# Archer Data Feed Monitor

The Archer Data Feed Monitor is a Tool & Utility that enables an Archer Administrator to call a data feed only when there is work to do (i.e., records that must be processed), reducing the resource load on the Archer Platform instance. Archer Data Feed Monitor is available to both Archer on-premises and Archer SaaS customers.

On this page

* [Release notes](#Releasenotes)
* [Overview of Archer Data Feed Monitor](#OverviewofArcherDataFeedMonitor)
  + [About Archer Data Feed Monitor](#AboutArcherDataFeedMonitor)
  + [Benefits](#Benefits)
  + [Prerequisites](#Prerequisites)
* [Archer Data Feed Monitor components](#ArcherDataFeedMonitorcomponents)
  + [Architecture diagram](#Architecturediagram)
* [Installing and Configuring Archer Data Feed Monitor](#Xf60a2c5eaaa07d9940f90aee4870c0077c0ad10)
  + [Installation overview](#Installationoverview)
  + [Installation and configuration](#Installationandconfiguration)
    - [Logging](#Logging)
    - [.NET Configuration](#NETConfiguration)
    - [General required monitor attributes](#Generalrequiredmonitorattributes)
    - [Monitor-specific required attributes](#Monitorspecificrequiredattributes)
    - [Configuration of data feed monitor tool and utility](#X4f62ec73678f9c33f25e6a79690bc511ad5faef)
    - [Test the connection](#Testtheconnection)
    - [Security considerations](#Securityconsiderations)
    - [Risk considerations](#Riskconsiderations)
* [Troubleshooting](#Troubleshooting)
  + [Configuration File (DataFeedMonitor.exe.config)](#X7cd685a6263137683a385990033b3cb000350c2)
* [Certification environment](#Certificationenvironment)

## Release notes

| Document Version | Published Date | Notes |
| --- | --- | --- |
| 1.0 | August 2018 | Initial release |
| 1.1 | May 2019 | The new version of Archer Data Feed Monitor uses the NLog framework to enable logging. There is no change in the business logic of how the Data Feed Monitor works, only a change in the logging functionality. The previous version of this tool did not use a logging framework and was writing data to one log file. Over time, this file will become quite large and difficult to work with. NLog provides a configurable option for logging such as setting the maximum log file size and options to configure a Rolling File Appender to create new log files at certain time intervals. |
| 1.2 | October 2019 | Added details for the Archer Record Statistical Monitor |
| 1.3 | September 2020 | The latest version of Archer Data Feed Monitor has the following updates:   1. Exceptions that occur while checking the criteria or executing a feed for a given monitor result in the rest of the monitors being skipped for that execution. This bug has been fixed. 2. App Settings in Config file are encrypted when the utility is run for the first time. Encrypt and decrypt batch files are no longer required. |
| 1.3.1 | March 2021 | Updated documentation with the added support for the use of Stored Procedures in the SQL Monitor. |
| 1.3.2 | April 2022 | Updated documentation with notes regarding implementation in Archer SaaS environments |

## Overview of Archer Data Feed Monitor

### About Archer Data Feed Monitor

In many Archer implementations, business users are looking for the ability to react to specific events in near real-time. For example, a file containing vulnerability scan results is placed on a network share, and the business wants to ingest the file as soon as it appears. The creation of the file can be unscheduled, and therefore unexpected. Data feeds that perform the ingestion are configured to run every five minutes (or less) in an effort to approximate real-time processing.

This use of minutely data feeds produces several undesirable results. Most notably, the rapid growth of affected system tables: tblAsyncInstrumentation, tblDataFeedHistory, and tblDataFeedMessage. In some cases, this can result in those tables landing in the top 10 by row count and size on disk.

Spinning up and shutting down a data feed requires system resources. The Job Engine must poll for runnable jobs, pop the data feed job from the queue, push the next occurrence to the queue, spin up a thread to process the job, unpack the named argument XML, constitute the data feed, check the source, determine that there is no work to do, log messages to tblDataFeedHistory and tblDataFeedMessage, and write the completed job information to tblAsyncInstrumentation. In one client instance, a data feed that was scheduled to run every five minutes was executed more than 17,000 times during a 60-day period. During that time there was not a single source row to process. This occupied more than three days of processing time on the services server and generated thousands of rows of data in the instance database to track the processing of the feeds. A better solution is needed.

