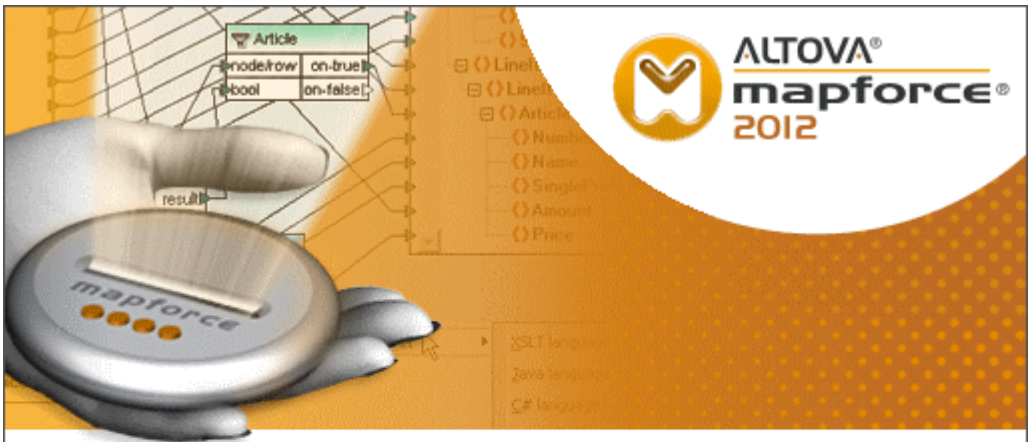
User and Reference Manual



Altova FlowForce User & Reference Manual

Published: 2012

Table of Contents

Index

Chapter 1

Altova FlowForce®

1 Altova FlowForce®

FlowForce® is a new Altova product that allows you to execute deployed mappings on dedicated high-speed servers.

FlowForce consists of the following modules:

- FlowForce Server
- FlowForce Administration Interface
- MapForce Beta
- MapForce Server (standalone Built-in execution engine)

For more details please see: FlowForce Architecture

This documentation is in multiple parts:

- The Administrator Guide describes how to install, setup and maintain the server, as well as how to define the access control settings.
- The Tutorial shows you how to deploy a mapping from MapForce, define a scheduled job in FlowForce Administration Interface, and execute that job to produce output files.
- The User Guide describes the browser application in more detail; the different trigger types and the various execution steps. It also describes how you can change the input/ output files supplied by the deployed mapping when the job executes.

What's new in Altova FlowForce® Beta3

- Jobs can be started via an HTTP request to the FlowForce Server. This allows FlowForce to be integrated into workflow solutions.
- Job input parameters allow values to be passed to individual execution steps of a job.
- Jobs can be called by other jobs.
- Timer triggers now support time zones.
- Jobs each have their own queue allowing job instances to run in parallel.
- A new system role, "Authenticated", is automatically assigned to all defined FlowForce users.
- Packages have been split up into individual functions.
- The new Configuration page unifies previously separate jobs, credentials, and packages pages.

Note: FlowForce Beta3 is not compatible with MapForceBeta2, or FlowForce Beta2.

Chapter 2

Administrator Guide

2 Administrator Guide

Download MapForce Beta3 and FlowForce Server Beta3 from the Altova download web page.

- **MapForceBeta3** consists of Altova MapForce, and FlexText
- **FlowForceServerBeta3** consists of FlowForce Server and FlowForce Administration Interface

  Both packages currently only run on Microsoft Windows. The applications will automatically stop working on 2013-03-31. MapForceBeta3 requires a license keycode for activation which you can request from the Download page.

The FlowForce Beta version does not include support for:

- XML Catalogs
- Global Resources
- Java and C# functions
- URL access in mappings

The Administrator guide focuses on the specifics of FlowForce namely:

- Getting started with FlowForce, i.e. how to install and start FlowForce
- Data storage and the various configuration files of FlowForce Server
- How to define the Access Control settings

2.1 Architecture

A FlowForce installation consists of several modules:

FlowForce Server
The FlowForce Server is the core of the FlowForce system 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. To execute job steps that use a deployed MapForce mapping, FlowForce sends an execution request to MapForce Server.

MapForce Server
MapForce Server is an implementation of the MapForce Built-in execution engine that executes mappings previously deployed via the MapForce graphical environment. MapForce Server is always installed on the same machine as FlowForce Server.

FlowForce Web Administration Interface
This is the module that acts as the front-end of FlowForce Server. It is a standalone web application that can be installed on the same machine as FlowForce Server or on any other machine in your environment.

The user interface runs in an internet browser and allows administrators to configure access control settings as well as the specific server operations including jobs, triggers, etc.

MapForce
The powerful MapForce graphical mapping application has been enhanced with an integrated deployment feature. Once a mapping has been tested and debugged, MapForce lets you deploy it to FlowForce server. The newly deployed mapping is then immediately available for use in any job on the server.

An administrator or developer runs MapForce on a personal Windows workstation to develop and deploy mappings onto the high-speed server.

2.2 FlowForce concepts

Configuration

Configuration data in FlowForce server's database are comprised of various objects that define the operation of FlowForce. This includes jobs, credentials, functions, triggers, and other objects.

Configuration objects are organized in a freely defined hierarchy of containers. Some configuration settings are edited together (e.g. jobs include triggers), and other settings can also be stored as standalone objects under their own name (e.g. credentials and functions).

Container
A container is similar to a folder in a commonly used file system. It is used to create a hierarchical structure for storing configuration objects and other containers. Containers can be assigned access permissions.

Two predefined containers exist in FlowForce: /system which contains system functions, e.g. copy, move, etc., and /public which is the default container when deploying a mapping to FlowForce from MapForce. Other containers can be created as needed, e.g. for departments or user groups.

Function
A FlowForce function performs a specific operation when used in a job execution step. It may have input parameters that need to be passed to it by the caller. Available functions include the system functions delivered with FlowForce, deployed MapForce mappings, and the execution steps of other jobs.

Job
A Job consists of Triggers, Execution steps, input parameters, and other settings. Triggers define when a job will be executed, and the execution steps define what the job actually does when it executes. Multiple triggers and execution steps can be defined per job.

Trigger
Triggers define under which circumstances a job will be executed. Three types of triggers can currently be defined: Timer triggers, File system triggers, and HTTP triggers. Multiple triggers can be defined per job.

Service
FlowForce permits exposing jobs as web services via the HTTP protocol. This allows interactive or automated access to these jobs.

Credential
Credentials are stored login data used to execute FlowForce jobs. Credentials can be defined as standalone "objects" and be assigned to various jobs, or they can be manually entered for a specific job.

Queue
The queue settings in a FlowForce job allow limiting the number of parallel job executions to control use of server resources.

Access Control

All important operations in FlowForce are linked to permissions or privileges which need to be assigned to the user to successfully execute them.

User

FlowForce users are persons that have been added to FlowForce by the FlowForce administrator with a login name and a password. Depending on the assigned rights and privileges, users can define FlowForce jobs, deploy mappings, or view logs.

Two special users are predefined by FlowForce: "root" is the initial administrator user, and "anonymous" is a special user account used for FlowForce services that should be available to users without explicit log in to FlowForce.

Role
Roles are used to manage privileges and object permissions for user groups as opposed to individual users.

Having defined users, you can assign them to a role thus creating user groups. The users become "members" assigned to the specific role.

Permission
Permissions control access to containers and configurations. Unlike privileges they can be redefined on every level of the container hierarchy, and are by default inherited from parent containers.

Permissions, like privileges, are inherited from all roles the user is a member of, as well as from permissions directly assigned to the user.

Privilege
Privileges control user rights globally. This means privilege settings cannot be overridden in the container hierarchy of FlowForce.

When a user logs into FlowForce, the set of effective privileges is determined by the user privileges and all role privileges the user is member of.

2.3 Getting started

This section deals with what the first-time user of FlowForce, the administrator, has to do to set up the software and configure it for multiple users.

Install the software

To install the software, see: Install.

Start FlowForce

To start FlowForce and change your default password to something new, see: Starting FlowForce.

Add new users

To add new users, see: Users.

- Users are persons that are allowed to define and/or start jobs. Users could be DeployUsr, Operator, etc.
- Note that users inherit privileges from all their roles in addition to the privileges defined here, so it is better to define them in the roles page.

Add new roles and define the role privileges

To add new roles and define privileges, see: Roles.

- The Role page lets you create new roles and define the role privileges.
- Roles are used to manage privileges and object permissions for user groups instead of individual users.
- This is the place that you define **role privileges,** as the privileges defined here are automatically inherited by users when you assign a role to a user.

- Having defined the users in the previous step, you can now assign users to a role, thus creating user groups assigned to the various roles. (The users become "members" assigned to the role).

  To assign users to a role, see: Assigning a user to a role.

Define the work environment (container structure) and the read/write/use permissions of your users

To add new permissions to the permission list see: Permissions.

- Containers are used to organize jobs and mapping deployments into a hierarchy similar to that of a file system composed of folders.
- Read/Write/Use permissions should generally only be assigned to **roles**, not to individual users (although this is possible).

Define the necessary credentials, i.e. the login data needed for FlowForce to access your operating system user accounts

To add new credentials, see: Credentials.

- Credentials are stored **login** data used to execute FlowForce jobs, and are stored in the FlowForce database as separate objects.
- Jobs are started automatically by FlowForce server, when the defined trigger conditions are met. FlowForce server then runs these jobs using a specific operating system **user account**, ensuring that execution steps do not access unauthorized data.

- Every job MUST have a credential assigned to it for the execution steps to be executed. This defines the **operating system** user account used to run the job execution steps.
- Since a job is defined to run with a certain credential, it is vital for the success of the job execution that the **windows user** which is assigned to this credential has sufficient access permissions.

Additional step when upgrading from FlowForce beta2
FlowForce beta3 does not yet support full backward compatibility with FlowForce beta2. Therefore MapForce mappings used by jobs must be deployed again to the same path using MapForce beta3.

See: Deploying a MapForce mapping.

If you run a job without redeploying it, it will fail with the log error message "No tool definition file found for tool 'MapForce' version '2012beta2'".

2.3.1 Installation

Having downloaded the software from the Altova website download page

1. Double click the installation file to start the installation process.
2. Follow the wizard instructions to install the software.

File paths in Windows XP, Windows Vista, and Windows 7

File paths given in this documentation will not be the same for all operating systems. You should note the following locations:

- FlowForce stores all data in the following locations:

Windows XP	`C:\Documents and Settings\All Users\Application Data\Altova\AltovaServersBeta`
Windows Vista, Windows 7	`C:\ProgramData\Altova\AltovaServersBeta`

- Application folder: The Application folder is the folder where your Altova application is located. The path to the Application folder is, by default, the following.

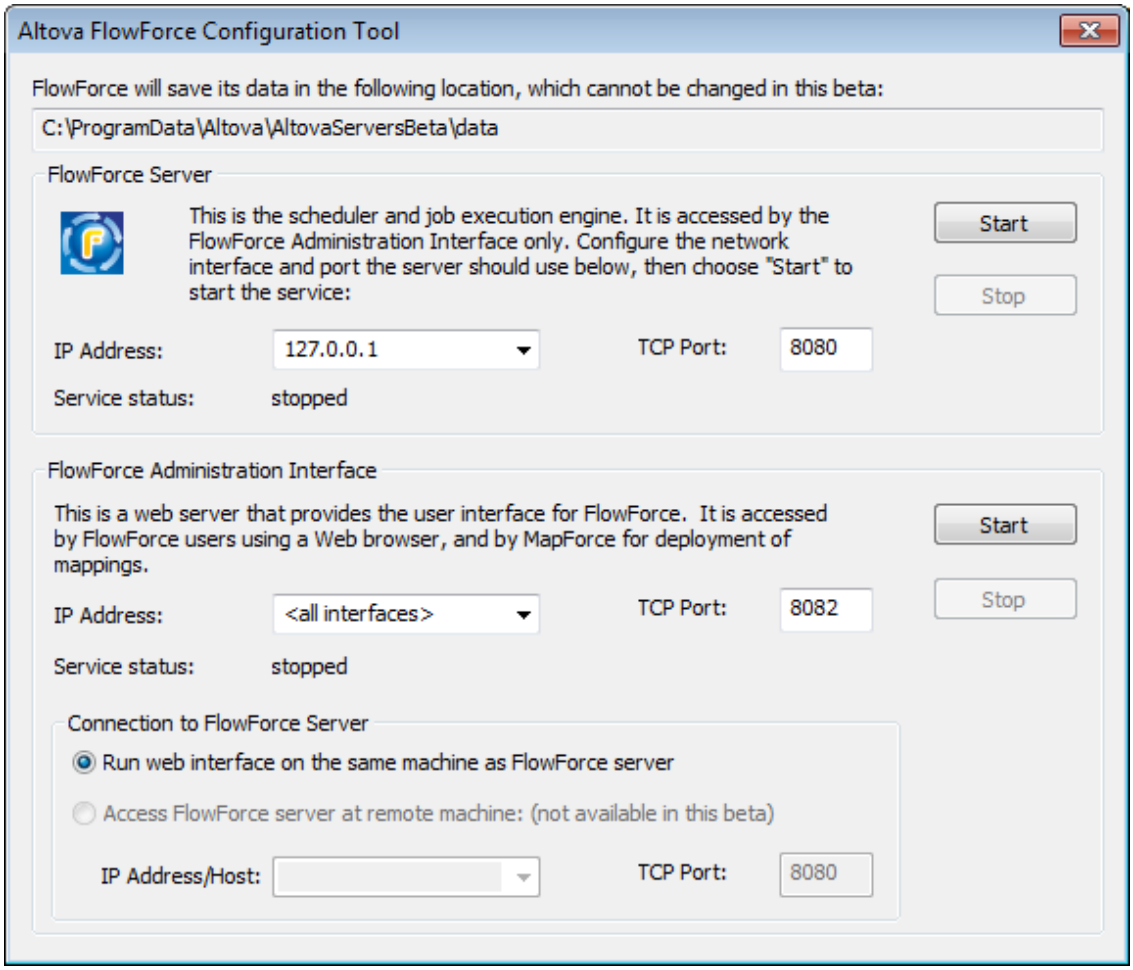
Windows XP	`C:\Program Files\Altova`
Windows Vista, Windows 7	`C:\Program Files\Altova`
32 bit Version on 64-bit OS	`C:\Program Files (x86)\Altova`

Next: Starting FlowForce

2.3.2 Starting FlowForce

To Start FlowForce Server Beta

Click the Windows "**Start"** button and select **| All Programs | Altova FlowForce Beta | Altova FlowForce Configuration Tool.**



The default addresses and ports will usually work fine, except if other services on the machine already use one of the ports, in which case you can change the ports used by FlowForce here.

Note: The default setting for the server accepts only requests from the same machine (127.0.0.1). If you intend to start jobs as web services via HTTP from remote machines, select <all interfaces> from the IP Address combo box.

Click the "Start" button for both FlowForce Server and FlowForce Administration Interface to start these services.

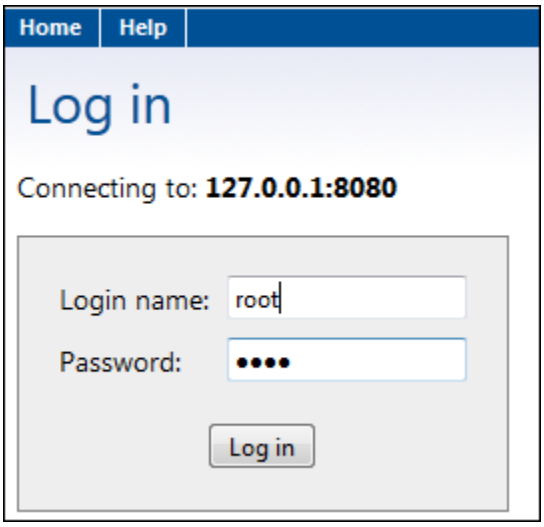
Note:
The FlowForce services are automatically started on every machine startup. Use the Windows control panel to disable the services.

The "Services" management console can be found in "Administrative Tools", and can also be started using Start | Run | services.msc.

To start FlowForce Administration Interface:

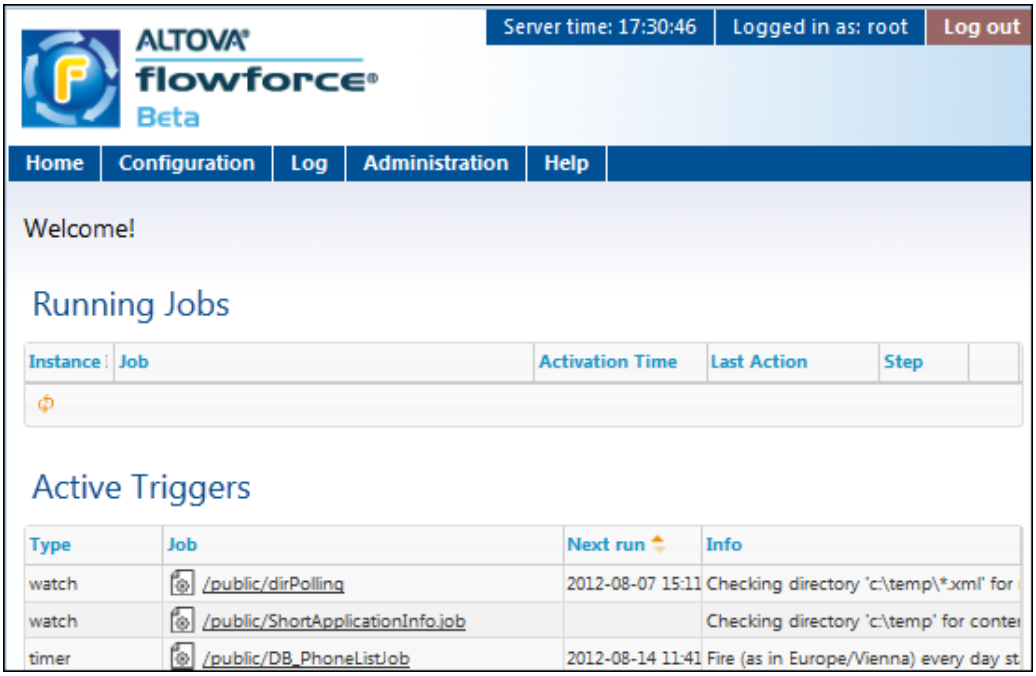
1. Start you browser and enter http://localhost:8082. If you changed the port in the

FlowForce Configuration tool, use the one you entered there.
This opens the Login mask.



Enter login name "**root**", as well as the password "**root**" if this is the first time that you have started FlowForce.

2. Click the "Log in" button to log in.
Connection information, as well as any running jobs and active triggers are visible on the Home screen.



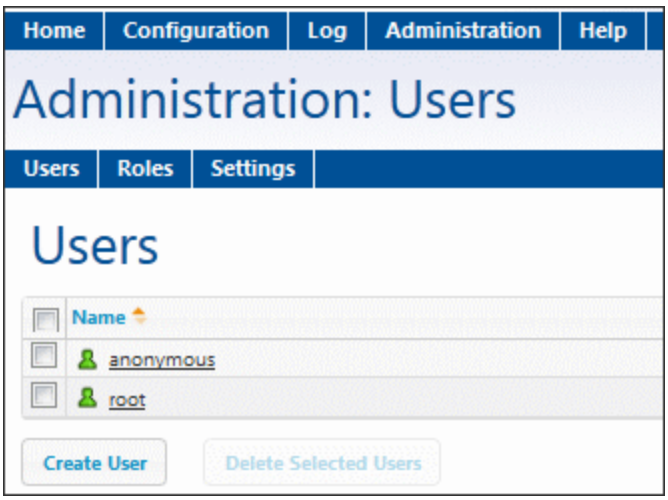
Logging off:

Click the "Log out" button at the far right of the browser window to log out.

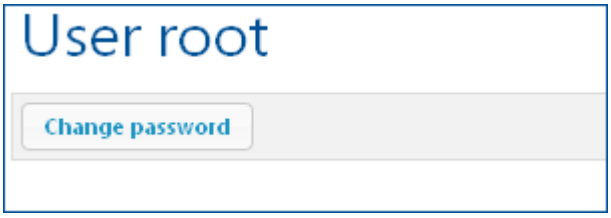
To change your default password:

From the Home page shown above:

1. Click the "Administration" button, then the "Users" button.



2. Click the "root" user entry in the Users table.
3. Click the "Change password" button and enter your old and new passwords.



4. Click Save to complete the process.

2.4 FlowForce Server

FlowForce Server is a background service that stores all configuration and other important data in a database. Configuration data can consist of: user profiles, roles, triggers and jobs. FlowForce Server has no user interface and is configured via FlowForce Administration Interface.

Data Storage / Server Instance
All data for FlowForce is stored in a single directory (with subdirectories).

Configuration:

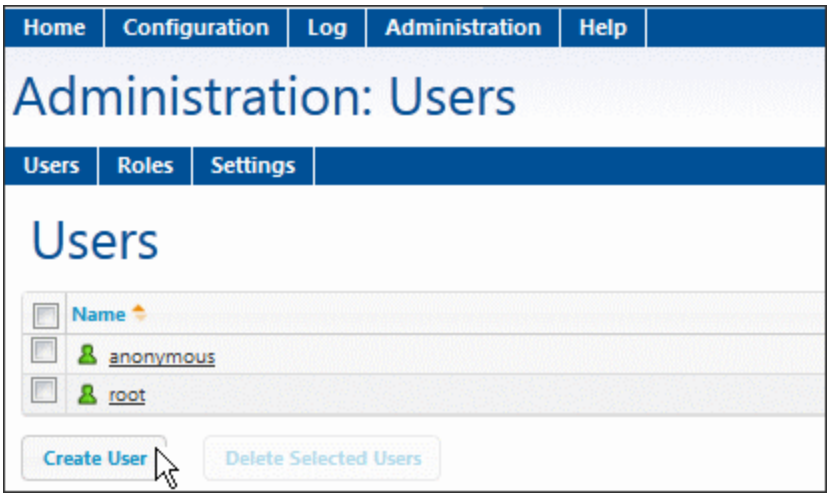
- **flowforce.ini**: the configuration file defining the port and listening interfaces of the server instance
- **flowforce.db**: the main database file storing the FlowForce object system, user data, active jobs, roles, etc.
- **flowforcelog.db**: the database that stores all FlowForce Server logs
- "**files**" subdirectory: stores files associated with deployed mapping functions
- "**tools**" subdirectory: contains a single file for each tool, e.g. "MapForce.tool", which is an ini file containing the server tool specification, i.e. file path
- "**logs**" subdirectory: contains captured output from job execution steps
- "**tmp**" subdirectory: stores temporary files

2.5 FlowForce Administration Interface

FlowForce Administration Interface is the module that acts as the front-end of FlowForce Server. FlowForce Administration Interface can be accessed using an internet browser and allows you to configure the specific server actions such as: jobs, triggers, etc.

