Performance DNA KPI Upload Application Guide

Performance DNA 9.0

1/15/2022

Table of Contents

Data Uploader Application Guide

Welcome to the *Data Uploader Application Guide*.

The Data Uploader Application is a standalone application that enables users to import both KPI data sources and OrgData file sources. It is similar to the DNA import service, but can be used with Performance DNA on-premises, Amazon Web Services (AWS), or Azure platforms.

Prerequisites

The following section lists the prerequisites required to run the **Data Uploader** application.

For KPI uploads:

- Must be a Performance DNA Administrator.
- An AWS API key if you are running an AWS instance of Performance DNA.
- Each imported KPI must have a matching Base Data Definition by name or alias.
- A manually generated OAuth2 Refresh Token if the tenant is using OAuth2 as an authentication method for their API and the OAuth2 provider does not support loop-back URLs.
- A base fact batch size for the Landlord. **Note**: The base fact batch size can be set by logging in as the Tenant Admin/Landlord and navigating to the **Global Settings** page. The default is `10,000`.

For OrgData uploads:

- Must be a Performance DNA Administrator.
- An AWS API key if you are running an AWS instance of Performance DNA.
- A manually generated OAuth2 Refresh Token if the tenant is using OAuth2 as an authentication method for their API and the OAuth2 provider does not support loop-back URLs.
- OrgData must be already setup within the system with a matching OrgData file name configuration.

Installing Data Uploader

You do not have to install the **Data Uploader** application explicitly. All you must do is, simply copy the `Apps.DataUploader` folder from the release package to the relevant machine where you want to run the application.

You can then run and configure the application from that location to import KPI data or OrgData by referring their procedures.

Configuring and running Data Uploader for KPI Uploads

This topic explains how to manually configure and run the **Data Uploader** Application for KPI Uploads.

You can use the **Data Uploader** Application by using the command line mode to run the `GSKM.Apps.DataUploader` application and a configuration name (if one exists).

For example, `cmd> GSKM.Apps.DataUploader example` where `example` is the name of the previously saved configuration you want to run.

If you have not provided configuration name in the above command line, you will be prompted to enter the following:

- A base URL for the Performance DNA API.
- A response to whether an AWS API key is required – Y if the API is hosted on AWS; N if it is not.
  - If your response is 'Y', that is, an AWS API key is required, then you will be prompted to enter the API key.
  - If your response is 'N', then you will be prompted to enter the other values given in the following section.
- The Performance DNA username of the user performing the upload.
  - For **OAuth2:** If the configured OAuth2 provider supports loop-back URLs, you will see a new browser window opening to perform authentication, else, you will be prompted to enter the OAuth2 refresh token.
  - For **Basic Auth:** You will be prompted to enter your password.
- The full file path and file name of the file containing the KPI data to import. The KPI data file must be in .**csv** or **.xlsx** format.
- The layout of the data in the file, you can specify either `Tabular` or `Linear` by typing their respective options.
- A response to which column in the imported file contains the unique identifier. A list of the columns will be displayed. You can type the number of the appropriate column which contains the unique identifier.
- The user field mapped to the unique identifier column. A list of the user fields defined in Performance DNA will be displayed. You can type the number of the user field in the list.
- A response to which column in the file contains the date information for the KPIs. You can select the appropriate column from the displayed list.
- If your selection is Linear layout, you will be prompted to select two additional columns mentioned below:
  - The column containing the names of the KPIs.
  - The column containing the values for the KPIs.
- A response to provide a name for the configuration you have just defined. This will save the

configuration parameters in that name and you can reuse it when needed by just specifying the name.

The **Data Uploader** Application then sends the information to the API and upload the new records. The Data Uploader folder contains a sub-folder, labelled `logs`, which stores all log as **.txt** files. The Data Uploader folder also contains the configuration file created in the configuration name you specified in the command line.

Supported data layouts for import

The **Data Uploader** Application supports importing two different data layouts such as Tabular and Linear. In both the layouts, the first row of the import file must be the header row that has header information for the columns, and the subsequent rows must be the data rows containing values for the columns.

The uploader supports both the **.xlsx** and **.csv** file formats.

> Important
>
> For **.xlsx** files, only the first sheet in the workbook will be considered for import.

KPI values must be numerical values including zero. If a KPI value is blank or non-numeric then it will not be imported for that user.