The Data Feed Monitor is intended to be run through the Windows Task Scheduler at a high frequency (every five minutes or less). When it runs, it checks a pre-configured list of monitors to see if any of them found work to do. Any monitor that finds pending work will call its associated data feed. This results in near real-time data feed execution, but without the expense of spinning up a data feed when there is no work to do. It also avoids the large database footprint associated with frequent data feed execution.

Revisiting our customer example, the Data Feed Monitor was deployed to watch for pending work. Over a subsequent 60-day period, the data feed was executed twice, resulting in a major savings of server resources and database storage.

### Benefits

Archer Data Feed Monitor enables you to:

* Implement a monitor that can call the data feed only when there is work to do.
* Reduce the instance processing requirements.
* Reduce the amount of data stored in the database data feed history/log tables.
* Improve instance performance.

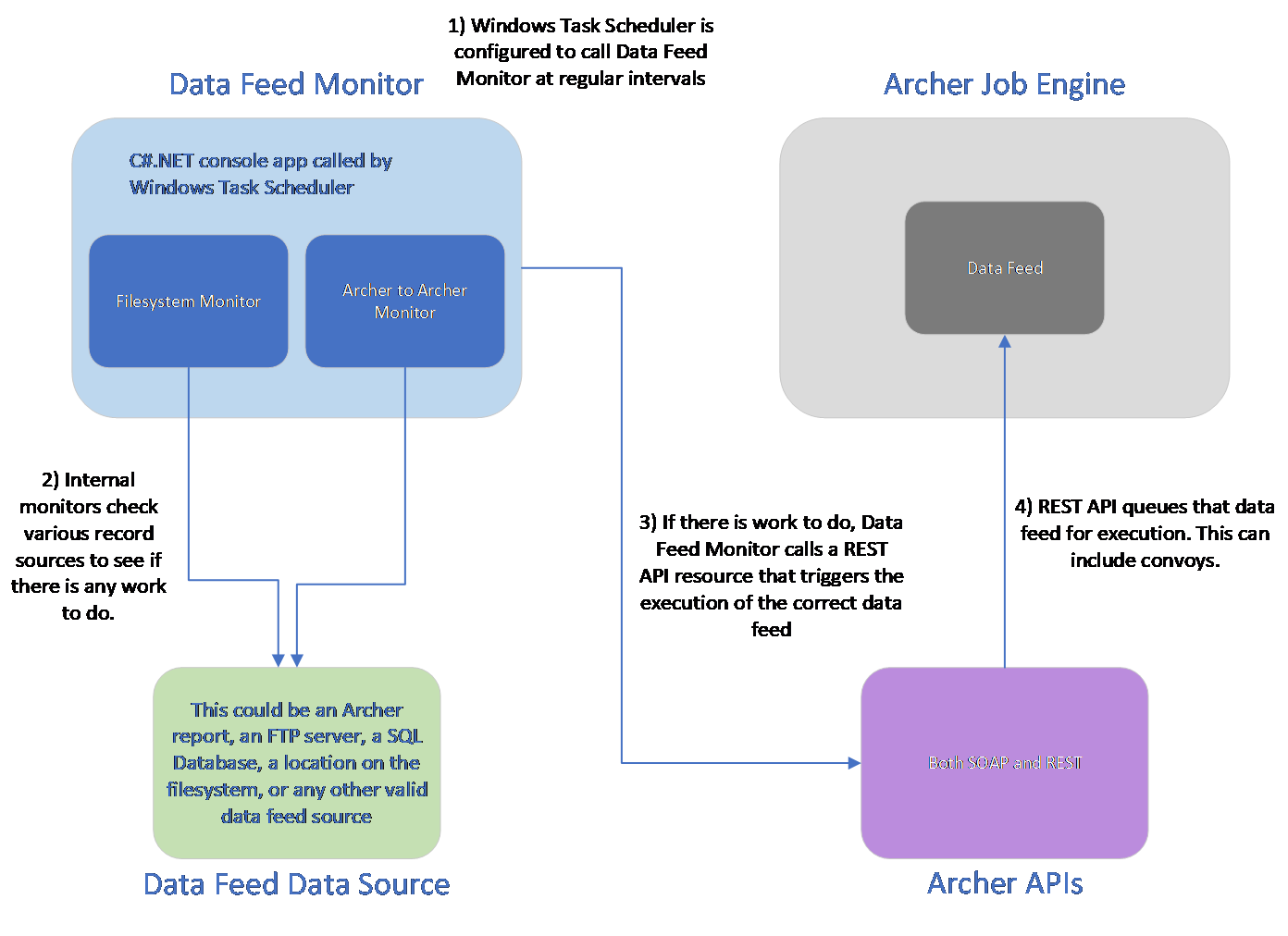
### Prerequisites

| Components | Recommended Software |
| --- | --- |
| Operating System | Windows Server 2012 R2 |
| Web Server | Microsoft Internet Information Services (Included in Windows Server 2012 R2 or 2016) |
| Framework | Microsoft .Net framework 4.6 and above |
| Deployment Requirements | on-premises implementations can deploy the web application on the server that is hosting the Archer platform.  However, Archer SaaS implementations must deploy this tool & utility on a separate server from the Archer SaaS server |
| Network Requirements | The Data Feed Monitor should have network access to the Archer |

## Archer Data Feed Monitor components

### Architecture diagram

The following diagram provides a high-level overview of the data flow process for the Archer Data Feed Monitor Tool & Utility.



## Installing and Configuring Archer Data Feed Monitor

### Installation overview

This section provides instructions for configuring the Archer Data Feed Monitor Tool & Utility within the Archer Platform.

