



Installation Guide for Windows

Kinetic - Edge & Fog Processing Module (EFM) 1.2.0

Revised: November 30, 2017

Contents

Introduction.....	2
Features and Functions	2
The Edge and Fog Processing Module Components	3
Hardware Requirements.....	4
EFM Components Protocols and Ports	4
Licensing installation and requests	5
Required Libraries for Installation on Windows	5
Installation preparation	5
Executing the Installer	5
Installation of the EFM Components	6
Install help or usage:	6
Installing the Smart Licensing Tool:.....	8
Installing the EFM Message Broker:	8
Installing the System Administrator:	9
Installing the System Monitor:.....	9
Installing all of the EFM Windows components:	10
Installing DGLux5:	12
Upgrading to EFM version 1.2.0 from an existing version installation.....	13
Starting and Stopping EFM Message Broker	14
Starting and stopping the DGLux5 and message broker	14
Connecting to the EFM System Components	14
server.json Configuration options	16

Obtaining documentation and submitting a service request 20

Introduction

The Cisco Edge and Fog Processing Module (EFM) allows you to create a reliable data communications messaging system on top of your data networking infrastructure that provides data delivery and allows you to rapidly deploy applications where needed that can be at the edge, fog or in the datacenter. The Edge and Fog Processing Module is an open platform that allows for the addition of micro services or applications by anyone, allowing for unlimited capability and growth by adding software components that optimize the results of the application, system or outcome.

The EFM addresses the complexity of building an enterprise ready scalable data messaging system for one or many applications to reside upon. The EFM comes with a series of tools to manage the system, the EFM system administrator and the EFM system monitor.

Features and Functions

The system's key capabilities include:

- A framework for edge and fog processing. High performance.
- Reusable micro services for collecting data from, and providing control over, devices and machines, as well as processing the data prior to delivery to its destination.
- Different options for reliable transport of data through the system, encompassing both batch and real-time streaming options.
- Flexible mechanisms for integration with IT systems, reporting, and analytics.
- An architectural framework to extend fog processing to multiple tiers: east west (fog to fog) and north south (hierarchical processing leveraging network topology).
- Easy-to-use GUI tools to simplify development, deployment, and operation for all aspects of the system.
- A pervasive control paradigm and flow of information back to micro services, devices and machines for management, control, optimization and specific actions.
- A completely open and polyglot system, where third parties can provide devices, processing storage, software modules, analytics, applications, or any combination thereof.

This is the technology that makes IoT approachable, and leads to much faster industry adoption

of the vision of IoT.

The Edge and Fog Processing Module Components

EFM Message Broker	<p>Provides reliable and flexible data delivery between any devices and micro services. The sources can be devices like sensors or other micro services and consumers can be micro services or user applications.</p> <p>The EFM Message Broker is a small footprint component working with other brokers to form a message bus.</p>
EFM Data Flow Editor	Defines message paths between devices and micro services.
EFM Data Flow Engine	Executes message paths between devices and micro services. It is recommended to be installed adjacent to the EFM Message Broker in order to perform data transformation and input sources that not in the canonical data format of the system.
EFM System Administrator	Configures and manages the message broker and micro services.
EFM System Monitor	A standalone tool for operators to obtain real-time functional status of a deployed solution.
Cisco ParStream (Historian Database)	Purpose-built database for scale to handle the massive volumes and high velocity of IoT data as well as analytics at the Edge.
EFM Tools Runtime Engine	A standalone runtime tool for visualizing dashboards and driving EFM System Administrator, EFM Data Flow Engine and EFM System Monitor.
Links	<p>DQL Link – DSA Query Language</p> <p>System Link – System Information</p> <p>ParStream Link v3 – ParStream Historian Database</p>
Smart License Agent Tool for Nodes	The Smart License Agent client that allows system users to manage license registration for Node Product IDs
Smart License Agent Tool for Devices	The Smart License Agent client that allows system users to manage license registration for Device Product IDs

Hardware Requirements

EFM Message Broker EFM Data Flow Engine DQL Link System Link ParStream Link	Red Hat Linux 7.2, CentOS 7 or Ubuntu 16.04, Windows 2016 Server 1GB RAM, 10 GB HD* - Recommended on the same system/VM
EFM Data Flow Editor	Automatically installs with EFM Message Broker and EFM Tools Runtime Engine. Access via a web browser
EFM System Administrator	Project installs on the same system as the EFM Message Broker and EFM Tools Runtime Engine. Accessed via a web browser
EFM System Monitor	Project installs on the same system as the EFM Message Broker and EFM Tools Runtime Engine. Accessed via a web browser
Cisco ParStream (Historian Database)	Red Hat Linux 7.2, CentOS 7 or Ubuntu 16.04, 6 CPU cores with 2GB RAM per core, 500 GB HD
EFM Tools Runtime Engine	Installs with EFM Message Broker
Smart License Agent Tool for Nodes and Smart License Agent Tool for Devices	Redhat Linux 7.2, CentOS 7 or Ubuntu 16.04, with 1GB RAM, 10 GB HD.

