XPE/Gecko Manual

Installation and Administration

XPE Version 2.05

Gecko Version 3.3

Release Date: 2009-02-02

Version: 0.5

SoftTouchIT http://www.softtouchit.com

Table of Contents

XPE Distribution

When you receive an XPE distribution from an authorized reseller, it should contain the following directories and file:

doc portal theme xars xpe2.05.tgz

1. xpe_2.0.5.tgz : this is the xpe server;
2. theme: portal themes
3. portal: portal applications
4. xars: XPE applications
5. doc: documents
6. license.xml: the license file

If any of the files are missing, please contact your software provider.

The license.xml is usually shipped to you via a separate media and is unique to the deployment. The license.xml file should not be copied and shared because it is against the law.

XPE Basics

This chapter covers the basics about XPE. You need those concepts in order to understand the following chapters.

XPE version

XPE version number consists of 3 digits in the following format: x.xx. Changes of the first digit indicate a major version change and applications developed for older version of XPE may need to change. Changes of the second digit indicate a minor version change. Application developed for older version of XPE should not need to change. XPE configurations may be changed. Changes to the last digit indicate bug fixes.

The current version of XPE is 2.03.

XPE Archive Resource (XAR)

XPE Archive Resource is a file format for XPE applications. Each XPE application is packaged into an XAR file with the file extension: “xar”.

Official xar files released by SoftTouchIT can be downloaded from http://www.softtouchit.com .

XPE Installation

Minimum Hardware Requirements

Before installing XPE, make sure that your hardware satisfies the minimum requirements of XPE:

1. Any desktop or server CPU manufactured by Intel or AMD after 2003 or faster;
2. Physical RAM 512 Mb or above
3. Diskspace: 5 GB or more for basic installation

Optimum Hardware Specifications

1. Two or more multi-core CPUs
2. Physical RAM greater than 4 GB
3. SCSI disks or SAN storage solutions

Prepare the Linux Environment

Create an Account for XPE

First, create an account with username 'xpe' . This is highly recommended. XPE should be run as this user. Never run XPE as root.

Unset the DISPLAY variable

When you connect to a Linux terminal, some ssh or telnet agents automatically sets the DISPLAY variable. This can cause problems to all image processing modules in XPE because Java will try to connect to the display specified by the DISPLAY variable and will most likely fail because those agents do not support X11 or they have been disconnected.

To unset the DISPLAY variable, simply add the following in the .bashrc file at the root of the xpe account:

unset DISPLAY

Set the limit for maximum number of open files

Linux usually limits the maximum number of open files. There are two limits: a global limit and a per user limit. We need to set both limits to larger numbers for XPE or a "too many open files" exception may be thrown by XPE.

To set the global limit, one needs to add this to /etc/sysctl.conf:

fs.file-max = 65536

You will have to run the following command before your settings become effective:

sysctl -p

To set the per user limit, you need to add the following lines to /etc/securitty/limits.conf:

xpe soft nofile 50000

xpe hard nofile 60000

Note that the numbers specified in the limits.conf must be smaller than the number specified in sysctl.conf .

To check the current per-user limit, run:

ulimit -n

Install JVM

First, make sure gcj is not installed or you have to remove it. Please consult the manual of your linux distribution to remove it.

Download Java from SUN and installed it. Please consult the manual of your linux distribution on how to install Java. This typically involves using rpm or other package manager.

XPE requires Java 1.5 or above.

Once java is installed, you must make sure that Java's bin directory is in the PATH of xpe account. We recommend that you do the following regardless how you have installed Java.

This can be done by editing the .bashrc file under the home directory:

export JAVA_HOME={the java home path}

export PATH=$JAVA_HOME/bin:$PATH

Install XPE

Warning: the following operations should be performed as the xpe user. DO NOT perform the following operations as root!

First, untar the xpe distribution by the following command under /home/xpe director:

tar -xzvf xpe_2.0.5.tgz

This should create a xpe2.0 directory under /home/xpe directory. Now, copy the license.xml file to xpe2.0/etc directory. Rename the license file to license.xml if necessary.

Second, edit etc/password.xml file and add a password attribute to the admin account:

<xpe>

<account username="admin" **password=”admin”** realm="xpe_console" ha1="1401078824318ccbcc612fdba5584c05"></account>

<account username="xpe" realm="xpe_console" ha1="b6e123b4bcbb7dee925355d86215c394"></account>

</xpe>

Finally, start xpe by issuing the following commend under /home/xpe/xpe2.0 directory:

sh run.sh

Now, point your browser to http://localhost:8888/xpe/console/about and login with username admin and the password you just set. If you see an about page, then congratulations. You have just installed XPE.

