the FTP server.
	This function returns Boolean TRUE if execution was successful. If the job execution fails, the outcome depends on the value of the <b>Abort on error</b> parameter, as follows:

	<ul> <li>If the Abort on error parameter is TRUE (default value), the job execution is aborted. In this case, you can still handle errors by means of protected blocks (see <u>Handling Step Errors</u>).</li> <li>If the Abort on error parameter is FALSE, the function returns FALSE.</li> </ul>	
Parameters	FTP server	Address of the remote FTP server, either as a URL or IP address.
	Port	The port number used to connect to the FTP server
	Directory on host	The name of the directory, on the host, from where you want to move the file.
	Login username	User name needed to connect to the host.
	Login password	Password needed to connect to the host.
	Use passive mode	Use passive mode if connection problems occur (for example, if routers or firewalls are set up to prevent active connections).
	Source file	Name of the source file that you want to move to a different location.
	Target file	Name of the copied file at the target location. Use a different name if you want to rename the copied file.
	Abort on error	This Boolean parameter determines what should be the return value of the function if the job fails. If <b>Abort on error</b> is FALSE, the function will return Boolean FALSE as well. If <b>Abort on error</b> is TRUE, the job execution is aborted. The default value is TRUE.
	Account	The FTP account name of the user allowed access to the files on the remote server.

# Tetrieve

Purpose	Retrieves a file from the FTP server.	
	This function returns Boolean TRUE if execution was successful. If the job execution fails, the outcome depends on the value of the <b>Abort on error</b> parameter, as follows:	
	<ul> <li>If the Abort on error parameter is TRUE (default value), the job execution is aborted. In this case, you can still handle errors by means of protected blocks (see <a href="Handling Step Errors">Handling Step Errors</a>).</li> <li>If the Abort on error parameter is FALSE, the function returns FALSE.</li> </ul>	

Parameters	FTP server	Address of the remote FTP server, either as a URL or IP address.
	Port	The port number used to connect to the FTP server
	Directory on host	The name of the directory, on the host, from where you want to retrieve the file.
	Login username	User name needed to connect to the host.
	Login password	Password needed to connect to the host.
	Use passive mode	Use passive mode if connection problems occur (for example, if routers or firewalls are set up to prevent active connections).
	Source file	Name of the source file that you want to retrieve.
	Target file	Name the file should have once it has been retrieved (change the name to rename it).
	Overwrite target	Clicking generates a check box. Activation causes destination file to be overwritten.
	Abort on error	This Boolean parameter determines what should be the return value of the function if the job fails. If <b>Abort on error</b> is FALSE, the function will return Boolean FALSE as well. If <b>Abort on error</b> is TRUE, the job execution is aborted. The default value is TRUE.
	Working directory	Specifies the working directory of the execution step.
	Account	The FTP account name of the user allowed access to the files on the remote server.

#### 🗊 rmdir

Purpose	Deletes a directory from the FTP server.	
	This function returns Boolean TRUE if execution was successful. If the job execution fails, the outcome depends on the value of the <b>Abort on error</b> parameter, as follows:	
	<ul> <li>If the <b>Abort on error</b> parameter is TRUE (default value), the job execution is aborted. In this case, you can still handle errors by means of protected blocks (see <u>Handling Step Errors</u>).</li> <li>If the <b>Abort on error</b> parameter is FALSE, the function returns FALSE.</li> </ul>	
Parameters	FTP server Address of the remote FTP server, either as a URL or IP	

	address.
Port	The port number used to connect to the FTP server
Directory on host	The name of the directory, on the host, from within which you want to remove a directory.
Login username	User name needed to connect to the host.
Login password	Password needed to connect to the host.
Use passive mode	Use passive mode if connection problems occur (for example, if routers or firewalls are set up to prevent active connections).
Target directory	The name of the directory that you want to remove on the server.
Abort on error	This Boolean parameter determines what should be the return value of the function if the job fails. If <b>Abort on error</b> is FALSE, the function will return Boolean FALSE as well. If <b>Abort on error</b> is TRUE, the job execution is aborted. The default value is TRUE.
Account	The FTP account name of the user allowed access to the files on the remote server.

#### 🗊 store

Purpose	Stores a file the FTP server.	
	This function returns Boolean TRUE if execution was successful. If the job execution fails, the outcome depends on the value of the <b>Abort on error</b> parameter, as follows:	
	execution means of	rt on error parameter is TRUE (default value), the job is aborted. In this case, you can still handle errors by protected blocks (see <a href="Handling Step Errors">Handling Step Errors</a> ). rt on error parameter is FALSE, the function returns
Parameters	FTP server	Address of the remote FTP server, either as a URL or IP address.
	Port	The port number used to connect to the FTP server
	Directory on host	The name of the directory, on the host, where you want to store the file.
	Login username	User name needed to connect to the host.

Login password	Password needed to connect to the host.
Use passive mode	Use passive mode if connection problems occur (for example, if routers or firewalls are set up to prevent active connections).
Source file	Name of the file that you want to store.
Target file	Name of the file to be stored at the target location. Use a different name if you want to rename the copied file.
Abort on error	This Boolean parameter determines what should be the return value of the function if the job fails. If <b>Abort on error</b> is FALSE, the function will return Boolean FALSE as well. If <b>Abort on error</b> is TRUE, the job execution is aborted. The default value is TRUE.
Working directory	Specifies the working directory of the execution step.
Account	The FTP account name of the user allowed access to the files on the remote server.

Built-in Functions /system/mail 287

# 11.6 /system/mail

The **system/mail** container includes the function send, used to send email.

**Note:** Before using this function, ensure the mail server settings are configured (see <u>Setting Mail Parameters</u>).

#### send sen

Purpose	Sends a mail to the specified recipients, generally the administrator.	
	This function returns Boolean TRUE if execution was successful. If the job execution fails, the outcome depends on the value of the <b>Abort on error</b> parameter, as follows:	
	<ul> <li>If the Abort on error parameter is TRUE (default value), the job execution is aborted. In this case, you can still handle errors by means of protected blocks (see <a href="Handling Step Errors">Handling Step Errors</a>).</li> <li>If the Abort on error parameter is FALSE, the function returns FALSE.</li> </ul>	
Parameters	From	Email address from which the mail is to be sent.
	То	Recipient's email address.
	Subject	Subject line of the message.
	Message body	Body text of the message. The message body supports ASCII as well as Unicode characters.
	Attachment	File name of the attachment sent with the email.
	Abort on error	This Boolean parameter determines what should be the return value of the function if the job fails. If <b>Abort on error</b> is FALSE, the function will return Boolean FALSE as well. If <b>Abort on error</b> is TRUE, the job execution is aborted. The default value is TRUE.

288 Built-in Functions /system/maintenance

# 11.7 /system/maintenance

The system/maintenance container includes functions used to perform maintenance operations on the server.

# archive-log

Purpose	Moves the older log records to an archive file on the server. Returns the name of the archive file that was created, as string value.	
Parameters	Older than, days	Archives files older than the number of days entered here. Default=30
	Archive directory	Archive directory name, (for example, c:\temp).  Mandatory.
	Archive file prefix	Prefix of the archive file. Default is "flowforcelog".
	Delete archived records	Deletes the archived records from the FlowForce database.
	Working directory	Specifies the working directory (for example, c:\temp).

# **1** cleanup-files

Purpose	Deletes those files that are not in use or referenced by any deployed objects (such as MapForce mappings and StyleVision transformations). Returns the number of files that were deleted, as numeric value.
	When you delete deployed objects, or when you re-deploy existing objects with modified files, any files associated with previously deployed objects become unused. By default, FlowForce Server does not delete the unused files. Therefore, in order to clean up the disk space, it is strongly recommended to create a job which periodically calls this function, especially in enterprise environments where multiple users deploy objects to FlowForce Server.
	To see the current disk space used by deployed objects, check the size of the <i>files</i> folder located in the FlowForce Server application data folder (see FlowForce Server Application Data).
Parameters	None

# Truncate-log

Purpose	Deletes log records older than the date supplied. Returns the number of
---------	---

Built-in Functions /system/maintenance 289

	records that were deleted, as numeric value.	
Parameters	Older than, Truncates (deletes) records older than the number of days entered here. The default value is 30.	

290 **Built-in Functions** /system/shell

#### 11.8 /system/shell

The system/shell container includes the function commandline, used to execute shell commands or scripts.

**Note:** To have FlowForce Server jobs read environment variables, they must be defined in scripts, and those scripts must be executed with the /system/shell/commandline function. Be aware that FlowForce Server is running a non-interactive shell, which means all behaviour specific to interactive shells is not applicable (such as executing .profile or .bashrc on Linux).

#### commandline

# Purpose Executes a shell command or a batch file. If the exit code from the last shell command is other than "0", the outcome is as follows: If the parameter **Abort on error** is TRUE (default), this function aborts execution. In this case, you can handle the error by means of protected blocks (see Handling Step Errors). If the parameter **Abort on error** is FALSE, the function returns the result of the shell command, including the standard output, the standard error, and the exit code. If the exit code from the last command is "0" (success), the function returns the result of the last shell command, as generic type. To handle the value returned by this function in another step or job, do the following: 1. Name the returned result by entering a value in the **Assign this** step's result to text box (for example, myresult). 2. Create a new step which executes either the function compute or compute-string, depending on what return type you need. 3. Enter as argument to the above function an expression which gets the desired part from the generic result by means of a FlowForce step result function. For example, enter the expression stdout(myresult) to get the standard output of the result as stream, and stderr(myresult) to get the standard error output stream. To get the same values as string, use content(stdout(myresult)) and content(stderr(myresult)), respectively. Note that the stdout function (and the job) will fail if the shell command does not return a standard output. Likewise, the stderr function will fail if there is no standard error. For further information, see <u>Handling Data Types in Steps</u> and <u>Step Result</u> Functions. Parameters Command Enter the shell command to execute.

Built-in Functions /system/shell 291

# Abort on error This Boolean parameter determines what should be the return value of the function if the job fails. If Abort on error is FALSE, the function will return the result of the shell command, including the standard output, the standard error, and the exit code. If Abort on error is TRUE, the job execution is aborted. The default value is TRUE. Working Specifies the working directory (for example, c:\temp). directory Example When you generate XSLT code in MapForce by clicking Generate code in XSLT2 on the File menu, a DoTransform.bat file is created. You can use the commandline function to execute the batch file at a certain time each day. **Execution Steps** Execute function /system/shell/commandline Parameters: Command: c:\temp3\DoTransform.bat Working directory: c:\temp3 You can also directly call the RaptorXML engine by entering a command line call into the Command field. n /system/shell/commandline Command: raptorxml xslt --input=c:/Test.xml --output=c:/Output.xml c:/Test.xslt For more information about RaptorXML Server, see the RaptorXML Server documentation (http://www.altova.com/documentation.html). For a step-by-step example of a job which calls the commandline function, see Example: Check if a Path Exists.

292 Built-in Functions /RaptorXML

### 11.9 /RaptorXML

The /RaptorXML container contains built-in functions applicable to RaptorXML Server. This container is present in FlowForce Server if you have licensed RaptorXML Server. The available functions are as follows (for more information, see the RaptorXML Server documentation).

