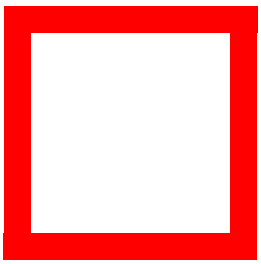
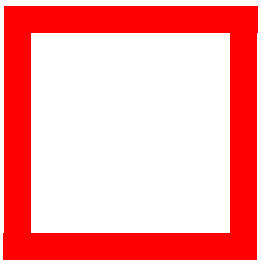
XTT Installation Guide

X-treme Testing Tool version 0.1

Created on 5 March 2009.



Contents



Introduction

X-treme Testing Tool (XTT) is a Java-based testing tool that helps test X-treme FrameWork (XF) products. It runs in both local and remote mode. Further details about XTT are available in the *User Interface Guide*, in the *Function Modules Guide* and in the *Utilities Guide*.

This Manual describes the installation prerequisites and procedure for XTT.

Purpose and Scope

This Manual is designed to help you install, configure and operate XTT.

It includes the following parts:

- Introduction (this chapter): provides an Overview of the current Manual.
- Installing a Brand New XTT System: describes the installation prerequisites and lists the installation steps to install XTT.
- Configuring XTT: describes the configuration file and its role in XTT installation.
- Building XTT Code: describes the build.xml file and its role in XTT installation.
- Appendix A: includes an example of the build.xml file.

Target Audience

This document is for regular users who use XTT, for software testers who use XTT, and for developers who extend XTT.

Conventions

This guide uses the following typographic conventions:

- **Enter** -- menu options, page and dialog titles, section and field names, and other visual elements of the GUI.
- `cmd` -- text that you enter using the keyboard, and other code.
- *Note* -- text that gives additional and relevant information.

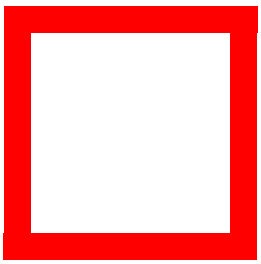
Related Documents

This Manual is part of the XTT v0.1 documentation set that includes the following documents:

- User Interface Guide
- Utilities Guide
- Function Modules Guide

Summary of Changes

Issue	Date	Release Changes
v 0.1	March 2, 2009	For XTT Release (Draft Version)



Installing a Brand New XTT System

Overview

X-treme Testing Tool (XTT) is a platform and branch independent Java-based testing tool. XTT is all Java-based and the same package can be used for all Java-enabled platforms. This chapter describes how to install and commission a brand-new XTT system. Follow the steps from Step 1 to Step 4.

This guide is for the regular user who is installing XTT for the first time; for the software tester who has used XTT for testing, and for the developer who is extending XTT.

There are dependencies between the different installation tasks. For example, you must install Java on your computer before you install the XTT package.

Installation Steps

Step 1: Check that you meet the Prerequisites

To install and commission a brand-new XTT system:

- You can use either the Windows or UNIX operating systems. If you are using Windows, we recommend that you use either the Windows 2000 or the Windows XP operating system.
- You must install JDK 1.5 on your computer.
- You must install ANT 1.6.5 or later (for developers extending XTT)
- You might require to install Oracle 9i client.

Note: It is necessary to install Oracle 9i client on your computer if you need to use FunctionModule_SQL together with an Oracle database. However, even then, the installation of the JDBC driver is enough.

Step 2: Download the XTT Package

The XTT package can be downloaded from the Fondue computer. The path to this package is:

http://fondue.len.tantau.com/Development/MAIN/LATEST_XTT/XTT_Package_2.0.0XXX.tar.gz

---

Note: If you are unable to view or access the above link, please contact 724 Professional Services. They will help you download the XTT package

---

The XTT package is in the form of a tar.gz file, where XXX comes from the xtt.buildnumber file and the number is always incremented when a new XTT build is made and put on the Fondue computer.

The build number of the package changes when XTT changes. So, any enhancements to XTT are reflected as a change in the build number.

Step 3: Extract and Install the XTT tar.gz file

To extract the XTT tar.gz file on a Windows computer, use any unzipping software to unzip and extract the XTT tar.gz file.

Note: Winrar should be the unzipping software of choice.

To extract the XTT tar.gz file on a UNIX computer, use the following command:

```
tar -zxvf  XTT_Package_2.0.0XXX.tar.gz
```

Extract the XTT package on your Windows or UNIX computer.