Install Common XPE applications

Point your browser to http://localhost:8888/xpe/console/doc/deploy and then upload each of the xar files under the xars directory of the xpe distribution.

Install Common XPE Gecko (Portal)

Point your browser to http://localhost:8888/xpe/console/doc/deploy and then upload xpe_portal.xar under the xars directory of the xpe distribution.

Make sure the following xar files are deployed:

xpe_portal.xar.,xpe_rms.xar, xpe_sso.xar, xpe_stat.xar, xpe_form.xar, xpe_ajax.xar

You can check the xar files installed from the URI: http://localhost:8888/xpe/console/deployed

Next, you need to initialise rms, sso and stat.

First, point your browser to http://localhost:8888/xpe/secu/init

Refresh this page at least 3 times. Ignore any error messages appeared in the first two times.

Second, point your browser to http://localhost:8888/xpe/stat/init

Refresh this page at least 3 times. Ignore any error messages appeared in the first two times.

Finally, point your browser to http://localhost:8888/xpe/rms/init

Refresh this page at least 3 times. Ignore any error messages appeared in the first two times.

Now, point your browser to http://localhost:8888/xpe/portal

Login as admin using password admin.

Next, deploy portal applications using the “Deploy xlets” menu or point your browser to http://localhost:8888/xpe/portal/admin/xlet/deploy .

Make sure the following portal applications are deployed using this feature:

xpe_portal.xar, xpe_portal_resource.xar, xpe_portal_rss.xar, xpe_ajax.xar, and any xar files under the portal directory of the xpe distribution.

You will then need to upload some common themes (css and image files) used by some portal applications. Just click the “Upload theme” menu and upload all the zip files under the theme directory of the xpe distribution.

Finally, we need to upload some default content type definitions: contentType.zip . This file is found under the portal directory of the xpe distribution.

Now, you have finished the installation of XPE Gecko.

Understanding XPE structure

Director Structure of XPE

Once installed, under the home directory XPE, you will find the following directories:

1. etc, this directory contains all the configuration files you may need to modify
2. work, this directory is used by XPE applications to store long term data
3. temp, this directory is used by XPE applications to store temporary data and it will be cleaned every time XPE restarts
4. xpe,this is the XPE engine directory and one should never change anything under this directory.

Configuring XPE

Under the XPE home directory, for example, xpe2.03, there is an etc directory. All configuration files are under this directory:

- log4j.xml, this file controls the logging level of XPE
- password.xml, this file configures accounts for XPE console
- server.xml, this file configures the run-time properties of the XPE server

Configuring server.xml

This XML document has the following structure:

```
<jvm  xmlns="http://www.xmlpipe.org/xpe"  args="-server -Xmx256m -Xms128m"  version="2.0" >
   <!-- any JVM properties can be defined here -->
    <!--You may add any additional global properties here -->
 <!--
   <property name="example.variable" value="example value" />
-->

   <!-- The following variables should not be changed -->
   <property name="javax.xml.transform.TransformerFactory" value="net.sf.saxon.TransformerFactoryImpl" />
   <property name="org.xml.sax.driver" value="org.apache.xerces.parsers.SAXParser" />
   <!-- This configures the maximum size for submitted form -->
   <property name="org.mortbay.http.HttpRequest.maxFormContentSize" value="500000" />
   <!-- the port is used for management -->
```

```
   <server port="2138" >
      <!-- One or more listeners can be configured to a server -->
      <!--three types of listeners are supported, http, https, ajp13 , if not specified, then it is http. -->
      <listener port="{port number}" minThreads="{minimum threads}" maxThreads="{maximum number of threads}" />
      <listener port="8888" minThreads="5" maxThreads="100" />

      <!--One or more contexts can be configured to a server.  A context is a HTTP context that maps a
      http request to a particular handler. -->
      <context path="/" base="ROOT" >
         <forward >
            <map from="/" to="/xpe/portal/66fd4782-1119-1000-80b6-46648aa451c5" />
            <map from="/echo" to="/xpe/mk/echo" />
         </forward>
         <resource dirAllowed="no" />
      </context>

      <context path="/xpe" >
         <!-- An XPE engine is effectively a handler for HTTP requests
               base - the base directory of XPE
               temp - relative to the installation directory, temp directory used by XPE
               work - working directory relative to the installation directory
               deploy - deployed applications  relative to the installation directory
               -->
         <xpe   base="./xpe" temp="temp" work="work" deploy="deploy"  >
         </xpe>
      </context>
      </server>
</jvm>
```

The jvm element

The jvm element defines the basic properties of the Java Virtual Machine that hosts the XPE server. The arg attributes may contain any arguments you wish to pass to the JVM. Please consult with your JVM manual for details of those applicable

arguments. The example given above is for SUN's JVM, which selects the server hot-spot VM with a maximum memory size of 256 Mb and minimum memory size of 128 Mb.