- valany valany
- valdtd
- valjson
- valjsonschema
- valxml-withdtd
- valxml-withxsd
- valxquery valxquery
- valxsd
- valxslt
- wfany
- wfdtd
- wfjson
- wfxml
- xquery
- xslt 🗇

The /RaptorXML container is organized as follows:

- Functions placed directly in the /RaptorXML container are compatible with the 2014 version of FlowForce Server (which is the first version supporting RaptorXML functions).
- When a subsequent release adds enhancements, the latest functions are available in a separate container, where the container name denotes the release in which they were added.

To benefit from the latest FlowForce Server features, after each product upgrade, it is recommended that you modify your jobs to use the latest available functions.

For examples of using RaptorXML Server functions in FlowForce Server jobs, see:

- Using RaptorXML to Validate a Document
- Validate an XML Document with Error Logging

Built-in Functions /RaptorXMLXBRL 293

# 11.10 /RaptorXMLXBRL

The /RaptorXMLXBRL container contains built-in functions applicable to RaptorXML+XBRL Server. This container is present in FlowForce Server if you have licensed RaptorXML+XBRL Server. The available functions are as follows (for more information, see the RaptorXML+XBRL Server documentation).

- valany (XML, DTD, XML Schema)
- valany (XBRL)
- valdtd
- valjson
- valjsonschema
- valxbrl
- valxbrltaxonomy
- valxml-withdtd
- valxml-withxsd
- valxquery
- valxsd
- valxslt
- wfanv
- wfjson
- wfdtd
- wfxml
- xquery
- xslt

The /RaptorXMLXBRL container is organized as follows:

- Functions placed directly in the /RaptorXMLXBRL container are compatible with the 2014 version of FlowForce Server (which is the first version supporting RaptorXML+XBRL functions).
- When a subsequent release adds enhancements, the latest functions are available in a separate container, where the container name denotes the release in which they were added.

To benefit from the latest FlowForce Server features, after each product upgrade, it is recommended that you modify your jobs to use the latest available functions.

# Chapter 12

**Expression Functions** 

#### 12 **Expression Functions**

This section provides reference to the FlowForce expression functions. For conceptual information that will help you understand how to use expressions, see The FlowForce Expression Language.

The expression functions are grouped as shown below:

- Step Result Functions
- **Stream Functions**
- File System Functions
- File Path Functions
- List Functions
- String Functions
- **Boolean Functions**
- Runtime Information Functions

# 12.1 Step Result Functions

Step result functions allow you to process the result returned by jobs (or the result returned by execution steps within jobs).

#### exitcode

Purpose	Returns the numeric exit code of the result.	
Signature	exitcode(result) of type number	
Parameters	result The result of the step whose exit code you want to get.	
Example	See Adding Error Handling to a Job for an example.	

#### failed-step

Purpose	Returns the step result. Using this function is meaningful when you are <a href="handling errors">handling errors</a> with protected blocks. The failed-step function must be part of the On Error handler, otherwise the step where you are using it will fail because there is no erroneous step.  As such, this function does not indicate the step where the error occurred. To find the result's attributes, pass this function as argument to functions such as <a href="stdout">stdout</a> or <a href="stdout">stderr</a> , for example:  stderr(failed-step()) stdout(failed-step())	
Signature	failed-step() of type result	
Parameters	None	
Example	See the following examples: <ul> <li>Adding Error Handling to a Job</li> <li>Validate an XML Document with Error Logging</li> </ul>	

#### results

Purpose	Returns an array of streams of the specified result, optionally filtered by name. Use the function	
---------	--	--

	name	Optional parameter. When provided, filters by name a particular value in the result.
Example	· ·	ponent <b>CompletePO</b> produces a result consisting of nd if you intend to access the first one, use:
	{as-file(nth(re	esults(MapForceMapping, "CompletePO"), 0))}
	component. The nt	ion picks the array of streams from the MapForce function picks the first item from this array. Finally, the treates a file from the stream.
	See Creating a Job	from a StyleVision Transformation for another example.

#### stdout

Purpose	Some execution steps (such as those that run shell commands) return standard output. For example, the shell command dir returns a list of directories.
	When a step returns a result, FlowForce Server automatically assigns to it the generic type <b>result</b> . With the stdout function, you can get access to the standard output of <b>result</b> , as follows:
	stdout(result)
	where <b>result</b> is the value returned by some execution step.
	This function fails if <b>result</b> does not provide standard output.
Signature	stdout(result) of type stream
Parameters	result The result of the step whose standard output you want to get.
Example	See the following examples:  • Adding Error Handling to a Job • Validate an XML Document with Error Logging

#### stderr

Purpose	Returns the standard error of the result. Fails if the result does not provide a standard error.	
Signature	stderr(result)	of type stream

Parameters	result	The result of the step whose standard error you want to get.
Example	See Adding Er	ror Handling to a Job for an example.

300 Expression Functions Stream Functions

## 12.2 Stream Functions

Stream functions are used to process streams of data. You can pass streams to FlowForce Server either by means of Web services or from step results.

#### as-file

Purpose	Creates a file if the stream source is a file. Creates a temporary file if the stream source is not a file.	
Signature	as-file(stream) of type string	
Example	See Validate an XML Document with Error Logging	

#### content

Purpose	Reads the contents of the specified stream as text in the specified encoding.		
Signature	content(stream, encoding = 'UTF-8') of type string		

#### empty-stream

Purpose	Creates an empty stream.	
Signature	empty-stream() of type stream	

#### stream-from-string

Purpose	Creates a stream from a string using the supplied encoding. The content type is associated to the stream. This type of stream is not automatically saved as a file.
Signature	<pre>stream-from-string (string, encoding = "UTF-8", contenttype="text/plain") of type stream</pre>

#### stream-open

Purpose	Creates a stream from an existing file.	
Signature	stream-open(name, contenttype="application/octet-stream") of type stream	
Example	<pre>stream-open("C:\files\textfile.txt")</pre>	

# 12.3 File System Functions

File system functions permit access to the file system. To execute these functions, the job must use the credentials of a user account with corresponding access rights on the operating system.

#### list-files

Purpose	Lists the files in the path (which may terminate with a wildcard) and returns the resulting string list.  If the path does not end with a path separator and is not a wildcard, a search is made for exactly the specified item in the parent directory.	
Signature	list-files(path) of type "list of string"	

#### list-directories

Purpose	Lists the subdirectories in the path (which may terminate with a wildcard) and returns the resulting string list.		
Signature	list-directories(path) of type "list of string"		

#### read-lines

Purpose	Reads the lines from the given file and returns them as a list of strings.				
Signature	read-lines(file name, encoding="UTF-8") string"	of	type	"list	of

302 Expression Functions File Path Functions

### 12.4 File Path Functions

File path functions allow you to extract specific portions of paths and file names. You may need to do this, for example, if you are <u>polling a directory</u> and want to extract the file name that triggered the job from the **triggerfile** parameter (see also <u>File System Triggers</u>).

#### extension

Purpose	Extracts the file extension from a path.		
Signature	extension(path) of type string		
Example	The following expression:		
	<pre>extension("c:\temp\file.txt")</pre>		
	returns ".txt".		

#### filename

Purpose	Extracts the file name from a path.		
Signature	filename(path) of type string		
Example	The following expression:		
	<pre>filename("c:\temp\file.txt")</pre>		
	returns "file".		

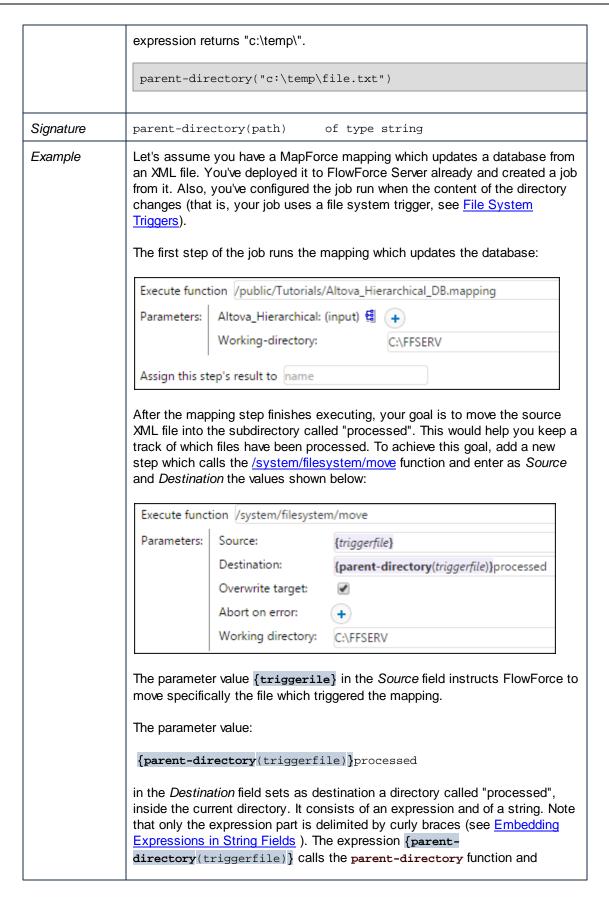
#### filename-with-extension

Purpose	Extracts the file name and extension from a path.	
Signature	filename-with-extension(path) of type string	
Example	The following expression:	
	filename-with-extension("c:\temp\file.txt")	
	returns "file.txt".	

#### parent-directory

Purpose Extracts the parent directory from a path. For example, the following
---

Expression Functions File Path Functions 303



304 Expression Functions File Path Functions

supplies to it the value "triggerfile" as argument.

Therefore, when the job runs, the following actions take place:

- 1. A script or a user copies a file (let's call it **source.xml**) into the current working directory (for example, **C:\FFSERV**).
- 2. The trigger fires and **source.xml** becomes the "triggerfile".
- 3. FlowForce Server executes the step which runs the mapping.
- 4. FlowForce Server executes the step which moves **source.xml** to the "processed" subdirectory. Note that the path **C:\FFSERV\processed** must exist.

Expression Functions List Functions 305

## 12.5 List Functions

List functions are used to create and disassemble lists. Lists always contain items of a single type (for example, only strings, only number, or only nested lists with the same item type); there are no mixed type lists.

#### char

Purpose	Returns a string that contains the Unicode character of the number supplied as argument. For example, char(10) returns a Line Feed.	
Signature	char(number)	of type string

#### code

Purpose	Returns the Unicode value of the first character of the string supplied as argument.	
Signature	code(string) of type number	

#### from-to

Purpose	Produces the list of integers between "from" and "to" inclusive. If "from" > "to", this list is empty.	
Signature	from-to(from, to) of type list of number	

#### join

Purpose	Concatenates the lists given by the first argument using the second argument as separator between each pair of lists.	
Signature	join(list of lists, separator = empty list) of type list	

## length

Purpose	Returns the number of items in the list.	
Signature	length(list)	of type number

#### list

Purpose	Builds a list from single items. All items must be of the same type, the
---------	--

306 Expression Functions List Functions

	resulting list is a list of items of that type.	
Signature	list(item1, item2,) of type list	

#### nth

Purpose	Returns the specified item from the list. The index is zero-based. Fails if the index is out of bounds.	
Signature	nth(list, index)	of type item

#### slice

Purpose	Selects a partial list from a list.	
	The argument "start" is the zero-based index of the first list item to include in the slice.	
	The argument "end" is the zero-based index of the first item to ignore in the slice.	
Signature	slice(list, start, end=length(list)) of type list	
Example	The following expression	
	slice(list(1,2,3,4),1,3)	
	returns list(2,3).	

