

Apache MiNiFi 3

MiNiFi Quick Start

Date of Publish: 2018-11-15



<https://docs.hortonworks.com/>

Contents

MiNiFi Java Agent Quick Start.....	3
Overview.....	3
Before You Begin.....	3
Installing and Starting MiNiFi.....	3
Installing MiNiFi on Linux.....	3
Installing MiNiFi as a Service on Linux.....	3
Starting MiNiFi on Linux.....	4
Installing MiNiFi on Windows.....	4
Configuring the MiNiFi MSI.....	5
Starting MiNiFi on Windows.....	7
Working with Dataflows.....	7
Setting up Your Dataflow.....	7
Using Processors Not Packaged with MiNiFi.....	8
Securing your Dataflow.....	10
Managing MiNiFi.....	10
Monitoring Status.....	10
Loading a New Dataflow.....	11
Stopping MiNiFi.....	11
Supported Processors.....	12
Processors Supported by Java.....	12
Supported Processors Requiring NAR.....	13

MiNiFi Java Agent Quick Start

This guide is intended to help you install and start using MiNiFi Java Agent quickly. For additional details, see the Administration Guide.

Overview

Apache NiFi, MiNiFi is an Apache NiFi project, designed to collect data at its source. MiNiFi was developed with the following objectives in mind:

- Small and lightweight footprint
- Central agent management
- Data provenance generation
- NiFi integration for follow-on dataflow management and chain of custody information

Before You Begin

MiNiFi is supported on the following operating systems:

- Red Hat Enterprise Linux / CentOS 6 (64-bit)
- Red Hat Enterprise Linux / CentOS 7 (64-bit)
- Ubuntu Trusty (14.04) (64-bit)
- Debian 7
- SUSE Linux Enterprise Server (SLES) 11 SP3 (64-bit)
- Windows

You can find download links for the following MiNiFi software in the *HDF Release Notes*.

- MiNiFi Java Agent
- MiNiFi C++
- MiNiFi Toolkit

Installing and Starting MiNiFi

You have several options for installing and starting MiNiFi.

Installing MiNiFi on Linux

To install MiNiFi on RHEL/CentOS, Ubuntu, Debian, SLES, complete the following steps:

Procedure

1. Download MiNiFi.
2. Extract the file to the location from which you want to run the application.

Installing MiNiFi as a Service on Linux

You can also install MiNiFi as a service:

Procedure

1. Navigate to the MiNiFi installation directory.

2. Enter:

```
bin/minifi.sh install
```

What to do next

You can also specify a custom name for your MiNiFi installation, by specifying that name during your install command. For example, to install MiNiFi as a service and named dataflow, enter:

```
bin/minifi.sh install dataflow
```

Starting MiNiFi on Linux

Once you have downloaded and installed MiNiFi, you need to start MiNiFi. You can start MiNiFi in the foreground, background, or as a service on Linux.

Procedure

1. Launching MiNiFi in the foreground: From a terminal window, navigate to the MiNiFi installation directory.

2. Enter:

```
bin/minifi.sh run
```

3. Launching MiNiFi in the background: From a terminal window, navigate to the MiNiFi installation directory.

4. Enter:

```
bin/minifi.sh start
```

5. Launch MiNiFi as a service: From a terminal window, enter:

```
sudo service minifi start
```

Installing MiNiFi on Windows

You can install MiNiFi using a Windows MSI:

Before you begin

Before you begin your MiNiFi installation, be sure you meet the following requirements:

- Install JDK 8.0 64 bit.
- Install Java to C:/java instead of C:/Program Files.
 - Recent Windows versions mark everything in C:\Program Files as read only.
- Set the JAVA_HOME environment variable using the 8.3 style name conventions.
 - For example: C:\Program\jdk1.8.0.
- Ensure JAVA_HOME is pointing to a 64-bit JRE/JDK.
- Ensure the Domain user has administrator privilege.
- Ensure your system meets the minimum memory requirement for Windows which is 4GB.

