| Altova FlowForce Server | |
|---------------------------|--|
| | |
| User and Reference Manual | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

Altova FlowForce Server 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: 2014

© 2014 Altova GmbH

Table of Contents

| 1 | Altova FlowForce Server® | 3 |
|------|---|----|
| 2 | What's New | 6 |
| 3 | FlowForce Server Overview | 8 |
| 3.1. | Basic Concepts | 9 |
| 3.2 | Security Concepts | 12 |
| 3.3 | How It Works | 14 |
| 3.4 | Web Administration Interface | 16 |
| 4 | Installation | 20 |
| 4.1 | System Requirements | 21 |
| 4.2 | Linux | 22 |
| | 4.2.1 Installing FlowForce Server | |
| | 4.2.2 Migrating to FlowForce Server 2015 | |
| | 4.2.3Viewing Currently Installed Products | |
| | 4.2.4Uninstalling FlowForce Server | |
| | 4.2.5 Installation Folder | |
| 4.3 | Mac | |
| | 4.3.1 Installing FlowForce Server | |
| | 4.3.2 Migrating to FlowForce Server 2015 | |
| | 4.3.3Uninstalling FlowForce Server | |
| 4.4 | 4.3.4 Installation Folder | |
| 4.4 | Windows | |
| | 4.4.1 Installing FlowForce Server | |
| | 4.4.3 Installation Folder | |
| 5 | Licensing | 44 |
| 5.1 | Altova LicenseServer | 45 |
| • | 5.1.1 Network Information | |

| | 5.1.2Installation (Windows) | 48 |
|------|--|-----|
| | 5.1.3Installation (Linux) | 49 |
| | 5.1.4Installation (Mac OS X) | 51 |
| | 5.1.5Altova ServiceController | 52 |
| | 5.1.6How to Assign Licenses | 53 |
| | Start LicenseServer | 53 |
| | Open LicenseServer's Config Page (Windows) | |
| | Open LicenseServer's Config Page (Linux) | |
| | Open LicenseServer's Config Page (Mac OS X) | |
| | Upload Licenses to LicenseServer | |
| | Register Product/s | |
| | Register FlowForce Server | |
| | Register MapForce Server | |
| | Register RaptorXML(+XBRL) Server | 74 |
| | Register MobileTogether Server | |
| | Assign Licenses to Registered Products | |
| | 5.1.7Configuration Page Reference | |
| | License Pool | |
| | Server Management | |
| | Server Monitoring | |
| | Settings Messages, Log Out | |
| 6 | Configuring the Server | 96 |
| 6.1. | FlowForce Server Application Data | 97 |
| | Opening the Setup Page | |
| | Setting the Network Address and Port | |
| | Setting the Default Time Zone | |
| | Setting Mail Parameters | |
| | Changing the Windows Active Directory Settings | |
| | Starting and Stopping Services (Linux) | |
| | SStarting and Stopping Services (Mac) | |
| | Starting and Stopping Services (Windows) | |
| | 0Localizing FlowForce Server | |
| 0.10 | ULocalizing Flowforce Server | 109 |
| 7 | Managing User Access | 112 |
| 7.1. | Users and Roles | 113 |
| | 7.1.1Creating Users | 114 |

| | 7.1.2Creating Roles | 115 |
|--|--|---|
| | 7.1.3Importing Domain Users and Roles | 116 |
| | 7.1.4Default Users and Roles | 118 |
| | 7.1.5Renaming Users and Roles | 119 |
| | 7.1.6Assigning Roles to Users | 120 |
| | 7.1.7Assigning Roles to Other Roles | 121 |
| 7.2 | .Privileges | 122 |
| | 7.2.1How Privileges Work | 123 |
| | 7.2.2Viewing Privilege Reports | 127 |
| 7.3 | Permissions and Containers | 129 |
| | 7.3.1How Permissions Work | 130 |
| | 7.3.2Understanding Containers | 133 |
| | 7.3.3Creating, Renaming, and Moving Containers | 135 |
| | 7.3.4Viewing Container Permissions | 138 |
| | 7.3.5Setting Container Permissions | 139 |
| | 7.3.6Restricting Access to the /public Container | 141 |
| 7.4 | Password Policies | 143 |
| | 7.4.1How Password Policies Work | 144 |
| | 7.4.2Creating and Assigning Password Policies | 145 |
| | | |
| | | |
| 8 | Working with Jobs 1 | 48 |
| 8 8.1 | Working with Jobs 1Credentials | |
| _ | | 149 |
| _ | Credentials | 149 150 |
| 8.1 | Credentials | 149 150 151 |
| 8.1 | Credentials | 149 150 151 152 |
| 8.1 8.2 8.3 | Credentials 8.1.1How Credentials Work 8.1.2Defining CredentialsCreating Jobs | 149 150 151 152 154 |
| 8.2 8.3 8.4 | Credentials 8.1.1How Credentials Work 8.1.2Defining CredentialsCreating JobsAdding Input ParametersAdding Steps | 149 150 151 152 154 155 |
| 8.2 8.3 8.4 | Credentials 8.1.1How Credentials Work 8.1.2Defining CredentialsCreating JobsAdding Input ParametersAdding StepsHandling Step Errors | 149 150 151 152 154 155 157 |
| 8.2 8.3 8.4 8.5 | Credentials 8.1.1How Credentials Work 8.1.2Defining CredentialsCreating JobsAdding Input ParametersAdding StepsHandling Step ErrorsDeclaring the Return Type of a Job | 149 150 151 152 154 155 157 |
| 8.2 8.3 8.4 8.5 8.6 | Credentials 8.1.1How Credentials Work 8.1.2Defining CredentialsCreating JobsAdding Input ParametersAdding StepsHandling Step ErrorsDeclaring the Return Type of a JobHandling Data Types in Steps | 149 150 151 152 154 155 157 159 160 |
| 8.2 8.3 8.4 8.5 8.6 | Credentials 8.1.1How Credentials Work 8.1.2Defining CredentialsCreating JobsAdding Input ParametersAdding StepsHandling Step ErrorsDeclaring the Return Type of a JobHandling Data Types in StepsAdding Triggers | 149 150 151 152 154 155 157 159 160 163 |
| 8.2 8.3 8.4 8.5 8.6 | Credentials 8.1.1How Credentials Work 8.1.2Defining CredentialsCreating JobsAdding Input ParametersAdding StepsHandling Step ErrorsDeclaring the Return Type of a JobHandling Data Types in StepsAdding Triggers 8.8.1Timer Triggers | 149 150 151 152 154 155 157 159 160 163 164 |
| 8.2 8.3 8.4 8.5 8.6 | Credentials 8.1.1How Credentials Work 8.1.2Defining Credentials Creating Jobs Adding Input Parameters Adding Steps Handling Step Errors Declaring the Return Type of a Job Handling Data Types in Steps Adding Triggers 8.8.1Timer Triggers 8.8.2File System Triggers | 149 150 151 152 154 155 157 159 160 163 164 166 |
| 8.2 8.3 8.4 8.5 8.6 8.7 | Credentials 8.1.1How Credentials Work 8.1.2Defining CredentialsCreating JobsAdding Input ParametersAdding StepsHandling Step ErrorsDeclaring the Return Type of a JobHandling Data Types in StepsAdding Triggers 8.8.1Timer Triggers | 149 150 151 152 154 155 157 159 160 163 164 166 168 |
| 8.2 8.3 8.4 8.5 8.6 8.7 | Credentials 8.1.1How Credentials Work 8.1.2Defining Credentials Creating Jobs Adding Input Parameters Adding Steps Handling Step Errors Declaring the Return Type of a Job Handling Data Types in Steps Adding Triggers 8.8.1Timer Triggers 8.8.2File System Triggers 8.8.3HTTP Triggers Exposing Jobs as Web Services | 149 150 151 152 154 155 157 159 160 163 164 166 168 170 |
| 8.2 8.3 8.4 8.5 8.7 8.8 8.9 | Credentials 8.1.1How Credentials Work 8.1.2Defining CredentialsCreating JobsAdding Input ParametersAdding StepsHandling Step ErrorsDeclaring the Return Type of a JobHandling Data Types in StepsAdding TriggersAdding TriggersAdding TriggersAdding TriggersAdding TriggersAdding TriggersAdding TriggersExposing Jobs as Web ServicesCaching Job Results | 149 150 151 152 154 155 157 160 163 164 166 168 170 |
| 8.2 8.3 8.4 8.5 8.6 8.7 8.8 8.9 | Credentials 8.1.1How Credentials Work 8.1.2Defining Credentials Creating Jobs Adding Input Parameters Adding Steps Handling Step Errors Declaring the Return Type of a Job Handling Data Types in Steps Adding Triggers 8.8.1Timer Triggers 8.8.2File System Triggers 8.8.3HTTP Triggers Exposing Jobs as Web Services | 149 150 151 152 154 155 157 159 160 163 164 166 168 170 173 |

| 8.13 | Stopping Jobs | |
|-------|------------------------------------|-----|
| 8.14 | Importing and Exporting Jobs | 180 |
| 9 | Command Line Interfese | 186 |
| _ | Command Line Interface | |
| | help | |
| | assignlicense | |
| | createdb | |
| | debug | |
| | exportresourcestrings | |
| | foreground | |
| 9.7 | initdb | 194 |
| 9.8 | install | 195 |
| 9.9 | licenseserver | 196 |
| 9.10 |)migratedb | 197 |
| 9.11 | repair | 198 |
| 9.12 | setdeflang (sdl) | 199 |
| 9.13 | start | 200 |
| 9.14 | uninstall | 201 |
| 9.15 | upgradedb | 202 |
| 9.16 | verifylicense | |
| 10 | Built-in Functions | 206 |
| 10.1 | /system/abort | 207 |
| 10.2 | /system/compute | 208 |
| 10.3 | /system/compute-string | 209 |
| 10.4 | /system/filesystem | 210 |
| 10.5 | /system/ftp | 212 |
| 10.6 | i/system/mail | 216 |
| | '/system/maintenance | |
| | S/system/shell | |
| | /RaptorXML | |
| 11 | Expression Functions and Operators | 222 |
| | | |
| | Step Result Functions | |
| 11.2. | Stream Functions | 225 |

| 11.3File System Functions | 226 |
|---|---------------------------------|
| 11.4File Path Functions | 227 |
| 11.5List Functions | 229 |
| 11.6String Functions | 231 |
| 11.7Boolean Functions | 234 |
| 11.8Runtime Information Functions | 236 |
| 11.9Operators | 237 |
| | |
| 12 Job Configuration Examples | 240 |
| 12.1Creating a Job from a MapForce Mapping | 241 |
| 12.2Using a Job as Step of Another Job | 246 |
| 12.3Creating a Directory Polling Job | 250 |
| 12.4Adding Error Handling to a Job | 253 |
| 12.5Exposing a Job as a Web Service | 257 |
| 12.6Caching Job Results | 260 |
| 12.7Creating a Job from a StyleVision Transformation | 263 |
| 12.8Using RaptorXML to Validate a Document | 268 |
| 12.9Using RaptorXML to Pass Key/Value Parameter Pairs | 272 |
| | |
| 13 Glossary | 276 |
| 13.1C | 277 |
| 13.2E | 279 |
| 13.3F | 280 |
| 13.4I | 201 |
| 13.5J | 201 |
| 13.3 | |
| 13.6P | 282 |
| | 282 283 |
| 13.6P | 282 283 284 |
| 13.6P | 282 283 284 285 |
| 13.6P | 282 283 284 285 |
| 13.6P | 282 283 284 285 286 |

Index

Chapter 1

Altova FlowForce Server® 3

1 Altova FlowForce Server®

FlowForce Server is a cross-platform software solution used to automate tasks on Windows, Linux, and Mac OS 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
- Command Line Interface
- Built-in Functions
- Expression Functions and Operators
- Job Configuration Examples
- Glossary

Chapter 2

What's New

2 What's New

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
- List parameters 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 <u>step variables</u> 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 09/15/2014 at 12:53 PM

Chapter 3

FlowForce Server Overview

3 FlowForce Server Overview

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

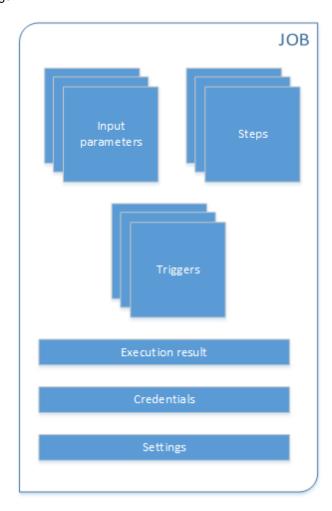
- Basic Concepts
- Security Concepts
- How It Works
- Web Administration Interface

FlowForce Server Overview Basic Concepts 9

3.1 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.



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 <u>Built-in Functions</u>)
- 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 FlowForce Server operating system where the 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.

Settings

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

FlowForce Server Overview Basic Concepts 11

Make the job available as a Web service (see also <u>Exposing Jobs as Web Services</u>)

• 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).

3.2 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: jobs, credentials, functions, and other containers. By setting permissions 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

FlowForce Server Overview Security Concepts 13

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.

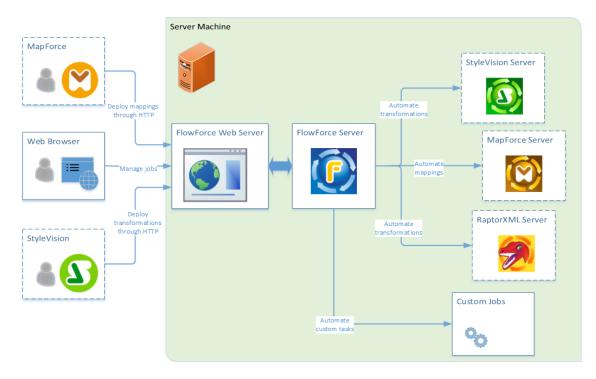
By default, permissions set on a container are inherited from the parent container. For example, let's assume that container A has a child container B. Users who have permission to access container A will have by default permission to access container B as well. However, an administrator can redefine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

3.3 How It Works

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



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.

FlowForce Web Server accepts and validates requests from HTTP 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.

FlowForce Web Server accepts HTTP 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 visual data mapping desktop application by Altova. You can deploy mappings created in MapForce 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. To execute jobs created from deployed MapForce mappings, FlowForce Server calls MapForce Server, whose role is to execute the mappings and produce the resulting output files. For further information, refer |

FlowForce Server Overview How It Works 15

| | to the MapForce and MapForce Server product pages. |
|--|---|
| 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, refer to the StyleVision and StyleVision Server product pages. |

RaptorXML Server

Altova RaptorXML Server (also called RaptorXML for short) is Altova's third-generation, super-fast XML and XBRL processor. It has been built to be 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 and XBRL data.

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 RaptorXML Server Functions.

3.4 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, <u>credentials</u>, 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:

| Date | Specifies the date when the entry was logged. |
|------------|--|
| Severity | Specifies the severity of the logged entry. Valid values: Info Warning Error |
| Module | Spefies 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 as a zipped JSON (JavaScript Object Notation) file, click Export.

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:

Users

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 4

Installation

4 Installation

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

Installation System Requirements 21

4.1 System Requirements

Platform support

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

Note that the Libidn library, available under GNU LGPL, must be installed.

▼ Mac OS X

Mac OS X 10.7 or newer

▼ Windows

Windows XP (SP2 for x64; SP3 for x86), Windows Vista, Windows 7, Windows 8, or newer

▼ 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. The Web administration interface does not support the Secure Sockets Layer (SSL) protocol.