FlowForce supports the current versions of Mozilla Firefox, Google Chrome, and Microsoft Internet Explorer 9 and 8. Note: When using Internet Explorer 9 as your browser, please disable the "Show friendly HTTP error messages" in the Advanced tab, to view the HTML form.

There are several menu items available in the browser window:



Home
Displays the connection details as well as a list of currently running/aborted jobs and active triggers.

Configuration
Displays the contents of the container hierarchy, and allows you to access containers, jobs, credentials, and functions by clicking them in the list. New containers, jobs, and credentials, are added using the "Create " button. See the User Guide for more details.

Log
Displays the log table. Can be filtered by time, job name and severity.

Administration
Lets you view and edit/access server-wide configuration, such as user accounts, roles and other settings.

Users
Allows you to create, remove, and maintain users.

Roles
Allows you to create, remove, and maintain access control roles.

Settings
Allows you to define your default time zone.

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

2.6 FlowForce Security

FlowForce grants access control to containers and configurations. Logged-in users are granted privileges and permissions based on explicit assignments to their user account, as well as the roles the user is a member of.

Privileges are global rights, independent of the FlowForce container hierarchy, whereas permissions are inherited down the hierarchy and can be refined at each level.

Users and Roles:
Users are persons that have been added to FlowForce by the FlowForce administrator. Depending on the assigned rights and privileges, users can define or run FlowForce jobs.

Roles are used to manage privileges and object permissions for user groups as opposed to individual users.

Privileges:
Privileges control user rights globally and can be defined in the Users and Roles pages.

Permissions:
Permissions control access to containers and configurations. Unlike privileges they can be redefined on every level of the container hierarchy, and are by default inherited from parent containers.

Credentials:
Credentials are stored login data used to execute FlowForce jobs.

2.6.1 Users and Roles

A user account defines a log-in name and has a set of roles the user is a member of. A role can be a member of another, broader role, which makes all members of the narrower role also members of the broader role.

This enables the definition of hierarchical access rights, the role Director of Marketing may be a member of Marketing, which in turn may be a member of Employees. Assigning a user, e.g. Bob, the role Director of Marketing, automatically makes him a member of Marketing and of Employees as well. When Bob logs into FlowForce he will be granted all privileges and permissions granted to any of his roles.

Two special users are predefined by FlowForce:

root is the initial administrator user. It is by default all-powerful and allows tasks such as adding other users and roles, as well as setting up privileges and permissions.

anonymous is a special user account for users that do not explicitly log in. Anonymous access to the FlowForce Administration Interface is not possible, but you can enable anonymous access for certain services exposed via the HTTP service interface.

Two special roles are also predefined by FlowForce:

authenticated is the role automatically assigned to every user **except** anonymous. It therefore includes every user who is authenticated using an existing user name and password.

all is the role automatically assigned to every user including anonymous.

While any users you create will be members of both all and authenticated, any roles you create are not by default members of any other role.

See also:
Privileges
Permissions
Credentials

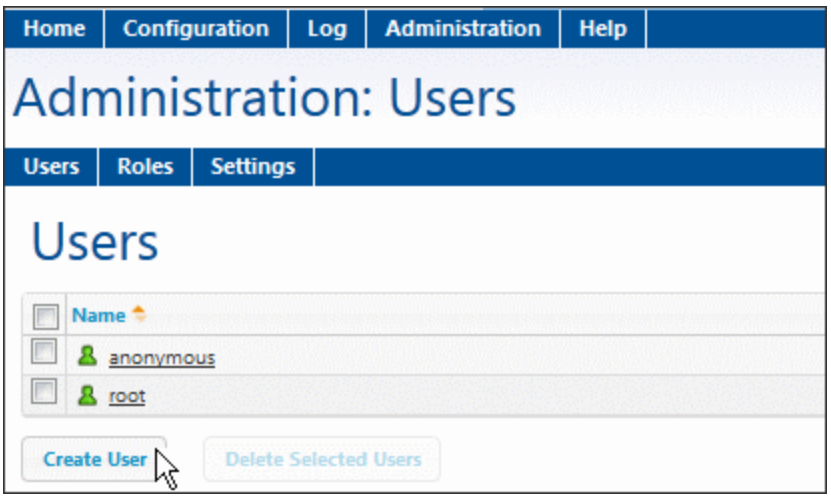
Defining restricted user rights

How to add Users

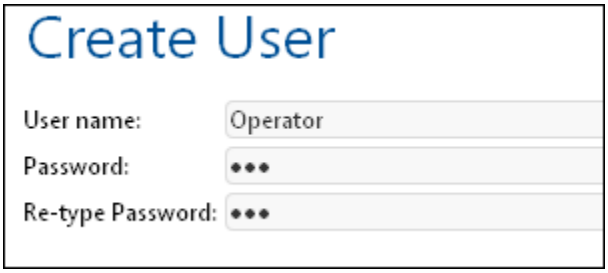
This page allows Administrators to define users and assign them user roles. The user name and password needed to access the web administration interface, or to deploy MapForce mappings, are defined here.

To add a user to FlowForce:

1. Click the "Create User" button on the Users page.



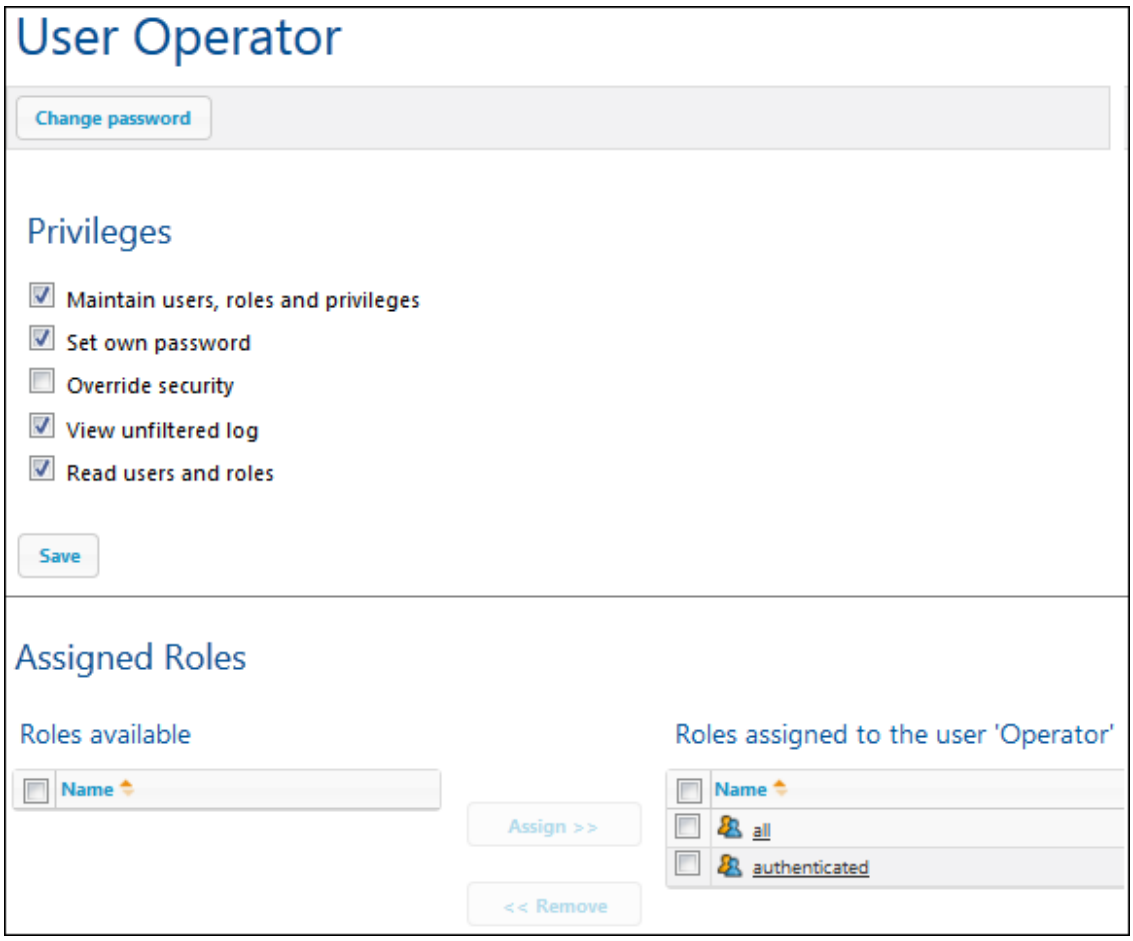
2. Enter the User name and password.



3. Define the privileges of this user (on the same browser page) by activating the Privileges check boxes. Note users inherit all privileges of their assigned roles; we therefore recommend assigning privileges to roles only to simplify maintenance.

4. Click Save to save the user.

Users and Roles
Roles make it easy to define user groups such as project teams, branch offices etc. To assign roles to a user on **this** page, roles must have been previously defined on the Roles page.



You can however assign a user to a role on the Roles page if you want to.

To assign one or more roles to a user:

1. Click the role name check box(esc) of the role(s) you want to assign (e.g. Deploy_mapping which is added in the following Roles topic).
2. Click the "Assign" button to assign the role(s) to this user.
   Role assignments are saved immediately.

The Deploy_mapping role has now been assigned to the user "Operator".

Note that the roles "all" and "authenticated" are default roles supplied with FlowForce.

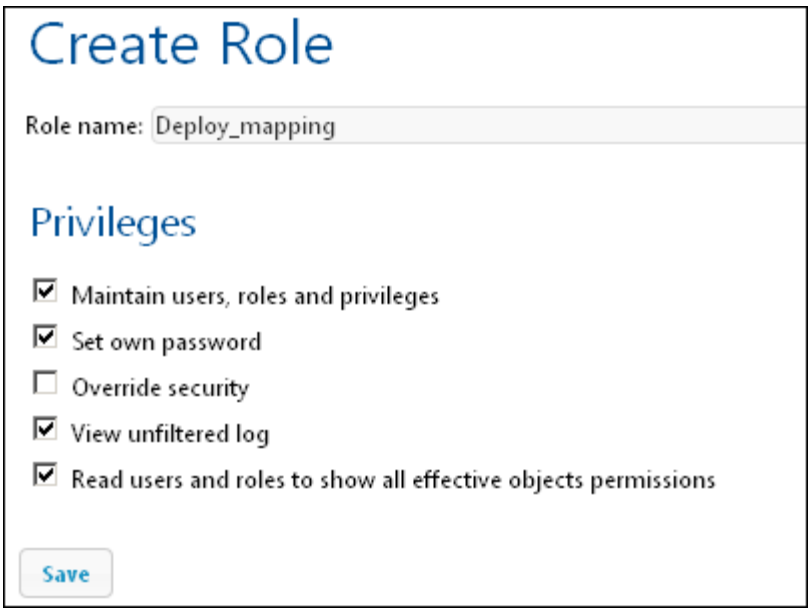
See also: How to add Roles

How to add Roles

To add a role to FlowForce:

1. Click the "Create Role" button on the Roles page.

2. Enter the Role name (e.g. Deploy_mapping) and define the privileges the members of this role should have, by activating the Privileges check boxes, then click Save to save the role.

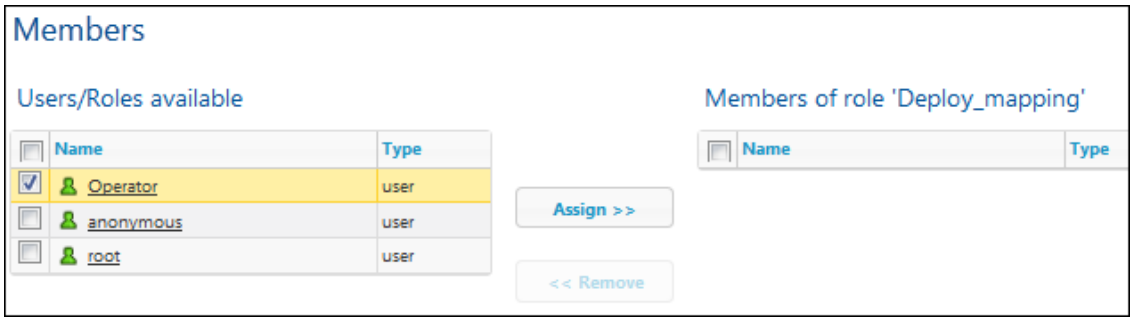


Note: Users and roles cannot have the same names in FlowForce.

Roles and Users
To assign a user to a role on **this** page, users must have been previously defined on the Users page.

To assign a user to a role:

1. Click the user (or role) check box that you want to assign the role to, in the "Users/ Roles available" table.
2. Click the "Assign" button to assign the user(s) to this role.



The user "Operator" has now been added to the Deploy_mapping role.

Assigning roles to roles
FlowForce makes it possible for you to assign roles to roles. What this does is allow indirect

inheritance of permissions/privileges.

- If user A is a member of role B, and role B is a member of role C, then user A is also an indirect member of C.

See also: Defining restricted user rights

Defining restricted user rights

To restrict access to the FlowForce /public container, it is recommended to perform the following steps:

- Take away permissions and privileges from the authenticated role.
- Group users requiring extra permissions in a new role.
- Assign extra permissions to the new role.

To take away permissions and privileges from the authenticated role:

1. Go to the Configuration page.
2. Click the "Permissions" button to the right of the /public container.
3. Click the "Change" button for authenticated.
4. Define the permissions that should apply to all users.
5. Click "Save Changes" to commit your changes.

To group users requiring extra permissions in a new role:

1. Go to the Administration page.
2. Click Roles to go to see the roles list.
3. Click the "Create Role" button to create a new role.
4. Enter a descriptive role name, e.g. Operators.
5. Click "Save" to save your changes.
6. Select the users you want to assign to the role from the list on the left (Users/Roles available).
7. Click "Assign" to assign them to the new role.

To assign extra permissions to the new role:

1. Go to Configuration page.
2. Click the "Permissions" button next to the /public container.
3. Click "Add Permissions".
4. Select the role from the combo box.
5. Define the extra permissions for this role.
6. Click "Save Changes" to commit your changes.

See also:
Privileges

2.6.2 Privileges

Privileges control several user rights globally. This means privilege settings cannot be overridden in the container hierarchy of FlowForce. When a user logs into FlowForce, the set of effective privileges is determined by the user privileges and all role privileges the user is member of.

It is recommended to assign privileges only to roles, and have the assignment to users take place via role membership.

Privileges are assigned on the user and role pages of the FlowForce Administration Interface.

Maintain users, roles and privileges
Any user having this privilege can create, delete and edit users and roles, their privilege assignments and passwords.

This is an administrative privilege and should only be assigned to FlowForce administrators.

By default, only the user "root" possesses this privilege.

Set own password
Any user having this privilege can change his own password. Users who do not have this privilege need to have their password set by a FlowForce administrator.

By default the "authenticated" role, and hence every user account except "anonymous", possesses this privilege.

Override security
Any user having this privilege can change permissions in the container hierarchy without needing "write" security permission. This allows FlowForce administrators to regain access to resources accidentally rendered inaccessible.

This is an administrative privilege and should only be assigned to FlowForce administrators.

By default, only "root" possesses this privilege.

View unfiltered log
By default users can only see log entries related to Configurations they have "read" access to. By granting this privilege a user can read all log entries, including those not associated with a specific configuration.

By default, only "root" possesses this privilege.

Read users and roles
By default users will only see their own user account and any roles they are member of. By granting this privilege a user can read all defined users and roles.

By default, only "root" possesses this privilege.

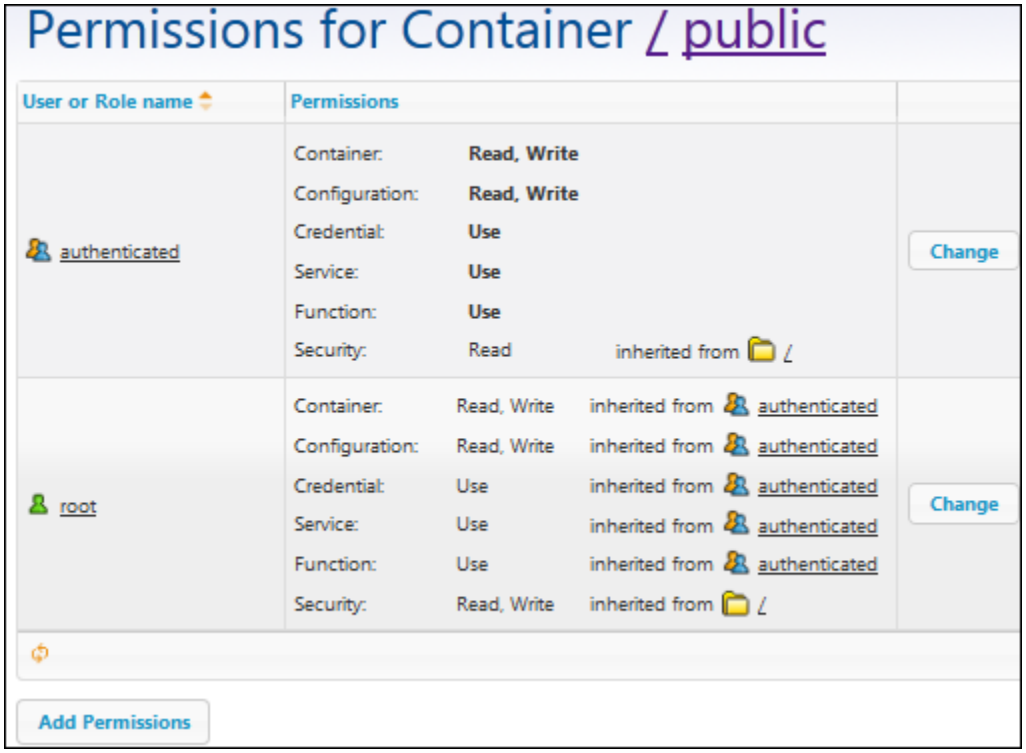
See also:
Permissions

2.6.3 Permissions

Permissions control access to containers and configurations. Unlike privileges they can be redefined on every level of the container hierarchy, and are by default inherited from parent containers.

Permissions, like privileges, are inherited from all roles the user is a member of, as well as from permissions directly assigned to the user.

This inheritance takes precedence over container hierarchy inheritance. If a permission is redefined for any role the user is a member of, container hierarchy inheritance for this particular permission is overridden.



Permission checks are performed on every user interaction. A user can only successfully edit a Configuration when all required permissions are granted. Permissions are not evaluated upon job execution, therefore any permission changes will not retroactively apply to already previously jobs.

FlowForce assigns permissions in several groups:

Container
The container permissions define what a user can do with objects in a container.

If the container has "read" access, the user can list the contents and find an object in the container.

If the container has "read"/"write" access, the user can additionally create new (and delete existing) objects within the container, depending on other permissions that may apply.

Configuration
The configuration permissions define what a user can do with a configuration (job, credential).

If the configuration has "read" access, the user can look at the details of that configuration, such as the defined execution steps or the defined triggers.

If the configuration has "read"/"write" access, the user can additionally modify that configuration. To successfully create a new configuration, or delete an existing one, the user must be permitted "write" access for the container and the configuration.

Credential
The credential permission defines what a user can do with a credential. This makes it possible to supply credentials for reuse.

If the credential has "use" access, the user is permitted to refer to this credential from another configuration.

Service
The service permission defines access to a job via the HTTP request interface.

If the service has "use" access, the user is permitted to access the service and thus execute the job via the request interface.

Note that service permission checks skip any container hierarchy checks. If a user is permitted to use a service he may do so without having "read" access to the container the corresponding job is defined in.

Also note that by granting service use to "anonymous", it becomes possible to use that service without any authentication.

Function
The function permission defines whether a user can invoke a function as an execution step in another function.

If the function has "use" access, it is permitted to call this function from another function.

Security
The security permission controls access to the container's child permission lists.

If security has "read" access, the user is permitted to read the permission list of any child of the container.

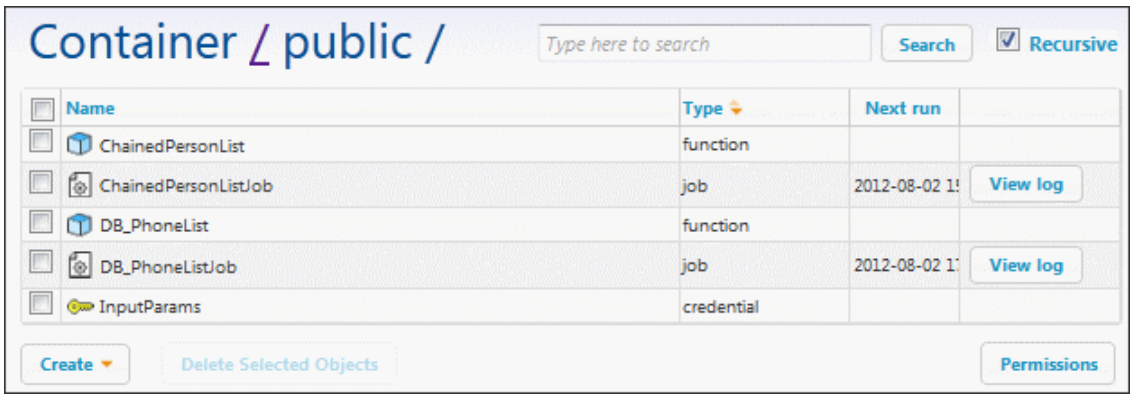
If security has "read"/"write" access, the user can additionally change the permission list of any child of the container.

By default users are permitted to read only permissions applicable to them. That means any permissions assigned to themselves or any role they are a member of. If the "Read Users and Roles" privilege is granted, users can read all permission entries.

