enginframe

Quick Start Guide

Version **2020.0**

NICE
an aws company

Copyright

We'd Like to Hear from You

You can help us make this document better by telling us what you think of the content, organization, and usefulness of the information. If you find an error or just want to make a suggestion for improving this document, please address your comments to <`documentation@nice-software.com`>. Please send only comments regarding NICE documentation.

For product support, contact <`helpdesk@nice-software.com`>.

Although the information in this document has been carefully reviewed, NICE s.r.l. ("NICE") does not warrant it to be free of errors or omissions. NICE reserves the right to make corrections, updates, revisions, or changes to the information in this document.

UNLESS OTHERWISE EXPRESSLY STATED BY NICE, THE PROGRAM DESCRIBED IN THIS DOCUMENT IS PROVIDED "AS IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL NICE BE LIABLE TO ANYONE FOR SPECIAL, COLLATERAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING WITHOUT LIMITATION ANY LOST PROFITS, DATA, OR SAVINGS, ARISING OUT OF THE USE OF OR INABILITY TO USE THIS PROGRAM.

Document Redistribution and Translation

Trademarks

EnginFrame, Neutro, Remote File Browsing, Service Definition File, EnginFrame Agent are registered trademarks or trademarks of NICE s.r.l. in Italy and other countries.

Amazon™ is a registered trademark of Amazon.com, Inc.

Apache®, Apache Derby®, Tomcat® are either registered trademarks or trademarks of the Apache Software Foundation in the United States and/or other countries.

Oracle®, Sun®, MySQL®, JavaScript® and Java™ are registered trademarks of Oracle and/or its affiliates.

Unix® is a registered trademark of The Open Group in the United States and other countries.

Linux® is the registered trademark of Linus Torvalds in the United States and other countries.

Microsoft®, Windows® and Internet Explorer® are either registered trademarks or trademarks of Microsoft Corporation in the United States and other countries.

Firefox® and Mozilla® are trademarks or registered trademarks of the Mozilla Foundation in the United States and/or other countries.

Apple®, Mac®, Mac® OS X® and Apple® Safari® are trademarks or registered trademarks of Apple, Inc. in the United States and other countries.

IBM®, IBM® Platform™ LSF® are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide.

Altair® PBS Professional® is a trademark of Altair Engineering, Inc.

Univa® and Univa® Grid Engine® (UGE) are trademarks of Univa Corporation.

SLURM™ is a trademark of SchedMD LLC.

RealVNC® and VNC® are trademarks of RealVNC Limited and are protected by trademark registrations and/or pending trademark applications in the European Union, United States of America and other jurisdictions.

HP® is a registered trademark of HP Inc.

Google™ and Chrome™ are trademarks of Google Inc.

Red Hat® is a trademark of Red Hat, Inc.

SUSE® is a registered trademark of SUSE Linux AG.

Last Update
November 17, 2020 (rev. 58)

Latest Version
`https://www.enginframe.com`

Contents

Welcome ... vii
About This Guide ... vii
Who Should Read This Guide ... vii
How to Find Out More ... vii
Get Technical Support ... vii
NICE Support Contacts ... vii
1. Introduction ... 1
2. Getting Started ... 3
Downloading EnginFrame ... 3
Obtaining a License ... 3
System Requirements ... 4
Third-party Software Prerequisites ... 4
Java™ Platform ... 4
Database Management Systems ... 4
Authentication Mechanisms ... 5
Distributed Resource Managers ... 6
Remote Visualization Technologies ... 10
Supported Browsers ... 12
Interactive Plugin Requirements ... 12
Session Managers ... 13
Single Application Desktop Requirements (Linux®) ... 13
Shared File System Requirements ... 13
3. Installation ... 15
Prepare for Installation ... 15
Installing ... 15
Updating ... 16
Upgrading from pre-2015 versions ... 16
Configuration File Mapping ... 17
Starting EnginFrame ... 19
Checking EnginFrame Status ... 19
Accessing EnginFrame ... 19
Troubleshooting ... 20

Welcome

About This Guide

This guide is your starting point for learning how to install and setup an instance of the NICE EnginFrame portal.

Who Should Read This Guide

This guide is intended for new EnginFrame administrators who want to familiarize themselves with the fundamentals of installing and setting up an instance of the NICE EnginFrame portal.

How to Find Out More

Refer to the *EnginFrame Administrator's Guide* to learn more about how to install and manage an instance of the NICE EnginFrame portal.