4.2 Linux

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

- Installing FlowForce Server
- Migrating to FlowForce Server 2015
- Viewing Currently Installed Products
- Uninstalling FlowForce Server
- Installation Folder

4.2.1 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-2015-debian.deb
```

Ubuntu

```
sudo dpkg --install flowforceserver-2015-ubuntu.deb
```

CentOS, RedHat

```
sudo rpm -ivh flowforceserver-2015-1.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/MapForceServer2015/etc/*.tool /opt/Altova/
FlowForceServer2015/tools
```

4. On Debian, set the FlowForce Server service to start automatically.

On CentOS, RedHat, and Ubuntu, the FlowForce service is set to start automatically by default. On Debian, use the update-rc.d command to configure the run levels.

Start the FlowForce Web Server service.

Debian

sudo /etc/init.d/flowforcewebserver start

Ubuntu

sudo initctl start flowforcewebserver

CentOS, RedHat

sudo initctl start flowforcewebserver

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/FlowForceServer2015/flowforceweb.html

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

grep running /var/opt/Altova/FlowForceServer2015/data/ffweb.log

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.

- Register FlowForce Server with Altova LicenseServer (see Register FlowForce Server).
- Optionally, configure the network address and port of the Web administration interface (see <u>Setting the Network Address and Port</u>). You may need to do this if the default interface and ports are used by other services on the local machine.
- 8. 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.

4.2.2 Migrating to FlowForce Server 2015

Prerequisites

- FlowForce Server 2015 must be installed (see Installing FlowForce Server).
- Perform data migraton either as root user or as a user with sudo privileges.

To migrate data to FlowForce Server 2015

1. Stop the FlowForce Web Server service.

Debian

sudo /etc/init.d/flowforcewebserver stop

Ubuntu

sudo initctl stop flowforcewebserver

CentOS, RedHat

sudo initctl stop flowforcewebserver

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

Debian, Ubuntu

sudo rm -rf /var/opt/Altova/FlowForceServer2015/data

CentOS, RedHat

sudo rm -rf /var/opt/Altova/FlowForceServer2015/data

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

sudo /opt/Altova/FlowForceServer2015/bin/flowforceserver migratedb
--olddatadir=/var/opt/Altova/FlowForceServer2014/data --datadir=/var/opt/
Altova/FlowForceServer2015/data

4. Start the FlowForce Web Server service.

Debian

sudo /etc/init.d/flowforcewebserver start

Ubuntu

sudo initctl start flowforcewebserver

CentOS, RedHat

sudo initctl start flowforcewebserver

4.2.3 Viewing Currently Installed Products

To check the currently installed Altova server products

• Run the following command:

```
Debian, Ubuntu

dpkg --list | grep Altova

CentOS, RedHat

rpm -qa | grep server
```

Installation Linux 29

4.2.4 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
```

30 Installation Linux

4.2.5 Installation Folder

On Linux, the location of the FlowForce Server application folder is as follows:

/opt/Altova/FlowForceServer2015/bin

4.3 Mac

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

- Installing FlowForce Server
- Migrating to FlowForce Server 2015
- Uninstalling FlowForce Server
- Installation Folder

4.3.1 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.

To install FlowForce Server

- 1. Download the installation package from the Altova website to a local directory.
- 2. In a terminal window, switch to the directory where you have copied the installer file, and double-click it. Go through the successive steps of the installer wizard. These are self-explanatory and include one step in which you have to agree to the license agreement before being able to proceed. The FlowForce Server package will be installed in the folder:

/usr/local/Altova/FlowForceServer2015/

3. 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/MapForceServer2015/etc/*.tool /usr/local/Altova/ FlowForceServer2015/tools

4. Start the FlowForce Server service.

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

- 5. Register FlowForce Server with Altova LicenseServer (see Register FlowForce Server).
- Optionally, configure the network address and port of the Web administration interface (see <u>Setting the Network Address and Port</u>). You may need to do this if the default interface and ports are used by other services on the local machine.
- 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.

4.3.2 Migrating to FlowForce Server 2015

Prerequisites

- FlowForce Server 2015 must be installed (see Installing FlowForce Server).
- Perform data migraton either as a user with administrative (root) privileges.

To migrate data to FlowForce Server 2015

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/FlowForceServer2015/data
```

4. Run the migratedb command:

```
sudo /usr/local/Altova/FlowForceServer2015/bin/FlowForceServer
migratedb
--olddatadir=/var/Altova/FlowForceServer2014/data --datadir=/var/
Altova/FlowForceServer2015/data
```

5. Start the FlowForce Server service.

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

6. Start the FlowForce Web Server service.

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

4.3.3 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 the Applications terminal, 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/FlowForceServer2015/
```

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

4.3.4 Installation Folder

On Mac, the location of the FlowForce Server installation folder is as follows:

/usr/local/Altova/FlowForceServer2015

4.4 Windows

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

- Installing FlowForce Server
- Migrating Data Manually
- Installation Folder

4.4.1 Installing FlowForce Server

Prerequisites

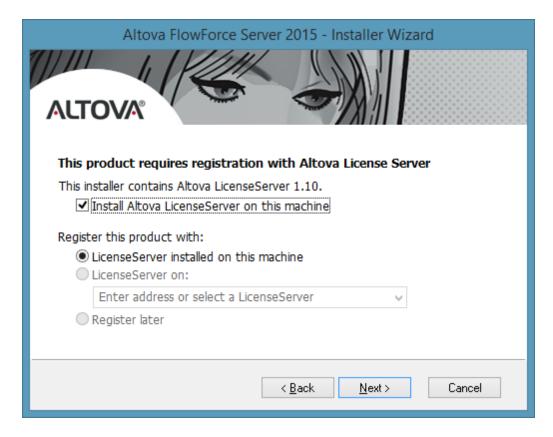
Perform installation as a user with administrative privileges.

To install FlowForce Server

- 1. Download the installation file from the Altova website and run it.
- Optionally, select the installation language from the box in the lower left area of the wizard. 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.



3. Click Next.



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:
 - 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**.
- 5. Click Next.
- 6. Select any additional Altova server products that you want to install.
- 7. Follow the wizard instructions to complete the installation. When the installation completes, the Altova ServiceController icon () is added in the system notification area.

8. If you haven't done so already, register FlowForce Server with Altova LicenseServer (see Register FlowForce Server).

- Optionally, configure the network address and port of the Web administration interface (see <u>Setting the Network Address and Port</u>). You may need to do this if the default interface and ports are used by other services on the local machine.
- 10. 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.

4.4.2 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

- 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 2015 installation wizard. The location of the data folder depends on your Windows version (see How FlowForce Server Stores Configuration Data).
- 4. At the command prompt, run the FlowForce executable with the migratedb comand. Make sure to specify the correct location of the FlowForce Server .exe file and the configuration data folders applicable to your Windows version (see How FlowForce Server Stores Configuration Data). For example, for FlowForce Server 32-bit running on 64-bit Windows 8, the command is:

```
"C:\Program Files(x86)\Altova\FlowForceServer2015\bin \FlowForceServer.exe" migratedb --datadir=C:\ProgramData\Altova\FlowForceServer2015\data -- olddatadir=C:\ProgramData\Altova\FlowForceServer2014\data
```

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

4.4.3 Installation Folder

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

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

Chapter 5

Licensing

5 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>.

5.1 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: 1.11

Licensing process with Altova LicenseServer

To assign an Altova server product a license 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 license/s</u> you have received from Altova to LicenseServer. Do this in the <u>License Pool</u> tab of the Configuration page.
- 4. Register Altova server products (<u>FlowForce Server</u>, <u>MapForce Server</u>, <u>StyleVision Server</u>, RaptorXML(+XBRL) Server) with LicenseServer.
- 5. Assign licenses to Altova server In the Server Management tab of the Configuration page.

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

Note: The <u>LicenseServer Configuration page</u> does not support SSL.

▼ LicenseServer versions and their compatibility with Altova server products

New versions of Altova server products can only be licensed with the version of LicenseServer that is the latest at the time of the server product's release. However, older versions of 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 this older version and install the latest version 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 server machine, and will be imported automatically into the newer version. 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: 1.11

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>.

- <u>How to Assign Licenses</u>, 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: 09-15-2014

5.1.1 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 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

The 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 the 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 the Altova LicenseServer will no longer permit the usage of any Altova software products connected to the 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 the 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.

5.1.2 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, and Altova RaptorXML(+XBRL).)

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

▼ LicenseServer versions and their compatibility with Altova server products

New versions of Altova server products can only be licensed with the version of LicenseServer that is the latest at the time of the server product's release. However, older versions of 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 this older version and install the latest version 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 server machine, and will be imported automatically into the newer version. 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: 1.11

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.

5.1.3 Installation (Linux)

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

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 Altova website 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-1.11-debian.deb
[Ubuntu]: sudo dpkg --install licenseserver-1.11-ubuntu.deb
[CentOS]: sudo rpm -ivh licenseserver-1.11-1.x86_64.rpm
[RedHat]: sudo rpm -ivh licenseserver-1.11-1.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</u> <u>Licenses</u>.

▼ LicenseServer versions and their compatibility with Altova server products

New versions of Altova server products can only be licensed with the version of LicenseServer that is the latest at the time of the server product's release. However, older versions of 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 this older version and install the latest version 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 server machine, and will be imported automatically into the newer version. 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: 1.11

5.1.4 Installation (Mac OS X)

Altova LicenseServer can be installed on Mac OS X systems (version 10.7 or higher). Since you might need to uninstall a previous version, uninstalling is described first.

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 the Applications terminal, 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

Download Altova LicenseServer from the <u>Altova website</u> (the installer file has a .pkg file extension), and double-click the installer package to start the installation. 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

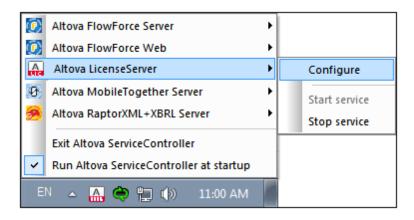
5.1.5 Altova ServiceController

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.

5.1.6 How to Assign Licenses

To assign an Altova server product a license using 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 License Pool tab of the LicenseServer Configuration page.
- 4. Register the Altova server product (<u>FlowForce Server</u>, <u>MapForce Server</u>, <u>StyleVision Server</u>, <u>RaptorXML(+XBRL) Server</u>) with LicenseServer. Depending on the product's type, the method of registering it with LicenseServer will be different: either via the product's Web UI or its command line. See the documentation of your Altova server product for additional information.
- 5. In the <u>Server Management</u> tab of the <u>LicenseServer Configuration page</u>, <u>assign a license</u> to the Altova server product according to the number of cores on the product machine.

Note on cores and licenses

The licensing of Altova server products, <u>except MobileTogether Server</u>***, 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 on a particular server machine must be greater than or equal to the number of cores available on that server, whether it's a physical or virtual machine.

For example, if a server has eight cores (an octa-core processor), you must purchase at least an 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 an 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 license can be used for only one client machine at a time, even if it 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 the license cannot be used simultaneously for another client machine.

MobileTogether Server licenses are assigned on the basis of the number of users, that is, the number of client devices that connect to MobileTogether Server.

Start LicenseServer

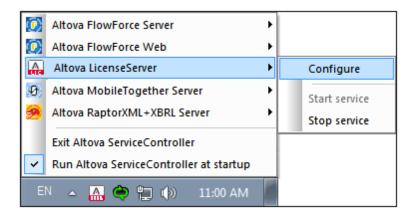
This section:

- 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.

(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

The 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 the 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 the Altova LicenseServer will no longer permit the usage of any Altova software products connected to the 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 the 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)

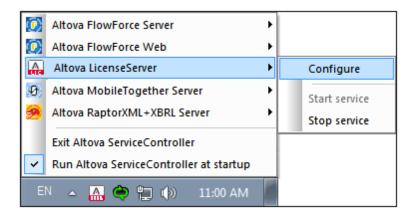
This section:

- 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 <u>Configuration</u> 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 Configuration page from some other Windows machine on the local network (than that on which LicenseServer is installed), enter the URL of the LicenseServer Configuration page 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.

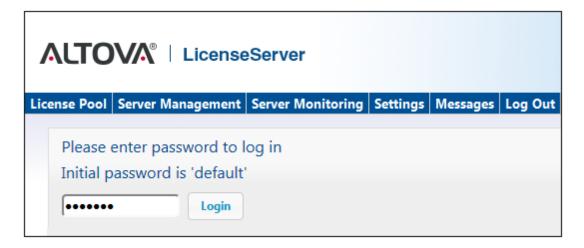
The dynamically generated URL in webur.html will have a form something like:

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://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 default. After you have

logged in, you can change your password in the Settings 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 (Windows)</u> and <u>Open LicenseServer Config Page (Linux)</u>).

| Web UI | | |
|--|--|--|
| Configure the host addresses where the web UI is available to administrators. | | |
| All interfaces and assigned IP addresses | | |
| © Local only (localhost) | | |
| Only the following hostname or IP address: 0.0.0.0 | | |
| Ensure this hostname or IP address exists or LicenseServer will fail to start! | | |
| Configure the port used for the web UI. | | |
| Dynamically chosen by the operating system | | |
| Fixed port 8088 | | |
| Ensure this port is available or LicenseServer will fail to start! | | |
| | | |

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*).

URL of the LicenseServer Configuration page

To open the LicenseServer <u>Configuration page</u> 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.

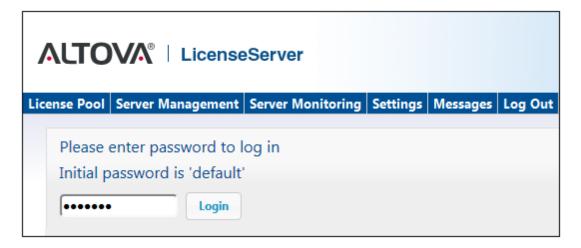
The dynamically generated URL in webul.html will have a form something like:

http://127.0.0.1:55541, 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 default. After you have

logged in, you can change your password in the Settings 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 (Windows)</u> and <u>Open LicenseServer Config Page (Linux)</u>).

| Web UI | | |
|--|--|--|
| Configure the host addresses where the web UI is available to administrators. | | |
| All interfaces and assigned IP addresses | | |
| © Local only (localhost) | | |
| Only the following hostname or IP address: 0.0.0.0 | | |
| Ensure this hostname or IP address exists or LicenseServer will fail to start! | | |
| Configure the port used for the web UI. | | |
| Dynamically chosen by the operating system | | |
| Fixed port 8088 | | |
| Ensure this port is available or LicenseServer will fail to start! | | |
| | | |

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*).

URL of the LicenseServer Configuration page

To open the LicenseServer <u>Configuration page</u> 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.

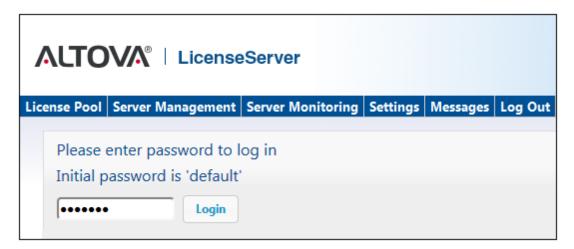
The dynamically generated URL in webui.html will have a form something like:

http://127.0.0.1:55541, 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 <u>Configuration page</u> 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 <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> (Windows) and <u>Open LicenseServer Config Page</u> (Linux)).

| Web UI | | | |
|--------|--|--|--|
| | Configure the host addresses where the web UI is available to administrators. | | |
| | All interfaces and assigned IP addresses | | |
| | Cocal only (localhost) | | |
| | Only the following hostname or IP address: 0.0.0.0 | | |
| | Ensure this hostname or IP address exists or LicenseServer will fail to start! | | |
| | Configure the port used for the web UI. | | |
| | Dynamically chosen by the operating system | | |
| | Fixed port 8088 | | |
| | Ensure this port is available or LicenseServer will fail to start! | | |
| | | | |

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

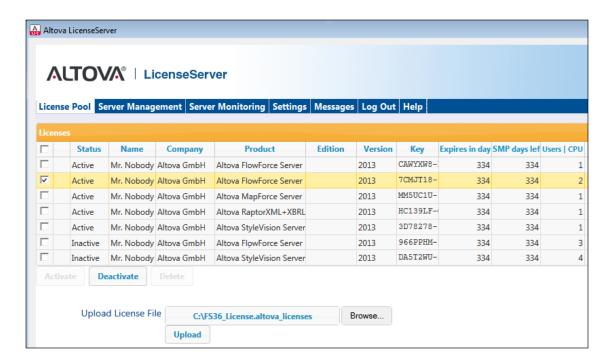
Uploading a license file to the license pool of LicenseServer

After you have obtained a license file from Altova, you must upload it to the Altova LicenseServer. (How to do this is described below.) 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 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 function of the License Pool tab (see screenshot below).