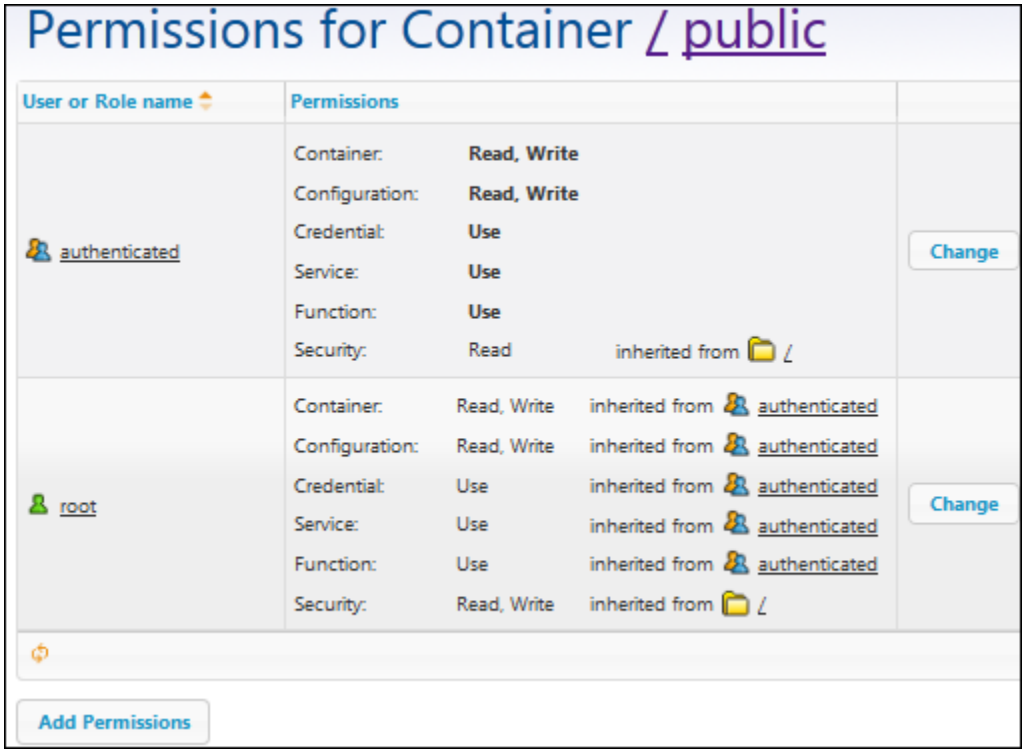
See also:
How to add Permissions

How to add Permissions

Permissions can be assigned to both a user and a role. Read/Write/Use permissions should generally only be assigned to **roles** and not to individual users, to simplify maintenance.



Permissions are accessed by clicking the "Permissions" button at the bottom of the right of the page, or next to the container that you want to define the permissions for. The screenshot below shows the permissions overview for the /public container.

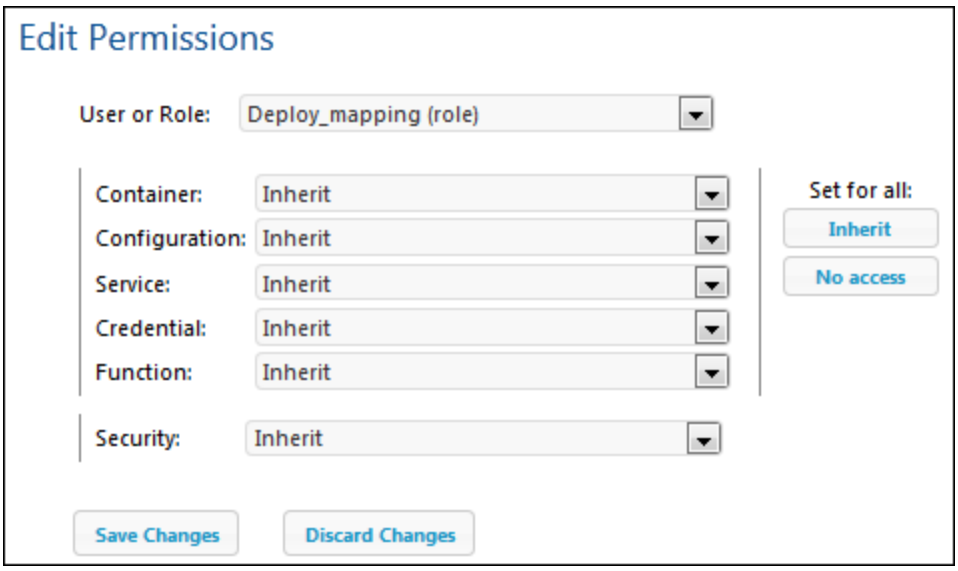


Permissions can be inherited from containers above the current container, if "Inherit" is selected (shown as the forward slash character **/** for the root container) on the Permissions overview page shown above.

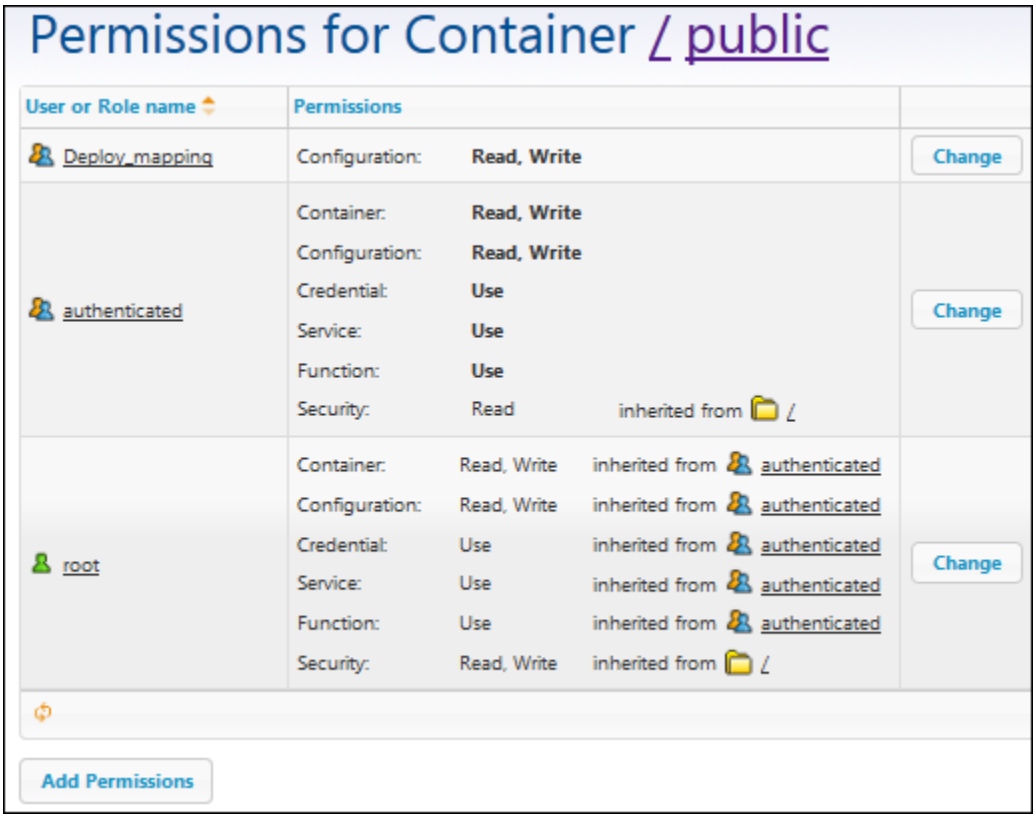
To add a new Permission to a user or role:

Having clicked the "Permissions" button on the Container page:

1. Click the "Add Permissions" button on the Permission overview page of the container you want the permission to be added to, e.g. Container /public.
2. Select a previously defined role (or user) from the "User or Role" combo box e.g. "Deploy_mapping (role)".



3. Change the "Configuration" permission to "Read/Write" and leave the other permissions as they are currently defined (Inherit for all), then click "Save Changes".



The Deploy_mapping permission list has now been added to the container "/public".

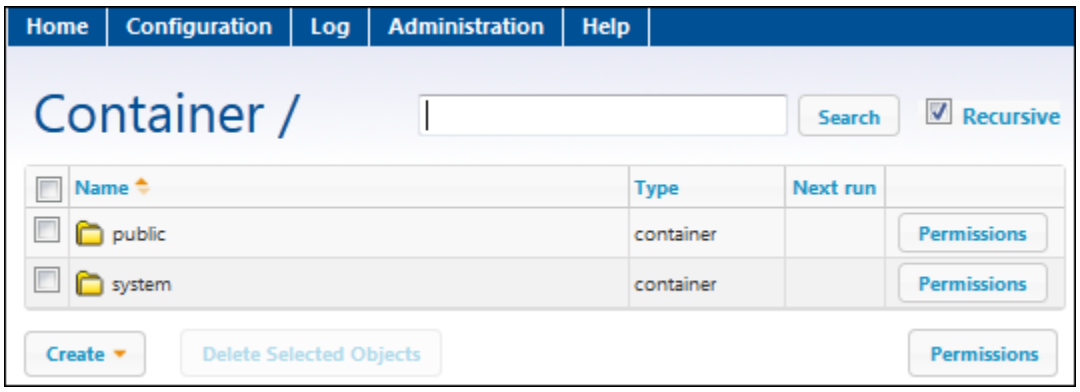
2.6.4 Credentials

Credentials are stored login data used to execute FlowForce jobs. Credentials can be defined as standalone "objects" and be assigned to various jobs, or they can be manually entered for a specific job.

Jobs are started automatically by FlowForce server, when the defined trigger conditions are met. FlowForce server then runs these jobs using a specific operating system user account, ensuring that job steps do not access unauthorized data. Note that file watch triggers are also assigned credentials.

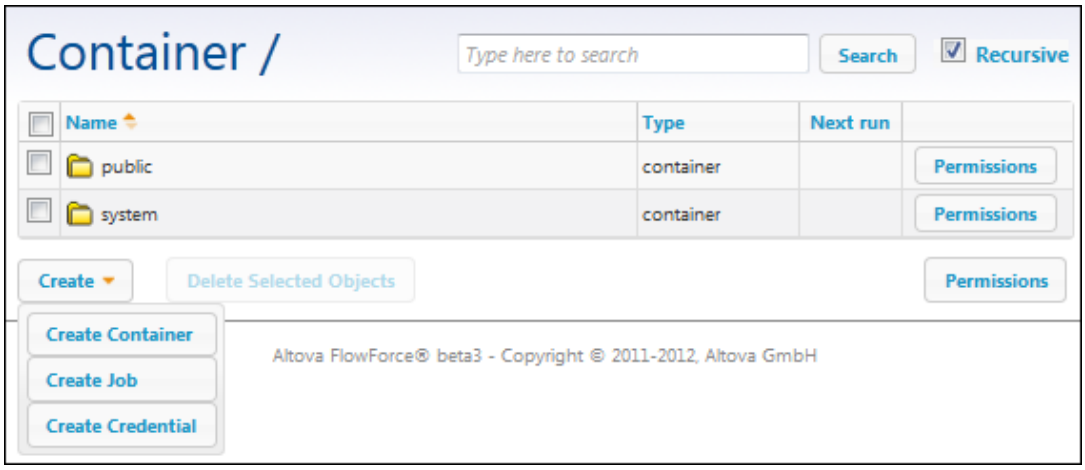
Credentials can be created, or deleted, on the Configuration (Container) page. Note that job credentials, i.e. username and password, can now also be entered for individual jobs on the Job page.

Any user that has "write" access to the Configuration permission, can edit or remove credentials.

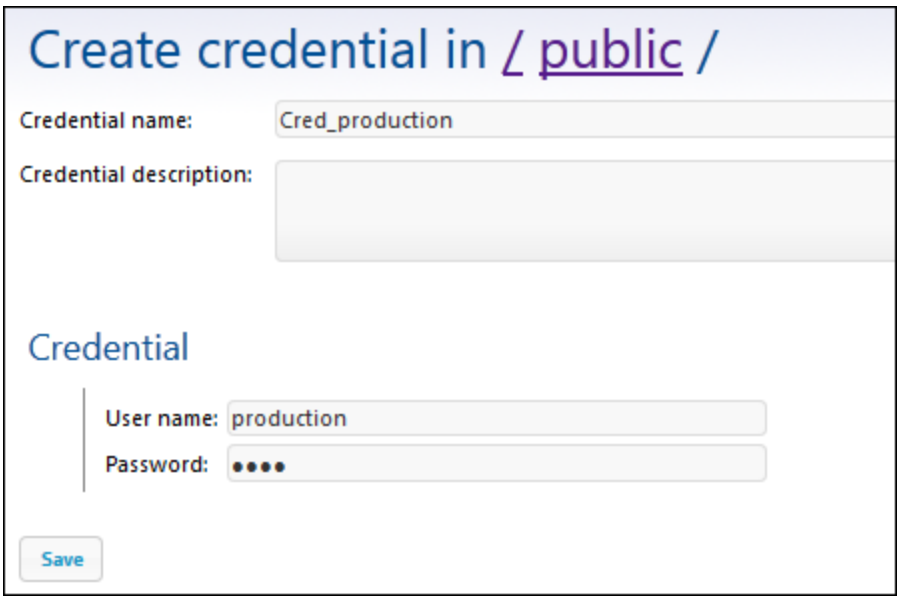


To add a credential to FlowForce:

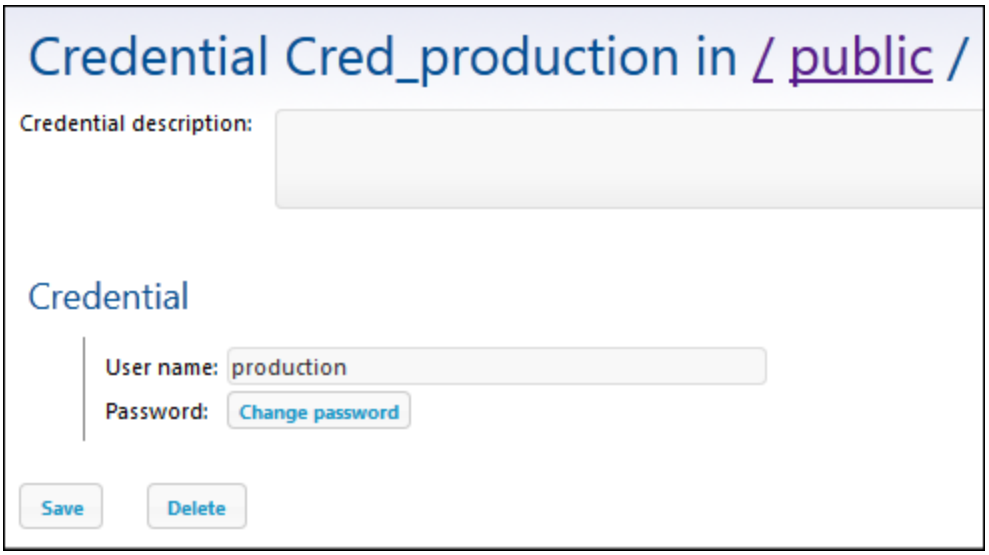
1. Click the container you want to create the new credential in, e.g. public.
2. Click the "Create" button and select the "Create Credential" entry.



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, please use the form **username@domain**.



4. Click Save.
   The new credential "Cred_production" has been saved in the /public container.
5. Click the "Configuration" button to return to the Container page.



Please see Permissions for information on the container permissions that can be defined.

Credentials and jobs
Every job MUST have a credential assigned to it for the job steps to be executed. This defines the **operating system** user account used to run the job steps.

A predefined credential can be selected using the combo box, or the job credentials can be manually entered in the "User name" and "Password" fields.

Note:
If you manually enter the user name and password, you will have to update them for those specific jobs, whenever your server credentials are changed.

Credentials:

- Credentials can be created in any container a user has access to.
- 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 database. The administrator should make sure to restrict access to the FlowForce database file.

Required Permissions for a user to execute a job
Since a job is defined to run with a certain credential, it is vital for the success of the job execution that the **windows user** which is assigned to this credential has sufficient access permissions.

The Windows user needs the following **file system** permissions:

- Execution permission for MapForce Server and all referred DLLs (implicitly set).
- Read permission for the folder where the temporary deployed mapping files are saved.

  ```
  C:\Documents and Settings\All Users\Application
  Data\Altova\AltovaServersBeta\tmp - WinXP

  C:\ProgramData\Altova\AltovaServersBeta - Windows Vista &
  Windows7
  ```

- Read permission for the FlowForce file data folder - This is beta3 limitation.

  ```
  C:\Documents and Settings\All Users\Application
  Data\Altova\AltovaServersBeta\files - WinXP.

  C:\ProgramData\Altova\AltovaServersBeta - Windows Vista &
  Windows7
  ```

- Read permission for all paths used in the input files of the job.
- Write permission for all paths used in the output files of the job.
- Read and/or Write permission for the working directory, depending on the specific job.

Chapter 3

FlowForce Tutorial

3 FlowForce Tutorial

The aim of the tutorial is to:

- Deploy a MapForce mapping to FlowForce Server, and create and execute a job at a specific time.
- Create a subjob that copies the previously generated output files into an archive directory.
- Check a directory for new files to be passed on to a job, that uses the new file as an input file.
- Supply job parameters at runtime, to a deployed mapping, that query a database.
- Use a deployed mapping as a web service, and view the mapping results in a browser.

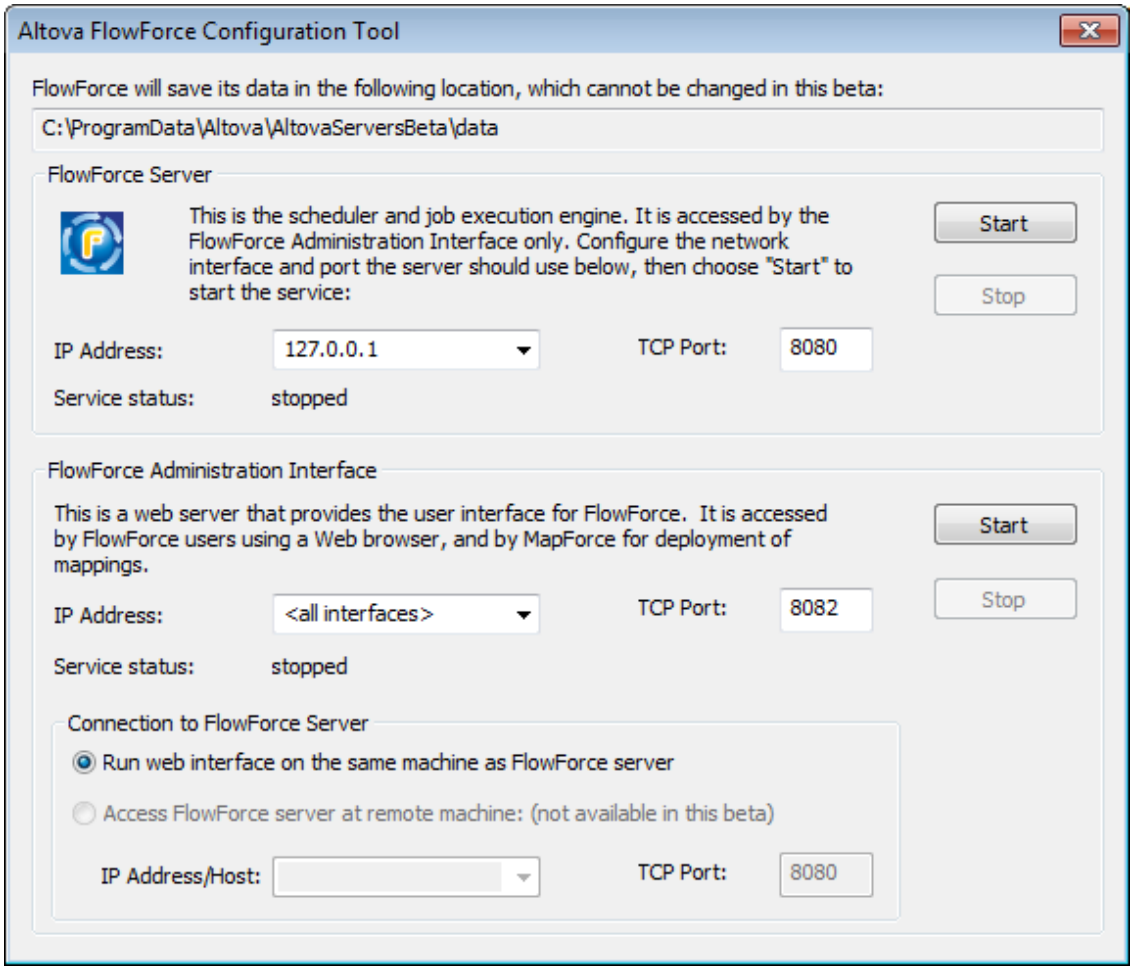
Both FlowForce Server and FlowForce Administration Interface need to be started to deploy mappings or to manage the server.

Note: When using Internet Explorer 9 as your browser, please disable the "Show friendly HTTP error messages" in the Advanced tab, to view the HTML form.

To Start FlowForce Server

If the services have not been configured and started by an administrator:

Click the Windows "**Start"** button and select **| All Programs | Altova FlowForce Beta | Altova FlowForce Configuration Tool.**



Click the "Start" button for both FlowForce Server and FlowForce Administration Interface to start these services.