Get Technical Support

Contact NICE or your EnginFrame reseller for technical support.

NICE Support Contacts

Use one of the following to contact NICE technical support.

Email
: <`helpdesk@nice-software.com`>

Phone
: +39 0141 901516

When contacting NICE, please include your company's full name.

1

Introduction

NICE EnginFrame 2020.0 is the leading grid-enabled application portal for user-friendly submission, control and monitoring of HPC jobs and interactive remote sessions.

It includes sophisticated data management for all stages of HPC job lifetime and is integrated with most popular job schedulers and middleware tools to submit, monitor, and manage jobs.

NICE EnginFrame provides a modular system where new functionality (e.g. application integrations, authentication sources, license monitoring, etc.) can be easily added. It also features a sophisticated Web Services interface which can be leveraged to enhance existing applications as well as developing custom solutions for your own environment.

A key advantage of NICE EnginFrame is the rapid migration from command-line to *Computing Portal* paradigm, leveraging existing scripting solutions where available.

In addition, NICE EnginFrame 2020.0 enables launching and managing interactive applications running on a remote cluster via a web browser.

It features a flexible architecture which supports different remote visualization technologies such as *NICE DCV*, *VNC®*, *HP® RGS* or *VirtualGL* on both Linux® and Windows® platforms.

Based on the latest and most advanced Web 2.0 standards, NICE EnginFrame provides a flexible infrastructure to support current and future computing and remote visualization needs. As you can expect from a web-enabling solution, it is flexible in content presentation, providing personalized experience for users according to their role or operational context.

2

Getting Started

Note

This guide covers the most common configuration where EnginFrame Server (including Apache Derby®) and EnginFrame Agent are installed on the same machine, refer to the *EnginFrame Administrator's Guide* for more complex configurations.

Downloading EnginFrame

EnginFrame packages can be downloaded from:

`https://www.enginframe.com`

You need a valid account to access the download area. If you do not have one yet, please contact NICE or your EnginFrame reseller.

Obtaining a License

You do not need a license to install and run EnginFrame on Amazon EC2. When used on an on-premises or alternative cloud-based server, you need a valid license to install and run EnginFrame.

EnginFrame licenses are classified as:

- *Demo licenses* - demo licenses are usually not bound to any IP address and are valid for one month.
- *Full licenses* - full licenses have time-unlimited validity and are bound to one or more IP addresses.
- *Year licenses* - year licenses have time-limited validity and are bound to one or more IP addresses.

Please, contact NICE or your EnginFrame reseller to purchase, renew, or update a license, perform a license change or request a demo license.

System Requirements

NICE EnginFrame supports the following operating systems[1]:

- Amazon™ Linux® release 2016.03 or above
- Red Hat® Enterprise Linux® 5.x, 6.x, 7.x (*x86-64*)
- SUSE® Linux® Enterprise Server 11 SP2, 12 SP3 (*x86-64*)

The installation machine must have at least 3 GB of RAM and one or more IP addresses (at least one of them reachable by each of the potential client machines, directly or via proxies).

To install EnginFrame you need at least 200 MB of free disk space, but 2 GB or more are suggested since, while operating, the software saves important data and logging information.

Please, make sure you have enough space for the service data stored inside the EnginFrame spoolers. By default, spoolers are located inside the EnginFrame installation directory (`$EF_TOP/spoolers`).

Third-party Software Prerequisites

Besides the standard packages installed with your operating system, NICE EnginFrame requires some additional third-party software.

Java™ Platform

NICE EnginFrame requires the *Linux® x64* version of *Oracle® Java™ Platform Standard Edition* (Java™ SE 8) or *OpenJDK Runtime Environment 8*.

From now on, we will call `JAVA_HOME` the Java™ installation directory.

The same Java™ version *must* be used for both EnginFrame Server and EnginFrame Agent.

Database Management Systems

EnginFrame requires a *JDBC-compliant* database. EnginFrame uses the RDBMS to manage *Triggers*, *Job-Cache* and *Applications* and *Views* users' groups. EnginFrame *Triggers* rely on Quartz[2] engine to schedule the execution of EnginFrame services. Triggers are used internally to execute periodic tasks as to check and update Interactive sessions status and to collect EnginFrame usage statistics informations. The *Job-Cache* feature is responsible for collecting and caching job statuses over time.

By default Apache Derby® 10.14 database is installed together with EnginFrame Professional, however using Apache Derby® in a production installation is not recommended.

