Updater

Software Requirements Specification

Interactive Laboratories

Updater

Change Log

Version	Author	Changes
0.0.0.0	Rico	Initial version
0.0.0.1	Rico	Added requirements 26 and 27
0.0.0.2	Rico	Added requirement 28

Updater

Table of Contents

Updater

Preface

Within this document, the functional requirement effort will be described and detailed. No implementation-specific details will be described other than from a conceptual viewpoint. No APIs or other interfaces shall be described herein other than a candidate list of names for

Updater is a console/terminal application to download new versions of files from an FTP site with no user interaction when the program is running.

Updater

Introduction

Project Scope

Updater can accompany an existing project or be distributed as a stand-alone program which can download any required content for a project. The latter is useful for minimising the download time for a project and is applicable to any project as it is a stand-alone program. *Updater* will run on Windows, Linux, SunOS, Xbox, and Pandora. SunOS and Xbox are not targeted for the initial release.

Updater

Overview

Product Perspective

Updater is a program for updating files for a project. It is a small program (<512KiB) intended to autonomously obtain files from an FTP server which follows a certain directory layout. It is written in C++ and works on the Windows, Linux, and Pandora platforms. There is no GUI for the program as it is unnecessary for the user to interact with the program once it is started.

Product Features

A summary of the product's features follows:
- Obtains new files from an FTP site
- Autonomous once started
- Use of a configuration data file to specify the following
  - Server address
  - Build Type
  - Platform
  - Project
- Accept configuration data on the command-line

User Classes and Characteristics

This program is intended for use by three user classes:
- Developers who are deploying various builds of their software for various testing purposes and do not wish to create a separate update server and handle the esoterica surrounding that implementation
- Testers who are either dedicated or casual software testers that wish to use the latest version of a developer's software
- Casual Users who want to get the latest release build of software

Operating Environment

Updater can be used on Windows, Linux, and Pandora platforms without modification or emulation.

Assumptions and Dependencies

A network connection is required of the user, while an FTP site correctly configured to the format expected by *Updater* is required for a developer's project. NCURSES is required for the Linux version, while PDCURSES is required for the Windows version.

Updater

Specific Requirements

External Interface Requirements

User Interfaces

Current Status and Download List Display

```
    Server Connection Status
     Current File Operation
    Total Time Running
    Running File Download Size
```

```
     File                Progress                  D/L Rate  Complete/Size    ETA
  Download Item 0    [------------]----------------  50%   100KiB/s  30MiB/60MiB   5:00
```

The download list will extend as far as the number of rows of the current terminal/console or until there are no more items in the queue.

At any time, the user can press the 'q' key on their keyboard to cancel the operation and exit the program. There will be a prompt to verify that the user wants to perform this action.

Hardware Interfaces

Updater requires a network connection with access to an FTP server and a keyboard to function.

Software Interfaces

A Linux, Windows, or Pandora operating system are required for their respective versions of *Updater* to run. Windows 2000 and greater are supported, as is Linux 2.4 and greater and the latest version of the Pandora operating system.

Communications Interfaces

As previously stated, a network connection with LAN, MAN, or WAN access is required for *Updater* to be useful.

Updater

Functional Requirements

List of Functional Requirements

<Req_1> - Software shall load a configuration state from a file.

<Req_2> - Software shall have the capability to establish a connection to the specified FTP site.

<Req_3> - Software shall implement an abort option to stop its actions and exit.

<Req_4> - Software shall prompt the user to confirm exit action.

<Req_5> - Software shall provide a command-line interface to define the configuration file to use, the FTP site to use, the project to use, the platform to use, the build type to use, and whether the software shall output a list of files downloaded and where if possible.

<Req_6> - Software shall display a list of files currently downloading with the file name, progress bar, per cent complete, estimated time to download file, download rate, and a split complete/size display of the progress.

<Req_7> - Software shall have the capability to query an FTP site for the files it hosts.

<Req_8> - Software shall have the capability to download a file from an FTP site.

<Req_9> - Software shall verify the MD5 hash for the current file downloaded matches the MD5 hash provided by the server.

<Req_10> - Software shall not exceed 1MiB in size.

<Req_11> - Software shall be able to run autonomously through to completion after it is started.

<Req_12> - Software shall include an example configuration file.

<Req_13> - Software shall output a list of possible commands if the user provides a help command to the program

<Req_14> - Software shall output a default configuration file if one is not present and there are no commands provided to the program.

<Req_15> - Software shall include a user's guide.

<Req_16> - Software shall take no longer than three seconds to initialise

<Req_17> - Software shall output a list of downloaded files and their MD5 hashes to a log file

<Req_18> - Software shall download files for the platform it is running on, unless a configuration option is used to alter the platform option.

<Req_19> - Software shall download the first build type it encounters, unless a configuration option is used to alter the build type option.

<Req_20> - Software shall download the first project it encounters, unless a configuration option is used to alter the project option.

<Req_21> - Software shall use a standard anonymous FTP account to connect to the specified FTP site.

<Req_22> - Software shall obey the directory layout the FTP site as proposed in the Design Notes document

<Req_23> - Software shall have the ability to generate an MD5 checksum for binary data

<Req_24> - Software shall be able to send an FTP command to a designated FTP site

<Req_25> - Software shall have the capability to receive data from an FTP site

<Req_26> - Software shall connect to an FTP site with an active connection using a specified port range

<Req_27> - Software shall connect to an FTP site with a passive connection

<Req_28> - Software shall utilise a CURSES-based library for displaying any graphical information

Functional Requirements By Component

Module 1 – Connect to FTP site
Requirements 1, 2, 5, 21, 26, 27, 28

Module 2 – List files from FTP site
Requirements 7, 22, 24

Module 3 – Download files from FTP site
Requirements 6, 8, 18, 19, 20, 25

Module 4 – Verification of file MD5 hashes
Requirements 9, 17, 23

Module 5 – Finishing touches
Requirements 3, 4, 11, 13, 14

Non-functional Requirements
Requirements 10, 12, 15, 16

Performance Requirements

There are no hard performance requirements.