Expression Functions String Functions 307

# 12.6 String Functions

The string functions perform basic string operations.

#### concat

Purpose	Concatenates/joins all of the separate strings into one string.	
	Same as string-join(list(string1, string2,)).	
Signature	concat(string1, string2,) of type string	

#### contains

Purpose	Returns TRUE if the first string contains at least one occurrence of substring, otherwise FALSE.	
Signature	contains(string, substring) of type boolean	

#### ends-with

Purpose	Returns TRUE if the string supplied in the <b>string</b> argument ends with the string supplied in the <b>end</b> argument.	
Signature	ends-with(string, end)	of type boolean

#### find-all

Purpose	Extracts all occurrences of pattern in the string, where pattern is a regular expression.	
Signature	find-all(string, pattern) of type list of string	

#### number

Purpose	Computes the number representation of the string, i.e. converts the string into a number.	
Signature	number(string)	of type number

#### split

Purpose	Splits the string at each occurrence of separator.
---------	--

**308** Expression Functions String Functions

Signature	split(string, separator)	of type list of string
-----------	--------------------------	------------------------

#### starts-with

Purpose	Returns TRUE if the string supplied in the <b>string</b> argument starts with the string supplied in the <b>start</b> argument.	
Signature	starts-with(string, start) of type boolean	

### string

Purpose	Computes the string representation of the given number, i.e. converts the number into a string.	
Signature	string(number)	of type string

### string-join

Purpose	Joins the "list of strings", inserts the separator in between each.	
Signature	string-join(list of strings, separator = an empty string) of type string	

## string-length

Purpose	Returns the number of characters in the string.	
Signature	string-length(string)	of type number

## substring

Purpose	Returns the specified substring. Start and end are zero-based character positions.
Signature	<pre>substring(string, start, end = string-length(string)) of type string</pre>

#### trim

Purpose	Removes leading and trailing whitespace from the string (Space, Tab, Line
	Feed, Carriage Return, Form Feed and Vertical Tab).

Expression Functions String Functions 309

Signature	trim(string)	of type string
-----------	--------------	----------------

#### trim-start

Purpose	Removes leading whitespace (see also trim).	
Signature	trim-start(string)	of type string

#### trim-end

Purpose	Removes trailing whitespace (see also trim).	
Signature	trim-end(string)	of type string

310 Expression Functions Boolean Functions

# 12.7 Boolean Functions

The Boolean functions are used to evaluate TRUE/FALSE expressions.

#### all

Purpose	Returns TRUE if all Boolean values are TRUE; stops evaluation after the first false value and returns FALSE.	
Signature	all(boolean1, boolean2,) of type boolean	

#### any

Purpose	Returns TRUE if any Boolean value is TRUE; stops evaluation after the first true value. Returns FALSE if all values are false.	
Signature	any(boolean1, boolean2,) of type boolean	

#### false

Purpose	Returns Boolean FALSE.	
Signature	false()	of type boolean

#### if

Purpose	Returns valueTrue if the Boolean is true, and valueFalse if false. Only the selected subexpression is evaluated. Both subexpressions must be of the same type, which is also the return type.  Example: To pass a Boolean as XML Schema conformant value, use   if(b, "true", "false") or if(b, "1", "0")
Signature	if(boolean, valueTrue, valueFalse) of type

#### not

Purpose	Returns the negation of the Boolean value supplied as argument.	
Signature	not(boolean) of type boolean	

Expression Functions Boolean Functions 311

#### true

Purpose	Returns Boolean TRUE.	
Signature	true()	of type boolean

# 12.8 Runtime Information Functions

The runtime information functions can be used to handle the details of the currently running jobs.

#### instance-id

Purpose	Returns a unique string for every job execution. This can be used to create a unique directory for each job execution, where the string is used to define the directory name.	
Signature	instance-id()	of type string

#### slot-number

Purpose	Returns the execution slot number of the queue currently running the job. This number should not be used as a file name. The number can be used to access different servers to execute parallel jobs (simple load balancing).  The slot number depends on the queue in which the slot execution was started. If the current job is called by another job, then it inherits the slot number of the calling job.	
Signature	slot-number() of type number	

# Chapter 13

**Job Configuration Examples** 

# 13 Job Configuration Examples

This section includes various FlowForce Server job configuration examples, as follows:

- Creating a Job from a MapForce Mapping
- Using a Job as Step of Another Job
- Creating a Directory Polling Job
- Adding Error Handling to a Job
- Exposing a Job as a Web Service
- Caching Job Results
- Creating a Job from a StyleVision Transformation
- Using RaptorXML to Validate a Document
- Validate an XML Document with Error Logging
- Using RaptorXML to Pass Key/Value Parameter Pairs

# 13.1 Creating a Job from a MapForce Mapping

This example shows you how to create a FlowForce Server job from a MapForce mapping. First, you will deploy a demo mapping file from MapForce to FlowForce Server. Deploying a mapping means that MapForce organizes the resources used by the mapping into an object and passes it to FlowForce Server. Once the mapping is deployed to FlowForce Server, you can create a server job from it. This particular example instructs you how to configure the mapping job to run daily at a specific time.

#### **Prerequisites**

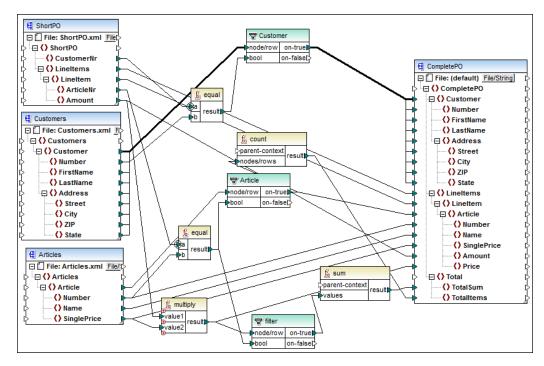
- Required licenses: MapForce Enterprise or Professional edition, MapForce Server or MapForce Server Accelerator Edition, FlowForce Server
- FlowForce Server is running at the configured network address and port (see <u>Setting the</u> Network Address and Port)
- You have a FlowForce Server user account with permissions to one of the <u>containers</u> (by default, the /public container used in this example is accessible to any authenticated user).
- The mapping job created in this example generates an XML file every time when it runs. Therefore, on the operating system where FlowForce Server runs, you must have rights to create files in some directory (this example uses the C:\temp directory).

#### **Demo files used**

 CompletePO.mfd, available at the following path: <Documents>\Altova\MapForce2017 \MapForceExamples.

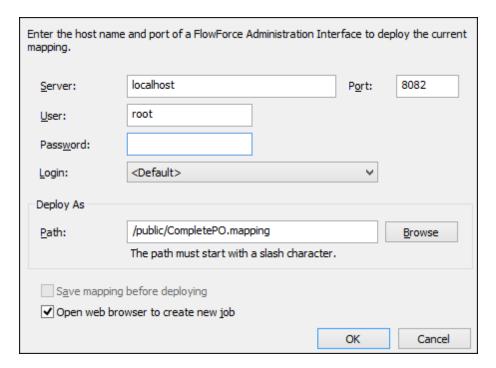
#### To create a job from a MapForce mapping:

1. Open the CompletePO.mfd file in MapForce.

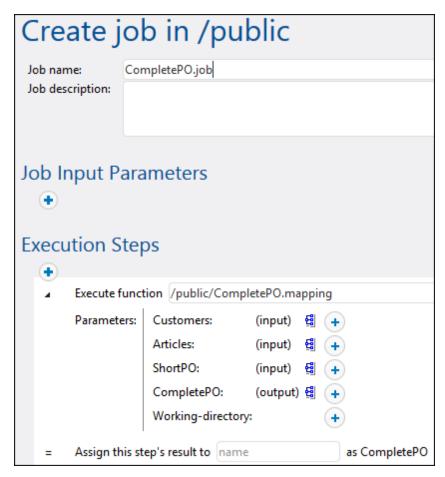


- 2. If you haven't done so already, set the transformation language of the mapping to "Built-in"
- 3. On the File menu, click Deploy to FlowForce Server.
- 4. Enter the server name and port of the Web administration interface in the respective fields (for example, localhost and 8082, if FlowForce Server is running on the same machine at the default host and port).
- 5. Enter your FlowForce Server user name and password, and, next to *Login*, leave the **Default** option as is.

If Windows Active Directory integration is enabled (see <u>Changing the Windows Active</u> <u>Directory Settings</u>), enter your Windows user name and password, and then select your Windows domain name from the *Login* drop-down list.



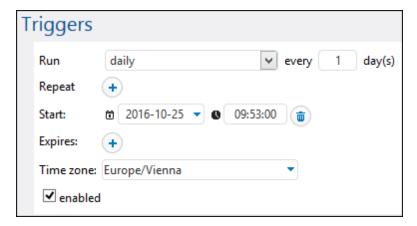
- 6. Make sure the **Open web browser to create new job** check box is selected. Although you can change the path where the mapping should be deployed, for the purpose of this example, leave the selected path as is.
- 7. Click **OK**. When deployment finishes, the FlowForce Server Administration Interface opens in your web browser, and a partially filled in job page is displayed. As shown below, an execution step is created automatically with some pre-filled parameters. For the scope of this example, leave the parameter values as is.



8. In the **Working-directory** box, enter the path to a directory where FlowForce Server must save the job output. This example uses C:\temp as working directory.

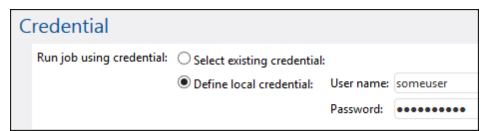
A working directory is a parameter required by <u>execution steps</u> if the job needs a location to unpack any input files or save output files. FlowForce Server also uses the working directory to resolve any relative paths that occur during step execution. When asked to provide a working directory, you should supply a valid path on the operating system where FlowForce Server runs. If you do not supply a working directory when creating the step, FlowForce Server uses a temporary directory.

- 9. Under Triggers, click **new Timer**.
- 10. Next to Run, set the timer to run **Daily** every **1** days. Next to Start, select a date and time when the job must start, for example:



11. Under Credentials, select an existing credential record or specify a local credential (see also Credentials).

A credential record is the combination of user name and password associated with a user account on the operating system where the FlowForce Server job runs. When you define a job in FlowForce Server, you must supply the credentials with which the job must be executed. Note that if the user account associated with the supplied credentials does not have sufficient rights on the operating system, the job cannot execute successfully.



12. Click Save. At the time and date specified in the trigger, FlowForce Server executes the mapping job. If the job executes successfully, the file generated as a result (CompletePO.xml) becomes available in the working directory specified previously. To see whether the job executed successfully, refer to the job log (see Viewing the Job Log).

## 13.2 Using a Job as Step of Another Job

This example shows you how to use a previously defined job as a step of another job. Since this example requires a previously created job, you should complete the <a href="Creating a Job from a MapForce Mapping">Creating a Job from a MapForce Mapping</a> example before completing this example.