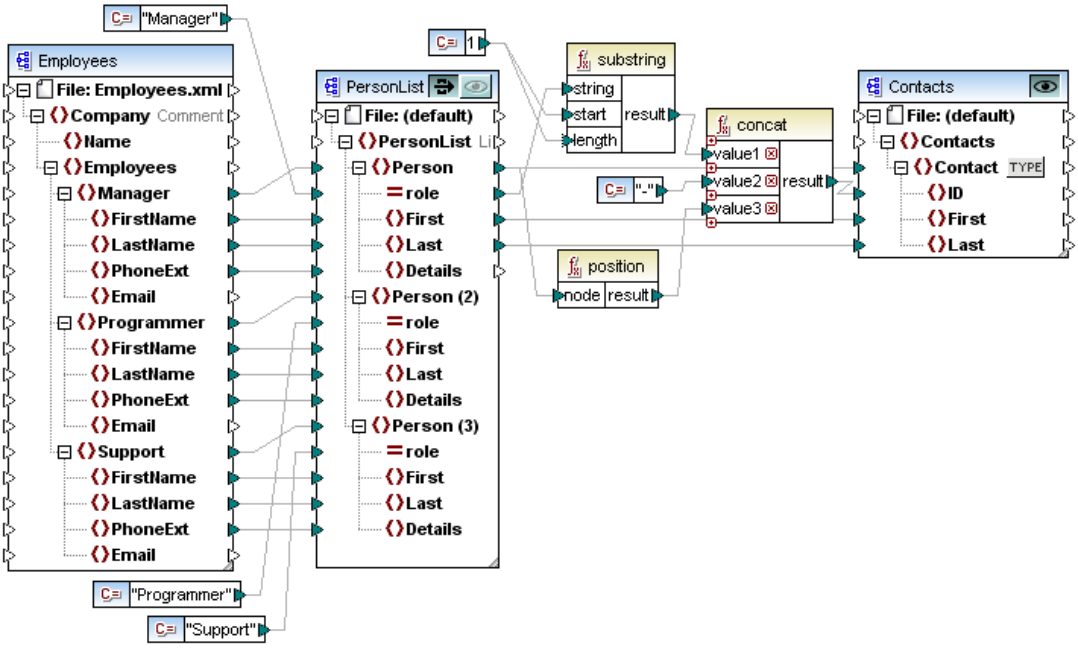
3.1 Deploying a MapForce mapping

Aim: to deploy a MapForce mapping.

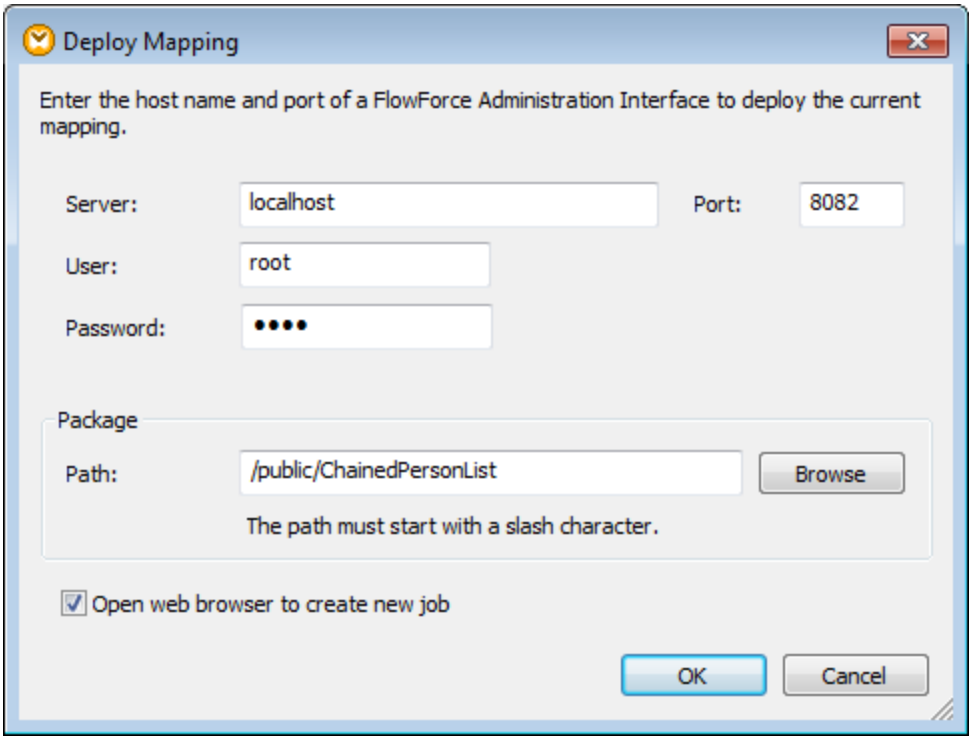
Deploying a mapping means that MapForce organizes all the mapping resources, used by the specific mapping, into an object and passes it on to the server/machine running FlowForce.

To deploy a mapping in MapForce

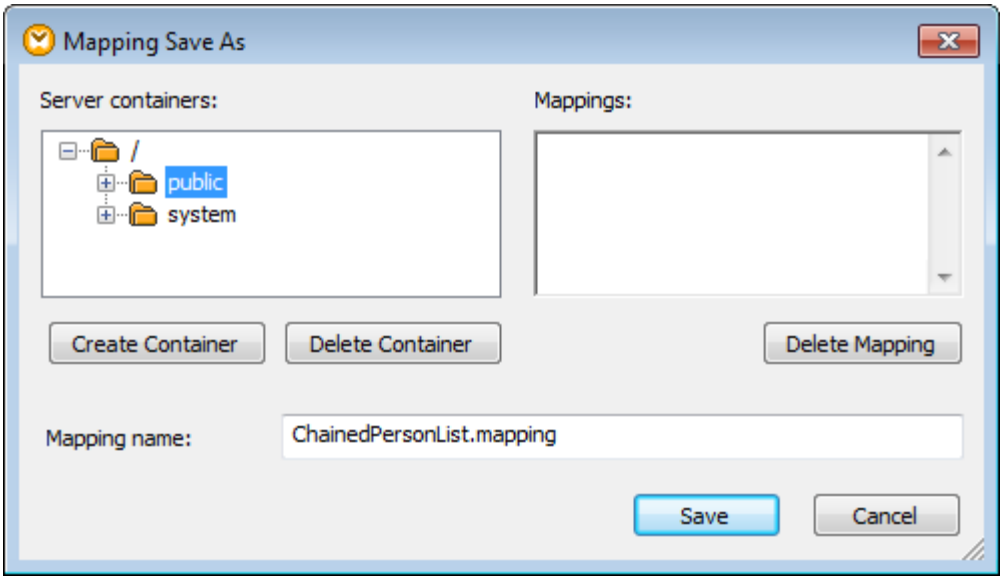
1. Open a mapping in MapForce e.g. ChainedPersonList.mfd.



2. Select the menu option **File | Deploy to FlowForce Server**.
3. Enter the Server name and Port of the web administration interface in the respective fields, e.g. localhost and 8082 if FlowForce is running on the same machine and the default port is used.
4. Enter the User Name and Password needed to access the server, e.g. "root" and "root".

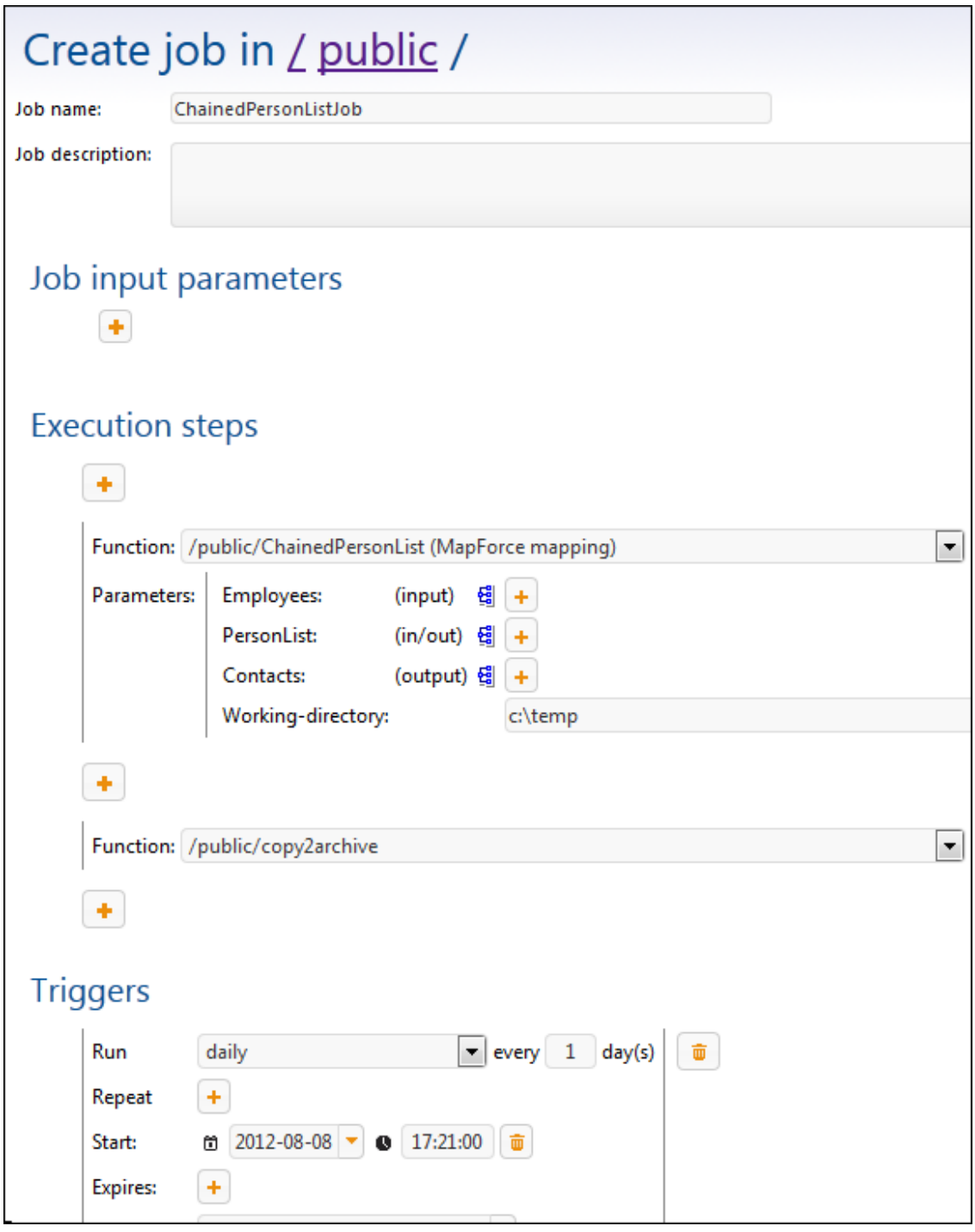


5. Optionally, click on the "Browse" button to define where the mappings are going to be placed inside the FlowForce server's object system ("public" is selected by default), then click Save.



6. Make sure the "Open web browser to create new job" check box is active.
7. Click OK to deploy.

   The messages window shows if the mapping deployed successfully. The FlowForce Administration Interface is automatically opened in your web browser and a partially filled in job page is displayed.



The next thing to do is to define the rest of the job, i.e. the Job Triggers, the Job Credentials, and the Execution Steps.

3.2 Defining a job - triggers & execution steps

Aim: To define a simple job that:

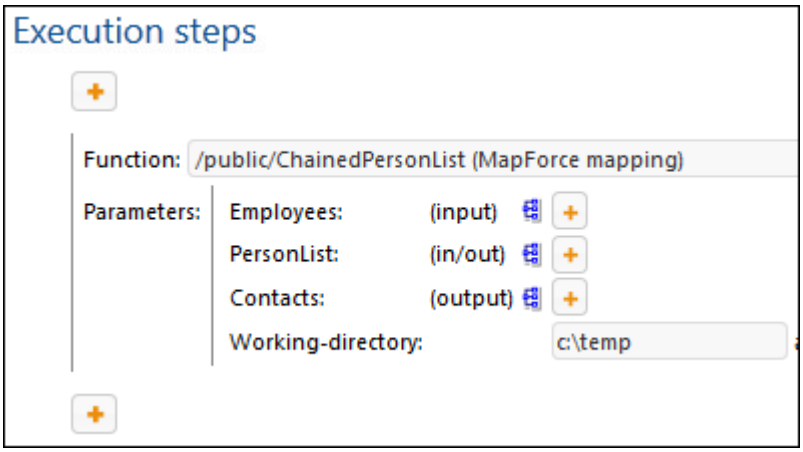
- Uses the deployed mapping function(s) from MapForce
- Uses a manual, or predefined credential: see Administration Guide
- Triggers the job at a specific time

As this mapping was deployed from MapForce and the "Open web browser..." check box was activated, you do not have to navigate to the Jobs page, it is automatically created for you.

To define the job execution steps:

Some of the fields of the Execution steps group have been filled out automatically. (You would normally have to click the "+" button to add a new Execution step.)

Function is: /public/ChainedPersonList.

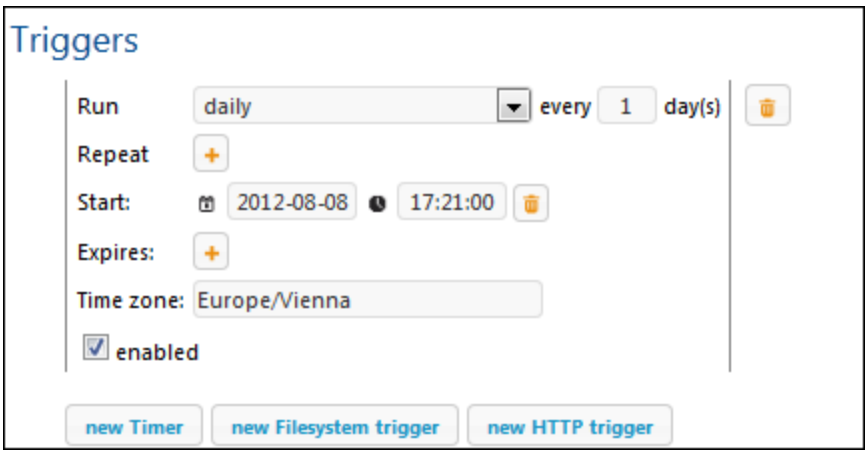


1. Click the "+" button of the working directory entry to enter a different directory, e.g. c:\temp. Note that this must be a path on the server machine (that runs FlowForce), not on your local machine.
   As we do not want to **override** any of the parameter settings defined by the deployed mapping, we are not going to change any other Parameter settings.
2. Click the "Save" button to save the job.

   As soon as the trigger time is reached, the job is executed and the output files, generated by the mapping, appear in the c:\temp directory. The output files are PersonList.xml and Contacts.xml.

To define the trigger:

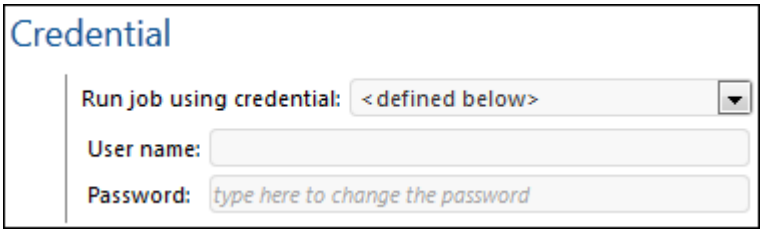
1. Click the "new Timer" button.
2. Click into the (Start) date field and select the start date from the date picker.
3. Enter the time the job is to be triggered. Note that the time is entered in 24 hour format. For testing purposes, use a time close to your current time.



To define the credentials:

There are two ways that you can define job credentials:

1. In the Credential group, click the "Run job using credential" combo box and select a previously defined credential e.g. "Cred_production".
2. Manually enter you personal server credentials in the User name and Password fields.

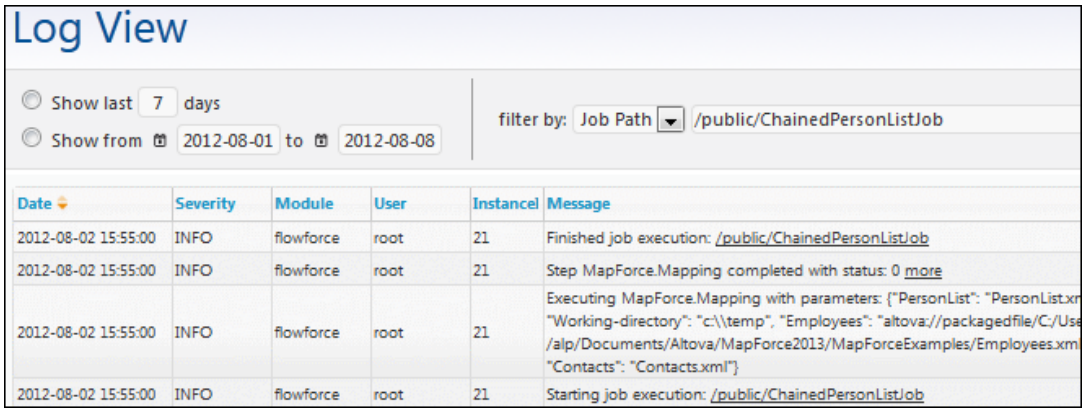


Entering credentials manually forces you to update the credentials here if your server login changes.

Viewing the job log

- Click the View log button near the top of the left of the Job page, to open the Log View filtered for the current job.

  The execution parameters and the execution status are displayed in the Log View table.

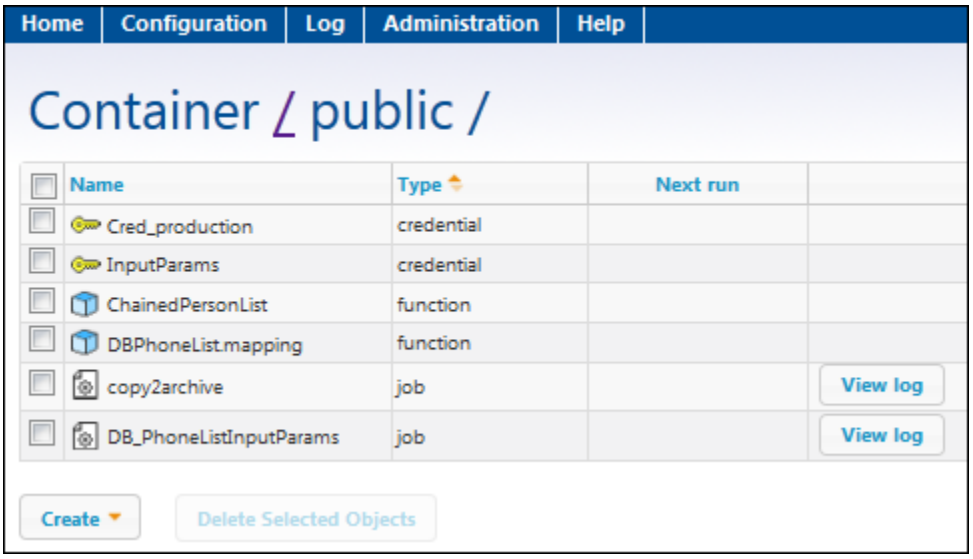


Viewing the defined job

1. Click the ChainedPersonListJob **link** in the Message column.
   This opens the previously executed job definition page.

Notes:

Click the "Configuration" button to see the various containers. Click a folder e.g. public, to see its contents. To see the root container contents, click "/".



Each container object, i.e. function, job, credential, etc., has a check box to the left of its name to select it. The check box to the left of the column header "Name", selects/ deselects **all** objects in the list.

The "Create" button lets you create Containers, Jobs, and Credentials.

The "Delete Selected Objects" button becomes active when objects have been selected in the list; clicking it deletes the selected objects.

See: Defining a subjob

3.3 Defining a subjob

Aim: To create and integrate a separate job that:

- Copies the mapping result files of the previous job into an archive directory.

  This job will act as a subjob to the calling job "ChainedPersonListJob".

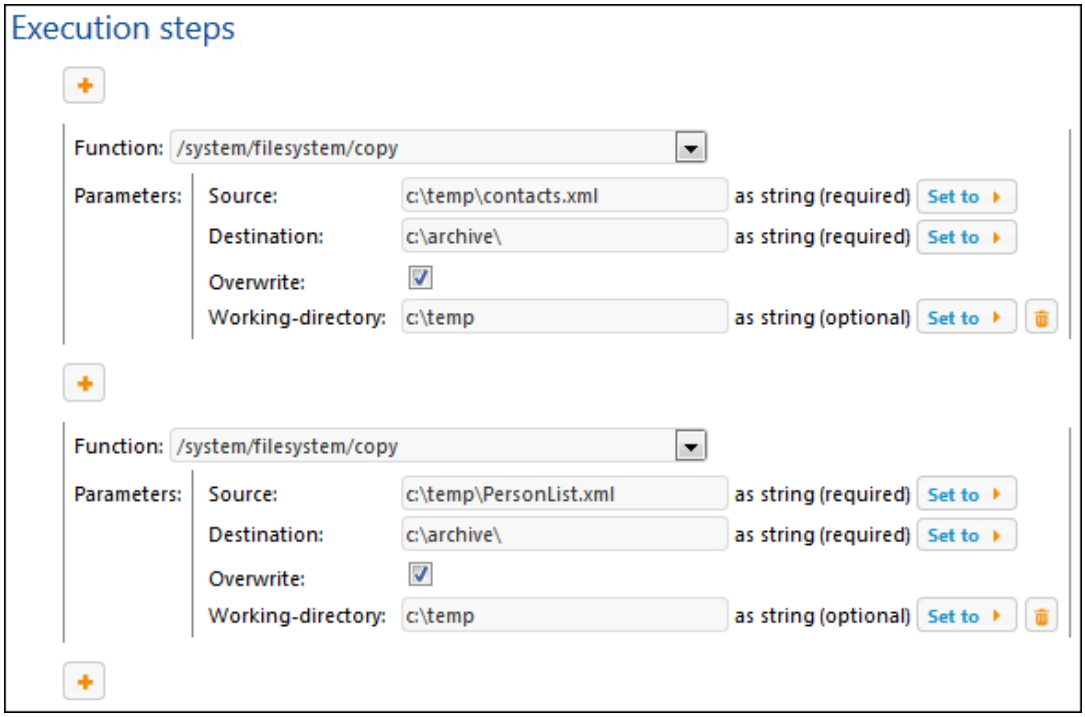
To create a new job:

1. Click the "Configuration" button, then click the container "public".
2. Click the "Create" button and select "Create Job" from the popup menu.
3. Enter the name of the job, e.g. "copy2archive".
   There is no need to define a trigger for this job, as it will be called by another job.

   Note: there is no need to define a new credential, as the credential of the calling job is always used.