EFM Components Protocols and Ports

The protocols and ports used by the EFM Broker and EFM Tools. The port values are configurable during and after installation.

EFM Broker and EFM Tools

Port No.	Exposure	Protocol	Description
8080	Public	HTTP	Default http or insecure port
8443	Public	HTTPS	Default https or secure port

Licensing installation and requests

This product uses the Smart License Agent Tool (for Nodes and Devices) to manage the corresponding licenses. After installation, refer to the Kinetic - Edge and Fog Processing Module 1.0 Smart License Agent User Guide.

Required Libraries for Installation on Windows

For the ParStream DSA Link: Java 8 JRE or JDK

Installation preparation

Place the EFM software image in the home directory.

The software should be downloaded from CCO at www.cisco.com under Support and Downloads.

Then **either** unpack the image by executing the unzip program directly (if on your system):

```
C:\Users\userid\> unzip -q EFM-1-2-0.zip
```

Or – alternatively - right click on the archive in the file explorer and select *extract all* (but do not create an additional folder as suggested per default, instead, we suggest to remove the EFM-1-2-0.zip from entry field)

Change into unzipped directory, e.g.:

```
C:\Users\userid\> cd %userprofile%\EFM-1-2-0
```

Executing the Installer

Execute the windows installer `efm-windows` as described in the following sections. Executing it without arguments yields a short usage info:

```
C:\Users\userid\EFM-1-1-0\> .\efm-windows
```

Installation of the EFM Components

The EFM installer `efm-windows` is designed to allow for interactive or non-interactive installation of the EFM components. The defaults allow for non-root users to operate the system.

All the examples below rely on default values and invocations refer to a fictitious `$HOME` being `C:\Users\userid` and further assume, that the package `EFM-1-2-0.zip` has been unpacked inside that folder (as described above).

Install help or usage:

Executing `.\efm-windows` displays a summary help screen.

```
Edge & Fog Processing Module - Installer and checksum tool v1.2.0

Synopsis: efm-windows [env|help|install|report|upgrade|verify|version]
[admin|broker|dart|dglux|monitor]

Note: Call with help for extended version including sample usage(s) or with env for environment info.

Copyright:
  (c) 2017 Cisco and/or its affiliates. All rights reserved.
```

Executing `.\efm-windows help` displays an extended help screen.

```
INFO: EFM_ROOT = C:\Users\userid\kinetic taken neither from file, nor from environment variable, but
instead from platform default!
INFO: Interactive mode enabled (reason default)
Edge & Fog Processing Module - Installer and checksum tool v1.2.0

Synopsis: efm-windows [env|help|install|report|upgrade|verify|version]
[admin|broker|dart|dglux|monitor]

Sample usage(s):

    efm-windows install          # -> install all platform components (dart, broker, admin, and
monitor)
    efm-windows upgrade        # -> upgrades all platform components (broker, admin, and monitor)

    efm-windows broker         # -> installs broker, data flow editor and dsa run-time
    efm-windows dart           # -> installs dart run-time
    efm-windows admin          # -> installs admin

    efm-windows verify         # -> report current settings and SHA512 checksums of components
    efm-windows verify dart    # -> calculate SHA512 checksum of dart run-time

    efm-windows version        # -> report version banner of this tool
    efm-windows [help]        # -> this help screen / usage info.
```



Notes:

Installing admin and monitor will also install Dart VM and Broker run-time.

Environment Variables:

EFM_ROOT should be set to the **absolute** path of the install root (default: \$HOME/kinetic)
(i.e. folders efm_server, parstream, and dart-sdk will be created below that path)
Current value is: <UNSET>

EFM_GUI_LOGIN may be set to the **name** of the EFM GUI Admin User (default: efmAdmin)
Current value is: <UNSET>

EFM_GUI_PHRASE may be set to the **pass phrase** of the EFM GUI Admin User (default: not set)
(This variable will be ignored for now in interactive install sessions)
Current value is: <UNSET>

EFM_BROKER_PRIV_KEY_PEM overwrites the default value key.pem for certKeyName in server.json
(i.e. with letsencrypt would suggest:
/etc/letsencrypt/live/fully.qualified.domain.name/privkey.pem)
Current value is: <UNSET>

