

Open Print Assist Project Plan

openprintassist.sourceforge.net

Project Admin: Michael Dawson (mmiissp@users.sourceforge.net)

1. Introduction

1.1 Summary

The Open Print Assist (OPA) project will develop a print management application that is highly flexible yet powerful. It will provide services to the instant print market such as customer, job and stock management. It will be based on a generic Java application framework for portability, maintainability and code re-use so as to reduce development costs and increase business productivity. At this stage the project is expected to take about 6 months.

1.2 Deliverables

In order to build a flexible application and to promote code re-use the OPA project will:

- Develop a generic application framework that is based on a modular architecture. This will provide core services that are common to most GUI client applications such as window management, options management and a directory service.
Duration: approx. 3 months
Success factor: a reliable and stable framework
- Develop user modules for use in the base framework. These modules will provide the print management functionality as noted above.
Duration: approx. 3 months
Success factor: core print management functions available, and stable

1.3 References

OPA Project Definition Statement: OPA-PDS.pdf
OPA Kernel Architecture Proposal

1.4 Contact Information

The project homepage is <http://openprintassist.sourceforge.net>. It maintains current contact details for the development team.

1.5 Definitions

Compile-Time – The time when a computer program is compiled. After the computer code is written it must be 'compiled' to generate instructions a computer can understand.

Deliverable – Something that is provided by a project. A deliverable(s) is the output of a project.

Kernel – The heart of a program. A kernel provides absolutely essential services to other parts of the program. Other parts access kernel services by method or function calls. The OPAKernel is a 'container' for the OPA manager objects.

Gantt Chart – A graphical tool for displaying project schedules and relationships between different project tasks. Tasks are usually shown as coloured bars on the Gantt chart.

Module – A part of a program that can be easily added, particularly at run-time (not compile time). By adding parts when a program starts it is much easier to develop them separately. Using the different parts

depends on them implementing a WELL DEFINED interface that all parts of the program know about. OPA Uses interfaces for the manager APIs.

OPA - Open Print Assist

PDS – see Project Definition Statement.

Project Definition Statement – A short description of a project. A PDS is usually prepared before the project plan to give people interested in a project important information such as what the project wants to do, why and how it wants to do it.

Run-Time – The time when a program starts after a user command. See compile time.

SDK – Software Development Kit

Software Development Kit – A collection of programs and files used in the development of software. Also the name given by Sun Microsystems to the collection used for developing Java programs.

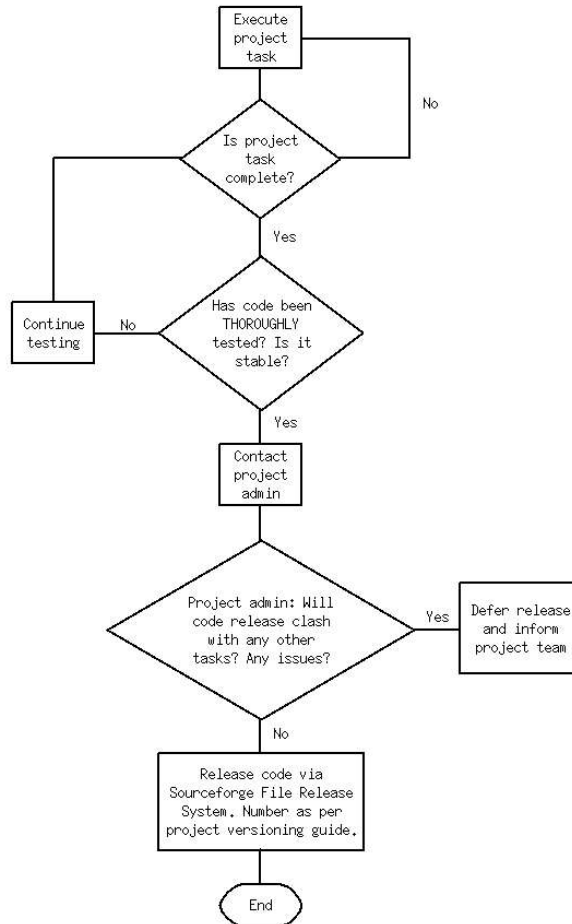
WBS – see Work Breakdown Structure

Work BreakDown Structure – An overview, usually a gantt chart or similar, of the tasks that make up a project. The WBS should show ALL tasks included in the project and possibly planned start/finish dates, resources required and people assigned to the tasks.

2. Organisation

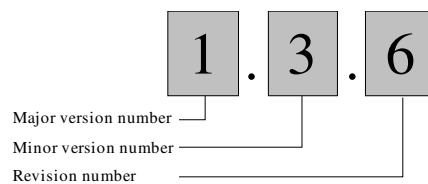
2.1 Public Sourcecode Release

Active development will use the *CVS repository services* hosted by Sourceforge. This is considered the 'working set' of project development code. This code is not suitable for general use. People wishing to use/run the OPA framework should download the public package. Public package releases of the sourcecode will be made available using the *File Release System* hosted by Sourceforge. The following illustrates the public release process:



2.2 Code Versioning System

OPA will identify source code by a three unit versioning system. The following demonstrates how to evaluate the release code:



Major versions are releases that provide a large change in the design of the system.
Minor versions are releases that provide a small change in the system.
Revisions are releases that update bugs or faulty code. They do not provide new functionality.

2.3 Task Responsibility

Throughout the project there may be many people in active development. In order to minimise duplicated efforts and repeated work, each task will be allocated to only a SINGLE DEVELOPER. Many developers may be active on the task however no more than one will be responsible to the project team. The allocated/available tasks will be viewable on the Sourceforge homepage.