Defining the subjob execution steps:

1. In the Function field, select /system/filesystem/copy.
2. Enter the name of the file that you want to copy in the Source field, e.g. c:\temp\contacts.xml.
3. Enter the name of the destination directory, e.g. c:\archive. (Click the "+" button next to the Overwrite field if you want to be able to overwrite the same file at the destination.)
4. Enter the name of the working directory, e.g. c:\temp.
5. Click the "+" button under the previously defined Execution step to add a second one. Source c:\temp\PersonList.xml, destination c:\archive, and working directory, c:\temp.
7. Click "Save" to save the job.

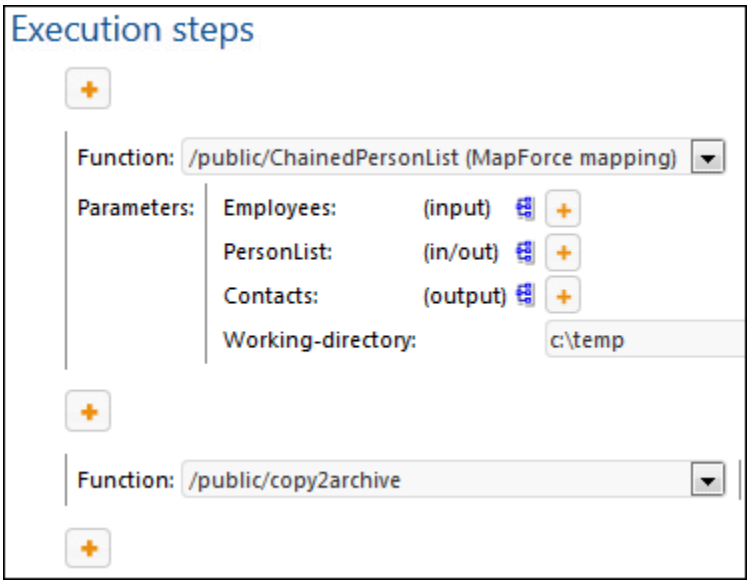


Calling a job from another job:

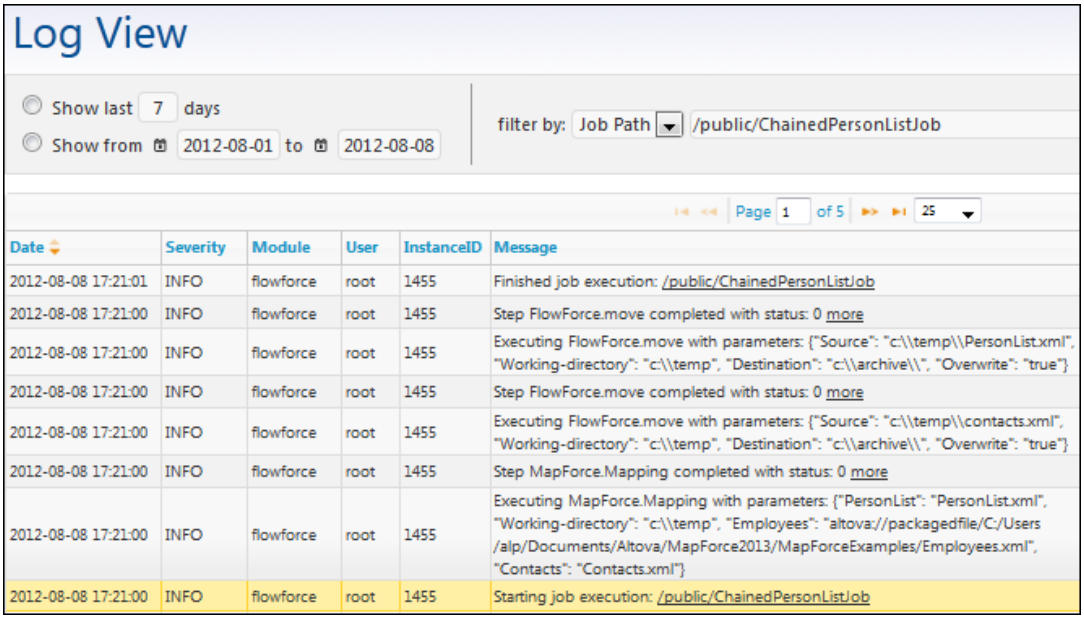
1. Click the "Configuration" button, then the public container, and select ChainedPersonListJob.
2. Scroll down to the Execution steps group and click the "+" button to add a new

execution step.

3. Click the Function combo box and select /public/copy2archive.



4. Update the timer trigger and click the "Save" button.
5. After the job has been triggered, click the "View Log" button at the top of the job page.



Date	Severity	Module	User	InstanceID	Message
2012-08-08 17:21:01	INFO	flowforce	root	1455	Finished job execution: /public/ChainedPersonListJob
2012-08-08 17:21:00	INFO	flowforce	root	1455	Step FlowForce.move completed with status: 0 more
2012-08-08 17:21:00	INFO	flowforce	root	1455	Executing FlowForce.move with parameters: {"Source": "c:\\temp\\PersonList.xml", "Working-directory": "c:\\temp", "Destination": "c:\\archive\\", "Overwrite": "true"}
2012-08-08 17:21:00	INFO	flowforce	root	1455	Step FlowForce.move completed with status: 0 more
2012-08-08 17:21:00	INFO	flowforce	root	1455	Executing FlowForce.move with parameters: {"Source": "c:\\temp\\contacts.xml", "Working-directory": "c:\\temp", "Destination": "c:\\archive\\", "Overwrite": "true"}
2012-08-08 17:21:00	INFO	flowforce	root	1455	Step MapForce.Mapping completed with status: 0 more
2012-08-08 17:21:00	INFO	flowforce	root	1455	Executing MapForce.Mapping with parameters: {"PersonList": "PersonList.xml", "Working-directory": "c:\\temp", "Employees": "altova://packagedfile/C:/Users/alp/Documents/Altova/MapForce2013/MapForceExamples/Employees.xml", "Contacts": "Contacts.xml"}
2012-08-08 17:21:00	INFO	flowforce	root	1455	Starting job execution: /public/ChainedPersonListJob

You can now see the status of the job and its subjob.

The two XML files generated by the first job have been copied to the archive directory.

Note:

If you want to rename a file when it is copied, enter the new file name in the "Destination" field.

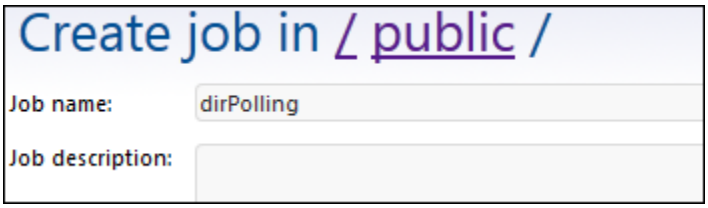
See: Directory polling - acting on a trigger file

3.4 Directory polling - acting on a trigger file

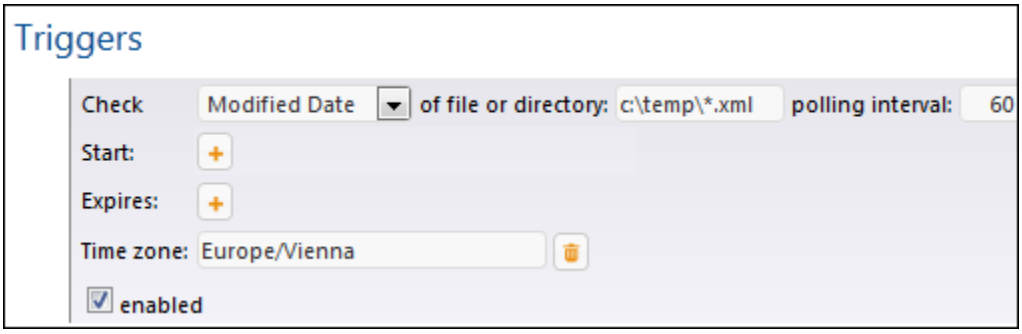
Aim: to check a directory for new XML files, execute the deployed mapping with those files, and copy the result files into an archive directory.

Deploying the mapping and creating the job:

1. Open the ShortApplicationInfo.mfd mapping in MapForce.
2. Select the menu option **File | Deploy to FlowForce Server**.
3. Enter the password in the Password field; make sure that the "Open web browser..." check box is active, then click OK.
   This generates a new job in the public directory of FlowForce.
4. Enter the name the job e.g. dirPolling.



5. Click the "new File System" trigger button.
6. Enter the directory name and file types that you want to check for, e.g. c:\temp*.xml.
7. Enter the polling interval, e.g. 60 seconds.

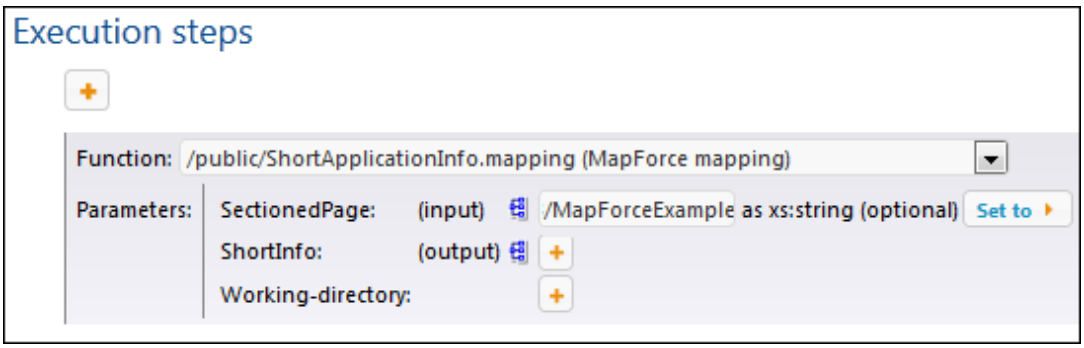


8. Select the credentials you want to use for this job, e.g. Cred_production.

Note:
When you select "File system trigger", the "triggerfile" entry is automatically added to the "Job input parameters" group.

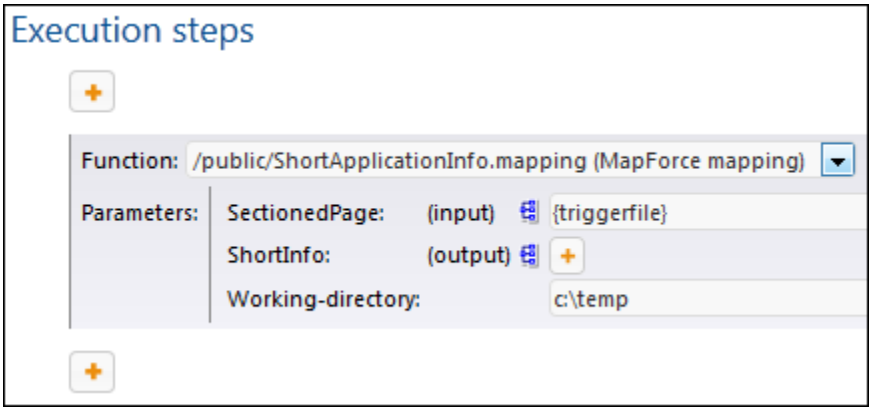
In the Execution steps group, click the "+" icon to add a new step, then:

1. Select "/public/ShortApplicationInfo.mapping" from the Function combo box to use the previously defined mapping.
2. Click the "+" icon to the right of the SectionedPage label.



This creates an edit field.

3. Click the Set to button to the right of the expanded SectionedPage field, and select "triggerfile"
4. This causes the input field contents to change to {triggerfile}.
5. Click the Working directory "+" button and enter the working directory e.g. c:\temp.

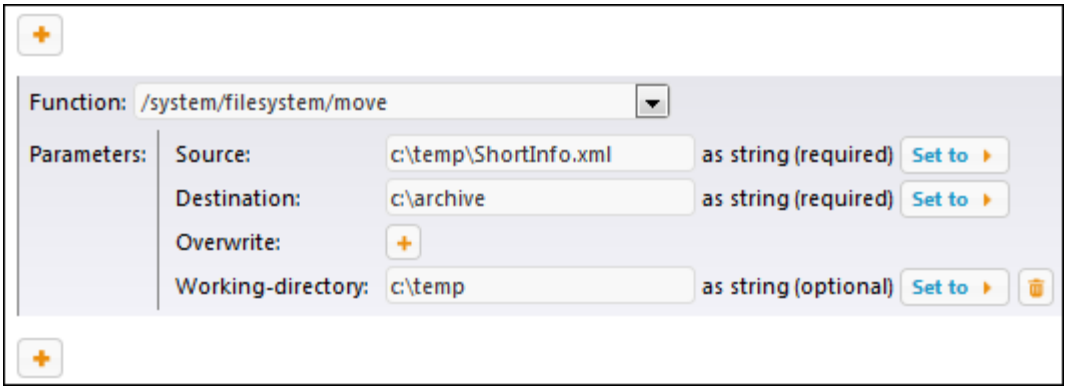


The file in the directory being polled, will now be used as the input file in Parameters group of the execution step.

Note that the name of the (output) file is also shown as a parameter in this group, i.e. "ShortInfo".

Adding a second job execution step:

1. Click the "+" button below the execution step that was just created.
2. Use the Function combo box to select "/system/filesystem/move".
3. In the Source field, enter the path/file name of file that you want to move, e.g. ShortInfo.xml
4. Enter the location of the archive directory, e.g. c:\archive.
5. Enter the working directory c:\temp.



6. Click the "Save" button to save the job.

   As soon as the trigger start time has been reached, the trigger will be active and c:\temp folder will be polled every 60 seconds.

To start the dirPolling job:

- Navigate to your ...\Altova\MapForceExamples folder and copy the **ApplicationsPage.xml** file to the c:\temp folder.

Result:

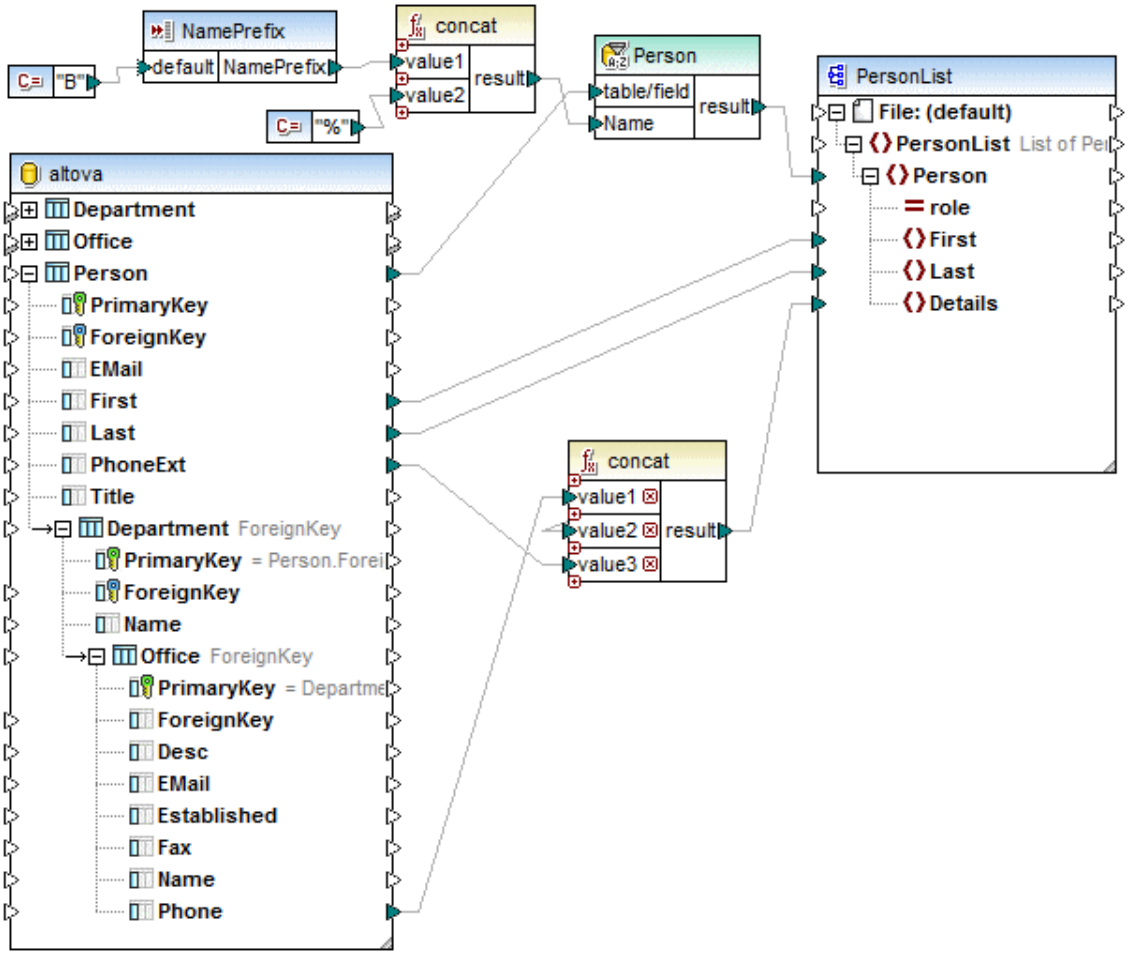
- As soon as a new XML file is found, that XML file becomes the input file for the deployed mapping.
- The job is started and the result of the processed input file (ShortInfo.xml) is moved to the archive directory.

See: Using parameters to query a database

3.5 Using parameters to query a database

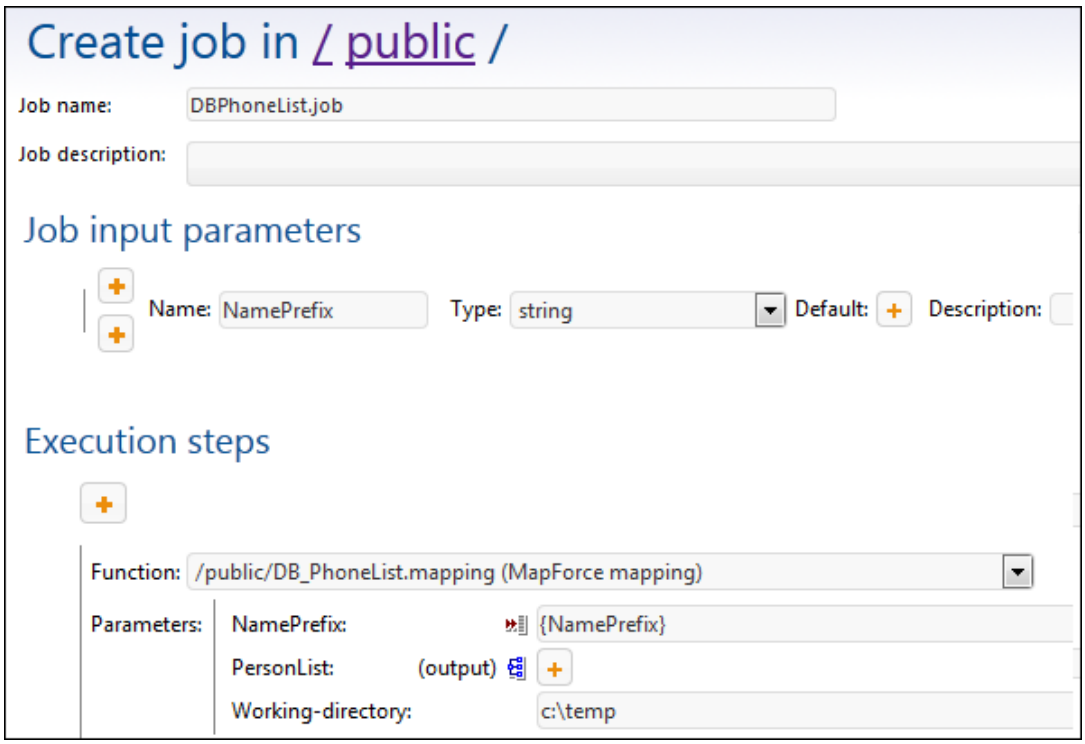
Aim: to query a database using job input parameters, via a web browser.

- This example uses the DB_PhoneList.mfd mapping available in the ... \MapForceExamples folder.
- The NamePrefix input parameter of the mapping, will be used to supply the query data in the browser client.

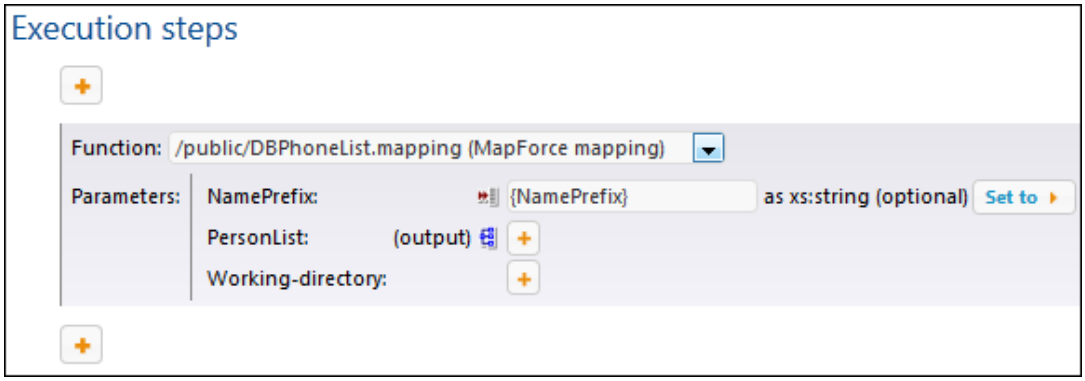


Deploying the mapping and creating the job:

1. Open the DB_PhoneList.mfd mapping in MapForce.
2. Select the menu option **File | Deploy to FlowForce Server**.
3. Enter the password (root) in the Password field; make sure that the "Open web browser..." check box is active, then click OK.
   This generates a new job in the public directory of FlowForce.
4. Enter the name of the job e.g. DBPhoneList.
5. Click the check box in the Service group and enter **myURL** in the text box.
6. Click the "+" button of Job input parameters and enter NamePrefix.
7. Select the credential you want to use for this job, e.g. cred_production.



8. Click the "+" button to the right of the NamePrefix entry in the Execution steps group.
9. Click the "Set to" button and select NamePrefix (the parameter name is automatically available).

