

# Developing Paper Products

---

*Livescribe™ Platform SDK*  
*Version 1.0.1*

## Copyright and Trademarks

LIVESCRIBE, PULSE, and PAPER REPLAY are trademarks or registered trademarks of Livescribe, Inc. Anoto is a trademark of Anoto Group AB. Microsoft and Microsoft Word are either registered trademarks or trademarks of Microsoft Corporation in the United States or other countries. MyScript is a registered trademark of Vision Objects, Inc. All other brand and product names are trademarks of their respective owners.

Copyright © 2007-2009 Livescribe, Inc. All rights reserved.

DevPaperProducts-1.0.1-SDK-1.0.1-REV-A

8/31/2009 3:56 PM

Preface .....	7
About this Document .....	7
About the Platform SDK Documentation Set .....	7
End-User Documentation .....	8
Introduction .....	9
About Paper Products .....	9
Terms and Definitions .....	11
Acceptable Smartpen Applications .....	11
Creating Paper Product Applications .....	11
Phase 1: Design and Create Page Images .....	12
Phase 2: Create a Penlet Project .....	13
Phase 3: Create a Paper Product Project .....	16
Set Up the Paper Product Project .....	16
Phase 4: Design the Paper Product .....	18
Add Images .....	19
Link Paper Products to the Penlets .....	20
Create Active Regions (Shaping) .....	21
Auto-generate Event Code for Active Regions .....	23
Add Standard Livescribe Controls .....	24

## *Developing Paper Products*

Phase 5: Complete the Penlet Programming .....	26
Phase 6: Test the Paper Product and Penlet .....	26
Phase 7: Request and Apply an Official Dot Pattern .....	28
Phase 8: Retest your Smartpen Application with Official Dot Pattern .....	30
Phase 9: Submit Your Smartpen Application to Livescribe.....	30
Other Paper Designer Tasks .....	31
Storing and Displaying Strokes in Livescribe Desktop .....	31
Creating Multi-Page Note Pads .....	31
Standard Livescribe Controls.....	32
Paper Replay Controls.....	33
Smartpen Settings .....	33
System-Level Controls.....	34
Common Keyboard and Mouse Shortcuts .....	35
Tools for Image Creation and Printing .....	36
Layout and Creation of Images .....	36
Conversion of Images to EPS .....	37
Conversion from EPS to PDF for Printing.....	37
Printer Settings.....	38
Setting Output Generation Properties.....	39
Pattern Settings .....	40

*Developing Paper Products*

Color Settings .....41

Miscellaneous Settings.....42

Layout Settings .....42

Recommended Printers .....43

Other Printer Models .....44

The Livescribe Pattern Server.....44

    Pattern License.....44

    User Accounts .....44

    Pattern Credits .....45

    Additional Pattern Server Information .....45

AFD Document Format .....45

    AFD Technical Details .....46

    Templates and Stride.....47

    Pattern Licenses .....47

    AFD Copies .....48

    Locked Property.....48

    AFD for the Smartpen.....49

    Expanded AFDs.....49

    Document GUID.....49

    Publish Document .....50

## *Developing Paper Products*

Printing Basics .....	50
Printing Terminology .....	50
Print Processes .....	51
Offset Printing .....	52
Toner Based Electrostatic Printing (Laser) .....	52
Inkjet Printing .....	53
Vector Based Formats.....	54
Color Theory and Smartpen Vision.....	55
FAQ and Troubleshooting.....	56

# Preface

## About this Document

This document, *Developing Paper Products*, introduces the Livescribe Paper Designer and processes for designing paper products for the Pulse smartpen.

## About the Platform SDK Documentation Set

The Livescribe Platform SDK documentation set includes the following documents.

Introductory Documentation:

- *Read Me First*: introduces the components of the Livescribe Platform, lists system requirements, and provides download instructions.
- *Livescribe Developer License Agreement*: The Livescribe license agreement for third-party developers, in TXT and PDF formats. All developers must accept this agreement before downloading the Platform SDK.
- *Introduction to the Livescribe Platform*: describes the Livescribe Platform, the Pulse smartpen and Livescribe dot paper, and Pulse smartpen applications.
- *README.txt*: provides a quick overview of how to set up your development environment. It is aimed at experienced developers and presents information with minimal context. For full instructions, see *Getting Started with the Livescribe Platform SDK* instead.
- *Getting Started with Livescribe Platform SDK*. describes the contents of the SDK, how to set up your IDE (Integrated Development Environment), and how to create some simple Hello World penlets. It also provides a reference for the wizards, views, and dialogs of the Penlet perspective within the IDE.
- *Eclipse Help*: basic coverage of the Livescribe IDE plug-in with Eclipse.
- *Release Notes*: covers late-breaking information, fixed bugs, and known issues.
- *Glossary*: terms and definitions (located in *Getting Started with the Livescribe Platform SDK*).

## *Developing Paper Products*

Penlet Development Documentation:

- *Developing Penlets*: the primary programming guide for open paper (OP) and fixed print (FP) penlet development.
- *Livescribe Platform API javadoc*: API reference for packages, classes, and methods.
- *User Experience Guidelines*: presents user interface guidelines for creating penlets to ensure a consistent user experience across all smartpen applications.
- *Development Guidelines*: presents guidelines for creating penlets and paper products that conform to Livescribe standards and legal requirements.

Paper Product Development Documentation:

- *(This Document) Developing Paper Product*

## End-User Documentation

To learn about the Pulse Smartpen and Livescribe Desktop from a user's perspective, please consult their user guides. They are available at <http://www.livescribe.com>. Click on the Support button and locate the Documents list in the right-hand column. Links to the following documents in PDF format are available:

- *Getting Started*: introduces the Pulse Smartpen. A dot-enabled version of this document is included in the Pulse Smartpen box.
- *Pulse Smartpen User Manual*: explains Pulse Smartpen features and use.
- *Livescribe Desktop User Manuals for Mac OS X and Windows*: explain Livescribe Desktop features and use.



## Introduction

This document describes how to use the Livescribe Paper Designer and the Livescribe Eclipse IDE to design and create Livescribe dot paper for Livescribe smartpen Fixed Print (FP) applications. Using the Paper Designer, you can:

- design paper products that contain pre-defined active regions that penlets can respond to.
- print test pages for your application with a development test dot pattern
- request product dot pattern specifically allocated for your application
- produce an Encapsulated Postscript (EPS) file to print physical pages you design

For background information about Livescribe paper products and penlets, refer to *Introduction to the Livescribe Platform* and *Getting Started with the Livescribe Platform SDK*.

## About Paper Products

The most basic definition of a Pulse smartpen application is a *Paper Product and one or more associated (linked) Penlets*.