2.4 CVS Write Access

The OPA project uses the CVS services available on Sourceforge. The sourcecode is freely available by anonymous pserver access. Write access to the source tree will be restricted in the early stages of the project. CVS writes will be achieved by:

- Verifying the accuracy of the code
- Forwarding the code to the project admin
- Project admin tests code again then writes to the CVS repository.

2.5 Project Licensing

The OPA project uses the following BSD license. This license MUST appear in ALL SOURCECODE FILES for the project and in a 'LICENSE' text file in the root distribution directory whether it includes source code or binaries. The license DOES NOT APPLY to non-OPA user supplied modules that plug into the OPAKernel.

Copyright (c) 2003, Open Print Assist (openprintassist.sourceforge.net)
All rights reserved.

Original design and code Copyright (c) 2003, Michael Dawson
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notices, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notices, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the names of MICHAEL DAWSON, OPEN PRINT ASSIST (OPA) nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR

OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

2.6 Development Life Cycle

The OPA project will use an iterative approach to development. Activities will be organised in small increments in order to minimise risk and increase the file release rate of the project:

- Short cycles of 9 days will be used
- A stable file release is desirable at the end of each cycle
- Project tasks should be able to be completed by a single developer, including testing and packaging within a single cycle.

See the project schedule section.

2.7 Project Plan Changes

Changes to the project plan may be made by the project admin. In the case of a change to the project plan all developers must be notified one (1) month in advance. All registered developers are free to suggest changes to the project, its plan or organisation via the non-public developer list on Sourceforge.

3. Managerial Process

3.1 Weekly updates

On the Friday of each week, a written update should be provided by all team members to the project developers mailing list. A simple text mail is sufficient. The update must however include the following:

- contact details
- sub-task being discussed
- summary of the progress
- any problems experienced or foreseen
- anything unusual that happened during the week that may affect the project

3.2 Monthly project report

By the 5th of each month, the project admin will release a formal status report onto Sourceforge and the project homepage for public viewing. This will include a summary of the projects progress, an updated project schedule and any other issues relevant to the project.

3.3 Staffing

OPA will seek to have 5~6 active developers involved in the project at any one time. Ideally, member developers will be able to contribute 10~15 hrs to the project per week and be contactable by email at least 3 times a week. The following are desirable for OPA developers:

Java development

- Knowledge of basic package, class and method rules.
- Familiarity with a Java integrated development environment (IDE), ideally NetBeans IDE.
- Familiarity with CVS services.

Analysis and Design

- Familiarity with UML diagrams including class, interaction and state diagrams
- Interest in planning and designing applications

Personal Qualities

- Self-motivated and able to contribute regularly
- Interested in working on a project with different types of people
- Keen to learn and to teach others who want to learn
- Enjoy developing!

4. Technical Processes

4.1 Code Format

The OPA Project will use the following code format for ALL Java sourcecode. NB Instructions in RED SHOULD NOT be included in source code:

```
/**
 * Class/Interface Javadoc comments go here. NB Javadoc (/**...*/) comments.
 */

/*
 * OPA licence notice goes here. NB simple (/*...*/) comments.
 */

modifiers ClassName(params) extends/implements statements {

    // Public class variables (section header)

    one line space...then the variables
    include 'help' comments (//...) to explain variables if necessary

    // Private class variables (section header)

    one line space...then variables here

    // Public instance variables (section header)

    one line space...then variables here

    // Private instance variables (section header)

    one line space...the variables here

    // Constructors (section header)

    /** All methods (including constructors) must have Javadoc comments.
     * one or two lines is enough. Also the params and return values.
     */
    modifiers returnValue constructorName(params) {
        // code starts here. Use 4 spaces (no tabs)
        ...more Java statements
    }

    // Public class methods (section header)

    /** All methods (including constructors) must have Javadoc comments.
     * one or two lines is enough. Also the params and return values.
     */
    modifiers returnValue someClassMethod(params) {
        // Java statements start on this line
        // an example if block
        if(needToUseAnIfBlock) {
            // four spaces indent
            someMethod();
        }
        else {
            someOtherMethod();
        }
    }
}
```

```

// Private class methods (section header)

/** All methods (including constructors) must have Javadoc comments.
 * one or two lines is enough. Also the params and return values.
 */
modifiers returnValue somePrivateClassMethod(params) {
    // Java statements start on this line
}

// Public instance methods (section header)

/** All methods (including constructors) must have Javadoc comments.
 * one or two lines is enough. Also the params and return values.
 */
modifiers returnValue somePublicInstanceMethod(params) {
    // Java statements start on this line
}

// Private instance methods (section header)

/** All methods (including constructors) must have Javadoc comments.
 * one or two lines is enough. Also the params and return values.
 */
modifiers returnValue somePrivateInstanceMethod(params) {
    // Java statements start on this line
}

```

4.2 Development Tools

Code and Libraries

The OPA project uses Java for development. The project will use the official SDK available from Sun Microsystems 1.4.2. This may change over the life of the project. In the event that such a change is necessary all project members will be informed one (1) month in advance.

IDE

The official Integrated Development environment for the OPA project is the NetBeans IDE 3.5.*. It is available free of charge at www.netbeans.org. '.fpm' files for use in the IDE GUI builder will be made available throughout the life of the project.

Groupware

The OPA project uses the freely available open source development facilities at sourceforge.net. This site provides task management, communication and web services to the OPA project.

5. Schedule

5.1 Project Gantt Chart

The project will follow an iterative, cycle based approach. Tasks have not yet been decided and so do not appear in the Gantt chart.