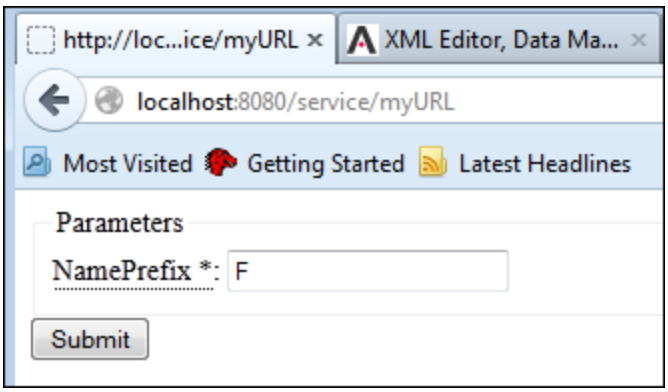


10. Click the "Save" button to save the job.

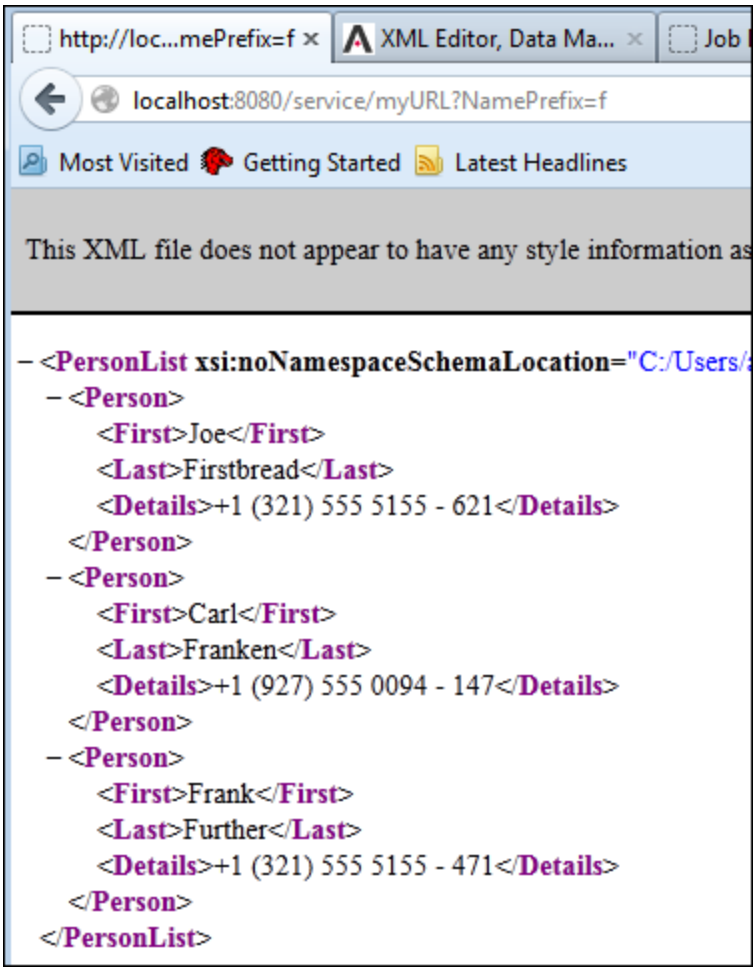
Note: No triggers were defined because the web browser is used to access the service.

Using the browser to run the job and query the database:

1. Open your browser and enter http://localhost:8080/service/myURL in the URL text box. If you changed the port number for FlowForce server, please use that one.
2. Enter the letter of the last name of the person(s) you are looking for, e.g. "F".



3. Click the "Submit" button to send the query to the server.



The resulting XML file is now displayed in the browser.

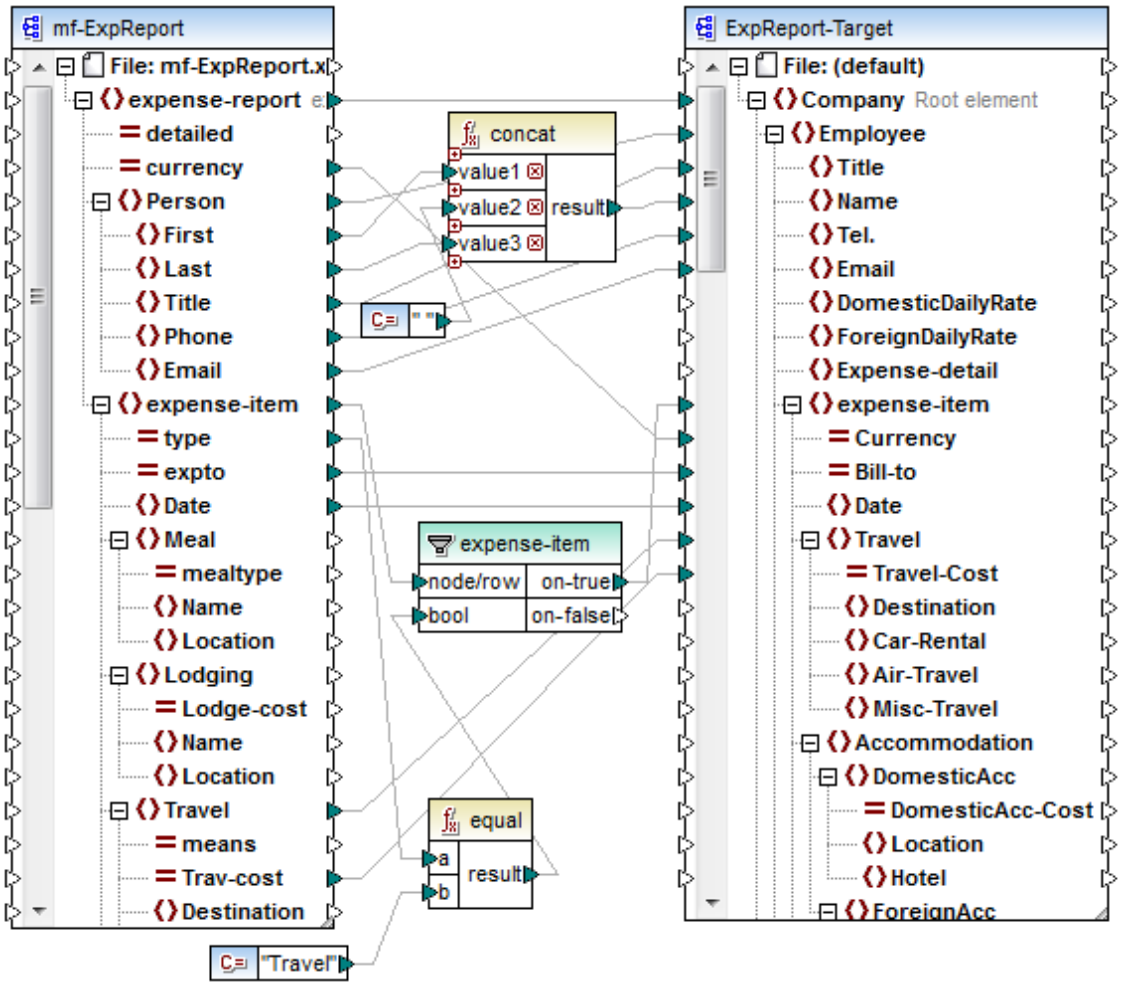
Please note:
You can press the "Back" button and reenter a query for a different Last name.

See: Using a deployed mapping as a web service

3.6 Using a deployed mapping as a web service

Aim: to supply various XML files as source files, and view the mapping result via a browser client.

- This example uses the Tut-ExpReport.mfd file available in the ... \MapForceExamples\Tutorial folder.
- The XML source file, mf-ExpReport.xml, will be replaced at runtime in the browser client.

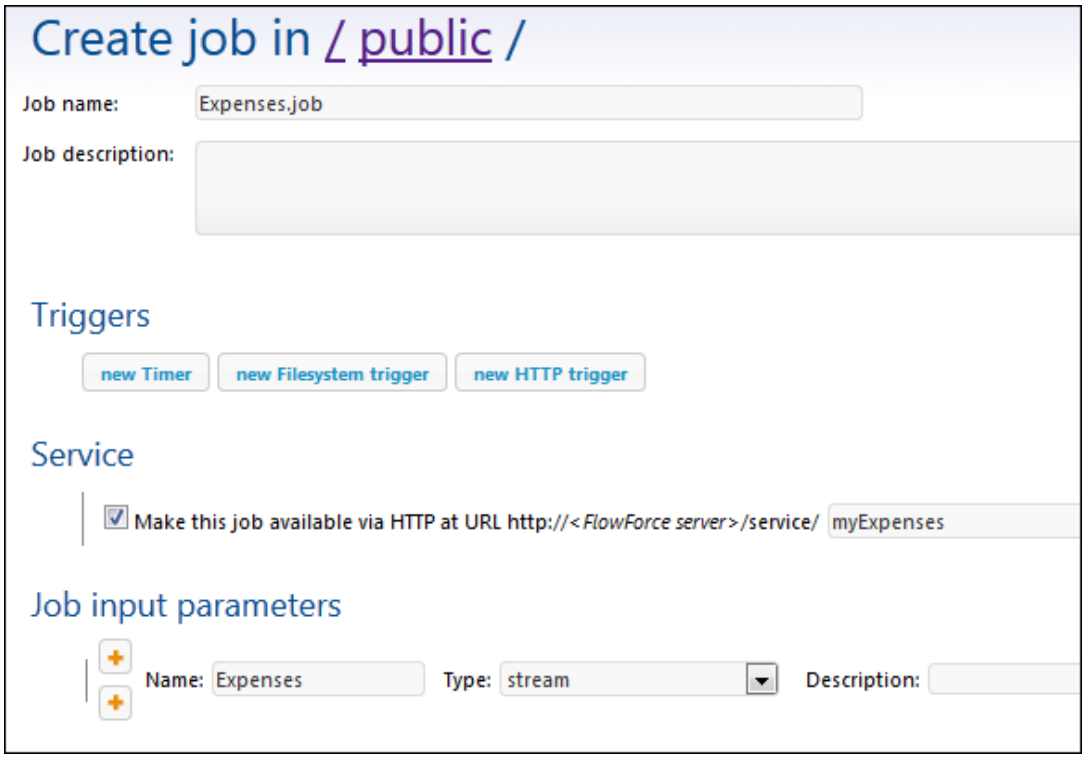


Deploying the mapping and creating the job:

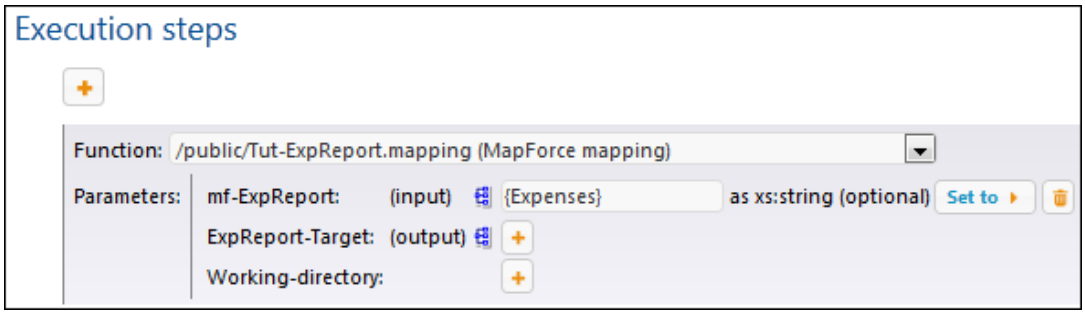
1. Open the Tut-ExpReport.mfd mapping in MapForce.
2. Select the menu option **File | Deploy to FlowForce Server**.
3. Enter the password (root) in the Password field; make sure that the "Open web browser..." check box is active, then click OK.
   This generates a new job in the /public directory of FlowForce.

4. Enter the name of the job e.g. Expenses.
5. Click the check box in the Service group and enter **myExpenses** in the text box.
6. Click the "+" button of Job input parameters and enter Expenses.
7. Click the "Type" combo box of Job input parameters and select **stream**. This defines the parameter to represent a file uploaded to the server with the HTTP POST request.

8. Select the credential you want to use for this job.



9. Click the "+" button to the right of the mf-ExpReport entry in the Execution steps group.
10. Click the "Set to" button and select Expenses (the parameter name is automatically available).

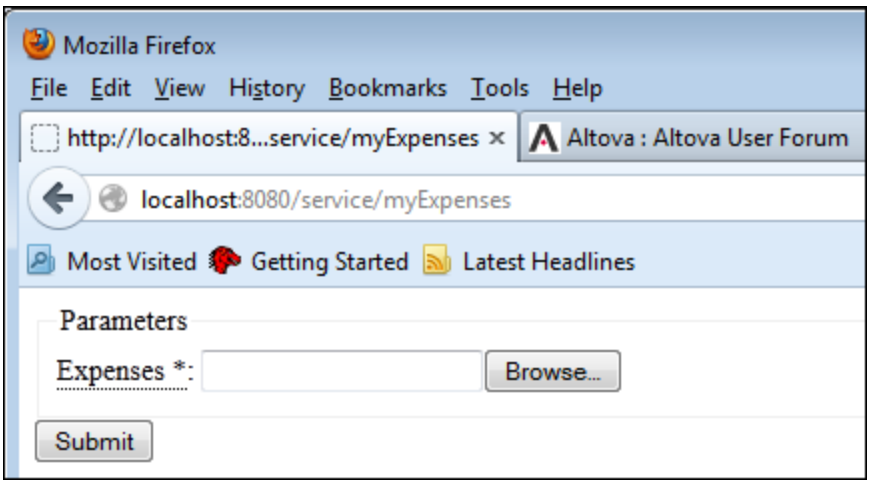


11. Click the "Save" button to save the job.

Note: No triggers were defined because the browser is used to access the service.

Using the browser to run the job and choose different XML input files:

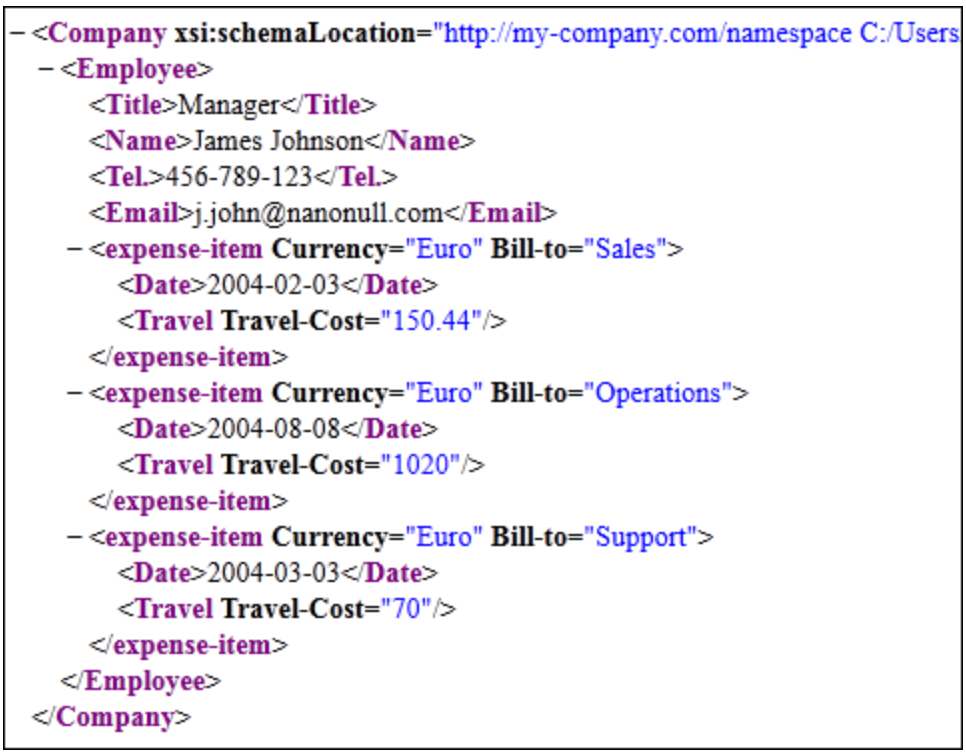
1. Open your browser and enter http://localhost:8080/service/myExpenses in the URL text box.
2. Click the "Browse" button.



3. Select mf-ExpReport.xml in the ...\MapForceExamples\Tutorial folder from the dialog box, and click "Submit".

```
- <Company xsi:schemaLocation="http://my-company.com/namespace C:/Users
  - <Employee>
      <Title>Project Manager</Title>
      <Name>Fred Landis</Name>
      <Tel.>123-456-78</Tel.>
      <Email>f.landis@nanonull.com</Email>
    - <expense-item Currency="USD" Bill-to="Development">
        <Date>2003-01-02</Date>
        <Travel Travel-Cost="337.88"/>
      </expense-item>
    - <expense-item Currency="USD" Bill-to="Accounting">
        <Date>2003-07-07</Date>
        <Travel Travel-Cost="1014.22"/>
      </expense-item>
    - <expense-item Currency="USD" Bill-to="Marketing">
        <Date>2003-02-02</Date>
        <Travel Travel-Cost="2000"/>
      </expense-item>
    </Employee>
</Company>
```

4. Click the browser "Back" button, then the myExpenses link, and click the "Browse" button.
5. Select mf-ExpReport2.xml from the dialog box, and click "Submit".



Two completely different Expense reports were processed by the deployed mapping and output to the browser.

Note:
The path of the XML output instance file is saved with the deployed mapping. There is therefore no need to supply the working directory.

Chapter 4

User Guide

4 User Guide

Object System
Jobs, functions, triggers, etc. are stored in the object system inside the FlowForce Server database in a hierarchical structure.

The properties and capabilities of the FlowForce Server object system are similar to those of commonly used file systems. File systems use folders, while FlowForce Server uses containers.

Containers can have access permissions assigned to them, or inherit permissions from their parent container(s).

- The root of the object system is the "**/**" container (the **root container**) which can contain other containers, or other user-defined objects.
- The predefined container "**/system**" contains the system functions(s) and should not be used for user-defined objects.
- The predefined container "**/public**" is the default location to create user-defined objects like jobs, functions, credentials and other containers.

Deploying mappings
Deploying a mapping means MapForce organizes all the mapping resources, used by the specific mapping, into a FlowForce function and passes it on to the server/machine running FlowForce, where it will be processed.

A deployed mapping function can then be used in a job execution step.

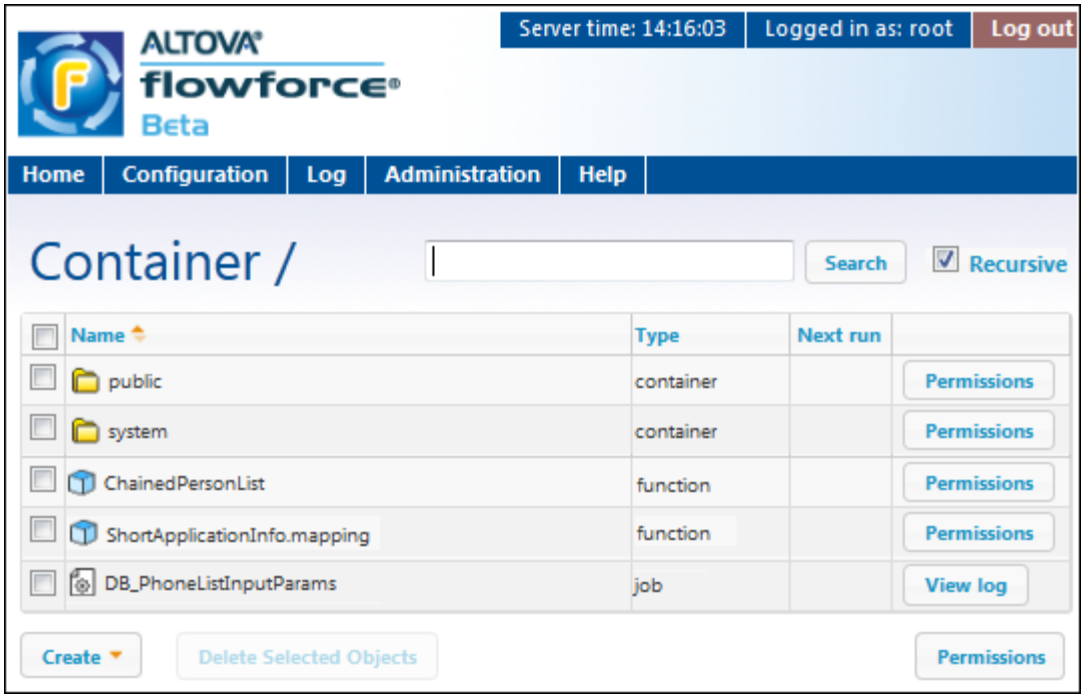
Jobs
Jobs consist of triggers, execution steps and various other settings. The triggers define when the job will be executed, and the execution steps define what the job actually does when it executes. Multiple triggers and execution steps can be defined per job.

Jobs can call other jobs allowing you to create subjobs.

Jobs can contain placeholder values that can be passed to the job at runtime. These placeholders are called "Job input values" and can supply values through a default, or through manual input via an HTTP client, e.g. internet browser.

4.1 FlowForce Administration Interface

The FlowForce Administration Interface is used to define jobs and triggers and to display the log table.



There are several menu items available in the browser window:

- **Home** shows you the connection details and any active jobs.
- **Configuration** shows containers and the objects they contain: jobs, credentials, functions, etc.
- **Log** shows you the server logs. The Log View contents can be configured.
- **Administration** shows you the Users, Roles, and Settings
- **Help** shows this help file

Containers
FlowForce has a hierarchy of "items" visible in the Configuration page.

- The top-level is the "Container". Containers can contain "objects".
- Objects can be other Containers, Jobs, Instances, or Functions. Clicking the "Name" check box selects/deselects all the objects in the list.
- Clicking a container displays the objects that it contains.
- Two default containers exist in the list: the public and system containers.
- Clicking a job displays the job definition page, showing the triggers, execution steps, and other settings that make up the job.

Function
A function is one or more executable steps of a mapping deployment, or an operating system function.

Job

A Job consists of Triggers and Execution steps. The triggers define when the job will be executed, and the execution steps define what the job actually does when it executes. Multiple triggers and execution steps can be defined per job.

4.2 FlowForce concepts

Configuration

Configuration data in FlowForce server's database are comprised of various objects that define the operation of FlowForce. This includes jobs, credentials, functions, triggers, and other objects.

