POLARION

Polarion® Application Lifecycle Management Platform

Installation Guide for LINUX

Version: **2016-SR1**

Polarion Application Lifecycle Management Platform Installation Guide for Linux

Updated: 13-June-2016

Table of Contents

About This Guide

Welcome and thanks for using Polarion application lifecycle management solutions.
This guide covers installation information and procedures for creating a **production installation** for all Polarion Software products based on the Polarion Application Lifecycle Management Platform. The list of products covered by this guide currently includes:

- **Polarion® ALM™**
- **Polarion® REQUIREMENTS™**
- **Polarion® QA™**

In general, the information is applicable to all the above products. Any product-specific differences will be explicitly noted. The information covers both new installations and, where applicable, updating of existing installations.

This guide applies to installation of the above Polarion products on supported Linux operating systems.

If you want to install a product on a Windows system, please see the separate *Installation Guide for Windows* document provided in the download archives for our Windows product versions, and also available on the Polarion Software web site.

Evaluation Installations

If you are installing Polarion for evaluation purposes, we recommend the ***Polarion Trial Guide*** (PDF), available for download on any product download page at www.polarion.com. This guide focuses on getting you up and running with an evaluation installation as quickly as possible, using the 1-click install option of the Windows installer. (This is recommended for the initial stage of any evaluation.)

Large-scale Installations

If you need a large-scale server environment with multiple clustered servers and failover capabilities, multiple repositories, etc. you should get the *Polarion Enterprise Setup Guide* (PDF), also available on product download pages on the Polarion Software web site.

1. System Requirements and Recommendations

Server Software

Requirement	Description
Operating System	SUSE Linux Enterprise Server 11 (SP 3 or higher) or 12, **or** Red Hat Enterprise Linux 6.x or 7.x (7.2 or higher)/CentOS 6.x or 7.x (7.2 or higher), **or** Debian GNU/Linux 7.0 or 8.0, **or** ▯Ubuntu Server 12.04 LTS or 14.04 LTS. Supported architectures are x86, x86_64 or amd64. URW fonts must be installed in the operating system. For SUSE, use the *ghostscript-fonts-std* package from http://tinyurl.com/pjt3vg3.
Java Runtime Environment	Java Platform, Standard Edition Development Kit 8, or Oracle JDK 8, 64-bit. (http://www.oracle.com/technetwork/java/javase/downloads/index.html) Note that OpenJDK 8 is not the same as Oracle Java 8 and is not officially supported. `JAVA_HOME/bin` should exist in the system paths.
Version control system	Subversion version 1.6.x or 1.7.x or 1.8.x or 1.9.x: http://subversion.apache.org/. If you are compiling Subversion yourself, compile using the *`--with-apsx`* or the *`--with-httpd`* option.
Web Server	Apache HTTPD server with mod_proxy_ajp and Subversion extension (WebDAV+SVN apache modules): http://httpd.apache.org/ In general, the Polarion server should run with whatever Apache version is present on a Windows system provided it is at least the minimum required version (2.2), and mod_proxy_ajp and Subversion extension modules are also installed.
Database	PostgreSQL version 8.4 or higher.

Server Hardware

Requirement	Description
RAM	● Minimum: 4 gigabytes for production installation. (This is maximum a 32-bit OS will run.) ● Recommended: 8 gigabytes or more and a 64-bit Operating System
Disk Storage Space	● Minimum: 10 GB (gigabytes) ● Recommended: 40 GB (gigabytes) or more There is no hard and fast rule for disk storage space. The actual amount you will require depends on the number and size of projects managed with Polarion. The more projects, and the larger they are, the more disk storage you will require.

Client Software

Requirement	Description
Operating System	Any operating system that can run the supported web browsers with support for the Flash plugin (see below). If the client user will use a Polarion product supporting data interchange with Microsoft Office®, the client user must run a Windows operating system compatible with a supported version of the Microsoft Office application(s) used. For details, please see *Appendix: Supported Microsoft Office Versions*.
Web Browser	All you need to use Polarion is a web browser. The most current list of supported browsers and versions is provided in the Release Notes section of the `README.html` file, delivered in all download distributions of all Polarion products.
Adobe Flash	The Polarion web portal displays charts and other graphical data about Polarion-managed projects using Adobe Flash. To view these properly using a web browser, the client computer must have Adobe Flash Player installed. You can download it free at http://www.adobe.com/products/flashplayer/.

Client Hardware

Requirement	Description
RAM	Minimum: 2 GB (4 GB recommended)
Display Resolution	Minimum: 1280 x 800 pixels
Server Connection	Not less than 1 Mbit/S

Additional Recommendations

The items described in this section are not critical for running and evaluating Polarion. However, the components described are needed to be able to fully utilize the product's capabilities.

Libraries required for building the demo projects

The distribution contains several demo projects. Each of them needs its particular set of 3rd party libraries to be correctly built and have the project reports generated. Any missing libraries are automatically downloaded from the Internet during project processing, so you may need a connection to the internet when you first try building the demo projects or running reports for them.

Library required for exporting *Highcharts* charts

Applicable only for 32-bit Linux: The Phantom JS library is used for preview and PDF export of *Highcharts* charts. Only the 64-bit version of the library is bundled with Polarion's Linux distributions. If you run Polarion on 32-bit Linux, you must obtain and install the 32-bit version of the library. The procedure is documented in online Help. In the Administrator's Guide chapter *Configuring PDF Export*, see the section *Configuring Highcharts Export*, and in particular the sub-section *Installing 32-bit Phantom.js*.

The *urw-fonts* package must be installed on CentOS and SuSE. See the same Help topic for information.

Enable Email Notifications

The Polarion server can send email notifications in response to various events in the system such as build completions, new work items, etc. It can also notify users about external changes.

To enable Polarion to send out email notifications you need to:

1. You can do this before starting Polarion server by setting the host name in the announcer.smtp.host property in the *polarion.properties* file located in `$POLARION_HOME$/etc/`. There you should also set the *announcer.smtp.user* and *announcer.smtp.password* properties to a valid email account on the SMTP host specified in *announcer.smtp.host*. If your SMTP server doesn't listen on default port 25, change the port setting in *announcer.smtp.port*. You may wish to create a special account on your SMTP host for use with Polarion notifications.

2. Provide a valid email address for each user in their user account. (Administration > User Management > Users.) This can be automated through user self-creation of accounts, or integration with LDAP. See Help, Administrator's Guide: *Managing Users and Permissions*.

