

## How to Configure a Parameter Set in a Taskflow

## Abstract

This article describes how you can use a parameter set in a taskflow.

## Supported Versions

- Informatica Cloud Data Integration May 2022

## Table of Contents

|  |    |
|--|----|
| Overview . . . . .                               | 2  |
| About ParamSetCli. . . . .                       | 3  |
| ParamSetCli utility requirements. . . . .        | 3  |
| ParamSetCli utility setup. . . . .               | 3  |
| Parameter set requirements. . . . .              | 5  |
| Parameter scope. . . . .                         | 5  |
| Sample parameter set. . . . .                    | 7  |
| Using the ParamSetCli utility. . . . .           | 7  |
| Running a taskflow with a parameter set. . . . . | 10 |
| Using the RunAJob utility. . . . .               | 13 |
| Rules and guidelines for parameter sets. . . . . | 14 |

## Overview

In PowerCenter (PC) to cloud migration, PC workflows are migrated to taskflows. PowerCenter supports parameter files that enable you to pass values for workflow variables and session parameters through an externalized file-based registry. However, taskflows accept inputs only through JSON or XML payload at run time. Therefore, to support PC workflow to taskflow migration, the concept of parameterization has been introduced in taskflows.

You can assign a parameter file to provide values for taskflow input parameters and run a taskflow. A parameter file is a list of user-defined parameters and their associated values. You can use a parameter file to define values that you want to update without having to edit the taskflow. The taskflow reads parameters from the parameter file, and when the taskflow runs, the values are applied.

Depending on the sections and parameters defined in the parameter file, you can configure parameterization in the following ways:

- **Parameter file:** If the parameter file contains sections and parameters for taskflows and mapping tasks, you can save the parameter file in the location where the Data Integration job runs. For more information about parameter files, see the [community article](#) and the [Parameter files](#) documentation.
- **Parameter set:** If the parameter file contains sections and parameters at the taskflow level, you can upload the parameter file to the Informatica managed cloud-hosted repository using the ParamSetCli utility and use them in taskflows. This uploaded parameter file is known as a parameter set. You can also define parameters for Subtaskflow steps in a parameter set.  
To convert PC parameter files to parameter sets, you must install the ParamSetCli utility. This article describes parameter sets.

Consider you have a taskflow where you want to orchestrate two mapping tasks, and you want to specify inputs for the taskflow and parameters for the mapping tasks in the parameter file. In this case, you can create a parameter file with sections for each asset and save the parameter file to the location where the Data Integration tasks run. For the

taskflow to use this parameter file, you must upload the parameter file to the cloud-hosted repository using ParamSetCli. The uploaded parameter file becomes a parameter set for the taskflow. You can then specify the parameter set name in the taskflow.

Additionally, if you have Data Integration tasks such as mapping tasks with a parameterized connection password, it is recommended that you create a separate parameter file with the sensitive information. You can create a separate parameter set with taskflow-related information.

## About ParamSetCli

ParamSetCli is Informatica's command line interface utility that enables you to perform the following actions on a parameter set:

- Upload a parameter set to the cloud-hosted repository.
- Download a parameter set from the cloud-hosted repository.
- Delete a parameter set from the cloud-hosted repository.
- List all the parameter sets available in the cloud-hosted repository.

The following video shows you how to upload, download, and delete a parameter set and list the uploaded parameter sets from the cloud-hosted repository:

<https://www.youtube.com/watch?v=zDPYS9e0ryM>

## ParamSetCli utility requirements

Before you run the ParamSetCli utility, you must have a Java Runtime Environment.

To use the ParamSetCli utility with proxy settings, you must have the Secure Agent installed on the same machine as the ParamSetCli utility.

You can download the ParamSetCli utility from the following community article:

<https://network.informatica.com/docs/DOC-19232>

## ParamSetCli utility setup

To set up the ParamSetCli utility, unzip the downloaded `paramsetcli.zip` to a directory other than the Secure Agent installation directory.

The ParamSetCli utility contains the following files:

- `restenv.properties`. Specifies login credentials.
- `log4j.properties`. Specifies the level of detail to return in log files.