### Penlet + Paper Product = Pen Application



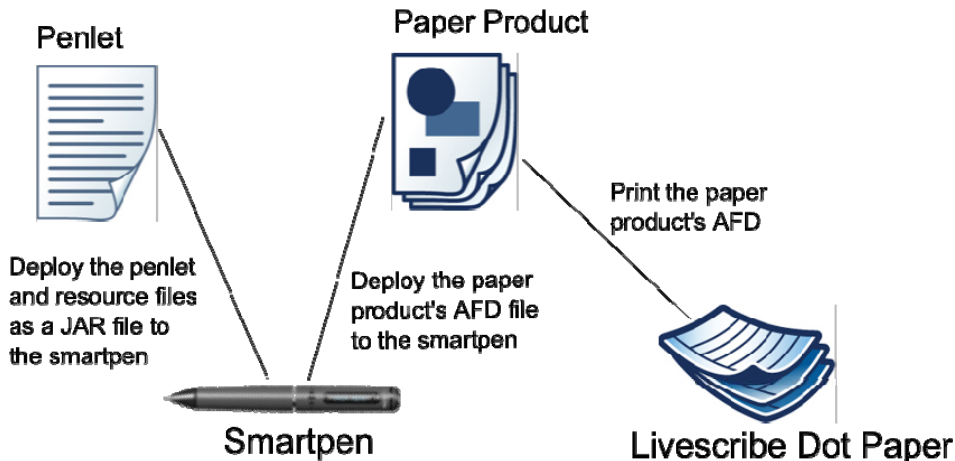
A *Paper Product* consists of:

- the physical dot paper a user interacts with using the Pulse smartpen
- an electronic file representation of the physical dot paper

The electronic file representation of a Paper Product is a proprietary container file called an Anoto Functionality Document (AFD). The AFD describes the Paper Product to Penlets and to other components in the Livescribe mobile computing platform. AFD files are installed on the Pulse smartpen along with Penlets that use them. This enables the Pulse smartpen to recognize and use the Paper Product.

## Developing Paper Products

As you know, a *Penlet* is a Java application developed to interact with specific Active Regions defined on a Paper Product. The active regions can be either pre-defined (static, Fixed Print) regions, or dynamic (established during the runtime interaction of a user with Open Paper). The Penlet handles smartpen events and performs actions dictated by the active regions.



The Livescribe Pulse smartpen can use two kinds of dot paper: Open Paper and Fixed Print. A printed page can contain Open Paper regions, Fixed Print regions, or both:

- **Open Paper (OP):** Open Paper refers to regions of a Livescribe dot page that various Pulse smartpen applications can claim at runtime. When a user runs a particular application and writes on those dots, the application can claim these dots.
- **Fixed Print (FP):** Fixed Print refers to regions of a Livescribe dot page that are defined by the developer when developing a Pulse smartpen application. No other application can claim them at run time. Printed graphics on the page identify these areas to the Pulse smartpen user as “paper controls” for operating the application.

Most of the paper products from Livescribe contain both Open Paper and Fixed Print regions. For instance, the pages of the Livescribe notebooks have Fixed Print controls at the bottom and Open Paper dots in the blank lines above them. These Fixed Print controls allow users to record, pause, and stop audio sessions while writing notes, access various points in the audio stream, manipulate playback speed, and adjust volume. The special Nav Plus control in the lower left-hand corner activates the main menu, visible on the Pulse smartpen's OLED display.



Open Paper areas (lined or not lined) above the Fixed Print controls can be used by any Open Paper application, such as Paper Replay to capture notes and audio, and Quick Commands, such as Quick Calc, where a user write math problems and have the Pulse smartpen calculate and display the answers.

## Terms and Definitions

There are new terms and definitions that relate specifically to creating Livescribe smartpen applications and paper products. This document uses many of these terms. Open Paper, Fixed Print, and Active Regions are among the most important. For a complete list of terms, see the *Glossary* chapter in *Getting Started with the Livescribe Platform SDK*.

## Acceptable Smartpen Applications

Before developing a smartpen application, review the guidelines and restrictions described in *Development Guidelines* and in the *Livescribe Developer License Agreement*. Pay particular attention to the guidelines for applications that use paper products because some of the restrictions can be subtle. If in doubt about a certain kind of penlet or paper product being acceptable to Livescribe, please contact your Livescribe Development Support representative.

## Creating Paper Product Applications

There are many possible workflows you can follow when developing smartpen applications. This section outlines the basic phases for developing applications that consist of paper products and their associated penlets. These kinds of applications

## *Developing Paper Products*

are generally Fixed Print (FP) applications. Each of these phases is explained in detail in subsequent sections of this document.

Phase 1: [Design and create page images \(background images and other page art\)](#).

Phase 2: [Create a penlet project](#).

Phase 3: [Create a paper product project](#).

Phase 4: [Design the paper product](#).

Phase 5: [Complete the penlet programming](#).

Phase 6: [Test the smartpen application](#).

Phase 7: [Request and apply an official dot pattern](#).

Phase 8: [Retest the smartpen application with official dot pattern](#).

Phase 9: [Submit Your Smartpen Application to Livescribe for approval](#).

## Phase 1: Design and Create Page Images

The first step in creating a paper product is to design its background and other page images. Often, this is the most distinctive aspect of a paper product to an end user. When designing your background image and other art, consider the functionality of your application and design:

- General background art (for example, lined, unlined, and so on)
- Art for custom controls and buttons needed by your application
- Art for Open Paper areas where users will write notes, capture handwriting, or claim dynamic active regions.
- Art for standard controls for Livescribe applications, such as Paper Replay, smartpen Settings, and the Nav Plus.

Livescribe does not provide tools to create artwork. Instead, you should use industry standard tools, like Adobe Illustrator, Indesign, and Photoshop to create your art. You must save the images in Encapsulated Postscript (EPS) format. The Livescribe Paper Designer requires that all images be in EPS format before they can be

imported and applied to a paper product. See [Tools for Image Creation and Printing](#) for details and suggestions.

## Phase 2: Create a Penlet Project

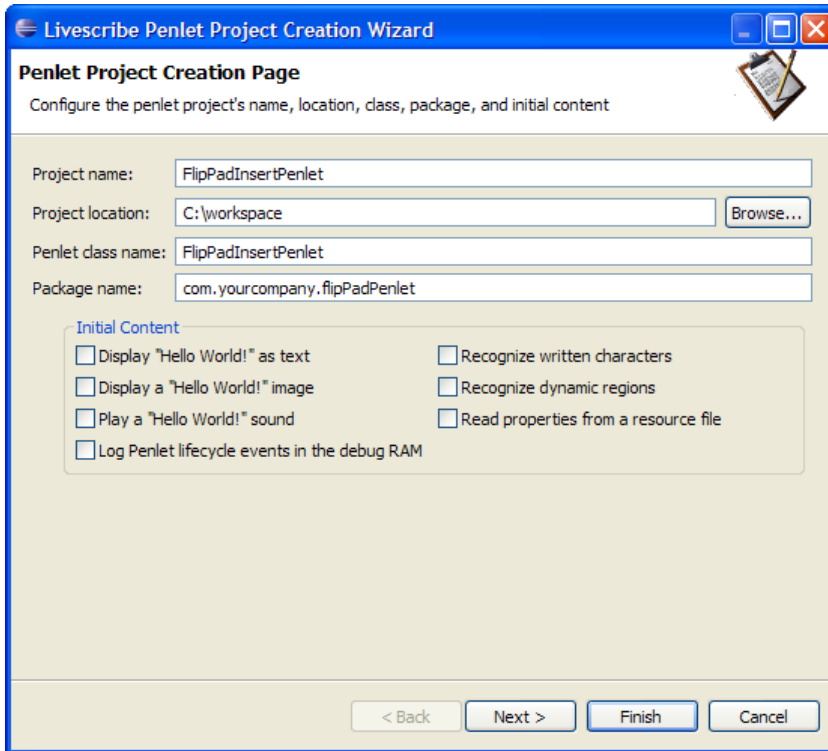
After creating your background and other art elements, you can begin either by creating a basic penlet project or by creating the paper product project. Either starting point is fine.

