

Release Notes - MiNiFi (C++)

[\[Version cpp-0.99.0 \]](#) [\[Version cpp-0.15.0 \]](#) [\[Version cpp-0.14.0 \]](#) [\[Version cpp-0.13.0 \]](#) [\[Version cpp-0.12.0 \]](#) [\[Version cpp-0.11.0 \]](#) [\[Version cpp-0.10.0 \]](#) [\[Version cpp-0.7.0 \]](#) [\[Version cpp-0.6.0 \]](#) [\[Version cpp-0.5.0 \]](#) [\[Version cpp-0.4.0 \]](#) [\[Version cpp-0.3.0 \]](#) [\[Version cpp-0.2.0 \]](#) [\[Version cpp-0.1.0 \]](#) [\[Version cpp-0.0.1 \]](#)

Version cpp-0.99.0

Release Date: **TBA**

New notable features:

- Added support for using NiFi 2.0 Python processors in MiNiFi C++
 - This also includes several improvements to the previous MiNiFi style python processors, like additional property options, custom relationships and virtualenv support
- Added new python based multiplatform bootstrap script
- Added encryption support for sensitive properties in flow configuration
- Releasing Windows installer now can be done (and will be done) under the Apache license
- Added support for service installation on MacOS
- Added C2 debug command to MiNiFi Controller
- Added support for setting MiNiFi properties from command line
- Added system load average field to C2 and Prometheus metrics
- Added support for manually configuring RocksDB options
- Added custom delimiter property for ListenTCP processor
- Added bandwidth limit properties to InvokeHTTP processor
- Added JSON flow configuration examples

New processors:

- Added PutSmb, FetchSmb and ListSmb processor for SMB networking protocol support
- Added PushGrafanaLokiGrpc and PushGrafanaLokiREST processors for pushing logs to Grafana Loki
- Added JoltTransform to use Jolt JSON transformations
- Added SplitText processor
- Added AttributeRollingWindow processor

Changes and improvements:

- Dropped support for disabling peer verification in InvokeHTTP
- Corrupt flow files are now filtered to avoid errors in the flow
- Using administrative yield duration instead of onschedule retry interval in scheduling adjusting to NiFi's functionality
- Fixed high disk IO usage issue with MergeContent
- Fixed the site-to-site transfer of large files
- Fixed memory leak caused by unused loggers
- Fixed yielding processors to still respect scheduling period

Upgraded dependencies:

- Upgraded OpenSSL to version 3.3.0
- Upgraded AWS SDK to version 1.11.219 with support for new AWS regions
- Upgraded libuv to version 0.0.7
- Upgraded docker base image to alpine:3.18
- Upgraded Sol2 to version 3.3.0

Version cpp-0.15.0

Release Date: **1 September 2023**

New features in this release:

- ConsumeWindowsEventLog can work from log files
- ConsumeWindowsEventLog resolve Security/UserID attribute
- TLS v1.3 support
- PutS3Object multipart upload support
- Use systemd service management on Linux
- Add ProcessContext::getStateManager to Lua/PythonR
- Reworked GetTCP
- SSL support for Prometheus reporter
- Multiarch docker support
- RFC3339 parsing with expression language
- Reworked Minifi controller
- GCC-13 support
- Fix for waking up prematurely after processor yields
- Fix system certificate store usage in SSLContextService on Linux
- Fix inconsistent naming in C2 machineArch
- Fix default CA path for S3 on CentOS
- Removed CronScheduler locale requirements

- Third party dependencies upgraded
 - Upgraded RocksDB to v8.1.1
 - Upgraded LibCurl to v8.1.0
 - Upgraded CivetWeb to v1.16
 - Upgraded OpenCV to v4.7.0
 - Upgraded GoogleCloud SDK to v2.10.1
 - Upgraded Azure SDK to v12.7.0
 - Replaced LibreSSL with OpenSSL 3.1.1

Version cpp-0.14.0

Release Date: **17 April 2023**

New features in this release:

- Several improvements on repository resource handling
 - Periodically run RocksDB repository compaction
 - Add compression options for RocksDB repositories
 - Add synchronous flow file reloading
 - Retry failed removals of unused resources
- Use Python stable ABI to support all libraries above Python 3.2+ for Python processors
- Add ProcessSession::remove to Python and Lua API
- Add support for JSON flow configuration format with option to generate JSON schema locally
- Add support for reverseDnsLookup in expression language
- Fix most issues for ARM64 support
- Improve performance of ListFile processor
- Add cache SID lookups in ConsumeWindowsEventLog processor
- Add default connection size limits of 2000 queue size and 100MB of queue data size
- Add support for MQTT 5 (mistakenly advertised for 0.13.0 release)
- Add failure relationship to SQL processors
- Add support for new AWS regions
- Add option to select processor metrics with regular expressions
- Add the UUID to the end of Processor and Controller Service log lines
- Make GetFile path attributes consistent with other processors
- Fix leaks and file lock issues on Windows
- Fix crashing in python processors

Version cpp-0.13.0

Release Date: **12 December 2022**

New features in this release:

- New processors:
 - [ListenTCP](#)
 - [PutTCP](#)
 - [PostElasticSearch](#)
 - [CollectKubernetesPodMetrics](#)
- Warn on SSL certificates about to expire
- Fix cron-based scheduling
- Improve metrics reporting and add support for Prometheus
- Improve the performance of several processors (ListenHTTP, AWS, Azure, GCS)
- Support swapping out flow files from memory to disk
- Support low-memory use cases with FileSystemRepository
- Improve the MQTT processors
- Improve communication with C2, eg. add alert capability
- Fix support of native packages in Python scripting
- Fix Python scripting on Windows
- Add SSL support to the ListenSyslog and ListenTCP processors
- Fix the 32-bit build on Windows
- Support POST/PUT of large files in InvokeHTTP
- Implement communication between process groups through ports

- Plus upgrade libraries, fix issues reported by clang-tidy, fix memory leaks etc

Build Notes:

- It is advised that you use bootstrap.sh when building on Linux or MacOS.
- This version does not compile with Clang + libc++ 14.0.0 due to a bug in this version of libc++: <https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=1008657>. Please use Clang (and libc++) version 14.0.6 or newer.

Version cpp-0.12.0

Release Date: **01 June 2022**

New features in this release:

- new processors:
 - [DeleteAzureBlobStorage](#)
 - [FetchAzureBlobStorage](#)
 - [FetchAzureDataLakeStorage](#)
 - [ListAzureBlobStorage](#)
 - [ListAzureDataLakeStorage](#)
 - [DeleteGCXObject](#)
 - [FetchGCXObject](#)
 - [ListGCSBucket](#)
 - [PutGCXObject](#)
 - [ProcFsMonitor](#)
 - [PutUDP](#)
 - [FetchFile](#)
 - [ListFile](#)
 - [PutSplunkHTTP](#)
 - [QuerySplunkIndexingStatus](#)

- Log collection from Kubernetes
- improved support for Lua processors
- platform independent [ListenSyslog](#)
- property update over C2 protocol

Build Notes:

It is advised that you use the bootstrap.sh when not building on windows.

Version cpp-0.11.0

Release Date: **13 December 2021**

New features in this release:

- new processors:
 - [AttributesToJson](#)
 - [DefragmentText](#)
 - [PutAzureDataLakeStorage](#)
 - [DeleteAzureDataLakeStorage](#)
 - [ReplaceText](#)
 - [RouteText](#)
- support for funnels
- shared RocksDB repository
- repository encryption (flow-file, content)
- support for Azure managed identity
- modularization of extensions
- ConsumeKafka security protocol
- SASL options for kafka processors
- platform independent AppendHostInfo
- agent configuration checksum in the C2 heartbeat
- We now use C++17 throughout the codebase and C++20 wherever possible.

Build Notes:

It is advised that you use the bootstrap.sh when not building on windows.

Version cpp-0.10.0

Release Date: **11 June 2021**

New features in this release:

- new processors:
 - ListS3
 - PutAzureBlobStorage
 - ConsumeKafka
 - PerformanceDataMonitor
 - ConsumeJournald
- add resource consumption data to heartbeats
- build with Visual Studio 2019 on Windows.

A few of the improvements and fixes:

- revive SQL processors
- fix expression language support in PublishKafka
- implement FollowRedirects and SendBody properties in InvokeHTTP
- add Initial Starting Position property to TailFile
- support credential refresh in AWSCredentialsService
- change default C2 protocol to REST
- plus bugfixes, compiler warning fixes etc.