To customize the ParamSetCli properties, you can update the existing `restenv.properties` and `log4j.properties` files.

### Login properties

Specify the Informatica Intelligent Cloud Services login credentials in the `restenv.properties` file. Or, you can pass the login parameters through cli using the `-u` and `-p` arguments.

You can use a password string or an encrypted password for the password parameter and the `PROXYPWD` parameter.

To create an encrypted password, use one of the following commands:

```
./paramsetcli.sh encryptText -t <password>
./paramsetcli.sh encryptText -text <password>
```

You can use `paramsetcli.bat` for Windows and `paramsetcli.sh` for Unix-like operating systems.

Copy the encrypted password string and replace the password or proxy password in the `restenv.properties` file with the encrypted string, and then set the `use.encrypted` flag to `true`.

The following example shows the `restenv.properties` file with an encrypted password, an encrypted proxy password, and the `use.encrypted` flag set to `true`:

```
baseUrl=https://dm-us.informaticacloud.com/ma
username=testuser
password=:1xCGDTC0oD9B2Rmd8Sr4IZWaWWkcEmiK5fy+GkycA==
ACTIVITYMONITORWAIT=2000
PROXYHOST=
PROXYPORT=
PROXYUSERNAME=
PROXYPWD=x2UNmQ1nxXbKvKs1+Z0kvQ==
RETRYCOUNT=
use.encrypted=true
paramSetBaseUrl=https://nal.dm-us.informaticacloud.com/active-bpel
```

Include the following parameters in the `restenv.properties` file or in action commands:

| Parameter                        | Description  |
|----------------------------------|--|
| <code>baseUrl</code>             | Informatica Intelligent Cloud Services MA service URL.<br>Default is <code>https://dm-us.informaticacloud.com/ma</code> .  |
| <code>username</code>            | Informatica Intelligent Cloud Services user name.  |
| <code>password</code>            | Informatica Intelligent Cloud Services password or encrypted password string.  |
| <code>ACTIVITYMONITORWAIT</code> | The amount of time the utility waits before retrying if an internal exception occurs, such as a login failure or network problem.<br>Default is <b>5000</b> milliseconds.  |
| <code>PROXYHOST</code>           | Host name or an IP address of the HTTP(s) proxy server.  |
| <code>PROXYPORT</code>           | Port number of the HTTP(s) proxy server.   |
| <code>PROXYUSERNAME</code>       | Authenticated user name for the HTTP(s) proxy server. Required if the proxy server requires authentication.  |
| <code>PROXYPWD</code>            | Password for the authenticated user or an encrypted password string for the authenticated user. Required if the proxy server requires authentication.  |
| <code>RETRYCOUNT</code>          | The number of times the utility polls for status. This parameter is used for polling the activity monitor and activity log for job status and for internal exceptions such as login failure or network problems.<br>Default is <b>6</b> .<br><b>Note:</b> Informatica Intelligent Cloud Services adds 10 seconds between each API call to prevent server issues. |
| <code>use.encrypted</code>       | Enables the use of an encrypted password. To use an encrypted password or an encrypted proxy password, set the value to <b>true</b> .  |
| <code>paramSetBaseUrl</code>     | Base URL of Informatica Cloud Data Integration suffixed with <code>active-bpel</code> .<br>Default is <code>https://nal.dm-us.informaticacloud.com/active-bpel</code> .  |

## Parameter set requirements

You can reuse parameter sets across taskflows. You can also define parameters for Subtaskflow steps in a taskflow. To reuse a parameter set, define local and global parameters within a parameter set.

You can group parameters in different sections of the parameter set. Each section is preceded by a heading that identifies the project, folder, and assets to which you want to apply the parameter values. You define parameters directly below the heading, entering each parameter on a new line.