Apache Derby® is not supported for EnginFrame Enterprise installations, it is strongly suggested to use an external JDBC-compliant RDBMS. Since EnginFrame Enterprise is part of a HA solution, also the RDBMS must have its own HA strategy. The external RDBMS is suggested to reside on a different node(s) than the EnginFrame servers and possibly configured to be fault tolerant.

[1] Other Linux® distributions and compatible Java™ versions might work but are not officially supported. Contact <`helpdesk@nice-software.com`> for more information.

[2] http://www.quartz-scheduler.org

EnginFrame supports MySQL® Database 8.0.x. EnginFrame has been used widely with other databases (Oracle® Database, SQL Server®, MariaDB®) but we only offer best effort support on them. In case of issues with a specific RDBMS version, please contact helpdesk@nice-software.com.

EnginFrame provides the JDBC driver for Apache Derby® only. In case a different DBMS is used, the JDBC driver must be added after the installation to the `$EF_TOP/<VERSION>/enginframe/WEBAPP/WEB-INF/lib` directory.

Please refer to the DBMS documentation for instructions on how to get the proper JDBC driver and configure it.

Authentication Mechanisms

EnginFrame supports different authentication mechanisms. Some of them require third-party software components.

Refer to Table 2.1, “Supported Authentication Mechanisms” to select the most appropriate authentication method for your system and check its third-party software prerequisites (if any).

Table 2.1. Supported Authentication Mechanisms

Name	Prerequisites	Notes
PAM	Linux® PAM must be correctly configured	It is the most common authentication method. It allows a system administrator to add new authentication methods simply by installing new PAM modules, and to modify authentication policies by editing configuration files. At installation time, you will be asked to specify which PAM service to use, **system-auth** is the default.
LDAP	The **ldapsearch** command must be installed and working appropriately on the EnginFrame Agent host	These methods allow you to authenticate users against a LDAP or Active Directory server. The EnginFrame installer will ask you to specify the parameters needed by **ldapsearch** to contact and query your directory server.
Active Directory		
HTTP Authentication	External HTTP authentication system	This method relies on an external authentication system to authenticate the users. The external system then adds an HTTP authentication header to the user requests. EnginFrame will trust the HTTP authentication header.
Certificate	SSL Certificates need to be installed and exchanged between EnginFrame Server and clients.	This method relies on the authentication accomplished by the web server, which requires the client authentication through the use of SSL certificates.

The EnginFrame installer can optionally verify if you have correctly configured the selected authentication method.

NICE EnginFrame can be easily extended to add support for custom authentication mechanisms.

Distributed Resource Managers

EnginFrame supports different distributed resource managers (DRM).

At installation time, you will need to specify which DRMs you want to use and provide the information required by EnginFrame to contact them. A single EnginFrame instance can access more than one DRM at the same time.

Refer to Table 2.2, “Supported Distributed Resource Managers” for a list of supported DRMs.

Table 2.2. Supported Distributed Resource Managers

Name	Version	Notes
IBM® Platform™ LSF®	6.x (Deprecated) - 10.1.x	The LSF client software must be installed on the EnginFrame Agent host. The installer will ask you to specify the LSF profile file.
Altair® PBS Professional®, OpenPBS®	Altair® PBS Professional®: 7.x - 14.x (Deprecated), 19.2.x - 2020.1.x OpenPBS®: 19.1.x - 20.0.x	The OpenPBS® or PBS Professional® client software must be installed on the EnginFrame Agent host. The installer will ask you to specify the directory where the OpenPBS® or PBS Professional® client software is installed.
Torque	6.0.x - 6.1.x (Deprecated)	The Torque client software must be installed on the EnginFrame Agent host. The installer will ask you to specify the directory where the Torque client software is installed.
NICE Neutro	2013 or later (Deprecated)	The NEUTRO master(s) must be reachable from the EnginFrame Server host. The installer will ask you to specify the IP address of your NEUTRO masters.
SLURM™	19.05.x - 20.0.x	SLURM™ binaries must be installed on the EnginFrame Server host. SLURM™ master host must be reachable from the EnginFrame Server host. The installer will ask you to specify the path where binaries are installed. On SLURM™ configuration, specifically related to compute nodes dedicated to interactive sessions, the Features: vnc,dcv,dcv2 and RealMemory parameters must be added to every required node. 'dcv2' stands for DCV since 2017.
Sun® Grid Engine (SGE)	6.2 (Deprecated), 8.1.x	The Grid Engine client software must be installed on the EnginFrame Agent host. The `$SGE_ROOT/$SGE_CELL/common` must be shared from SGE master to EF nodes.

