

Monarch Server

Version 14.1

 DATAWATCH

[Visual Process Designer API Guide]

 DATAWATCH

Datawatch Corporation makes no representation or warranties with respect to the contents of this manual or the associated software and especially disclaims any implied warranties of merchantability or fitness for any particular purpose. Further, Datawatch Corporation reserves the right to revise this publication and make changes from time to time to its contents without obligation to notify anyone of such revisions or changes.

Monarch Server software is offered and is to be used in accordance with a SOFTWARE LICENSE AND MAINTENANCE AGREEMENT. This agreement stipulates that this software be used only in the computer system designated in that agreement. The agreement further stipulates that the customer shall not copy or alter, or permit others to copy or alter, the software or related materials in whole or in part, in any media for any purpose, except to make an archive (back-up) copy or to make a copy as an essential step in the use of the software with the customer's computer.

Datawatch Corporation hereby grants the buyer the right to reprint this documentation for internal uses only. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, for any other purposes, without the prior written permission of Datawatch Corporation.

Monarch Server v14.1 Visual Process Designer API guide

Copyright © 2017 by Datawatch Corporation

All rights reserved. Printed in the U.S.A.

Unpublished - Rights reserved under the copyright law of the United States.

Monarch Server is a trademark of Datawatch Corporation. Other products mentioned herein may be trademarks or registered trademarks of their respective owners in the United States or other countries.

For U.S. Government End Users, the software is a "Commercial Item(s)," as that term is defined at 48 C.F.R. Section 2.101, consisting of "Commercial Computer Software" and "Commercial Computer Software Documentation," as such terms are used in 48 C.F.R. Section 12.212 or 48 C.F.R. Section 227.7202, as applicable. Consistent with 48 C.F.R. Section 12.212 or 48 C.F.R. Sections 227.7202-1 through 227.7202-4, as applicable, the Commercial Computer Software and Commercial Computer Software Documentation are being licensed to U.S. Government end users (a) only as Commercial Items and (b) with only those rights as are granted to all other end users pursuant to the Datawatch Software License and Maintenance Agreement.

DATAWATCH CORPORATION

CORPORATE HEADQUARTERS

4 Crosby Drive
Bedford, MA 01730, USA

Tel.: +1 800.445.3311 / +1 978.441.2200
Investor Relations: +1 978.441.2200 ext. 8323
Fax: 978.441.1114

Sales: sales@datawatch.com

Support: support@datawatch.com

GREENSBORO, NC

101 S Elm #225
Greensboro, NC 27401
Tel: +1 800.445.3311

NEW YORK

415 Madison Avenue, Suite 1421
New York, NY 10017
Tel.: +1 800.445.3311

UNITED KINGDOM

DATAWATCH INTERNATIONAL LIMITED.

Siena Court, Broadway
Maidenhead, Berkshire SL6 1NJ
Tel: +44 845 362 3270

Floor 42 Level 39
One Canada Square
Canary Wharf
London E14 5AB
Tel: +44 845 362 3270

SWEDEN

DATAWATCH AB

Eriksbergsgatan 10
Stockholm, Sweden SE-114 30
Tel: +46 853 480 483

ASIA PACIFIC

MANILA

U2011 20th Flr Jollibee Plaza Condominium
F. Ortigas Jr. Rd., Ortigas Center
Pasig City 1605
PH
Phone +63 2 633 5583

Table of Contents

Introduction	1
Principle and purpose	1
API testing	1
Working with the Monarch Server API	2
Request authentication	2
API entry point	2
Response type and error handling	2
Monarch Server API Usage Sample	3
API reference.....	4
Login	4
Logging in.....	4
Autologin	5
Logging out.....	5
ProcessDesigner	7
Start Process.....	7
Start Process Using Manifest.....	7
Get Process Status	16
Get Process Log	17
Get Item Log.....	18

Introduction

Monarch Server provides the API for external applications based on the Hypertext Transfer Protocol (HTTP).