Tabular Layout

A tabular layout must have the following columns:

- User identifier,
- Date of the KPI data
- A column for each KPI

An example for the tabular layout is shown in the following table.

User Id	Data Date	KPI 1	KPI 2	KPI 3	KPI 4
00001	20/10/2018	12.5	125	412	
00002	20/10/2018	22.1	351	217	
00003	20/10/2018	17	199		
00004	20/10/2018	14.2			1044

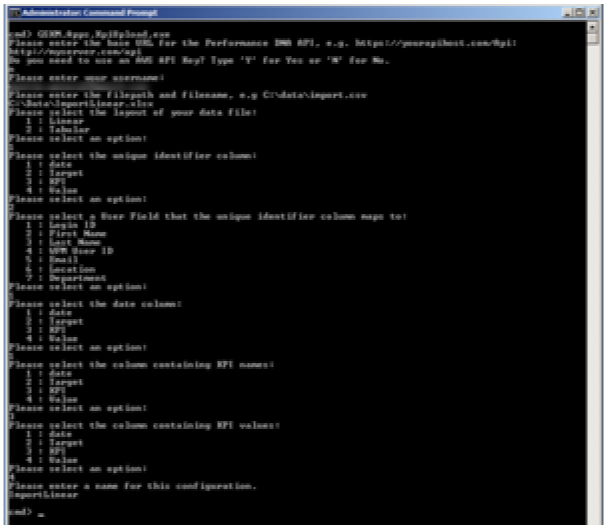
Linear Layout

A linear layout contains a row per user per piece of KPI data.

An example for the linear layout is shown in the following table.

User Id	Data Date	Kpi Name	Value
00001	20/10/2018	Kpi 1	12.5
00001	20/10/2018	Kpi 2	125
00001	20/10/2018	Kpi 3	412
00002	20/10/2018	Kpi 1	22.1
00002	20/10/2018	Kpi 2	351
00002	20/10/2018	Kpi 3	217
00003	20/10/2018	Kpi 1	17
00003	20/10/2018	Kpi 2	199
00004	20/10/2018	Kpi 1	14.2
00004	20/10/2018	Kpi 4	1044

The following screenshot is an example showing a linear layout configuration.



```
cmd> GSKM.Apps.KpiUpload.exe
Please enter the base URL for the Performance DNA API, e.g. https://yourapihost.com/Api:
http://myserver.com/api
Do you need to use an AWS API Key? Type 'Y' for Yes or 'N' for No.
N
Please enter your username:
[redacted]
Please enter the filepath and filename, e.g C:\data\import.csv
C:\Data\ImportLinear.xlsx
Please select the layout of your data file:
   1 : Linear
   2 : Tabular
Please select an option:
1
Please select the unique identifier column:
   1 : date
   2 : Target
   3 : KPI
   4 : Value
Please select an option:
2
Please select a User Field that the unique identifier column maps to:
   1 : Login ID
   2 : First Name
   3 : Last Name
   4 : UPM User ID
   5 : Email
   6 : Location
   7 : Department
Please select an option:
1
Please select the date column:
   1 : date
   2 : Target
   3 : KPI
   4 : Value
Please select an option:
1
Please select the column containing KPI names:
   1 : date
   2 : Target
   3 : KPI
   4 : Value
Please select an option:
3
Please select the column containing KPI values:
   1 : date
   2 : Target
   3 : KPI
   4 : Value
Please select an option:
4
Please enter a name for this configuration.
ImportLinear

cmd> _
```

Configuring and running Data Uploader for OrgData uploads

This topic explains how to manually configure and run the **Data Uploader** Application for OrgData Uploads.

You can use the **Data Uploader** Application by using the command line mode to run the `GSKM.Apps.DataUploader` application with `-orgdata` and a configuration name (if one exists), as shown below.

For example,

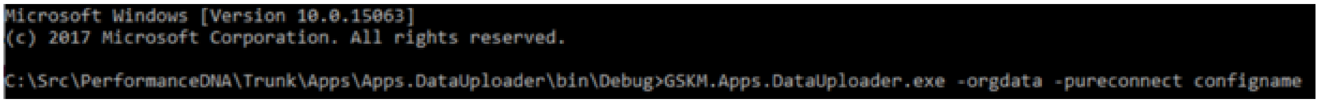
```
cmd> GSKM.Apps.DataUploader -orgdata -pureconnect configname
```

If you are running **Data Uploader** with **PureConnect**, ensure to add `-pureconnect` to the command line as shown below.