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 below shows multiple licenses, uploaded from multiple license files.



License status

License status values are as follows:

- Activating: When a license is uploaded into 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 in the pool will change to Active.
- Inactive: If a license has been verified, but is present on another LicenseServer on the network, the status in the pool will be shown as Inactive. An Inactive status also results when a license is manually deactivated in the license pool by the administrator.
- 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 Activate button). |
| Active | Verification was successful, the license is active. |
| Inactive | Verification was successful, but the license is on another LicenseServer on the network. Licenses can be made inactive with the Deactivate button. |
| 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 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:

Register the Altova server product (<u>FlowForce Server</u>, <u>MapForce Server</u>, <u>StyleVision 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 server product, you must register the product installation with LicenseServer. The registration is done from the Altova server product, and the process is different for those server products that have Web UIs and those that are run from the command line only. 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 server products:

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

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 2015 | 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 Server 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 licenseserver command of the FlowForce Server CLI. Note that FlowForce Server must be started with root rights.

sudo /opt/Altova/FlowForceServer2015/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/MapForceServer2015/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. Assign a license 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. Only after it has been registered with LicenseServer can a <u>license be assigned</u> to it from LicenseServer. On Windows systems, if MapForce Server was installed as part of the FlowForce Server package, it will automatically be registered when FlowForce is registered. On Linux systems, only if MapForce Server is installed after FlowForce Server will it be registered automatically when

FlowForce Server is registered subsequently.

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.

 $\verb|sudo|/opt/Altova/MapForceServer2015/bin/mapforceserver licenseserver localhost|$

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/MapForceServer2015/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 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 licenseserver command of the StyleVision Server CLI. Note that StyleVision Server must be started with root rights.

sudo /opt/Altova/StyleVisionServer2015/bin/stylevisionserver licenseserver
localhost

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

/opt/Altova/StyleVisionServer2015/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. Assign a license to your Altova product that has been registered with the LicenseServer.

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 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/RaptorXMLServer2015/bin/raptorxmlserver licenseserver
localhost

sudo /opt/Altova/RaptorXMLXBRLServer2015/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/RaptorXMLServer2015/bin
/opt/Altova/RaptorXMLXBRLServer2015/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>.

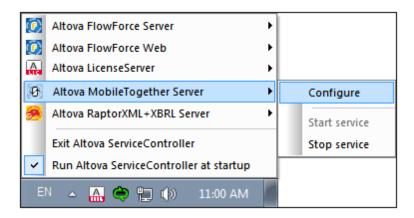
Next Steps

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. <u>Assign a license</u> 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 Setup page; (iii) 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>Server Management tab of LicenseServer's configuration</u> page to assign a license to MobileTogether Server.

Assign Licenses to Registered Products

This section:

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

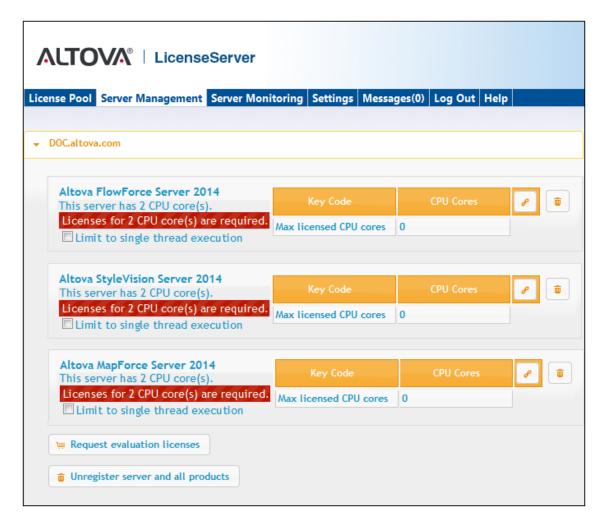
Before assigning a license

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 Server Management tab

Licenses are assigned in the Server Management tab of the LicenseServer Configuration page (*screenshot below*). The screenshot shows that three Altova products have been registered with LicenseServer. (Since MapForce Server and StyleVision Server are bundled with FlowForce Server, registering FlowForce Server with LicenseServer automatically also registers MapForce Server and StyleVision Server. No additional registration of the latter two products are required if FlowForce Server is registered.)



Note the following points about the Server Management tab:

- Each product is listed under the name of its client machine. In the screenshot above, one client machine, named <code>Doc.altova.com</code>, is listed. This client machine (<code>Doc.altova.com</code>) has three Altova products registered with the 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 Server Management tab.
- 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's key code is assigned by clicking
 its Edit Assigned Licenses button (see icon list below) and selecting the required
 license from those available for that product (for example, FlowForce Server) in the
 license pool. This procedure is explained in more detail below.
- Each product also 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 (see screenshot above). (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 Server 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 latest version of LicenseServer 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 server, 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 Server Management tab



Edit Assigned Licenses. Available with each product. Pops up the Manage Licenses dialog, in which new licenses can be assigned to the product and already assigned licenses can be edited.



Show Licenses. Appears with each license. Switches to the License Pool tab and highlights the selected license, so that license details can be read.



Unregister This Product. Available with each product. The selected product (on the selected client machine) will be unregistered from LicenseServer.

Note on cores and licenses

The licensing of Altova server products, <u>except MobileTogether Server</u>***, 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 on a particular server machine must be greater than or equal to the number of cores available on that server, whether it's a physical or virtual machine.

For example, if a server has eight cores (an octa-core processor), you must purchase at least an 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 an 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 license can be used for only one client machine at a time, even if it 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 the license cannot be used simultaneously for another client machine.

*** MobileTogether Server licenses are assigned on the basis of the number of users, that is, the number of client devices that connect to MobileTogether Server.

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 FlowForce Server 2013.
- The dialog displays all the currently active licenses for that product in the license pool. In our screenshot, four currently active FlowForce Server licenses are in the license pool. LicenseServer will automatically detect from each license in the pool the product for which it has been issued.
- The licenses in the screenshot above have been licensed, respectively, for 3 CPU cores, 2 CPU cores, 4 CPU cores, and 4 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. This license could be, for example, the second license in the list shown in the screenshot above. You can also combine licenses. So, if the machine's processor is octa-core (eight-core), you can combine two 4-core licenses; for example, the third and fourth licenses in the list shown in the screenshot above.

- The Manage 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.
- CPU cores indicates for how many CPU cores 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 Server Management tab (see screenshot below). The screenshot below shows that a 2-CPU-core license for Altova FlowForce Server has been assigned (to the client machine Doc.altova.com).



Unregistering products

Each Altova product registered with LicenseServer is listed in the Server Management tab under its client machine name and has an **Unregister** icon to its right. Click this icon to unregister the product. 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 Server and All Products** button at the bottom of the Server Management tab (see first screenshot in this section).

To re-register a product, go to the product's pre-configuration page.

5.1.7 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 (FlowForce Server, MapForce Server, StyleVision Server, RaptorXML(+XBRL) Server). 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) and Open LicenseServer Config Page (Linux).

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

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

For a step-by-step guide of how to assign licenses with LicenseServer, see the section <u>How to Assign Licenses</u>.

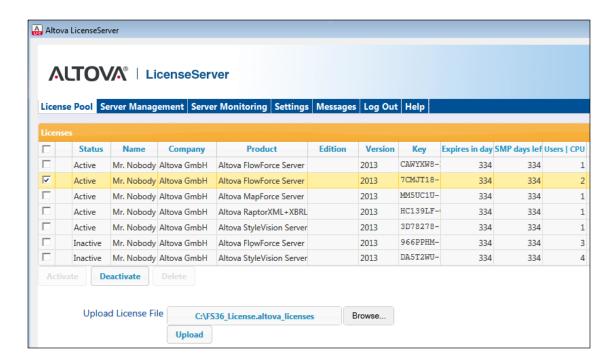
License Pool

This section:

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

The **License Pool** tab displays all the licenses that are currently on the LicenseServer (see screenshot below). When a license file is uploaded to the LicenseServer with the **Upload** button on this page, all the licenses contained in the license file are placed in the license pool on the server and are displayed on the License Pool page.

The License Pool page displays information about all the licenses currently on the LicenseServer and thus provides a convenient overview of all Altova product licenses. On this page you can also activate, deactivate, and delete selected licenses.



Uploading a license

To upload a license file (which you receive from Altova GmbH for your Altova server 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 into 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 in the pool will

change to Active.

• *Inactive:* If a license has been verified, but is present on another LicenseServer on the network, the status in the pool will be shown as *Inactive*. An *Inactive* status also results when a license is manually deactivated in the license pool by the administrator.

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 Activate button). |
| Active | Verification was successful, the license is active. |
| Inactive | Verification was successful, but the license is on another LicenseServer on the network. Licenses can be made inactive with the Deactivate button. |
| 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 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

The 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 the 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 the Altova LicenseServer will no longer permit the usage of any Altova software products connected to the Altova LicenseServer.

Any such loss of connection with the <code>altova.com</code> master servers will be logged in the Messages tab of the Configuration page of the Altova LicenseServer. In addition, the administrator can configure the Altova LicenseServer to automatically send an alert email when the connection to <code>altova.com</code> 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



Edit Assigned Licenses. Appears with each license (in the Assignments column). Pops up the Manage Licenses dialog, in which new licenses can be assigned to the product and already assigned licenses can be edited.



Show License Information. Appears with each license (in the Assignments column). Provides information about the currently active clients.

License information

The following license information is displayed:

- Status: Can be one of the following values: Failed Verification | Activating | Active |
 Inactive | Blocked. See <u>License status</u> 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.
- Key, Expires in days, SMP (days left): The license key to unlock the product, and 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.
- Users | CPU Cores: The number of users or CPU cores that the license allows. In the
 case of Altova's MobileTogether Server product, licenses are 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
 (see note below).

• Assignments: Access to editing dialogs and information of individual licenses.

Note on cores and licenses

The licensing of Altova server products, <u>except MobileTogether Server</u>***, 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 on a particular server machine must be greater than or equal to the number of cores available on that server, whether it's a physical or virtual machine.

For example, if a server has eight cores (an octa-core processor), you must purchase at least an 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 an 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 license can be used for only one client machine at a time, even if it 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 the license cannot be used simultaneously for another client machine.

*** MobileTogether Server licenses are assigned on the basis of the number of users, that is, the number of client devices that connect to MobileTogether Server.

Server Management

This section:

- Icons in the Server Management tab
- Assigning licenses
- One client machine under different names
- Requesting an evaluation license
- Unregistering products

In the **Server Management** tab (*screenshot below*), you can assign licenses to <u>registered products</u>.



Note the following points about the Server Management tab:

- Each product is listed under the name of its client machine. In the screenshot above, one client machine, named <code>Doc.altova.com</code>, has three Altova products registered with the 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 Server Management tab.
- 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's key code is assigned by clicking
 its Edit Assigned Licenses button and selecting the required license from those
 available for that product (for example, FlowForce Server) in the license pool. This
 procedure is explained in more detail below.
- Each product (except MobileTogether Server) also 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 (see screenshot above). (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.)

Single thread execution

If a 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 as 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.

Icons in the Server Management tab



Edit Assigned Licenses. Available with each product. Pops up the Manage Licenses dialog, in which new licenses can be assigned to the product and already assigned licenses can be edited.



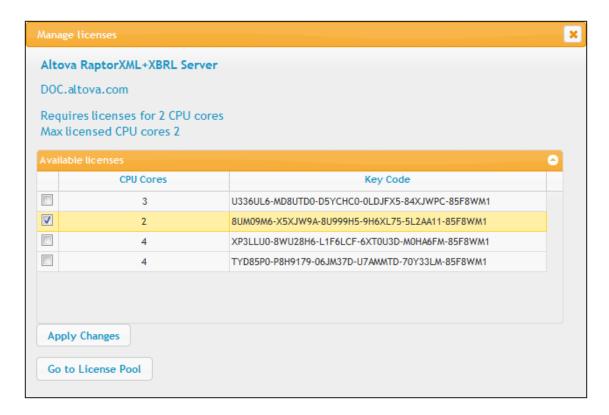
Show Licenses. Appears with each license. Switches to the License Pool tab and highlights the selected license, so that license details can be read.



Unregister This Product. Available with each product. The selected product (on the selected client machine) will be unregistered from LicenseServer.

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*).



Select the license you wish to assign. After selecting the license/s, click **Apply Changes**. The license/s will be assigned to that product and displayed in the Server Management tab (see screenshot below).



One client machine under different names

If a client machine is registered more than once with LicenseServer, it might appear in the Server 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 server and all products** button of these machines. (Note: While the client machines are considered for the purposes of this documentation to be clients of LicenseServer, they are in effect servers of their own products.) 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 Server 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 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 the 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 near the bottom of the Server Management tab. A dialog pops up containing a list of the Altova server 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. The number of cores for which the license will be valid per product will be exactly the number required by the product at the time the request is sent. Save the license/s to disk and upload to the license pool.

Unregistering products

Each Altova product registered with LicenseServer is listed in the Server Management tab under its client machine name and has an **Unregister** icon to its right. Click this icon to unregister the product. 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 Server and All Products** button at the bottom of the Server Management tab (see first screenshot in this section).

To re-register a product with LicenseServer, go to the product's Setup page or its CLI and register it. See: Register FlowForce Server, Register MapForce Server, Register StyleVision Server, and Register RaptorXML(+XBRL) Server.

For more information, see the section, <u>Assigning Licenses to Registered Products</u>.

Server Monitoring

The **Server Monitoring** tab provides an overview of servers currently running licensed Altova products. It contains product information along with information about users and licenses.

Settings

This section:

- Network settings
- Alert Mail settings
- Miscellaneous settings

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

- 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.
- Client statistics
- Network settings for the web-based configuration page (Web UI), the proxy server used to connect to the Internet (if any), for and for LicenseServer (License Service). These settings are described in Network settings 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. A changed setting will not take effect till it is saved.

Network settings

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

| Changing these settings will cause the LicenseServer | to restart and any currently running and licensed applications will be shut down! |
|---|---|
| Configure the host addresses where the web UI is | is available to administrators. |
| All interfaces and assigned IP addresses | |
| Cocal only (localhost) | |
| Only the following hostname or IP address: | 0.0.0 |
| Ensure this hostname or IP address exists or Licens | iseServer will fail to start! |
| Configure the port used for the web UI. | |
| Dynamically chosen by the operating system | |
| Fixed port 8088 | • |
| Ensure this port is available or LicenseServer will f | fail to start! |
| | a proxy server is needed to communicate with Altova's servers. |
| • | a proxy server is needed to communicate with Altova's servers. If the port number is left blank the default port 1080 will be used. |
| Configure the proxy server connection details if a | |
| Configure the proxy server connection details if a Hostname myproxy Port Number 1285 | |
| Configure the proxy server connection details if a Hostname myproxy Port Number 1285 User Name myusername | If the port number is left blank the default port 1080 will be used. |
| Configure the proxy server connection details if a Hostname myproxy Port Number 1285 User Name myusername Password •••••• | If the port number is left blank the default port 1080 will be used. Leave the user name and password blank if no authentication is requ |
| Configure the proxy server connection details if a Hostname myproxy Port Number 1285 User Name myusername Password •••••• | If the port number is left blank the default port 1080 will be used. Leave the user name and password blank if no authentication is requ |
| Configure the proxy server connection details if a Hostname myproxy Port Number 1285 User Name myusername Password •••••• cense Service Configure the host addresses where the LicenseS | If the port number is left blank the default port 1080 will be used. Leave the user name and password blank if no authentication is requ |

- Web UI: Allowed IP addresses can vary from all interfaces and IP addresses on that
 machine to a fixed address, and 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). These fields need to be filled in only if a proxy server is being used. Also, proxy servers often do not need authentication (a user-name and password), in which case these two fields can be left blank. To configure LicenseServer for the proxy server, enter the proxy server's host name, and, if required, a port number.
- License Service: IP addresses can vary from all interfaces and IP addresses on that machine to a fixed address. If you list hostnames and/or IP addresses, use a commaseparated list without any spaces (for example: hostname1, IPAddress1, hostname2). The port number is fixed at 35355.