We suggest, however, that you first create the basic penlet project. This makes linking your paper product to your penlet a little more convenient. When you later create your paper product, you will need to associate (link) it to one or more penlets. At that point, it is easiest to have already started your penlet so you can then link to it. However, if you instead first create the paper product project, you can later create your penlet and link the paper product to the penlet.

At any point during the development cycle, you can switch back and forth between paper design and penlet development. You can explicitly switch between the Paper Product and Penlet perspectives, or simply double-click on the item in the Project Explorer to open the item in the main viewer.

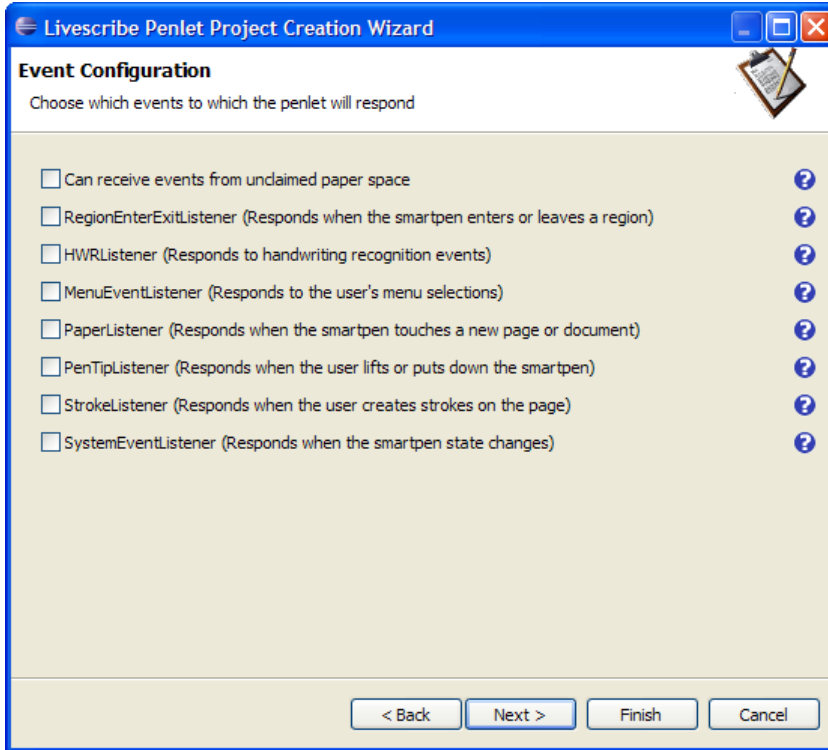
Below are the basic steps for creating a penlet project. For detailed instructions, refer to *Getting Started with the Livescribe Platform SDK* and *Developing Penlets*. Here are the basic steps:

1. Launch Eclipse.
2. Choose **File > New > Livescribe Penlet Project**. This starts the Livescribe Penlet Project Creation Wizard, and opens up a Penlet Perspective view in the IDE.
3. Follow the wizard dialogs to create the basic framework for the kind of penlet you want to interact with your paper product. The dialogs will help set up your basic penlet code



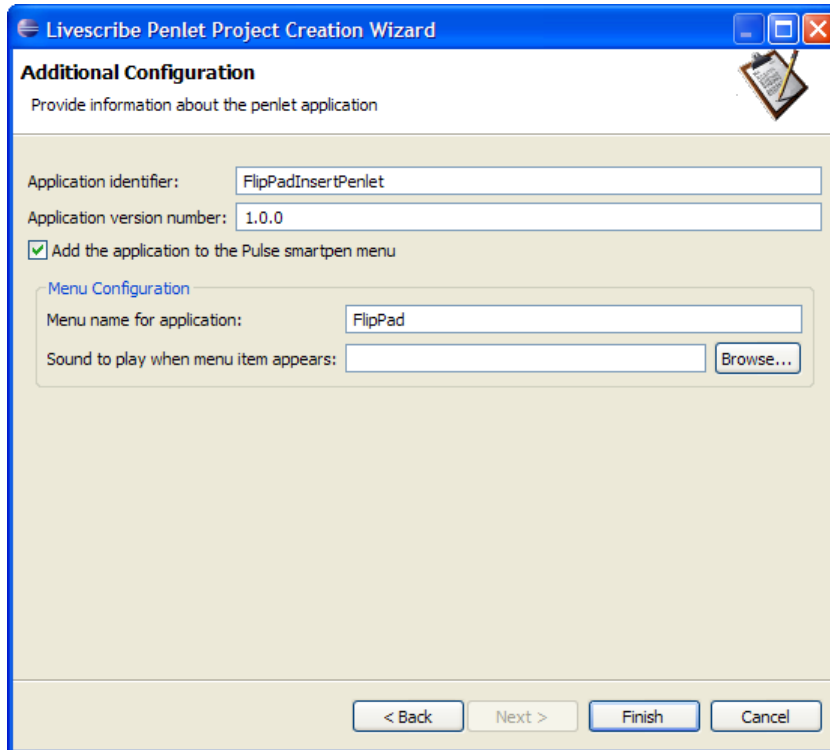
4. Specify the project name and location, the class name for the penlet, and the package name. In general, follow the standard naming conventions for Java. Be careful that the package name is unique in some way to avoid name collisions with other developers creating applications of the same name. You can generally accomplish this using a package name like `com.<your_company>.<your_project_name>`. For example, the Livescribe package name for Paper Replay is: `com.livescribe.paperreplay`.
5. Optionally, have the wizard add initial content to support the kinds of penlet you want to make. The wizard will create stub code for handwriting recognition, penlets that define dynamic regions at runtime, and those that read properties or other information from a resource file. You can also optionally have the penlet lifecycle logged in the smartpen's debug RAM. This is generally a good idea to do.
6. Select the smartpen events you want the penlet to respond to. The wizard will auto-generate code to respond to typical smartpen events. For example, a typical

application that responds to a user tapping on an active region requires



7. Click **Next**.
8. Provide additional configuration information, including a specific application identifier and version number. You can also select whether or not to include the penlet in the smartpen menu, specify the penlet menu name, and optionally play

a sound when the menu is invoked.



9. Click **Finish**.

## Phase 3: Create a Paper Product Project

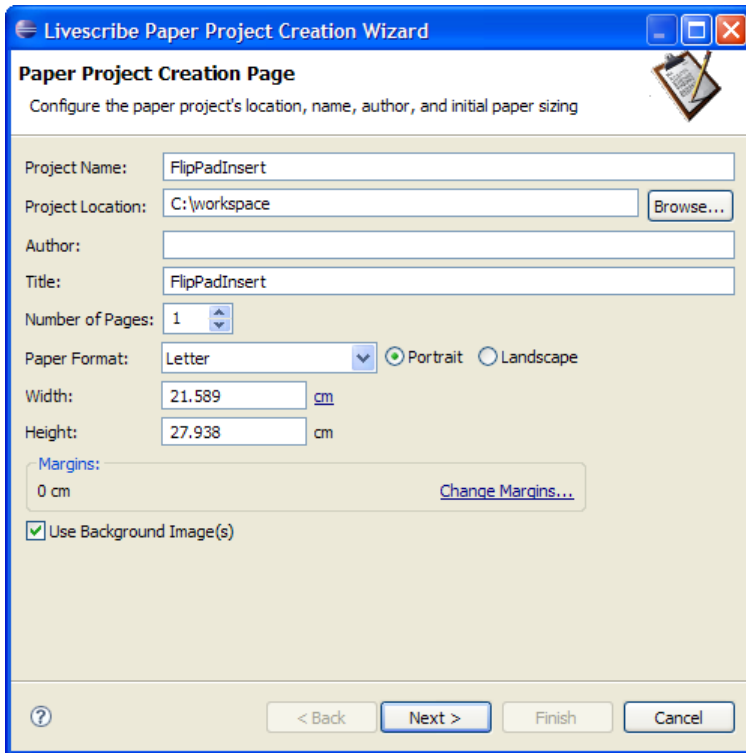
The easiest way to create a new paper project is to use the Livescribe Paper Project Creation Wizard.