The jvm element may contain a number of property elements. Each property element specifies a Java property. Unless you are absolutely sure, the following properties should not be changed.

Common Java Properties

Property Name	Value	Note
javax.xml.transform.TransformerFactory	net.sf.saxon.TransformerFactoryImpl	This specifies the default XLST engine to use. In theory, one can switch from one engine to another. However, different engines may behavior differently.
org.xml.sax.driver	org.apache.xerces.parsers.SAXParser	This specifies the default XML parser. The apache parser has been chosen for its speed and reliability. We suggest to keep this value unchanged.
org.mortbay.http.HttpRequest.maxFormContentSize	500000	This specifies the maximum size of a form in bytes that is allowed to be submitted to XPE server. If your applications have large forms, then you need to increase this value. Otherwise, keep it small in order to minimise the risk of DoS attack.

One may also add additional global properties and those properties can be used by other filters or xlets.

The server element

The server element configures the dynamic properties of the XPE server. It has a port attribute, which defines the management port that is used to stop the the server.

A server can be configured with one or more listeners. Each listener listens to a network port. A default listener is HTTP listener. The listener element allows a type attribute. The value of the type can be "socket", "ssl" or "ajp13". If not specified, the default value is "socket", which listens to standard HTTP protocol. If the type is set to "ssl", it then supports HTTPS. The type, "ajp13", is used to integrate the server with Apache, which would require mod_jk to be installed.

A server should always have a default listener at port 8888. This is like a "loop-back" port for the consumption of internal services.

A server can be configured with a number of HTTP contexts. A HTTP context maps a HTTP request to a handler. In the example shown above, the first context maps the root context to a forward handler and a resource handler. The forward handler forwards any request matches the from URI to the new URI specified by the "to" attribute. The resource handler will deliver any static contents that are not matched by the forward hander.

The xpe element

The xpe element configures the dynamic properties of XPE engine. We recommend that you do not change any of its attributes.

Configuring password.xml

This file contains a list of accounts for the XPE console application, which is used as Web Administration tool for application deployment and server monitoring tool.

The file may look like this:

<?xml version="1.0" encoding="UTF-8"?><xpe>

<account username="admin" realm="xpe_console" ha1="1401078824318ccbcc612fdba5584c05"></account>

</xpe>

The xpe element is the top element and it has no attributes. It may have one or more account elements. Each account element represents an account. The ha1 attribute is generated by XPE. To add a new account, one adds an account element to the xpe element. For example,

<account username="{a unique username}" realm="xpe_console" password="{the password of the user}" />

Once the file is saved, XPE will remove the password attribute and replace with a ha1 attribute.

Running XPE

Start and Stop XPE

To start XPE, just execute the run.sh on Unix/Linux or run.bat script.

To stop XPE, one needs to simply terminate the correspond Java process or execute the stop.sh script on Unix/Linux or stop.bat script on Windlows.

Deploy XPE Applications

An XPE application comes in one single XML ARchive (XAR) file, which always has a .xar suffix. There are two ways to deploy an XPE application: using OS's command line or using the web interface provided by XPE Console (to be discussed in the next chapter).

To deploy an XPE application with the command line option, one simply copies an XAR file to the *deployable* directory under the home directory of XPE. A deployment report named,deploy.xml, is created under the deploy directory under the home directory of XPE and the XAR file is removed from the deployable directory. If the XAR file contains any error that XPE cannot deploy, the fill will remain in the deployable directory.

Deploying XPE applications can be done when XPE is running.

Remove XPE Applications

Remove an XPE application can only be done when the XPE server is stopped. Changing directory to the deploy directory under XPE home directory, one will see a list of directories. Each directory corresponds to an XPE application except the dataSource directory, which contains all the database connection pools. Each XPE application directory contains a readme.txt and an app.xar file. The readme.txt contains a brief description of the application and its application URI. The app.xar file contains the application itself. To remove the application, one simply remove the corresponding application directory. Once a directory, one needs to start XPE again.

Integrating XPE with Apache

Apache is a popular web server. Apache can be used in combination with XPE with Apache acting as a gateway to XPE.

We use mod_jk as the connector between Apache and XPE. Hence, one must first intall the mod_jk module for Apache.

The server.xml of XPE also needs to be modified and an ajp13 listener must be added:

```
<listener port="8001" minThreads="5" maxThreads="100" type=”ajp13” />
```

Next, the httpd.conf of Apache needs to be modified to use the specified ajp13 port.