The following table describes the headings that define each section in the parameter set and the scope of the parameters that you define in each section:

| Heading  | Description   |
|--|---|
| #USE_SECTIONS  | Use this heading as the first line of a parameter set that contains sections. Otherwise, the taskflow reads only the first global section and ignores all other sections.   |
| [Global]   | Defines parameters for all projects, folders, taskflows, and Subtaskflow steps.   |
| [project name].[folder name].[taskflow name]<br>-or-<br>[project name].[taskflow name]   | Defines parameters for a specific taskflow.<br>If a parameter is defined in a taskflow section and in a global section, the value in the taskflow section overrides the global value. If a parameter value is not defined in the taskflow section, the global value is considered.  |
| [project name].[folder name].[taskflow name].[subtaskflow step name]<br>-or-<br>[project name].[taskflow name].[subtaskflow step name] | Defines parameters for a specific Subtaskflow step.<br>If a parameter is defined in a Subtaskflow step section and in a global section, the value in the Subtaskflow step section overrides the global value.<br>If a parameter is defined in a Subtaskflow step section and in a taskflow section, and the taskflow uses the Subtaskflow step, the value in the Subtaskflow step section is considered.<br>If a parameter value is not defined in the Subtaskflow step or in the taskflow section, the global value is considered. |

If the parameter set does not contain sections, the taskflow reads all parameters as global.

Precede the parameter name with two dollar signs, as follows:

```
$$<parameter>
```

Define parameter values as follows:

```
$$<parameter>=value1
```

```
$$<parameter2>=value2
```

The parameter value includes any characters after the equals sign (=), including leading or trailing spaces. Parameter values are treated as string values.

## Parameter scope

When you define values for the same parameter in multiple sections in a parameter set, the parameter with the smallest scope takes precedence over parameters with a larger scope.

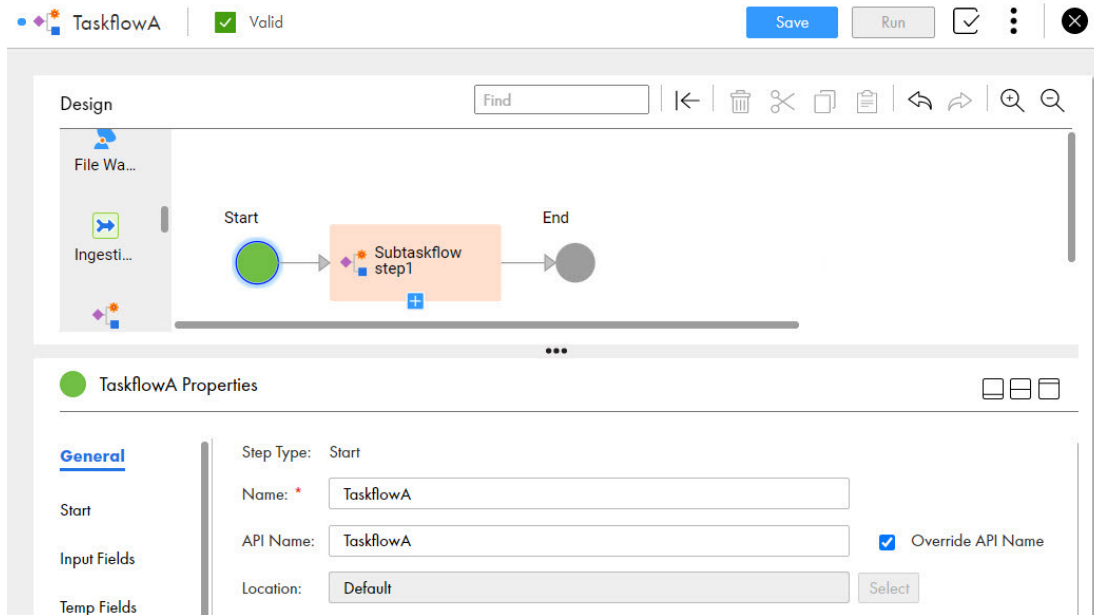
In this case, the taskflow gives precedence to parameter values in the following order:

1. Value defined in the Subtaskflow step section.

2. Values defined in a taskflow section.
3. Values defined in the #USE\_SECTIONS section.
4. Values defined in a global section.

If you define a parameter in a Subtaskflow step section and in a taskflow section, and the taskflow uses the Subtaskflow step, the taskflow uses the parameter value defined in the Subtaskflow step section.

