



Byosphere[®] Automated Processor Guide

December 2025

5.12

Protein Metrics LLC, Boston, Massachusetts, USA

Contents

Introduction.....	3
System Requirements.....	3
Install the Automation Server Package.....	3
Configure the Automation Server.....	4
Configure the Watcher	9
Find Byosphere Upload Folder ID	11
Run as Windows Service.....	11
Windows Credential for Automation Server	13
Application Notes	15
Auto Processor for WA	16
Introduction	16
High-Level Functions	16
System Architecture & Data Flow	16
Current Auto Processor Data Flow	16
Auto Processor for Web Analysis Data Flow	17
Auto Processor for WA Web Configuration	17

Introduction

The Protein Metrics Byosphere® Automated Processor is a server-based product which automatically generates analyses from MS files as they are written to source directories. When MS files are copied to designated source folders, workflows are generated in a designated directory and the MS files used in them are automatically uploaded to the Byosphere server and compressed to the proprietary *.pacq format. New workflows are then generated which use the server sample references and these workflows are then submitted to the server.

The product contains two parts:

- **Watchers** identify newly added or updated MS files and generates workflows based on provided templates, FASTA files and specialized instructions. Different Watchers can be used to create different kinds of workflows based on the kinds of MS files and their source directories.
- **Automation Service**, when in AutoProcess mode, identifies workflows created by the Watchers, uploads and compresses the MS files and submits equivalent workflows as analyses, using the server versions of the MS files. The same Automation Service, when in AutoUpload mode, serves as the Data Uploader, which is available with all Byosphere systems.

For more information about the Data Uploader, see **Byosphere 05 Data Uploader Guide.pdf**.

System Requirements

The minimum hardware/software requirements for the machine that will host the Data Uploader is listed below:

CPU	Memory	Storage	OS	Network
>= 2 cores	>= 16 GB	>= 100 GB	Windows Server 2022	>= 1 GB/s

The machine will require access to the file system with the files/folders to be uploaded.

Install the Automation Server Package

Ideally, the Automation Server should be installed in an environment that is independent from the Analysis Service so that they do not compete for the same resources.

1. Download the Automation Server installer to the local environment and run the installer.
2. On the local environment, run Byosphere to initialize the server URL in the configuration file and test the connection:
 - a) Run `\ProteinMetrics\PMI-Byosphere-Client\Base\PMI-Byosphere-Client.exe`
 - b) Choose **Server > Configure**, enter the Byosphere server URL, and click **OK**

Configure the Automation Server

Modify `appsettings.json` in the installed `PMIAutomationService` directory as follows:

Parameter, Req'd & Default value	Note
<i>Parameter:</i> Host <i>Required:</i> Yes	Byosphere Application Server IP address
<i>Parameter:</i> UseAPIToken <i>Required:</i> Yes, if using API Token	If true, set Windows Credential to use API token set on Application Server (see below). An API token is recommended for Byosphere v 4.1 and above.
<i>Parameter:</i> KeychainNameSpace <i>Default:</i> PmiAutomationCredentials	Must be consistent with the credentials set up using <code>PMI-Keychain-Manager.exe</code> in Windows Credential Store; see notes below
<i>Parameter:</i> LocalWorkingPath <i>Default:</i> <code>Environment.GetFolderPath(Environment.SpecialFolder.CommonApplicationData)\ProteinMetrics\AutomationServer</code>	Folder location to store temporary files generated and then deleted during uploads. The default value will be used if the customer value is not specified or accessible.
<i>Parameter:</i> ExecPath <i>Default:</i> <code>Environment.GetFolderPath(Environment.SpecialFolder.ProgramFiles)\ProteinMetrics\PMI-Byosphere-Client\Base</code>	The <code>console.exe</code> executable path from the Protein Metrics Byosphere Client software installation
<i>Parameter:</i> BaseDir <i>Required:</i> On prem only	This is a required parameter for OnPrem only. This is the NAS mount drive. This setting in Windows must match its counterpart in App Server's <code>base_dir</code> in Linux. Example: <code>"BaseDir": "C:"</code>
<i>Parameter:</i> AutomationConfigurationID	The value is the Automation configuration ID in the Admin Portal. This is a required parameter for Data Uploader and Auto Processor. It is not required for Watcher.

Parameter, Req'd & Default value	Note
<i>Parameter:</i> ProjectUploadFolderID <i>Required:</i> Yes <i>Default:</i> 0	ID of the Byosphere server destination folder for submitted analyses using the Automated Processor; The user specified by the API token must be a Super User to create a root-level folder (the default value ID=0), or have File Editor, Folder Editor and Viewer privileges for a specified Byosphere Web Client folder. For the Automated Processor, project files created by the analysis are uploaded to this folder tree. Output Folder and ProjectUploadFolderID can be the same folder. To find a folder's ID, see the Find Byosphere Upload Folder ID section below.
<i>Parameter:</i> AutomatedMode	This parameter is required for Watcher. To run the executable as a Watcher, set it as: <code>"AutomatedMode": "BuildWflw",</code> In this mode, samples, workflow templates and other inputs are used to generate workflows in the directory defined by the Output parameter (see the Configure the Watcher section below).

Parameter, Req'd & Default value	Note
<i>Parameter:</i> SearchPattern <i>Required:</i> Yes	Extensions of all files (including MS files) to be uploaded. all other file types will be ignored. If AutoProcess is true, only MS files will be uploaded.
<i>Parameter:</i> Comment	Optional text to save in the Comment field of the uploaded file version
<i>Parameter:</i> DeleteAfterUpload <i>Default:</i> false	If true, input file or directory will be deleted upon successful upload to save space. Only SearchPattern/MSFile matched files/folders are deleted.
<i>Parameter:</i> WriteLastScan <i>Default:</i> false	If true, last scan time will be auto-created / updated by the program and written to the configuration file. Note: when set to true, the LastScanTime line should be removed (along with the comma after "WriteLastScan": true) before restarting the service.
<i>Parameter:</i> LastScanTime <i>Required:</i> No	This this value is auto-created / updated by the last program run. If both Modified After and LastScanTime exist, later of the two values will be in use. Format is "yyyy-MM-dd HH:mm:ss" Example: "LastScanTime": "2021-06-01 19:58:56" (HH is a 24 hour clock)
<i>Parameter:</i> RollingWindowDelta	The unit is second. The Automation will check for new data using time stamp setup in Modified After or LastScanTime. However, sometimes NAS data is not immediately available for cloud instance to be picked up. The delta window makes the Modified After or LastScanTime more tolerant in such cases. If there is data with timestamp 12-20-22 11:50 AM, data will not be uploaded using "Modified After": "2022-12-20 12:30:00". Data will be uploaded using the following, given a 3600 second or 1 hour delta effectively checking the file against 12-20-22 11:30 AM. "RollingWindowDelta": 3600, "Modified After": "2022-12-20 12:30:00" If no value is specified, no delta window will be applied. Example: "RollingWindowDelta": 3600
<i>Parameter:</i> ApplyMetadataToExistingFile	When ApplyMetadataToExistingFile is true, additional metadata might be applied to a file already in Byosphere. When the value is false, the Byosphere file will not be updated with metadata. Default value is false. Example: "ApplyMetadataToExistingFile": true