Step 4: Run XTT to display the GUI

1. Click **Start** and then click **Run**.
2. Enter `cmd` and click **OK**. The command prompt appears.
3. Navigate to the folder where you have extracted the XTT package.
4. Set the following paths at the command prompt if you are using a Windows computer:

   ```
   set JAVA_HOME="<Your JDK installed folder>"
   set PATH="<Your JDK installed folder>\bin";%PATH%
   ```

   OR

   Set the following paths at the command prompt if you are using a UNIX computer:

   ```
   export JAVA_HOME="<Your JDK installed folder>"
   export PATH="<Your JDK installed folder>/bin":$PATH
   ```

5. Enter `java -cp lib\jdom.jar -jar lib\xtt.jar -g` at the command prompt and press **Enter**.

The XTT application GUI appears.

Command Line Options

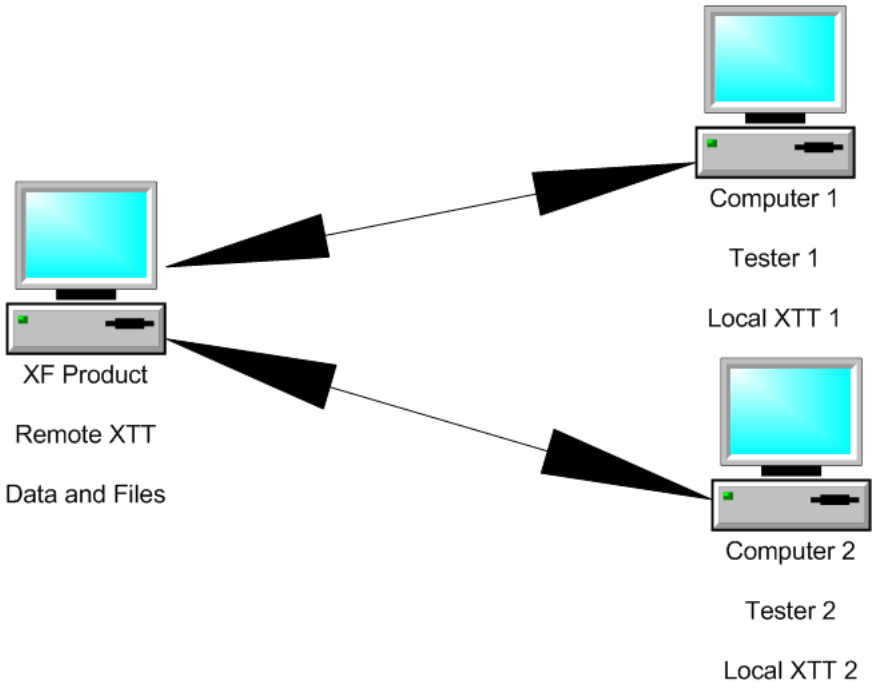
There are several command line options that are used when XTT runs in command line mode. These options are:

Command Line Option	Description
-c <config>, --config <config>	Specifies the file where the configuration is
--dumpfunctions	Lists all available functions of Function Modules
-g, --gui	Loads the interactive XTT GUI
--help	This help
--nostartonload	Removes any 'start on load' options in a configuration
-l <tests>,--list <tests>	Specifies a test file list
-s <test>,--single <test>	Runs the test <test>
--selftest	Runs self tests, and shows version numbers

Running XTT in Remote Mode

Remote XTT runs on a different computer from the ones that the software tester uses to test XTT. Remote XTT is used as a conduit for data and files.

The following diagram shows how XTT runs in remote mode.



The diagram above is a pictorial representation that shows how XTT runs in Remote mode. Remote XTT is installed on the same computer where the X-treme FrameWork (XF) product is installed. There are two software testers, each on Computer 1 and Computer 2, where local versions of XTT are installed.

Both testers need to test two different functionalities of the XF product installed on the computer where Remote XTT is installed. Both testers use Remote XTT to upload Data and Files onto the computer that houses the XF product. In this way, they are able to test the various functionalities of different XF products using Remote XTT.

Tip: While running XTT In Remote mode, modify the system IP address and remote IP address in the remoteconfig.xml file.

- To start Remote XTT in command-line mode on a Windows, Solaris or Linux computer, type the following command and press **Enter**:

  ```
  ./bin/xtt RemoteXTT -c remoteconfig.xml
  ```

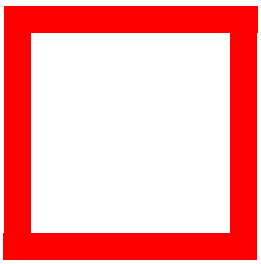
  Note: Ensure that the path of JDK is set correctly in the PATH environment variable.

For example, for a Windows computer,

```
set PATH=<Your JDK installed folder>\bin;%PATH%
```

For example, for a UNIX computer,

```
export PATH=<Your JDK installed folder>/bin:$PATH
```



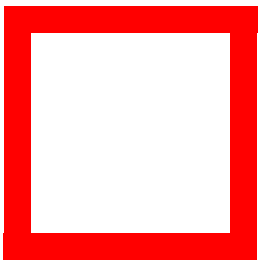
Configuring XTT

This section details how to configure XTT using the configuration file, and is meant for all customers who intend to extract and install XTT. The configuration file is a static file and includes the entire configuration for all the function modules. For example, it may contain the IP address for the HTTP protocol, as well as the port number.