For example, consider a taskflow that contains a Subtaskflow step named Subtaskflow step1 as shown in the following image:



You define the following parameter values in a parameter set:

```
#USE_SECTIONS
$$input=employee_table
[GLOBAL]
$$location=USA
$$department=R&D
[Default].[Folder1].[TaskflowA]
$$input=Leads_table
$$department=HR
$$designation=Manager
[Default].[Folder1].[TaskflowA].[Subtaskflow step1]
$$input=Associates_table
$$department=Finance
```

TaskflowA contains Subtaskflow step1.

When you run TaskflowA, TaskflowA uses the following parameter values:

| Parameter      | Section                         | Value       |
|----------------|---------------------------------|-------------|
| \$\$input      | [Default].[Folder1].[TaskflowA] | Leads_table |
| \$\$department | [Default].[Folder1].[TaskflowA] | HR          |
| \$\$location   | [GLOBAL]                        | USA         |

When you run Subtaskflow step1, the taskflow uses the following parameter values:

| Parameter       | Section   | Value            |
|-----------------|---|------------------|
| \$\$input       | [Default].[Folder1].[TaskflowA].[Subtaskflow step1] | Associates_table |
| \$\$department  | [Default].[Folder1].[TaskflowA].[Subtaskflow step1] | Finance          |
| \$\$location    | [GLOBAL]  | USA              |
| \$\$designation | [Default].[Folder1].[TaskflowA]                     | Manager          |

For all other assets that contain the \$\$input parameter, the taskflow uses the value employee\_table.

## Sample parameter set

The following example shows a sample parameter set that contains parameter values for a taskflow:

```
#USE_SECTIONS
$$input=customer_table
[GLOBAL]
$$Country=USA

[Project1].[Folder1].[Taskflow1]
$$FirstName=John
$$LastName=Williams
$$State=Karnataka
$$Country=India

$$Company=CompanyA
$$ShippingID=1234
$$CustomerID=1
$$ContactNumber=1234567890

[Project1].[Folder1].[Taskflow1].[Subtaskflow1]
$$FirstName=Jones
$$LastName=Greg
$$ContactNumber=0123456789
$$ShippingID=4321
```

The taskflow uses parameters to pass different values for the country, state, first name, last name, shipping ID, customer ID, and contact number.

## Using the ParamSetCli utility

To use the ParamSetCli utility, type the `runParamSetCli` command followed by the arguments.

Use `paramsetcli.bat runParamSetCli` for Windows. Use `paramsetcli.sh runParamSetCli` for Unix-like operating systems.

For each action that you want to perform, you must specify the action type.

## ParamSetCli utility arguments

You can use the following arguments in a ParamSetCli command:

| Parameter                 | Argument | Description   |
|---------------------------|----------|---|
| username                  | -u       | Required. The user name to log in to Informatica Intelligent Cloud Services.  |
| password                  | -p       | Required. The password or encrypted password string to log in to Informatica Intelligent Cloud Services.  |
| baseUrl                   | -bu      | Required. Informatica Intelligent Cloud Services MA service URL. Default is <a href="https://dm-us.informaticacloud.com/ma">https://dm-us.informaticacloud.com/ma</a> .   |
| paramsetbaseUrl           | -psbu    | Base URL of Informatica Cloud Data Integration suffixed with active-bpel. For example, <a href="https://na1.dm-us.informaticacloud.com/active-bpel">https://na1.dm-us.informaticacloud.com/active-bpel</a><br>You can provide the paramset base URL in the <code>restenv.properties</code> file or you can pass it through the CLI using the <code>-psbu</code> argument. If it is available in both locations, the <code>-psbu</code> value in the action command takes precedence.  |
| debug                     | -d       | Optional. Displays debugging information.   |
| insecure                  | -k       | Optional. Enables the insecure mode.  |
| action                    | -a       | Required. Defines the action to be performed on the parameter set. You can specify one of the following actions:<br>upload- Upload a parameter set to the cloud-hosted repository by using the unique name. Optionally, you can specify the <code>-f</code> force argument to override the existing parameter set.<br>download- Download a parameter set from the cloud-hosted repository by using the unique name of an uploaded parameter set. If the specified file already exists on your local machine, the CLI does not download the parameter set. To forcefully download and overwrite the existing file, you can use the <code>-f</code> force argument.<br>list- List the parameter sets with basic information such as created by, creation date, parameter set name, parameter file name, and parameter directory available in the cloud-hosted repository.<br>delete- Permanently delete the specified parameter set from the cloud-hosted repository. |
| Force                     | -f       | Optional. Enables you to force upload or download the parameter set.  |
| Parameter Set Unique Name | -un      | Required. A name used to identify a parameter set when performing different operations such as upload, download, or delete a parameter set, or use it in a taskflow. The name must be unique within the IICS organization.  |
| Parameter directory       | -pd      | Specifies the directory where the parameter files are located.  |
| Parameter file name       | -pf      | Parameter file name.  |
| Parameter Set List Page   | -page    | Optional. Page number. Specifies the page number that you want to list.   |
| Page size                 | -ps      | Optional. Specifies the number of entries per page. The default and the maximum number of entries that you can list is 50 per page.   |