Name	Version	Notes
Oracle® Grid Engine (OGE)	7.0 (Deprecated)	The installer will ask you to specify the Grid Engine shell settings file.
Univa® Grid Engine® (UGE)	8.6.x	
Son of Grid Engine (SoGE)	8.1.x	
Open Grid Scheduler	2011.x (Deprecated)	
AWS Batch	The AWS Batch cluster must be created with AWS ParallelCluster 2.1.0 or later	The installer will ask you to specify the AWS ParallelCluster cluster name and the AWS region.

Some schedulers like Torque, PBS Professional® and Univa® Grid Engine® (UGE) 8.2.0 have job history disabled by default. This means that a job will disappear when finished. It is strongly suggested to configure these distributed resource managers to retain information about the finished jobs. For more information on the configuration check the section called “Required DRM Configuration”.

Deprecated integrations will not be supported beginning with a later release currently planned for July 2021, except for NEUTRO which will end support December 2021.

Required DRM Configuration

Altair® PBS Professional®

Applies to versions: 11, 12, 14

Altair® PBS Professional® by default does not show finished jobs. To enable job history, a server parameter must be changed:

```
qmgr -c "set server job_history_enable = True"
```

Once enabled, the default duration of the job history is 2 weeks.

Torque

Applies to versions: 4, 5, 6

Torque by default does not show finished jobs. To enable job history, a queue parameter must be changed:

```
qmgr -c "set queue batch keep_completed=120"
```

The `keep_completed` parameter specifies the number of seconds jobs should be held in the Completed state after exiting.

Once enabled, the default duration of the job history is 2 minutes.

Applies to versions: all

In order to get the *Display Output* functionality for *Torque* jobs, *qpeek* tool should be configured properly. By default *qpeek* uses the *rsh* command to remote access the so-called "mother superior" node.

Choose between installing *rsh* on the nodes or configure *qpeek* to use *ssh* and install *ssh* configured passwordless among the nodes.

Univa® Grid Engine® (UGE)

Applies to versions: 8.2.x

Univa® Grid Engine® (UGE) by default does not show finished jobs. To enable job history:

- *(8.2.0 only)* disable reader threads:

  edit file `SGE_ROOT/SGE_CELL/common/bootstrap`

  set `reader_threads` to 0 instead of 2

- enable finished jobs:
  run

  ```
  qconf -mconf
  ```

  set `finished_jobs` to a non-zero value according to the rate of finishing jobs.

  The `finished_jobs` parameter defines the number of finished jobs stored. If this maximum number is reached, the eldest finished job will be discarded for every new job added to the finished job list.

  By default EnginFrame grabs the scheduler jobs every minute. The `finished_jobs` parameter must be tweaked so that a finished job stays in the job list for at least a minute. Depending on the number of jobs running in the cluster a reasonable value is in between *the medium number of running jobs* and *the amount of jobs ending per minute*.

- restart qmaster

SLURM™

Applies to versions: all

SLURM™ show finished jobs for a default period defined by the `MinJobAge` parameter in file `slurm.conf` (under `/etc/slurm` or the SLURM™ configuration directory). The default value is *300* seconds, i.e. five minutes, which is acceptable.

In case you changed this parameter, ensure it is not set to a value lower than *300*.

Also check the `MaxJobCount` parameter is not set.

After changing this parameter restart SLURM™ with:

```
/etc/init.d/slurm stop
/etc/init.d/slurm start
```

The setting must be done on all SLURM™ nodes.

IBM® Platform™ LSF®;

Applies to versions: all

IBM® Platform™ LSF® shows finished jobs for a default period defined by the `CLEAN_PERIOD` parameter in file `lsb.params`. The default value is *3600* seconds, i.e. one hour, which is acceptable.

In case you changed this parameter, ensure it is not set to a value lower than *300*.

After changing this parameter run:

```
badmin reconfig
```

AWS Batch

To integrate EnginFrame with AWS Batch it is required to create a Batch cluster with AWS ParallelCluster and give the user running the EnginFrame Server the permission to interact with the cluster. Here the details of the required steps:

- Install AWS ParallelCluster and configure it following the instruction here.
  AWS CLI will be installed as dependency of AWS ParallelCluster.

- Taking into account the network requirements, create a new cluster for AWS Batch scheduler.