As you may recall from the <u>Creating a Job from a MapForce Mapping</u> example, the **Complete PO.job** generates an XML file in a temporary folder every time when it runs. This example shows you how to do the following:

- 1. Create a job that copies the generated file from the temporary folder to an archive folder (we will call this job **copy2archive**).
- 2. Modify the **CompletePO.job** to include the **copy2archive** job as an additional execution step.

#### **Prerequisites**

- Required licenses: MapForce Enterprise or Professional edition, MapForce Server or MapForce Server Accelerator Edition, and FlowForce Server
- FlowForce Server is running at the configured network address and port (see <u>Setting the</u> Network Address and Port)
- You have a FlowForce Server user account with permissions to one of the <u>containers</u> (by default, the /public container is accessible to any authenticated user).
- The mapping job created in this example copies files from one directory to another. Therefore, on the operating system where FlowForce Server runs, ensure that both directories exist and that you have rights to create files in both directories (this example uses the C:\temp and C:\archive directory).
- Complete the steps described in the Creating a Job from a MapForce Mapping example.

#### **Demo files used**

 Complete PO.job available in the /public container of FlowForce Server administration page.

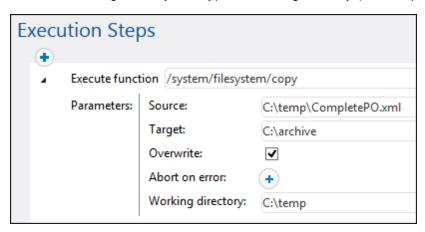
#### To create the "copy2archive" job:

- 1. Click **Configuration**, and then navigate to the **/public** container.
- 2. Click Create, and then select Create Job.
- 3. Enter the name of the job (in this example, "copy2archive").
- 4. Under Execution steps, add the first execution step, as follows:
  - a. Next to Execute function, navigate to the **system/filesystem/copy** function (for further information about functions, see <u>Built-in Functions</u>).
  - b. In the Source text box, type the path of the file to be copied (for example, "C:\temp \CompletePO.xml").
  - c. In the Target text box, type the destination path (for example, "C:\archive"). This must be an existing directory on the operating system where FlowForce Server runs. If you want to rename the file when it is copied, add the file name to the path, for example "C:\archive\PurchaseOrders.xml".
  - d. Click to select the **Overwrite** check box. This instructs FlowForce Server to overwrite any file with the same name found at the destination path.

e. Leave the Abort on error option as is.

This Boolean parameter determines what should be the return value of the function if the job fails. If **Abort on error** is FALSE, the function will return Boolean FALSE as well. If **Abort on error** is TRUE, the job execution is aborted. The default value is TRUE.

f. In the Working directory box, type the working directory (for example, "C:\temp").

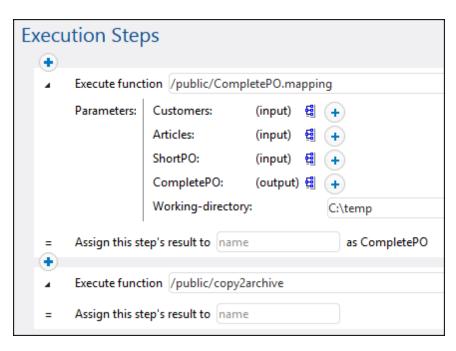


- 5. Under credentials, select an existing credential record or specify a local credential (see Credentials).
- 6. Click Save.

As you may have noticed, the job we just created does not have any trigger. We did not define any trigger because we will call this job from another job.

#### To add the "copy2archive" job as a step of "CompletePO" job:

- 1. In the /public container, click to open the CompletePO.job record.
- 2. Under Execution Steps, click **new Execution step** to add a new step after the existing one
- 3. Next to Execute function, browse for the location of the **copy2arhive** job. Now the execution steps look as follows:



- 4. Update the time trigger, and then click Save.
- 5. At the time entered in the trigger, FlowForce Server executes the job and copies the **CompletePO.xml** file from the temporary location to the archive location. To see whether the job executed successfully, refer to the job log (see Viewing the Job Log).

## 13.3 Creating a Directory Polling Job

This example shows you how to monitor a directory for new XML files using a file system trigger (see also <u>File System Triggers</u>). Whenever a new XML file is added to the directory, FlowForce Server executes a mapping job that takes the XML file as input parameter. The output of the mapping job is then copied to an archive directory.

#### **Prerequisites**

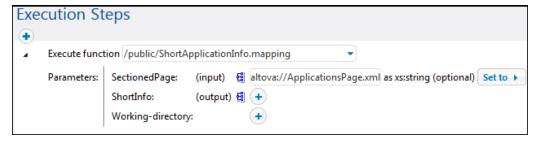
- Required licenses: MapForce Enterprise or Professional edition, MapForce Server or MapForce Server Accelerator Edition, and FlowForce Server
- FlowForce Server is running at the configured network address and port (see <u>Setting the</u> <u>Network Address and Port</u>)
- You have a FlowForce Server user account with permissions to one of the <u>containers</u> (by default, the **/public** container is accessible to any authenticated user).
- The mapping job created in this example copies files from one directory to another. Therefore, on the operating system where FlowForce Server runs, ensure that both directories exist and that you have rights to create files in both directories (this example uses the C:\temp and C:\archive directory).

#### **Demo files used**

- ShortApplicationInfo.mfd, available at the following path relative to the MapForce examples project: MapForceExamples > Mapping Folder > XML Schemas > Mixed Content. (To open the MapForce examples project in MapForce, click MapForceExamples.mfp on the Project menu.)
- **ApplicationsPage.xml**, available in the MapForce application data folder (the location of this folder depends on the operating system).

#### To create the directory polling job:

- Open the ShortApplicationInfo.mfd in MapForce and deploy it to a FlowForce Server container (for an example of how to deploy a mapping, see <u>Creating a Job from a</u> <u>MapForce Mapping</u>).
- In the FlowForce Server Web administration interface, open the mapping record from the container where you deployed it, and then click **Create Job**. The job page is displayed with some pre-filled parameters.



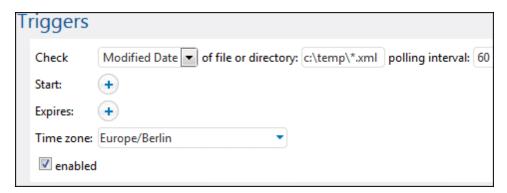
3. In the **Working-directory** box, enter the path to the working directory applicable to this step. This example uses C:\temp as working directory.

A working directory is a parameter required by <u>execution steps</u> if the job needs a location to unpack any input files or save output files. FlowForce Server also uses the working directory to resolve any relative paths that occur during step execution. When asked to provide a working directory, you should supply a valid path on the operating system where FlowForce Server runs. If you do not supply a working directory when creating the step, FlowForce Server uses a temporary directory.

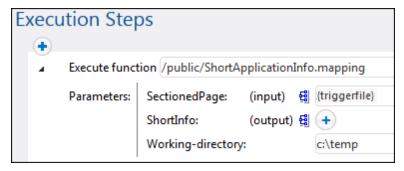
4. Under Triggers, click **new Filesystem trigger**. Notice that FlowForce Server automatically adds a new **triggerfile** parameter under Input Parameters.



- 5. Set the following trigger values:
  - · Check: Modified Date
  - File or directory: c:\temp\\*.xmlPolling interval: 60 seconds

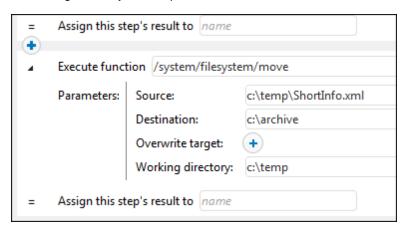


6. Under Execution Steps, pass the **{triggerfile}** parameter as input value to the **SectionedPage** parameter. To do this, click the **SectionedPage** parameter, and then select **{triggerfile}**. As a result, the value of the **SectionedPage** parameter changes to **{triggerfile}**.



7. Under Execution Steps, add a new execution step. This step will move the mapping output (**ShortInfo.xml**) from the working directory to the archive directory. Configure the step as follows (note that the source and destination fields are case-sensitive):

- Execute function: /system/filesystem/move
- Source: c:\temp\ShortInfo.xml
- Destination: c:\archive • Working directory: c:\temp



- 8. Under Credentials, select an existing credential record or specify a local credential (see Credentials).
- 9. Click Save.

You can now test the job by copying the file ApplicationsPage.xml to the working directory. When you do this, FlowForce Server executes the mapping job and copies the resulting output file to the archive directory.

To see whether the job executed successfully, refer to the job log (see Viewing the Job Log).

## 13.4 Adding Error Handling to a Job

This example shows how to add error handling to a simple job that lists the contents of a directory. More specifically, it shows you how to configure FlowForce Server as follows:

- Whenever the job fails to execute due to any reason, send an email notification to a named recipient.
- Whenever the job execution finishes, regardless of the execution status, log the job internal ID to a file on the local system.

In FlowForce Server terms, in this example you create a <u>protected block</u> with two error handling conditions: "On Error" and "Always" (each will handle one of the scenarios mentioned above).

#### **Prerequisites**

- Required licenses: FlowForce Server
- FlowForce Server is running at the configured network address and port (see <u>Setting the</u> Network Address and Port)
- The FlowForce Server mail settings have been configured (see <u>Setting the Mail</u> Parameters)
- You have a FlowForce Server user account with permissions to one of the <u>containers</u> (by default, the /public container is accessible to any authenticated user).

#### Tips

- This example uses FlowForce Server expressions, which are required to handle the job return values. To understand this example better, you might want to have a look at the instance-id, stderr, stdout, and failed-step expression functions (see <a href="Step Result Functions">Step Result Functions</a>).
- Although this example uses Windows paths and commands, you can still test it on other operating systems, if you change the paths and the commands accordingly.

#### To create the error handling job:

- On the machine where FlowForce Server runs, create a file called JobLog.txt (this file will be used in subsequent steps).
- 2. Log on to FlowForce Server and navigate to the container where you have permission to create new jobs (in this example we are using **/public**).
- 3. Click Create, and then select Create job.
- 4. Under Execution Steps, click the button, and then select **new error/success** handling step.
- 5. Under "Execute with error/success handling", click the button, and choose to add a new execution step, with the following settings:

Execute	Browse for the /system/shell/commandline function.
function	

Command	Enter the following shell command:			
	On Windows, this command lists recursively the contents of the working directory (see the next setting).			
Working directory	Set the value to an existing location on the machine where FlowForce Server runs, for example c:\			

6. Under the "On error" condition, click the button and choose to add a new execution step, with the following settings:

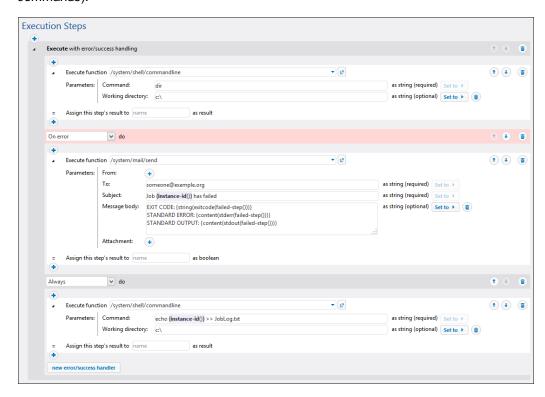
Execute function	Browse for the O/system/mail/send function.		
From	Enter the email address of the sender. Leave this field empty if you have configured the mail settings from the administration page.		
То	Enter your email address.		
Subject	Enter the subject of the notification email as follows:  Job {instance-id()} has failed		
Message body	Enter the following FlowForce Server expression:  Exit Code: {string(exitcode(failed-step()))} Standard Error: {content(stderr(failed-step()))} Standard Output: {content(stdout(failed-step()))} For information about the purpose of each function used here, see Step Result Functions.		