When this configuration is correctly set up, the system will send notification e-mails about various events according to the notification targets configuration. For information on configuring email notifications, see Help, Administrator's Guide: *Configuring Notifications.*

> **NOTE**
>
> *If a Work Item is modified outside of the Polarion portal, (e.g. manually in SVN), email notifications are sent the same as they would be if the modification has been done in the portal.*

Enable support for Javadoc

The demo projects (and your projects) can be configured to provide Javadoc reports. Javadoc must also be enabled for the descriptors.xml file. Access it in the Repository perspective: `Repository/.polarion/reports/descriptors.xml`. Refer to comments in the file for enabling Javadoc.

2. Linux Installation and Overview

This chapter covers installation and startup procedures for Linux. There is also some information about initial configuration of Polarion on Linux, as well as reference information about the Polarion installation structure and third-party components.

Installation Types

- **Automated installation** means that the entire installation process is driven automatically by an installation script. At most you have to answer some questions by typing, but you do not pre-install anything or do any other manual work in order to satisfy the prerequisites and the success of the installation process before you start the script. Automated installation is possible on most, but not all, of the supported Linux platforms.

- **Manual installation** means you must pre-install required infrastructure software before you attempt to install the Polarion platform. You follow steps in documentation as opposed to interactive instructions from a script. You may also need to write some shell scripts by hand or modify the `manual_install.sh` file before running it.

- **Semi-automated installation** installation is a third type in which the automated installation process script is not able to install all the prerequisites for some platform automatically, so it requests you to install something manually before proceeding.

For installation types and the platforms for which they are applicable, see *Supported Platforms and Installation Types* in the next section.

Automated Installation

The automated installer scripts detect the operating system. The OS version is not checked, however, so you need to ensure that you have the minimal required version of your system as specified in Server Software Requirements.

If you have different version of the OS from the supported distributions, it is probably still possible to perform an automatic installation. However, during the process you will need to manually install required third-party software before you respond no to installation default dependencies.

If your OS is supported, the automated scripts will install the prerequisite third-party software for the specific supported Linux platform from existing package repositories on your system during the Polarion installation process, after which they will then install the Polarion platform itself. Please ensure that you have set up package repositories defaults provided for your system.

Supported Linux versions for automated installation

If you have one of the supported Linux platforms, you will find installation on Linux easier, faster and smoother. Each supported platform comes with predefined configurations of the recommended version of third party software. If the installer script finds that your operating system is not one of the supported variants, it will inform you and allow you to install Polarion manually.

The table below lists the Linux platforms that are fully supported as of this writing. Installation support for other platforms may become available. Be sure to check the list of available installers on the web page where you download Polarion.

We highly recommend using stable repositories and installing tested versions of third party software rather than installing newer, non-tested versions.

Supported Platforms and Installation Types

Platform	Type	Default repositories necessary
SUSE	Automated	All default repositories including SDK.
RedHat	Semi-automated	Default repositories from installation, more information about mirrors and repositories can be found at □http://www.centos.org/download/mirrors/
Ubuntu	Automated	Main, Universe and Multiverse. For more information see: https://help.ubuntu.com/community/Repositories/Ubuntu
Debian	Automated	main, updates
All other	Manual	N/A

Automated Installation Process

The automated installation scripts perform the following installation steps:

1. Create a special system account for Polarion.
2. Set valid permissions for Polarion.
3. Add Polarion as a system service at default runlevels (according to the Linux platform and ensure that it starts after Apache).
4. Set up the default Apache configuration for Polarion and Subversion used by Polarion.
5. For a new/clean installation, create a new Subversion repository with initial or sample demo data.
6. Create symbolic links into default system places such as `/var/logs`, `/var/run/`, `/srv/polarion` and so on.
7. Start the Polarion and third-party servers.

Automated installation steps are detailed in chapter 3: *Automated Installation Procedure*.

Manual Installation

A manual approach (i.e. no automated installation script) installs only the Polarion platform without any third party software (SVN, Apache, etc.) Therefore manual installation requires that you obtain all the required third-party software needed to support Polarion, install and configure it, and then install and start the Polarion server. Manual installation may be necessary in some cases: when the computer does not yet have an internet connection, for example. For more information see Appendix: *Manual Installation on Linux*.

3. Automated Installation Procedure

This section describes the basic procedures for performing a Polarion installation using an automated installation script on one of the platforms for which one is available (see Supported Platforms and Installation Types).

> **IMPORTANT**
>
> *If you already have one or more of the required third-party software systems installed, you can respond NO to the prompts asking if you want to install them, BUT…*
>
> *BE ABSOLUTELY SURE YOU DO HAVE IT INSTALLED, otherwise you can break the automated process and it can be problematic to resume.*
>
> *If you are* not *ABSOLUTELY SURE you have the required installation and configuration, it is better to respond YES to such prompts.*

To perform automated installation:

1. Unpack the archive you download from the Polarion Software web site into an empty directory.

2. If SELinux is bundled with/installed on your Linux OS, make sure it is not activated. Check the status of SELinux with command **`□/usr/sbin/sestatus -v`**.

3. Log in to the root account.

4. Navigate into the unpacked directory.

5. Run **`chmod +x install.sh;`** (after unzip, necessary only 1 time)

6. Run the installation script: **`./install.sh`**

7. Continue with the installation, answering questions when prompted by the script.

8. After the installation process the Polarion server will start automatically, which you can verify by opening **http://localhost/polarion** in a supported web browser.

> **IMPORTANT**
>
> *Before using the system in production, see the section in Chapter 6: “Securing the Polarion Activation Application”.*

4. System Startup and Shutdown

Polarion includes an activation application that makes it possible to install or update a license during runtime of Polarion server. Access to this application is NOT protected by any username or password. For production use it is highly recommended to secure access to this application. See Securing the *Polarion Activation Application* in chapter 5.

Starting Polarion

The first time you start the Polarion server, no time estimates appear in the console because there is no data on which to base the estimate. Instead, Not enough data for startup estimation is written to a log file at special log `/var/log/polarion.log`.

First startup after an automated installation is automatic. This section provides the procedure for subsequent startups, which includes first startup after manual installations.