Parameter: CustomSettings

CustomSettings is an array of custom codes. Custom code will be executed in the order they are listed. Parameter includes

- CustomCodeExePath, optional CustomCode, optional MetadataPath. This set is executed for each sample.
- CustomCodeExePathBefore, optional CustomCodeBefore, optional MetadataPathBefore. This set is executed before Automation starts scanning for new data.
- CustomCodeExePathAfter, optional CustomCodeAfter, optional MetadataPathAfter. This set is executed after Automation has processed current scan data.
- CustomCodeExePathAutomationTrigger, optional CustomCodeAutomationTrigger, optional MetadataPathAutomationTrigger. This set is executed for each Automation Trigger file.

MetadataPath can contain output from the custom code. Contact support@proteinmetrics.com for more information on integration with third party tools through custom codes.

Examples,

- First custom code executes python code,
- Second custom code executes python code with a parameter,
- Third example is an executable,
- Fourth example is an executable with a parameter and a metadata output path

```
"CustomSettings": [
  {
    "CustomCodeBeforeExePath": "py-
thon.exe",
    "CustomCodeBefore":
    "PMIGetMetadata.py"
  },
  {
    "CustomCodeAfterExePath": "py-
thon.exe",
    "CustomCodeAfter": "\"D:\\Test
Data\\PMIGetFasta.py\" --protein"
  },
  {
    "CustomCodeAutomationTrigger":
    "GenedataIntegration.exe"
  },
  {
```

	<pre> "CustomCodeExePath": "BioviaIntegra- tion.exe", "CustomCode": "-f", "MetadataPath": "E:\\Pro- gramData\\ProteinMetrics\\BioviaMetadata" }], </pre>
Parameter: ApplySampleMetadataToProject	False When True, the metadata for the first sample added to a workflow is automatically included in the project.
Parameter: LocalInputRoot (optional)	<p>Automation by default honors local folder hierarchy and will create a matching folder hierarchy in Byosphere. LocalInputRoot parameter allows user to define common path that will be skipped during Byosphere subfolder creation processor therefore redefine a root folder.</p> <p>LocalInputRoot is case insensitive. LocalInputPath and LocalInputRoot are expected to have the same path format of Windows, Linux, or UNC path.</p> <p>Examples in Linux or Windows filepath format:</p> <p>LocalInputPath c:/a/b/z, LocalInputRoot c:/a/b => common path c:/a/b -> data will be uploaded to Output Folder > z</p> <p>LocalInputPath c:/a/b, LocalInputRoot empty => no common path -> data will be uploaded to Output Folder > a > b</p> <p>LocalInputPath c:\a\b, LocalInputRoot c:\m => no common path -> data will be uploaded to Output Folder > a > b</p> <p>LocalInputPath c:\a\b\z, LocalInputRoot c:\a\b\z\y => common path c:/a/b/z -> data will be uploaded to Output Folder</p>
Parameter: UserGroupIds (optional)	<p>Specifies the ids of user groups that should be assigned to the new folders created when uploading a file. The groups will be assigned to all uploaded folders or to subfolders from the level specified by UserGroupSubfolderLevel.</p> <p>Example:</p> <p>"UserGroupIds": "2,3"</p>
Parameter: UserGroupSubfolderLevel (optional) Default: 1	<p>Specifies the level of LocalInputPath directory from where UserGroupIds should be applied when uploading folders.</p> <p>When this parameter is not provided or its value is less than or equal to 1, UserGroupIds will be applied to all new folders.</p> <p>Examples:</p> <p>"LocalInputPath": "C:/a/b/c/d/e",</p> <p>"UserGroupIds": "2,3"</p>

	<p>Example 1:</p> <p>"UserGroupSubfolderLevel ": 1</p> <p>Result: User groups will be applied to all subfolders: a,b,c,d,e</p> <p>Example 2:</p> <p>"UserGroupSubfolderLevel ": 3</p> <p>Result: User groups will be applied to the following subfolders: c,d,e</p> <p>Example 3 (when LocalInputRoot is provided):</p> <p>"LocalInputRoot": "C:/a/b/c",</p> <p>"UserGroupSubfolderLevel ": 3</p> <p>Result: User groups will be applied to the following subfolders: d,e (subfolder 'c' will not be uploaded due to LocalInputRoot value)</p>
--	--

Note: The Automation Server supports metadata. The configuration can be customized so that the folder and file names are parsed into strings which can be saved with the file to specific metadata fields. For information about metadata customization, contact support@proteinmetrics.com.