It is assumed that the reader has both working knowledge of all products involved, and the ability to perform the tasks outlined in this section. Administrators must have access to the documentation for all products in order to install the required components.

**Important:** The integration described in this guide is being provided as a reference implementation for evaluation and testing purposes. It may or may not meet the needs and use cases for your organization.  If additional customizations or enhancements are needed, it is recommended that customers contact Archer Help for assistance.

### Installation and configuration

The Data Feed Monitor is an ASP.NET console application. The solution includes two XML configuration files. These allow for the specification of authentication credentials, along with other necessary details.

#### Logging

The solution makes use of a well-known logging framework called NLog. NLog allows for the creation of one or more log <target> entries, and one or more associated <logger>entries. The targets define potential “listeners” for log information, and the loggers define the rules associated with each of them, including log level. Although a sample config file is provided with the solution, a full discussion of the configuration file and its use is outside the scope of this document. For more details, please see the [NLog documentation](https://github.com/NLog/NLog/wiki/NLog-config-Example).

#### .NET Configuration

An ASP.NET console application uses an XML configuration file in the form of the DataFeedMonitor.exe.config. This file contains the following configuration details:

<appSettings>

<addkey="archerBaseUrl"value="https://[YourArcherHostURL]"/>

<addkey="cacheFilePath"value="[cacheFilePath]"/>

<addkey="cacheFileEncryptionPassphrase"value="[EncryptionPassphrase]"/>

<addkey="useSslValidation"value="[true/false]"/>

<addkey="securityProtocolVersion" value="[Ssl3/Tls/Tls11/Tls12]" />

<addkey="archerInstance"value="[InstanceName]"/>

<addkey="archerUser"value="[UserName]"/>

<addkey="archerPwd"value="[Password]"/>

<addkey="httpTimeoutMilliseconds" value="[httpTimeoutMilliseconds]" />

</appSettings>

The XML configuration code below offers a configuration example.