Principle and purpose

The API allows external applications to use various functions of Monarch Server and access the Monarch Server database data. Using the API, a new front-end or automation application can be built easily on top of Monarch Server.

API testing

You can test how the API works on the interactive API help page at

`http://localhost/MSAdmin/api/help`

On this page, you can view detailed information about each method and test all the methods by clicking the **Test API** button.



NOTES

In Monarch Server v14.1 that is upgraded from previous versions (e.g., v13.2), the virtual folders could still be labeled as "DSAdmin" (or "DSClient"). In this case, use "DSAdmin" (or "DSClient") instead of "MSAdmin" (or "MSClient") to open the Admin (or Client) page.



Working with the Monarch Server API

To work with the Monarch Server API successfully, review the information below.

Request authentication

For authentication, use the [Login](#) method.

API entry point

The Monarch Server HTTP API is accessed through an entry point.

The format of the request URL is

```
http://<host>:<port>/MSAdmin/api
```

Here, <host> is the name of the computer where Monarch Server is running, and <port> is the port number on which the Monarch Server is listening.

The request can be sent using either an HTTP GET or an HTTP POST method (unless specified otherwise). The target area can be either the top window or a new browser window.

Parameter values that contain URL-prohibited symbols must be encoded as required by the URL specification.

Some parameters depend on a particular operation type.

Response type and error handling

A response to a request can be either one of the following depending on the success condition:

- Request succeeded (the response has a corresponding MIME-type or an HTTP redirect)
- Request failed

Depending on the operation nature and status, the response can be one of the following:

- Standard Monarch Server error page describing the nature of the error (for HTML-based views)
- HTTP error code with an optional message
- XML error description (non-HTML views)



In case of an XML error description, the response type is text/xml. The format of an XML error message is

```
<Error>
  <Message>[Message]</Message>
  <ExceptionMessage>[ExceptionMessage]</ExceptionMessage>
  <ExceptionType>[ExceptionType]</ExceptionType>
  <StackTrace>[StackTrace]</StackTrace>
</Error>
```

The admin application can check the root node's name if text/xml content is returned to determine whether an error occurred.

Monarch Server API Usage Sample

Use the [Login](#) method for user authentication.

```
http://localhost/MSAdmin/api/login
```

Upon logging in, the user can either stay in the system or leave via the [Logout](#) method.

```
http://localhost/MSAdmin/api/logout?redirectUrl={redirectUrl}
```

The user can use different methods to get information about a visual process. See example below.

1. Use the [Start Process](#) method to start a visual process and get a tracking ID.

```
http://localhost/MSAdmin/api/v1/visualprocesses/start
```

2. Then, use the [Get Status](#) method to get the status of the visual process.

```
http://localhost/MSAdmin/api/v1/visualprocesses/{trackingId}/status
```

3. Use the [Get Log](#) method to get a log of the visual process.

```
http://localhost/MSAdmin/api/v1/visualprocesses/{trackingId}/log
```

The user can utilize the Visual Process Designer API Samples installed using the Single Installer.



API reference

The following is a complete reference of the methods supported by the Monarch Server API.

Login

Use the following methods to begin and end a user session to work with Monarch Server.

LOGGING IN

Used to establish a "session" — a secure environment for further operation.

POST method

The entry point for this method is
`api/login`

Request Information

URI Parameters: None

The body parameters are as follows:

Name	Description	Type	Additional information
Username	Username	string	Required
Password	Password	string	None
Domain	User domain	string	None



Request format sample (application/json, text/json):

```
{
  "Username": "sample string 1",
  "Password": "sample string 2",
  "Domain": "sample string 3"
}
```

Request format sample (application/xml, text/xml):

```
<LoginRequest xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://schemas.datacontract.org/2004/07/Datawatch.Web.Enterprise
Server.Common.Models.Api">
  <Domain>sample string 3</Domain>
  <Password>sample string 2</Password>
  <Username>sample string 1</Username>
</LoginRequest>
```

AUTOLOGIN

Used to establish a "session" when the Active directory authentication is enabled with the selected SSO option.