By default, these 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

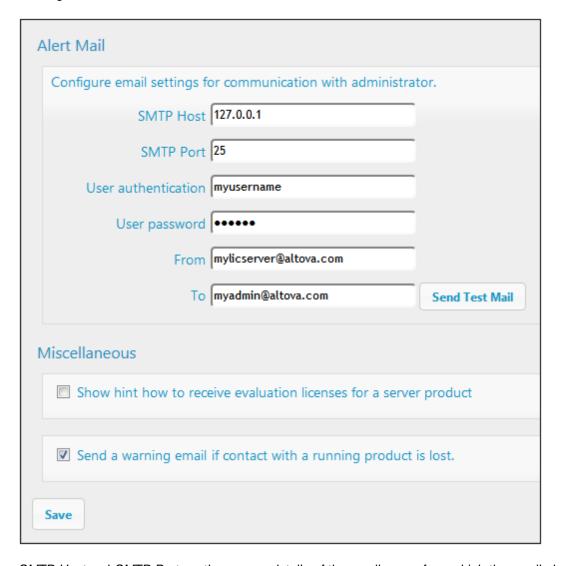
either LicenseServer or its configuration page, enter the appropriate settings and click Save.

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 secreenshot 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.

Chapter 6

Configuring the Server

6 Configuring the Server

This section includes the following FlowForce Server configuration instructions:

- Setting the Network Address and Port
- Setting the Default Time Zone
- Setting the Mail Parameters
- Changing the Windows Active Directory Settings
- Localizing FlowForce Server

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 Command Line Interface). For information about how FlowForce Server stores user or application data, see FlowForce Server Application Data.

In some cases (for example, after editing the .ini files manually), you might need to restart the FlowForce services manually. If you need to restart services manually, refer to the following additional topics:

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

6.1 FlowForce Server Application Data

This topic describes the contents of the FlowForce Server application data folder. 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 folder stores data generated by both FlowForce Server and its users, such as jobs, triggers, system functions, server logs, and other files.

The application data folder 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 location of the application data folder depends on the operating system and platform and is as follows.

| Linux | /var/opt/Altova/FlowForceServer2015 |
|--|---|
| Mac | /var/Altova/FlowForceServer2015 |
| Windows XP | C:\Documents and Settings\All Users\Application Data\Altova\FlowForceServer2015 |
| Windows Vista, Windows 7, Windows 8 | C:\ProgramData\Altova\FlowForceServer2015 |

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

| flowforceserver.ini | Stores global configuration settings of FlowForce Server (currently, the language used in server logs and in error messages). | |
|--|--|--|
| flowforceweb.ini | Stores global configuration settings of FlowForce Web Server (currently, the language of the Web administration interface). | |
| 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/ffweb.log (or data/flowforce.log) | Stores the log of messages from both FlowForce Web Server and FlowForce Server. | |
| data/flowforce.db | This is the database file that stores the FlowForce Server object system, user data, active jobs, roles, and others. | |
| data/flowforcelog.db | This is the database file that stores the FlowForce Server logs. | |
| data/flowforce.ini | This configuration file defines the port and listening interfaces of FlowForce Server. | |
| data/flowforceweb.ini | This configuration file defines the port and listening | |

| | interfaces of FlowForce Web Server. |
|------------|--|
| data/files | Stores files associated with deployed functions. |
| data/logs | Contains captured output from job execution steps. |
| data/tmp | Stores temporary files. |
| data/tools | This folder stores any .tool files copied manually from the installation directory of other Altova server products during FlowForce Server installation (applicable to Linux and Mac). If you installed FlowForce Server before installing any other server products (which is the recommended order), this folder is empty. |

6.2 Opening the Setup Page

The FlowForce Server setup page allows you to specify the default network address and port of the Web administration interface, as well as the network address of LicenseServer. 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, 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/FlowForceServer2015/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 configuration folder (the URL has the format http://127.0.0.1:34597/setup?key=52239315203).

To open the setup page (Mac)

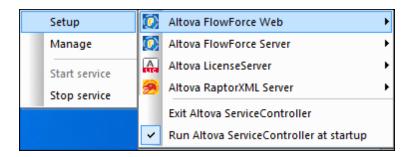
Do one of the following:

- 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).

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 http://127.0.0.1:34597/setup?key=52239315203).

6.3 Setting the Network Address and Port

By default, the FlowForce Server Web administration interface runs at http://
<ServerAddress>:<ServerPort>. If you are accessing the Web administration interface from a browser on the same machine, the default URL is http://localhost:8082. However, you may need to change the default address and port of FlowForce Server or FlowForce Web Server if they are already in use by other services on the same machine.

To configure the network address and port

- 1. Open the FlowForce setup page (see Opening the Setup Page).
- 2. Set the bind address and port for the FlowForce Web Server. By default, the Web administration interface is available to users on all network interfaces on port 8082.



- 3. Optionally, set the default time zone to use in the Web interface.
- 4. Set the bind address and port for the FlowForce Server. The default setting for the server accepts only requests from the same machine (127.0.0.1) on port 4646. 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.
- Click Apply settings and restart FlowForce services. The FlowForce services will restart, and your browser will be redirected to the Login page.

© 2014 Altova GmbH Altova GmbH

6.4 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.

6.5 Setting Mail Parameters

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

To change the mail settings

- 1. Click Administration.
- 2. Click Settings.
- 3. Under **SMTP Server** and **SMTP port**, enter the name and port of the mail server, respectively.
- 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 mail/send function.

6.6 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 Importing Domain Users and Roles). 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 How Privileges Work and How Permissions Work). Administrators can also assign FlowForce Server roles to Windows group or user accounts (see Assigning Roles to Users).
- 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

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

| Α | Active Directory login | | | | |
|-----------------|--|--------------------------------|--|--|--|
| ☑ Enable | | | | | |
| | Allow any domain users to log in | | | | |
| | Suggested domain suffix: my.domain.com | ✓ Set domain suffix as default | | | |
| | Suggested domain suffix: my.domain.com | ✓ Set domain suffix as default | | | |

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 Login . 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. | | |
|----------------------------------|--|--|--|
| Allow any domain users to log in | Select this check box if a user's Windows account should be imported into the FlowForce user database when users log on to FlowForce with their Windows domain credentials. If this option is disabled, Windows users can log on to FlowForce Server only if their Windows account is already imported into FlowForce Server. | | |
| Suggested domain suffix | Enter the suffix of the Windows domain that must be available for selection in the Login drop-down list, so that users can conveniently select a particular Windows domain when logging in. For example, to enable Windows principals from the example.company.com domain suffix to log on (a sample Windows principal would be john.doe@example.company.com), enter the value example.company.com in this field. | | |
| Set domain suffix as default | Select this check box if the domain suffix should be selected as the default choice in the Login drop-down list of the FlowForce Server authentication page. Login name: Password: Log in If you disable the check box, the built-in FlowForce Server authentication ("Directly") is the default choice. | | |

See also

Importing Domain Users and Roles

6.7 Starting and Stopping Services (Linux)

To start the FlowForce Web Server service

Debian

sudo /etc/init.d/flowforcewebserver start

Ubuntu

sudo initctl start flowforcewebserver

CentOS, RedHat

sudo initctl start flowforcewebserver

To stop the FlowForce Web Server service

Debian

sudo /etc/init.d/flowforcewebserver stop

Ubuntu

sudo initctl stop flowforcewebserver

CentOS, RedHat

sudo initctl stop flowforcewebserver

6.8 Starting and Stopping Services (Mac)

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 launchetl unload /Library/LaunchDaemons/
com.altova.FlowForceWebServer.plist

6.9 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).

6.10 Localizing FlowForce Server

FlowForce Server is delivered with support for four languages: English, German, Spanish, and Japanese. So you do not need to create a localized version of these languages. To set any of these four languages as the default language, use FlowForce Server's setdeflang 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>\FlowForceServer2015\bin folder. Your DLL file will have a name of the form FlowForceServer2015_lc.dll. The _lc part of the name contains the language code. For example, in FlowForceServer2015_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.

Chapter 7

Managing User Access

7 Managing User Access

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

7.1 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

7.1.1 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 | Enter the name of the user. The following restrictions apply: It must not be empty It must not begin with or end with spaces The allowed characters are letters, digits, underscore (_), dash (-), and full stop (.) | |
|-------------------------------|--|--|
| 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.

7.1.2 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.

7.1.3 Importing Domain Users and Roles

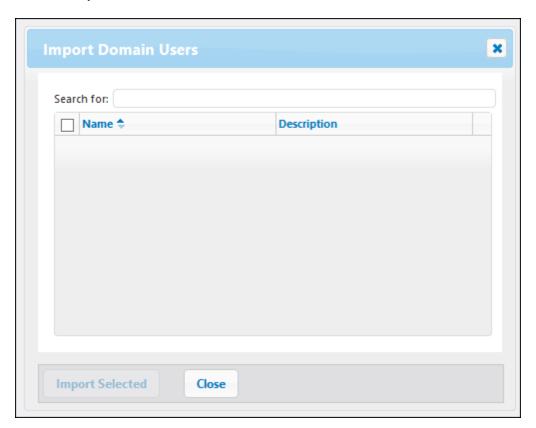
In addition to creating FlowForce Server users, you can import domain user accounts and groups from the Windows Active Directory.

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.

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.

 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

7.1.4 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 Exposing Jobs 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 anonymous 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 anonymous . 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).

7.1.5 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.

7.1.6 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

- 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**.

7.1.7 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

7.2 Privileges

This section includes the following topics:

- How Privileges Work
- Viewing Privilege Reports

7.2.1 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: • Users • Roles • Privileges • Passwords This is an administrative privilege and should only be granted to FlowForce Server administrators. By default, only the user 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 aroot 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 root 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. | |

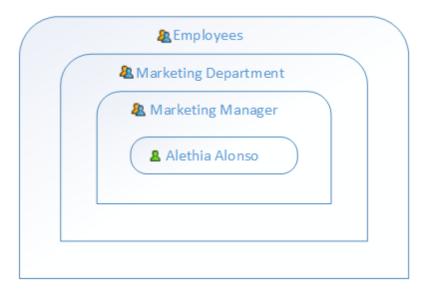
| Stop any job | This privilege grants the right to stop any running FlowForce Server job, regardless of the user who created it. |
|---------------------|--|
| View unfiltered log | By default, users can see log entries related to configurations to which they have "read" access. If granted this privilege, users can read all log entries, including those not associated with a specific configuration. |
| | By default, only the user 4 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 AMARKETING 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

7.2.2 Viewing Privilege Reports

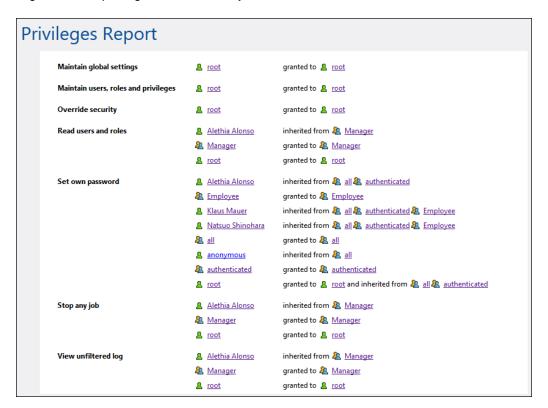
As a FlowForce Server administrator, you might find it difficult to keep track of privileges assigned to each and every role or user, especially when the number of users and roles increases. To help you get a quick overview of all privileges currently assigned to users and roles, FlowForce Server provides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

- Privileges Report
- Privileges by User Report

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

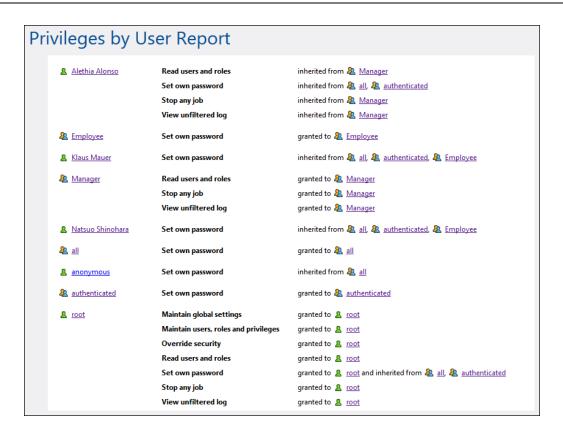
Privileges Report

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 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.



7.3 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

7.3.1 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 permissions define what users can do 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. 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 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 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 W 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. |
| | | Notes 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. If you grant Use permission to user anonymous, the service becomes publicly available and does not require authentication. |
| | 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 | | ission defines whether a user can invoke a ecution 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. Note: 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 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. |

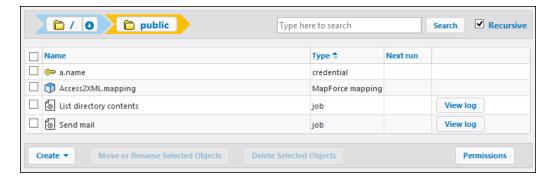
7.3.2 Understanding Containers

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: jobs, credentials, functions, and other containers. By setting permissions 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 /public 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 / public container serves as default location in the following cases: |
|------------|--|
| | When you deploy mappings from MapForce to FlowForce Server. When you deploy transformations from StyleVision to FlowForce Server. |
| | 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 system 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.



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).

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.

7.3.3 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.

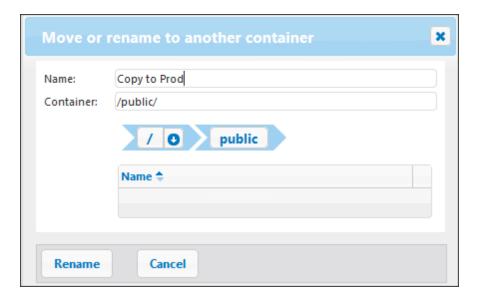


- **4.** Enter the name of the container. The following name restrictions apply:
 - o It must not be empty
 - It must not begin or end with space characters
 - It can contain letters, digits, single space, underscore (_), dash (-), and full stop (.
) characters.
- 5. Click Save.

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.

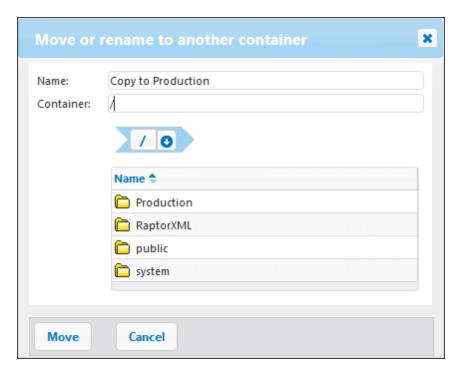
© 2014 Altova GmbH Altova GmbH



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

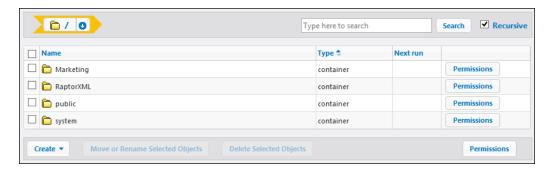
• Click the topmost check box.