Configure the Watcher

The installed PMIAutomationService folder is also used by the Watchers. After configuring this folder for the Automation Service, make a copy of the folder for each Watcher and name them accordingly. Next, configure the `appsettings.json` in each of the Watcher directories. Most of the Watcher `*.json` parameters that are shared with the Automation Server `*.json` parameters above will be assigned the same values. There are also specialized parameters for the Watchers to configure as follows:

Parameter, Req'd & Default value	Note
<i>Parameter:</i> LocalInputPath <i>Required:</i> Yes	The path of the folder or root folder of subfolders containing the MS files to be uploaded and processed as analyses.
<i>Parameter:</i> SubFolderLevel <i>Default:</i> 1	The subfolder level of the LocalInputPath directory where the MS files are found. For a value of 1, the MS files are found in the LocalInputPath directory; for a value of 2, the MS files are in folders under that directory, etc.
<i>Parameter:</i> ProcessEntireFolder <i>Default:</i> false	If true, all the MS files in folder at the SubFolderLevel as well as its subfolders are loaded into a single workflow Note: This must be set to false for *.raw and *.d folder samples.
<i>Parameter:</i> WflwOutputDir <i>Required:</i> Yes	The directory where workflows are generated when BuildWflw is set to true, typically the LocalInputPath directory specified for the Automation Service
<i>Parameter:</i> WflwTemplate <i>Required:</i> Yes	The path or file name of the workflow file containing the workflow parameters, apart from samples, which are loaded from the LocalInputPath tree, and sequences, which are loaded from the Sequence path and fasta file.
<i>Parameter:</i> Sequence <i>Required:</i> Yes	The path or file name of the fasta file(s) containing any sequences to load.
<i>Parameter:</i> ReportConfig <i>Required:</i> No	The full path and file name of the report configuration *.rptc file to be loaded into the workflows. If no report configuration is specified, the default report for that project type (*.blgc, *.ntms or *.bmap) Note: Local report configuration files can be added directly to the workflow templates. This is recommended if the Watcher applies to more than one workflow with different report types. In that use case, use absolute paths since relative paths are not supported.
<i>Parameter:</i> SampleProteinCSV <i>Required:</i> No	The full path and file name of the sample-protein *.csv file to be loaded into the workflow.
<i>Parameter:</i> CustomJS <i>Required:</i> Yes	The path and file name of a Custom.js file containing instructions for naming and parsing files and workflows. For help with customizing generated workflows, contact support@proteinmetrics.com
<i>Parameter:</i> AutomationTrigger <i>Required:</i> No	When ProcessEntireFolder is true , a more precise way to trigger the Watcher is to use an automation trigger file. A sub-folder will be processed if and only if the filename designated as the AutomationTrigger is present in the folder. Example: "AutomationTrigger": "WatersmabEnd.raw"

Parameter:
ContinueAfterCustomCode
Required: No
Default: true

If true, we will continue with BuildWflw after custom code is executed. If false, no action will be taken after the custom code is executed.

Find Byosphere Upload Folder ID

Byosphere folder IDs are displayed when uploading to or downloading from a selected folder. To identify the folder ID, do the following:

1. Create the upload folder in Byosphere. If the Keychain credentials for the Data Uploader is not a Super User, then add a user group for that user name with File Editor, Folder Editor and Viewer privileges.
2. In Byosphere Desktop, login to the server and choose **File > Upload**.
3. In the **Server file/folder** cell, click the "..."/>

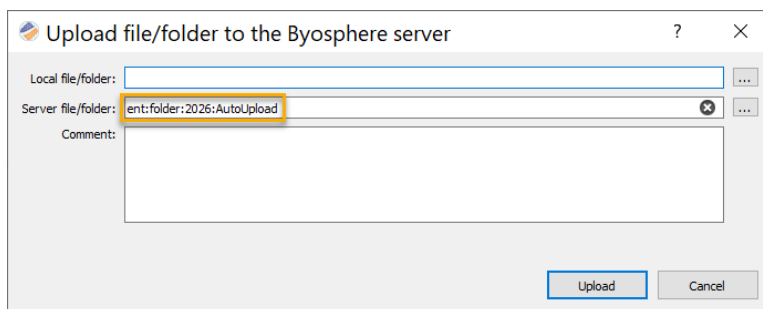


Figure 1: Identify the Byosphere upload folder ID

Run as Windows Service

The Protein Metrics Automation Server and the Watchers are recommended to run as Windows Services rather than as console applications. The executable sc.exe in CMD can also be used to create, start, stop, or delete a Windows Service:

:: Create a Windows Service for the Automation Server

```
sc create PmiAutomationService DisplayName="PMI Automation Server"
binPath="C:\Program Files\ProteinMetrics\Pmi-Automation-
Service\PmiAutomationService.exe"
```

:: Start a Windows Service

```
sc start PmiAutomationService
```

:: Stop a Windows Service

```
sc stop PmiAutomationService
```

:: Delete a Windows Service

```
sc delete PmiAutomationService
```

Repeat these steps for each Watcher Server. After configuring and verifying the setup, it is recommended to set the Protein Metrics Automation Server **Start Type** to **Auto**. Windows services can be managed in the Windows Services App:

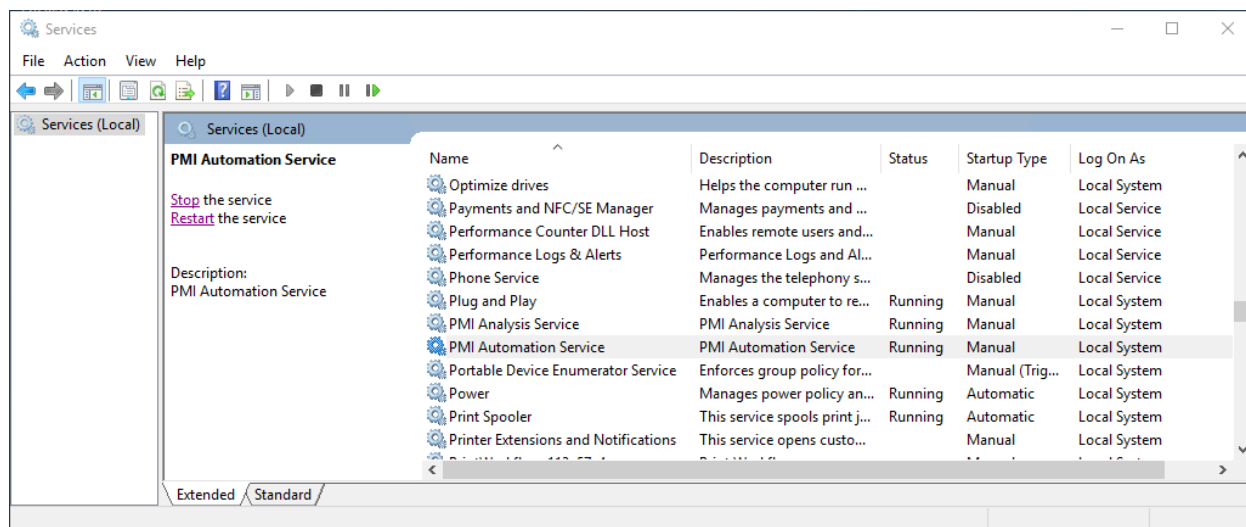


Figure 2: Window Services App

The Log On account for Protein Metrics Automation Service can be a Local Service:

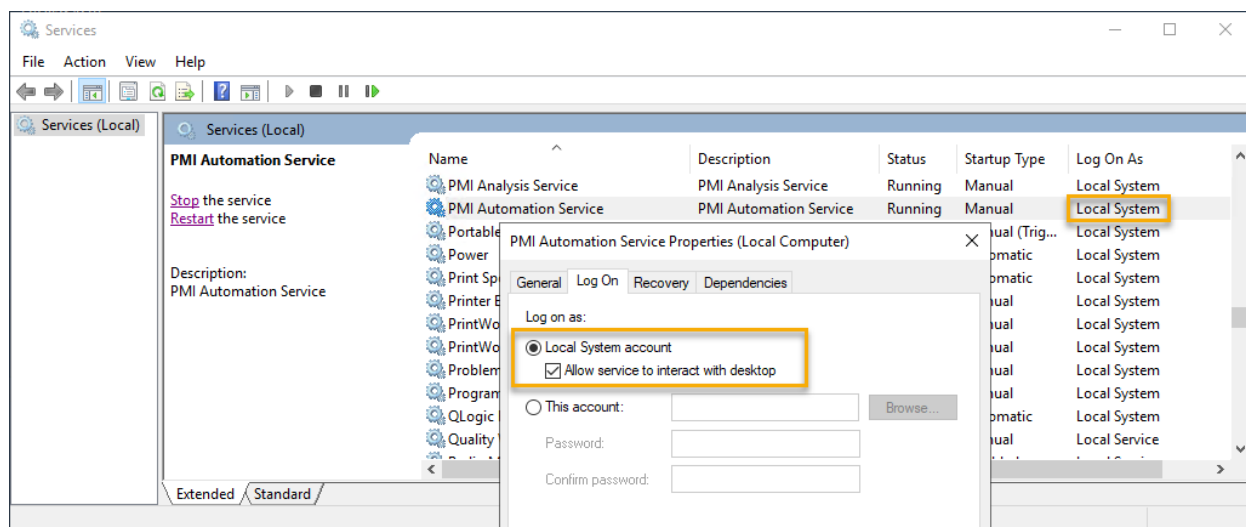


Figure 3: Protein Metrics Automation Service Local logon account

Alternatively, the Log On account for Protein Metrics Automation Service can be an individual user:

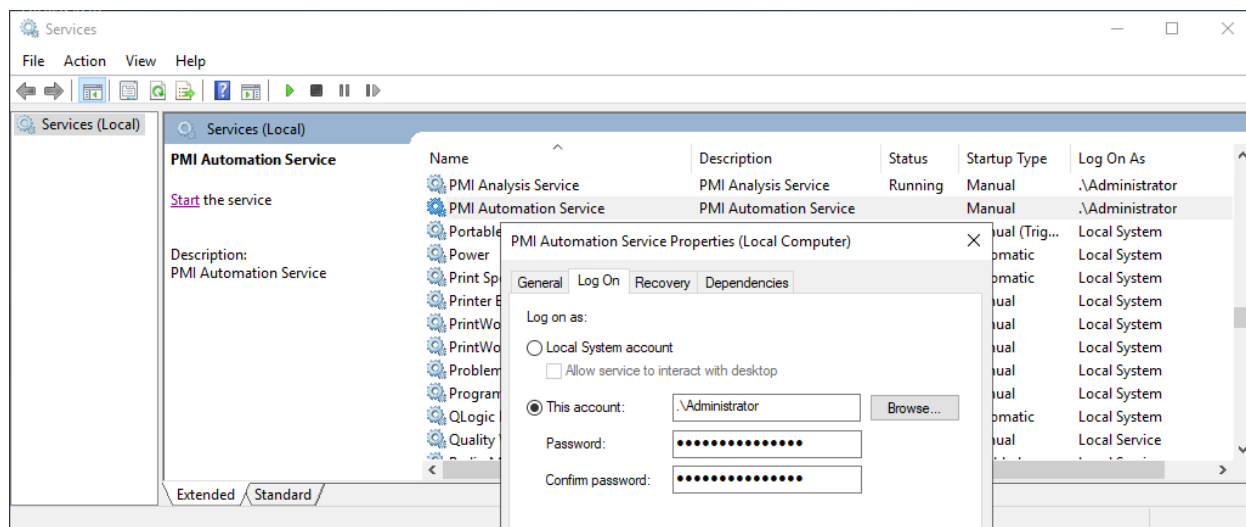


Figure 4: Protein Metrics Automation Service Administrator logon account

Note: After verifying the setup, Automation Server failure recovery can be adjusted according to Company IT policies.