## Running the ParamSetCli commands

You can use the following command to run an action using the action name.

```
paramsetcli.bat runParamSetCli -a <actionName>
```

You can perform different actions such as upload, download, and delete a parameter set and list all of the uploaded parameter sets. For each action, you must provide the `-a` argument and appropriate arguments specific to the action as shown below:

### Upload a parameter set

You can use the upload action to upload a parameter set to the cloud-hosted repository. You can upload a file with a size up to 5 MB.

To upload a parameter set, use the following syntax:

```
paramsetcli.bat runParamSetCli -un <unique_param_set_name> -pf <parameter_file_name> -pd  
<parameter_directory> -a <action>
```

For example:

```
paramsetcli.bat runParamSetCli -un Paramset.params -pf Paramsetfile.params -pd C:\files -a  
upload
```

Here, `-un`, `-pf`, and `-pd` are mandatory arguments. If a parameter set already exists in the cloud-hosted repository, you cannot upload a parameter set with the same name. However, if you want to force an update of the content in the existing parameter set, you can use the `-f` force argument.

If you update the parameter set after uploading it to the cloud-hosted repository, you must reupload the parameter set for the changes to take effect.

After you run this command, the parameter set is uploaded to the cloud-hosted repository. The unique name of the parameter set is used in a taskflow to read the inputs.

### Download a parameter set

You can use the unique name of an uploaded parameter set to download it from the cloud-hosted repository.

To download a parameter set, use the following syntax:

```
paramsetcli.bat runParamSetCli -un <unique_param_set_name> -pf <parameter_file_name> -pd  
<parameter_directory> -a <action>
```

For example:

```
paramsetcli.bat runParamSetCli -un Paramset.params -pf Paramsetfile.paramsDownload -pd  
C:\files -a download
```

Here, `-un`, `-pf`, and `-pd` are mandatory arguments. `-pf` specifies the file name with which you want to download the parameter set, and `-pd` specifies the directory where the parameter set is to be downloaded. If the specified file already exists on the local machine, the ParamSetCli utility does not download the parameter set. To forcefully download and overwrite the existing file, you can use the `-f` force argument.

After you run this command, the parameter set is downloaded to the specified directory.

### Delete a parameter set

You can use the unique name of the parameter set to delete it from the cloud-hosted repository.

To delete a parameter set, use the following syntax:

```
paramsetcli.bat runParamSetCli -un <unique_param_set_name> -a <action>
```

For example:

```
paramsetcli.bat runParamSetCli -un Paramset.params -a delete
```

Here, `-un` is a mandatory argument.

After you run this command, the parameter set is permanently deleted from the cloud-hosted repository.

### List the parameter sets

You can list the parameter sets that you have uploaded to the cloud-hosted repository.

To list the parameter sets, use the following syntax:

```
paramsetcli.bat runParamSetCli -a list
```

When you run the above command, the first page with 50 entries is listed. It displays basic information such as created by, creation date, parameter set name, parameter file name, and parameter directory for the parameter sets available in the cloud-hosted repository.