Procedure

1. Extract the MiNiFi MSI files in the repo location at <http://public-repo-1.hortonworks.com/HDF/windows/3.x/updates/3.2.0.0/nifi-3.2.0.0-520.msi> to the location from which you want to run the application.
2. Execute the MSI.

Configuring the MiNiFi MSI

The MSI adds the Windows service for MiNiFi. The service is configured to be executed by either a local user in the computer, or a domain user in ActiveDirectory.

Using a Local User for MiNiFi Windows Service

There is no prerequisite to use a Local user for the Windows service. The installer automatically sets up the user.

About this task

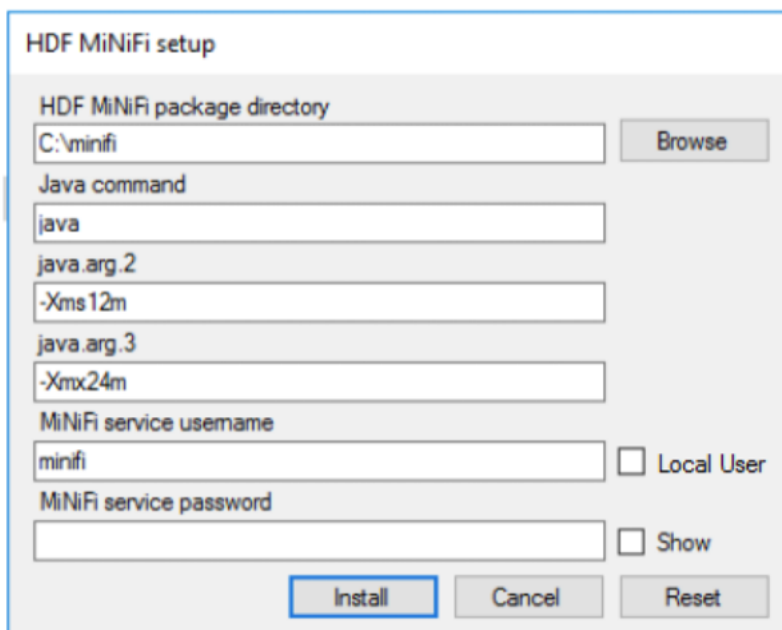
The installer also grants the following privileges to the specified user:

- SeCreateSymbolicLinkPrivilege
- SeServiceLogonRight

Procedure

If the computer is a part of a domain, then **Local User** checkbox appears in the **HDF NiFi setup** window. Check the **Local User** checkbox to specify that Local user is used to execute the installed service.

HDF_MiNiFi_setup.png



The screenshot shows the 'HDF MiNiFi setup' dialog box. It contains the following fields and controls:

- HDF MiNiFi package directory:** Text box containing 'C:\minifi' and a 'Browse' button.
- Java command:** Text box containing 'java'.
- java.arg.2:** Text box containing '-Xms12m'.
- java.arg.3:** Text box containing '-Xmx24m'.
- MiNiFi service username:** Text box containing 'minifi' and a checkbox labeled 'Local User'.
- MiNiFi service password:** Text box and a checkbox labeled 'Show'.
- Buttons:** 'Install', 'Cancel', and 'Reset' buttons at the bottom.

If a user specified at **MiNiFi service username** does not exist, the installer creates one with the specified **MiNiFi service password**. If the user already exists, the installer updates its password with the specified password.