- 7. Click new error/success handler, and then select Always.
- 8. Under the "Always" condition, click the button and choose to add a new execution step, with the following settings:

Execute function	Browse for the /system/shell/commandline function.		
Command	Enter the following shell command:  echo {instance-id()} >> JobLog.txt  On Windows, this command writes the job ID to a file called  JobLog.txt. If the file contains data, the new text will be added after  the existing data.		
Working directory	Set the value to the directory where the <b>JobLog.txt</b> file created in previous steps is located. This must be an existing directory on the machine where FlowForce Server runs, for example c:\		

At this stage, the job should look as follows (provided you did not use different paths or shell

#### commands).



Under Triggers, add a Timer trigger which runs every N minutes (for example, 5 minutes).



- Under Credentials, select an existing credential record or specify a local credential (see <u>Credentials</u>).
- 11. Click Save.

At this stage, you have completed the job configuration, and you might want to do one of the following:

- To test the "Always" condition, wait for the trigger condition to be met. Whenever the
  trigger condition is met, a new job ID is appended to the contents of the JobLog.txt file.
- To test the "On Error" condition, change the parameters of the first step to some
  deliberately incorrect value (perhaps, by specifying a path that does not exist). If this
  case, FlowForce Server will send an email to the address specified in the recipient field of
  the "On Error" handler. Additionally, it will log the job ID in the JobLog.txt file, since this
  behaviour has been configured to happen in the "Always" condition.
- To see whether the job executed successfully, refer to the job log (see <u>Viewing the Job Log</u>).

## 13.5 Exposing a Job as a Web Service

This example shows you how to expose a sample mapping job as a Web service. After completing this example, you will be able to invoke the Web service from a browser. Once you supply a parameter to the Web service and click the **Submit** button, the result returned by the Web service is displayed in the browser.

#### **Prerequisites**

- Required licenses: MapForce Enterprise or Professional edition, MapForce Server or MapForce Server Accelerator Edition, FlowForce Server
- FlowForce Server is running at the configured network address and port (see <u>Setting the</u> Network Address and Port)
- You have a FlowForce Server user account with permissions to one of the <u>containers</u> (by default, the /public container is accessible to any authenticated user).

#### **Demo files used**

DB\_PhoneList.mfd, available at the following path relative to the MapForce examples project: MapForceExamples > Mapping Folder > Databases > WHERE/ORDER Component. (To open the MapForce examples project in MapForce, click MapForceExamples.mfp on the Project menu.)

#### To expose the mapping job as a Web service:

- Open the **DB\_PhoneList.mfd** in MapForce and deploy it to a FlowForce Server container (for an example of how to deploy a mapping, see <u>Creating a Job from a MapForce</u> <u>Mapping</u>). For the purpose of this example, it is recommended that you deploy the mapping to the **/public** container.
- 2. In FlowForce Server Web administration interface, open the mapping record from the container where you deployed it, and then click **Create Job**.
- Under Input Parameters, create a new input parameter of type string and name it NamePrefix.



Under Execution Steps, next to NamePrefix, click Set to, and then select NamePrefix.
 This sets the value of the mapping parameter NamePrefix to the value of the NamePrefix input parameter created in previous step.



5. Under Service, click to select the Make this job available via HTTP check box, and

enter GetPhoneList as name of the service.



- 6. Under Credentials, select an existing credential record or specify a local credential (see Credentials).
- 7. Click Save.

You can now invoke the Web service you just created.

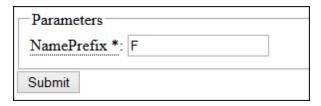
#### To invoke the Web service:

1. Open a Web browser and type the following URL in the address bar (replace [FlowForceServer] and [Port] with the settings configured in the administration page):

http://[FlowForceServer]:[Port]/service/GetPhoneList

If you use Internet Explorer to test FlowForce Server jobs exposed as Web services, disable the "Show friendly HTTP error messages" option in the **Advanced** tab.

- 2. When prompted to supply credentials, enter the credentials you use to access the FlowForce Server Web administration interface.
- 3. When prompted to enter the parameters of the Web service, enter F.



4. Click **Submit**. FlowForce Server processes the job and returns the result.

```
This XML file does not appear to have any style informa
- <PersonList xsi:noNamespaceSchemaLocation="C:/</p>
/PersonList.xsd">
 -<Person>
     <First>Joe</First>
     <Last>Firstbread</Last>
     <Details>+1 (321) 555 5155 - 621</Details>
   </Person>
 -<Person>
     <First>Carl</First>
     <Last>Franken</Last>
     <Details>+1 (927) 555 0094 - 147</Details>
   </Person>
 -<Person>
     <First>Frank</First>
     <Last>Further</Last>
     <Details>+1 (321) 555 5155 - 471</Details>
   </Person>
 </PersonList>
```

To display the parameter form again, click the browser's Back button.

To see whether the job executed successfully, refer to the job log (see Viewing the Job Log).

## 13.6 Caching Job Results

This example shows you how to cache the result of a job (referred to as <u>cache producer</u>) and use it in another job (referred to as <u>cache consumer</u>). Both jobs will be exposed as Web services with the following behavior:

- When the cache producer Web service is invoked, it lists recursively the contents of the directory, creates or updates the cache, and then outputs the result in the browser;
- When the cache consumer Web service is invoked, it reads the cache created by the cache producer service and outputs the result in the browser.

Our goal is to compare the execution time of both jobs, and see that the second job executes significantly faster than the first job, since it consumes cached data.

#### **Prerequisites**

- Required licenses: FlowForce Server
- FlowForce Server is running at the configured network address and port (see <u>Setting the</u> Network Address and Port)
- You have a FlowForce Server user account with permissions to one of the <u>containers</u> (by default, the /public container is accessible to any authenticated user).

#### Tip

Although this example uses Windows paths and commands, you can still test it on other
operating systems, if you change the paths and the commands accordingly.

#### To create the cache producer and the cache consumer jobs:

- 1. Click **Configuration**, and then navigate to the **/public** container.
- 2. Click **Create**, and then select **Create Job**.
- 3. In the Job Name box, enter **DirectoryListing**.
- 4. Under Execution Steps, add a new execution step with the following settings:

Execute function	Browse for the /system/shell/commandline function.		
Command	Enter the following shell command:		
	dir /s		
	On Windows, this command lists recursively the contents of the working directory (see the next setting).		
Working directory	Set the value to a directory on the machine where FlowForce Server runs, for example:		
	c:\		
Assign this step's result to	We will need to refer to the value returned by the execution step in a subsequent step, so it must have a name. For the		

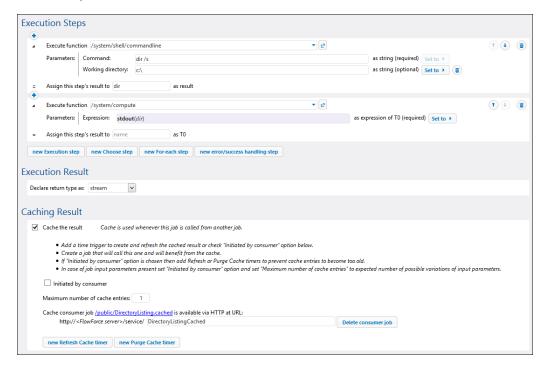
scope of this example, enter dir as value of this field.

5. Under Execution Steps, add a new execution step with the following settings:

Execute function	Browse for the /system/compute function.		
Expression	Enter the following FlowForce Server expression:		
	stdout(dir)		
	The stdout function converts the raw result returned by the previous execution step into a stream of data (see <a href="Step">Step</a> <a href="Result Functions">Result Functions</a> ).		

- 6. Under Execution Result, set the return type to **stream**. As you might have noticed, we set it to the same data type as returned by the last execution step of the job.
- 7. Under Caching Result, select the **Cache the result** check box.
- 8. Select the **Auto-create a new cache consumer job** check box, and then enter **DirectoryListingCached** as the name of the Web service.
- 9. Under Service, click to select the **Make this job available via HTTP** check box, and enter **DirectoryListing** as name of the service.

At this stage, the job should look as follows (provided you did not use different paths or shell commands).



- Under Credentials, select an existing credential record or specify a local credential (see <u>Credentials</u>).
- 11. Click Save.

At this stage, you have completed the configuration of both the cache producer and the cache consumer jobs. To compare the performance of both jobs, you can do the following:

- Access the URL of the cache producer service (DirectoryListing) in the browser, that
  is http://[FlowForceServer]:[Port]/service/DirectoryListing (if you are having
  trouble accessing the Web service, check the configured network address and port). Note
  that, because the job was configured to list the contents of the c:\ directory recursively, it
  might take up to several minutes to complete.
- 2. Refer to the job log to see how long it took for the job to complete (see <u>Viewing the Job Log</u>).
- 3. Next, do the same for the cache consumer service (DirectoryListingCached). Since this service consumes the cache rather than executing the directory listing, it is expected to take significantly less time to complete.

## 13.7 Creating a Job from a StyleVision Transformation

This example shows you how to create a FlowForce Server job from a StyleVision transformation. First, you will deploy a demo transformation file from StyleVision to FlowForce Server. Deploying a transformation file means that StyleVision organizes the resources used by the transformation into an object and passes it to FlowForce Server. Once the transformation is deployed to FlowForce Server, you will create a server job from it. The job will consist of three steps, namely:

- 1. The first step will execute the StyleVision transformation.
- 2. Because the transformation returns an array of multiple streams, the second step will access one of the several files created by the transformation, using a FlowForce Server expression.
- 3. The third step will copy the file to an archive folder.

#### **Prerequisites**

- Required licenses: StyleVision Enterprise or Professional edition, StyleVision Server, FlowForce Server
- FlowForce Server is running at the configured network address and port (see <u>Setting the</u> Network Address and Port)
- You have a FlowForce Server user account with permissions to one of the <u>containers</u> (by default, the /public container used in this example is accessible to any authenticated user).
- On the operating system where FlowForce Server runs, the following directories must exist and you must have rights to create files in them:
  - o c:\archive
  - o c:\temp

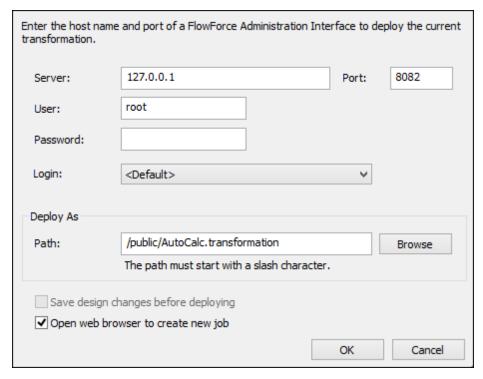
#### Demo files used

 AutoCalc.sps, available at the following path relative to the StyleVision examples project: Examples > Basics AutoCalc.sps. (To open the StyleVision examples project in StyleVision, click Examples on the Project menu.)

