System Requirements

Version **2020.1**

NICE
an aws company

Copyright

Copyright © 2000-2021, NICE s.r.l.

All right reserved.

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

This document is protected by copyright and you may not redistribute or translate it into another language, in part or in whole, without the express written permission of NICE s.r.l.

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.

Other names mentioned in this document may be trademarks of their respective owners.

Last Update
September 21, 2021 (rev. 762)

Latest Version
https://www.enginframe.com

Contents

1

EnginFrame System Requirements

This documents lists the hardware and software prerequisites for installing EnginFrame.

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.

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, 8.x (*x86-64*)
- SUSE® Linux® Enterprise Server 11 SP2, 12 SP3 (*x86-64*)

  SUSE® Linux® Enterprise Server 12 SP5 (*x86-64*)

  SUSE® Linux® Enterprise Server 15 SP2 (*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`).

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

Notes on Suse Linux

EnginFrame PAM standard user authentication (system) expects to find the file `system-auth` in the folder `/etc/pam.d/`, however in SUSE® Linux® Enterprise Server this file is called `common-auth`. So to make the standard authentication work a symbolic link is required:

```
ln -s /etc/pam.d/common-auth /etc/pam.d/system-auth
```

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.

EnginFrame supports MySQL® Database 8.0.x and later with the InnoDB storage engine. EnginFrame has been used widely with other databases (Oracle® Database, SQL Server®, MariaDB®) even though they are not officially supported. In case of issues with a supported 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.

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

Authentication Mechanisms

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

Refer to Table 1.1, "Supported Authentication Mechanisms" to select the most appropriate authentication method for your system and check its third-party software prerequisites (if any).

Table 1.1. 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 Active Directory	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.
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 1.2, "Supported Distributed Resource Managers" for a list of supported DRMs.

Table 1.2. Supported Distributed Resource Managers

Name	Version	Notes
IBM® Platform™ LSF®	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®: 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.
NICE Neutro	2013 or later (Will be discontinued)	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)	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. 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	
AWS Batch	The AWS Batch cluster must be created with AWS ParallelCluster 2.x (2.1.0 or later)	The installer will ask you to specify the AWS ParallelCluster cluster name and the AWS region.

Some schedulers like 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”.

Support for NEUTRO will be discontinued on 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.

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 1.3. Supported Remote Visualization Technologies

Name	Version	Notes
TurboVNC	2.2 (Will be discontinued)	Linux® only (server side).
NICE DCV	2017.x or later	It allows to share sessions both in full access or view only mode.
VirtualGL	2.5 or later	

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

Starting from March 2022 we will discontinue support for TurboVNC in favor of DCV Session Manager. Customers using versions of EnginFrame released after March 2022 will be unable to use TurboVNC for accessing interactive sessions. Support for NEUTRO will be discontinued on 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.

Network Requirements

EnginFrame is a distributed system. Your network and firewall configuration must allow EnginFrame components to communicate with each other and with user's browsers.

The specific requirements depend on how EnginFrame is deployed on your system. Please refer to *EnginFrame Administrator's Guide* for more detailed information. The following table summarizes network requirements for a basic EnginFrame deployment.

Table 1.4. Network Requirements

Port (Default)	Protocol	From Host	To Host	Mandatory
8080/8443	HTTP/HTTPS	User's clients	EnginFrame Server	Mandatory
9999 and 9998	RMI (TCP)	EnginFrame Server	EnginFrame Agent	Optional [1]
8080/8443	HTTP/HTTPS	EnginFrame Agent	EnginFrame Server	Optional[1]
7800	TCP	EnginFrame Server	EnginFrame Server	Mandatory only for EnginFrame Enterprise[2]

[1]Required if EnginFrame Agent and EnginFrame Server run on separate hosts

[2]EnginFrame Servers use the port to communicate with each other

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 1.5, “Supported Browsers”.

Table 1.5. Supported Browsers

Name	Version	Notes
Microsoft® Edge	41 and 44	
Microsoft® Internet Explorer®	10 and 11 (Will be discontinued)	
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.

By the end of December 2021 we will discontinue support for Internet Explorer 10. Support for Internet Explorer 11 will be discontinued by the end of June 2022.

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 manager, see the section called “Session Managers”
- 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 1.6. Shared File-System Requirement

Distributed Resource Manager	Linux®	Windows®
NICE Neutro	-	Not required
IBM® Platform™ LSF®	Not required	-
SLURM™	Required	-
Altair® PBS Professional®	Required	-
Grid Engine (SGE, SoGE, OGE, UGE)	Required	-