7.3.4 Viewing Container Permissions

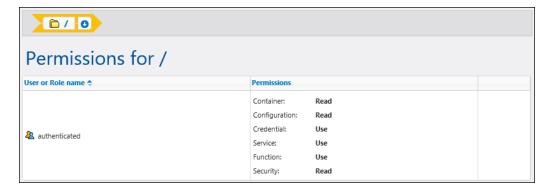
You can view the permissions of containers where you have the relevant permissions to do so (see also How Permissions Work). By default, you can see your own permissions with respect to the container. If you are member of any role, you can also see the permissions available to roles of which you are member. If you have the privilege Read users and roles, 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:
 - 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.

7.3.5 Setting Container Permissions

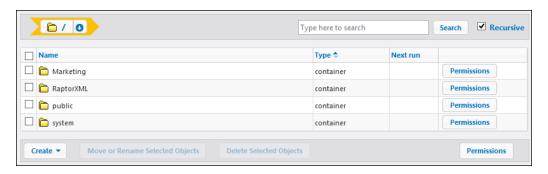
Prerequisites

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

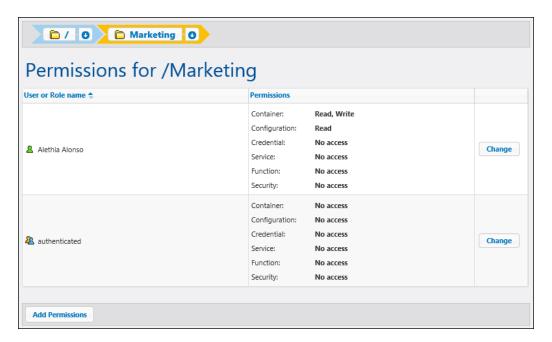
- You (or any roles you are member of) have the Security / Read and Write Security permission on the parent container relative to the one where you want to change permissions. For example, to change the permission of container "Jobs" which is a child of container "Marketing", you must have the permission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

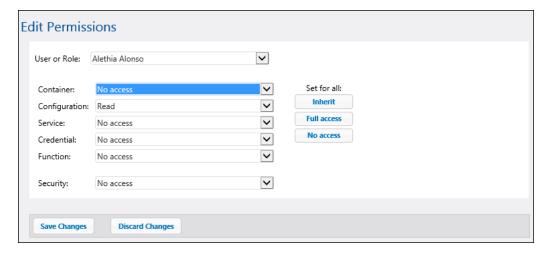
- 1. Click Configuration.
- 2. Do one of the following:
 - o Click the **Permissions** button adjacent to the container record.
 - Click the container record, and then click the click the Permissions button available in the lower right corner of the page.



- 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.



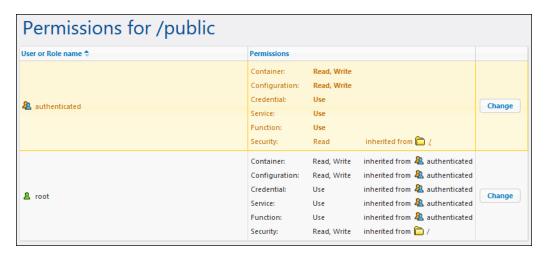
4. In the Edit Permissions section, select from the User and Role list the user or role whose permissions you want to change.



- 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.

7.3.6 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.



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 **A 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).

7.4 Password Policies

This section includes the following topics:

- How Password Policies Work
- Creating and Assigning Password Policies

7.4.1 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.

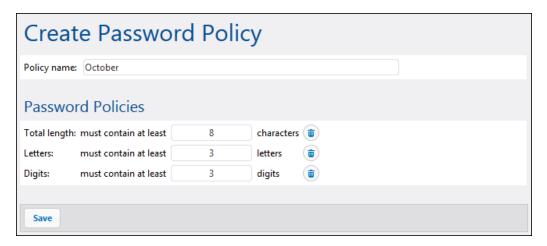
See also

Creating and Assigning Password Policies

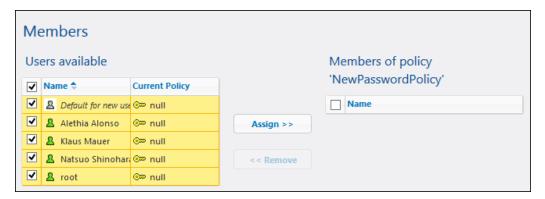
7.4.2 **Creating and Assigning Password Policies**

To create a new password policy

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



- 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 8

Working with Jobs

8 Working with Jobs

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

Working with Jobs Credentials 149

8.1 Credentials

This section includes the following topics:

- How Credentials Work
- Defining Credentials

150 Working with Jobs Credentials

8.1.1 How Credentials Work

A credential record is the combination of user name and password associated with a user account on the FlowForce Server operating system where the 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 151

8.1.2 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.

152 Working with Jobs Creating Jobs

8.2 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 Viewing Container Permissions)
 - 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, get familiar with FlowForce expressions.

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 Adding Input Parameters).
- 5. Under Execution Steps, add the steps of the job (see <u>Adding Steps</u>). 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 Declaring the return type of a job).
- 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>Adding 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**).

Working with Jobs Creating Jobs 153

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 Caching Job Results).
- 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.

8.3 Adding 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>Adding 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 Exposing Jobs as Web Services).

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. | |
|-------------|---|--|
| Type | 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. Describe the purpose of the parameter. | |

To add or remove parameters, click the • and • buttons.

To change the order of parameters, click the the and buttons.

Working with Jobs Adding Steps 155

8.4 Adding 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.

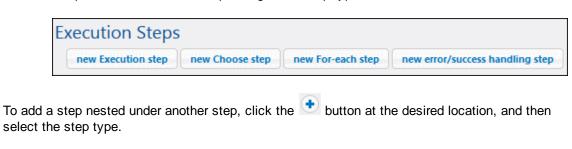
| Execution step | A <u>step</u> of type "Execution step" lets you execute a specific FlowForce function. Available functions include the <u>built-infunctions</u> delivered with FlowForce Server, deployed MapForce mappings or StyleVision transformations, and the execution step of other jobs. | |
|----------------|---|--|
| 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 (subconditions), for example: | |
| | When {expression} When {expression} Execute (step) Otherwise Execute (step) | |
| | Otherwise Execute (step) | |
| | maedite (<u>step</u>) | |
| | 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-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) | |
| | FlowForce executes the step until it finishes looping through all items of the sequence expression. | |

156 Working with Jobs Adding Steps

Error/Success Handling
step

When a step of a job fails, FlowForce Server aborts the job. You can use steps of type "Error/success handling step" to perform clean-up actions before the job finally exits (see Handling Step Errors).

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



To delete a step, click the button.

To change the order of steps, click the and buttons.

Working with Jobs Handling Step Errors 157

8.5 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.



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.
- 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.

158 Working with Jobs Handling Step Errors

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)

8.6 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. | |

8.7 Handling Data Types in Steps

When you pass the result of one step to another step or job, you may 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

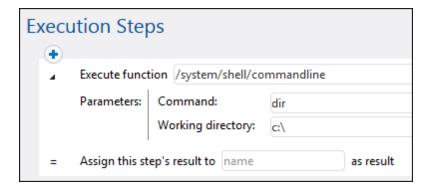
```
stdout(result)
```

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 Expression Functions and Operators, and, more specifically, the Step Result Functions.

Example

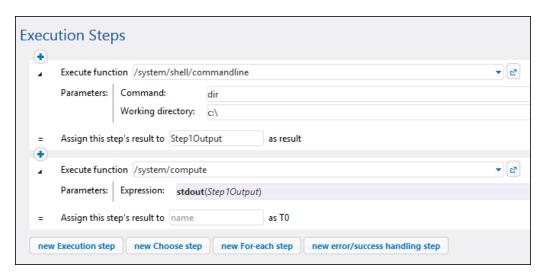
Let's assume that you have created an execution step which lists the contents of a directory, as follows:



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.

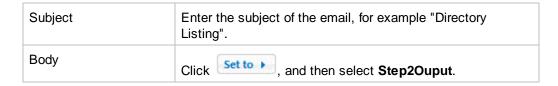


At this stage your job output data type is stream, as intended. You can now create a new execution step that will send the result of the second step to an email address.

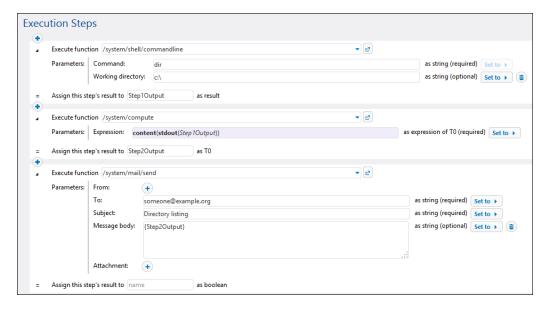
To send the directory listing to an email address

- 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 system/mail/send function):

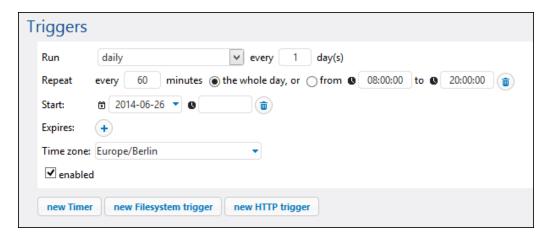
| Execute function | Browse for the <a>This is a second se |
|---|--|
| From Enter the sender's email address. | |
| To Enter the destination email address. | |



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.

8.8 Adding 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.

Adding and deleting triggers

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



To delete a trigger, click the button next to the trigger you want to delete.

See also

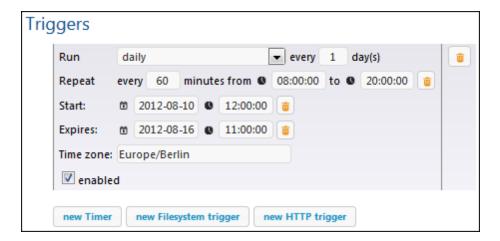
- <u>Timer Triggers</u>
- File System Triggers
- HTTP Triggers

© 2014 Altova GmbH Altova GmbH

8.8.1 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.



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. Defines the date and time when the trigger expires. | |
| 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 enabled check box allows you to enable or disable the trigger. |
|---------|--|
| | This option is useful when creating and testing new jobs. |

8.8.2 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.



File system triggers have the following structure.

| Check | Specifies how the trigger should poll the directory or file. Va options: | |
|--------------------------|---|---|
| | Modified date | The trigger checks the last modification timestan files. If any dates have changed, or a new file has trigger fires. |
| | Content | This option computes and stores a hash code for After the polling interval has passed, the hash co and compared to the stored value. If there is a diffires. Note that this can place considerable load of any dates have changed, or a new file has been 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 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 | The <i>enabled</i> check box allows you to enable or disable the trigger. This option is useful 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 <u>file path expression</u> functions.

For an example of a job that uses the triggerfile parameter, see Creating a Directory Polling Job.

8.8.3 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.



HTTP triggers have the following structure.

| Check | Specifies how the trigger should poll the URI. Valid options: | | |
|---------------------------|---|---|--|
| | Modified date | Instructs the system to check the Last-Modifie the Last-Modified HTTP header is missing, the header is checked (see next option). | |
| | Content | Instructs the system to check the Content-MD5 header. This is a 128-bit "digest" used as a mess If the MD5 header has changed after the polling if then the trigger fires. If the header is not provided 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 | The <i>enabled</i> check box allows you to enable or disable the trigger. This option is useful 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.

8.9 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.



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 http://localhost:4646 (assuming you are accessing it from the same machine). The server name and port are as defined in the administration page (see Setting the Network Address and Port).

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 9 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)

Working with Jobs Caching Job Results 173

8.10 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.

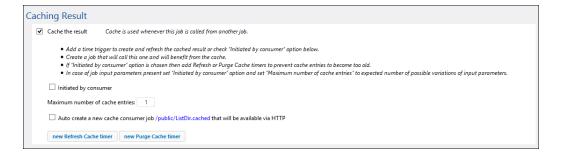
Caching the result of a job essentially means that FlowForce Server prepares and stores the job result in some internal repository (that is, the cache). If the job has parameters, the system creates a cache entry for every parameter combination, up to a limit that can be configured. When the job with 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:

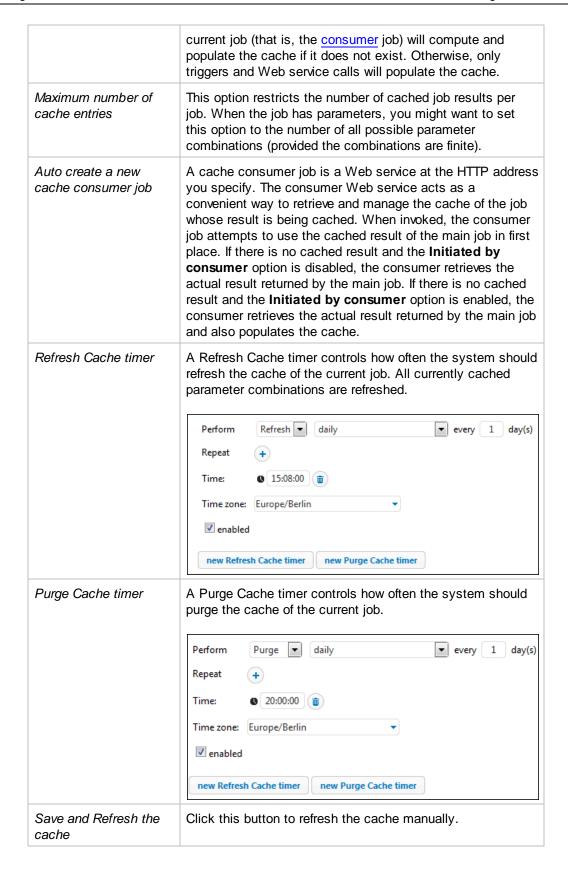
- It is mandatory to declare the data type returned by the job whose result is cached (see Declaring the return type of a job).
- Both the job whose result is cached and the consumer job must use the same credentials. If the credentials differ, then the job executes as if no cache were defined.
- When you change the configuration of the cached job, the existing cache data is invalidated (flushed).

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 |

174 Working with Jobs Caching Job Results



Working with Jobs Caching Job Results 175

See also

<u>Caching Job Results</u> (example)

8.11 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.



| 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 | Enter the number of times the same job may be executed in parallel on the server. |

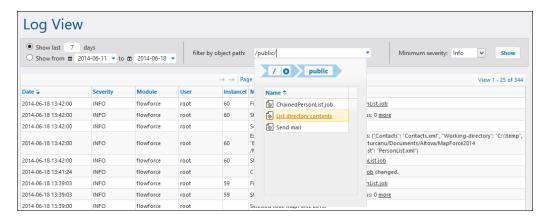
Working with Jobs Viewing the Job Log 177

8.12 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".)
- 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".)
- Click Show.

178 Working with Jobs Stopping Jobs

8.13 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

Working with Jobs Stopping Jobs 179

job.

8.14 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 single .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.

Before exporting objects, FlowForce displays the list of objects for your review, on a separate page. In addition to objects that are to be exported, this page displays also any objects on which the exported jobs depend, so that you can include them if they were accidentally missed. If you are exporting objects to a running FlowForce Server, additional information is available on whether each object already exists on the destination server. In this way, you can decide individually for each job whether it should be included or excluded. Note, however, that if you choose to export jobs without exporting all their dependent objects, your jobs might fail if dependencies do not exist at destination.