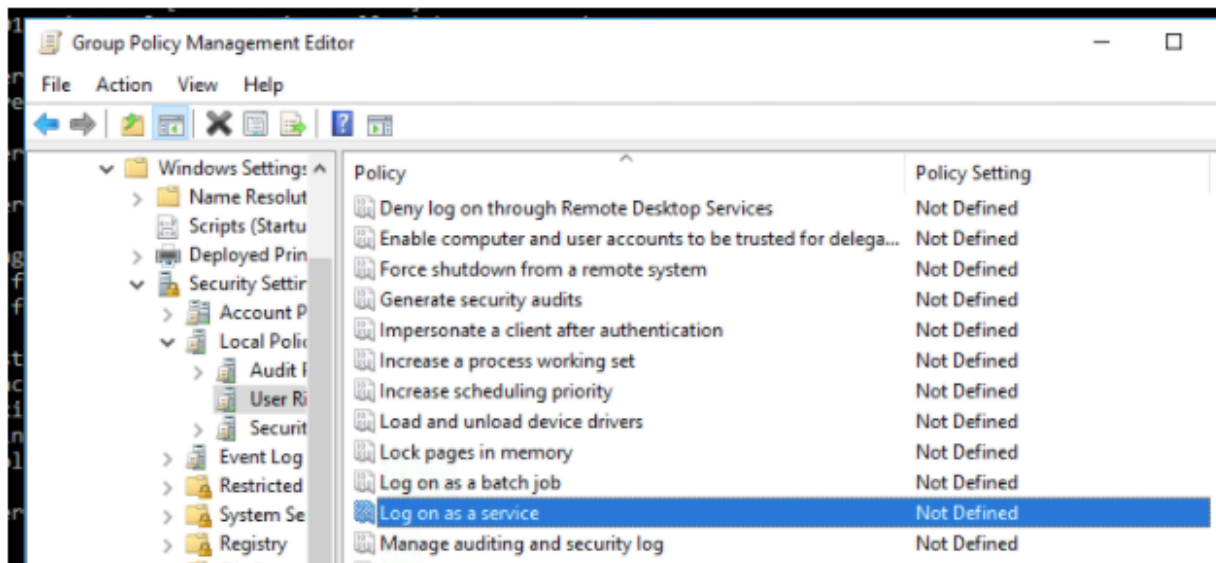
Using a Domain User for MiNiFi Windows Service

Before you begin

- The computer must be part of the domain.
- The specified user must exist in the domain, and a correct password must be provided.
- ActiveDirectory PowerShell module must be available.

Procedure

1. In the Group Policy Management Editor, set permission to 'Log on as a service.'



2. Navigate to a machine on which MiNiFi is installed and enter the following command:

```
gpupdate
```

The gpupdate command is a machine-wide command and can be executed from any directory on the MiNiFi machine.

3. Install the ActiveDirectory PowerShell module by entering the following in the PowerShell console:

```
Add-WindowsFeature RSAT-AD-PowerShell
```

4. In the HDF NiFi setup window, uncheck 'Local User' checkbox then click **Install**.

HDF MiNiFi setup

HDF MiNiFi package directory

Java command

java.arg.2

java.arg.3

MiNiFi service username
 Local User

MiNiFi service password
 Show

What to do next

After installation, you can update Java options at `nifi-install-dir\conf\bootstrap.conf` file. Repository locations are at `nifi-install-dir\conf\nifi.properties` file.

Starting MiNiFi on Windows

Once you have downloaded and installed MiNiFi, you can start MiNiFi in the foreground or as a service on Windows.

About this task

Launching MiNiFi in the foreground:

Procedure

1. From a command prompt window, navigate to the MiNiFi installation directory.
2. Enter the following command to launch MiNiFi in the foreground:

```
bin\run-minifi.bat
```

What to do next

Launching MiNiFi as a service:

You can start or stop the installed MiNiFi service from the Windows Service Manager.

Working with Dataflows

When you are working with a MiNiFi dataflow, you should design it, add any additional configuration your environment or use case requires, and then deploy your dataflow. MiNiFi is not designed to accommodate substantial mid-dataflow configuration.

Setting up Your Dataflow

You can use the following information to set up your MiNiFi dataflow.

About this task

Before you begin, you should be aware that the following NiFi components are not supported in MiNiFi dataflows:

- Funnels
- Multiple source relationships for a single connection
- Process groups

Additionally, each processor requires a distinct name.