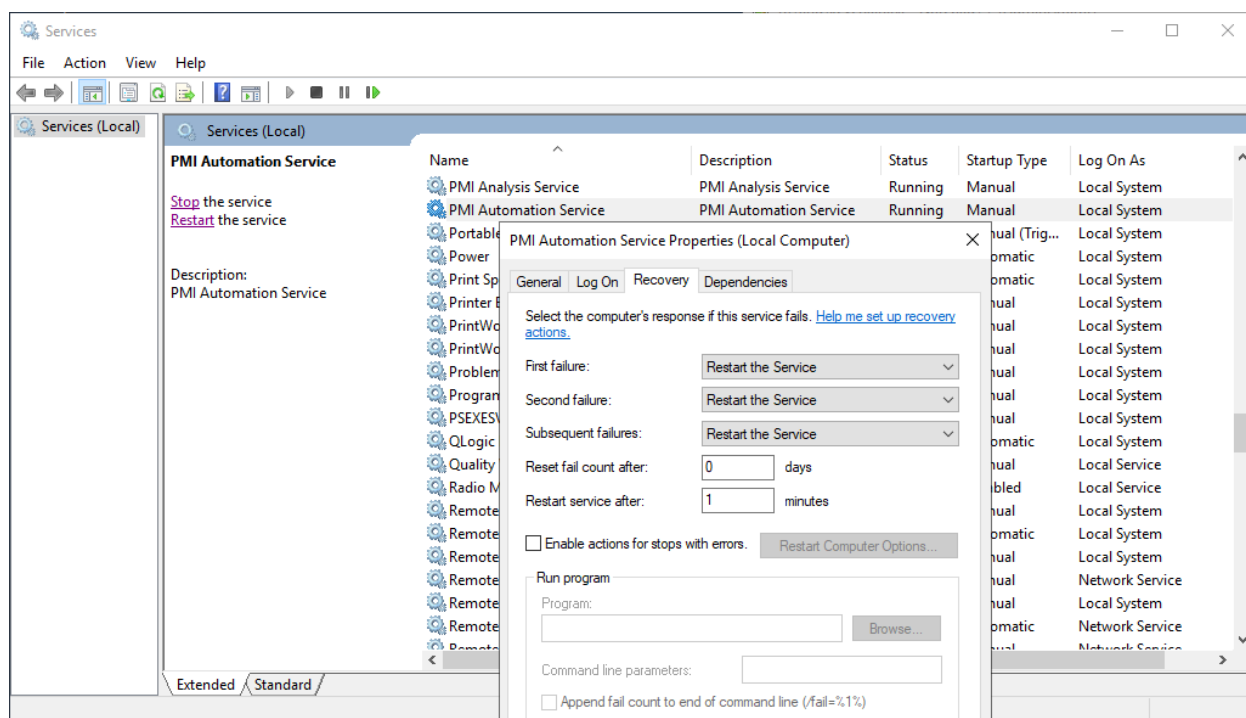


Figure 5: Automation Server failure recovery

Windows Credential for Automation Server

Automation Server uses Windows Credential Store to avoid exposing user passwords in the configuration. Commands are run in the CMD console. For Administrator log on services, a simple CMD window is used. For Local System log on services, there are many tools to access Local System account. One way is to use <https://docs.microsoft.com/en-us/sysinternals/downloads/pstools>

- Start CMD, Run as administrator

- Execute the following command:
`psexec -i -s cmd.exe`
- This will launch a new CMD window granting Local System account access
- Windows Credentials setup can be executed from the Local System account

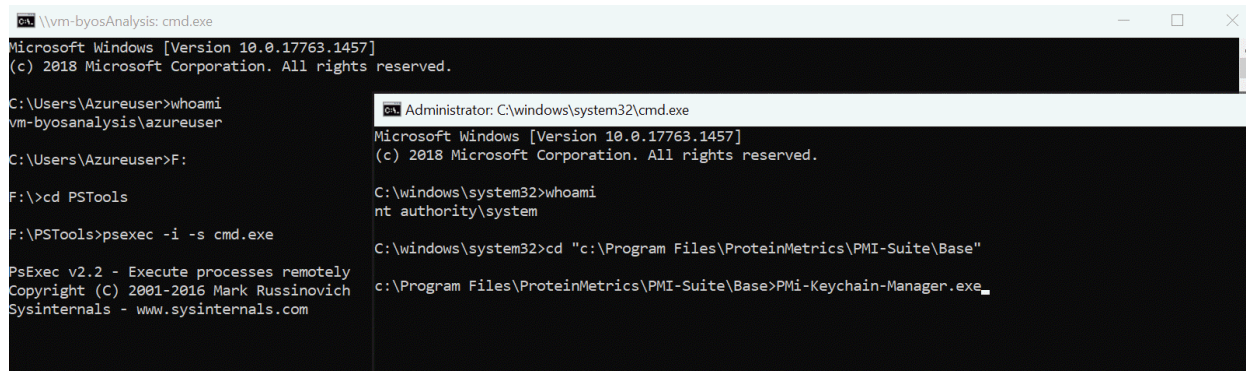


Figure 6: Accessing Local System account

To configure the password used in App Server authentication for Automation Server:

- Follow App Server Guidance on setup API Token
- Open a CMD window as shown above.
- Customize and run the following commands:

```
PMi-Keychain-Manager --delete --namespace=PmiAutomationCredentials
PMi-Keychain-Manager --add --namespace=PmiAutomationCredentials --
key=server --value="https://byosphere.company.com"
PMi-Keychain-Manager --add --namespace=PmiAutomationCredentials --
key=api-token --value=ABCDEabcde
PMi-Keychain-Manager --read --namespace=PmiAutomationCredentials --
key=api-token --context-data="server=https://byosphere.company.com"
```

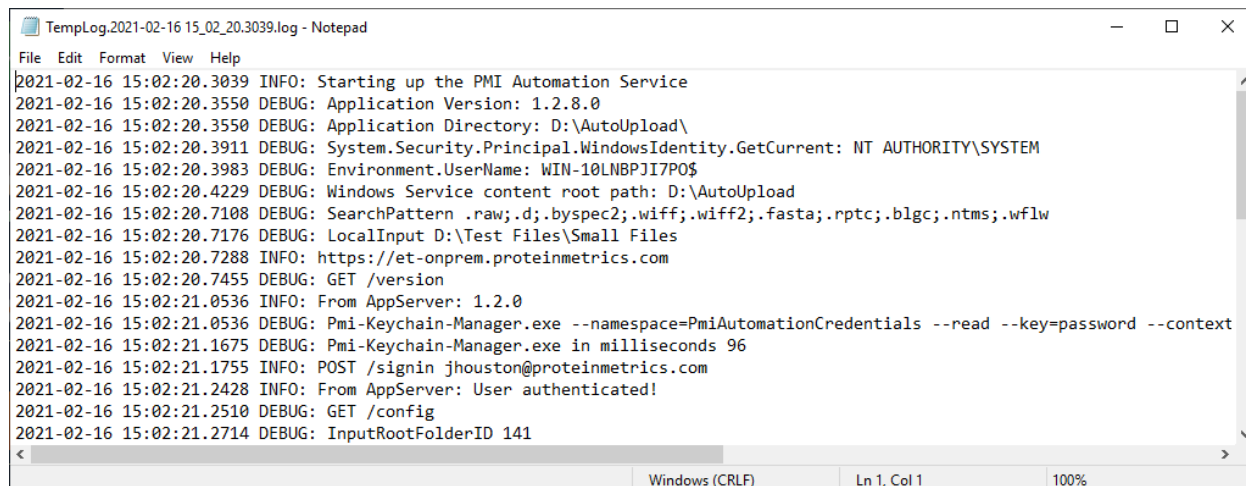
The final line will display the API Token value for the Byosphere server URL. This user account assigned to the API Token requires the Super User privilege and the Contributor entitlement.