Configuration objects are organized in a freely defined hierarchy of containers. Some configuration settings are edited together (e.g. jobs include triggers), and other settings can also be stored as standalone objects under their own name (e.g. credentials and functions).

Container
A container is similar to a folder in a commonly used file system. It is used to create a hierarchical structure for storing configuration objects and other containers. Containers can be assigned access permissions.

Two predefined containers exist in FlowForce: /system which contains system functions, e.g. copy, move, etc., and /public which is the default container when deploying a mapping to FlowForce from MapForce. Other containers can be created as needed, e.g. for departments or user groups.

Function
A FlowForce function performs a specific operation when used in a job execution step. It may have input parameters that need to be passed to it by the caller. Available functions include the system functions delivered with FlowForce, deployed MapForce mappings, and the execution steps of other jobs.

Job
A Job consists of Triggers, Execution steps, input parameters, and other settings. Triggers define when a job will be executed, and the execution steps define what the job actually does when it executes. Multiple triggers and execution steps can be defined per job.

Trigger
Triggers define under which circumstances a job will be executed. Three types of triggers can currently be defined: Timer triggers, File system triggers, and HTTP triggers. Multiple triggers can be defined per job.

Service
FlowForce permits exposing jobs as web services via the HTTP protocol. This allows interactive or automated access to these jobs.

Credential
Credentials are stored login data used to execute FlowForce jobs. Credentials can be defined as standalone "objects" and be assigned to various jobs, or they can be manually entered for a specific job.

Queue
The queue settings in a FlowForce job allow limiting the number of parallel job executions to control use of server resources.

Access Control

All important operations in FlowForce are linked to permissions or privileges which need to be assigned to the user to successfully execute them.

User

FlowForce users are persons that have been added to FlowForce by the FlowForce administrator with a login name and a password. Depending on the assigned rights and privileges, users can define FlowForce jobs, deploy mappings, or view logs.

Two special users are predefined by FlowForce: "root" is the initial administrator user, and "anonymous" is a special user account used for FlowForce services that should be available to users without explicit log in to FlowForce.

Role
Roles are used to manage privileges and object permissions for user groups as opposed to individual users.

Having defined users, you can assign them to a role thus creating user groups. The users become "members" assigned to the specific role.

Permission
Permissions control access to containers and configurations. Unlike privileges they can be redefined on every level of the container hierarchy, and are by default inherited from parent containers.

Permissions, like privileges, are inherited from all roles the user is a member of, as well as from permissions directly assigned to the user.

Privilege
Privileges control user rights globally. This means privilege settings cannot be overridden in the container hierarchy of FlowForce.

When a user logs into FlowForce, the set of effective privileges is determined by the user privileges and all role privileges the user is member of.

4.3 Job input parameters

Job input parameters are placeholders for values (or files) that can be supplied at runtime.

Creating an input parameter, e.g. NamePrefix, automatically makes it available for selection in the "Set to" popup window, of any of the execution step parameters.

For an example please see: Using a deployed mapping as a web service, or Using parameters to query a database.

When adding a file system or HTTP trigger to a job, FlowForce automatically creates a parameter named "triggerfile" that contains the name of the file that activated the trigger.

Type: string
Use the string type for most parameters that you use.

Type: stream
Allows you to select files when using the job as a service.

Default
Allows you to specify a default value for the parameter that is used when no value is passed to the job at runtime.

Job parameter values are determined at runtime as follows:

- File system and HTTP triggers set the parameter "triggerfile".
- If a job is called as service via HTTP, all parameters are passed from the HTTP request. Please see the Service section for more information.
- If a job is called from an execution step in another job, parameters are passed from the step definition in the calling job.
- Parameters that are not supplied any value get the default value defined in the "Default" field.

4.4 Job execution steps

Steps define what actions the job executes when it is started by a trigger or an HTTP request.

Function:
The Function combo box displays a list of available FlowForce functions.

The following functions are available:

- Any MapForce mapping can be deployed to FlowForce from the MapForce graphical environment. This allows you to use all the data transformation features of MapForce, including XML, EDI, databases, ...
- Built-in functions in the "/system" container include copying or moving files, creating directories, or executing arbitrary command lines.
- Any other job defined in FlowForce can be used as subjob.

  Note that all file paths in job execution steps must be a path on the server machine (that runs FlowForce), not on your local machine.

Function - System commands

/system/shell/commandline - execute any command line
Parameters:

Command	Enter any command line to execute, e.g. batch files or other executables.
Working-directory	Enter a working directory, e.g. C:\Temp. If this is empty, the temporary directory is used. Note that this must be a path on the server machine (that runs FlowForce), not on your local machine.

/system/filesystem/mkdir - create directory
Parameters:

Path	Enter the path/location of the new directory
MakeParents	Clicking generates a check box. Activation allows a hierarchical path to be created in one step. E.g. working directory is c:\temp, and path is temp2\temp3. Creates the new directory c:\temp\temp2\temp3.
Working-directory	Enter a working directory, e.g. C:\Temp. If this is empty, the temporary directory is used.

/system/filesystem/copy - copy file(s)
Parameters:

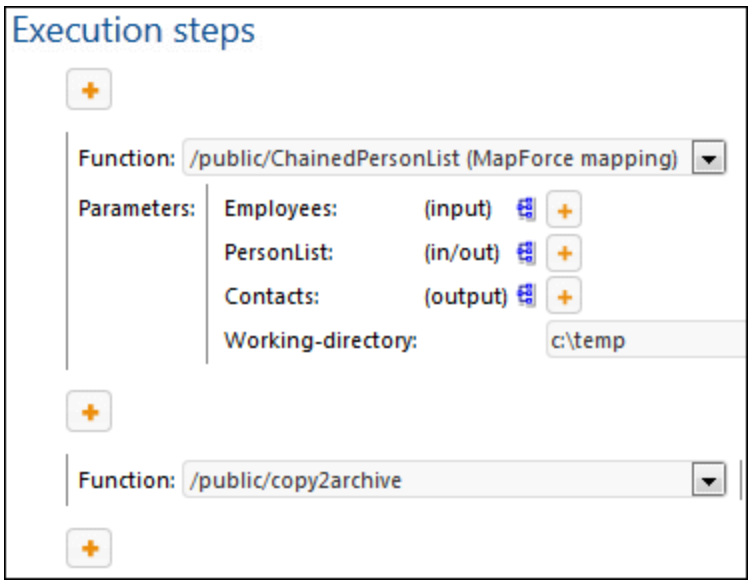
Source	Enter the path and file name of the source file that you want to copy.
Destination	Enter the path and file name of the destination directory. You can enter a different file name in the destination field if you want to rename it as well.
Overwrite	Clicking generates a check box. Activation causes destination file to be overwritten
Working-directory	Enter a working directory, e.g. C:\Temp. If this is empty, the temporary directory is used.

/system/filesystem/move - move or rename file(s)
Parameters:

Source	Enter the path and file name of the source file that you want to move.
Destination	Enter the path and file name of the destination directory. If you only supply the directory name in this field, then the original file name will be retained.
Overwrite	Clicking generates a check box. Activation causes destination file to be overwritten.
Working-directory	Enter a working directory, e.g. C:\Temp. If this is empty, the temporary directory is used.

Note - Working directory:
This entry must be a path on the server machine (that runs FlowForce), not on your local machine.

Function - Subjob
Once a job has been defined it can be used in an execution step of another job as a subjob. In the screen shot below, the subjob is added as an extra execution step, by clicking the Function combo box and selecting /public/copy2archive. If the selected subjob has job input parameters, they appear below the function and can be filled with values.



Function - MapForce mapping

Parameters:
Parameters for MapForce actions are defined by the specific mappings.

Parameters are defined by:

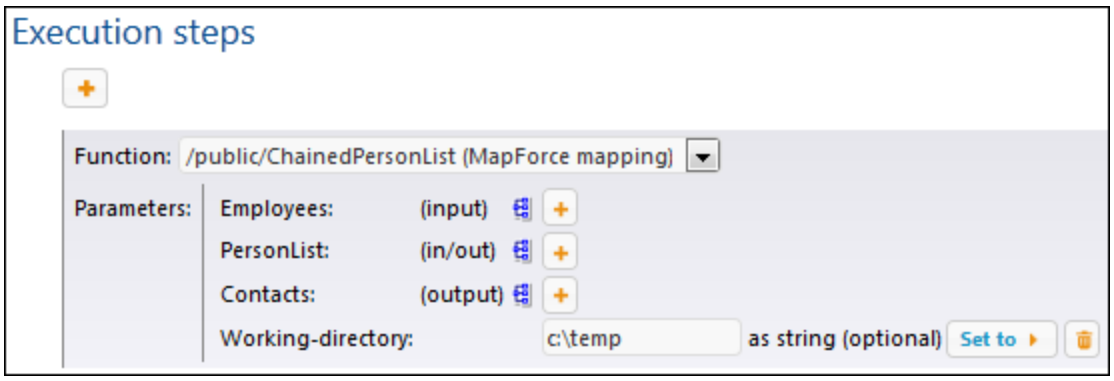
- input components
- in/out components
- output components

Parameters allow you to **override** the input and output files that were defined when the mapping was deployed from MapForce. This means that you can replace the those files by any others when you are defining the job execution steps.

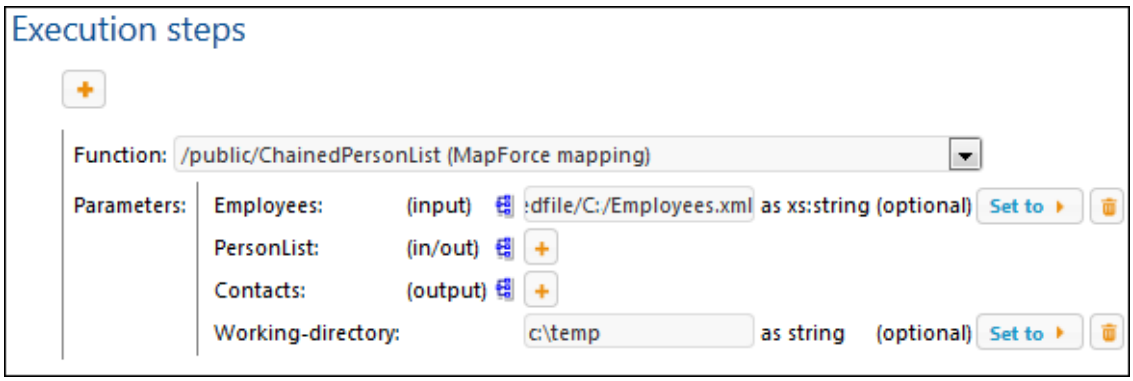
When the job executes, the new files will be used instead of the ones defined in the mapping.

E.g.
Using the deployed ChainedPersonList as an example:

Employees (input)	is the source component with Employees.xml as the instance document.
PersonList (in/out)	is the intermediate document and therefore shown as in/out, because it is used as both a source and target document in the chained mapping.
Contacts (output)	is the target document

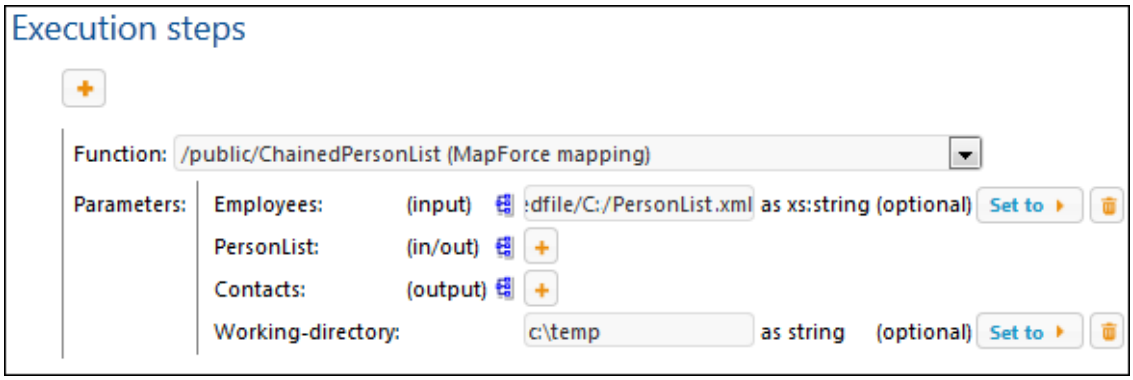


To change the source/destination files:
Click the "+" button next to the list of parameters to expand the optional fields.



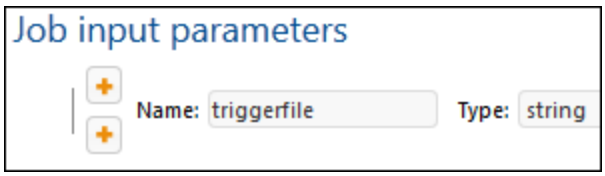
Click in the Employees (input) field and delete the "altova://packagedfile/C:/Documents and Settings.../Employees.xml" file and replace it with the file you want to use instead (e.g.

PersonList.xml).



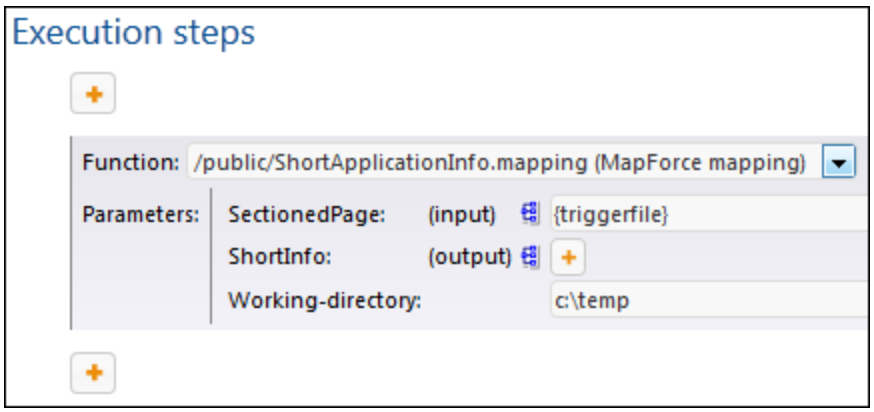
Note:
Any path starting with "**altova://packagedfile/**" refers to the file content that was **deployed** together with the mapping, and not to the current version of that file in any path on the server.

Acting on files that cause the trigger to fire
If you create a "File system trigger" this automatically adds the "triggerfile" entry into the Job input parameters field.



Select the deployed mapping file you want to use in the Function field.
Clicking the "+" icon to the right of SectionedPage creates an edit field.

Click the Set to button to the right of the expanded SectionedPage field, and select "triggerfile" entry.



This causes the input field contents to change to {triggerfile}.
The file, in the directory being polled, will then be used as the input file for the execution step.

Note:
Using a file that caused the trigger to fire, does not work with time-based triggers.

Please see: **Directory change - act on trigger file**

4.5 Triggers

Three types of triggers can currently be defined: Timer, File system triggers, and HTTP triggers.

Timer trigger
This type of trigger allows you to schedule your jobs. Timer triggers have a Start date/time, Expire date/time and period of recurrence (daily, weekly etc.).

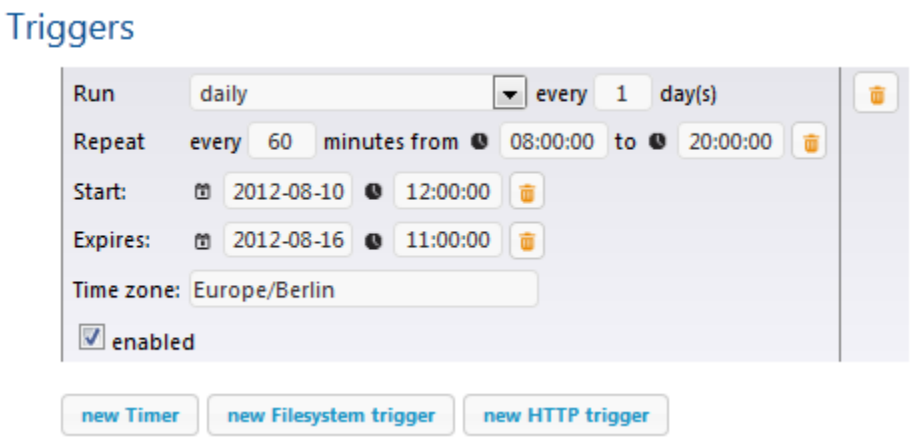
File system trigger
This type of trigger lets you check a specified directory, as well as check the content of a specific file(s).

HTTP trigger
This type of trigger lets you poll a specified URI for changes.

You do not need to define any triggers if you intend to make the job available as a service via HTTP.

4.5.1 Timer trigger types & common settings

Multiple triggers can be defined per job allowing maximum flexibility. This means that several triggers can be active simultaneously, and that whenever any of the triggers is fired, all execution steps of the job associated with the trigger are processed.



Run
The "Run" combo box allows you to define the specific days that the trigger can be activated. This option only refers to dates! There is no time component when you make this selection.

Run period options:

- Once
- Daily
- On days of week
- On days of months
- On days of weeks of months

Common trigger properties:

Repeat
The Repeat options define the interval between successive trigger firings, per day. The days when this will occur, are defined by the selection made with the "Run" combo box.

The "every" field lets you define the period between the job runs, in minutes.

The "from" and "to" fields define the time range between which the triggers will fire.

Start - date and time
Start date/time entries are only mandatory for a timer trigger that uses the "Run - Once option.

Start date/time are optional for file system and HTTP triggers. Clicking in the Date field opens a pop-up calendar from which you can select the start date.

E.g. Repeat every "60" minutes from "08:00" to "20:00" with start time at 09:33. This means that the trigger will become active at 09:33, and as the repeat interval is 60 minutes, it will fire for the first time at 10:00, with repeats at every full hour.

Expires date - time.
The Expires fields allows you to define the date/time from when the job is to expire. The trigger will not fire after this date/time.

Time Zone
This field allows you to define timers that will fire at the same time of day, even if there are daylight saving time switches. Clicking in the field opens a pop-up time zone picker. The default time zone is defined in the server administration settings.

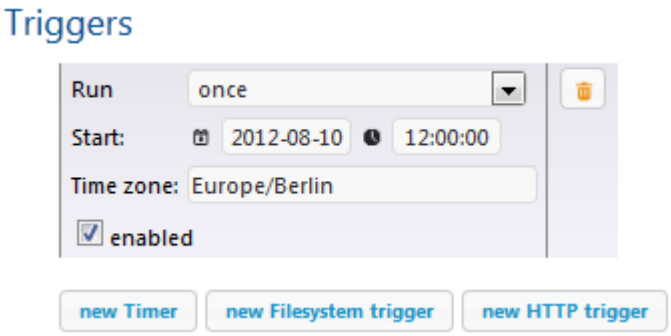
Enabled
The "enabled" check box allows you to enable/disable each individual trigger in the trigger list. This option is useful when creating and testing new jobs.

Waste basket 
The waste basket allows you to delete the whole trigger, or any of the sub elements that are part of it.

Note: Triggers and defaults
If you use job parameters with triggers, make sure that all parameters have defaults or the job will not execute.

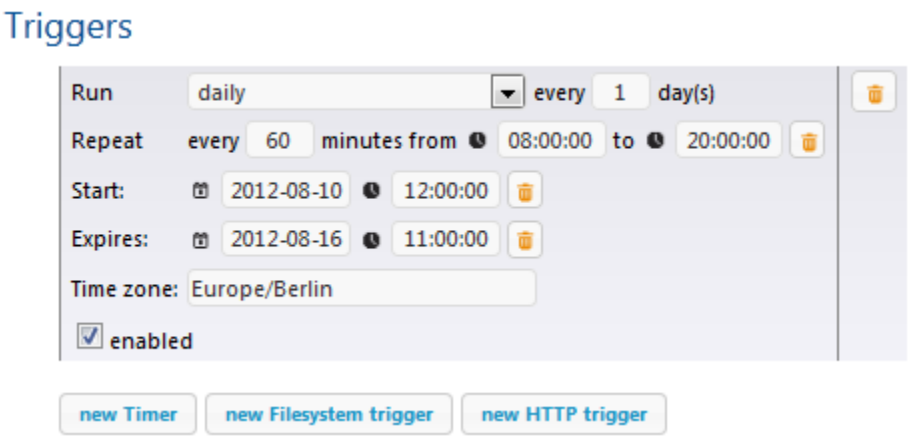
4.5.2 Run Once

This type of trigger will fire once on the day specified, at the exact time given in the time field.



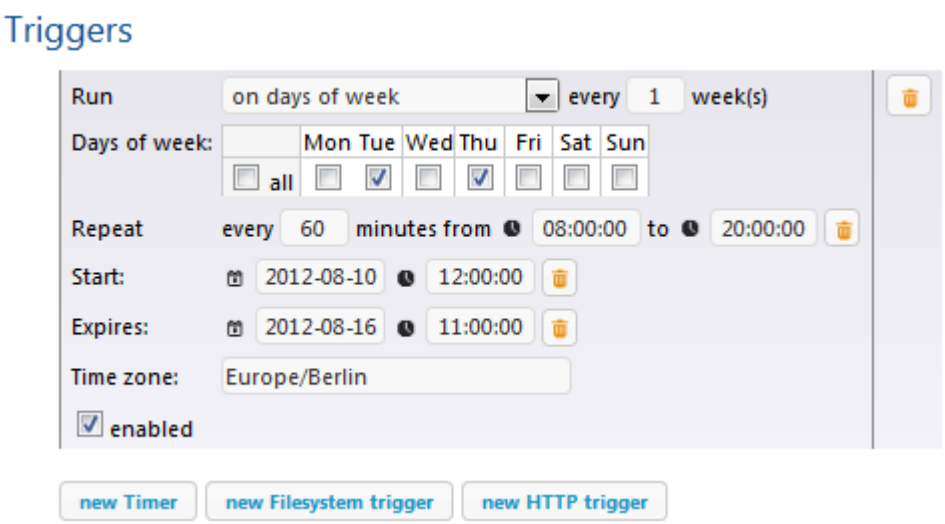
4.5.3 Run Daily

This type of trigger will fire every day between the dates specified, with the first firing at 12:00, and repeat every full hour.