To start the Polarion server, execute whichever of this following is applicable:

- Common command: **`$POLARION_HOME$/bin/polarion.init start`** or system supporting systemd: **`systemctl start polarion`**
- On CentOS or RHEL or Debian: **`service polarion start`**
- On SUSE: **`rcpolarion start`**

On startup and on re-index operations, Polarion estimates and reports the amount of time the operation will take. You will see this estimate in the console and log file on subsequent startups, but the first time, no data exists on which to base the estimate. The following startup phases are reported in the console and log file:

- Platform startup
- Context recognition
- Context initialization
- Revisions processing
- Build artifacts recognition
- BIR inspection
- Data indexing
- Polarion startup

Shutting Down Polarion

To shut down Polarion execute whichever of this following is applicable:

- Common command: **`$POLARION_HOME$/bin/polarion.init stop`** or system supporting systemd: **`systemctl stop polarion`**
- On CentOS or RHEL or Debian: **`service polarion stop`**
- On SUSE: **`rcpolarion stop`**

Starting and Stopping the PostgreSQL Database

The automated installation process creates a service for the PostgreSQL database named *postgresql-polarion*. If you manually start/restart your Polarion system, this service must be started after Apache and before the Polarion server. You can use one of the following commands to start the service:

- CentOS/RHEL or Debian: **`service postgresql-polarion start`**
- SUSE/SLES 11: **`rcpostgresql-polarion start`**
- SUSE/SLES 12: **`service postgresql-polarion start`**

You can stop or restart the service by replacing **start** with **stop** or **restart** in the above commands.

If you install Polarion ***manually***, the *postgresql-polarion* service is not present, so it is necessary use utilities provided by PostgreSQL to start/stop the database. Starting/stopping of the PostgreSQL server is common for all Linux distributions and needs to be done on behalf of the *postgres* user. You need to initialize the database storage prior to running the server for the first time (see Configuration of Third-party Components: PostgreSQL). All PostgreSQL utilities must be run on behalf of the *postgres* system user. Command syntax:

```
su postgres
pg_ctl -D <PATH_TO_PG_DATA> -l <PG_LOGFILE> start/stop/restart
```

In a typical installation this would translate to:

```
su postgres
pg_ctl -D /opt/polarion/data/postgres-data -l
/opt/polarion/data/postgres-data/log.out -o "-p 5433" start
```

(The *pg ctl* command should be written to a single command line.)

You can define additional options, for example on which port the database should run. For an overview of options, see http://www.postgresql.org/docs/9.4/static/app-pg-ctl.html.

Integration with *systemd*

On Linux systems that support *systemd*, Polarion installs support for it and uses it for system startup. Integration with *systemd* ensures that other necessary software (Apache and PostgreSQL, for example) is started before Polarion.

With systemd, commands `start`, `stop`, and `restart` can be used. (These are are also delegated to `polarion.init.`) Other commands like `reindex`, `demo` etc. must be used with /`opt/polarion/bin/polarion.init reindex` etc. The `start/stop/restart` commands can also be used with `polarion.init in` systems supporting *systemd*.

Integration with init system

The script `/opt/polarion/bin/polarion.init` is suitable for integration with the init system to start the Polarion server if systemd is not used. It supports start, stop and restart commands using the special system account *polarion*.

IMPORTANT

It is important to note that `polarion.init` *does not start the Apache and PostgreSQL servers. You should always make sure to start these by other means before starting Polarion server.*

5. After Installation

This chapter covers some first things you will probably want to do immediately following installation and initial configuration (if applicable) of Polarion.

Configuring PostgreSQL Database

This component must be properly configured before you can start the Polarion server. Automated installation scripts attempt to perform basic configuration so that Polarion can run. If the script was unable to configure PostgreSQL you will need to configure it manually. See *Appendix: Manual Configuration of Third-party Components: PostgreSQL*. Even if the script succeeds, some additional configuration of the PostgreSQL database is recommended for optimal performance.

Optimizing the PostgreSQL Database

Beginning with version 2015 SR2, Polarion integrates the PostgreSQL database in all new installations. After a new Polarion installation containing this database, it is highly recommended that the administrator adjust some PostgreSQL settings to optimize performance. You should make the following changes in `postgresql.conf`. The database server needs to be restarted following this configuration.

Default Linux path: `/opt/polarion/data/postgres-data/`

```
max_connections = 80  # should be < 10 * number of CPUs
shared_buffers = 2GB  # should be 10% - 15% of total system RAM
work_mem = 10MB       # should be 10MB - 100MB
maintenance_work_mem = 200MB

fsync = off
synchronous_commit = off
full_page_writes = off

wal_buffers = 256kB    # should be more than size of common
                      # transaction

checkpoint_segments = 32
effective_cache_size = 4GB        # should be approx 1/3 of total
                                  # system RAM
max_locks_per_transaction = 100  # specific for Polarion
# Optimal planner performance setting
# For HDD, keep default setting. Otherwise, uncomment the
# applicable setting below:
# For SSD:
# random_page_cost = 1.5
# For SAN:
# random_page_cost = 2.0
```

Note that on some Linux environments, the automatic configuration of PostgreSQL's *shared_buffers* parameter may calculate a value that exceeds the available shared memory configured by kernel parameter *SHMMAX*. In automated installations, the installer script assigns a default value of 24 MB for *shared_buffers* and posts this message:

> *"NOTICE: Polarion attempted to change value of shared_buffers in postgresql.conf to $v_shared_buffers_old but $v_shared_buffers_new was used instead because of low value in /proc/sys/kernel/shmmax."*

If you see this message during the installation, you are advised to review your SHMMAX configuration and adjust *shared_buffers* for PostgreSQL manually.

Securing the Polarion Activation Application

Beginning with version 2015, Polarion includes an activation application that makes it possible to install or update a license during runtime of the Polarion server, without need to copy the license file manually to the target machine. **Access to this application is NOT initially protected by username and password.** For production use it is highly recommended to secure access to this application directly in the Apache configuration.

Beginning with version 2015, there is a template Apache configuration file in the Polarion installation folder:
`/polarion/polarion/install/polarion.activation.conf.template`

To ensure that a user name and password is requested when accessing the activation application (`/polarion/activate/online` and `/polarion/activate/offline`), copy this file to the Apache configuration folder, on Linux usually `/etc/httpd/conf.d/`.

After copying the file, rename it to remove the extension ".template". Then open the file in any text editor and modify it according to the instruction comments provided.

The template configuration is prepared for both users file authentication (like Polarion uses for Subversion by default, with user passwords data in a file) and for authentication against an LDAP server.

In a multi-instance setup (coordinator + one or more instances, that can be clustered), it is necessary to use this configuration only on the coordinator server (the activation application runs only on the coordinator). For additional information about this type of setup, see the Polarion Enterprise Setup Guide.

Changing Default System Passwords

To help ensure the security of your Polarion system, you should change the default password of the system administrator user account, and polarion user in Subversion. The following two sections explain how to make these changes.

Changing the Default System Administrator Password

The default Polarion user account System Administrator has access to all administrative functions of Polarion, including read-write access to the Subversion repository. After installing Polarion for actual production use, you should change the password of this account.