<appSettings>

<addkey="archerBaseUrl"value="https://localhost/archer"/>

<addkey="cacheFilePath"value="c:\temp\"/>

<addkey="cacheFileEncryptionPassphrase"value="d0n7TryTh1s@h0m3!"/>

<addkey="securityProtocolVersion" value="Tls" />

<addkey="useSslValidation"value="false"/>

<addkey="archerInstance"value="Archer"/>

<addkey="archerUser"value="adminuser"/>

<addkey="archerPwd"value="Password@123"/>

<addkey="httpTimeoutMilliseconds" value="30000" />

</appSettings>

The following table describes each setting in DataFeedMonitor.exe.config file.

| Setting | Description |
| --- | --- |
| archerBaseUrl | The root URL for the Archer instance from which information is pulled. |
| cacheFilePath | Path to the location where the cache file will be written. This flat file is used to cache Archer ID values. When the solution is started, values are looked up from the configured Archer instance and stored in the cache file. This enables correctly referenced fields, modules, and other resources to use IDs appropriate to the Archer instance. |
| cacheFileEncryptionPassphrase | If provided, the cache file is encrypted using this setting as the passphrase, providing security for data at rest. |
| securityProtocolVersion | This value dictates which security protocol should be used. Options include: SSL3, TLS 1.0, TLS 1.1, and TLS 1.2. |
| useSslValidation | When set to true, custom validation is used for the SSL certificate. This is necessary when using a self-signed certificate to avoid validation dialogs. |
| archerInstance | The name of the Archer instance from which information is pulled. |
| archerUser | The name of the account used to pull information from Archer. |
| archerPwd | The password for the account used to pull information from Archer. |
| httpTimeoutMilliseconds | The httpTimeoutMilliseconds provides a configuration setting that enables the user to override the default HTTP timeout of 30,000 milliseconds (30 seconds). This value is in milliseconds and should be set to the number of seconds you want HTTP time out to take \* 1000. |

The configuration file also allows you to define monitors, along with other necessary details.

The following is a description of each of these settings.

<sectionname="monitorSection"…– this entry associates a custom configuration section with the assembly that will be used to process that section. This entry should not be modified.

<monitorSection><monitors>…– this section contains configuration information for each individually configured monitor.

<monitor…– each monitor node defines the configuration for a single data feed monitor. The configuration file can contain as many monitors are needed.

#### General required monitor attributes

monitorType – must be one of five supported values corresponding to the type of transporter being used by the feed to be monitored.

| archerRecordMonitor | Archer Web Services Transporter (standard report) |
| --- | --- |
| archerRecordStatisticalMonitor | Archer Web Services Transporter (statistical report) |
| fileSystemMonitor | File Transporter |
| sqlDatabaseMonitor | Database Query Transporter |
| ftpMonitor | FTP Transporter |

name – the name of the monitor is used only to allow the administrator to determine what has been configured. It is recommended that monitors be named for the feeds they will run.

associatedFeedGuid –the GUID associated with the feed that should be run when the monitor finds pending work.

isFeedConvoy –a Boolean (true/false) value indicating whether or not this feed has others that reference it. When this value is set to “true”, the next feed will be automatically executed upon completion of the reference feed.

#### Monitor-specific required attributes

There are several monitor attributes that may be required by the type of monitor being configured. If these attributes are not specified in the configuration for the monitor in question, an exception will be logged, and the application will exit.