EFM_BROKER_FULL_CHAIN_PEM overwrites the default value cert.pem for certName in server.json
(i.e. with letsencrypt would suggest:
/etc/letsencrypt/live/fully.qualified.domain.name/fullchain.pem)
Current value is: <UNSET>

EFM_BROKER_SECURE_PORT will overwrite the default value of 8443 for httpsPort in server.json
Current value is: <UNSET>

EFM_BROKER_CLEARTEXT_PORT will overwrite the default value of 8080 for port in server.json
(disable this insecure access port by setting value 0 in production systems!)
Current value is: <UNSET>

EFM_BROKER_IS_ALWAYS_OFFLINE will overwrite the default value to set isAlwaysOffline true in
server.json if set
(isAlwaysOffline indicates that a server is expected to never have a full internet connection)
Current value is: <UNSET>

EFM_BROKER_WORKERS may be set to positive integers in [1, 128] and should match the number of
logical cpu cores
as maximum and only if machine is dedicated to broker use and does not e.g. run local links or
ParStream DB
(if set and valid will overwrite the CPU core count derived one as value of the workers key in
servers.json)
Current value is: <UNSET>

EFM_INSTALL_LOGS may be set to the **absolute** path of a folder for install logs (default:
C:\Users\mdeleo\Desktop\EFM-1-2-0\EFM-1-2-0)
(e.g. needed in case the unpacked install components are stored on a read-only medium)
Current value is: <UNSET>

EFM_UNATTENDED may be set to fast enable unattended operation solely controlled by eff.json file
content.
Setting this to anything else than an empty value is equivalent to setting INTERACTIVE to false
in eff.json

EFM_DEBUG: For execution in debug mode, please set EFM_DEBUG environment variable to nonempty value

Copyright:

(c) 2017 Cisco and/or its affiliates. All rights reserved.

Executing `.\efm-windows` version displays the version.



Edge & Fog Processing Module - Installer and checksum tool v1.2.0

Executing `.\efm-windows env` reports the effective environment variable values and their source.

```
INFO: EFM_ROOT = C:\Users\userid\kinetic taken neither from file, nor from environment variable, but
instead from platform default!
INFO: Interactive mode enabled (reason default)
INFO: Effective environment variable status detected:
REPORT: - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8<
- - - -
REPORT: EFM_ROOT(from platform default): C:\Users\mdeleo\kinetic
REPORT: EFM_GUI_LOGIN(from config): efmAdmin
REPORT: EFM_GUI_PHRASE(from config): <empty>; Format: PBKDF2
REPORT: EFM_BROKER_PRIV_KEY_PEM(from config): key.pem
REPORT: EFM_BROKER_FULL_CHAIN_PEM(from config): cert.pem
REPORT: EFM_BROKER_SECURE_PORT(from config): 8443
REPORT: EFM_BROKER_CLEARTEXT_PORT(from config): 8080
REPORT: EFM_BROKER_IS_ALWAYS_OFFLINE(from config): BOOLEAN_FALSE
REPORT: EFM_BROKER_WORKERS(from config): 1
REPORT: EFM_INSTALL_LOGS(from config):
REPORT: EFM_UNATTENDED(from env): BOOLEAN_FALSE
REPORT: EFM_DEBUG: <UNSET>
REPORT: - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - -
- - >8 -
```

Installing the Smart Licensing Tool:

Executing `.\efm-windows license` reports the following error. The EFM licensing agent is not supported on Windows.

```
ERROR: License is only a valid install target on linux and osx platform
```

Installing the EFM Message Broker:

In this example, no global variables are set, all defaults are used.

This installs the EFM Message Broker and the DQL, System, and Dataflow engine DLinks.