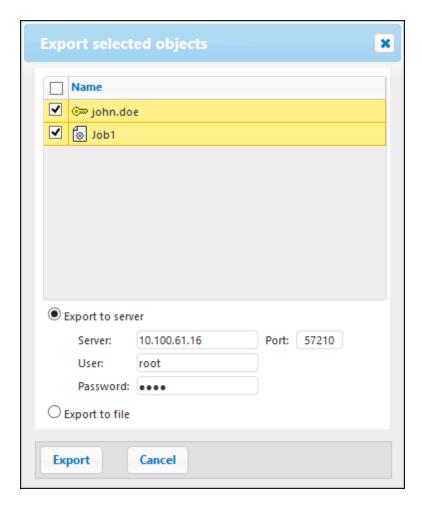
For security reasons, FlowForce Server does not export the following categories of sensitive data:

- 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.

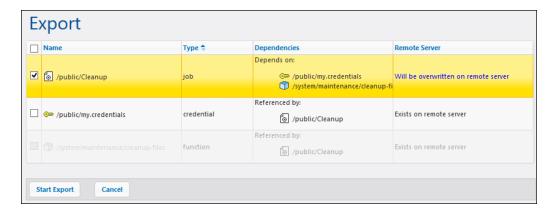
To export jobs to a running FlowForce Server instance

- 1. Click **Configuration**, and open the container which stores the records you want to export.
- 2. Click to select the records you want to export, and then click **Export Selected Objects**.



- 3. In the pop-up dialog box, 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**.

The system displays all records to be exported on a page where you can choose whether to include or exclude the objects that are to be exported. Any detected job dependencies are also displayed, as well as information on whether the file exists at destination, as shown below.



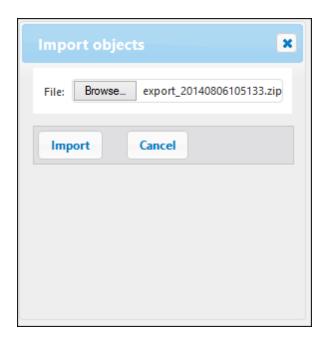
5. Click to select the objects to be exported, and then click **Start Export**.

To export jobs to a file

- Click Configuration, and navigate to the container which stores the records you want to export.
- 2. Select the check box adjacent to the records you want to export, and then click **Export Selected Objects**.
- 3. In the pop-up dialog box, click **Export to file**.
- 4. Browse for a location on the local drive, and then click **Export**.

To import jobs from a file

1. Click **Configuration**, and then click **Import Objects**. A pop-up dialog box appears.



2. Browse for the location of the archive on your local drive, and then click **Import**.

Chapter 9

Command Line Interface

9 Command Line Interface

This section describes the command line interface of FlowForce Server. The command line interface includes the following commands:

| help | Displays help for the command supplied as argument. |
|-----------------------|--|
| assignlicense | This command is applicable to Windows platforms only. It can be used to upload and assign a license file to FlowForce Server. |
| 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. |
| install | 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. |
| setdeflang sdl | Sets the default language. |
| start | Starts the application as a service. |
| uninstall | Uninstalls the application as a Windows service. |
| upgradedb | Upgrades the FlowForce Server database to the latest version. |
| <u>verifylicense</u> | 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. |

Location of the FlowForce Server executable

| Linux | On Linux systems, the FlowForce Server executable (flowforceserver) is located by default at: |
|-------|---|
| | /opt/Altova/FlowForceServer2015/bin/ flowforceserver |
| | where the first forward slash indicates the root directory. |

Command Line Interface 187

| Mac | On Mac systems, the FlowForce Server executable (flowforceserver) is located by default at: /usr/local/Altova/FlowForceServer2015/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

General command line syntax for FlowForceServer is as follows:

FlowForceServer --h | --help | --version | <command> [options] [arguments]

The options and arguments in square brackets are optional.

where

| h help | Displays the help text. |
|---------|---|
| version | Displays the version of FlowForce Server. |

188 Command Line Interface help

9.1 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 189

9.2 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. |
|---|
|---|

Options

| 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. |

190 Command Line Interface createdb

9.3 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.

Options

| datadir=VALUE | VALUE is the path of the data directory. | |
|----------------|--|--|
| datadii viiioi | VALUE IS the path of the data directory. | |

Command Line Interface debug 191

9.4 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.

Options

| datadir=VALUE | VALUE is the path of the data directory. | |
|----------------|--|--|
| dataarr viiior | VALUE IS the path of the data directory. | |

192 Command Line Interface exportresourcestrings

9.5 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

Command Line Interface foreground 193

9.6 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.

194 Command Line Interface initdb

9.7 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.

Options

| datadir=VALUE | VALUE is the path of the database directory. | |
|---------------|--|--|
| | | |

Command Line Interface install 195

9.8 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.

196 Command Line Interface licenseserver

9.9 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> <u>documentation</u>.

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-parsable JSON object. |
|---|------|--|
| | | Form:json=true/false |

Command Line Interface migratedb 197

9.10 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{lowForceServer2015} $$ \end{align*} $$ "C:\Program Files(x86)\Altova\FlowForceServer2015\bin\FlowForceServer.exe" $$ migratedb$

 $--datadir=C:\ProgramData\Altova\FlowForceServer2015\data --olddatadir=C:\ProgramData\Altova\FlowForceServer2014\data$

198 Command Line Interface repair

9.11 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. | |
|---------------|--|--|
| | ļ , , , , , , , , , , , , , , , , , , , | |

Example

 $\label{lowForceServer} FlowForceServer a constraint of the const$

Command Line Interface setdeflang (sdl) 199

9.12 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 |
|----|----------|
| de | German |
| es | Spanish |
| ja | Japanese |

Example

FlowForceServer setdeflang de

200 Command Line Interface start

9.13 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.

Options

| datadir=VALUE | VALUE is the path of the database directory. | |
|---------------|--|---|
| uataurr-value | VALUE IS the path of the database directory. | П |

Command Line Interface uninstall 201

9.14 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.

202 Command Line Interface upgradedb

9.15 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. | datadir=VALUE value is the | path of the database directory. |
|--|----------------------------|---------------------------------|
|--|----------------------------|---------------------------------|

Example

Command Line Interface verifylicense 203

9.16 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]

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 10

Built-in Functions

10 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

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

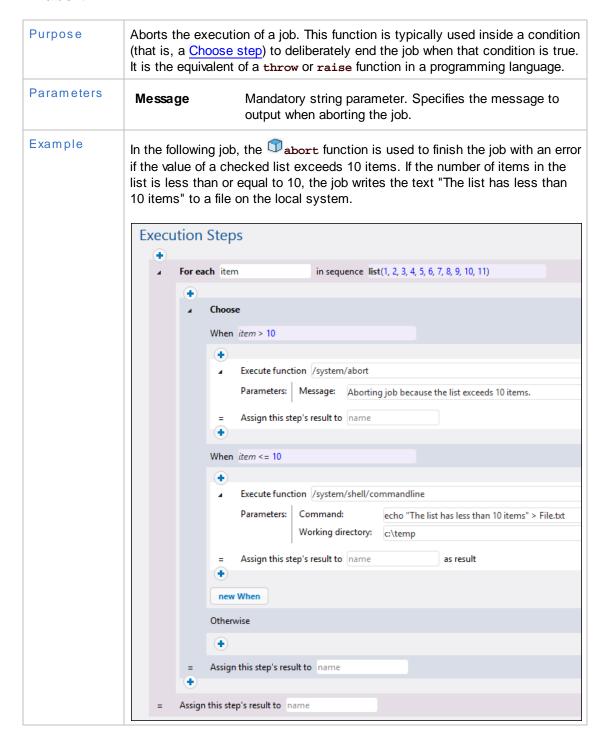
Where:

- 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 207

10.1 /system/abort

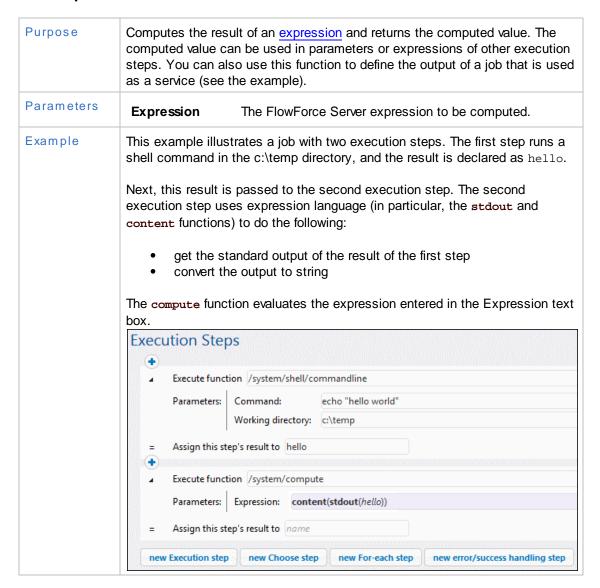
abort



208 Built-in Functions /system/compute

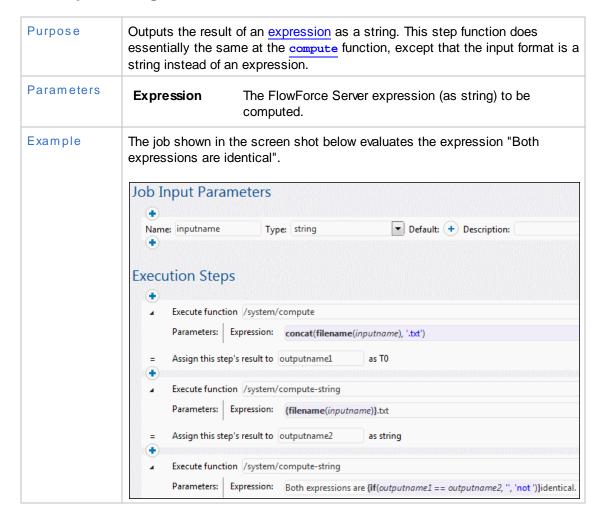
10.2 /system/compute

compute



10.3 /system/compute-string

compute-string



210 Built-in Functions /system/filesystem

10.4 /system/filesystem

The **system/filesystem** container includes functions used to manage files and directories on the operating system where FlowForce Server runs.

Note: All file paths in job execution steps are paths on the operating system where FlowForce Server runs, not on your local machine.

🗇 сору

| Purpose | Copy file(s). | |
|------------|-----------------------|---|
| Parameters | Source | The path and file name of the source file that you want to copy. |
| | Destination | 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 | Clicking generates a check box. Activation causes destination file to be overwritten. |
| | Working- directory | Specifies the working directory (for example, c:\temp). |

delete

| Purpose | Delete file(s). | |
|------------|-----------------------|---|
| Parameters | Path | The path and file name of the file you want to delete. |
| | Working- directory | Specifies the working directory (for example, c:\temp). |

mkdir

| Purpose | Create directory. | |
|------------|-----------------------|--|
| 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. |
| | Working- directory | Specifies the working directory (for example, c:\temp). |
| Example | If Working-Direc | tory is c:\temp, and Path is temp2\temp3, the function |

Built-in Functions /system/filesystem 211

| creates the new directory c:\temp\temp2\temp3. |
|---|
| creates the new directory c: \temp\temp2\temp3. |

move

| Purpose | Move or rename file(s). | |
|------------|-------------------------|--|
| 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 | Clicking generates a check box. Activation causes destination file to be overwritten. |
| | Working- directory | Specifies the working directory (for example, c:\temp). |

rmdir

| Purpose | Remove directory. | |
|------------|-----------------------|---|
| Parameters | Path | The name of the directory you want to delete. |
| | Working- directory | Specifies the working directory (for example, c:\temp). |

10.5 /system/ftp

The $\mbox{system/ftp}$ $\mbox{container}$ includes functions used to perform FTP operations on a remote FTP server.

delete

| Purpose | Delete a file from the FTP server. | |
|------------|------------------------------------|--|
| 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. |
| | Account | The FTP account name of the user allowed access to the files on the remote server. |

mkdir

| Purpose | Create a directory on the FTP server. | |
|------------|---------------------------------------|--|
| 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). |

| | arget lirectory | The name of the directory that you want create on the server. |
|---|--------------------|--|
| A | 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. | |
|------------|---------------------------------|--|
| 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. |
| | Account | The FTP account name of the user allowed access to the files on the remote server. |

retrieve

| Purpose | Retrieves a file from the FTP server. | |
|------------|---------------------------------------|---|
| 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. |
| 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. |
| | |

Tmdir

| Purpose | Deletes a directory from the FTP server. | |
|------------|--|--|
| 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. |
| | Account | The FTP account name of the user allowed access to the files on the remote server. |

store

| Purpose |
|---------|
|---------|

| 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. |
| | 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. |

216 Built-in Functions /system/mail

10.6 /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

| Purpose | Sends a mail to the specified recipients, generally the administrator. | |
|------------|--|--|
| 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. |
| | Attachment | File name of the attachment sent with the email. |

Built-in Functions /system/maintenance 217

10.7 /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. | |
|------------|--|---|
| Parameters | Older than, days | Archives files older than the number of days entered here. Default=30 |
| | Archive directory | Archive directory name, (for example, $c:\neq emp$). 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). |

cleanup-files

| Purpose | Deletes those files that are not in use or referenced by any deployed objects (such as MapForce mappings and StyleVision transformations). 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 | |
|---------|---|--|
| | records that were deleted. | |

218 Built-in Functions /system/maintenance

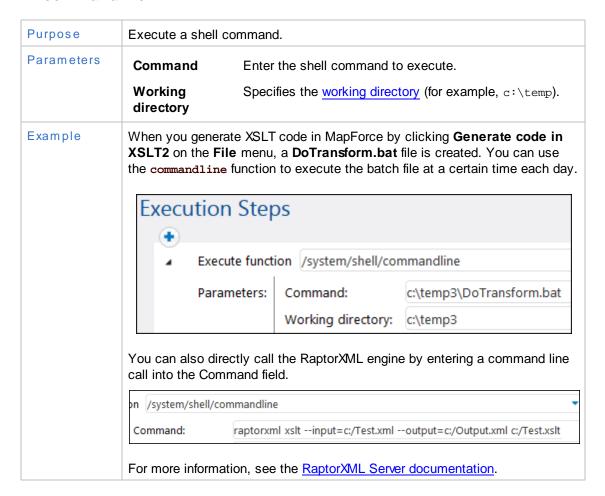
| Parameters Olde days | than, Truncates (deletes) records older than the num days entered here. Default=30 | ber of |
|----------------------|--|--------|
|----------------------|--|--------|

Built-in Functions /system/shell 219

10.8 /system/shell

The **system/shell** container includes the function commandline, used to execute shell commands.

commandline



220 Built-in Functions /RaptorXML

10.9 /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.

Altova RaptorXML Server (also called RaptorXML for short) is Altova's third-generation, super-fast XML and XBRL processor. It has been built to be 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 and XBRL data.

The functions applicable to RaptorXML Server are as follows:

- valany
- valdtd
- valxml-withdtd
- valxml-withxsd
- valxquery
- valxsd
- valxslt
- wfany
- wfdtd
- wfxml
- xquerv
- xslt 🗇

The list of available functions correspond to the product flavour you have licensed (either RaptorXML or RaptorXML+XBRL).

For examples of using RaptorXML Server functions in FlowForce Server jobs, see <u>Job</u> <u>Configuration Examples</u>.

For more information, see the RaptorXML Server documentation.