### Set Up the Paper Product Project

1. Launch Eclipse.
2. Choose **File > New > Livescribe Paper Project**. This starts the Livescribe Paper Project Creation Wizard, and opens up a Paper Product Perspective view in the IDE.
3. Using the wizard, specify properties, such as the project name, author, project title, number of pages, page format, and background images. The wizard will create the electronic version of your paper product known as an AFD. Livescribe paper products are defined by an AFD file, which contains paper attributes, dot space, images, application linkages, and other relevant information. The wizard will create the main source version of the AFD (under the src folder) from which



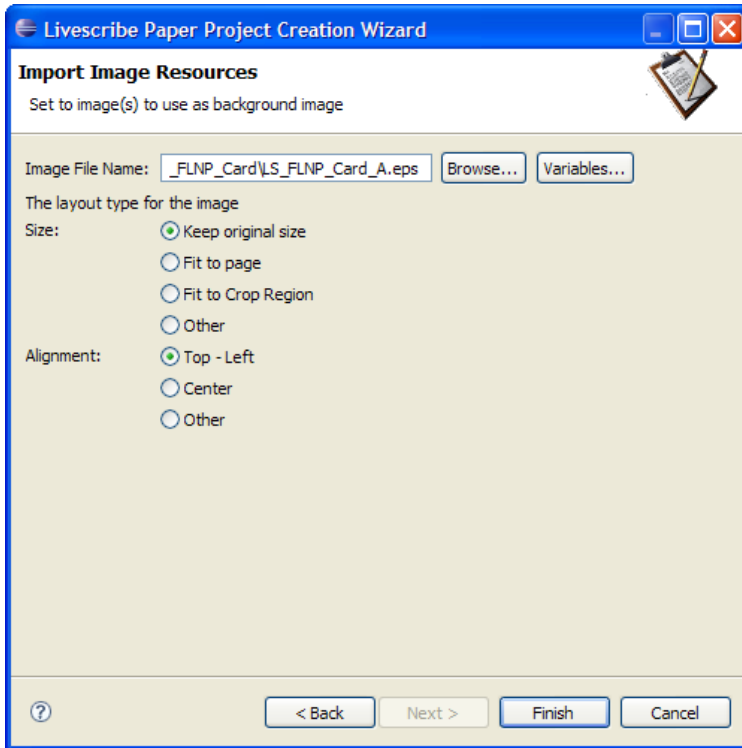
you can print, and a separate AFD file (under the dist folder) that will be deployed on the smartpen.



4. Specify the paper format using the **Paper Format** drop-down, or manually set the width, height and orientation directly. Note: Every location and size in the Paper Designer will be an integer number of an Anoto Unit (AU), roughly 677 dpi. If you enter an integer value in mm or inches it will be rounded the closest whole AU and then converted back.
5. Optionally, change the page margins by clicking the **Change Margins** link and enter the values you want for your paper product.
6. If you want to use an image as a background for your paper product, check **Use Background Image(s)**. In some circumstances, you may not want to use a background image, and instead compose your page art using separate images on the page. In most cases, however, you will want to specify a background image.
7. If you did not check **Use Background Image(s)**, click Finish to exit the wizard and finish creating your Paper Product.

To specify a background image, click **Next** to open the **Import Image Resources** dialog. In this dialog, specify which image to use. To use a single image as a background, select **Fit to Crop Region** for the size. The Crop Region is the rectangle that represents the viewable or finished paper product size. In general, this is smaller than or equal to the whole page (also called the Page

Bounding Box). See [Printing Basics](#).



8. Click **Browse** to locate the background image. **Note:** Currently, you must use a background image in encapsulated postscript (EPS) format. See [Design and Create Page Images](#).
9. When you have finished the wizard dialogs, click **Finish**. The wizard will then create and display your paper product and open the Eclipse perspective (windows and views) for Paper Editing. You can also manually select this perspective from **Window > Open Perspective > Paper Editing**.

## Phase 4: Design the Paper Product

After creating the paper product project with the wizard, you can then use the Paper Designer to complete your design. Tasks include:

- [adding images](#)
- [linking the paper product to penlets](#)
- [create active regions \(shaping\)](#)
- [auto-generate event code for active regions](#)

- [optionally, add standard Livescribe controls](#)

## Add Images

Add images as needed to your paper design, such as the background, buttons, or other artistic elements. In most circumstances, paper products include buttons and controls as part of the default background image. However, you have the option of adding other images to your page, or even compose your entire page design from separate images.

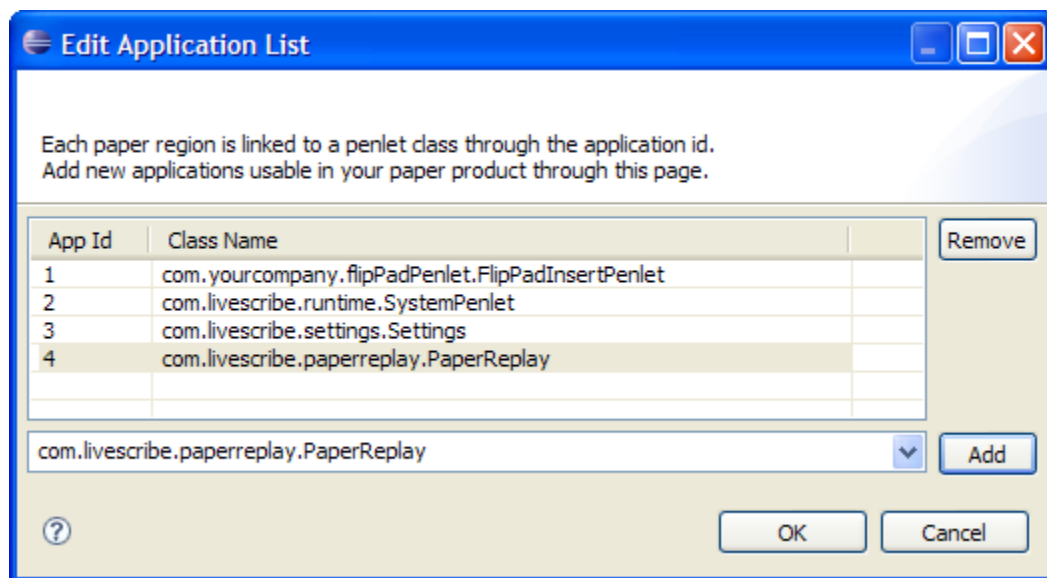
To insert a graphical element on your page, like an image for a button:

1. Select **Insert Image** under **Graphical Elements** in the palette.
2. Click in the editor canvas to place the image on your page.
3. In the image dialog, locate the image to insert. Note that currently, the image must be in EPS format. See [Tools for Image Creation and Printing](#).
4. After inserting the image, resize and move it by clicking and dragging it in the canvas, or edit the properties in the Property Inspector.
5. In the Property Inspector, name the image and either accept the default Graphic Id or specify a new one. Set other parameters as needed, including:
  - **Z-Order:** the depth relative order of the objects on the page. This option is available for Regions and Graphical Elements. The Z-order for Regions and Graphical elements is independent from each other. The Z-order of graphical elements decides the drawing order on printer and screen. A lower z-order is drawn on top of a higher one. For Regions, the z-order decides the order of intersection and a lower Z-order means that the shape is on top of the other shape.
  - **Graphics element is a user control:** This is a property of a Graphics object. When enabled, Livescribe Desktop will not display the image. For example, the Paper Replay controls on a notebook page are not displayed in Livescribe Desktop. What you see in Livescribe Desktop are similar page controls that are part of the desktop application, not the actual controls from the paper product.