Executing `.\efm-windows broker`

```
INFO: EFM_ROOT = C:\Users\userid\kinetic taken neither from file, nor from environment variable, but
instead from platform default!
INFO: Interactive mode enabled (reason default)
INFO: Sub-task install Dart VM (required by Dart broker) ...
INTERACTION: Installed platform dart run-time. Check that the Dart VM can execute and expose its
version? [Y/n]:
INFO: Dart VM check succeeded with Dart VM version: 1.21.1 (Fri Jan 13 01:50:41 2017) on "windows_x64"
INFO: Sub-task install Dart VM (required by Dart broker) succeeded
INFO: EFM Message Broker server configuration template copied successfully from
C:\Users\mdeleo\Desktop\EFM-1-2-0\EFM-1-2-0\Components\server.json
INTERACTION: Reconfigure the EFM Message Broker server configuration? [y/n]: y
INFO: EFM Message Broker server configuration reconfiguration requested, continuing ...
INFO: Set server json value of certName from EFM_BROKER_FULL_CHAIN_PEM to cert.pem
INFO: Set server json value of certKeyName from EFM_BROKER_PRIV_KEY_PEM to key.pem
INFO: Set server json value of httpsPort from EFM_BROKER_SECURE_PORT 8443
```




```
INFO: Set server json value of isAlwaysOffline from EFM_BROKER_IS_ALWAYS_OFFLINE to false
INFO: Set server json value of port from EFM_BROKER_CLEARTEXT_PORT to 8080
INFO: Set server json value of workers from EFM_BROKER_WORKERS to 1
INFO: EFM Message Broker server configuration updated with path
INFO: Installation created secure by default setup, good.
INTERACTION: Keep blocking unsecured creation of upstream connections over HTTP using the EFM Message
Broker installation? [Y/n]:
INFO: Setup kept secure as installed by default, good.
INFO: EFM Message Broker user configuration template copied ...
INTERACTION: Perform EFM Message Broker custom user configuration now? [Y/n]: y
INFO: EFM Message Broker custom user configuration requested, continuing ...
INTERACTION: EFM GUI Admin login is (efmAdmin). Change? [y/N]:

INTERACTION: Enter Password:
INTERACTION: Enter Password (verify):
INFO: Map the given login credentials into the EFM Message Broker user configuration and grant admin
rights ...
INFO: Creation of EFM GUI Admin User efmAdmin succeeded with ["User 'efmAdmin' was successfully
created."]
INFO: Grant admin rights for EFM GUI Admin User efmAdmin succeeded with ["User 'efmAdmin' was
successfully granted superuser."]
INFO: Users entry 1/1
REPORT: - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8<
- - - -
REPORT: Username is: efmAdmin
REPORT: Admin rights granted: true
REPORT: Passphrase in PBKDF2 format:
044153DuK1EJCScd5XGjzQVPu5ytn2iHEufABxaUj1YjJw=CRR/r8fFvNCzEw/h09EprU7AtpvE8y7wbi4S8CGaEzQ=
REPORT: - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - -
- - >8 -
INTERACTION: Installed broker and run-time. Check that the EFM Message Broker can execute and expose
its version? [Y/n]:
INFO: EFM Message Broker check succeeded with ['DSA Version: 1.1.2', 'DGLux5 Build: r7725', 'DGLux
Server Build: 1175', 'Verifying Environment...', 'Verifying Configuration...']
INFO: The component EFM Message Broker has been installed
```

Installing the System Administrator:

In this example, no global variables are set, all defaults are used.

The System Administrator is an option component that can be added to the broker for managing the system. It requires that the broker previously has been installed and configured to properly function and thus will trigger install of broker otherwise.

Executing .\efm-windows admin

```
INFO: EFM_ROOT = C:\Users\userid\kinetic taken neither from file, nor from environment variable, but
instead from platform default!
INFO: Interactive mode enabled (reason default)
INFO: The component EFM System Administrator has been installed
```

Installing the System Monitor:

In this example, no global variables are set, all defaults are used.