For example,

```
cmd> GSKM.Apps.DataUploader -orgdata -pureconnect configname
```

In both the examples, `configname` is the name of the previously saved configuration you want to run. Note that there is no "—" before `configname`



```
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Src\PerformanceDNA\Trunk\Apps\Apps.DataUploader\bin\Debug>GSKM.Apps.DataUploader.exe -orgdata -pureconnect configname
```

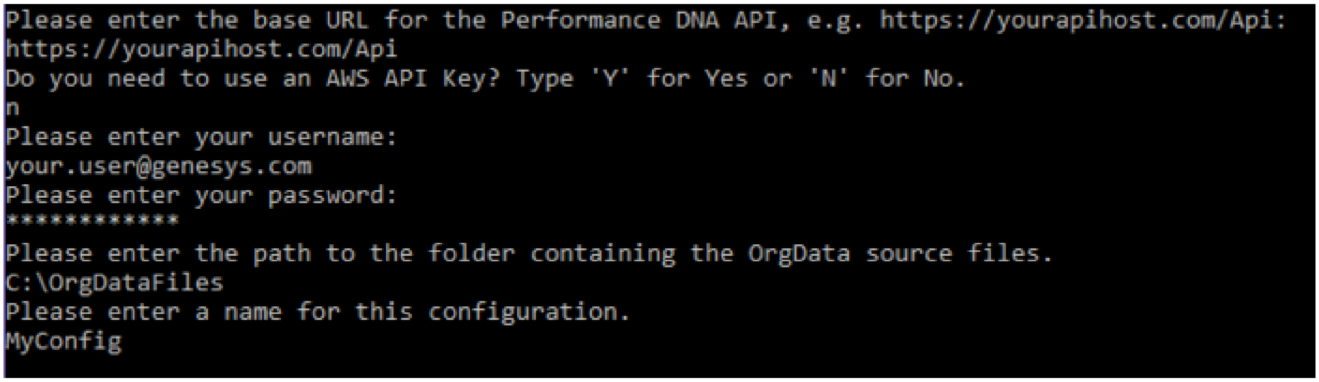
If you have not provided a configuration name in the above command line, you will be prompted to enter the following:

- A base URL for the Performance DNA API.
- A response to whether an AWS API key is required – Y if the API is hosted on AWS; N if it is not.
  - If your response is 'Y', that is, an AWS API key is required, then you will be prompted to enter the API key.
  - If your response is 'N', then you will be prompted to enter the other values given in the following section.
- The Performance DNA username of the user performing the upload.
  - For **OAuth2:** If the configured OAuth2 provider supports loop-back URLs, you will see a new browser window opening to perform authentication, else, you will be prompted to enter the OAuth2 refresh token.
  - For **Basic Auth:** You will be prompted to enter your password.
- The full file path and file name of the file containing the OrgData to import. The OrgData file must be in **.csv** format. Note that the specified OrgData file name must match the file name that already exists for

OrgData uploads set in the **OrgData Setup** screen in Performance DNA.

> **Important**
>
> If you are uploading to **PureConnect**, the file name must follow the format: **FileName_YYYYMMDDHHMMSS.csv** for example, MyOrgData_20190603120000.csv