XTT generates a new configuration file when you open the internal configuration editor either by choosing the default configuration or by adding a new one.

You can edit the configuration file for your personalized configuration for running XTT. Once you generate the configuration file, you can save the file and change it as you wish. You can edit the IP address of the computer where XTT is going to run. This IP address is present under the <system> tag, between the <ip> and </ip> tags.

You can also edit the RemoteIP address, which is the IP address of the computer where RemoteXTT is running. This IP address is present under the <system> tag between the <remoteip> and </remoteip> tags.

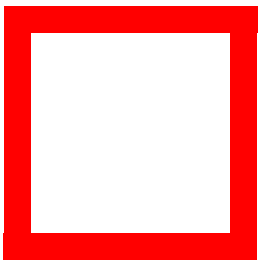


Building XTT Code

This section details how to build XTT code using the build.xml file and is intended only for developers who extend XTT. The build.xml file is used by developers to compile and package XTT. It is used during build and release to compile and package code and put it onto a build server, that is, fondue.len.tantau.com. XTT developers should use the pared down version of build.xml, as the build.xml checkout from CVS does several things, such as FTP the newly-built package to a particular computer, send emails that the package is completed and at a particular place, and so on. For development purposes, the pared down version of build.xml MUST be used. It is recommended that you do not make a build of XTT source code using the checkout version of build.xml. Instead you should use the pared down version of build.xml.

Note: You should check out the source code from the CVS repository using your credentials.

For an example of the pared down build.xml file, refer to Appendix A.

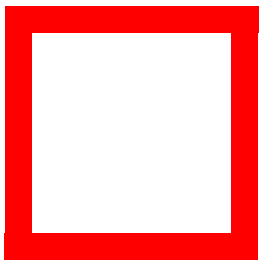


Appendix A

```
- <project name="xtt" default="jar" basedir=".">
  <property name="dest" value="dest/" />
  <property name="src" value="com/mobilgw/xtt" />
  <property name="src.gui" value="com/mobilgw/xtt/gui" />
  <property name="src.jacorb" value="org/jacorb/naming" />
  <property name="src.jar" value="xtt.jar" />
- <target name="compile" depends="prepare" description="Compile all the
java code">
- <javac destdir="${dest}" debug="on">
  <classpath refid="classpath" />
  <src path="${src}" />
  <src path="${src.gui}" />
  <src path="${src.jacorb}" />
  </javac>
  </target>
- <target name="jar" depends="compile" description="Creates the xtt.jar
file">
  <property file="xtt.buildnumber" />
- <tstamp>
  <format property="build.time" pattern="EEE, d MMM yyyy HH:mm:ss z"
locale="en,CH" />
  </tstamp>
- <!--
  <manifest file="META-INF/MANIFEST.MF" mode="update">
            <attribute name="Class-Path" value="${manifest.classpath}"/>
            <attribute name="XTT-Build" value="${build.number}"/>
            <attribute name="XTT-BuildTimeStamp" value="${build.time}"/>
        </manifest>
```

```
  -->
  <echo message="Build Number: ${build.number}" />
  <jar jarfile="${src.jar}" basedir="${dest}"
manifest="META-INF/MANIFEST.MF" excludes="" />
  <copy file="${src.jar}" todir="lib/"/>
  </target>
- <target name="javadoc" depends="compile" description="Generate the java
doc">
- <javadoc destdir="doc" verbose="false" classpathref="classpath">
  <link href="http://www.jdom.org/docs/apidocs/" />
  <link href="http://java.sun.com/j2se/1.5.0/docs/api/" />
- <packageset dir="." defaultexcludes="yes">
  <include name="com/mobilgw/xtt/**" />
  </packageset>
  </javadoc>
  </target>
- <target name="clean" description="Clean up">
- <delete includeEmptyDirs="true" failonerror="false">
  <fileset dir="." includes="${src.jar}" />
  <fileset dir="${dest}" />
  <fileset dir="./doc" includes="**/*" />
  </delete>
  </target>
- <target name="prepare" depends="clean" description="Prepare the build
directory">
- <!--  Create classes and generated directories if they do not exist
  -->
  <mkdir dir="${dest}" />
  </target>
- <path id="classpath">
- <fileset dir="lib">
  <include name="**/*.jar" />
  </fileset>
  </path>
- <pathconvert property="manifest.classpath" pathsep="">
- <mapper>
- <chainedmapper>
  <flattenmapper />
- <!-- globmapper from="*" to="lib/*" /
```

```
-->
</chainedmapper>
</mapper>
<path refid="classpath" />
</pathconvert>
</project>
```



Index