Build Notes:

It is advised that you use the bootstrap.sh when not building on windows.

Version cpp-0.9.0

Release Date: **1 March 2021**

Highlights of 0.9.0 release include:

- Added support for RocksDB-based content repository for better performance
- Added SQL extension
- Improved task scheduling
- Various C2 improvements
- Bug fixes and improvements to TailFile, ConsumeWindowsEventLog, MergeContent, CompressContent, PublishKafka, InvokeHTTP
- Implemented RetryFlowFile and smart handling of loopback connections
- Added a way to encrypt sensitive config properties and the flow configuration
- Implemented full S3 support
- Reduced memory footprint when working with many flow files

Build Notes:

It is advised that you use the bootstrap.sh when not building on windows.

Version cpp-0.7.0

Release Date: **12 January 2020**

Highlights of 0.7.0 release include:

- Added several processors (OPC, SFTP and OpenCV to name a few)
- Added Windows specific processors (ConsumeEventLog for eg.)
- Improved Windows support (SiteToSite, processor features, OpenSSL support)
- Improved performance (related to usage of repositories)
- Windows build made trivial
- Fixed many issues including some security vulnerabilities

Build Notes:

It is advised that you use the bootstrap.sh when not building on windows.

Version cpp-0.6.0

Release Date: **22 March 2019**

Highlights of 0.6.0 release include:

- Began structuring library known as NanoFi
- Added several processors (WEL Reader, Sensor Readers, and HashContent to name a few)
- Added JNI Support to run NiFi processors in MiNiFi C++ flows, when Java is allowed and enabled
- Added Support for Natively written Python processors
- Added a CoAP package
- Windows support through Visual Studio with appveyor build
- Added docker builds for centos, debian, fedora, and u16
- Improved static build process across distros

Build Notes:

It is advised that you use the bootstrap.sh when not building on windows.

OS X: Certain versions of OS X have an incompatible version of Bison installed. The work around is to install Bison via brew and tell the build to use that version. Those steps are.

1. brew link bison --force
2. echo 'export PATH="/usr/local/opt/bison/bin:\$PATH"' >> ~/.bash_profile
 - a. Allows the brew Bison version to take precedence
3. source ~/.bash_profile

Ubuntu 16.X: Linking errors occur near the end of the build. A simple cmake flag can be used to ensure that the bundled version of RocksDB will be built and linked to the resulting binary

- cmake <all your desired flags> -DBUILD_ROCKSDB=true

Caveats:

This release represents an iterative developmental step in the MiNiFi C++ effort and is not considered ready for production and only provides a subset of functionality of the Java version.

- A full listing of additional caveats can be found at: <https://github.com/apache/nifi-minifi-cpp#caveats>

A full list of issues that were resolved can be found at <https://issues.apache.org/jira/secure/ReleaseNote.jspa?projectId=12321520&version=12343363>

Version cpp-0.5.0

Release Date: **6 June 2018**

Highlights of 0.5.0 release include:

- Added several functions to expression language
- Support for synchronous and asynchronous C2 control functions
- Support for UpdateAttribute
- Dynamic property support for processors
- SUSE support to bootstrap process
- Added RouteOnAttribute processor
- PutSQL (lite) implementation
- ExecuteSQL (lite) implementation

Build Notes:

OS X: Certain versions of OS X have an incompatible version of Bison installed. The work around is to install Bison via brew and tell the build to use that version. Those steps are.

1. brew link bison --force
2. echo 'export PATH="/usr/local/opt/bison/bin:\$PATH"' >> ~/.bash_profile
 - a. Allows the brew Bison version to take precedence
3. source ~/.bash_profile

Ubuntu 16.X: Linking errors occur near the end of the build. A simple cmake flag can be used to ensure that the bundled version of RocksDB will be built and linked to the resulting binary

- cmake <all your desired flags> -DBUILD_ROCKSDB=true

Caveats:

This release represents an iterative developmental step in the MiNiFi C++ effort and is not considered ready for production and only provides a subset of functionality of the Java version.