Typing .\efm-windows monitor



Since the System Monitor is envisioned to operate on an operations console, the installation takes the necessary steps of installing the message broker and system monitor project as a bundle. So, in contrast to above admin install, the below sample starts with an empty target folder.

```
INFO: EFM_ROOT = C:\Users\userid\kinetic taken neither from file, nor from environment variable, but
instead from platform default!
INFO: Interactive mode enabled (reason default)
INFO: Sub-task install EFM Message Broker (required by Monitor) ...
INFO: Sub-task install Dart VM (required by Dart broker) ...
INTERACTION: Installed platform dart run-time. Check that the Dart VM can execute and expose its
version? [Y/n]:
INFO: Dart VM check succeeded with Dart VM version: 1.21.1 (Fri Jan 13 01:50:41 2017) on "windows_x64"
INFO: Sub-task install Dart VM (required by Dart broker) succeeded
INFO: EFM Message Broker server configuration template copied successfully from
C:\Users\mdeleo\Desktop\EFM-1-2-0\EFM-1-2-0\Components\server.json
INTERACTION: Reconfigure the EFM Message Broker server configuration? [y/n]: y
INFO: EFM Message Broker server configuration reconfiguration requested, continuing ...
INFO: Set server json value of certName from EFM_BROKER_FULL_CHAIN_PEM to cert.pem
INFO: Set server json value of certKeyName from EFM_BROKER_PRIV_KEY_PEM to key.pem
INFO: Set server json value of httpsPort from EFM_BROKER_SECURE_PORT 8443
INFO: Set server json value of isAlwaysOffline from EFM_BROKER_IS_ALWAYS_OFFLINE to false
INFO: Set server json value of port from EFM_BROKER_CLEARTEXT_PORT to 8080
INFO: Set server json value of workers from EFM_BROKER_WORKERS to 1
INFO: EFM Message Broker server configuration updated with path
INFO: Installation created secure by default setup, good.
INTERACTION: Keep blocking unsecured creation of upstream connections over HTTP using the EFM Message
Broker installation? [Y/n]:
INFO: Setup kept secure as installed by default, good.
INFO: EFM Message Broker user configuration template copied ...
INTERACTION: Perform EFM Message Broker custom user configuration now? [Y/n]: y
INFO: EFM Message Broker custom user configuration requested, continuing ...
INTERACTION: EFM GUI Admin login is (efmAdmin). Change? [y/N]:

INTERACTION: Enter Password:
INTERACTION: Enter Password (verify):
INFO: Map the given login credentials into the EFM Message Broker user configuration and grant admin
rights ...
INFO: Creation of EFM GUI Admin User efmAdmin succeeded with ["User 'efmAdmin' was successfully
created."]
INFO: Grant admin rights for EFM GUI Admin User efmAdmin succeeded with ["User 'efmAdmin' was
successfully granted superuser."]
INFO: Users entry 1/1
REPORT: - 8< - - - - - 8< - - - - - 8< - - - - - 8< - - - - - 8< - - - - - 8< - - - - - 8< - - - - - 8<
- - - - -
REPORT: Username is: efmAdmin
REPORT: Admin rights granted: true
REPORT: Passphrase in PBKDF2 format:
044jyzhGbtI8VdrCJBLunkNhEoCl8CB6kr1Bnh8wYR35Y0=FvL+OHYwY0pGjX0PhS6MHNFMakDq/7WPvZgDR27a6Pk=
REPORT: - - - - - >8 - - - - - >8 - - - - - >8 - - - - - >8 - - - - - >8 - - - - - >8 - - - - - >8 - - - - -
- - >8 -
INTERACTION: Installed broker and run-time. Check that the EFM Message Broker can execute and expose
its version? [Y/n]:
INFO: EFM Message Broker check succeeded with ['DSA Version: 1.1.2', 'DGLux5 Build: r7725', 'DGLux
Server Build: 1175', 'Verifying Environment...', 'Verifying Configuration...']
INFO: Sub-task install EFM Message Broker (required by Monitor) succeeded
INFO: The component EFM System Monitor has been installed
```

Installing all of the EFM Windows components:

In this example, no global variables are set, all defaults are used.



For a complete installation of all the components on a host system, you can use this command option. We do not envision production systems will be designed with all the components on a single system, there are circumstances for development testing and learning environments that desire a quicker installation with one command.

This command installs the EFM Message Broker, EFM System Administrator, EFM System Monitor and EFM DataFlow Editor.