POST method

The entry point for this method is
api/autologin

Request Information

URI Parameters: None

Body Parameters: None

LOGGING OUT

This operation terminates the current API session. It can be useful when the maximum number of user sessions is limited.

GET method

The entry point for this method is



api/logout?redirectUrl={redirectUrl}

Request Information

The URI parameter is as follows:

Name	Description	Type	Additional information
redirectUrl	URL for redirection after session termination. If this parameter is absent, then no redirect is sent.	string	None

Body Parameters: None



ProcessDesigner

This allows the use of Visual Process Designer as a web service.

START PROCESS

This initiates a visual process.

POST method

The entry point for this method is
`api/v1/visualprocesses/start`

Request Information

URI parameters: None

Body parameters: None

Request format sample (application/json, text/json)

```
"sample string 1"
```

Here, "sample string 1" is a name of a visual process.

Request format sample (application/xml, text/xml)

```
<string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">sample  
string 1</string>
```

Response Information

Resource Description: None

START PROCESS USING MANIFEST

This initiates a visual process as described by a given manifest.

POST method

The entry point for this method is
`api/v1/visualprocesses/startwithmanifest`



Request Information

URI Parameters: None

Body parameters: None

Request format sample (application/json, text/json)

```
"sample string 1"
```

Request format sample (application/xml, text/xml)

```
<string  
xmlns="http://schemas.microsoft.com/2003/10/Serialization/">sample  
string 1</string>
```

For Model, Input, RunTime fields, Export, Credentials, DataPrep items, "sample string 1" may be replaced as follows.

Model:

```
<![CDATA[<ProcessStartParams ProcessName="sample string 1">  
  <ProcessItems>  
    <ProcessItem>  
      <ItemType>ModelFile</ItemType>  
      <Model>  
        <Name>sample string 2</Name>  
        <Location>sample string 3</Location>  
        <ExternalLookupOptions>  
          <ExternalLookupOption>  
            <Name>sample string 4</Name>  
            <Password>sample string 5</Password>  
          </ExternalLookupOption>  
        </ExternalLookupOptions>  
      </Model>  
    </ProcessItem>  
  </ProcessItems>  
</ProcessStartParams>]]>
```



NOTES

Sample string 3 (Location) is the file location, definition ID, or web address. Its value depends on the model item type.

Input:

```
<![CDATA[<ProcessStartParams ProcessName="sample string 1">  
  <ProcessItems>  
<!--File Input-->  
    <ProcessItem>  
      <ItemType>InputFile</ItemType>  
      <FileInput>  
        <Name>sample string 2</Name>
```



```

        <FileLocation>sample string 3</FileLocation>
        <PdfPassword>sample string 4</PdfPassword>
    </FileInput>
</ProcessItem>

<!--Web Input-->
    <ProcessItem>
        <ItemType>InputFile</ItemType>
        <WebInput>
            <Name>sample string 5</Name>
            <FileLocation>sample string 6</FileLocation>
            <PdfPassword>sample string 7</PdfPassword>
        </WebInput>
    </ProcessItem>

<!--Database Input-->
    <ProcessItem>
        <ItemType>InputFile</ItemType>
        <DatabaseInput>
            <Name>sample string 8</Name>
            <DataSource>sample string 9</DataSource>
            <Password>sample string 10</Password>
            <TableName>sample string 11</TableName>
            <Filter>sample string 12</Filter>
            <ReplaceNulls>true/false</ReplaceNulls>
            <FirstRowContainsNames>true/false</FirstRowContainsNames>
            <TextParameters>
                <LinesToIgnoreAtStart>int</LinesToIgnoreAtStart>
            </TextParameters>
        </DatabaseInput>
    </ProcessItem>

<AllowEmbeddedLineBreaks>true/false</AllowEmbeddedLineBreaks>
    <Delimiter>sample string 13</Delimiter>
    <TextQualifier>sample string 14</TextQualifier>
    <TextEncoding>sample string 15</TextEncoding>
</TextParameters>
</DatabaseInput>
</ProcessItem>

<!--ContentSystem Input-->
    <ProcessItem>
        <ItemType>InputFile</ItemType>
        <ContentSystemInput>
            <Name>sample string 13</Name>
            <Since>sample string 14</Since>
            <Till>sample string 15</Till>
            <DocType>
                <Id>1</Id>
            </DocType>
            <SelectCriteria>ByDate </SelectCriteria>
            <DateType>FilingDate</DateType>
        </ContentSystemInput>
    </ProcessItem>
</ProcessItems>
</ProcessStartParams>]]>