The default and the maximum number of entries that you can list is 50 per page. If you want a customized list, you can specify the page number and page size as per the requirement.

To customize the parameter set list, use the following syntax:

```
paramsetcli.bat runParamSetCli -a list -page <page_number> -ps <page_size>
```

For example: `paramsetcli.bat runParamSetCli -a list -page <5> -ps <10>`

Here, `-page` is the page number that you want to list and `-ps` is the number of entries you want to display on that page.

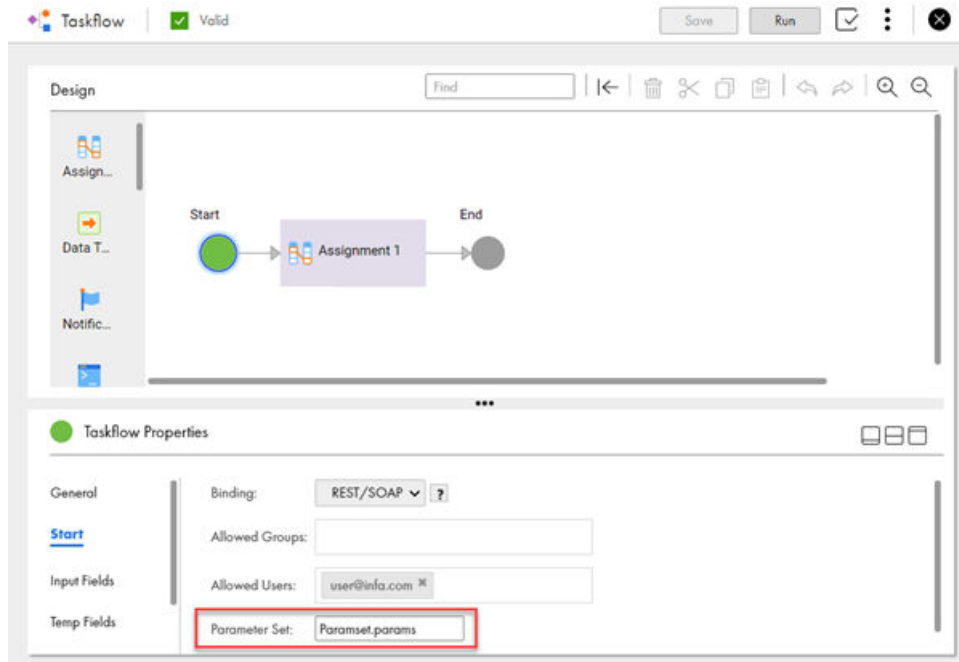
After you run this command, the fifth page with 10 entries are listed.

## Running a taskflow with a parameter set

You can specify the parameter set name while designing the taskflow. However, you can override the parameter set name at run time.