- Go to the CloudFormation console from the AWS Account and click on the created Stack.
  Get the `Stack ID` from the Stack Info tab view. (e.g. `arn:aws:cloudformation:<REGION>:<ACCOUNT>:stack/<STACK_NAME>/<UID>`)

- Get the `BatchUserRole` from the Outputs tab view, using the AWS CloudFormation console, or through the status command of the AWS ParallelCluster command line. (e.g. `arn:aws:iam::<ACCOUNT>:role/<STACK_NAME>-suffix`).

- Use the `BatchUserRole` and the `Stack ID` (by replacing the latest UID with an asterisk) to create, through the Identity and Access Management (IAM) console, a new IAM Policy like the following:

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
        "sts:AssumeRole",
        "cloudformation:DescribeStacks"
      ],
      "Resource":[
        "<BatchUserRole>",
        "<StackID with the UID replaced by *>"
      ]
    }
  ]
}
```

- Create a new IAM User and assign the created Policy to it.

- From the IAM Console, click on the created user and get the user credentials from the Security Credentials tab view. Use them to configure the AWS CLI for the user running the EnginFrame Server (e.g. `efnobody`).

```
[efnobody]$ aws configure
```

- Follow EnginFrame installer steps to configure AWS Batch EnginFrame plugin to contact the created cluster.

Remote Visualization Technologies

EnginFrame supports different remote visualization technologies, and the same EnginFrame instance can manage multiple of them. Please refer to the following table for the supported ones.

Table 2.3. Supported Remote Visualization Technologies

Name	Version	Notes
RealVNC® Enterprise Edition	4.x or 5.x (Deprecated)	It allows to share sessions both in full access or view only mode.
TigerVNC	1.x (Deprecated)	Linux® only (server side).
TurboVNC	2.2	Linux® only (server side).
RealVNC® Free Edition	4.x (Deprecated)	Linux® only (server side).
NICE DCV	2012.x to 2016.x (Deprecated), 2017.x or later	It allows to share sessions both in full access or view only mode.
VirtualGL	2.5 or later	
HP® RGS	5.x, 6.x or 7.x (Deprecated)	Please refer to EnginFrame Administrator's Guide for more information on how to set up your DRM system on Linux® nodes to work with HP® RGS.

For detailed instructions on how to install and configure these remote visualization technologies please refer to their respective manuals.

Deprecated integrations will not be supported beginning with a later release currently planned for July 2021, except for NEUTRO which will end support December 2021.

Remote Visualization Technologies Configuration

NICE DCV 2017.0 or later on Linux

For Linux environments the configuration of the authentication to use with NICE DCV must correspond to the authentication system set on the DCV server in the remote visualization hosts.

On EnginFrame the authentication to use with DCV on Linux can be set in the `INTERACTIVE_DEFAULT_DCV2_LINUX_AUTH` configuration parameter inside the `$EF_TOP/conf/plugins/interactive/interactive.efconf` file.
Default value and documentation can be found in the static configuration file `$EF_TOP/<VERSION>/enginframe/plugins/interactive/conf/interactive.efconf`.

The auto authentication system, providing seamless authentication with self-generated strong passwords, requires the following configuration on the visualization hosts running the DCV server:

- The DCV simple external authenticator provided with NICE DCV must be installed and running.

  The simple external authenticator installation package is distributed as an rpm, e.g. nice-dcv-simple-external-authenticator-2017.x...x86_64.rpm.

  Once installed you can manage the service as root user:

  - On systems using SystemD (e.g. RedHat 7):

    ```
    systemctl [start|stop|status] dcvsimpleextauth
    ```

  - On systems using SysVInit (e.g. RedHat 6):

    ```
    /etc/init.d/dcvsimpleextauth [start|stop|status]
    ```

- The DCV server must be configured to use the simple external authenticator dcvsimpleextauth instance running on the same host, e.g. inside /etc/dcv/dcv.conf, under the security section, there should be a setting like this:

  ```
  [security]
  auth-token-verifier="http://localhost:8444"
  ```

- Restart the DCV server after any changes made to /etc/dcv/dcv.conf configuration file.

NICE DCV 2017.0 or later on Windows

For Windows environments the configuration of the authentication to use with NICE DCV must be configured on EnginFrame in the INTERACTIVE_DEFAULT_DCV2_WINDOWS_AUTH configuration parameter inside the $EF_TOP/conf/plugins/interactive/interactive.efconf file.
Default value and documentation can be found in the static configuration file $EF_TOP/<VERSION>/enginframe/plugins/interactive/conf/interactive.efconf.

The auto authentication system, providing seamless authentication with self-generated strong passwords, does not require any other configuration on the visualization hosts running the DCV server.

The DCV server service is managed by the interactive session job landing on the node:

- If the DCV server service is not running, it will be started.

- If the DCV server service is running but with different authentication configuration than the one set on the EnginFrame side, the configuration will be changed and the service restarted. This includes the case when the DCV server is configured to automatically launch the console session at system startup. This setting will be removed by the interactive session job.

- If DCV session is running but there is no logged user, the session will be closed by the interactive session job.

- It is not possible to submit an interactive session to a node with a DCV session running and a user logged in.

Supported Browsers

NICE EnginFrame produces HTML which can be viewed with most popular browsers. NICE EnginFrame has been tested with the browsers listed in Table 2.4, “Supported Browsers”.

Table 2.4. Supported Browsers

Name	Version	Notes
Microsoft® Edge	41 and 44	
Microsoft® Internet Explorer®	10 and 11	
Mozilla Firefox®	3.6 and above	
Apple® Safari®	6.0 and above and iOS 6 version	Tested on Mac® OS X® and iPad® only.
Google™ Chrome™	25 and above	

JavaScript® and Cookies must be enabled on browsers.

Interactive Plugin Requirements

Interactive Plugin requires the following components to be successfully installed and configured:

- at least one supported resource manager software, see the section called “Distributed Resource Managers” or a session broker software, see ???
- at least one supported remote visualization middleware, see the section called “Remote Visualization Technologies”

The Interactive Plugin can be used without a license when running EnginFrame on an Amazon EC2 instance. When running on an on-premises or alternative cloud-based server, a proper license must be installed on the EnginFrame Server.

Each node running interactive sessions should have all the necessary software installed. On Linux® this usually means the packages for the desired desktop environment (gnome, kde, xfce, etc).

In addition, to let the portal show screen thumbnails in the session list, the following software must be installed and available in the system PATH on visualization nodes:

- Linux®: *ImageMagick tool* (`http://www.imagemagick.org`) and the `xorg-x11-apps`, `xorg-x11-utils` packages
- Windows®: *NICE Shot tool* (`niceshot.exe` available under `$EF_TOP/<VERSION>/enginframe/plugins/interactive/tools/niceshot`). Not required on NICE Neutro hosts since Neutro Agent installer already includes it.

Session Managers

Starting from version 2020.0, EnginFrame supports DCV Session Manager as Session Broker.

At installation time you can choose to use DCV Session Manager as session broker and provide the configuration parameters required by EnginFrame to contact the remote DCV Session Manager Server.

For detailed instructions on how to install and configure the DCV Session Manager please refer to its chapter.

Single Application Desktop Requirements (Linux®)

Sometimes you may prefer to run a minimal session on your interactive nodes consisting in a minimal desktop and a single application running. In that case, instead of installing a full desktop environment like GNOME or KDE, you may want to only install some basic required tools, a Window manager, a dock panel and the applications you intend to use.

For this intent the `minimal.xstartup` script can be configured to be a Window Manager choice for the Applications and Views service editors.
Here is a reference list of the tools used by the `minimal.xstartup` file provided by EnginFrame under `$EF_TOP/<VERSION>/enginframe/plugins/interactive/conf`:

- basic tools: `bash`, `grep`, `cat`, `printf`, `gawk`, `xprop`
- window managers: `metacity`, `kwin` (usually provided by package `kdebase`), `xfwm4`
- dock panels: `tint2`, `fluxbox`, `blackbox`, `mwm` (usually provided by package `openmotif` or `lesstif` or `motif`)

Shared File System Requirements

Depending on the deployment strategy, EnginFrame may require some directories to be shared between the cluster and EnginFrame nodes. This guide covers the simplest scenario where both EnginFrame Server and EnginFrame Agent run on the same host. For more complex configurations or to change the mount points of the shared directories, please check the *"Deployment Strategies"* section in the EnginFrame Administrator's Guide.

In this scenario the EnginFrame Server, EnginFrame Agent and visualization nodes may require the `$EF_TOP/sessions` directory to be shared. Please refer to the following table to check if you need to share this directory or not.

Table 2.5. Shared File-System Requirement

Distributed Resource Manager	Linux®	Windows®
NICE Neutro	-	Not required
IBM® Platform™ LSF®	Not required	-
SLURM™	Required	-
Torque	Required	-
Altair® PBS Professional®	Required	-

Distributed Resource Manager	Linux®	Windows®
Grid Engine (SGE, SoGE, OGE, UGE)	Required	-

3

Installation

Prepare for Installation

Before installing and assuming your system matches one of the supported configurations, the following prerequisites have to be met:

1. Select an installation location (called `$NICE_ROOT` hereafter) with at least 200 MB free disk space.

2. Select an EnginFrame administrator user (`efadmin` hereafter). This user must exist prior to the installation. He must be an administrator of the selected distributed resource managers, i.e. must be able to retrieve job information (for all users) and cluster/host information.

3. Select an unprivileged user for running EnginFrame Portal (`efnobody` hereafter). This user must exist prior to the installation.

> **EnginFrame Users**
>
> `efadmin` and `efnobody` must be operating system valid accounts: you must be able to login to the system with those accounts and they *must not* be disabled.

Installing

1. Login on the machine designated to host EnginFrame

2. Change working directory to the EnginFrame installer *`enginframe-2020.0.x.y.jar`* directory location

3. Be sure the umask is 022

4. As `root`, run:

```
$JAVA_HOME/bin/java –jar enginframe-2020.0.x.y.jar
```

5. Answer the installer questions.

Updating

Starting from EnginFrame 2015, the installer supports updating from a previous 2015.x version. The installer does not support upgrading from EnginFrame 2013 or older versions.

To update from an older 2015.x version launch the installer and select the `$NICE_ROOT` containing the `enginframe` folder. The installer will perform some checks on the current installation and then will propose to update.

In case the installer detects any changes in the `$EF_TOP/<VERSION>` directory with respect to the installation status, it will print a warning with information on the detected changes. The user can decide to continue the installation.

The installation process will create a new `$EF_TOP/<VERSION>` directory containing the new EnginFrame files.

Upgrading from pre-2015 versions

Automatic upgrade from pre-2015 versions is not supported.

To install a new EnginFrame and port the old instance configuration to the new one, backup the old installation and follow these suggested steps:

1. If possible, make sure all users close their interactive sessions.

   Running sessions will not be closed by EnginFrame and the users can continue to use them. They will be able to reconnect to existing sessions as long as they have a valid connection file with credentials (*.vnc, etc).

   However the new EnginFrame instance will *NOT* import any of the old sessions and *no action on old sessions will be possible from the portal*. Users will *NOT* be able to connect or close sessions launched from the old EnginFrame instance.

   A session launched from the old EnginFrame instance will need to be closed manually by connecting to it and logging out of the desktop session.

2. Stop the current EnginFrame instance

3. Install a new EnginFrame instance using different installation paths, in particular note you *MUST NOT* select the same location for the Spoolers, Sessions and Repository directories

4. Any change to the configuration of the old EnginFrame instance has to be manually ported to the new installation. Please check the section called “Configuration File Mapping” for reference

5. Start the new EnginFrame instance

> ### Note
>
> Note the new EnginFrame instance will *NOT* use any spooler or session from the old instance. Users will not be able to browse the spoolers of the old installation or retrieve or delete data through the new portal.

Configuration File Mapping

Starting from EnginFrame 2015 the directory structure changed with respect to the previous versions.

In the case changes applied to a pre-2015 installation want to be ported to a new 2020.0 installation use the following reference.

Please note only the *changes* to the default configuration must be applied, *DO NOT* copy the whole files from the previous installation to the new one.

Identify the old installation directory: the default value is `/opt/nice` and contains the `enginframe`, `conf`, `tomcat` sub-directories. We will refer to this directory as `OLD_INSTALL` in the next paragraphs.

Identify the root configuration directory of the new installation: the default value is `/opt/nice/enginframe/conf` and contains `enginframe`, `derby`, `plugins`, `tomcat` subdirectories. We will refer to this directory as `EF_CONF_ROOT` in the next paragraphs.

Main configuration

Changes to the configuration files in `OLD_INSTALL/conf` must be ported to the corresponding files in `EF_CONF_ROOT`. The files which may have been changed in the old instance are:

- `enginframe.conf`

Changes to the configuration files in `OLD_INSTALL/enginframe/conf` must be ported to the corresponding files in `EF_CONF_ROOT/enginframe`. The files which may have been changed in the old instance are:

- `server.conf`
- `agent.conf`
- `log.server.xconf`
- `log.agent.xconf`
- `authorization.xconf` (please check the section called “Changes to authorization.xconf”)

Tomcat® configuration

Changes to the configuration files in `OLD_INSTALL/tomcat/conf` must be ported to the corresponding files in `EF_CONF_ROOT/tomcat/conf`. The files which may have been changed in the old instance are:

- `Catalina/localhost/enginframe.xml`
- `catalina.policy`
- `catalina.properties`
- `context.xml`
- `logging.properties`

- `server.xml`
- `tomcat-users.xml`
- `web.xml`

Changes to the configuration files in `OLD_INSTALL/tomcat/webapps/ROOT` must be ported to the corresponding files in `EF_CONF_ROOT/tomcat/webapps/ROOT`. The files which may have been changed in the old instance are:

- `favicon.ico`
- `index.html`
- `WEB-INF/web-xml`

Apache Derby® configuration

Changes to the configuration files in `OLD_INSTALL/derby/db` must be ported to the corresponding files in `EF_CONF_ROOT/derby`. The files which may have been changed in the old instance are:

- `derby.properties`
- `server.policy`

Plug-in configuration

Changes to the configuration files in `OLD_INSTALL/enginframe/plugin/[plugin-id]/conf` must be ported to the corresponding files in `EF_CONF_ROOT/plugin/[plugin-id]`. The files which may have been changed in the old instance are:

- `log.xconf`
- `authorization.xconf` (please check the section called “Changes to authorization.xconf”)
- `ef.auth.conf` (activedirectory, ldap, os, pam)
- `grid.conf` (grid)
- `dcv.gpu.balancer.conf` (interactive)
- `dcv2.gpu.balancer.conf` (interactive)
- `interactive.efconf` (interactive)
- `nat.conf` (interactive)
- `ef.lsf.conf` (lsf)
- `ef.sge.conf` (sge)
- `ef.pbs.conf` (pbs)

- ef.torque.conf (torque)

Changes to the configuration files in OLD_INSTALL/enginframe/plugin/neutro/conf must be ported to EF_CONF_ROOT/plugin/neutro/neutro.efconf. The files which may have been changed in the old instance are:

- properties.conf

Changes to authorization.xconf

When porting changes from pre-2015 authorization.xconf files note in EnginFrame 2020.0 the ACLs and Actors are merged and, in case different files define the same ACL or Actor ID, the conflicts are resolved by applying the following precedence rule, from the most to the least authoritative:

- EF_CONF_ROOT/enginframe/authorization.xconf
- EF_TOP/<VERSION>/enginframe/conf/authorization.xconf (read-only)
- EF_CONF_ROOT/plugins/[plugin-id]/authorization.xconf
- EF_TOP/<VERSION>/enginframe/plugins/[plugin-id]/conf/authorization.xconf (read-only)

Starting EnginFrame

After a successful installation, you can start EnginFrame as follows:

```
$EF_TOP/bin/enginframe start
```

or, in case EnginFrame was added to the startup scripts:

```
service enginframe start
```

Checking EnginFrame Status

You can check EnginFrame status by executing:

```
$EF_TOP/bin/enginframe status
```

or, in case EnginFrame was added to the startup scripts:

```
service enginframe status
```

Accessing EnginFrame

Open the URL, in your browser, configured during installation with the following syntax:

http://<serverhost>:<serverport>/

The default for `<serverport>` is `8080`.

Important

JavaScript® and Cookies must be enabled on browsers.

Troubleshooting

Useful information can be found inside EnginFrame logs. They are located under `$EF_TOP/logs/<HOSTNAME>`. These files contain *EnginFrame Portal log messages*.

After a successful startup the `ef.log` should print two lines containing the Java and EnginFrame versions:

```
2017/Mar/14 09:51.42 INFO [...] : Java Version (Oracle Corporation 1.8.0_101)
2017/Mar/14 09:51.42 INFO [...] : EnginFrame Server STARTED (version: 2017.0.41391)
```

Since EnginFrame Portal runs inside the Tomcat® web server (bundled with installer), also Tomcat® log files can help. Tomcat® web server log messages are located under `$EF_TOP/logs/<HOSTNAME>/tomcat`.

By default EnginFrame Portal installs and uses a Apache Derby® DB instance. Derby log files are located under `$EF_TOP/logs/<HOSTNAME>/derby`.

If any of these log files contain error messages that you do not understand, feel free to contact helpdesk@nice-software.com or your EnginFrame reseller.