To change the default administrator password:

1. Log in to the Polarion portal with the default login described above.

2. Click on **My Polarion**. The *My Polarion* page for the System Administrator account loads in the content area.

3. In the System Administrator's *My Polarion* page, click the link: **My Account**.

4. In the *My Account* page click the **Edit** button.

5. Enter the new password in the *New Password* field, and again in the *Re-enter Password* field.

6. If you want to continue using this account as the main system administrator account, you may wish to add your email address in the *Email* field, and add a description for the account in the *Description* field.

7. When finished editing the System Administrator profile, click the **Save** button. The password is now changed and you will need to use it next time you log in.

DON'T LOSE THE NEW PASSWORD
If you lose the changed password, you will not be able to log in as system administrator. If no other accounts exist with administrator rights, it will not be possible to change the configuration, add projects, manage user accounts, etc.

Changing the password for SVN user 'polarion'

A Subversion repository user named *polarion* is created by default when you install Polarion. This user acts on behalf of the Polarion application and consequently has extensive permissions including read permission for all projects. Access to this user by unauthorized people would compromise the security of your Polarion system, so it is advisable to change this password before putting the system into production use.

The following steps assume you use `passwd` file authentication, which is the most common method.

1. Stop the Polarion server before changing this password.

2. Use the *htpasswd* utility to change the password for the user *polarion*. The utility is installed on your system together with the Apache server binaries.

   Utility syntax: `htpasswd path/passwdfilename username`

   Example: `htpasswd /opt/polarion/data/svn/passwd polarion`

3. Next, change the value of the password property in the `polarion.properties` file to the password you set with the htpasswd utility. Typical location of this file is `/opt/polarion/etc/polarion.properties`.

Note for LDAP Users

The typical setup for most Polarion users is *passwd* file authentication for the *polarion* user with failover to LDAP for company users (this is also the default Polarion setup). For such setups, you do not need to enter the *polarion* user to your LDAP users.

Configuring OLE Object Support

It is possible to import Microsoft Word documents that contain OLE objects. Polarion can display OLE Object thumbnails during Word document import, including EMF thumbnails for Visio diagrams. OLE Objects in documents must contain their thumbnails in.emf or.wmf file formats, and the image converter used must support their conversion into JPEG. OLE Objects themselves are not imported, only their thumbnails.

Some additional third-party software must be installed, and Polarion's system configuration modified before you can import such Word documents.

Prerequisites

- A computer or virtual machine running Microsoft Windows, accessible as a network share to the Linux machine hosting Polarion.
- Winexe for Linux: http://www.aldeid.com/wiki/Winexe
- ImageMagick image converter: http://www.imagemagick.org/script/binary-releases.php

Installation and Configuration

1. Install Winexe on your Linux machine. You must have administrator permissions on the target machine. For more information, see:□http://serverfault.com/questions/72082/winexe-gives-error-error-failed-to-install-service-winexesvc-nt-status-acces

2. Install ImageMagick as a standard program on the Windows computer or VM.

3. Make a Windows share accessible from Linux. The *polarion* user must have write access to this share.

4. Create a shell script on Linux named oleconvert.sh, per the following code example. Replace blue text with correct values for your system. Use forward slashes to delimit Windows folders. Long lines have a blank line before and after them.

   ```
   #!/bin/sh
   infile=$1
   inname='basename "$1"'
   outfile=$2
   outname='basename "$2"'
   linux_path="LINUX_PATH_TO_WINDOWS_SHARE"
   win_path="WINDOWS_PATH_TO_WINDOWS_SHARE"
   cp "$infile" "$linux_path/$inname"

   PATH_TO_WINEXE_EXECUTABLE -U DOMAIN/WINDOWS_USER%WINDOWS_PASSWORD
   //WINDOWS_HOSTNAME "\"WINDOWS_IMAGEMAGICK_INSTALLATION_FOLDER/

   convert.exe\" -flatten \"$win_path/$inname\" \"$win_path/$outname\""
   rm "$linux_path/$inname"
   mv "$linux_path/$outname" "$outfile"
   ```

5. In the system configuration file polarion.properties, configure the OLE converter using the following example. Again, replace blue text with correct values for your system.

   ```
   com.polarion.oleconverter.usefiles=true
   com.polarion.oleconverter.app=PATH_TO_OLECONVERT_SH
   com.polarion.oleconverter.param1=$in
   com.polarion.oleconverter.param2=$out
   com.polarion.oleconverter.convertedImageFormat=png
   ```

6. Restart the Polarion server.

If OLE thumbnails do not appear in imported documents, check the Polarion error log files to make sure there are no invalid path or other errors.

Configuring Polarion for OLE

There are some system properties in the _common.properties system configuration file that you will need to review and set after installing the image converter. Refer to comments in the section for the *com.polarion.oleconverter.app* property that explain the settings.

There is an additional system property in the polarion.properties file that specifies the target image format for the OLE thumbnail conversion. Set the property *com.polarion.oleconverter.convertedImageFormat=png* to have thumbnails converted to PNG, or *com.polarion.oleconverter.convertedImageFormat=jpg* to have thumbnails converted to JPEG format.
The system properties are located on path [POLARION]/etc/polarion.properties.

Licensing and Activation

In order to use Polarion you must obtain a license. A license with the necessary key and file is normally delivered by email to the address provided by the person who purchased the license. If you need help obtaining a license please contact sales@polarion.com.
Polarion installs with a 30-day evaluation license. After obtaining a license for production use, you must activate your Polarion installation. The login page provides a action options leading to online and offline activation instructions. You will need the information provided by Polarion Software to complete the activation.

Using different license types

Several different license types are available - Evaluation, Site, User-limited, etc. If you begin using Polarion with one type of license key (Evaluation, for example), and want to continue using it with a different license type, simply remove the current license key file from the license folder and copy the new license key file there. If Polarion server is running, you will need to restart it for the new license to take effect. You may install multiple license keys for different license types and/or Polarion products on the same server.

Assigning named and concurrent users

If your license provides for named and/or concurrent users, you will need to add assignments for each type of user in the appropriate section of the users file. By default this file is located in the license folder of your Polarion installation. If you change the location for license key files, be sure to move the `users` file to the same folder that stores your license key file.

You can edit it in the *License* topic in global Administration in the portal. The file contains comments with complete instructions on how to find the user IDs of your named/concurrent users, and make the relevant assignments. Be sure that write permission is set for the folder `/opt/polarion/polarion/license`.

License usage log file

Administrators and managers can monitor license usage by checking the license usage log file `log4j-licensing-TIMESTAMP.log`. This file is located in the directory `/opt/polarion/data/workspace/.metadata` by default. (The `/opt/polarion/data` location is configurable during installation)
When a concurrent user logs in/out, a license usage statistics report is written to the licensing log. (Note that concurrent licensing is not supported for all products.)