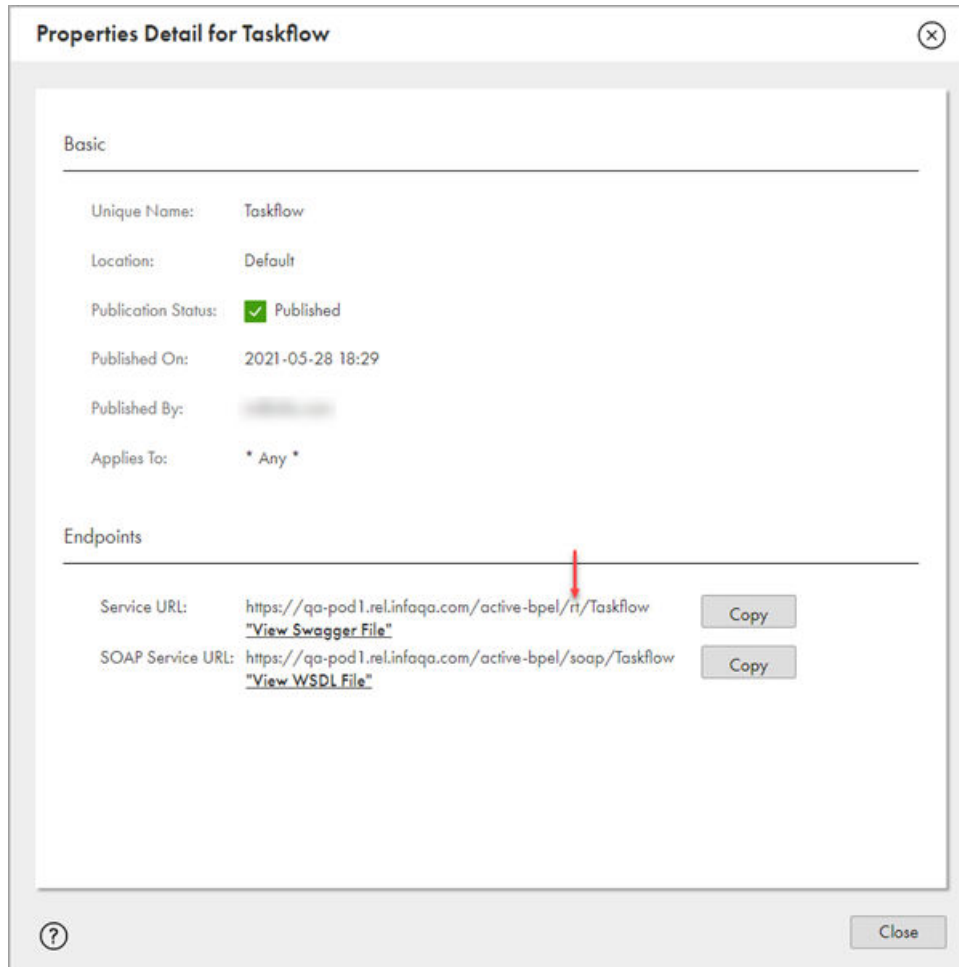
Perform the following steps to use a parameter set in a taskflow:

1. In Data Integration, select **Explore**.  
The **Explore** page opens.
2. Navigate to the taskflow for which you want to assign a parameter set.
3. Select the **Start** step, and then in the **Start** tab, enter the parameter set name in the **Parameter Set** field as shown in the following image:



4. Add the input fields and temporary fields for which you want to read the values from the parameter set.
5. From the **Actions** menu, select **Publish** to publish the taskflow and generate the service URLs.
6. From the **Actions** menu, select **Properties Detail**.

The **Properties Detail** dialog box appears displaying the Service URL and the SOAP Service URL as shown in the following image:



**Note:** You can use only the Service URL to run the taskflow with a parameter set. The parameter set cannot be run through the SOAP Service URL.

- To run the taskflow, use the Service URL and the API definition available in the Swagger file to send a request through the REST client.

Before you run the taskflow with a parameter set, you must change `rt` to `tf` in the Service URL as shown below:

**Change:** <Informatica Intelligent Cloud Services URL>/active-bpel/**rt**/**<API\_name>**

**To:** <Informatica Intelligent Cloud Services URL>/active-bpel/**tf**/**<API\_name>**

For example:

`https://qa-pod1.rel.infaqa.com/active-bpel/tf/Taskflow`

You receive the taskflow run ID as the response.

- To override the parameter set at run time with a different parameter set that is available in the cloud-hosted repository, append the Service URL as shown below:

For example:

<Informatica Intelligent Cloud Services URL>/active-bpel/tf/<API\_name>/**paramset/**  
**<new\_parameter\_set\_name>**

`https://qa-pod1.rel.infaqa.com/active-bpel/tf/Taskflow/paramset/overrideParamset.params`

9. To monitor the taskflow with the run ID, perform one of the following steps:
  - Use the run ID to monitor the taskflow run in the **My Jobs** page. The instance name uses the following format:  
`<taskflow name>-<run ID>`
  - Use the status resource to query the status of the taskflow based on the run ID. For more information about using the status resource, see [Monitoring taskflow status with the status resource](#).

## Using the RunAJob utility

You can use the RunAJob utility to run jobs or check job status instead of making calls directly through the Informatica Intelligent Cloud Services REST API.