```

RunTime field:

```

<![CDATA[<ProcessStartParams ProcessName="sample string 1">

```



```

<RuntimeFields>
  <RuntimeField>
    <Name>sample string 2</Name>
    <Value>sample string 3</Value>
  </RuntimeField>
</RuntimeFields>
</ProcessStartParams>]]>

```

Credentials:

```

<![CDATA[<ProcessStartParams ProcessName="sample string 1">
<ProcessItems>
<!--FTP Credentials-->
  <ProcessItem>
    <ItemType>Credentials</ItemType>
    <Credentials>
      <Name>sample string 2</Name>
      <FtpCredential>
        <UserName>sample string 3</UserName>
        <Password>sample string 4</Password>
        <Domain>sample string 5</Domain>
        <UseCurrent>true</UseCurrent>
      </FtpCredential>
    </Credentials>
  </ProcessItem>

<!--HTTP Credentials-->
  <ProcessItem>
    <ItemType>Credentials</ItemType>
    <Credentials>
      <Name>sample string 6</Name>
      <HttpCredential>
        <UserName>sample string 7</UserName>
        <Password>sample string 8</Password>
        <Domain>sample string 9</Domain>
        <UseCurrent>true</UseCurrent>
      </HttpCredential>
    </Credentials>
  </ProcessItem>
</ProcessItems>
</ProcessStartParams>]]>

```

Exports:

```

<![CDATA[<ProcessStartParams ProcessName="sample string 1">
<ProcessItems>
<!--Report export-->
  <ProcessItem>
    <ItemType>ExportReport</ItemType>
    <ExportReport>
      <Name>sample string 2 </Name>
      <Destination>sample string 3</Destination>

<ExportFileExistingOptions>Overwrite</ExportFileExistingOptions>
    </ExportReport>
  </ProcessItem>

<!--Table export-->

```



```

<ProcessItem>
  <ItemType>ExportTable</ItemType>
  <ExportTable>
    <Name>sample string 4</Name>
    <OutputType>FileSystem/Database</OutputType>
    <Destination>sample string 5</Destination>
    <FileName>sample string 6</FileName>
    <ExportFileType>Unspecified</ExportFileType>
    <WhenFileExist>Overwrite</WhenFileExist>
    <ConnectionString>sample string 7</ConnectionString>
    <ConnectionStringPassword>sample string
8</ConnectionStringPassword>
    <TableName>sample string 9</TableName>
    <WhenTableExist>Overwrite</WhenTableExist>
    <ByFilesNaming>Unspecified</ByFilesNaming>
    <ByTablesNaming>Filter</ByTablesNaming>
    <CurrentSort>
      <Name>sample string 10</Name>
      <NoSort>>false</NoSort>
    </CurrentSort>
    <CurrentFilter>
      <Name>sample string 11</Name>
      <NoFilter>>false</NoFilter>
      <AllFilters>>false</AllFilters>
    </CurrentFilter>
  </ExportTable>
</ProcessItem>

<!--Summary export-->
<ProcessItem>
  <ItemType>ExportSummary</ItemType>
  <ExportSummary>
    <Name>sample string 12</Name>
    <OutputType>FileSystem/Database</OutputType>
    <Destination>sample string 13</Destination>
    <FileName>sample string 14</FileName>
    <ExportFileType>Unspecified</ExportFileType>
    <WhenFileExist>Overwrite</WhenFileExist>
    <ConnectionString>sample string 15</ConnectionString>
    <ConnectionStringPassword>sample string
16</ConnectionStringPassword>
    <TableName>sample string 17</TableName>
    <WhenTableExist>Overwrite</WhenTableExist>
    <ByFilesNaming>Unspecified</ByFilesNaming>
    <ByTablesNaming>Filter</ByTablesNaming>
    <CurrentSort>
      <Name>sample string 18</Name>
      <NoSort>>true</NoSort>
    </CurrentSort>
    <CurrentFilter>
      <Name>sample string 19</Name>
      <NoFilter>>true</NoFilter>
      <AllFilters>>true</AllFilters>
    </CurrentFilter>
    <CurrentSummary>
      <Name>sample string 20</Name>
      <AllSummaries>>false</AllSummaries>