The following example shows one user currently using an enterprise concurrent license type, the greatest number of users of this license during the current server session (peak), and the maximum number of users allowed by the license (*limit*).

```
2008-05-14 11:12:29,609 [TP-Processor2] INFO  PolarionLicensing
- STATS:enterpriseConcurrentUsers,current:1,peak:2,limit:20
```

Subversion Optimization

Polarion uses a Subversion (SVN) repository as its main data storage. There are 2 topics in the Administrator's Guide component of Polarion online Help that provide guidance for administrators about optimizing Subversion for best performance. It is recommended that you review them before going into production with a new or updated installation.

- Administrator's Guide: Advanced Administration: Topics: Optimizing Subversion
- Administrator's Guide: System Maintenance: Topics: Maintaining Subversion (SVN)

Multiple Repository Setup

There are two Polarion features that enable you to work with multiple repositories, but they are fundamentally different. You need to understand the basics of each feature before deciding which approach to multiple repositories best meets your needs.

The **External Repository** feature gives you the ability to link Polarion artifacts stored in Polarion’s integrated repository with changes (revisions) in source code stored on one or more external SVN or Git repositories.
After installation, you can configure Polarion to use one or more external repositories in addition to the SVN repository bundled and installed with Polarion. For information on this feature, see Help, Administrator’s Guide: *Configuring Repositories*.

The **Clustering** feature enables you to run Polarion on multiple servers, either physical, virtual, or a combination. The topography can be set up to host multiple Polarion servers running on separate machines each with its own Polarion repository, and/or multiple machines all accessing a single Polarion repository. (Polarion servers on any node can optionally be configured to access external repositories, as described above.)

Special installation and configuration procedures beyond the scope of this guide are required to set up a clustered multi-server environment. These are covered fully in the *Polarion Enterprise Setup Guide* (PDF), available for download on all product download pages on the Polarion Software web site.

Accessing the Polarion Portal

Open a supported web browser and enter the following URL:
`http://localhost/polarion/`

On your first login after installation, you can login with the default system administrator credentials:

- User ID: **admin**
- Password: **admin**

LDAP Authorization

In a new installation, users are authorized using the Subversion integrated policy access functions (directives *AuthzSVNAccessFile* and *AuthUserFile* in the `polarionSVN.conf` file). If you have LDAP infrastructure, you can make Polarion authorize users against the LDAP database.

Information on performing this configuration, together with some examples, is provided in the `polarionSVN.conf` configuration file. The file is located on one of the following paths, depending on your Linux distribution:

- `/etc/apache2/conf.d`
- `/etc/httpd/conf.d`

After modifying the configuration file, the Apache server must be restarted to reflect the changes.

For more information about the Apache LDAP modules and their capabilities, visit these web pages: http://httpd.apache.org/docs/2.4/mod/mod_authnz_ldap.html
and http://httpd.apache.org/docs/2.4/mod/mod_ldap.html.

You can find information on configuring Polarion to work with LDAP in the Polarion Help topic Administrator’s Guide: User Management: *Integrating Polarion ALM Server with LDAP/Active Directory*.

Next steps after installation

Once you have installed Polarion and logged in to the Portal, consider taking a look at the demo projects (assuming you installed demo data). Click on the drop-down control in the Navigation panel on the left, select *Open Project or Project Group*, and open any project in the *Demo Projects* group. The information in the Getting Started with Projects topic of the User Guide will help you get going.

You may want to do some initial global customizations such as custom Work Item types, Workflow, Reports, SSL Support, etc. You will find topics on these configurations in the Administrator's Guide provided in the online Help system, and also in the Polarion PDF manual set (available as a separate download at the Polarion Software download site).

Once you have your Polarion system running, and any global customizations done, you are ready to begin setting up your own projects and user accounts. Look up the following topics in the Administrator's Guide: *Creating and Managing Projects* and *Managing Users and Permissions*.

Removing Polarion

An automated uninstaller script uninstall.sh is provided for Linux platforms.

Remove the `/var/log/polarion` folder and `/var/run/polarion.pid`. Polarion should then be uninstalled.

WARNING
Polarion's Subversion repository is stored in the `/opt/polarion` *folder. Be sure this repository does not contain production data that must be preserved! If it does, be sure to make a backup before uninstalling Polarion.*

6. Technical Support

Polarion is problem-free for most people... at least that's been our experience. However, it's impossible to anticipate all the conditions and environments where Polarion may be used.

For those cases when issues arise, Polarion Software's Technical Support team maintains the Customer Self-service Portal which includes an extensive Knowledge Base of common problems and solutions and troubleshooting info, as well as the possibility to submit, manage, and review your own specific support cases.

For information about the portal and technical support options, please visit:
http://www.polarion.com/techsupport/

You can access the Customer Service Portal from the above page, or directly at:
http://support.polarion.com/ssp/

Login credentials for the portal are provided to the Technical Contact person for your company upon purchase of a support/maintenance subscription.

APPENDIX

Default Users and Groups

OS	User	Group	Passwd utility
RedHat (CentOS)	apache	apache	htpasswd
Debian	www-data	www-data	htpasswd
Ubuntu Server	www-data	www-data	htpasswd
SUSE	wwwrun	www	htpasswd2

Enabling Email Notifications

If you did not configure email notification settings in the installation program, you can do this after installation. by setting the host name in the *announcer.smtp.host* property in the `polarion.properties` file located in `$POLARION_HOME$/polarion/configuration`. There you should also set the *announcer.smtp.user* and *announcer.smtp.password* properties to a valid e-mail account on the SMTP host specified in announcer.smtp.host. You may want to create a dedicated email address on your SMTP host for use by the Polarion notifications system.

When this configuration is correctly set up, the system will send notification e-mails about various events according to the notification targets configuration. For information on configuring email notifications, see Help, Administrator's Guide: *Configuring Notifications*.

Manual Installation on Linux

It is highly recommended that you run Polarion on one of the supported Linux operating systems and use the automated installation scripts provided in the product distributions for Linux.

There may be cases when Polarion must be installed manually. For example, perhaps an automated installation script doesn't recognize your system as one of the supported Linux systems due to customization or other factors. Or possibly your server does does not yet have an internet connection, so various packages downloaded during automated installation are not accessible, so automated installation cannot proceed. (Scripts will inform if automated installation is not possible).

The remainder of this section covers full manual installation on Linux systems.

1. System Requirements and Recommendations

Before starting with manual installation, please consult the preceding sections in this manual, listed below:

- Server software requirements (OS, Java, VCS, web server).
- Server hardware requirements (RAM, disk storage).
- Client software requirements (OS, web browsers, Adobe Flash).
- Client hardware requirements (RAM, display, connection).