## Link Paper Products to the Penlets

Associate the paper product to one or more penlets, including optionally standard Livescribe penlets (like Paper Replay) and system penlets known as syslets. See [Standard Livescribe Controls](#) for more information about standard controls.

1. In the project explorer, open the folder for your paper product.
2. Open the src folder.
3. Select the .afd file.
4. Select **File > Properties**.
5. Select the **Document** tab.
6. Click **Edit Application List**.



7. Type the application class name for the custom penlet you created in [Create a Penlet Project](#). Click **Add**.

**Tip:** When adding penlets to this list, you should add the penlet with the most Active Regions first. This is because the Paper Designer automatically assumes any Active Region you create is for the first application in the list. This makes it easier when setting the properties for your Active Regions because the regions will be automatically linked to this application.

8. Optionally, add standard Livescribe applications like Paper Replay: `com.livescribe.paperreplay.PaperReplay` in the bottom entry field. For information

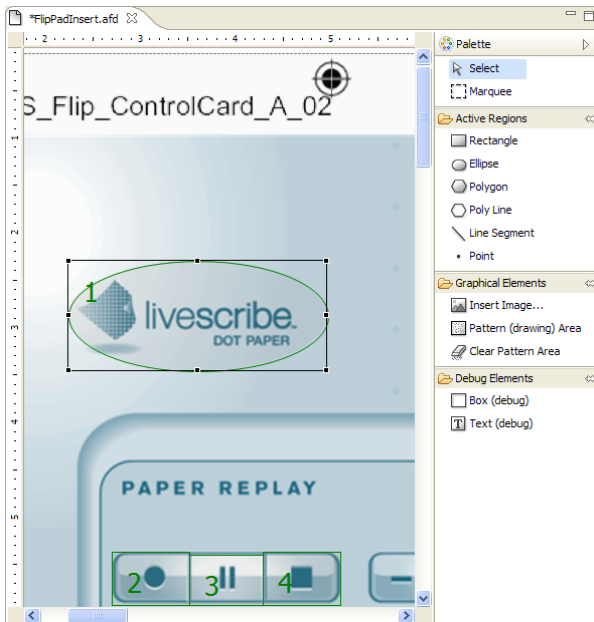
and package/class names of the standard Livescribe applications, see [Standard Livescribe Controls](#).

9. Click **Add** for each application you add.
10. Click **OK** when finished.

## Create Active Regions (Shaping)

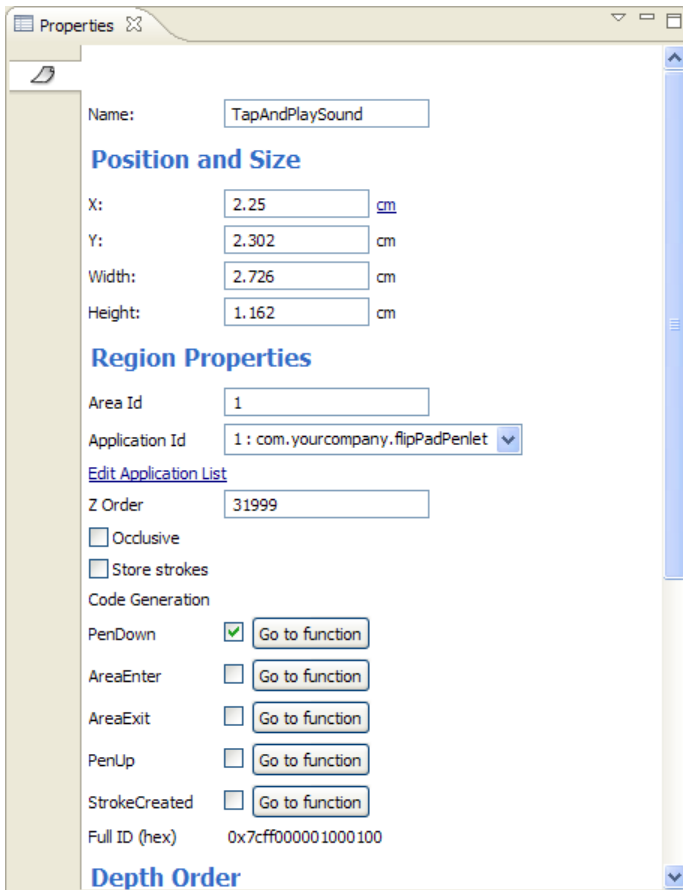
Use the Paper Designer graphical tools to draw active regions (controls, tap-and-play areas, and so on) that your penlet will respond to. This process is also known as *shaping*.

1. To create an active region on your page:
2. Select a shape (such as a Rectangle) from the **Active Regions** folder in the palette.
3. Click in the editor canvas to place the element on your page.
4. Resize and move the element by clicking and dragging them in the canvas, or edit the properties in the Property Inspector. Note that after creating an Active Region, you can save time by copying and pasting it within the page or to other pages.



5. In the Property Inspector, for each active region (shape), specify its name, Area Id, and the penlet that will respond to this region. The Area Id is what links specific regions on the page to the penlet. You can set it to any integer value as

long as it is unique for the penlet it is linked to.



6. Set other parameters as needed:

- **Z-Order:** the depth relative order of the objects on the page. This option is available for Regions and Graphical Elements. The Z-order for Regions and Graphical elements is independent from each other. The Z-order of graphical elements decides the drawing order on printer and screen. A lower z-order is drawn on top of a higher one. For Regions, the Z-order decides the order of intersection and a lower Z-order means that the shape is on top of the other shape
- **Occlusive:** When two Active Regions (Shapes) overlap, two events will be thrown in the Java API, one for each of the regions. Certain applications do not want this behavior. To get an event only for the closest one of the regions, set the occlusive bit. You can also use the occlusive bit for some more exotic things such as cutting holes in shapes and creating more complex shapes by overlapping.

- **Store Strokes:** Indicates if end-user smartpen strokes will be stored by the smartpen for this active region and displayed in Livescribe Desktop. In the Property Inspector, uncheck the Store strokes checkbox. See [Storing and Displaying Strokes in Livescribe Desktop](#).