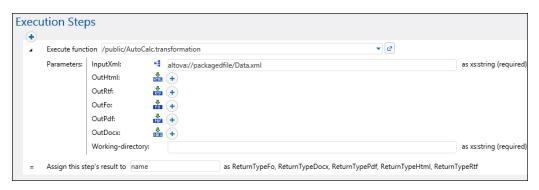
#### To create a job from a StyleVision transformation:

- 1. Open the **AutoCalc.sps** file in StyleVision.
- 2. On the **File** menu, click **Deploy to FlowForce...**. If this option is disabled, make sure the Design tab is currently selected.
- 3. When prompted to save the transformation as PXF file, leave the default settings as is, and click OK.
- 4. Enter the server name and port of the Web administration interface in the respective fields (for example, localhost and 8082, if FlowForce Server is running on the same machine at the default port).
- 5. Enter your FlowForce Server user name and password, and, next to *Login*, leave the **Default** option as is.

If Windows Active Directory integration is enabled (see <u>Changing the Windows Active</u> <u>Directory Settings</u>), enter your Windows user name and password, and then select your Windows domain name from the *Login* drop-down list.



- 6. Make sure the **Open web browser to create new job** check box is selected. Although you can change the path where the transformation should be deployed, for the purpose of this example, leave the selected path as is.
- 7. Click **OK**. When deployment finishes, the FlowForce Server Administration Interface opens in your web browser, and a partially filled in job page is displayed. As shown below, an execution step is created automatically with some pre-filled parameters.



- 8. Next to Parameters / OutHtml, click the button. This declares the AutoCalc.html as output file.
- In the Assign this step's result box, enter step1. This instructs FlowForce Server that
  the result returned by the first execution step of the job has the name step1. We will
  need to refer to this result subsequently.
- 10. In the Working-directory box, enter the path to a directory where FlowForce Server must

save the job output. This example uses c:\temp as working directory.

A working directory is a parameter required by <u>execution steps</u> if the job needs a location to unpack any input files or save output files. FlowForce Server also uses the working directory to resolve any relative paths that occur during step execution. When asked to provide a working directory, you should supply a valid path on the operating system where FlowForce Server runs. If you do not supply a working directory when creating the step, FlowForce Server uses a temporary directory.

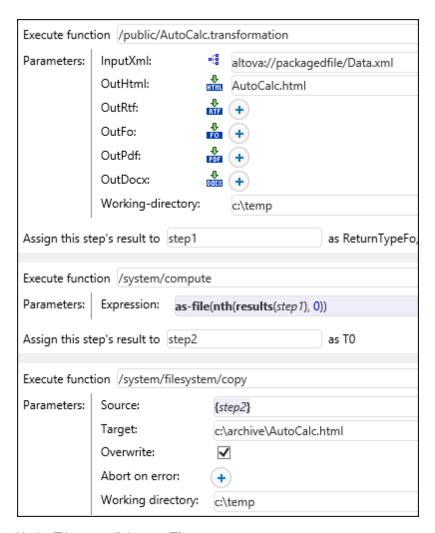
11. Under the execution step that is already created, click **new Execution step** and configure the step as follows:

Execute function	Browse for the <a>Total</a> <a></a>			
Expression	Enter the following FlowForce Server expression:  as-file(nth(results(step1), 0))  This expression instructs FlowForce Server to do the following:  1. Call the expression function results to get the array returned by step1 declared previously.  2. Pass this array to function nth to get the first item in the array. Since the array index is zero-based, we are using 0 as second argument of function nth.  3. Pass the value to function as-file to declare it as a file.			
Assign this step's result to	Enter <b>step2</b> . This instructs FlowForce Server that the result returned by the step has the name <b>step2</b> . We will need to refer to this result subsequently.			

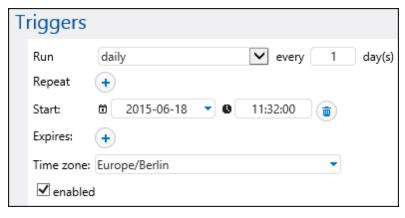
12. Click **new Execution step** and configure the step as follows:

Execute function	Browse for the 7/system/copy function.		
Source	Click Set to , and then select step2.		
Target	c:\archive\AutoCalc.html		
Overwrite	Select the <b>Overwrite</b> check box.		
Working directory	c:\temp		

At this stage, the Execution Steps section of the job page should look as follows:

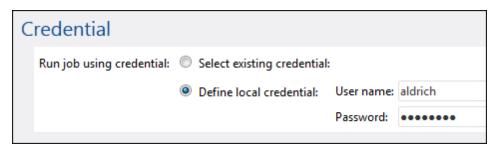


- 13. Under Triggers, click **new Timer**.
- 14. Next to Run, set the timer to run **Daily** every **1** days. Next to Start, select a date and time when the job must start, for example:



15. Under Credentials, select an existing credential record or specify a local credential (see also <u>Credentials</u>).

A credential record is the combination of user name and password associated with a user account on the operating system where the FlowForce Server job runs. When you define a job in FlowForce Server, you must supply the credentials with which the job must be executed. Note that if the user account associated with the supplied credentials does not have sufficient rights on the operating system, the job cannot execute successfully.



16. Click **Save**. At the time and date specified in the trigger, FlowForce Server executes the StyleVision transformation job. If the job executes successfully, the AutoCalc.html file becomes available in the c:\archive directory. To see whether the job executed successfully, refer to the job log (see Viewing the Job Log).

## 13.8 Using RaptorXML to Validate a Document

This example shows you how to create a job which validates an XML Schema file. The validation job uses the **valany** RaptorXML function built into FlowForce Server. For a list of RaptorXML functions, see RaptorXML Server Functions.

#### **Prerequisites**

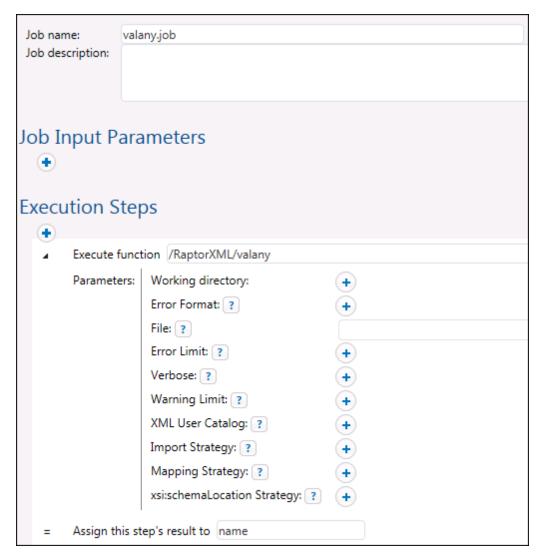
- Required licenses: FlowForce Server, RaptorXML (or RaptorXML+XBRL) Server
- FlowForce Server is running at the configured network address and port (see <u>Setting the</u> Network Address and Port)
- You have a FlowForce Server user account with permissions to one of the <u>containers</u> (by default, the /public container used in this example is accessible to any authenticated user).

#### **Demo files used**

• This example uses the address.xsd file available in the RaptorXML Server installation folder. On a Windows 8 system running 32-bit FlowForce Server, the path would be c: \Program Files (x86)\Altova\RaptorXMLServer2017\examples\address.xsd, unless you installed RaptorXML Server in a different folder.

#### To create the validation job:

- 1. Click **Configuration**, and then click the "RaptorXML" container.
- 2. Click the valary function (you can also find this function in any container that corresponds to a specific RaptorXML release, for example **2016r2**).
- 3. Click the **Create Job** button at the bottom of the page. This creates a job with the default name "valany.job". Edit the job name if necessary.



The parameters applicable to the **valany** function are now visible on the page. Note that the **mandatory** parameter that you need to supply is shown as an expanded field.

4. In the **File** field, enter the path and file name of the file that you want to validate, for example, C:\Program Files (x86)\Altova\RaptorXMLServer2017\examples \address.xsd.



- 5. Under Triggers, click **new Timer** and create a trigger that will run the job at a specific time in future (see also Timer Triggers).
- 6. Under Credentials, select an existing credential record or specify a local credential (see also <u>Credentials</u>).

7. Click **Save**. The job will run at the time specified in the trigger. To see whether the job executed successfully, refer to the job log (see <u>Viewing the Job Log</u>). Specifically, in the log entry details page, result="OK" means that the validation process was successful, for example:

```
file:///C:/Program%20Files%20(x86)/Altova/RaptorXMLServer/examples/
address.xsd: runtime="0ms" result="0K"
```

If the file did not validate, the log entry displays result="Fail".

## 13.9 Validate an XML Document with Error Logging

This example shows you how to create a job which validates an XML file against a schema. If the job fails due to any reason, the error details will be written to a log file. For validation, we will use the **valxml-withxsd** function of RaptorXML Server. (The RaptorXML Server functions become available in FlowForce Server when RaptorXML Server is licensed.) Note that, for the error logging part, the technique illustrated in this example is not dependent on RaptorXML Server and can be applied to other job types.

In this particular example, the job will be defined as a Web service, so that you can trigger it on demand, by accessing a URL from the browser. However, if necessary, you can also add to the job a timer (or file system) trigger, similar to how this is done in other examples. You could even add to the same job a combination of a trigger and a Web service. This way, the job will run not only as defined by the trigger rules, but also on demand, when the Web service is called.

#### **Prerequisites**

- Required licenses: FlowForce Server, RaptorXML (or RaptorXML+XBRL) Server
- FlowForce Server is running at the configured network address and port (see <u>Setting the</u> <u>Network Address and Port</u>)
- Your FlowForce Server user account has permissions to one of the <u>containers</u> (by default, the **/public** container used in this example is accessible to any authenticated user).
- The job created in this example generates a log file every time when it runs. Therefore, on the operating system where FlowForce Server runs, you must have rights to create files in some directory (this example uses the **C:\FlowForce** directory).

#### **Demo files used**

• The XML file validated in this example is available in the RaptorXML Server installation folder, at the following path: C:\Program Files\Altova\RaptorXMLServer2017 \examples\Nanonullorg.xml. On a 64-bit Windows running 32-bit FlowForce Server, the path would be C:\Program Files (x86)\Altova\RaptorXMLServer2017\examples \Nanonullorg.xml, unless you installed RaptorXML Server in a different folder.

#### To create the validation job:

- 1. Log on to FlowForce Server and navigate to a container where you have permission to create new jobs (in this example, we are using /public).
- 2. Click Create, and then select Create job.
- 3. Add a Job name (ValidateFile, in this example) and, optionally, a job description.