2. Required Third-party Software Components

In order to run Polarion, you need to have several other software components installed and configured:

- Apache HTTP Server version 2.2 or higher (latest 2.4.x is recommended)
- Subversion server version 1.6.x or 1.7.x or 1.8.x or 1.9.x
- Oracle Java SE 8 JDK
- PostgreSQL version 8.4 or higher.
- Fonts package urw-fonts (on SUSE and CentOS)

Apache Server

Apache HTTP Server is a required application for Polarion. We highly recommend installing it first. You can download the appropriate distribution at the Apache website at http://httpd.apache.org/. Installation Help for Apache HTTP server is available from Apache at http://httpd.apache.org/docs/2.4/install.html.

htpassword utility

After installation you must ensure that the *htpasswd* utility is executable. This means that the `apache/bin` folder must be present among other paths in *PATH* system variable, OR the full path needs to be specified in the system configuration file `polarion.properties`.

NOTE
On SUSE the utility may be called **httpasswd2**.

Subversion

The Subversion version control system is also required for Polarion (see above for recommended version). You can download the appropriate distribution at http://subversion.apache.org/. Full user/administrator documentation is available on a third-party website at http://svnbook.red-bean.com/.

PostgreSQL

In new installations beginning with version 2015 SR2, Polarion server cannot start if PostgreSQL is not installed and configured. Please use your distribution's package manager to download these two packages: *postgresql* and *postgresql-contrib*. On RedHat you will need the *postgresql-server* package instead of the *postgresql* package.

You can follow the instructions posted online at http://www.postgresql.org/download/ or use one of the following command sets:

CentOS, RedHat:

- **`yum install postgresql`**
- **`yum install postgresql-contrib`**

Debian, Ubuntu:

- **`apt-get install postgresql`**
- **`apt-get install postgresql-contrib`**

SUSE: Available from openSuse Build Service (https://build.opensuse.org/) in the server:database:postgresql project:

- **`yast --install postgresql`**
- **`yast --install postgresql-contrib`**

After installing PostgreSQL, you must configure it and Polarion. This configuration is covered in the following section *Configuration of Third-party Components: PostgreSQL*.

Java SE JDK

You should also download and install the Java SE 8 JDK at http://www.oracle.com/technetwork/java/javase/downloads/index.html.

Fonts Package

Phantom JS, used for preview and PDF export of highcharts, is installed on all OS platforms. On SUSE and CentOS, URW fonts must be explicitly installed (see *Server Software Requirements*), otherwise exported diagrams rendered by Phantom JS will display rectangles instead of letters. Phantom JS is not supported on 32-bit Linux systems by default. If you need it on such systems, contact Polarion support.

3. Configuration of Third-party Components

This section covers various configuration requirements for third-party software that enable Polarion to run with it.

For further inspiration you may also inspect supported OS configurations located in `[Polarion_Unpack_Dir]/libinstall/predefined/[OS]/**`

Apache HTTPD

1. Apache must be compiled with WebDAV module (*mod_dav.so*).

2. Modules from Subversion must be properly installed (*mod_authz_svn.so* and *mod_dav_svn.so*)
   The following lines must be added into an Apache configuration file into the LoadModule section. This can be the default configuration file `httpd.conf` (`[Apache2_Dir]/conf/httpd.conf`), or any other *conf* file, including a new one you create for the purpose. Lines:

```
LoadModule dav_svn_module modules/mod_dav_svn.so
LoadModule authz_svn_module modules/mod_authz_svn.so
```

3. Add the following lines to your httpd.conf:

```
<Location /repo>
DAV svn
SVNPath "/opt/polarion/data/svn/repo"
# our access control policy
AuthzSVNAccessFile "/opt/polarion/data/svn/access"
# try anonymous access first, resort to real
# authentication if necessary.
Satisfy Any
Require valid-user

# How to authenticate a user. (NOTE: Polarion does not
# currently support HTTP Digest access authentication.)
AuthType Basic
AuthName "Subversion repository on localhost"
AuthUserFile "/opt/polarion/data/svn/passwd"
SVNAutoversioning on

</Location>

#polarion specific configuration of apache for svn
# and maven2 repo
# Maven 2 shared repository
<Directory "/opt/polarion/data/shared-maven-repo">
Options Indexes
Order allow,deny
Allow from all
</Directory>
Alias /maven2 "/opt/polarion/data/shared-maven-repo"
```

Pay attention to the name of repository (`/repo`) in the first tag (`<Location/repo>`). It's the name (an alias) of your Subversion repository.

Don't forget to specify correct paths to the real repository folder (in the *SVNPath* parameter), to the access file (in the *AuthzSVNAccessFile* parameter), and to the `passwd` file (in the *AuthUserFile* parameter).

Pay attention to the paths used in the `httpd.conf` example above. They must be the same as those used by default by the `libinstall/default.sh script` - this script won't work if different paths are used.

4. Create a file named `workers.properties` in your Apache server's configuration directory (where the `httpd.conf` file resides) with the following:

```
worker.list=worker1
worker.worker1.type=ajp13
worker.worker1.host=127.0.0.1
worker.worker1.port=8889
worker.worker1.lbfactor=50
worker.worker1.cachesize=50
worker.worker1.cache_timeout=600
worker.worker1.socket_keepalive=1
worker.worker1.recycle_timeout=300
```

5. Create a file named `polarion_mounts.properties` in your Apache server's configuration directory (where the `httpd.conf` file resides). Add the following lines:

   ```
   /polarion/*=worker1
   /polarion=worker1
   /svnwebclient/*=worker1
   /svnwebclient=worker1
   ```

Subversion

Either the Subversion binary (`svn`) must be on PATH or the *javahl* libraries must be properly installed.

Java JRE and SDK

The configuration requires the Java Runtime Environment (JRE) and several files from the JDK (SDK).

- If you use only the JRE (runtime), you must add bin/javac and lib/tools.jar from Java SE JDK, and, optionally, bin/javadoc if you want Javadoc generation to work in Polarion.

- The environment variable JDK_HOME must be properly set.

JDK is required only for running Polarion as a service. For installing Polarion you can use just JRE.

Java Virtual Machine Memory Limit

If you allocate too much memory for the Java Virtual Machine (JVM), the operating system will not initialize it. Diagnosing the issue can be difficult because the service simply does not start and no error log is written. On some Linux versions, a message may appear in the console indicating that the Java Virtual Machine could not be initialized, but without indicating the reason.

The amount of memory you can allocate to the JVM depends on what OS you have and how much total memory exists on the computer. The more total memory the more you can allocate to the JVM before the operating system imposes a limit the better.

PostgreSQL

The location of PostgreSQL utilities depends on the specific Linux distribution. The command examples below assume that the folder containing PostgreSQL binaries is added to your system PATH variable.