Note: Credentials only need to be setup one. In the future updates, run the last line to verify.

Note: Automation Server is set to log on as **Local System** by default. Depending on Company IT policies, Windows Credential might be necessarily to setup from Local System account. To find out more, please refer to <https://docs.microsoft.com/en-us/windows/win32/services/localsystem-account> and

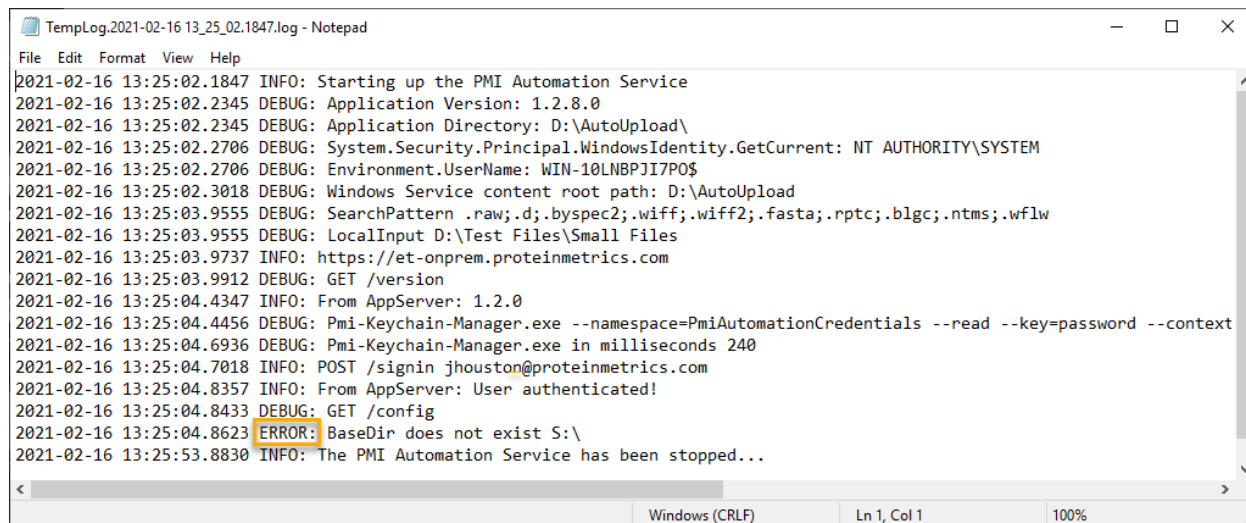
<https://docs.microsoft.com/en-us/windows/win32/ad/the-localsystem-account>

Example of a successful Automation Server startup using username and password: expect to see no errors:



```
TempLog.2021-02-16 15_02_20.3039.log - Notepad
File Edit Format View Help
2021-02-16 15:02:20.3039 INFO: Starting up the PMI Automation Service
2021-02-16 15:02:20.3550 DEBUG: Application Version: 1.2.8.0
2021-02-16 15:02:20.3550 DEBUG: Application Directory: D:\AutoUpload\
2021-02-16 15:02:20.3911 DEBUG: System.Security.Principal.WindowsIdentity.GetCurrent: NT AUTHORITY\SYSTEM
2021-02-16 15:02:20.3983 DEBUG: Environment.UserName: WIN-10LNBPJ17P0$
2021-02-16 15:02:20.4229 DEBUG: Windows Service content root path: D:\AutoUpload
2021-02-16 15:02:20.7108 DEBUG: SearchPattern .raw;.d;.byspec2;.wiff;.wiff2;.fasta;.rptc;.blgc;.ntms;.wflw
2021-02-16 15:02:20.7176 DEBUG: LocalInput D:\Test Files\Small Files
2021-02-16 15:02:20.7288 INFO: https://et-onprem.proteinmetrics.com
2021-02-16 15:02:20.7455 DEBUG: GET /version
2021-02-16 15:02:21.0536 INFO: From AppServer: 1.2.0
2021-02-16 15:02:21.0536 DEBUG: Pmi-Keychain-Manager.exe --namespace=PmiAutomationCredentials --read --key=password --context
2021-02-16 15:02:21.1675 DEBUG: Pmi-Keychain-Manager.exe in milliseconds 96
2021-02-16 15:02:21.1755 INFO: POST /signin jhouston@proteinmetrics.com
2021-02-16 15:02:21.2428 INFO: From AppServer: User authenticated!
2021-02-16 15:02:21.2510 DEBUG: GET /config
2021-02-16 15:02:21.2714 DEBUG: InputRootFolderID 141
```

Figure 7: Example of Successful Automation Server startup



```
TempLog.2021-02-16 13_25_02.1847.log - Notepad
File Edit Format View Help
2021-02-16 13:25:02.1847 INFO: Starting up the PMI Automation Service
2021-02-16 13:25:02.2345 DEBUG: Application Version: 1.2.8.0
2021-02-16 13:25:02.2345 DEBUG: Application Directory: D:\AutoUpload\
2021-02-16 13:25:02.2706 DEBUG: System.Security.Principal.WindowsIdentity.GetCurrent: NT AUTHORITY\SYSTEM
2021-02-16 13:25:02.2706 DEBUG: Environment.UserName: WIN-10LNBPJ17P0$
2021-02-16 13:25:02.3018 DEBUG: Windows Service content root path: D:\AutoUpload
2021-02-16 13:25:03.9555 DEBUG: SearchPattern .raw;.d;.byspec2;.wiff;.wiff2;.fasta;.rptc;.blgc;.ntms;.wflw
2021-02-16 13:25:03.9555 DEBUG: LocalInput D:\Test Files\Small Files
2021-02-16 13:25:03.9737 INFO: https://et-onprem.proteinmetrics.com
2021-02-16 13:25:03.9912 DEBUG: GET /version
2021-02-16 13:25:04.4347 INFO: From AppServer: 1.2.0
2021-02-16 13:25:04.4456 DEBUG: Pmi-Keychain-Manager.exe --namespace=PmiAutomationCredentials --read --key=password --context
2021-02-16 13:25:04.6936 DEBUG: Pmi-Keychain-Manager.exe in milliseconds 240
2021-02-16 13:25:04.7018 INFO: POST /signin jhouston@proteinmetrics.com
2021-02-16 13:25:04.8357 INFO: From AppServer: User authenticated!
2021-02-16 13:25:04.8433 DEBUG: GET /config
2021-02-16 13:25:04.8623 ERROR: BaseDir does not exist S:\
2021-02-16 13:25:53.8830 INFO: The PMI Automation Service has been stopped...
```