```

```

        </CurrentSummary>
    </ExportSummary>
</ProcessItem>
</ProcessItems>
</ProcessStartParams>]]>

```

DataPrep items:

Workspace:

```

<![CDATA[<ProcessStartParams ProcessName="sample string 1">
  <ProcessItems>
    <ProcessItem>
      <ItemType>DataPrepWorkspace</ItemType>
      <DataPrepWorkspaceItem>
        <Name>sample string 2</Name>
        <WorkspacePath>sample string 3</WorkspacePath>
      </DataPrepWorkspaceItem>
    </ProcessItem>
  </ProcessItems>
</ProcessStartParams>]]>

```

DataPrep export:

```

<![CDATA[<ProcessStartParams ProcessName="sample string 1">
  <ProcessItems>
<!--CSV export-->
    <ProcessItem>
      <ItemType>DataPrepExport</ItemType>
      <DataPrepExportItem>
        <Name>sample string 2</Name>
        <AllLoadPlans>>false/true</AllLoadPlans>
        <ExportPlanNames>sample string1</ExportPlanNames>
        <ExportPlanNames>sample string2</ExportPlanNames>
        <ExportType>Delimited</ExportType>
        <DelimitedExportOptions>
          <Destination>sample string 4</Destination>
          <!--For MS v13.2 and 13.5-->
          <OverwriteFile>>true</OverwriteFile>
          <!--For MS v14.0+-->
          <WhenFileExist>Overwrite</WhenFileExist>
          <IncludeHeader>>true</IncludeHeader>
          <Delimiter>sample string 5</Delimiter>
          <Qualifier>sample string 6</Qualifier>
        </DelimitedExportOptions>
      </DataPrepExportItem>
    </ProcessItem>

<!--Tableau export-->
    <ProcessItem>
      <ItemType>DataPrepExport</ItemType>
      <DataPrepExportItem>
        <Name>sample string 7</Name>
        <AllLoadPlans>>false/true</AllLoadPlans>
        <ExportPlanNames>sample string1</ExportPlanNames>
        <ExportPlanNames>sample string2</ExportPlanNames>
        <ExportType>Tableau</ExportType>
        <TableauExportOptions>
          <Destination>sample string 9</Destination>
          <!--For MS v13.2 and 13.5-->