The OS user *postgres* is normally created as a result of the PostgreSQL installation. To be able to use the *psql* utility, you must access it as this user.

1. Create the PostgreSQL database storage:

1. Create a folder to store PostgreSQL data and make the postgres user the owner of this folder on Linux. Any location will be fine, as long as the postgres user owns the folder. You can use the following example where the folder for PostgreSQL data is: `/opt/polarion/data/postgres-data`
   ```
   mkdir /opt/polarion/data/postgres-data
   chown postgres:postgres /opt/polarion/data/postgres-data
   ```

2. As the *postgres* user, initialize the database using the `initdb` command. The syntax differs for PostgreSQL versions, as shown in the following examples, with our `/opt/data/postgres-data` folder. Use `psql --version` to check for the version.

   For PostgreSQL version 9.1 and earlier:
   ```
   su postgres
   initdb -D /opt/polarion/data/postgres-data -E utf8
   ```

   For PostgreSQL versions after 9.1:
   ```
   initdb -D /opt/polarion/data/postgres-data -E utf8
   --auth-host=md5
   ```

   In some cases `initdb` is not available immediately after the installation and you need to add to your path.You can use the locate command to find the location of psql. If it cannot be found, please try to use the `updatedb` command and search for it again. More information about `initdb` can be found at: http://www.postgresql.org/docs/9.4/static/app-initdb.html.

3. Start the PostgreSQL server. Remember that all PostgreSQL utilities must be run on behalf of the *postgres* system user, and you must initialize the database storage as described in step 2 above prior to running the server for the first time.
   ```
   su postgres
   pg_ctl -D /opt/polarion/data/postgres-data -l
   /opt/polarion/data/postgres-data/log.out -o "-p 5433" start
   ```

(The *pg ctl* command should be written to a single command line.) You can define additional options, such as the port on which the database should run. Check http://www.postgresql.org/docs/9.4/static/app-pg-ctl.html for an overview of options.

2. Connect to the root database of PostgreSQL:

For configuration of Polarion databases, you need to connect to the root database of PostgreSQL *postgres* using your favorite SQL client (for example: Squirrel[1]), or just use PostgreSQL's built in `psql` tool.

1. On Linux, the operations with PostgreSQL database must be run on behalf of the *postgres* user: **`su postgres`**

2. Connect to the PostgreSQL *postgres* root database on behalf of the *postgres* user (or the database's superuser): `psql -p <port> postgres`

The syntax for psql is: `psql -p <port> -U <user> <database>`
Example: `psql -p 5433 -U polarion polarion_history`

3. Create the *polarion* User and the Databases:

Polarion uses two databases, *polarion* and *polarion_history*, which will be owned by the *polarion* user, so the first step is to create the *polarion* user. To do this, execute the following three SQL statements using your SQL client:

1. Create the *polarion* user (replace the <*your-password*> with a password of your choice[2]):
   ```
   CREATE USER polarion WITH PASSWORD '<your-password>' CREATEROLE;
   ```

2. Create the *polarion* database:
   ```
   CREATE DATABASE polarion OWNER polarion ENCODING 'UTF8';
   ```

3. Create the *polarion_history* database:
   ```
   CREATE DATABASE polarion_history OWNER polarion ENCODING 'UTF8';
   ```

Quit the psql utility (using command: **`\q`**)

4. Install *dblink* extension and PLPGSQL language:

After you have created the two databases for Polarion, you need to install the DBLink extension on both of them. The DBLink extension is used to enable Lucene queries within SQL queries. Some functions used by Polarion use the PLPGSQL language, which needs to be installed as well.

[1] ttp://squirrel-sql.sourceforge.net/

[2] Password cannot contain any of the following characters: @, :, \, ', " or leading/trailing spaces.

To install the extension and language:

1. Connect to the polarion database via psql client as postgres database user (or the superuser): **`psql -p 5433 polarion`**

2. Create the DBLink extension.
   a. If your PostgreSQL version is 9.1 or higher, just execute the following SQL command: **`CREATE EXTENSION dblink;`**

   b. If your PostgreSQL version is 9.0 or lower, use the *psql* tool to set up the extension using the following command:
   - If not connected to psql: **`psql -p <port> polarion -f dblink.sql`**
   - If already connected: **`\i dblink.sql;`**

   The `dblink.sql` file was installed during your installation of PostgreSQL. The location depends on your Linux distribution. You can use the locate command to find it.

3. To verify that the dblink extension was successfully installed, execute this SQL query in your database client:

   ```
   SELECT p.proname
   FROM pg_catalog.pg_namespace n
   JOIN pg_catalog.pg_proc p
   ON p.pronamespace = n.oid
   WHERE n.nspname = 'public';
   ```

   It should return rows containing *db_link* as a prefix.

4. Install the PLPGSQL language by executing the following SQL command:
   `CREATE LANGUAGE plpgsql;`

5. Quit the *psql* utility (**`\q`**) and repeat the above steps for the *polarion_history* database.

6. If you did not run the initialization of the database with `--auth-host=md5`, open `[PG_DATA]/pg_hba.conf` and change all the "host" entries with "trust" to "md5", then restart PostgreSQL. (In our examples, `[PG_DATA]` was `/opt/polarion/data/postgres-data` (the location on which the database was initialized).

5. Optimize the PostgreSQL Database:
Optimization is the same as described for automated installation. Please refer to *After Installation: Optimizing the PostgreSQL Database*.

6. Configure Polarion to use PostgreSQL database:
Add the following property to the `polarion.properties` system configuration file:

```
com.polarion.platform.internalPG=polarion:<password>@<hostname>:<port>
```

Example:

```
com.polarion.platform.internalPG=polarion:polarion@localhost:5433
```

If you have the *polarion* and *polarion_history* databases running on different servers, you can use these two properties rather than the above property:

```
com.polarion.platform.sql.internalDB.head.url=jdbc:postgresql:
//<hostname>:<port>/polarion?user=<userName>&password=<password>

com.polarion.platform.sql.internalDB.historical.url=jdbc:postgresq
l://<hostname>:<port>/polarion_history?user=<userName>&password=
<password>
```

Example:[3]

```
com.polarion.platform.sql.internalDB.head.url=jdbc:postgresql://lo
calhost:5432/polarion?user=polarion&password=polarion

com.polarion.platform.sql.internalDB.historical.url=jdbc:postgresq
l://localhost:5432/polarion_history?user=polarion&password=polario
n
```

After you have set the system properties, reindex your Polarion installation so that the PostgreSQL database will be populated. Using any SQL client after the reindex, you should see the database filled in the *polarion* schema.