| Archer Record Monitor/Archer Record Statistical Monitor | archerBaseUrl, cacheFilePath, cacheFileEncryptionPassphrase, useSslValidation, archerInstance, archerUser, and archerPassword are all discussed in detail in the AppSettings section. They are included as attributes in the Archer Record Monitor or Archer Record Monitor Statistical config to enable the use case where an Archer to Archer data feed is pulling data from one Archer instance and pushing it to another. The attributes configured for the monitor will be used to pull the source information. The attributes configured in AppSettings will be used to execute the data feed. If you are not moving data between instances, the config for the monitor and the config in AppSettings will have the same values. |
| --- | --- |
| reportGuid– the GUID associated with the Archer report that supplies the source data for the configured feed. This report will be executed using the SOAP API. If the report is a statistics report, the Archer Record Statistical Monitor must be used. |  |
| Filesystem Monitor | directoryToWatch– the path to the directory that will be monitored. |
| filePatternToMatch– a file specifier that identifies which files to watch for in the configured watch directory. For example, “\*.csv”. |
| SQL Database Monitor | connectionString– a valid connection string for a Microsoft SQL Server database. This is using the .NET SqlClient class under the covers and does not currently support Oracle connections. |
| command– an XML structure specifying the details of the command to be executed. |
| command/isProc – a boolean [true/false] value indicating whether the configured command text is a stored procedure to be executed. |
| command/isScalar – a boolean [true/false] value indicating whether the configured stored procedure returns a scalar value. Set this value to false for stored procedures that return row results. |
| command/commandText – the actual text of the command to be executed. For SQL queries, this will be the text of the query. For stored procedures, this will be the name of the procedure. |
| command/params – a collection of Param objects specify any parameters that need to be passed to the configured stored procedure |
| command/Param – an individual parameter to be passed to the configured stored procedure. |
| command/Param/name – the name of the parameter |
| command/param/type – the SQL DB Type associated with the parameter. Valid values are: BigInt, Binary, Bit, Char, DateTime, Decimal, Float, Image, Int, Money, NChar, NText, NVarChar, Real, UniqueIdentifier, SmallDateTime, SmallInt, SmallMoney, Text, Timestamp, TinyInt, VarBinary, VarChar, Variant, Xml, Udt, Structured, Date, Time, DateTime2, DateTimeOffset |
| command/param/value – the value of the parameter |
| FTP Monitor | ftpServerUrl– the URL to the FTP server to be monitored. The current version of the FTP monitor does not support secure FTP or monitoring in folders other than the root. This is provided as a reference implementation. |
| isAnonymous– a Boolean (true/false) value indicating whether the connection should be made using anonymous credentials. When false, username and password should be supplied. |
| User– a valid FTP user id |
| Password– the password associated with the configured FTP user id |
| filePatternToMatch– a regular expression identifying the files to watched for in the configured FTP server. Since FTP does not support file filtering in the same way that the filesystem does, a directory listing is requested and each row of the result is checked for a match against the regular expression. The first row that matches will cause the feed to be executed. For example, “^.\*\.(csv)$” would be equivalent to “\*.csv “. |

#### Configuration of data feed monitor tool and utility

1. Extract the zip file DataFeedMonitor.zipto a location/folder.
2. Edit the app Settings section of the XML configuration file “DataFeedMonitor.exe.config”. If a named config file is being used, edit that file instead.
   1. Navigate to the web application folder DataFeedMonitor, and locate the DataFeedMonitor.exe.config file.
   2. Open DataFeedMonitor.exe.config file and provide the following information:
   * <appSettings>
   * <addkey="archerBaseUrl"value="https://[YourArcherHostURL]"/>
   * <addkey="cacheFilePath"value="[cacheFilePath]"/>
   * <addkey="cacheFileEncryptionPassphrase"value="[EncryptionPassphrase]"/>
   * <addkey="useSslValidation"value="[true/false]"/>
   * <addkey="securityProtocolVersion" value="[Ssl3/Tls/Tls11/Tls12]" />
   * <addkey="archerInstance"value="[InstanceName]"/>
   * <addkey="archerUser"value="[UserName]"/>
   * <addkey="archerPwd"value="[Password]"/>
   * <addkey="httpTimeoutMilliseconds" value="[httpTimeoutMilliseconds]" />
   * </appSettings>

The following table describes each setting.

| Setting | Description and Example |
| --- | --- |
| [YourArcherHostURL] | Archer host URL  Example: company.com or 10.20.30.40:99 |
| [cacheFilePath] | Cache file path  Example:  c:\temp\ |
| [EncryptionPassphrase] | Encryption pass phrase  Example:  d0n7TryTh1s@h0m3! |
| [true/false] | True or false to validate SSL  Example: false |
| [Ssl3/Tls/Tls11/Tls12] | The version of security protocol to be used incommunicating with the Archer web server.  Example:Tls12 |
| [InstanceName] | Instance name of your Archer environment  Example: ArcherProd  **Note:** Instance name is case sensitive. |
| [UserName] | Username of the account  Example: adminuser |
| [Password] | Password of the user account  Example:  Password@123 |
| [httpTimeoutMilliseconds] | httpTimeoutMilliseconds  Example: 30000 |