XPE Administration

XPE Console is an XPE application that can be used to manage other XPE applications. Developers will need to this application to deploy XPE applications.

To access the console, one just points a browser to http://localhost:{port}/{xpe}/console/about

The application is password protected and the account is configured by the password.xml found under the etc directory of XPE home directory. The realm attribute must be set to “xpe_console”.

This application provides a number of useful tools for system administrator and XPE developers:

URI Mapping

One or more external URI patterns can be mapped to an XML Pipe. This page displays all the mapping rules. The wildcard rules are displayed first followed by exact rules. Rules are also grouped by their corresponding HTTP methods.

About XPE

This page shows the current version of XPE and its build number.

XPE License

This page displays the end user license of XPE.

System Information

This page displays the current memory usage of XPE and the current JVM properties.

The current memory usage is a very good indicator of the health of XPE. There are four numbers here: “Maximum”, which indicates the total amount of memory allocated to the JVM; “Total used”, which is the largest amount of memory the JVM recently actually used; “Free” means how much memory the JVM has freed from the “Total used”; The last number “Used percentage” is the percentage of memory currently in use by the JVM. For most tasks, XPE is very memory efficient so the value of “Used percentage” should be very small, typically less than 30%. If this value is above 80% most of the time, then it is a good idea to allocate more memory to the JVM. Obviously, the memory usage is highly application dependent. For example, it is well known that the generation of PDF using the FOP technology is highly memory hungary and it is good idea to have 1 G of RAM for every 100 pages generated.

The JVM properties display the actual values used by the JVM.

Environment

This page displays the applications deployed to XPE, pipes associated with each application, and filters used by each pipe.

XPE pipes use lazy binding. This means that a pipe can refer to a filter even if it does not exist yet. Of course, if a pipe that refers to an non-existing filter gets executed, an exception will be thrown.

Deployed Data Source

This page lists all the data sources available to applications. A data source is a JDBC connection pool. This is typically used by an application to connect to a JDBC compliant database engine.

Deploy Application

This page allows one to upload an XPE application to XPE. If the application contains any error that XPE cannot accept, error messages with the cause of the error will be displayed.

Check Deployed XPE Applications

Deployed XPE applications can be viewed from the XPE console under the following URI:

http://localhost:8888/xpe/console/deployed

The application name if it has one or its URI is displayed together with the version, build number, and release date of the application.

Update XPE Applications

To update XPE applications, one can access the “Update Applications” menu from the console. If an application is already installed, one can re-install the application. If an update is available for an installed application, one can upgrade it. If an upgrade has problem, it is always possible to downgrade to a previous version.

However, a major upgrade may not be able to reverse to the previous version because some of the changes may not be reversible.

This feature is only available if the XPE server can access the internet.

Gecko Administration

Gecko is a portal engine built on top of XPE. It uses XPE Content Management System (CMS) as its storage engine.

General Portal Administration

Different types of Gecko files

Gecko accepts a few different types of files:

1. Portal Application: this is an XPE application plus additional Xlets. The file should have a ".xar" suffix.
2. Theme Pack: a theme pack contains CSS, image, Java Scripts or other static resources. A theme pack is a zipped file and should have a ".zip" suffix.
3. Site Backup: A site backup contains site definitions, templates, attribute groups, content types, and other files needed for a site excluding users and roles, and theme packs. A site backup can optionally exclude contents as well.
4. Resources: A resource file is a zip file that conforming to the Gecko CMS data format. It can be used to upload any data into Gecko.
5. Security file: A security file contains security information including group definition, role definition, and users. Security file is useful for migrating security information from one environment to another.

Deploy XPE Portal Applications

This is found under "Xlet Management" of General Portal Administration.

Upload a theme pack

A theme pack is a zip file that contains CSS files and images for the styling of a site. Typically, a theme pack is organised under a root directory named after the site it intends to style. Under the root directory, there are usually a css directory for CSS files and an image directory for images.

A theme pack can be uploaded under "CMS administration" using the "Upload theme" link.

Site Backup

A site can be packed in one single zip file. There are two methods for backing up a site: backup with contents or backup without contents. The first method allows a full backup of a site including all the contents stored in the CMS. The second method only backups the site structure and associated configurations.

Import a Site

This is found under the "CMS administration". This feature allows a site backup to be imported into Gecko.

User Administration and Access Control

Gecko uses a Role Based Access Control mechanism. To control the access rights of a user, an administrator assigns roles to the user. A role is associated with access privileges. When developing applications on Gecko, developers usually define new application specific roles on Gecko. Those roles are then associated with page controls and xlet controls.