To use the RunAJob utility, type the RunAJob utility command followed by the arguments.

The following string is the RunAJob utility command:

```
cli.bat runAJobCli
```

### RunAJob utility arguments

You can use the following arguments in a RunAJob command to run a taskflow:

| Parameter              | Argument | Description   |
|------------------------|----------|---|
| username               | -u       | Informatica Intelligent Cloud Services user name.   |
| password               | -p       | Informatica Intelligent Cloud Services password.  |
| baseUrl                | -bu      | Base URL. Default is <code>https://dm-us.informaticacloud.com/ma</code> . Required.   |
| taskflowUniqueName     | -un      | Taskflow unique name. Required to run a Data Integration taskflow.  |
| parameterSetUniqueName | -pun     | Unique name of the parameter set from which you want to provide values to the taskflow input parameters.  |
| parameterDir           | -pd      | Parameter file directory. Include with parameterFile. Default is: <code>&lt;SecureAgentInstallDir&gt;/apps/Data_Integration_Server/data/userparameters</code> |
| parameterFile          | -pf      | Parameter file. Can be used for mapping tasks.  |
| waitFlag               | -w       | Wait flag. Determines whether to wait for the job to complete or run the job in the background.   |
| debug                  | -d       | Display debugging information.  |
| insecure               | -k       | Insecure mode.  |

### Using the parameter set argument in the RunAJob utility

You can use a parameter set to provide values for taskflow input parameters and run the taskflow using the RunAJob utility.

You can download the RunAJob utility from the following community article:

<https://network.informatica.com/docs/DOC-19232>

When you run a taskflow that uses a parameter set, type the RunAJob utility command followed by the `-pun` or `--parameterSetUniqueName` argument.

To run a taskflow using the RunAJob utility, the taskflow must be published and you must include values for **Allowed Users** and **Allowed Groups** in the taskflow designer. For more information about taskflows, see the [Taskflows](#) documentation.

To run a taskflow that uses a parameter set, use one of the following syntaxes:

```
cli.bat runAJobCli -t TASKFLOW -un <taskflow_name> -fp <folder_path> -pun
<unique_param_set_name>

cli.bat runAJobCli -t TASKFLOW -un <taskflow_name> -fp <folder_path> --parameterSetUniqueName
<unique_param_set_name>
```

For example:

```
cli.bat runAJobCli -t TASKFLOW -un myTaskflow -fp myproject/folder1 -pun sampleParamSet
cli.bat runAJobCli -t TASKFLOW -un myTaskflow -fp myproject/folder1 --parameterSetUniqueName
sampleParamSet
```

Here, `-pun` and `--parameterSetUniqueName` specify the unique name of the parameter set from which you want to provide values to the input parameters for the taskflow.

**Note:** After or effective from the November 2021 release, all taskflows run by using the `/tf` endpoint from the RunAJob utility.

For more information about RunAJob utility, see the [RunAJob utility](#) documentation.

## Rules and guidelines for parameter sets

Consider the following rules and guidelines when you use parameter sets in taskflows:

- If the parameter set that you use in a taskflow is not available in the cloud-hosted repository, the taskflow fails at run time. You can upload the parameter set and run the taskflow again, or override the parameter set name at run time with a valid parameter set name.
- If the input field that you pass does not have a value specified in the parameter set, the taskflow takes the default value of the data type. For example, in case of Boolean, it takes the value as False, integer as zero, string as null, and so on.
- If a parameter value is another parameter defined in the set, the taskflow uses the first value of the variable in the most specific scope. For example, a parameter set contains the following parameter values:

```
[GLOBAL]
$$ffconnection=my_ff_conn
$$var2=California
$$var6=$$var5
$$var5=North
[Default].[folder5].[sales_accounts]
$$var2=$$var5
$$var5=south
```

In "sales\_accounts," the value of "var5" is "south." Since var2 is defined as var5, var2 is also "south."

- The value of a parameter is global unless it is present in a section.
- If a parameter set contains the same heading more than once, the taskflow uses the parameters from the first section.

## Author

Sonali Kumbhalkar