- A full listing of additional caveats can be found at: <https://github.com/apache/nifi-minifi-cpp#caveats>

A full list of issues that were resolved can be found at <https://issues.apache.org/jira/secure/ReleaseNote.jspa?projectId=12321520&version=12342659>

Version cpp-0.4.0

Release Date: **27 January 2018**

Highlights of 0.4.0 release include:

- Introduction of Expression Language into the framework and initial functions to perform string and numeric manipulations
- Introduction of a bootstrap script to help configure desired modules for a binary build
- Several new processors to provide functionality for GPS, MQTT, USB Cameras, Handlebars templates, and Tensorflow
- A C API for interacting with the framework
- Support for version 3 of the MiNiFi YAML config format
- Scripts for interrogating the controller API to get information about the instance's operation

Caveats:

This release represents an iterative developmental step in the MiNiFi C++ effort and is not considered ready for production and only provides a subset of functionality of the Java version.

- A full listing of additional caveats can be found at: <https://github.com/apache/nifi-minifi-cpp#caveats>

A full list of issues that were resolved can be found at <https://issues.apache.org/jira/secure/ReleaseNote.jspa?projectId=12321520&version=12341641>

Version cpp-0.3.0

Release Date: **2017 November 30**

Highlights of 0.3.0 release include:

- Creation of ExecuteScript processor using Python and Lua as underlying scripting engines
- Creation of a Lib Archive module that contains the MergeContent, CompressContent, and Focus/Unfocus archive processors
- PutKafka was created to support writing directly to Kafka
- Modules extensions are facilitated via CMAKE so that features can be enabled or disabled
- Command and Control facilities were created to support base C2 capabilities

Caveats:

This release represents an iterative developmental step in the MiNiFi C++ effort and is not considered ready for production and only provides a subset of functionality of the Java version.

- A full listing of additional caveats can be found at: <https://github.com/apache/nifi-minifi-cpp#caveats>

A full list of issues that were resolved can be found at <https://issues.apache.org/jira/secure/ReleaseNote.jspa?projectId=12321520&version=12341640>

Version cpp-0.2.0

Release Date: **2017 May 11**

Highlights of 0.2.0 release include:

- Incorporation of Catch testing framework and Google linting for code quality and enhanced test coverage
- Providing support for reporting tasks and an initial implementation of Site to Site Provenance reporting
- New Processors inclusive of PutFile, LlistenHTTP
- Modularization and namespacing of codebase

Caveats:

This release represents an iterative developmental step in the MiNiFi C++ effort and is not considered ready for production and only provides a subset of functionality of the Java version.

- A full listing of additional caveats can be found at: <https://github.com/apache/nifi-minifi-cpp#caveats>

A full list of issues that were resolved can be found at <https://issues.apache.org/jira/secure/ReleaseNote.jspa?version=12338790&projectId=12319921>

Version cpp-0.1.0

Release Date: **2016 December 2**

Highlights of 0.1.0 release include:

- Introduction of an ExecuteProcess processor
- Conversion to the CMake build system
- Framework support for generation of provenance

Caveats:

- This release represents an iterative developmental step in the MiNiFi C++ effort and is not considered ready for production and only provides a subset of functionality of the Java version.
 - A full listing of additional caveats can be found at: <https://github.com/apache/nifi-minifi-cpp#caveats>

A full list of issues that were resolved can be found at <https://issues.apache.org/jira/secure/ReleaseNote.jspa?version=12338046&projectId=12319921>.

Version cpp-0.0.1

Release Date: **2016 September 2**

Highlights of 0.0.1 release include:

- Initial framework implemented in C/C++
- Declarative configuration of processing flows through a YAML configuration file
- Initial set of processors consisting of
 - TailFile
 - GetFile
 - GenerateFlowFile
 - LogAttribute
 - ListenSyslog
- Site to Site Client implementation in C++ for talking to NiFi instances

Caveats:

- This release represents the first of the MiNiFi C++ effort and is not considered ready for production and only provides a subset of functionality of the Java version.
 - A full listing of additional caveats can be found at: <https://github.com/apache/nifi-minifi-cpp/blob/8f853c94db2ba15c8fff6af6a6dbf30defe1e83e/README.md#caveats>

A full list of issues that were resolved can be found at <https://issues.apache.org/jira/secure/ReleaseNote.jspa?version=12337990&projectId=12319921>.