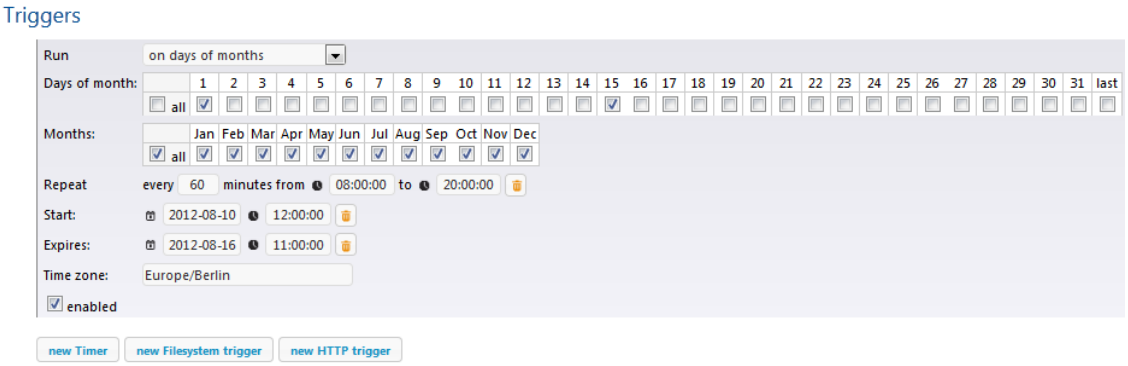
4.5.4 Run On days of week

This type of trigger will fire every week on Tuesday and Thursday between the dates specified. The first time it fires will be at 12:00, and repeat every full hour.



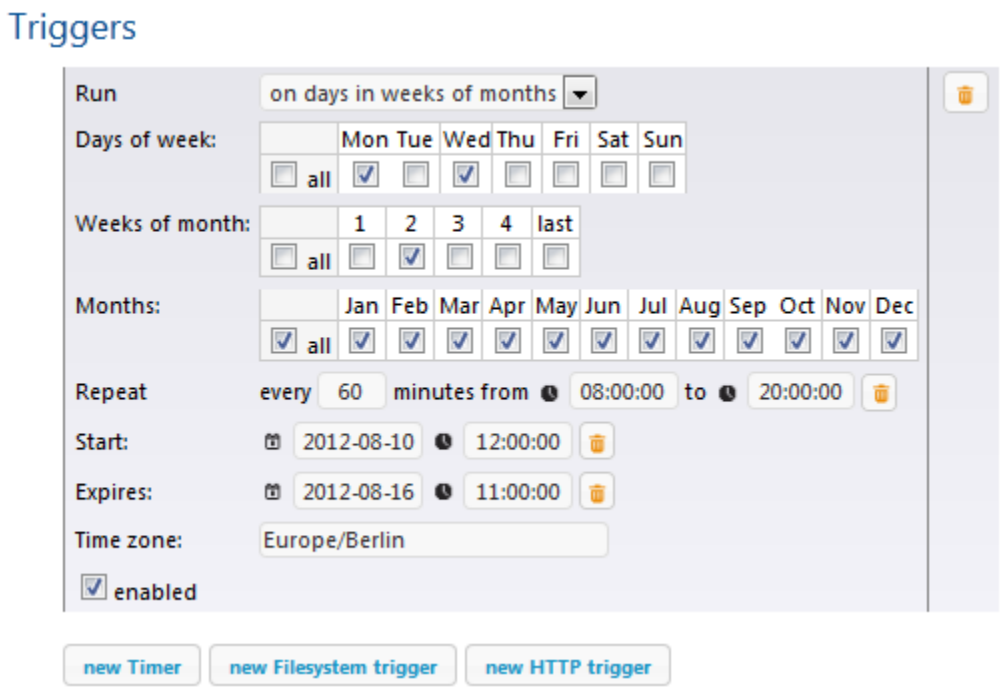
4.5.5 Run On days of months

This type of trigger will fire on the 1st and 15th every month between the dates specified. The first time it fires will be at 12:00, and repeat every full hour.



4.5.6 Run On days in weeks of months

This type of trigger will be fired on Monday, and Wednesday of every second week, of every month between the dates specified. The first time it fires will be at 12:00, and repeat every full hour.



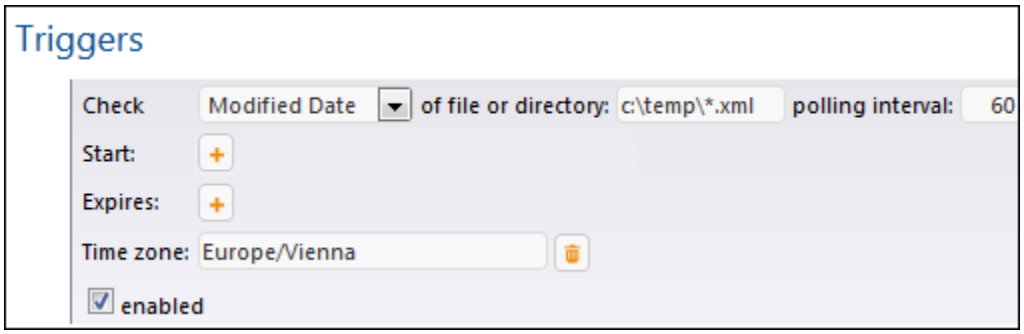
4.5.7 File system trigger

File system trigger:
This type of trigger lets you check a specified directory, as well as check the content of a specific file(s). A directory can be checked for updated or new files (you cannot check for deleted files). Wildcards can be used to filter specific files of the directory.

Check - Modified date:
The trigger checks the last modification timestamp of all the specified files. If any dates have changed, or a new file has been added, then the trigger fires.

Check - Content:
This option computes and stores a hash code for the specified file. After the polling interval has passed, the hash code is recomputed and compared to the stored value. If there is a difference then the trigger fires. Note that this can place considerable load on the server.

If any dates have changed, or a new file has been added, then the trigger also fires.



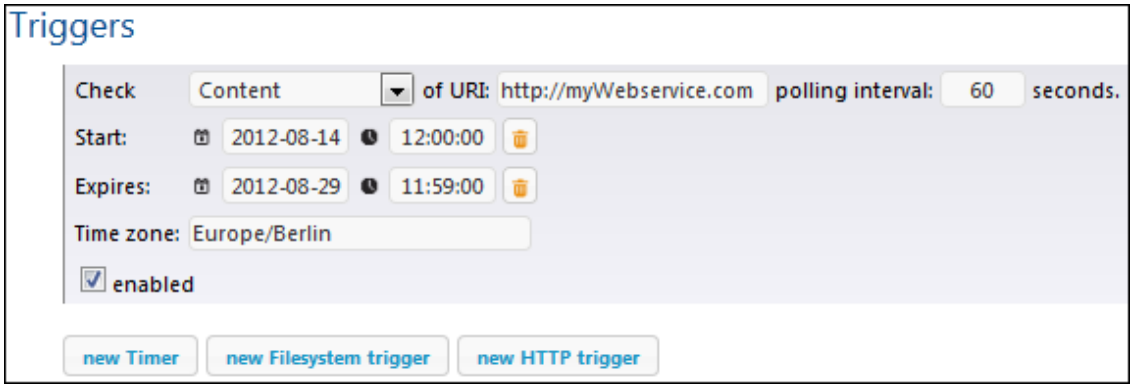
When adding a file system trigger, FlowForce automatically adds the "triggerfile" parameter to the job. This parameter is set at runtime to the name of the file that triggered the job execution. This file name can then be passed to an execution step to process the file.

4.5.8 HTTP trigger

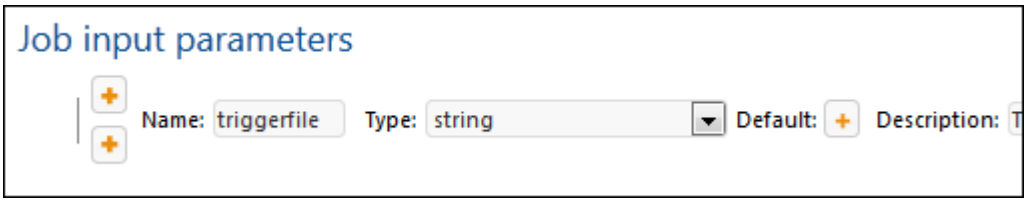
HTTP trigger
This type of trigger lets you poll a specified URI for changes.

Check - Modified date: If a URI is being polled, then the "Last-Modified" HTTP header is checked. If the HTTP header is missing, then "Check Content" is used.

Check - Content: If a URI is being polled, then the "Content-MD5" optional header field is checked. This is a 128 bit "digest" used as a message integrity check. If the MD5 header has changed after the polling interval has passed, then the trigger fires. If the header is not provided by the server, the content is retrieved and hashed locally.



When adding an HTTP trigger, FlowForce automatically adds the "triggerfile" parameter to the job. This parameter is set at runtime to the name of a temporary file that contains the downloaded content at the URI that triggered the job execution. This file name can then be passed to an execution step to process the file.



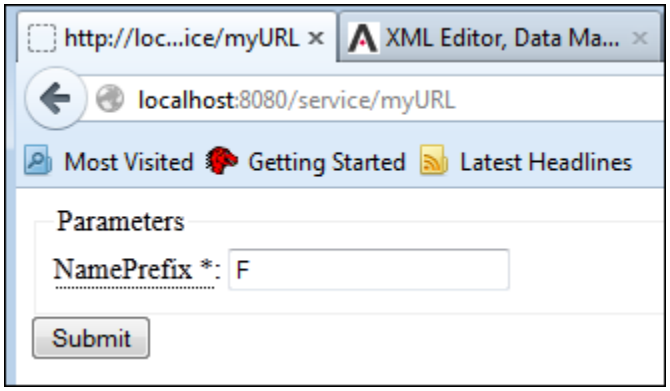
4.6 Service

FlowForce permits exposing jobs as web services via the HTTP protocol. This allows programmatic and interactive access to these jobs, making it possible to use them on-demand.

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; otherwise it is optional and can be provided, taking the default value if it is not provided.

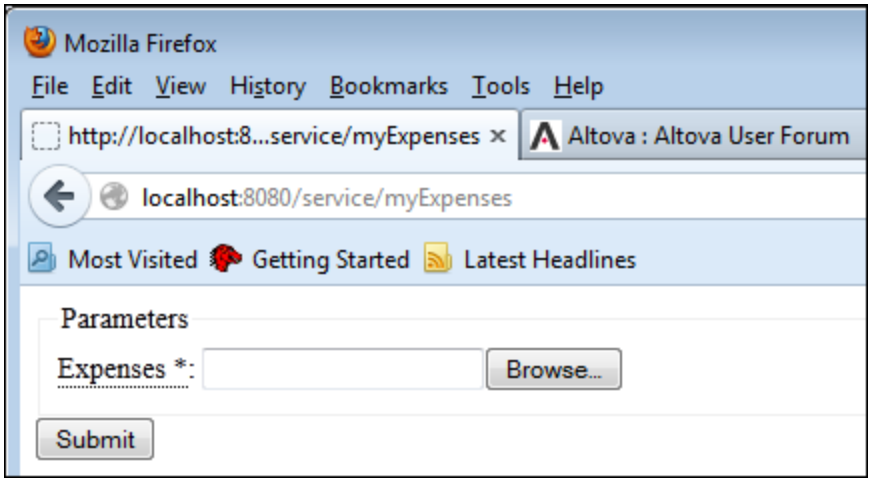
FlowForce checks if all mandatory parameters are provided when the service is invoked. If some are missing the service execution fails. For testing purposes FlowForce supplies a simple HTML form allowing parameters to entered manually. Note: When using Internet Explorer 9 as your browser, please disable the "Show friendly HTTP error messages" in the Advanced tab, to view the form.

This form allows you to enter a value for all parameters.



Please see the tutorial example: Using parameters to query a database.

Streams need a file to be uploaded using the "Browse" button. When clicking "Submit" the data is transferred to FlowForce, and if accepted, FlowForce starts the job and waits for the result.



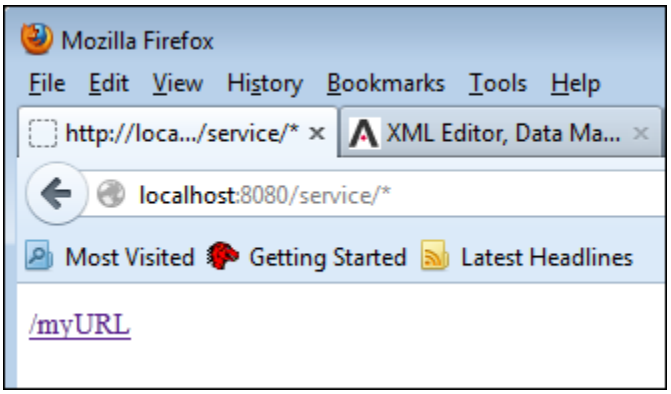
After the job has finished executing FlowForce delivers a response. If the job failed for any reason, FlowForce will return an error message, otherwise it will return the first result file of the last execution step executed, or the last execution step's standard output, if no result file is available.

Services remain active as long as FlowForce server is running.

Please see the tutorial example: Using a deployed mapping as a web service.

To access FlowForce Server services through your browser:

1. Open your browser and enter http://localhost:8080/service/* in the URL text box. If you are using a remote FlowForce server installation, make sure it accepts connections from other machines.
   This command shows all the services currently running on the server.



See also:
Technical details

4.6.1 Technical details

The service interface is primarily meant for machine consumption. At the request URL specified, FlowForce starts a listener service which accepts HTTP GET and POST requests.

It then runs the job execution steps specified and returns the first result file of the last step or the standard output of the last step, if no result files are produced (e.g. for FlowForce system commands).

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.

Authentication:
FlowForce uses HTTP Basic authentication as the means of user authentication. If you want a service available without credentials, you have to grant "use service" permission to the default "anonymous" user. Any other user credentials are checked against the FlowForce user database (so you can use the same usernames and passwords you use for logging into the FlowForce configuration GUI).

You can still supply HTTP credentials when a service is available for anonymous use. The credentials are then checked against the FlowForce user database and the service execution is attributed to the authenticated user instead of user anonymous.

Invalid credentials
If you supply invalid credentials, the request interface will return 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 will also return an HTTP status of 401.

If you supplied valid credentials, but the authenticated user is not granted "use" access 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.

When the client is permitted to use the service, FlowForce will verify the supplied request parameters against the defined parameters of the job. Every parameter that does not have a default must be specified, parameters having a default may also be specified. If parameter validation fails, FlowForce will return a 5xx HTTP status. For debugging and testing purposes FlowForce also returns a simple HTML parameter form in this case.

Two parameter names are used internally by FlowForce to alter the default behavior:

- A parameter named wsdl, when present (regardless of value) will result in FlowForce describing the service inputs as WSDL 2.0 description.
- A parameter named showform, when present (regardless of value) will display the testing HTTP form regardless of any parameter validation errors. In both these cases FlowForce will not return a failure status.

Requests can generally be sent as both HTTP GET or HTTP POST (with multipart/form-data Content-Type), with the exception of parameters of type **stream**, which are only supported for HTTP POST requests.

Service execution behaves like execution via trigger, and is subject to the same queue constraints. You should set the queue limits accordingly.

Execution errors are reported as HTTP 5xx status with a generic error message; detailed information can be found in the FlowForce log.

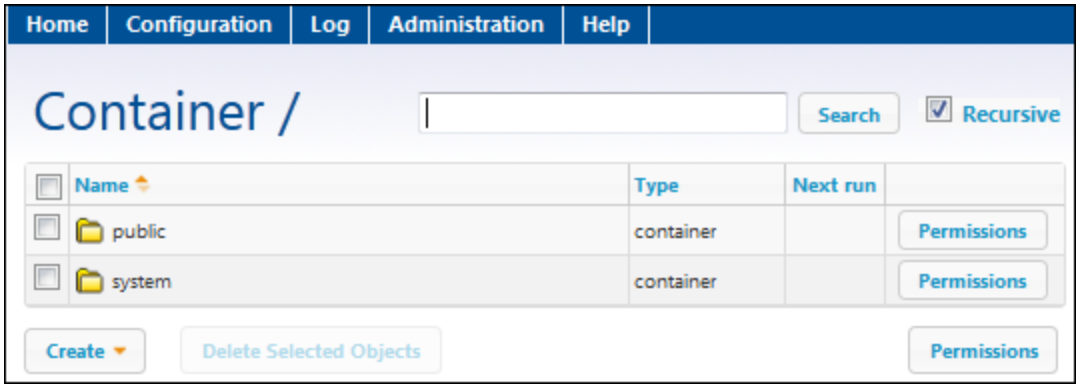
4.7 Credentials

Credentials are stored login data used to execute FlowForce jobs. Credentials can be defined as standalone "objects" and be assigned to various jobs, or they can be manually entered for a specific job.

Jobs are started automatically by FlowForce server, when the defined trigger conditions are met. FlowForce server then runs these jobs using a specific operating system user account, ensuring that job steps do not access unauthorized data. Note that file watch triggers are also assigned credentials.

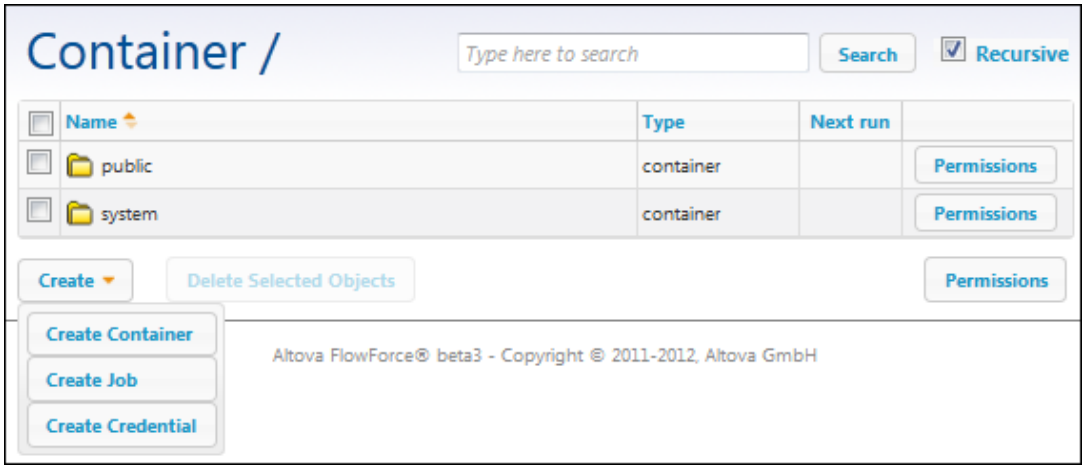
Credentials can be created, or deleted, on the Configuration (Container) page. Note that job credentials, i.e. username and password, can now also be entered for individual jobs on the Job page.

Any user that has "write" access to the Configuration permission, can edit or remove credentials.

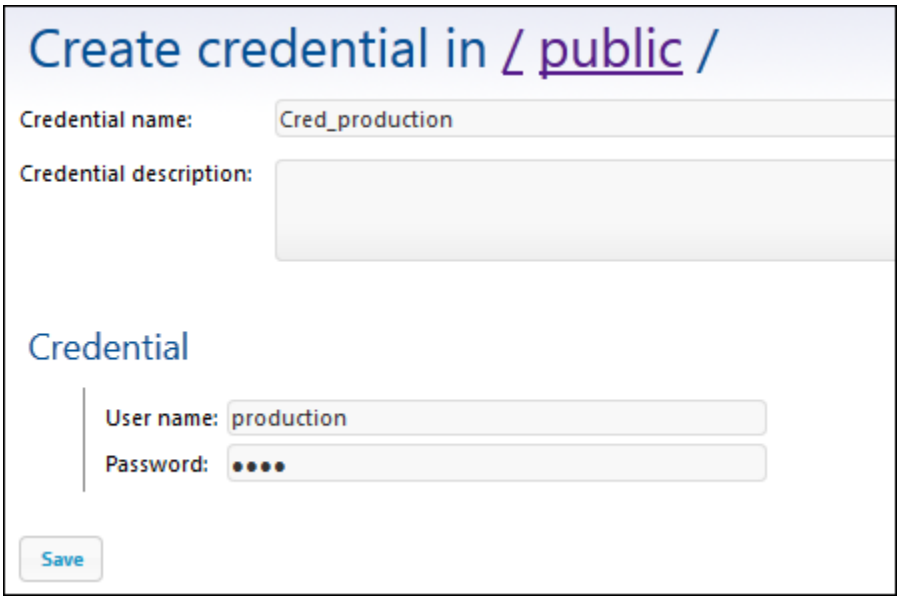


To add a credential to FlowForce:

1. Click the container you want to create the new credential in, e.g. public.
2. Click the "Create" button and select the "Create Credential" entry.



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, please use the form **username@domain**.



4. Click Save.
   The new credential "Cred_production" has been saved in the /public container.
5. Click the "Configuration" button to return to the Container page.

Please see Permissions for information on the container permissions that can be defined.

Credentials and jobs
Every job MUST have a credential assigned to it for the job steps to be executed. This defines the **operating system** user account used to run the job steps.

A predefined credential can be selected using the combo box, or the job credentials can be manually entered in the "User name" and "Password" fields.

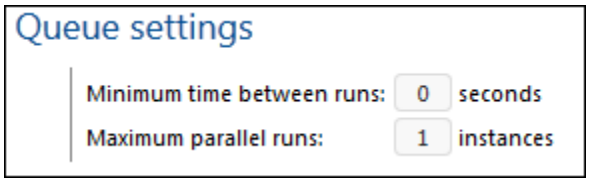
Note:
If you manually enter the user name and password, you will have to update them for those specific jobs, whenever your server credentials are changed.

Credentials:

- Credentials can be created in any container a user has access to.
- 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 database. The administrator should make sure to restrict access to the FlowForce database file.

4.8 Queue settings

Each Job has a queue assigned to it allowing you to define how many instances of the same job can be run in parallel.



Maximum parallel runs:
Enter the number of times the same job may be executed in parallel on the server.

Minimum time between runs:
Enter the time in seconds that must pass after each of the parallel jobs starts before another one may start.

Index

8

A

C

D

E

F

G

H

I

J

L

M

O

P

Q

R

S

T

U

V

W