```




```

        <OverwriteFile>true</OverwriteFile>
        <!--For MS v14.0+-->
        <WhenFileExist>Overwrite</WhenFileExist>
    </TableauExportOptions>
</DataPrepExportItem>
</ProcessItem>

<!--Qlik export-->
<ProcessItem>
    <ItemType>DataPrepExport</ItemType>
    <DataPrepExportItem>
        <Name>sample string 10</Name>
        <AllLoadPlans>false/true</AllLoadPlans>
        <ExportPlanNames>sample string1</ExportPlanNames>
        <ExportPlanNames>sample string2</ExportPlanNames>
        <ExportType>Qlik</ExportType>
        <QlikExportOptions>
            <Destination>sample string 12</Destination>
            <!--For MS v13.2 and 13.5-->
            <OverwriteFile>true</OverwriteFile>
            <!--For MS v14.0+-->
            <WhenFileExist>Overwrite</WhenFileExist>
        </QlikExportOptions>
    </DataPrepExportItem>
</ProcessItem>

<!--Excel export-->
<ProcessItem>
    <ItemType>DataPrepExport</ItemType>
    <DataPrepExportItem>
        <Name>sample string 13</Name>
        <AllLoadPlans>false/true</AllLoadPlans>
        <ExportPlanNames>sample string1</ExportPlanNames>
        <ExportPlanNames>sample string2</ExportPlanNames>
        <ExportType>MicrosoftExcel</ExportType>
        <ExcelExportOptions>
            <Destination>sample string 15</Destination>
            <!--For MS v13.2 and 13.5-->
            <OverwriteFile>true</OverwriteFile>
            <!--For MS v14.0+-->
            <WhenFileExist>Overwrite</WhenFileExist>
            <TableName>sample string 16</TableName>
            <WhenTableExist>Overwrite</WhenTableExist>
        </ExcelExportOptions>
    </DataPrepExportItem>
</ProcessItem>

<!--Designer export-->
<ProcessItem>
    <ItemType>DataPrepExport</ItemType>
    <DataPrepExportItem>
        <Name>sample string 17</Name>
        <AllLoadPlans>false/true</AllLoadPlans>
        <ExportPlanNames>sample string1</ExportPlanNames>
        <ExportPlanNames>sample string2</ExportPlanNames>
        <ExportType>DatawatchDesigner</ExportType>
        <DesignerExportOptions>

```

```

        <Destination>sample string 19</Destination>
        <!--For MS v13.2 and 13.5-->
        <OverwriteFile>true</OverwriteFile>
        <!--For MS v14.0+-->
        <WhenFileExist>Overwrite</WhenFileExist>
        <TableName>sample string 20</TableName>
        <WhenTableExist>Overwrite</WhenTableExist>
    </DesignerExportOptions>
</DataPrepExportItem>
</ProcessItem>

<!--Access export-->
<ProcessItem>
    <ItemType>DataPrepExport</ItemType>
    <DataPrepExportItem>
        <Name>sample string 21</Name>
        <AllLoadPlans>false/true</AllLoadPlans>
        <ExportPlanNames>sample string1</ExportPlanNames>
        <ExportPlanNames>sample string2</ExportPlanNames>
        <ExportType>MicrosoftAccess</ExportType>
        <AccessExportOptions>
            <Destination>sample string 23</Destination>
            <!--For MS v13.2 and 13.5-->
            <OverwriteFile>true</OverwriteFile>
            <!--For MS v14.0+-->
            <WhenFileExist>Overwrite</WhenFileExist>
            <TableName>sample string 24</TableName>
            <WhenTableExist>Overwrite</WhenTableExist>
        </AccessExportOptions>
    </DataPrepExportItem>
</ProcessItem>

<!--OleDb export-->
<ProcessItem>
    <ItemType>DataPrepExport</ItemType>
    <DataPrepExportItem>
        <Name>sample string 25</Name>
        <AllLoadPlans>false/true</AllLoadPlans>
        <ExportPlanNames>sample string1</ExportPlanNames>
        <ExportPlanNames>sample string2</ExportPlanNames>
        <ExportType>OleDb</ExportType>
        <OleDbExportOptions>
            <IncludePassword>true</IncludePassword>
            <Password>sample string 27</Password>
            <TableName>sample string 28</TableName>
            <WhenTableExist>Overwrite</WhenTableExist>
            <Destination>sample string 29</Destination>
        </OleDbExportOptions>
    </DataPrepExportItem>
</ProcessItem>
</ProcessItems>
</ProcessStartParams>]]>

<!--Cognos export-->
<ProcessItem>
<ItemType>DataPrepExport</ItemType>
    <DataPrepExportItem>