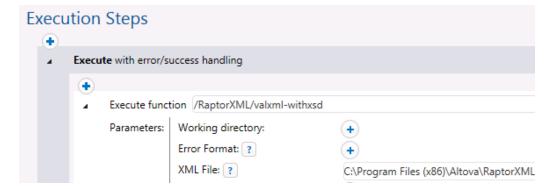
4. Under "Execution Steps", click the button, and then select new error/success

#### handling step.



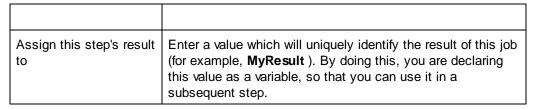
5. Under "Execute with error/success handling", click the button, and choose to add a new execution step, with the following settings:

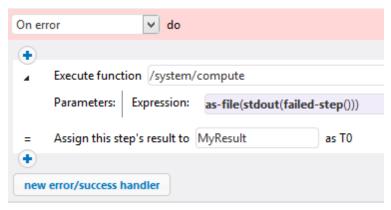
Execute function	Set the value to:		
	/RaptorXML/valxml-withxsd		
	You can also find this function in any container that corresponds to a specific RaptorXML release, for example / RaptorXML/2016r2/valxml-withxsd.		
Parameters	Set the value of the XML File parameter to:		
	C:\Program Files (x86)\Altova \RaptorXMLServer2017\examples\NanonullOrg.xml		



6. Under the "On error" condition, click the button and choose to add a new execution step, with the following settings:

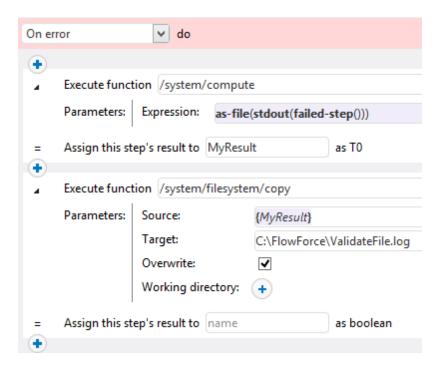
Execute function	/system/compute		
Parameters	Set the value of <b>Expression</b> to:		
	as-file(stdout(failed-step()))		





7. Under the previous execution step, click the button and choose to add a new execution step, with the following settings:

Execute function	/system/filesystem/copy			
Parameters	Next to the <b>Source</b> parameter, click and select the <b>MyResult</b> variable declared earlier.			
	Next to the <b>Target</b> parameter, type the path where the log will be saved (in this example, the path is <b>C:\FlowForce</b> \ <b>ValidateFile.log</b> ). Note that the <b>C:\FlowForce</b> directory must exist on the server where FlowForce Server runs, and you must have permission to write to it.			
	Next to the <b>Overwrite</b> parameter, select the check box. The log file is generated each time the job runs, so this ensures that the job does not fail when the log file already exists.			



- 8. Under Service, click to select the **Make this job available via HTTP** check box, and enter **ValidateFile** as name of the service.
- 9. Under "Credentials", select an existing credential record or specify a local credential (see <u>Credentials</u>).
- 10. Click Save.

To test the job, enter <a href="http://127.0.0.1:4646/service/ValidateFile">http://127.0.0.1:4646/service/ValidateFile</a> in the browser's address bar, provided that FlowForce Server runs at the default host and port name. If you have defined other host and port settings in the Configuration page (see <a href="Setting the Network Address and Port">Setting the Network Address and Port</a> ), change the address accordingly. If prompted for credentials when accessing the Web service, supply the same credentials you use to log on to FlowForce Server.

If the job executes successfully (that is, if it returns the exit code "0"), the browser displays the standard output of the job, for example:

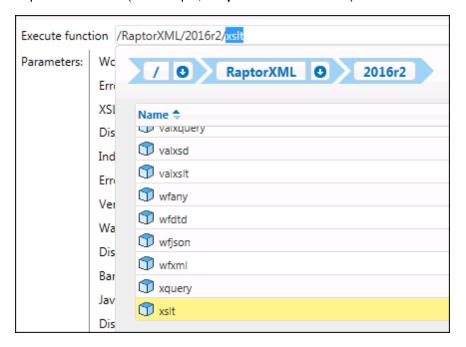
```
file:///C:/Program%20Files%20(x86)/Altova/RaptorXMLServer/examples/
NanonullOrg.xml: runtime="16ms" result="OK"
```

If the job returns an exit code other than "0" (for example, due to an incorrect path, validation errors, and so on), the browser displays a "Server execution failed" message and the output is written to the **C:\FlowForce\ValidateLog.log** file. In the event that the log file was not generated, check the log of the job in FlowForce Server to identify the error (see <u>Viewing the Job Log</u>). It may be the case, for example, that the **/system/filesystem/copy** function has failed because you have no permission to write to the target path, in which case the log file cannot be generated.

## 13.10 Using RaptorXML to Pass Key/Value Parameter Pairs

FlowForce Server can pass key and value pair parameters to XSLT stylesheets that are then executed by RaptorXML. In this generalized example, an XSLT file transforms an input XML file into a different language. A second XML file contains the original and translated values/terms. Two sets of related parameters are thus needed for this transformation.

In the job configuration page, browse for the Txslt function located in the /RaptorXML container. You can also find this function in any container that corresponds to a specific RaptorXML release (for example, /RaptorXML/2016r2/xslt):

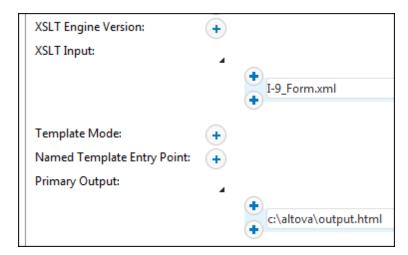


All the XSLT parameters are listed in the Parameters section.

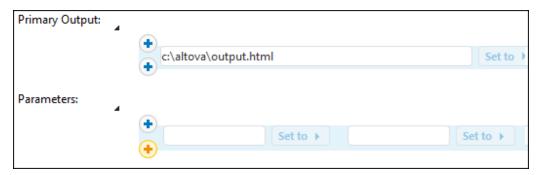
2. Under **XSLT File**, enter the path to the XSLT document (in this example, **MultiLangBy2ndFile.xslt**).



3. Under XSLT Input, enter the path to the XSLT input file (in this example, I-9\_Form.xml). Also, under Primary Output, enter the path to the generated output file (in this example, c:\altova\output.html).



4. Click the "+" button to the right of the Parameters label to create the first parameter pair.



5. Enter the "key" and then the "value" of that parameter in the field to the right. E.g. Language and 'G'.



6. Click the "+" button below these two fields and enter the second key and value pair, e.g. Translation and 'FormTranslation.xml'. The Translation element contains both language values of the FormTranslation.xml file.



Once you add a trigger to the job and run it, the XSLT transformation is executed with the supplied key/value parameter pairs.

Note: The contents of the "value" fields are XPath expressions and both must therefore be

enclosed in quotes to define them as of type "string" (single, or double quotes are OK). Alternatively, if you click the "Set to" button at the right of the parameters block, you can enter the XSLT expression in a single field, for example:



# Chapter 14

Glossary

## 14 Glossary

The glossary section includes the list of terms pertaining to FlowForce Server.

Glossary C 353

## 14.1 C

#### Cache consumer

A job that fetches data from the cache if it is available (in other words, it "consumes" the cache) . If the cached result is not available, the consumer executes the job and retrieves the actual job result.

#### Cache producer

A job that populates cache with data (in other words, it "produces" the cache).

#### **Choose step**

A step of type "Choose step" lets you define the conditions under which other job steps should be executed. Choose steps have the following structure:

```
When {some expression}
Execute (some step)
Otherwise
Execute (some other step)
```

Under each "Choose step", you can nest other Choose steps (sub-conditions), for example:

```
When {expression}
When {expression}
Execute (step)
Otherwise
Execute (step)
Otherwise
Execute (step)
```

Any number of conditional steps can be defined.

Within any when / Otherwise pair, FlowForce Server executes only the condition that is true; the other condition is ignored.

#### Container

As the name of the term implies, a container is data packaged together. In FlowForce Server, containers can be roughly compared to folders on an operating system. Containers can contain any of the following: <a href="jobs">jobs</a>, <a href="credentials">credentials</a>, <a href="functions">functions</a>, and other containers. By setting <a href="permissions">permissions</a> on containers, you can control who can view or access the data inside them. Organizing data into containers and setting up the relevant permissions for each container is a good security practice.

#### **Credentials**

A credential record is the combination of user name and password associated with a user account on the operating system where the FlowForce Server job runs. When you define a job in

**354** Glossary C

FlowForce Server, you must supply the credentials with which the job must be executed. Note that if the user account associated with the supplied credentials does not have sufficient rights on the operating system, the job cannot execute successfully.

Glossary E 355

## 14.2 E

#### **Error/Success Handling step**

When a step of a job fails, FlowForce Server aborts the job. You can use <u>steps</u> of type "Error/ success handling step" to perform clean-up actions before the job finally exits (see <u>Handling Step Errors</u>).

#### **Execution result**

In FlowForce Server, a step's execution result defines what is returned after the step is executed (for example, a file, or some text). When working with jobs, you can explicitly declare a step's execution result to be of a specific data type (such as String or Boolean), or be discarded. You typically need to declare the data type of the execution result if you intend to use it in other jobs, or if you want to cache the result.

#### **Execution step**

A <u>step</u> of type "Execution step" lets you execute a specific FlowForce function. Available functions include the <u>built-in functions</u> delivered with FlowForce Server, deployed MapForce mappings or StyleVision transformations, and the execution steps of other jobs.

356 Glossary F

## 14.3 F

#### For-Each step

A <u>step</u> of type "For-each step" lets you iterate through a sequence (for example, a list of files within a directory) and repeat an execution step any number of times. "For-Each" steps has the following structure:

```
For each item in sequence \{\text{sequence } \frac{\text{expression}}{}\}
Execute (\text{step})
```

FlowForce executes the step until it finishes looping through all items of the sequence expression.

#### **Function**

In the context of a <u>job</u>, a function is an instruction understood by FlowForce Server that performs some operation on the target file system. A function can be one of the following:

- A built-in FlowForce function (see Built-in Functions)
- A StyleVision transformation
- A MapForce mapping
- The execution step of a job

Most functions have input parameters. Any mandatory input parameters must be supplied by the caller in order for the step to execute successfully.

Glossary I 357

## 14.4

### **Input Parameters**

In the context of a FlowForce Server job, input parameters represent some arbitrary information supplied to the job based on which the job will execute in a particular way. Job input parameters are similar to function arguments in a programming language. Input parameters can be of various types (for example, file or directory references, text, numbers, Boolean values, and others). For some job types, FlowForce Server creates the input parameters automatically.

358 Glossary J

## 14.5 J

#### Job

A job is a core concept in FlowForce Server. It represents a task or a sequence of tasks to be executed by the server. Jobs can be as simple as one-step tasks such sending an email. However, you can also create jobs that perform multiple actions and pass the result (for example, a file) as parameter to another job. A job consists of <a href="input parameters">input parameters</a>, <a href="steps">steps</a>, <a href="triggers">triggers</a>, and other settings.

Glossary P 359

## 14.6 P

#### **Password policy**

FlowForce Servers uses password policies to help administrators manage the complexity of user passwords. A password policy is a set of minimum requirements that a user password must meet in order to be valid (for example, at least *N* characters long).