You can use the MiNiFi Toolkit, located in your MiNiFi installation directory, and any NiFi instance to set up the dataflow you want MiNiFi to run:

Procedure

1. Launch NiFi.
2. Create a dataflow.
3. Convert your dataflow into a template.



Note:

You can use one template at a time, per MiNiFi instance.

4. Download your template as an .xml file.
5. From the MiNiFi Toolkit, run the following command to turn your .xml file into a .yaml file:

```
config.sh transform input_file output_file
```

6. Move your new .yaml file to minifi/conf.

7. Rename your .yml file config.yml.

Results

Once you have your config.yml file in the minifi/conf directory, launch that instance of MiNiFi and your dataflow begins automatically.

Using Processors Not Packaged with MiNiFi

You should be familiar with the processors that you can use out of the box, and how to

About this task

MiNiFi is able to use the following processors out of the box:

- UpdateAttribute
- AttributesToJSON
- Base64EncodeContent
- CompressContent
- ControlRate
- ConvertCharacterSet
- ConvertJSONToSQL
- DetectDuplicate
- DistributeLoad
- DuplicateFlowFile
- EncryptContent
- EvaluateJsonPath
- EvaluateXPath
- EvaluateXQuery
- ExecuteProcess
- ExecuteSQL
- ExecuteStreamCommand
- ExtractText
- FetchDistributedMapCache
- FetchFile
- FetchSFTP
- GenerateFlowFile
- GetFTP
- GetFile
- GetHTTP
- GetJMSQueue
- GetJMSTopic
- GetSFTP
- HandleHttpRequest
- HandleHttpResponse
- HashAttribute
- HashContent
- IdentifyMimeType
- InvokeHTTP
- ListFile
- ListSFTP
- ListenHTTP
- ListenRELP

- ListenSyslog
- ListenTCP
- ListenUDP
- LogAttribute
- MergeContent
- ModifyBytes
- MonitorActivity
- ParseSyslog
- PostHTTP
- PutDistributedMapCache
- PutEmail
- PutFTP
- PutFile
- PutJMS
- PutSFTP
- PutSQL
- PutSyslog
- QueryDatabaseTable
- ReplaceText
- ReplaceTextWithMapping
- RouteOnAttribute
- RouteOnContent
- RouteText
- ScanAttribute
- ScanContent
- SegmentContent
- SplitContent
- SplitJson
- SplitText
- SplitXml
- TailFile
- TransformXml
- UnpackContent
- ValidateXml

Procedure

1. Set up your dataflow as described above.
2. Copy the desired NAR file into the NiFi lib directory.
3. Restart your NiFi instance.

What to do next

**Note:**

Currently only the StandardSSLContextService is supported as a controller service. It is created automatically if the "Security Properties" section is set and can be referenced in the processor configuration using the ID "SSL-Context-Service".

Securing your Dataflow

You can secure your MiNiFi dataflow using keystore or trust store SSL protocols, however, this information is not automatically generated. You will need to generate your security configuration information yourself.

About this task

To run a MiNiFi dataflow securely, modify the Security Properties section of your config.yml file.

Procedure

1. Create your dataflow template as discussed above.
2. Move it to minifi.conf and rename config.yml.
3. Manually modify the Security Properties section of config.yml.

Example

```
Security Properties:
keystore:
keystore type:
keystore password:
key password:
truststore:
truststore type:
truststore password:
ssl protocol: TLS
Sensitive Props:
key:
algorithm: PBEWITHMD5AND256BITAES-CBC-OPENSSL
provider: BC
```

Managing MiNiFi

You can also perform some management tasks using MiNiFi.

Monitoring Status

You can use the minifi.sh flowStatus option to monitor a range of aspects of your MiNiFi operational and dataflow status. You can use the flowStatus option to get information dataflow component health and functionality, a MiNiFi instance, or system diagnostics.