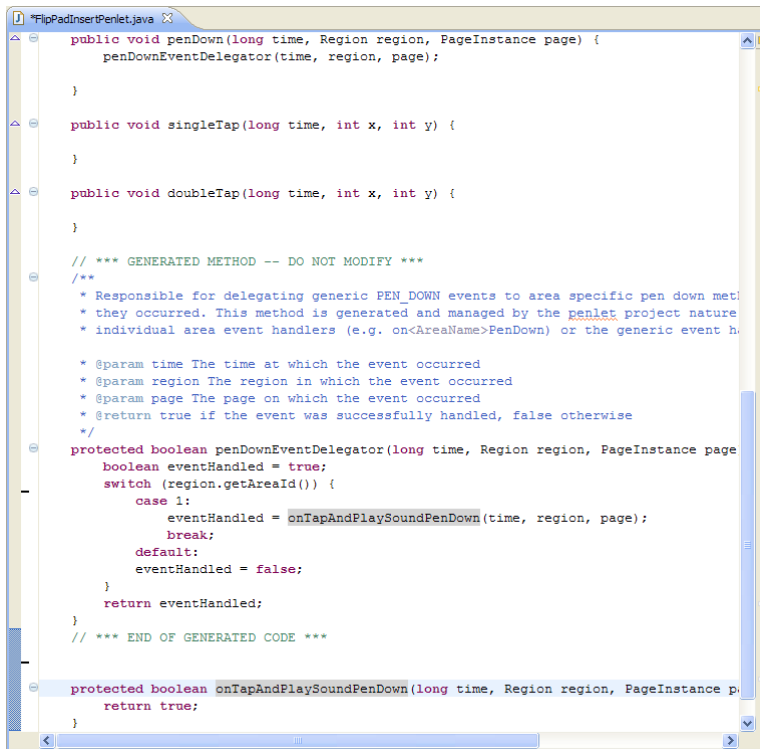
**Note:** When multiple active regions overlap, if any of them is set to not store strokes, no strokes are stored for any of the overlapped active regions.

## Auto-generate Event Code for Active Regions

For each custom active region used by your penlet, have the Paper Designer auto-generate event code. This process saves time when programming your penlet. To trigger auto-generation of event code for an Active Region:

1. Select the Active Region in the Outline palette.
2. In the Property Inspector, check the events you want to define – **PenDown**, **AreaEnter**, **AreaExit**, **PenUp**, and/or **StrokeCreated**. Checking these events automatically generates stub code in your penlet.
3. Next to each of the event names, click **Go to Function** to open the penlet for further editing.

For example, if you checked **PenDown** in the Property Inspector for a custom Active Region called TapandPlaySound, the Paper Designer will automatically generate a stub penlet method called onTapandPlaySoundPenDown to respond to a PenDown and other pen tip listener events on this Active Region. You can then use the Penlet editor to define what the behavior should be.



```
public void penDown(long time, Region region, PageInstance page) {
    penDownEventDelegator(time, region, page);
}

public void singleTap(long time, int x, int y) {
}

public void doubleTap(long time, int x, int y) {
}

// *** GENERATED METHOD -- DO NOT MODIFY ***
/**
 * Responsible for delegating generic PEN_DOWN events to area specific pen down methods
 * they occurred. This method is generated and managed by the penlet project nature
 * individual area event handlers (e.g. on<AreaName>PenDown) or the generic event handler
 */
 * @param time The time at which the event occurred
 * @param region The region in which the event occurred
 * @param page The page on which the event occurred
 * @return true if the event was successfully handled, false otherwise
 */
protected boolean penDownEventDelegator(long time, Region region, PageInstance page) {
    boolean eventHandled = true;
    switch (region.getAreaId()) {
        case 1:
            eventHandled = onTapAndPlaySoundPenDown(time, region, page);
            break;
        default:
            eventHandled = false;
    }
    return eventHandled;
}
// *** END OF GENERATED CODE ***

protected boolean onTapAndPlaySoundPenDown(long time, Region region, PageInstance page) {
    return true;
}
```

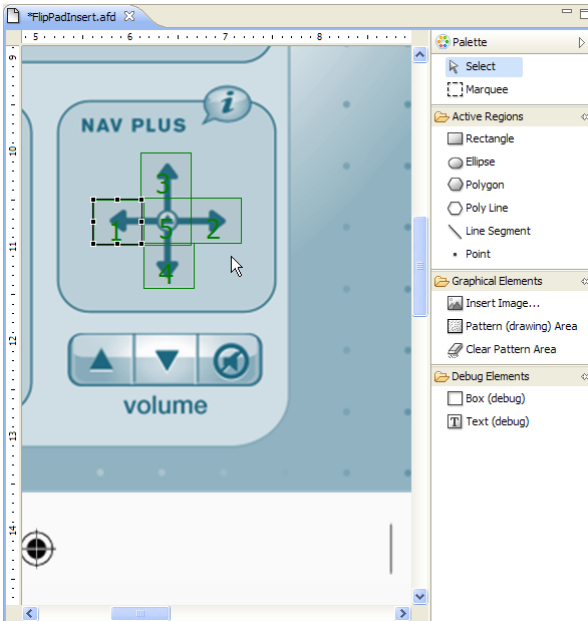
## Add Standard Livescribe Controls

Optionally, you can add standard Livescribe penlet and system controls to your paper product. For example, you will likely want to add a Nav Plus control to allow your users to access your application through the Main Menu on their smartpens. You also might want your paper product to control volume level and mute options, or enable Paper Replay controls for any Open Paper regions on your paper product.

1. Refer to [Standard Livescribe Controls](#) for a list of penlet and system classes and Area Ids you can add to your paper product.
2. In your paper product, insert an image to represent the control, or use an existing art element on your background image.
3. Draw active region or regions over the image for the control. For guidelines on shaping Livescribe controls, open one of the standard AFDs provided with the SDK. Note the size of the shaped regions. In particular, look at how the Nav Plus is shaped. There are five regions to shape for the center, and the left, right, top,



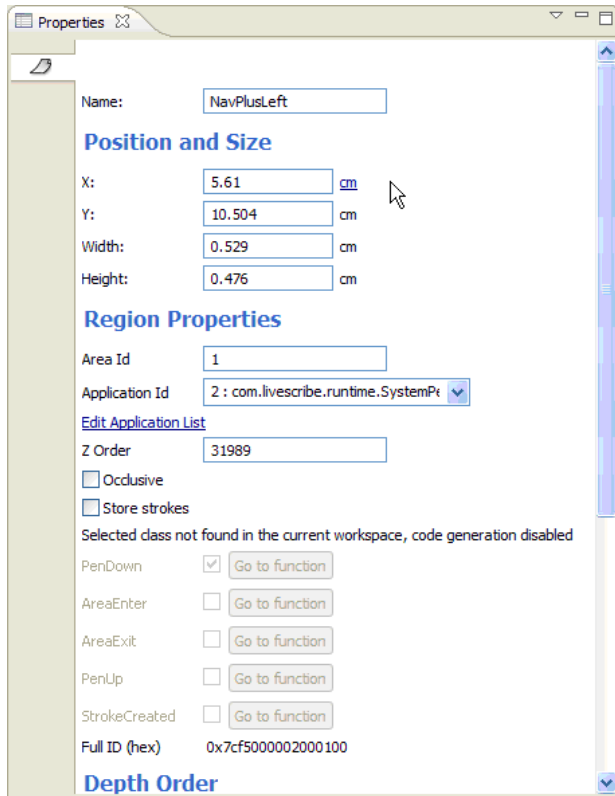
and bottom.



4. In the properties area, use the application dropdown to select the desired standard penlet or system class to associate with the active region. Use the class names specified in [Standard Livescribe Controls](#).

Note: If you have not added the penlet to your paper product, click **Edit Applications List** in the Property Inspector and type in the class name.

5. Set the Area Id of the active region to match the desired functionality of the penlet or system control.



## Phase 5: Complete the Penlet Programming

After completing your paper product design, return to the Penlet perspective to complete your penlet code. Refer to the Livescribe SDK javadoc, *Developing Penlets*, and *User Experience Guidelines* for information about how to design the best penlet for your application goals.