Typing .\efm-windows install

```
INFO: EFM_ROOT = C:\Users\mdeleo\kinetic taken neither from file, nor from environment variable, but
instead from platform default!
INFO: Interactive mode enabled (reason default)
INTERACTION: Dart VM already exist. Overwrite? [y/n]: no
Please inspect folders below C:\Users\mdeleo\kinetic and try install again after inspection or changes
have been made.
INFO: The component Dart VM has been installed
INFO: EFM Message Broker server configuration template copied successfully from
C:\Users\mdeleo\Desktop\EFM-1-2-0\Components\server.json
INTERACTION: Reconfigure the EFM Message Broker server configuration? [y/n]:
C:\Users\mdeleo\Desktop\EFM-1-2-0>efm-windows.exe install
INFO: EFM_ROOT = C:\Users\mdeleo\kinetic taken neither from file, nor from environment variable, but
instead from platform default!
INFO: Interactive mode enabled (reason default)
INTERACTION: Installed platform dart run-time. Check that the Dart VM can execute and expose its
version? [Y/n]:
INFO: Dart VM check succeeded with Dart VM version: 1.21.1 (Fri Jan 13 01:50:41 2017) on "windows_x64"
INFO: The component Dart VM has been installed
INFO: EFM Message Broker server configuration template copied successfully from
C:\Users\mdeleo\Desktop\EFM-1-2-0\Components\server.json
INTERACTION: Reconfigure the EFM Message Broker server configuration? [y/n]: y
INFO: EFM Message Broker server configuration reconfiguration requested, continuing ...
INFO: Set server json value of certName from EFM_BROKER_FULL_CHAIN_PEM to cert.pem
INFO: Set server json value of certKeyName from EFM_BROKER_PRIV_KEY_PEM to key.pem
INFO: Set server json value of httpsPort from EFM_BROKER_SECURE_PORT 8443
INFO: Set server json value of isAlwaysOffline from EFM_BROKER_IS_ALWAYS_OFFLINE to false
INFO: Set server json value of port from EFM_BROKER_CLEARTEXT_PORT to 8080
INFO: Set server json value of workers from EFM_BROKER_WORKERS to 1
INFO: EFM Message Broker server configuration updated with path
INFO: Installation created secure by default setup, good.
INTERACTION: Keep blocking unsecured creation of upstream connections over HTTP using the EFM Message
Broker installation? [Y/n]:
INFO: Setup kept secure as installed by default, good.
INFO: EFM Message Broker user configuration template copied ...
INTERACTION: Perform EFM Message Broker custom user configuration now? [Y/n]:
INFO: EFM Message Broker custom user configuration requested, continuing ...
INTERACTION: EFM GUI Admin login is (efmAdmin). Change? [y/N]:
INTERACTION: Enter Password:
INTERACTION: Enter Password (verify):
INFO: Map the given login credentials into the EFM Message Broker user configuration and grant admin
rights ...
INFO: Creation of EFM GUI Admin User efmAdmin succeeded with ["User 'efmAdmin' was successfully
created."]
INFO: Grant admin rights for EFM GUI Admin User efmAdmin succeeded with ["User 'efmAdmin' was
successfully granted superuser."]
INFO: Users entry 1/1
REPORT: - 8< - - - - - 8< - - - - - 8< - - - - - 8< - - - - - 8< - - - - - 8< - - - - - 8< - - - - - 8<
- - - -
```



```
REPORT: Username is: efmAdmin
REPORT: Admin rights granted: true
REPORT: Passphrase in PBKDF2 format:
044eFJF00775xkZze68csrCqld0Sgbp9AFgez1q9YVy5VM=cVM5keJuewfxZqftb+Vjv8k3u9N3fJ/VUFZcyfT/PqY=
REPORT: - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - -
- - >8 -
INTERACTION: Installed broker and run-time. Check that the EFM Message Broker can execute and expose
its version? [Y/n]:
INFO: EFM Message Broker check succeeded with ['DSA Version: 1.1.2', 'DGLux5 Build: r7732', 'DGLux
Server Build: 1179', 'Verifying Environment...', 'Verifying Configuration...']
INFO: The component EFM Message Broker has been installed
INFO: The component EFM System Administrator has been installed
INFO: The component EFM System Monitor has been installed
```

Installing DGLux5:

In this example, no global variables are set, all defaults are used.

Typing `.\efm-windows dglux`

```
INFO: EFM_ROOT = C:\Users\mdaleo\kinetic taken neither from file, nor from environment variable, but
instead from platform default!
INFO: Interactive mode enabled (reason default)
INFO: AcuityBrands DGLux Server server configuration template copied successfully from
C:\Users\mdaleo\Desktop\EFM-1-2-0\EFM-1-2-0\Components\server_dglux_certs.json
INFO: The component AcuityBrands DGLux Server has been installed
```

This installs the DGLux5 server only and the dataflow dslink. See the document *EFM and DGLux5 Installation Guide* for details on installing and running the EFM message broker and the DGLux5 on the same host.

If the EFM broker has already been installed with the DART or the DART VM has not been installed as a standalone, then it must be installed before the DGLux5 can be run. The DART VM is used both the EFM message broker and DGLux5.

Typing `.\efm-windows dart`

```
INFO: EFM_ROOT = C:\Users\mdaleo\kinetic taken neither from file, nor from environment variable, but
instead from platform default!
INFO: Interactive mode enabled (reason default)
INTERACTION: Installed platform dart run-time. Check that the Dart VM can execute and expose its
version? [Y/n]:
INFO: Dart VM check succeeded with Dart VM version: 1.21.1 (Fri Jan 13 01:50:41 2017) on "windows_x64"
INFO: The component Dart VM has been installed
```

Upgrading to EFM version 1.2.0 from an existing version installation

If an existing version 1.0.0, 1.0.1 or 1.1.0 is installed and running on a Windows server it is possible to upgrade to the version 1.2.0. It is highly recommended performing a backup of the host prior to proceeding with the upgrade.

Assuming that the EFF version 1.0.x or 1.1.0 is already installed and running. Here are the prerequisites to upgrading:

- Download the EFM 1.2.0 package
- Stop the running message broker (see below)

Please take note of the following name changes that might impact the upgrade process:

- If no explicit EFF_ROOT was set for an installation with EFF 1.0.x, or 1.1.0, all components were installed below C:\cisco\iotdc\
 - The environment variable name for the install root folder has been changed from EFF_ROOT to EFM_ROOT
- If you now upgrade to EFM 1.2, the new default path %userprofile%\kinetic is empty. So, you have two alternatives:
 - either set EFM_ROOT to C:\cisco\iotdc to upgrade the setup in the prior location
 - or move C:\cisco\iotdc\ to %userprofile%\kinetic (and adapt config entries in eff_server/server.json where applicable)
- be aware, that the efm-tool will rename *eff_* subfolders during the upgrade process. An existing *eff_server* will be renamed to *efm_server*, *eff_license* will become *efm_license*.