1. Edit the monitorSection section of the XML configuration file.

* <monitorSection>
* <monitors>
* <monitor
* monitorType="ArcherRecordMonitor"
* name="[Name of the datafeed. Value should be Unique]"
* associatedFeedGuid="[Data Feed Guid]"
* isFeedConvoy="[true/false]"
* archerBaseUrl="http://[YourArcherHostURL]"
* cacheFilePath="[cacheFilePath]"
* cacheFileEncryptionPassphrase="[EncryptionPassphrase]"
* useSslValidation="[true/false]"
* archerInstance="[InstanceName]"
* archerUser="[UserName]"
* archerPassword="[Password]"
* reportGuid="[Archer Report Guid]"/>
* <monitor
* monitorType="ArcherRecordStatisticalMonitor"
* name="[Name of the datafeed. Value should be Unique]"
* associatedFeedGuid="[Data Feed Guid]"
* isFeedConvoy="[true/false]"
* archerBaseUrl="http://[YourArcherHostURL]"
* cacheFilePath="[cacheFilePath]"
* cacheFileEncryptionPassphrase="[EncryptionPassphrase]"
* useSslValidation="[true/false]"
* archerInstance="[InstanceName]"
* archerUser="[UserName]"
* archerPassword="[Password]"
* reportGuid="[Archer Statistical Report Guid]"/>
* <monitor
* monitorType="FileSystemMonitor"
* name="[Name of the datafeed. Value should be Unique]"
* associatedFeedGuid="[Data Feed Guid]"
* isFeedConvoy="[true/false]"
* filePatternToMatch="[File Type to watch]"
* directoryToWatch="[Directory Path to be monitored]" />
* <monitor
* monitorType="SqlDatabaseMonitor"
* name="[Name of the datafeed. Value should be Unique]"
* associatedFeedGuid="[Data Feed Guid]"
* isFeedConvoy="[true/false]"
* connectionString="[a connection string to the SQL DB Source]">
* <command
* isProc="[true/false]"
* isScalar="[true/false]"
* commandText="[The SQL Query or stored procedure name to be executed]">
* <params>
* <paramname="[the name of the SQL Parameter]"type="[The SQL DB Type of the SQL Parameter]"value="[the value of the SQL Parameter]"/>
* </params>
* </command>
* </monitor>
* <monitor
* monitorType="FtpMonitor"
* name="[Name of the datafeed. Value should be Unique]"
* associatedFeedGuid="[Data Feed Guid]"
* isFeedConvoy="[true/false]"
* filePatternToMatch="[Regex identifying the files to be watched]"
* ftpServerUrl="[FTP Url]"
* isAnonymous="[Data Feed Guid]"
* user="[FTP UserName]"
* password="[FTP Password]" />
* </monitors>
* </monitorSection>

**Option:** Users can also use a separate config file for the <monitorSection>. To use a separate file for monitor section:

* Edit and replace the <monitorSection> in the xml configuration file with
* <monitorSection configSource="[FolderName\monitor.config]" />.
* “FolderName” is the folder where the monitor config files are placed and “monitor.config” is the file containing the feed details.
* Ex: <monitorSection configSource="ConfigFiles\Test1.config">
* The monitor file should have the following format:
* <?xmlversion="1.0"?>
* <monitorSection>
* <monitors>
* [Described above]
* </monitors>
* </monitorSection>

1. Save and Close the DataFeedMonitor.exe.config file.

* **Note:** Changing the Archer URL in the Config file requires the deletion of the previously used cache file.

1. Open NLog.config. Locate file target and provide the following information.

* filename="[logFilePath]"
* Example: fileName="c:\temp\DataFeedMonitor\_Nlog.log"
* Name of the file to write to.Provide name of the file to write to with the full path.
* archiveFileName="[archivelogFilePath]"
* Example: archiveFileName="c:\temp\logArchive\DataFeedMonitor\_Nlog.{#}.log"
* Name of the file to be used for an archive. It contains a special placeholder {#} that will be replaced with a sequence of numbers depending on the archiving strategy. The number of hash characters used determines the number of numerical digits to be used for numbering files.