Chapter 11

Expression Functions and Operators

11 Expression Functions and Operators

FlowForce uses a simple expression language similar to that of Microsoft Excel, to compute values, create streams, create/disassemble lists, etc. 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
- Operators

11.1 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 handling errors 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 stdout or stderr , for example: |
|------------|---|
| | <pre>stderr(failed-step()) stdout(failed-step())</pre> |
| Signature | failed-step() of type result |
| Parameters | None |
| Example | See Adding Error Handling to a Job for an example. |

results

| Purpose | Returns an array of streams of the specified result, optionally filtered by name. Use the function | |
|---------|--|--|
|---------|--|--|

| Example | If a MapForce component CompletePO produces a result consisting of several streams, and if you intend to access the first one, use: |
|---------|---|
| | {as-file(nth(results(MapForceMapping, "CompletePO"), 0))} |
| | The results function picks the array of streams from the MapForce component. The nth function picks the first item from this array. Finally, the as-file function creates 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 result . With the stdout function, you can get access to the standard output of result , as follows: |
| | stdout(result) |
| | where result is the value returned by some execution step. |
| | This function fails if result 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 Adding Error Handling to a Job for an example. |

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 Error Handling to a Job for an example. | |

11.2 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 |

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 | stream-from-string (string, encoding = "UTF-8", contenttype="text/plain") of type stream |

stream-open

| Purpose | Creates a stream from an existing file. |
|-----------|---|
| Signature | <pre>stream-open(name, contenttype="application/octet-stream") of type stream</pre> |
| Example | <pre>stream-open("C:\files\textfile.txt")</pre> |

11.3 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") of type "list of string" |

11.4 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: |
| | extension("c:\temp\file.txt") |
| | returns ".txt". |

filename

| Purpose | Extracts the file name from a path. |
|-----------|-------------------------------------|
| Signature | filename(path) of type string |
| Example | The following expression: |
| | filename("c:\temp\file.txt") |
| | 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: |
| | <pre>filename-with-extension("c:\temp\file.txt")</pre> |
| | returns "file.txt". |

parent-directory

| Purpose | Extracts the parent directory from a path. |
|-----------|---|
| Signature | parent-directory(path) of type string |
| Example | The following expression: |
| | <pre>parent-directory("c:\temp\file.txt")</pre> |
| | returns "c:\temp". |

11.5 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 |
|---------|--|
| | |

| | 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 |
| | The following expression |
| | slice(list(1,2,3,4),1,3) |
| | returns list(2,3). |

11.6 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 string argument ends with the string supplied in the end 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 num | ber |

split

| Purpose | Splits the string at each occurrence of separator. |
|---------|--|
|---------|--|

| Signature | split(string, separator) | of type list of string | |
|-----------|--------------------------|------------------------|--|

starts-with

| Purpose | Returns TRUE if the string supplied in the string argument starts with the string supplied in the start argument. | |
|-----------|---|--|
| Signature | starts-with(string, start) of type boolean | |

string

| Purpose | Computes the string representation of the given number, i.e. conver 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 speficied 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). | |
|-----------|---|----------------|
| 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 | |

11.7 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 |

true

| Purpose | Returns Boolean TRUE. | |
|-----------|-----------------------|-------|
| Signature | true() of type bo | olean |

11.8 Runtime Information Functions

The runtime information functions can be used to handle the details of the currently running jobs.

instance-id

| Purpose | | for every job execution. This can be used to create a job execution, where the string is used to define the |
|-----------|---------------|---|
| 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 |

11.9 Operators

Basic mathematical operators can be used to work on strings and numbers.

```
checks if a and b are equal (numerically equal for numbers, code-point equal for strings)

a != b
is equivalent to not (a == b). a <> b is another variation.

a < b
checks if a is less than b (numerically less for numbers, see below for strings)

a >= b
is equivalent to not (a < b)

a > b is equivalent to b < a

a <= b is equivalent to b >= a.
```

String comparisons are performed as follows:

- The common prefix of the two strings are ignored (evaluated on codepoints)
- If both remaining strings are non-empty, their first codepoints are compared numerically
- Empty strings are less than non-empty strings

Operators on numbers:

a + b, a - b, a * b, a / b

All compute the normal arithmetic results.

Chapter 12

Job Configuration Examples

12 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
- Using RaptorXML to Pass Key/Value Parameter Pairs

12.1 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

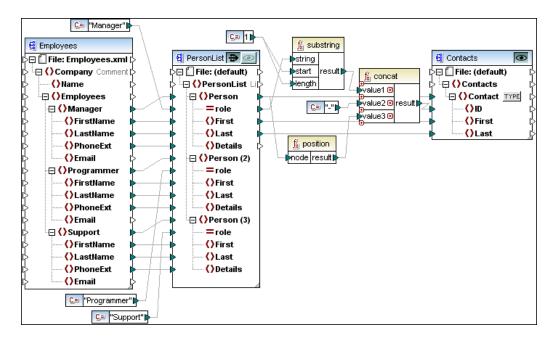
- Required licenses: MapForce Enterprise or Professional edition, MapForce Server, FlowForce Server
- FlowForce Server is running at the configured network address and port (see <u>Setting the</u> Network Address and Port)
- You have a FlowForce Server user account with permissions to one of the <u>containers</u> (by default, the /public container used in this example is accessible to any authenticated user).
- The mapping job created in this example generates two XML files 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

ChainedPersonList.mfd, available at the following path relative to the MapForce examples project: MapForceExamples > Mapping Folder > XML Schemas > Chained Mapping. (To open the MapForce examples project in MapForce, click MapForceExamples.mfp on the Project menu.)

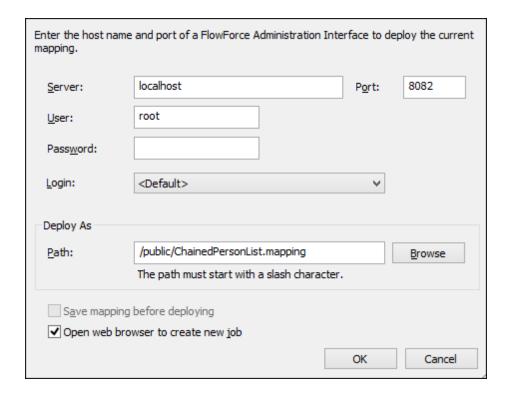
To create a job from a MapForce mapping

Open the ChainedPersonList.mfd file in MapForce.



- If you haven't done so already, set the transformation language of the mapping to "Builtin".
- 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 Directory Settings</u>), enter your Windows user name and password, and then select your Windows domain name from the *Login* drop-down list.



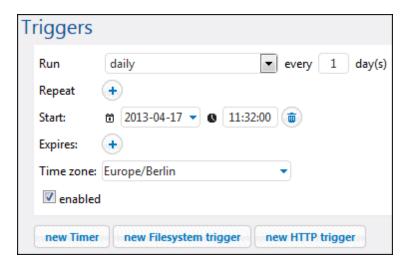
- 6. Make sure the **Open web browser to create new job** check box is selected. Although you can change the path where the mapping should be deployed, for the purpose of this example, leave the selected path as is.
- 7. Click **OK**. When deployment finishes, the FlowForce Server Administration Interface 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.

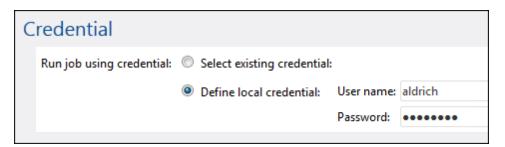
A working directory is a parameter required by <u>execution steps</u> if the job needs a location to unpack any input files or save output files. FlowForce Server also uses the working directory to resolve any relative paths that occur during step execution. When asked to provide a working directory, 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 <u>Credentials</u>).

A credential record is the combination of user name and password associated with a user account on the FlowForce Server operating system where the 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 two files generated as a result (Contacts.xml and PersonList.xml) become available in the working directory specified previously. To see whether the job executed successfully, refer to the job log (see Viewing the Job Log).

12.2 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 Creating a Job from a MapForce Mapping example before completing this example.

As you may recall from the <u>Creating a Job from a MapForce Mapping</u> example, the **ChainedPersonList.job** generates two XML files 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 files from the temporary folder to an archive folder (we will call this job **copy2archive**).
- 2. Modify the **ChainedPersonList.job** to include the **copy2archive** job as an additional execution step.

Prerequisites

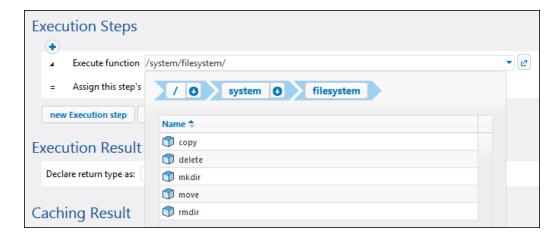
- Required licenses: MapForce Enterprise or Professional edition, MapForce Server, and FlowForce Server
- FlowForce Server is running at the configured network address and port (see <u>Setting the</u> Network Address and Port)
- You have a FlowForce Server user account with permissions to one of the <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

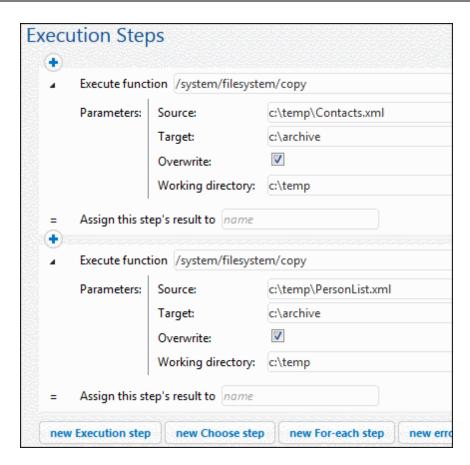
 ChainedPersonList.job available in the /public container of FlowForce Server administration page.

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 **Built-in Functions**).



- b. In the Source text box, type the path of the file to be copied (for example, "C:\temp \Contacts.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 to the path, for example "C:\archive\Contacts2.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. In the Working-directory box, type the working directory (for example, "C:\temp").
- 5. Add the second execution step, as follows:
 - a. Next to Execute function, navigate to the system/filesystem/copy function.
 - b. In the Source text box, type the path of the file to be copied (for example, "C:\temp \PersonList.xml").
 - c. In the Target text box, type the destination path (for example, "C:\archive").
 - d. Click to select the Overwrite check box.
 - e. In the Working-directory box, type the working directory (for example, "C:\temp").

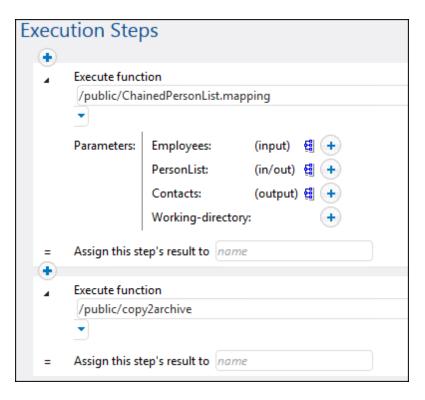


- 6. Under credentials, select an existing credential record or specify a local credential (see Credentials).
- 7. Click Save.

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 "ChainedPersonList" job

- 1. In the /public container, click to open the ChainedPersonList.job record.
- 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**.
- At the time entered in the trigger, FlowForce Server executes the job and copies the Contacts.xml and PersonList.xml files from the temporary location to the archive location. To see whether the job executed successfully, refer to the job log (see <u>Viewing</u> the Job Log).

12.3 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

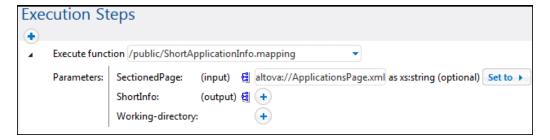
- Required licenses: MapForce Enterprise or Professional edition, MapForce Server, and FlowForce Server
- FlowForce Server is running at the configured network address and port (see <u>Setting the</u> Network Address and Port)
- You have a FlowForce Server user account with permissions to one of the <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).

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 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.



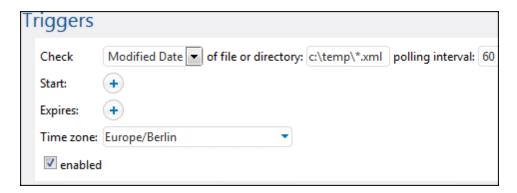
3. In the **Working-directory** box, enter the path to the working directory applicable to this step. This example uses C:\temp as working directory.

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

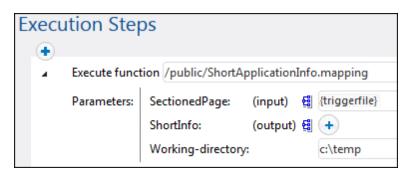


6. Under Execution Steps, pass the {triggerfile} parameter as input value to the

SectionedPage parameter. To do this, click the

SectionedPage parameter, and then select {triggerfile}. As a result, the value of the

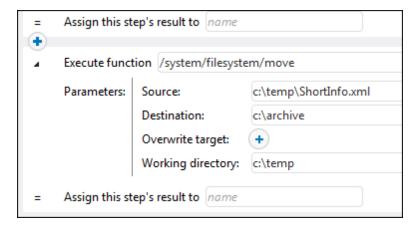
SectionedPage parameter changes to {triggerfile}.



7. Under Execution Steps, add a new execution step. This step will move the mapping

output (**ShortInfo.xmI**) 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:\archiveWorking directory: c:\temp



- 8. Under Credentials, select an existing credential record or specify a local credential (see <u>Credentials</u>).
- 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).

12.4 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 Step Result Functions).
- 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 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 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:

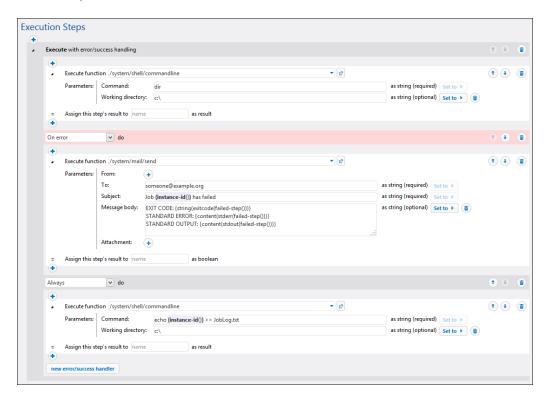
| Execute function | Browse for the /system/mail/send function. |
|------------------|--|
| From | Enter the email address of the sender. Leave this field empty if you have configured the mail settings from the administration page. |
| То | Enter your email address. |
| Subject | Enter the subject of the notification email as follows: Job {instance-id()} has failed |
| Message body | <pre>Enter the following FlowForce Server expression: Exit Code: {string(exitcode(failed-step()))} Standard Error: {content(stderr(failed-step()))} Standard output: {content(stdout(failed-step()))}</pre> |
| | For information about the purpose of each function used here, see <u>Step Result Functions</u> . |

- 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 | <pre>Enter the following shell command: echo {instance-id()} >> JobLog.txt</pre> |
| | 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 JobLog.txt file created in previous steps is located. This must be an existing location on the machine where FlowForce Server runs, for example: |

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).



- 10. Under Credentials, select an existing credential record or specify a local credential (see Credentials).
- 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</u> Log).

12.5 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, 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).

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 as 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**.
- 3. 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 <u>Credentials</u>).
- 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 9 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**.

| Parameters | |
|-----------------|--|
| NamePrefix *: F | |
| Submit | |

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).

12.6 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 udates 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 <u>working directory</u> (see the next setting). |
| Working directory | Set the value to a directory on the machine where FlowForce Server runs, for example: |
| | c:\ |

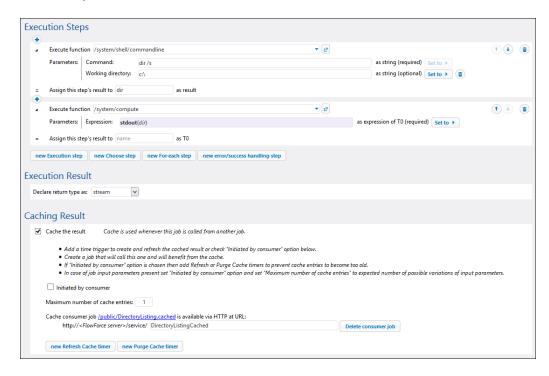
| Assign this step's result | We will need to refer to the value returned by the execution |
|---------------------------|--|
| to | step in a subsequent step, so it must have a name. For the |
| | scope of this example, enter dir as value of this field. |

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 Step Result Functions). |

- 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).