Perform the following steps to upgrade in a Command Window:

Prepare upgrade like a fresh install: Unpack the EFM-1-2-0.zip accordingly and change current working directory to EFM-1-2-0 folder all as described above in the install section (cf. there if unzip not present or something does not work):

```
C:\Users\userid\> unzip -q EFM-1-2-0.zip
C:\Users\userid\> cd %userprofile%\EFM-1-2-0
C:\Users\userid\EFM-1-2-0\>
```

Note directory of the destination as the default version 1.0.x installation, for example:

```
C:\Users\userid\EFM-1-2-0\> set EFM_ROOT=C:\cisco\kinetic\
```



Note: If your installation path differs from this default path, please adapt the corresponding paths in the following commands. If it is already %userprofile%\kinetic no setting of EFM_ROOT variable is required, but then below commands must be adapted to function properly.

Stop the broker:

```
C:\> %EFM_ROOT%\dart-sdk\bin\dart %EFM_ROOT%\efm_server\bin\daemon.dart stop
```

Upgrade the components:

```
C:\Users\userid\EFM-1-2-0\> .\efm-windows upgrade
```

Update the static URL path to the “.well-known” resource inside efm_server/server.json config file, so that it matches the upgraded location (just change eff_server to efm_server inside path)

Start the broker:

```
C:\> %EFM_ROOT%\dart-sdk\bin\dart %EFM_ROOT%\efm_server\bin\daemon.dart start
```

Starting and Stopping EFM Message Broker

Note: Assuming the default installation path root stored in variable EFM_ROOT in the following examples.

Start the broker:

```
C:\> %EFM_ROOT%\dart-sdk\bin\dart %EFM_ROOT%\efm_server\bin\daemon.dart start
```

To stop the broker (when demo or test complete):

```
C:\> %EFM_ROOT%\dart-sdk\bin\dart %EFM_ROOT%\efm_server\bin\daemon.dart stop
```

Starting and stopping the DGLux5 and message broker

Note: Assuming the default installation path root stored in variable EFM_ROOT in the following examples.

Start the broker:

```
C:\> %EFM_ROOT%\dart-sdk\bin\dart %EFM_ROOT%\dglux_server\bin\daemon.dart start
```

To stop the broker (when demo or test complete):

```
C:\> %EFM_ROOT%\dart-sdk\bin\dart %EFM_ROOT%\dglux_server\bin\daemon.dart stop
```

Connecting to the EFM System Components



All the EFM tools require a username and login for access. The administrator can add additional users in the System Administrator after the first login.

Application	Insecure Port (if supported) ¹	Secure Port
EFM System Administrator	http://[Server IP Address]/efm-admin	https://[Server IP Address]/efm-admin
EFM System Monitor	http://[Server IP Address]/efm-monitor	https://[Server IP Address]/efm-monitor
EFM Data Flow Editor	http://[Server IP Address]/dataflow.html	https://[Server IP Address]/dataflow.html

¹ Note if the server configuration is using Safe Mode, income connections will automatically redirect to the secure port if configured. If no secure port is defined or certificates and passwords are not properly configured, the secure port will not accept connections and therefore in Safe Mode the access to the system is unavailable.