7. Allowing remote connections to the PostgreSQL server (optional):

By default the PostgreSQL database is only available on *localhost* after a fresh install. If you need to have access from outside the host machine, please follow these steps to allow remote access:

1. Change the listener addresses in the `postgresql.conf` file. This file can be found in the PostgreSQL data folder, the one you have defined under *Starting/Stopping PostgreSQL*. In this file, change: `#listen_addresses='localhost'` to `listen_addresses='*'`

2. In the same folder, you will find the `pg_hba.conf` file. Open it in a text editor and add the following line: `host all all  0.0.0.0/0  md5`

[3] The properties should be written on a single line.

SECURITY ADVISORY

If you allow external connection to the database, the external user has blanket access to the entire database content and effectively to all Polarion data - all projects, the HEAD revision as well as history. There is no further access control. All Polarion user names will also be visible. (Polarion user passwords will not be visible as those are not stored in a database or SVN.) Access is read-only.

Administrators are advised to consider carefully before opening database access in a system in use for production.

8. Check Shared Buffers Configuration

Check that the value of the *shared_buffers* parameter in `postgresql.conf` does not exceed the available shared memory configured by the kernel parameter *SHMMAX.*

4. Installing Polarion

After the required third-party software is installed and configured, you are ready to undertake the installation of the Polarion platform itself.

The Linux distribution comes in a Zip archive. The archive file name includes the version number and the word *linux*. For example: `PolarionALM_nnnn_linux.zip` (where *n* is a number 0-9).

Installation Procedure

1. Be sure that all required third-party components are properly installed and configured.

2. Unpack the distribution Zip archive to temporary location

3. Change current directory to *Polarion*

4. If SELinux is bundled with/installed on your Linux OS, make sure it is not activated. Check the status of SELinux with command □**`/usr/sbin/sestatus -v`**.

5. Create a Unix system account named *polarion* for the Polarion server.

6. Review the path variables defined in the `libinstall/default.sh`. Set up *v_web_user* and *v_web_group*.

7. Execute (under root):
   ```
   chmod +x manual_install.sh; ./manual_install.sh
   ```

This will install Polarion into the standard location /opt/polarion. Data will be installed into the standard location `/opt/polarion/data`.

Setting Permissions

After manual installation you need to set up permissions on unpacked folders.
Standard permissions are:

- Read only for everybody, owner and group root:root on folders and its subfolders:
  `/opt/polarion/polarion`
  `/opt/polarion/maven`

- Writeable on folder and its subfolders: `/opt/polarion/data`

For “polarion”:”web user”, where *polarion* is a system account, which you need to create manually and web user is user used by Apache.

For more details see `[UnpackDir]/libinstall/helper_functions.sh`, function *setPermissions()*.

Notes on the Manual Installation

Note that if Polarion is already installed, any user data (repository, builds, reports) will not be overwritten. However, it is recommended to back up any changed files in `/opt/polarion` (typically `polarion.properties` and any customized shell scripts), as these will be overwritten.

An example of some possible changes is given below:

- Change the value of the *host.name* parameter from *localhost* to the host name specified in `httpd.conf`.

- Check that the repository alias in the svn.url parameter corresponds to the one in the `httpd.conf` file (see *Required Configuration*).

- Check that the correct paths to the files passwd and access are specified in the relevant parameters *svn.passwd.file* and *svn.access.file*.

- Note that you may need to reconfigure the *htpasswd.path* parameter and set it to `/usr/sbin/htpasswd2` if necessary (instead of the usual `/usr/local/apache2/bin/htpasswd`).

- Check Apache's *htpasswd* utility path in the *htpasswd.path* parameter.

- Check that the port specified in the *tomcat.ajp13-port* parameter corresponds to the port specified in the *worker.worker1.port* parameter of the `workers.properties` file.

- Specify the correct SMTP host in the *announcer.smtp.host* parameter.

5. Configuring Polarion

1. Execute commands from installation temporary directory to create new Subversion repository:

   ```
   svnadmin create /opt/polarion/data/svn/repo -fs-type fsfs
   chown -R "webuser":"webgroup" /opt/polarion/data/svn/repo
   ```

   The items “webuser” and “webgroup” are the user and group under them on the Apache server.

2. Import production data *or* demo data configuration. *You cannot import both*. You can execute both scripts *only* on fresh repository!

   ```
   [POLARION]/bin/polarion.init init
   ```
   or
   ```
   [POLARION]/bin/polarion.init demo
   ```

   Note: Parameters could be different on your OS. For example values of User and Group directives from httpd.conf: the value of Group in the standard Apache distribution is #-1 (which is the numerical group ID -1) can't be used. Use apache/apache as user and group instead.

   WARNING

   Once the script (`[POLARION]/bin/polarion.init init`, or `[POLARION]/bin/polarion.init demo`) has been executed, it is not possible to install demo data without deleting the Subversion repository in `/srv/polarion/data/svn/repo`.

3. Before using the system for production, see *Securing the Polarion Activation Application* in the *Polarion Installation Guide for Linux* and perform the recommended configuration as described.

6. Next Steps

After completing manual installation, the following sections in this guide are useful to go over:

- System Startup and Shutdown
- After Installation
- Appendix: Default Users and Groups
- Appendix: Enabling Email Notifications

Supported Microsoft Office® Versions

Users of a Polarion product supporting current or legacy Microsoft Office data interchange features for Microsoft Word™ and/or Excel®, please note the following table which outlines compatible versions of Microsoft Office applications.

				Microsoft Office Version			
Feature	Action	Extension	Format	2003	2007	2010	2013
Polarion LiveDocs (2011)	Word Import & round-trip	DOCX	DOCX	UNSUPPORTED [1]	SUPPORTED	SUPPORTED	SUPPORTED
	Excel Import & round-trip	XLSX	XLSX	UNSUPPORTED [1]	SUPPORTED	SUPPORTED	SUPPORTED
Polarion LiveDocs (2010) [2]	Excel export	XLS	XML	UNSUPPORTED [2]	UNSUPPORTED [2]	UNSUPPORTED [2]	UNSUPPORTED [2]
	Word export	DOC	XML	UNSUPPORTED [2]	UNSUPPORTED [2]	UNSUPPORTED	UNSUPPORTED

NOTES:

1. Beginning with Polarion version 2014-SR1, Microsoft Office 2003 is no longer supported in Polarion LiveDocs.

2. Prior to Polarion version 2011, “Live Documents” referred to Microsoft Office Word and Excel documents based on special document templates developed by Polarion Software, which could define and store Polarion Work Items. Beginning with version 2011, Polarion refactored the technology completely but maintained backward compatibility. Beginning with Polarion version 2014-SR1 support for this legacy format is dropped completely.