1. Open Windows Task Scheduler and create a task to run DataFeedMonitor.exe according to the desired schedule. If you are only scheduling a single task to run the application, you can use the default configuration file. If you need to schedule multiple tasks to run data feeds on different schedules, store a separate configuration file for each task and pass the name and path of that configuration file to the application on the command line in the task. Whether using the default or a separate named configuration file, create unique Archer user accounts for each task instance so that no other process can authenticate with the same credentials and invalidate the session being used by your task.

To create a Windows Scheduler Task for the application:

1. Open Windows Task Scheduler
2. On the right side of the window under Action, click on Create Task.
3. In the dialog, enter the following:

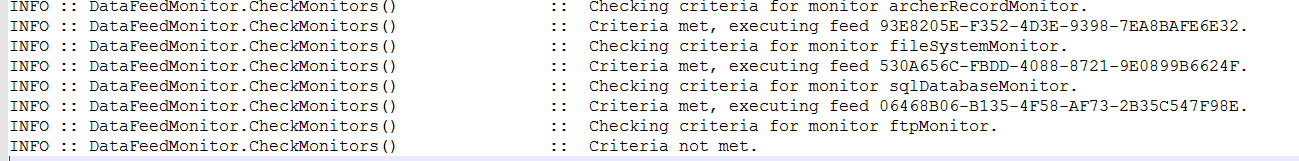
| Section | Description |
| --- | --- |
| General | Provide the DataFeed Name |
| Triggers | Create a new trigger by providing the Frequency to run the tool |
| Actions | Select the DataFeedMonitor.exe from extracted application folder Archer Data Feed Monitor. Also select a config file in “Add arguments” if you need to set up multiple schedules and are using named configuration files. |

1. Once all the properties for windows task schedule is complete, click on OK to start the schedule.

**Note:** If Data Feed Monitor is configured to run in Windows Task Scheduler, all associated data feeds referenced in the config file should have start dates several years in the future in Archer to prevent unexpected executions.

#### Test the connection

If the process is successful, it displays the message in the DataFeedMonitor.log as seen in the following example:



#### Security considerations

* Use HTTPS/TLS for secured connection between the DFM Application and Archer.
* Run the DFM application with the least privileged account (Logon Service permission).
* Run the DFM Utility immediately after configuring to encrypt the monitor and app settings in the config file.

#### Risk considerations

The Data Feed Monitor must be executed immediately after configuring. Failure to do so will result in a risk exposure, leaving parameters entered in monitor Section and app Settings (includes Archer credentials) in the config file in plain text.

## Troubleshooting

### Configuration File (DataFeedMonitor.exe.config)

* Ensure the values in the configuration file match your Archer environment.
* Do not include default.aspx in the Archer URL.
* Note that the Archer instance name is case sensitive.
* If you are changing the Archer URL in the config file, ensure that you have deleted the previously used cache file.
* Ensure that dedicated, active Archer user account credentials are used in the configuration file.

Others:

* When configuring a file transport monitor, if the directory being monitored is not in the same location where the Data Feed Monitor is running, ensure that you have specified the full path in the “directoryToWatch” parameter. This will be the case when the directory to watch is on a network share, or when the it is not in the same location where the Data Feed Monitor is running. If the folder has security, you can schedule the task in Windows Task Scheduler to run under a privileged account. Or you can just have it run the task under a service account and then grant that account access to the share.
* If you are running an Archer Record Monitor and the utility is reporting that the criteria are not met, even when you think they should be, ensure that the report that the monitor is watching is a standard report, not a statistics report. If the report is a statistics report, you must configure an Archer Record Statistical Monitor instead. This is because the two reports return different XML schemas, so a specialized monitor is required to correctly parse the XML and count results.

## Certification environment

Date tested: September 2020

| Product Name | Version Information | Operating System |
| --- | --- | --- |
| Archer | 6.7 | Windows Server 2012 R2 |