Figure 8: Example of Failed Automation Server startup

Application Notes

- New files are identified by content, not by name. If a file is edited, it will be uploaded as new version over the existing file with that name.
- Files which have MS sample extensions but are not recognized by the Byos application as MS files or folders (for example, corrupted data) are uploaded but not compressed into *.pacq format. Multiple versions of these files may then be generated.
- Log files written to the PMIAutomationService\Logs directory can be renamed or configured by editing the file PMIAutomationService\NLog.config. For more information, please refer to <https://github.com/nlog/nlog/wiki/Configuration-file> or contact Protein Metrics support. Note that log level can be Debug, Info, Warning, Error, and Fatal. It is recommended to start with Debug level logging during initial deployment to enable Protein Metrics to better troubleshoot any issues. It is recommended to use Info in normal operation; Warning or high to minimize logging. User can delete log files at PMIAutomationService\Logs subdirectory if log information is no longer needed.

Auto Processor for WA

The current Automation suite successfully provides "Data Watcher" and "Auto Processor" functionalities, enabling automated processing of datasets into Byos workflows (.wflw). Auto Processor for Web Analysis extends this automation capability directly into the next-generation Byosphere Web Analysis (WA) platform.

This new automation type will monitor local file systems, apply sophisticated business logic to group and prepare data, and automatically create, compute, and publish Web Analysis projects. It is designed to provide a future-proof path for all customers adopting the Web Analysis platform. A key innovation is its flexible customization model, which uses JSON-based input/output and Python scripts to handle complex business logic like sample grouping and template selection.

Introduction

The shift from desktop-centric workflows to the cloud-native Byosphere Web Analysis platform requires a corresponding evolution in automation tools. The current Auto Processor creates projects that are ingested into the Deep Query (DQ) database, which is incongruent with the WA architecture.

Auto Processor for Web Analysis bridges this gap. It combines the monitoring capabilities of Data Watcher with the processing logic of Auto Processor, re-engineered to leverage the Web Analysis REST API. This provides customers with:

Seamless Transition: A familiar automation experience for users moving from Byos workflows to Web Analysis.

Operational Efficiency: Eliminates manual steps in creating and publishing analyses for recurring data types.

Scalability: Leverages the cloud-based Byosphere platform for compute-intensive analysis.

Dual-Mode Support: Organizations can run Auto Processor (for Byos workflows) and Auto Processor for Web Analysis side-by-side. A dataset will be processed by one or the other, but not both.

High-Level Functions

1. Monitor specified input directories (local file/folder/NAS).
2. Apply configurable business logic to group samples and select appropriate WA templates.
3. Automatically acquire samples and data (e.g., FASTA files) into Byosphere.
4. Create a new Web Analysis project using a selected template.
5. Populate the analysis with the acquired samples and sequences.
6. Submit the compute and publish operation as a WA job.
7. To provide user feedback, the automation will send notification once a WA job is submitted.
8. Provide a configuration experience consistent with existing Automation features.
9. Support advanced customization via Python scripts using a JSON-in/JSON-out model.

System Architecture & Data Flow

The following data flow illustrates the new process for Web Analysis, contrasting with the existing Byos flow.

Current Auto Processor Data Flow

1. Data Watcher monitors the Input Directory.

2. Business Logic (via custom.js) determines workflow templates and sample grouping.
3. Workflow Manipulation: Samples are added to local .wflw templates; sequences are added.
4. Populated workflows are placed in a Wflw Output Directory.
5. Auto Processor picks up the .wflw file, uploads data to Byosphere, translates local file references to ent references, and submits the job.

Auto Processor for Web Analysis Data Flow

The new process merges and modifies the above steps to work with the WA API and centralized customization hooks. Key architectural changes include:

- No Local Workflows: Replaces local .wflw templates with cloud-based WA templates.
- Early Data Acquisition: Samples and FASTA files are uploaded to Byosphere to obtain their ent (enterprise) references.
- Direct Publication: The analysis is computed and published directly within Byosphere.

Auto Processor for WA Web Configuration

Parameter	Required	Default	Note	Example
Type		Auto Processor		
Name	Yes		Automation Web Configuration name	Data Uploader for QA testing
Input Directory	Yes		Local input directory	D:\AutoProcessing\Data
Output Folder	Yes		Byosphere Output Folder	Samples
Modified After		1 year before the current date when the service is setup	Used to filter out incoming data for consideration	2/20/2020 14:22
Polling Interval		3600 seconds	An integer defines how often the Automation service will query for data. The unit is seconds. The default is 3600 seconds or 1 hour	3600
Parsing Rule			User can select one or more predefined parsing rules, see Parsing Rules section for details.	
Description			User-entered description of the configuration	Automation for QA testing
Advanced Properties				
Project Upload Folder			Byosphere Output Folder for WA projects	Projects
Apply Sample		TRUE	If True, apply the first sample metadata to the created project	