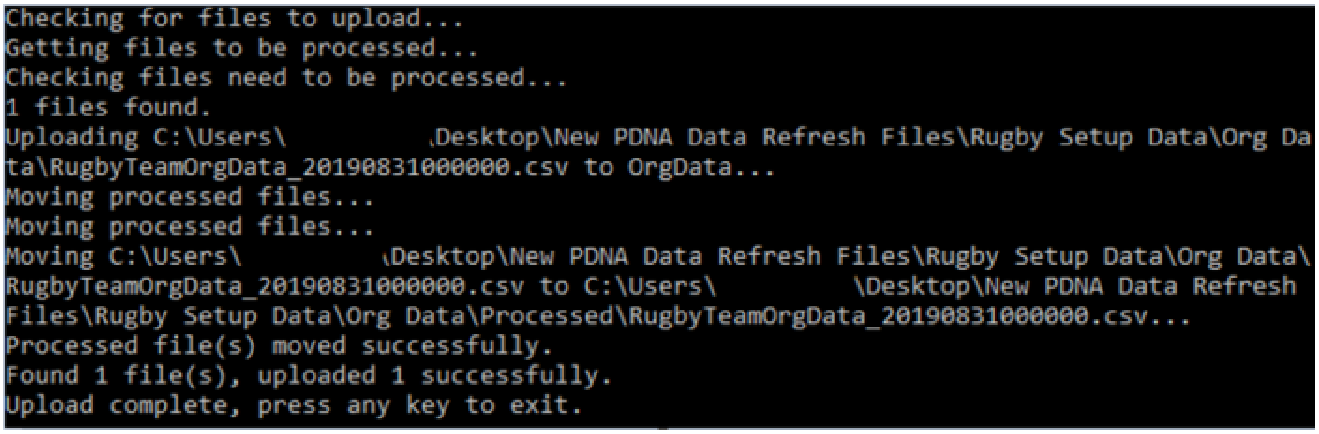
- A response to provide a name for the configuration you have just defined. This will save the configuration parameters in that name and you can reuse it when needed by just specifying the name.



```
Please enter the base URL for the Performance DNA API, e.g. https://yourapihost.com/Api:
https://yourapihost.com/Api
Do you need to use an AWS API Key? Type 'Y' for Yes or 'N' for No.
n
Please enter your username:
your.user@genesys.com
Please enter your password:
************
Please enter the path to the folder containing the OrgData source files.
C:\OrgDataFiles
Please enter a name for this configuration.
MyConfig
```

The **Data Uploader** Application will then process the files, identifies the latest, detects the changes, and uploads the changes from the OrgData file to the system. Based on the changes identified in the uploaded OrgData file, the new/updated users and hierarchy will be updated.

Once the file has been successfully uploaded, the processed file will be moved into the **Processed** folder.



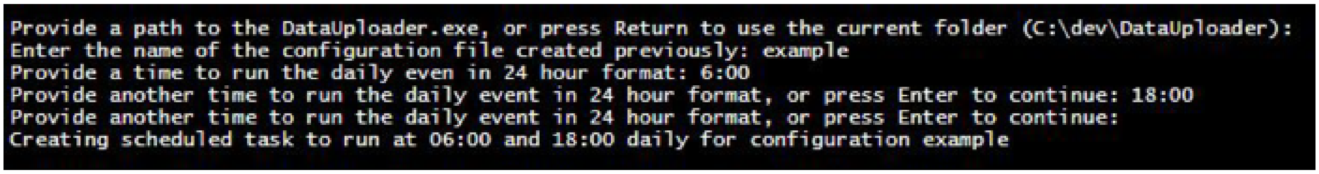
```
Checking for files to upload...
Getting files to be processed...
Checking files need to be processed...
1 files found.
Uploading C:\Users\          \Desktop\New PDNA Data Refresh Files\Rugby Setup Data\Org Da
ta\RugbyTeamOrgData_20190831000000.csv to OrgData...
Moving processed files...
Moving processed files...
Moving C:\Users\          \Desktop\New PDNA Data Refresh Files\Rugby Setup Data\Org Data\
RugbyTeamOrgData_20190831000000.csv to C:\Users\          \Desktop\New PDNA Data Refresh
Files\Rugby Setup Data\Org Data\Processed\RugbyTeamOrgData_20190831000000.csv...
Processed file(s) moved successfully.
Found 1 file(s), uploaded 1 successfully.
Upload complete, press any key to exit.
```

Scheduling the Data Uploader to run daily

The **Data Uploader** application can be run on Windows as a scheduled event by running the following PowerShell script: `CreateScheduleForDataUploader.ps1` for example: `cmd> CreateScheduleForDataUploader.ps1`

You will then be prompted to enter the following:

- The path to the `DataUploader.exe` (and thus the folder with the configuration files).
- The name of the configuration to use in the scheduled event.
- The time in which the run should take place. This should be specified using the standard 24-hour format - for example 18:00 for 6 p.m. **Note:** Because multiple runs can take place in a day, this prompt will appear several times until the user stops entering a time.



```
Provide a path to the DataUploader.exe, or press Return to use the current folder (C:\dev\DataUploader):
Enter the name of the configuration file created previously: example
Provide a time to run the daily even in 24 hour format: 6:00
Provide another time to run the daily event in 24 hour format, or press Enter to continue: 18:00
Provide another time to run the daily event in 24 hour format, or press Enter to continue:
Creating scheduled task to run at 06:00 and 18:00 daily for configuration example
```