```



```

    <Name>sample string 1</Name>
    <AllLoadPlans>>false/true</AllLoadPlans>
    <ExportPlanNames>sample string1</ExportPlanNames>
    <ExportPlanNames>sample string2</ExportPlanNames>
    <ExportType>Cognos</ExportType>
    <CognosExportOptions>
        <Destination>sample string3</Destination>
        <!--For MS v13.2 and 13.5-->
        <OverwriteFile>>true</OverwriteFile>
        <!--For MS v14.0+-->
        <WhenFileExist>Overwrite</WhenFileExist>
        <TableName>sample string 4</TableName>
        <WhenTableExist>Overwrite/Skip</WhenTableExist>
        <ServerUrl>sample string 5</ServerUrl>
        <Namespace>sample string 6</Namespace>
        <UserName>sample string 7</UserName>
        <Password>sample string 8</Password>
        <IsAnonymous>>true/false</IsAnonymous>
    </CognosExportOptions>
</DataPrepExportItem>
</ProcessItem>

<!--Watson export-->
<ProcessItem>
<ItemType>DataPrepExport</ItemType>
<DataPrepExportItem>
    <Name>sample string 1</Name>
    <AllLoadPlans>>false/true</AllLoadPlans>
    <ExportPlanNames>sample string1</ExportPlanNames>
    <ExportPlanNames>sample string2</ExportPlanNames>
    <ExportType>IBMWatsonAnalytics</ExportType>
    <WatsonExportOptions>
        <Destination>sample string 3</Destination>
        <!--For MS v13.2 and 13.5-->
        <OverwriteFile>>true</OverwriteFile>
        <!--For MS v14.0+-->
        <WhenFileExist>Overwrite</WhenFileExist>
        <TableName>sample string 4</TableName>
        <WhenTableExist>Overwrite/Skip</WhenTableExist>
        <AccessToken>sample string 5</AccessToken>
        <RefreshToken>sample string 6</RefreshToken>
    </WatsonExportOptions>
</DataPrepExportItem>
</ProcessItem>

<!--Power BI export-->
<ProcessItem>
<ItemType>DataPrepExport</ItemType>
<DataPrepExportItem>
    <Name>sample string 1</Name>
    <AllLoadPlans>>false/true</AllLoadPlans>
    <ExportPlanNames>sample string1</ExportPlanNames>
    <ExportPlanNames>sample string2</ExportPlanNames>
    <ExportType>PowerBI</ExportType>
    <PowerBIExportOptions>
        <Destination>sample string 3</Destination>
        <!--For MS v13.2 and 13.5-->

```

```

        <OverwriteFile>true</OverwriteFile>
        <!--For MS v14.0+-->
        <WhenFileExist>Overwrite</WhenFileExist>
        <TableName>sample string 4</TableName>
        <WhenTableExist>Overwrite/Skip</WhenTableExist>
        <AccessToken>sample string 5</AccessToken>
        <RefreshToken>sample string 6</RefreshToken>
    </PowerBIExportOptions>
</DataPrepExportItem>
</ProcessItem>

```

Response Information

Resource Description: None

GET PROCESS STATUS

This returns the status of a visual process provided by a specific tracking ID.

GET method

The entry point for this method is

`api/v1/visualprocesses/{trackingId}/status`

Request Information

The URI parameter is as follows:

Name	Description	Type	Additional information
trackingId	Tracking ID returned from the initial call to the Start Process or Start Process with Manifest methods	globally unique identifier	None

Body Parameters: None

Response Information

Definition of returned process log status:

- "100" PlanBuilding
- "200" Ready
- "300" Running
- "400" Completed



"500" PartiallyFailed
"600" PrerequisitesFailed
"700" ExportFailed

GET PROCESS LOG

This returns the log of a visual process.

GET method

The entry point for this method is
`api/v1/visualprocesses/{trackingId}/log`

Request Information

The URI parameter is as follows:

Name	Description	Type	Additional information
trackingId	Tracking ID returned from the initial call to the Start Process or Start Process with Manifest methods	globally unique identifier	Required

Body Parameters: None

Response Information

Resource Description: None



GET ITEM LOG

This returns the log of a visual process item.

GET method

The entry point for this method is

```
api/v1/visualprocesses/{id}/logitem
```

Request Information

The URI parameter is as follows:

Name	Description	Type	Additional information
Id	ID of a process log's item	integer	Required

Body Parameters: None

Response Information

Resource Description: None