Metadata to Project				
Apply Metadata to Project Excludes			An array of Byosphere Metadata Fields that will be excluded when we apply sample's metadata to a project. Parameter is applied only when ApplySampleMetadataToProject is true.	["plate_position", "sample_code"]
Apply Metadata to Existing File		FALSE	If True, additional metadata might be applied to a file already in Byosphere. When the value is false, the Byosphere file will not be updated with metadata. Default value is false.	
Metadata Path			Local directory for metadata files	D:\PMIMetadata
Write Last Scan		TRUE	If True, last scan time will be auto-created / updated by the program and written to the configuration file.	
Integration			User can select one or more predefined Chromeleon Integration, UNIFI Integration, or Genedata Biologics Integration	
Process Entire Folder		TRUE	If true, all the MS files in folder at the SubFolderLevel as well as its subfolders are loaded into a single WA	
Sub Folder Levelpro		1	The subfolder level of the LocalInputPath directory where the MS files are found. For a value of 1, the MS files are found in the LocalInputPath directory; for a value of 2, the MS files are in folders under that directory, etc.	
Sample Protein CSV			The full path and file name of the sample-protein *.csv file to be loaded into the WA.	
Search Pattern		.raw;.d;.wiff;.wiff2;.byspec2	Used to filter out incoming data by file extension	
WA Template Folder			Byosphere WA template folder	
WA Template files			Byosphere WA template files	
Sequence File or Directory			The path or file name of the fasta file(s) containing any sequences to load.	
Custom Code Directory			The path and file name of a Custom.py file containing instructions for processing samples. For help with customizing generated workflows, contact support@proteinmetrics.com	

Automation Trigger			<p>When ProcessEntireFolder is true, a more precise way to trigger the Watcher is to use an automation trigger file.</p> <p>A sub-folder will be processed if and only if the filename designated as the AutomationTrigger is present in the folder.</p>	*standby.raw
RollingWindowDelta		0	<p>The unit is second. The Automation will check for new data using time stamp setup in Modified After or LastScanTime. However, sometimes server data is not immediately available for cloud instance to be picked up. The delta window makes the Modified After or LastScanTime more tolerant in such cases.</p>	
Folder Starts With			Include a folder if the beginning of the name matches the specified string.	
Folder Ends With			Include a folder if the end of the name matches the specified string.	
Folder Includes			Include a folder if a specified string occurs within its name.	
Folder Excludes			Exclude a folder if a specified string occurs within its name.	
File Starts With			Include a file if the beginning of the name matches the specified string.	
File Ends With			Include a file if the end of the name matches the specified string.	
File Includes			Include a file if a specified string occurs within its name.	
File Excludes			Exclude a file if a specified string occurs within its name.	
Custom Codes			<p>CustomSettings is an array of custom codes. Custom code will be executed in the order they are listed. Parameter includes CustomCodeExePath, optional CustomCode, optional MetadataPath, CustomCodeTemplateProcessExePath, optional CustomCodeTemplateProcess. This set is executed for WA creation. MetadataPath can contain output from the custom code. Contact support@proteinmetrics.com for more information on integration with third party tools through custom codes.</p> <p>Examples,</p> <p>First custom code executes python code,</p>	

			<p>Second custom code executes python code with a parameter,</p> <p>Third example is an executable,</p> <p>Fourth example is an executable with a parameter and a metadata output path</p> <pre> "CustomSettings": [{ "CustomCodeExePath": "python.exe", "CustomCode": "PMIGetMetadata.py" }, { "CustomCodeExePath": "python.exe", "CustomCode": "\"D:\\Test Data\\PMIGetFasta.py\" --protein" }, { "CustomCodeExePath": "GenedataIntegration.exe" }, { "CustomCodeExePath": "BioviaIntegration.exe", "CustomCode": "-f", "MetadataPath": "E:\\ProgramData\\ProteinMetrics\\BioviaMetadata" }, { "CustomCodeTemplateProcess": "D:\\custom.py", "CustomCodeTemplateProcessExePath": "C:\\Python312\\python.exe" }], </pre>	
System Metadata Fields			An array of System Metadata Fields that Automation will set for each sample based	

			<p>on sample metadata found at Metadata Path</p> <p>Example: ["is_biophysical", "instrument", "project", "c_b_number", "c_c_code", "c_channel_id", "c_plate_barcode"],</p>	
Manual Custom Fields			<p>An array of Manual Custom Fields that Automation will set for each sample based on sample metadata found at Metadata Path</p> <p>Example: ["LotNumber", "Analyst Workbook Autoname", "Analyst Username", "Task ID"]</p>	
Metadata Mapping RegEx			<p>Regular expressions are used to match samples to .csv column data. The .csv column is defined in the MetadataMappingRegExField. We'll apply one regular expression (MetadataMappingRegExSample) to the sample, and another (MetadataMappingRegExMetadata) to the column's cell. If there's a match, we'll use that row's data to add metadata. Multiple .csv columns can be mapped to Byosphere metadata fields.</p>	<pre>{ "Sample": "Explores-02\\.*\\..*?\\..*?-(?ccode'[A-Za-z0-9]+)_[A-Za-z]+[A-Za-z0-9]+.raw", "Metadata": "(?ccode'[A-Za-z0-9]+)", "Field": "SampleName/Ccode" }</pre>
Metadata Settings			<p>This parameter contains an array of settings for metadata. In the following example, we have 3 file naming conventions based on regular expressions under MetadataPattern and a set of key value pairs with keys being the Byosphere Metadata Fields, the value can be from a metadata .tsv file.</p> <pre>[{ "MetadataPattern": "\\(?(instrument'RapidFire)\\)\\(Pipeline\\(?(project'.*)\\)\\(?(injection'.*)\\)\\(?(c_plate_barcode'.*)\\)\\(?(assay'[A-Za-z]+)\\)\\(?(plate_position'[A-Za-z0-9]+).d" }, { "MetadataPattern": "\\(?(instrument'Explores-02)\\)\\(?(project'.*)\\)\\(?(request_id'.*)\\)\\(?(protein'.*)\\)\\(?(info'.*)\\)\\(?(assay'[A-Za-z]+)\\)\\(?(plate_position'[A-Za-z0-9]+).raw"</pre>	

			<pre> }, { "MetadataPattern": "\\(\\('instrument'Exploris- 01)\\)\\(\\('info1'.*)\\)\\(\\('digestion'.*)\\)\\(\\('reque st_id'.*?\\)_\\('protein'.*?\\)- \\('info'.*?\\)_\\('assay'[A-Za- z]+\\)\\('plate_position'[A-Za-z0-9]+\\).raw" }, { "request_id": "AD#", "c_plate_barcode": "Plate Barcode", "sample": "Sample Name", "protein": "B#", "clone": "C Code" } } </pre>	
--	--	--	--	--

Table 1: Auto Processor Web Configuration