Gecko has two system roles: admin and portalAdmin. Any user with the admin role can create manage roles and users. Any user with the portalAdmin role can log into Gecko backend and perform general Gecko administration tasks. The user administration facilities are available under the "Security administration" tag and is available only if a user has the "admin" role.

A user must have both admin and portalAmin roles to be able to use the user management features in Gecko. For a person who only needs to develop applications on Gecko, then only a portalAdmin role is needed.

A default admin account is created for Gecko when it is initialised. The account has a username "admin" with password "admin". One should immediately change the password after login.

Performance Tuning

Common Performance Tuning Procedure

You should do performance tests after each step to see if there is any improvements.

1. Check the maximum number of open file limit. This can be done by the command 'ulimit -n'. Set the number to a large one such as 20000. See the XPE Installation chapter for details.
2. Check the RAM assigned to XPE. This is defined in etc/server.xml. Increase the RAM to 2G if you have at least 4G physical RAM.
3. Check the logging level of XPE. This is defined in etc/log4j.xml. Increase the logging level to 'fatal' .
4. Making Gecko xlets static.

XPE Tuning

XPE is a Java application so the common Java tuning tips can be applied here:http://java.sun.com/performance/reference/whitepapers/tuning.html .

Some of the related tuning tips are copied here:

Tuning for Throughput

In most cases, increasing the memory allocated to Java will improve the performance. Here is an example on a system with 4 GB of memory:

<jvm xmlns="http://www.xmlpipe.org/xpe" args="-server -Xmx3000m -Xms3000m " >

Comments:

* -Xmx3000m -Xms3000m

  Configures a large Java heap to take advantage of the large memory system.

If your system is capable of running 32 threads simultaneously (CPU's and cores or contexts, you may try the following:

<jvm xmlns="http://www.xmlpipe.org/xpe" args="-server -Xmx3000m -Xms3000m -Xmn2g -Xss128k -XX:+UseParallelGC -XX:ParallelGCThreads=20" >

Comments:

* -Xmn2g

Configures a large heap for the young generation (which can be collected in parallel), again taking advantage of the large memory system. It helps prevent short lived objects from being prematurely promoted to the old generation, where garbage collection is more expensive.

* -Xss128k

Reduces the default maximum thread stack size, which allows more of the process' virtual memory address space to be used by the Java heap.

* -XX:+UseParallelGC

Selects the parallel garbage collector for the new generation of the Java heap (note: this is generally the default on server-class machines)

* -XX:ParallelGCThreads=20

Reduces the number of garbage collection threads. The default would be equal to the processor count, which would probably be unnecessarily high on a 32 thread capable system.

As a simple test of XPE performance, one can test the echo URI on the host by using the ab command:

ab -c 10 -k -n 100000 http://localhost:8888/xpe/mk/echo

The above command sends 10 concurrent requests and 100000 requests in total to the server. XPE should be able to reach a performance over 8000 requests on a server with Core 2 Duo 2.0 GHz CPU.

Using AJP13 Connector

There are two approaches to connect XPE to Apace: apache proxy or AJP13. AJP13 is much more efficient but apache proxy is simpler to use and offers better flexibility. However, for performance reason, AJP13 should be used.

Using the 80 port directly

Add the following line to your server.xml:

```
<listener port="80" minThreads="5" maxThreads="100" />
```

Restart XPE and it will listen on port 80. However, Linux has an access control policy that prevents normal user from accessing this port. Please consult your Linux distribution to remove this restriction.

Increasing XPE Logging Level

Low level logging can be quite verbose and very expensive. Increasing the logging level to info or fatal will significantly increase the performance of XPE.

Gecko Tuning

Making an Xlet Static

When developing a web application, most xlets are configured as dynamic. This means that each time a page is loaded, every xlet is executed. Some pages may contains tens of xlets which result in slower performance. However, most xlets

can be considered static so the result of an xlet can be cached. A static xlet will still be executed every hour so any xlet that does not need update every time a page is loaded can be configured as static.

Trouble Shooting

XPE not starting up correctly

Use your browser and open http://{host}:8888/xpe/mk/echo, if you get the following messages:

<error cause="org.xml.pipe.PipeNotFoundException">

<message>No matched pipe found for the URI:/xpe/license</message>

</error>

This means that the license file is missing. Just place the license.xml file to etc directory and restart XPE.

Gecko not starting up correctly

It has been reported that XPE and Gecko may not start up correctly after reboot. When this happens, just manually kill the XPE process and then restart it manually.

Need More Information?

SoftTouchIT

General information about XPE, updates and latest downloads can be found from the http://softtouchit.com site.