FlowStatus accepts the following flags and options:

- processors
 - health
 - bulletins
 - status
- connections
 - health
 - stats
- remoteProcessGroups
 - health
 - bulletins
 - status

- authorizationIssues
- inputPorts
- controllerServices
 - health
 - bulletins
- provenancereporting
 - health
 - bulletins
- instance
 - health
 - bulletins
 - status
- systemdiagnostics
 - heap
 - processorstats
 - contentrepositoryusage
 - flowfilerepositoryusage
 - garbagecollection

For example, this query gets the health, stats, and bulletins for the TailFile processors

```
minifi.sh flowStatus processor:TailFile:health,stats,bulletins
```

**Note:**

Currently the script only accepts one high level option at a time.

Any connections, remote process groups or processors names that contain ":", ";" or "," will cause parsing errors when querying.

Loading a New Dataflow

You can load a new dataflow for a MiNiFi instance to run:

Procedure

1. Create a new config.yml file with the new dataflow.
2. Replace the existing config.yml in minifi/conf with the new file.
3. Restart MiNiFi.

Stopping MiNiFi

You can stop MiNiFi at any time.

Procedure

1. From a terminal window, navigate to the MiNiFi installation directory.
2. Enter:

```
bin/minifi.sh stop
```

3. To stop MiNiFi running as a service:
4. From a terminal window, enter:

```
sudo service minifi stop
```

What to do next

Stopping NiFi as a service:

Supported Processors

When you first obtain the Java version of NiFi, several processors are supported without you taking any further action. Other processors are a part of the default distribution but require you to add a NAR for a controller service not packaged by default.

Processors Supported by Java

When you first obtain the Java version of NiFi, several processors are supported without you taking any further action. Other processors are a part of the default distribution but require you to add a NAR for a controller service not packaged by default.

- UpdateAttribute
- AttributesToJSON
- Base64EncodeContent
- CompressContent
- ControlRate
- ConvertCharacterSet
- ConvertJSONToSQL
- DetectDuplicate
- DistributeLoad
- DuplicateFlowFile
- EncryptContent
- EvaluateJsonPath
- EvaluateRegularExpression
- EvaluateXPath
- EvaluateXQuery
- ExecuteProcess
- ExecuteSQL
- ExecuteStreamCommand
- ExtractText
- FetchDistributedMapCache
- FetchFile
- FetchSFTP
- GenerateFlowFile
- GetFTP
- GetFile
- GetHTTP
- GetJMSQueue
- GetJMSTopic
- GetSFTP
- HandleHttpRequest
- HandleHttpResponse
- HashAttribute
- HashContent
- IdentifyMimeType

- InvokeHTTP
- ListFile
- ListSFTP
- ListenHTTP
- ListenRELP
- ListenSyslog
- ListenTCP
- ListenUDP
- LogAttribute
- MergeContent
- ModifyBytes
- MonitorActivity
- ParseSyslog
- PostHTTP
- PutDistributedMapCache
- PutEmail
- PutFTP
- PutFile
- PutJMS
- PutSFTP
- PutSQL
- PutSyslog
- QueryDatabaseTable
- ReplaceText
- ReplaceTextWithMapping
- RouteOnAttribute
- RouteOnContent
- RouteText
- ScanAttribute
- ScanContent
- SegmentContent
- SplitContent
- SplitJson
- SplitText
- SplitXml
- TailFile
- TransformXml
- UnpackContent
- ValidateXml

Supported Processors Requiring NAR

The following processors are also a part of the default distribution but require you to add a NAR for a controller service not packaged by default. The processors are grouped by the NAR that is required.

- nifi-dbcp-service-nar
 - ConvertJSONToSQL
 - PutSQL
 - GenerateTableFetch

- ListDatabaseTable
- QueryDatabaseTable
- ExecuteSQL
- nifi-distributed-cache-services-nar
 - DetectDuplicate
 - FetchDistributedMapCache
 - PutDistributedMapCache
- nifi-http-context-map-nar
 - HandleHttpRequest
 - HandleHttpResponse