#### **Permission**

Permissions control user access to <u>containers</u>. Like privileges, <u>permissions</u> can be granted both to users and to <u>roles</u>. Therefore, if a user is a part of a role, any permissions granted to the role will automatically apply to the user as well.

By default, permissions set on a container are inherited from the parent container. For example, let's assume that container A has a child container B. Users who have permission to access container A will have by default permission to access container B as well. However, an administrator can redefine the permissions of any user or role at every level of the container hierarchy.

#### **Privilege**

Privileges define what users can do in FlowForce Server (for example, set own password, read users and roles, stop any job, and so on). Privileges are different from <u>permissions</u> in the sense that permissions control user access to <u>containers</u>, whereas privileges are effective globally across FlowForce Server. The following simple rule might help you distinguish quickly between privileges and permissions: privileges are global, permissions are local.

Like permissions, privileges can be assigned both to individual users and to <u>roles</u>. Therefore, when users log on to FlowForce Server, their set of effective privileges is determined by:

- a) the privileges they have been assigned directly
- b) the privileges assigned to any roles that the user is member of.

360 Glossary R

## 14.7 R

#### RaptorXML Server

Altova RaptorXML Server (also called RaptorXML for short) is Altova's third-generation, super-fast XML processor, optimized for the latest standards and parallel computing environments. Designed to be highly cross-platform capable, the engine takes advantage of today's ubiquitous multi-core computers to deliver lightning-fast processing of XML. RaptorXML is available in two editions: (i) RaptorXML Server and (ii) RaptorXML+XBRL Server. The RaptorXML+XBRL Server edition includes support for validating and processing XBRL (eXtensible Business Reporting Language) documents, in addition to XML.

#### RaptorXML+XBRL Server

RaptorXML+XBRL Server is a special edition of RaptorXML Server, which includes support for validating and processing XBRL (eXtensible Business Reporting Language) documents, in addition to XML.

#### Role

Roles are named sets of <u>privileges</u> that help enforce security based on the business need. The typical role-based security involves at least two roles: an administrator and a standard user. Each role is defined by the privileges granted to that role. For example, administrators can change their own password and that of other users, whereas standard users can change only their own password. You can assign roles to users and revoke roles from users as necessary.

Glossary S 361

## 14.8 S

#### Step

Steps define what the FlowForce Server job must actually do (for example, delete a file, execute a MapForce mapping, or send an email). In its simplest form, a step is just an operation with either failed or successful outcome, and it requires a <u>function</u> to be executed (see <u>Execution step</u>). However, there are also steps that provide means to execute other steps conditionally or in a loop (see <u>Choose step</u>, <u>Error/success Handling step</u>, and <u>For-Each step</u>). You can create as many steps as required within the same job, and you can set the order in which they must take place.

**362** Glossary T

## 14.9 T

## **Trigger**

When creating a job, you must specify the conditions (or criteria) that will trigger the job, known as triggers. FlowForce Server monitors any defined triggers and executes the job whenever the trigger condition is met.

Glossary U 363

## 14.10 U

#### User

Users are persons who log on to FlowForce Server to configure jobs, deploy MapForce or StyleVision transformations, or manage the FlowForce Server. The actions available to users in FlowForce Server depend on the following:

- a) Their assigned permissions or privileges
- b) The permissions and privileges assigned to any roles that users are members of.

364 Glossary W

## 14.11 W

#### Web administration interface

The FlowForce Server Web administration interface is the front end of FlowForce Server from where you administer the server and configure jobs. You can access the Web administration interface from a Web browser at the configured address and port.

#### **Working directory**

A working directory is a parameter required by <u>execution steps</u> if the job needs a location to unpack any input files or save output files. FlowForce Server also uses the working directory to resolve any relative paths that occur during step execution. When asked to provide a working directory, you should supply a valid path on the operating system where FlowForce Server runs. If you do not supply a working directory when creating the step, FlowForce Server uses a temporary directory.

# Index

## A

#### abort,

as built-in function, 273

#### Active Directory,

integration with FlowForce Server, 126

#### Administrator interface, 90

Alert emails, 102

#### Altova LicenseServer,

(see LicenseServer), 51

migrating to a new machine, 44

#### Altova ServiceController, 59

Assigning licenses, 84

## B

#### **Built-in functions**,

RaptorXML Server, 292

RaptorXML+XBRL Server, 293

## C

#### Cache consumer,

definition, 353

#### Cache producer,

definition, 353

#### Choose step,

definition, 353

#### Client machine monitoring, 101

Client management pane, 96

#### Command line interface,

getting help on, 252

### compute,

as built-in function, 274

## compute-string,

as built-in function, 275

#### Configuration page, 90

opening on Linux, 65

opening on Mac OS X, 67

opening on Windows, 62

URL of, 62

URL of (Linux), 65

URL of (Mac OS X), 67

#### Consumer,

definition, 353

#### Container,

definition, 353

restricting access to /public, 164

#### Containers,

creating, 158

moving, 158

overview of, 156

renaming, 158

setting permissions on, 162

viewing permissions of, 161

#### Credentials,

adding, 173

definition, 353

overview of, 172

## D

#### Data types,

in FlowForce, 226

#### Default password, 62

Default time zone,

configuration of, 124

#### Desktop products,

registering with LicenseServer, 72

#### Directory polling job,

example of, 323

## E

#### Environment variables,

setting, 132, 290

#### Error handling,

adding to a job, 326

#### Error logging,

adding to a job, 343

#### Error/success handling step,

definition, 355

Evaluation licenses, 96

Execution result,	
definition, 355	
Execution Step,	11
definition, 361	Н
Expressions, 296	
built-in functions, 274, 275	HTTP Triggers,
	overview of, 198
	HTTS connections,
	configuring FlowForce to accept, 117, 121
Г	
File System Triggers,	
overview of, 196	1
HowForce Server,	lowest Developmentage
application data folder, 113	Input Parameters,
architecture, 13	adding, 176
basic job concepts, 9	built-in, 176 definition, 357
basic security concepts, 11	removing, 176
built-in functions, 272	Installation,
changing the language of, 265	installation, installation folder (Linux), 28
command line interface, 250	installation folder (Mac), 34
configuration of, 112	installation folder (Windows), 43
definition, 3	on Linux, 22
glossary, 352	on Mac, 29
introduction to, 3	on Windows, 35
localization of, 131, 257	system requirements, 21
maintenance, 288	viewing currently installed products (Linux), 26
migrating to a new machine, 44	item,
migrating to latest version of, 262	as FlowForce data type, 226
new features in, 4	as 1 low1 ofce data type, 220
opening the setup page of, 115	
registering with LicenseServer, 73	•
setting the network address and port of, 117, 121	J
system requirements, 21	
Web administration interface, 15	Job,
FlowForce Server Function,	definition, 358
definition, 356	Jobs,
FlowForce Web Server,	caching results of, 202, 332
setting the network address and port of, 117, 121	copying, 175
For-each step,	creating, 174
definition, 356	creating from MapForce mapping, 315
FTP,	creating from StyleVision transformations, 335
built-in functions, 281	declaring the return type of, 188
Function,	defining queue settings of, 205
definition, 356	duplicating, 175
	executing shell commands with, 290
	exporting to another FlowForce Server instance, 208
	exporting to file, 208

Jobs.	starting services on, 129
handling result of, 297	stopping services on, 129
importing from file, 208	Mail,
stopping, 207	as built-in function, 287
viewing the log of, 206	Mail parameters,
viewing the log of, 200	configuration of, 125
	MapForce Server,
1.7	calling from a job, 238
K	migrating to a new machine, 44
	registering with LicenseServer, 78
Key/value pairs,	Messages, 108
passing to XSLT sheets, 347	Migrating to a new machine,
	Altova LicenseServer, 44
	FlowForce Server, 44
L	MapForce Server, 44
L	MobileTogether Server, 44
Lineway Road CO OO	RaptorXML(XBRL) Server, 44
License Pool, 69, 90	StyleVision Server, 44
Licenses,	MobileTogether Server,
assigning, 84, 96	migrating to a new machine, 44
managing, 96	registering with LicenseServer, 80
uploading, 69, 90	registering with Electiseserver, ou
LicenseServer,	
Configuration page, 90	<b>.</b> .
installation on Linux, 56	N
installation on Mac OS X, 58	
installation on Windows, 54	Network information, 53
interface with, 90	Network settings, 102
registering desktop products with, 72	
registering FlowForce Server with, 73	
registering MapForce Server with, 78	D
registering MobileTogether Server with, 80	Γ
registering RaptorXML(+XBRL) Server with, 81	B
registering StyleVision Server with, 82	Password,
settings, 102	default at startup, 62
starting, 61	Password policies,
steps for assigning licenses, 60	creating, 168
LicenseServer configuration page,	overview of, 167
(see Configuration page), 62, 65, 67	Password policy,
Linux,	definition, 359
starting services on, 128	Password reset, 109
stopping services on, 128	PDF files,
Logout, 108	generating, 238
	Permission,
	definition, 359
M	Permissions,
I V I	list of, 153
	overview of, 153

Mac,

Privilege,	
definition, 359	
Privileges,	C
inheritance, 146	3
list of, 146	
overview of, 146	Server Management tab, 84
viewing reports about, 150	ServiceController, 59
	Settings, 102
	Shell commands,
D	executing as jobs, 290
K	SSL,
	encrypting connections with, 117, 121
RaptorXML functions,	Step,
creating jobs with, 340, 343	definition, 361
RaptorXML Server,	Steps,
definition, 360	creating, 178
RaptorXML(+XBRL) Server,	example of using in jobs, 320
registering with LicenseServer, 81	handling errors in, 186
RaptorXML(XBRL) Server,	StyleVision Server,
migrating to a new machine, 44	calling from a job, 238
RaptorXML+XBRL Server,	migrating to a new machine, 44
definition, 360	registering with LicenseServer, 82
Registering desktop products with LicenseServer, 72	
Registering FlowForce Server with LicenseServer, 73	
Registering MapForce Server with LicenseServer, 78	Т
Registering MobileTogether Server with LicenseServer, 80	
Registering RaptorXML(+XBRL) Server with LicenseServer, 81	<b>T0,</b> as return value in functions, 274
Registering StyleVision Server with LicenseServer, 82	Timer Triggers,
Resettin the password, 109	overview of, 194
result,	Trigger,
as FlowForce data type, 226	Definition, 362
results,	triggerfile parameter,
as FlowForce data type, 226	purpose of, 176
Role,	Triggers,
definition, 360	creating, 192
Roles,	deleting, 192
assigning roles to, 143	
assigning to users, 142	
built-in roles, 140	U
creating, 137	U
renaming, 141	Unas vista visa and disease and disease and
root user,	Unregistering products and clients, 96
resetting the password of, 144	Uploading licenses, 69, 90
	Users,
	built-in roles, 140
	creating, 136
	renaming, 141

## W

### Web administration interface,

definition, 364

### Web Services,

exposing jobs as, 200, 329

#### Windows,

referring to network paths on, 272 starting services on, 130 stopping services on, 130

### Windows domain users,

importing into FlowForce Server, 138

### Working directory,

definition, 364