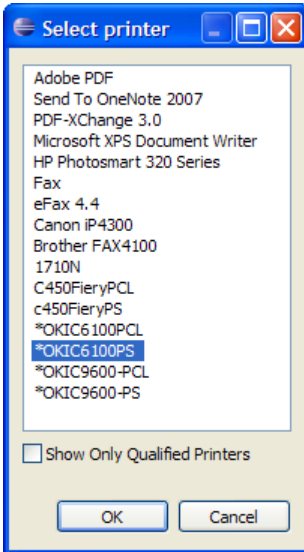
## Phase 6: Test the Paper Product and Penlet

1. Deploy the penlet to your smartpen by right-clicking the Penlet project and selecting **Deploy to Smartpen**.
2. From the Project Explorer, select the source AFD file in the src folder, and right-click to select **Deploy to Smartpen**.

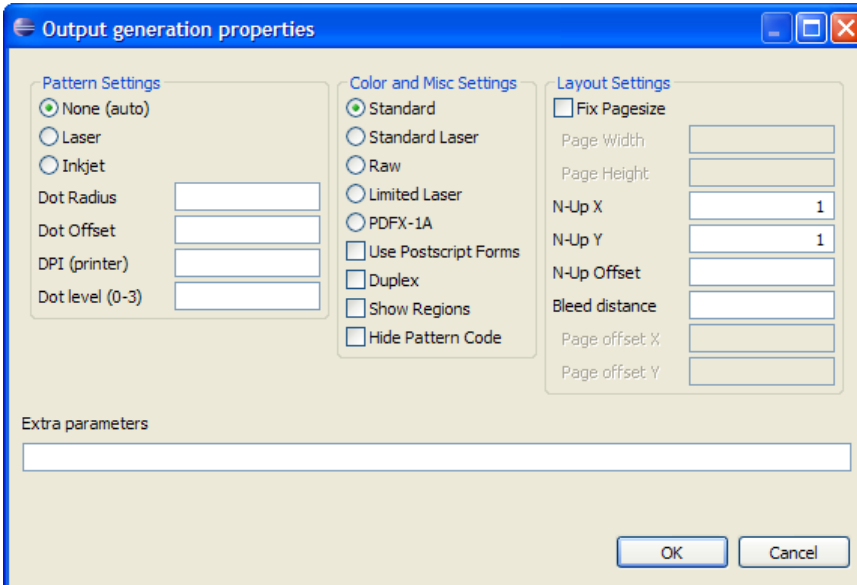
## Developing Paper Products

- Print your paper product using the AFD. Right-click the .afd file in the src folder and choose **Print**.
- Select the printer you want to use and click **OK**.

When printing, you should use a recommended printer. However, other printers can also work. See [Recommended Printers](#) for more information.



- In the next dialog, select the printing properties. See [Printer Settings](#) for details about which printer settings to use.



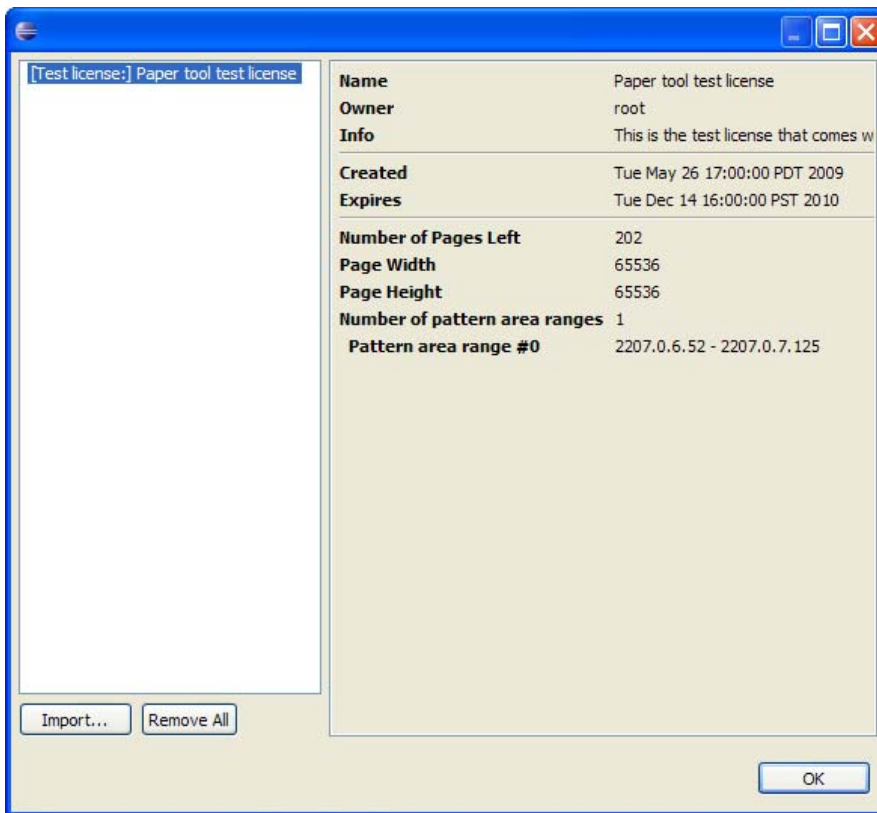
**Note:** When testing your paper product, you will use the default developer test pattern provided by the Paper Designer. When you are ready to publish the paper product for distribution, you will need to obtain a unique dot pattern license from

Livescribe using the Livescribe Pattern Server. For more information, see [The Livescribe Pattern Server](#).

6. After deploying your penlet, and deploying and printing your paper product, thoroughly test your application.

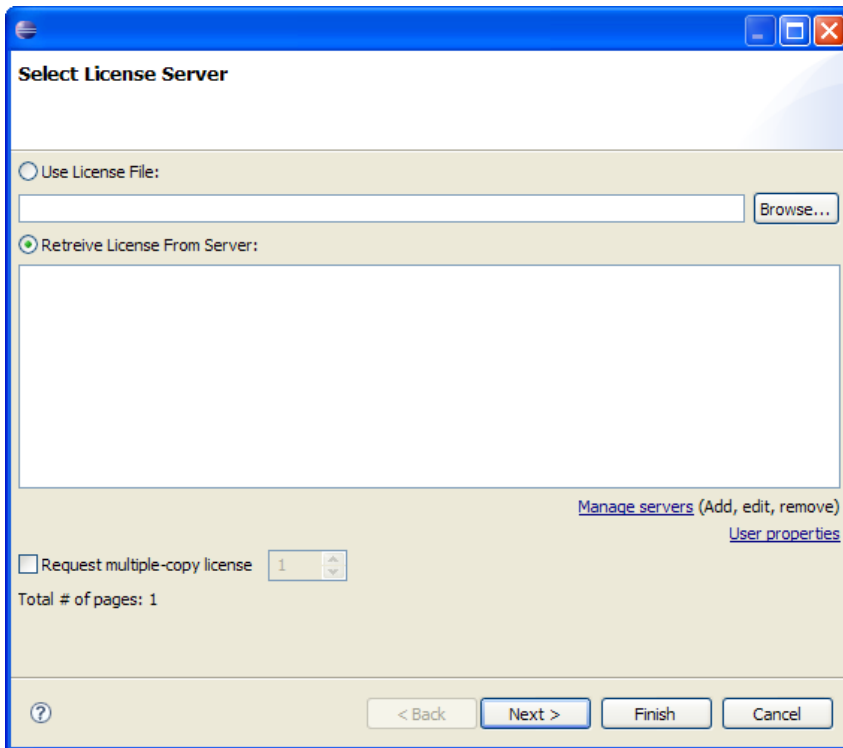
## Phase 7: Request and Apply an Official Dot Pattern

1. When your application is fully tested and ready for production, you will request a unique dot pattern for your paper product from Livescribe and apply the pattern to your paper product.
2. Choose **File > Manage License**.
3. In the License dialog, select **Import**.