- 10. Under Credentials, select an existing credential record or specify a local credential (see Credentials).
- 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.

12.7 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

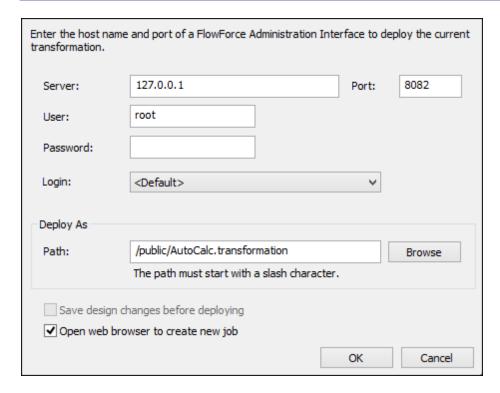
Demo files used

 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.)

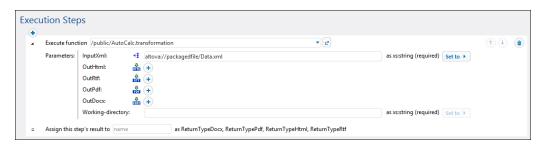
To create a job from a StyleVision transformation

- 1. Open the AutoCalc.sps file in StyleVision.
- 2. On the **File** menu, click **Deploy to FlowForce Server**. If this option is disabled, make sure the Design tab is currently selected.
- 3. When prompted to save the transformation as PXF file, 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 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.
- 9. 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.

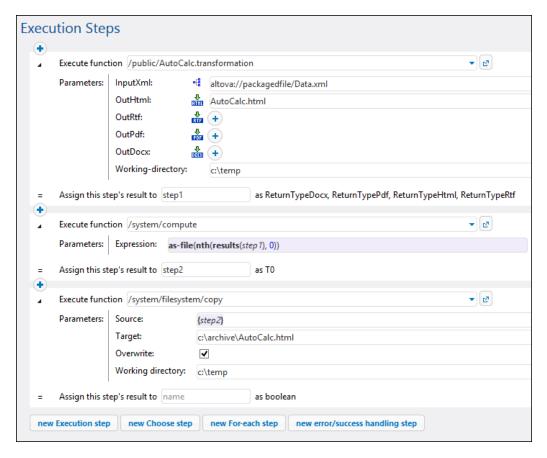
11. Under the execution step that is already created, click **new Execution step** and configure the step as follows:

| Execute function | Browse for the /system/compute function. |
|------------------------------|--|
| 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 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 step2 . This instructs FlowForce Server that the result returned by the step has the name step2 . We will need to refer to this result subsequently. |

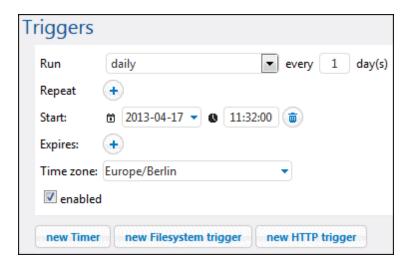
12. Click **new Execution step** and configure the step as follows:

| Execute function | Browse for the /system/copy function. |
|-------------------|---------------------------------------|
| Source | Click Set to , and then select step2. |
| Target | c:\archive\AutoCalc.html |
| Overwrite | Select the Overwrite check box. |
| Working directory | c:\temp |

At this stage, the Execution Steps section of the job page should look as follows:



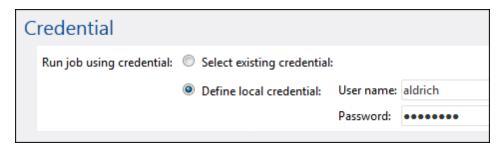
- 13. Under Triggers, click new Timer.
- 14. Next to Run, set the timer to run **Daily** every **1** days. Next to Start, select a date and time when the job must start, for example:



15. Under Credentials, select an existing credential record or specify a local credential (see also Credentials).

A credential record is the combination of user name and password associated with a user

account on the FlowForce Server operating system where the 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.



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).

12.8 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

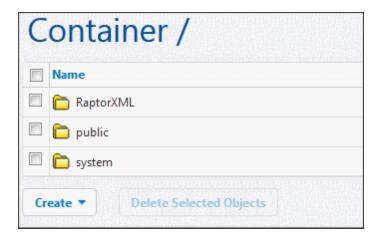
- Required licenses: FlowForce Server, RaptorXML Server
- FlowForce Server is running at the configured network address and port (see <u>Setting the</u> Network Address and Port)
- You have a FlowForce Server user account with permissions to one of the <u>containers</u> (by default, the /public container 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\RaptorXMLServer2015\examples\address.xsd, unless you installed RaptorXML Server in a different folder.

To create the validation job

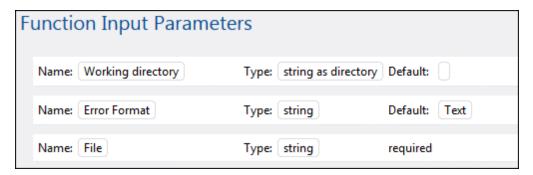
1. Click **Configuration**, and then click the "RaptorXML" container.



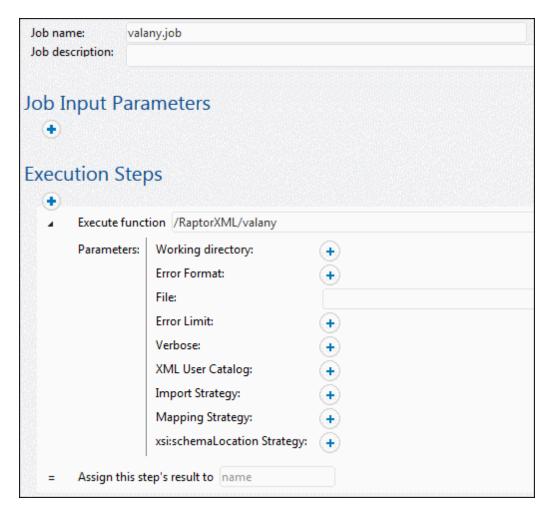
This opens the container and displays all the default functions available for RaptorXML.



2. Click the valany function. This displays a window showing all the Function Input Parameters along with their types and default values.



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\RaptorXMLServer2015\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 Credentials).
- 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. If the file did not validate, you will see result="Fail".

Log entry details: 2013-06-06 15:52:19

Date: 2013-06-06 15:52:19

Severity: INFO
Module: flowforce
User: root
Instance ID: 13

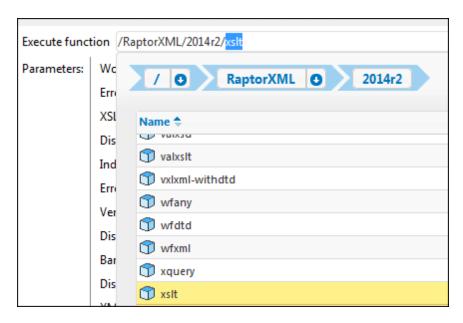
Message: Step RaptorXML.raptor.validateany completed with status: 0

file:///c:/Program%20Files%20(x86)/Altova/RaptorXMLServer2013/examples/address.xsd: result="OK"

12.9 Using RaptorXML to Pass Key/Value Parameter Pairs

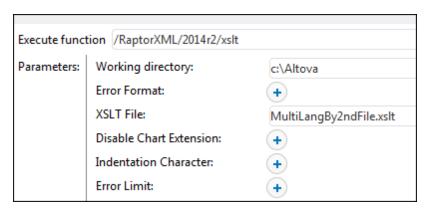
FlowForce Server can pass key and value pair parameters to XSLT stylesheets that are then executed by RaptorXML. In this generalized example, an XSLT file transforms an input XML file into a different language. A second XML file contains the original and translated values/terms. Two sets of related parameters are thus needed for this transformation.

Browse for the Txslt function located in the /RaptorXML container.

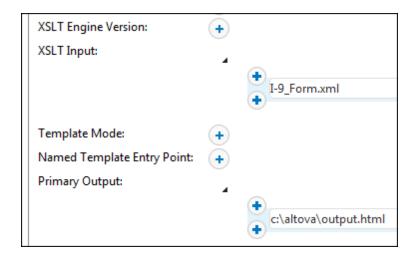


All the XSLT parameters are listed in the Parameters section.

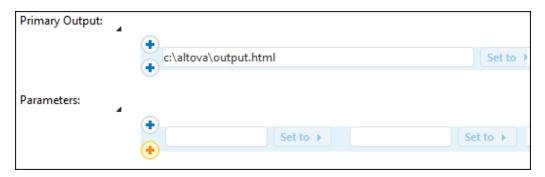
2. Enter the name of the XSLT file, e.g. MultiLangBy2ndFile.xslt.



3. Enter the name of the XSLT input file, e.g. I-9_Form.xml and then the name of the generated output file, e.g. 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.



All you need to do now is create a trigger that is to start the XSLT transformation.

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.

list(("Language","'G'"),("Translation","'FormTranslation.xml""))

Parameters:

| list(("Language","'G'"),("Translation","'FormTranslation.xml""))

Chapter 13

Glossary

13 Glossary

The glossary section includes the list of terms pertaining to FlowForce Server.

Glossary C 277

13.1 C

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

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: jobs, credentials, functions, and other containers. By setting permissions 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.

278 Glossary C

Credentials

A credential record is the combination of user name and password associated with a user account on the FlowForce Server operating system where the 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.

Glossary E 279

13.2 E

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

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.

280 Glossary F

13.3 F

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

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 281

13.4

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

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.

282 Glossary J

13.5 J

A B <u>C</u> D <u>E F</u> G H <u>I J</u> K L M N O <u>P</u> Q <u>R S T U</u> V <u>W</u> X Y Z

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 <u>input parameters</u>, <u>steps</u>, <u>triggers</u>, and other settings.

Glossary P 283

13.6 P

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

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.

284 Glossary R

13.7 R

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

RaptorXML Server

Altova RaptorXML Server (also called RaptorXML for short) is Altova's third-generation, super-fast XML and XBRL processor. It has been built to be 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 and XBRL data.

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 285

13.8 S

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

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.

286 Glossary T

13.9 T

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

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 287

13.10 U

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

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.

288 Glossary W

13.11 W

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

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

Active Directory,

integration with FlowForce Server, 104

Administrator interface, 81

Alert emails, 90

Altova LicenseServer,

(see LicenseServer), 45

Altova ServiceController, 52

Assigning licenses, 76, 85

B

Built-in functions,

RaptorXML Server, 220

C

Cache consumer,

definition, 277

Cache producer,

definition, 277

Choose step,

definition, 277

Command line interface,

getting help on, 188

Configuration page, 81

opening on Linux, 58

opening on Mac OS X, 60

opening on Windows, 55

URL of, 55

URL of (Linux), 58

URL of (Mac OS X), 60

Consumer,

definition, 277

Container,

definition, 277

restricting access to /public, 141

Containers,

creating, 135

moving, 135

overview of, 133

renaming, 135

setting permissions on, 139

viewing permissions of, 138

Credentials,

adding, 151

definition, 278

overview of, 150

D

Default password, 55

Default time zone,

configuration of, 102

Directory polling job,

example of, 250

Error handling job,

example of, 253

Error/success handling step,

definition, 279

Execution result,

definition, 279

Execution Step,

definition, 285

Expressions, 222

built-in functions, 208, 209

F

File System Triggers,

overview of, 166

FlowForce Server,

application data folder, 97

architecture, 14

basic job concepts, 9

basic security concepts, 12

built-in functions, 206

changing the language of, 199

| HowForce Server, | on Windows, 36 |
|--|---|
| command line interface, 186 | system requirements, 21 |
| configuration of, 96 | viewing currently installed products (Linux), 28 |
| definition, 3 | |
| glossary, 276 | |
| introduction to, 3 | • |
| localization of, 109, 192 | J |
| maintenance, 217 | |
| migrating to latest version of, 197 | Job, |
| new features in, 6 | definition, 282 |
| opening the setup page of, 99 | Jobs, |
| registering with LicenseServer, 65 | caching results of, 173, 260 |
| setting the network address and port of, 101 | creating, 152 |
| system requirements, 21 | creating from MapForce mapping, 241 |
| Web administration interface, 16 | creating from StyleVision transformations, 263 |
| FlowForce Server Function, definition, 280 | declaring the return type of, 159 defining queue settings of, 176 |
| | |
| setting the network address and port of, 101 | exporting to another FlowForce Server instance, 180 |
| For-each step, | exporting to file, 180 |
| definition, 280 | handling result of, 223 |
| FTP. | importing from file, 180 |
| built-in functions, 212 | stopping, 178 |
| Function, | using with RaptorXML, 268 |
| definition, 280 | viewing the log of, 177 |
| • | V |
| Н | K |
| HTTP Triggers, | Key/value pairs, |
| overview of, 168 | passing to XSLT sheets, 272 |
| • | • |
| 1 | L |
| Input Parameters, | License Pool, 62, 81 |
| adding, 154 | Licenses, |
| built-in, 154 | assigning, 76, 85 |
| definition, 281 | uploading, 62, 81 |
| removing, 154 | LicenseServer, |
| Installation, | Configuration page, 81 |
| installation folder (Linux), 30 | installation on Mac OS X, 51 |
| | installation on Windows, 48, 49 |
| installation folder (Windows) 41 | interface with, 81 |
| installation folder (Windows), 41 | registering FlowForce Server with, 65 |
| on Linux, 22 | registering Map Force Server with, 70 |
| on Mac, 31 | registering StyleVision Server with, 72 |

| LicenseServer, | Privilege, |
|--|---|
| settings, 90 | definition, 283 |
| starting, 53 | Privileges, |
| steps for assigning licenses, 53 | inheritance, 123 |
| LicenseServer configuration page, | list of, 123 |
| (see Configuration page), 55, 58, 60 | overview of, 123 |
| Linux, | viewing reports about, 127 |
| starting services on, 106 | |
| stopping services on, 106 | |
| Logout, 93 | R |
| | |
| R/I | RaptorXML Server, |
| M | definition, 284 |
| Mara | Registering FlowForce Server with LicenseServer, 65 |
| Mac, | Registering MapForce Server with LicenseServer, 70 |
| starting services on, 107 | Registering StyleVision Server with LicenseServer, 72 |
| stopping services on, 107 | Role, |
| Mail, | definition, 284 |
| as built-in function, 216 | Roles, |
| Mail parameters, | assigning roles to, 121 |
| configuration of, 103 | assigning to users, 120 |
| MapForce Server, | built-in roles, 118 |
| registering with LicenseServer, 70 Messages, 93 | creating, 115 renaming, 119 |
| Messages, 33 | renaming, 119 |
| N | S |
| Network information, 47 | Server Management tab, 76, 85 |
| Network settings, 90 | Server Monitoring tab, 90 |
| | ServiceController, 52 |
| | Settings, 90 |
| D | Shell commands, |
| P | executing as jobs, 219 |
| | Step, |
| Password, | definition, 285 |
| default at startup, 55 | Steps, |
| Password policies, | creating, 155 |
| creating, 145 | example of using in jobs, 246 |
| overview of, 144 | handling errors in, 157 |
| Password policy, | StyleVision Server, |
| definition, 283 | registering with LicenseServer, 72 |
| Permission, | |
| definition, 283 | |
| Permissions, | |

list of, 130 overview of, 130

T

Timer Triggers,

overview of, 164

Trigger,

Definition, 286

triggerfile parameter,

purpose of, 154

Triggers,

creating, 163

deleting, 163

U

Uploading licenses, 62, 81

Users,

built-in roles, 118

creating, 114

renaming, 119

W

Web administration interface,

definition, 288

Web Services,

exposing jobs as, 170, 257

Windows,

referring to network paths on, 206

starting services on, 108

stopping services on, 108

Windows domain users,

importing into FlowForce Server, 116

Working directory,

definition, 288