server.json Configuration options

Example server.json configuration file.

```
{
  "allowAllLinks": true,
  "allowBrowserCaching": false,
  "allowPasswordChanges": true,
  "alternativeBrokerUrl": null,
  "authType": "file",
  "broadcast": false,
  "brokerName": "broker-",
  "certKeyName": "key.pem",
  "certName": "server.pem",
  "certPassword": "",
  "corsProxyRules": "",
  "dartRuntimeManagerVmFlags": [],
  "debug": false,
  "defaultPermission": null,
  "disableFileSecurity": false,
  "disabledLinks": [],
  "distributionUrl": "NO",
  "downstreamName": "downstream",
  "enableCertificateGeneration": true,
  "enableGit": false,
  "enableIPv6": false,
  "enableSingleSignOnServer": false,
  "enableUptimeChecker": true,
  "formatDg5": false,
  "generatedCertificateSubject": "/C=US/ST=California/L=Oakland/O=DGLogik Inc./OU=Customers/CN=*",
  "guestLoginRedirectPath": "/assets/",
  "hooks": {},
  "host": "0.0.0.0",
  "httpPathClassification": {},
  "httpsPort": 8443,
  "isAlwaysOffline": false,
  "javaRuntimeManagerVmFlags": [],
  "keepCustomAssets": true,
  "linkConfig": {},
  "linkManagerEnvironment": {},
  "linkRepositoryUrl": "https://dsa.s3.amazonaws.com/links/links.json",
  "logRotationInterval": 0,
  "loggers": [],
  "loginRedirectPath": "/",
  "observe": false,
  "passwordHasherIterations": 1000,
  "passwordHasherKeyLength": 32,
  "port": 8080,
  "proxies": {},
  "quarantine": false,
  "runBrokerInMain": true,
  "runPortChecks": true,
  "serverLogLevel": "INFO",
  "serverVmFlags": [],
  "ssoProviderUrl": null,
  "static": {
    "/.well-known": "/opt/cisco/kinetic/efm_server/.well-known"
  },
  "storageDriver": "simple",
  "timeHttpRequests": false,
  "twoFactorAuth": "none",
  "updateInterval": 200,
  "upstream": {},
}
```




```
"uptimeCheckUrl": null,  
"useDartRuntimeManager": false,  
"useJavaRuntimeManager": false,  
"useRuntimeManager": false,  
"userTimeout": 525600,  
"workers": 1  
}
```

Option	Description	Default Value	Comments
debug	Enable/Disable Debugging Mode	false	For production site, this should always be false, debug:true may result in memory leak and bugs. port
port	HTTP Port to listen on. If this is less than or equal to 0, then the server does not listen on any port for HTTP.	8080	At least one of port or httpsPort must have a valid port number assigned.
httpsPort	HTTPS port to listen on. If this is less than or equal to 0, and/or certName or certPassword is not provided, then the server does not listen on any port for HTTPS. Ensure that if you install a custom certificate, you fill in the certName, certKeyName and certPassword fields.	8443	At least one of port or httpsPort must have a valid port number assigned. certName
certName	SSL certificate file name. Leave blank to disable HTTPS		
certPassword	SSL certificate password. Set to null to disable HTTPS		
certKeyName	SSL private key file name. Leave blank to disable HTTPS		
disableFileSecurity	When this value is true, then any user can access any file. When this is false, file permissions are checked.	false	
broadcast	When this value is true, the server's broker is broadcast to the local network for discovery by other machines. When this value is false, the broadcast service is not enabled.	true	
workers	Number of Server Workers. For low end devices, this should stay	For single-core machines, this is 1,	



	at 1. For large machines, this can be set up to a maximum of 128. It is recommended that you do not exceed the number of logical processors on your machine.	for other devices, this is 2.	
static	Configures a static directory mapping. This is used to serve files and directories on the server. Example: { "/static": "/srv/http/static" }	{"/.well-known": "/path/to/dsa/dglux-server/.well-known"}	
defaultPermission	Default permission setting for the root node. When this value is null, permissions are disabled, and everything has the config permission.		
allowAllLinks	When the value is true, all incoming DSLink connections will be accepted to /downstream. When the value is false, an incoming DSLink without proper authentication will be rejected unless quarantine is enabled.	true	
quarantine	** This setting has no effect when allowAllLinks is true ** When the value is true, a new incoming DSLink without a token will be put in /sys/quarantine. A quarantined DSLink can only work as a responder. Use the /sys/quarantine/authorize to move a quarantined DSlink to /downstream.	false	
isAlwaysOffline	Indicates that a server is expected to never have a full internet connection.	false	
useDartRuntimeManager	When the value is true, the Dart Runtime Manager is used for Dart DSLinks. The Dart runtime manager reduces resource consumption by merging Dart	false	



	DSLlinks into a single process.		
useJavaRuntimeManager	When the value is true, the Java Runtime Manager is used for Java DSLinks. The Java runtime manager reduces resource consumption by merging Java DSLinks into a single process.	false	
guestLoginRedirectPath	Determines the URI that a user is redirected to when login is complete.	/	
authType	Determines the authentication provider to use.	file	
twoFactorAuth	Determines the two factor authentication provider to use. Supported Two-Factor Authentication Providers <ul style="list-style-type: none">• none: Don't enable two factor authentication.• duo: Duo Two-Factor Authentication	none	
enableIPv6	Toggles support for IPv6 connections	false	
keepCustomAssets	When the value is true, custom assets in www/assets are kept upon updating EFM Server.	false	
formatDg5	When this value is true, efm client will save dg5 in a formatted and json with key sorted, makes it easy to track changes.	false	



Obtaining documentation and submitting a service request

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at:

<http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html>

Subscribe to the *What's New in Cisco Product Documentation* as a Really Simple Syndication (RSS) feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service and Cisco currently supports RSS Version 2.0.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.