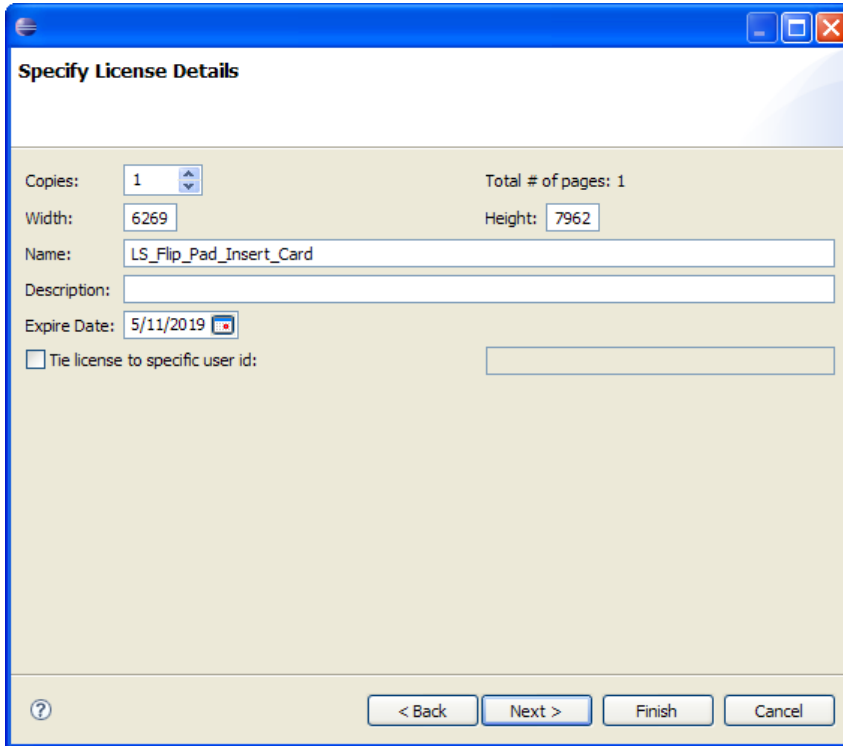
4. If you have received a license file from Livescribe, enter it in the **User License File field**. If not, Click **Retrieve License** from Server to enter a license server path and name.

5. Click **Manage Servers**.



6. In the **Manage License Servers** dialog, select **Livescribe License Server** and click **Add**.
7. In the **Livescribe Server Credentials** dialog, enter the URL and your username for the license server.
8. Click **OK**.

9. In the **Specify License Details** dialog, enter the number of license copies you need and the other license parameters as needed.



10. Click **Finish**.

## Phase 8: Retest your Smartpen Application with Official Dot Pattern

1. Reprint your paper product and redeploy the updated AFD to your smartpen.
2. Thoroughly test your application with the official, licensed pattern using your deployed penlet, AFD, and printed paper product.

## Phase 9: Submit Your Smartpen Application to Livescribe

After testing using the new dot pattern, submit your Paper Product and Penlet as a Smartpen Application to Livescribe for approval and distribution through the Livescribe Store. For now, please contact your Livescribe Developer Support representative about how to submit your application.

## Other Paper Designer Tasks

This section describes some other tasks and workflows you may want to use when designing your paper product.

### Storing and Displaying Strokes in Livescribe Desktop

You can specify if end user strokes should be stored by the smartpen. If strokes are stored, they will appear in Livescribe Desktop when users dock their smartpens and transfer their content.

To disable stroke storage for the entire paper product:

1. For all your pages, select the Crop region from the Palette or drawing area.
2. In the Property Inspector, uncheck the **Store strokes** checkbox.

To disable stroke storage for specific regions:

3. Select the active region from the Palette or drawing area.
4. In the Property Inspector, uncheck the **Store strokes** checkbox.

**Note:** When multiple active regions overlap, if any of them is set to not store strokes, no strokes are stored for any of the overlapped active regions.

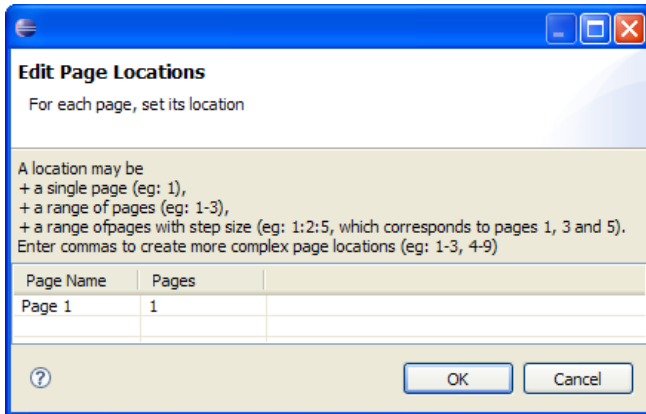
### Creating Multi-Page Note Pads

If you want to create a multi-page note pad that has identical designs for all left page and identical designs for all right pages, you can just create a basic two page note pad with the desired left and right page designs, and then replicate those pages for as many as 50 pages (25 sheets). If you require a larger note pad, please contact your Livescribe Developer Support representative.

To create a multi-page note pad:

1. Create a paper product project with two base pages using the Paper Product Creation Wizard.
2. Right-click on the page Thumbnails view for each page and choose **Edit Page Allocation**. This allows you to reuse the page specification for multiple physical

pages. They are all edited as a group.



3. Change the "1" in the pages column to "1:2:49". This indicates that the left page will start with page 1, continue every second page, and end on page 49.
4. Change the "2" in the pages column to "2:2:50". This indicates that the right page will start with page 2, continue every second page, and end on page 50.
5. Click **OK**
6. Add Active Regions to your two pages to add custom controls and any standard Livescribe controls.
7. In the project explorer, open the folder for your paper product.
8. Open the src folder and select the .afd file.
9. Select File >Properties.
10. Select the **Document** tab.
11. Click Edit Application List.
12. Add the desired application class names. For example, for Paper Replay, type com.livescribe.paperreplay.PaperReplay in the bottom entry field.
13. Click **Add** and then click **OK** to exit the dialog.
14. Return to your Active Regions and edit their properties by selecting their application and setting their Area Ids to match the expected Area Ids for their applications. See [Standard Livescribe Controls](#) for details.
15. After completing your paper design, follow the guidelines for printing the paper product in [Printer Settings](#).

## Standard Livescribe Controls

You can include certain standard Livescribe controls for built-in applications, such as Paper Replay, smartpen Settings, and system controls like the Nav Plus.















## Paper Replay Controls

You should consider adding these buttons only when it is appropriate to provide a link to the Paper Replay application. When a user performs a penDown event on any of the Paper Replay active regions, the system will de-active the current application and activate Paper Replay.

To include functionality that is similar to Paper Replay in your own penlet, you must code it yourself. It is not provided by inclusion of these areas in an AFD.

Class name: **com.livescribe.paperreplay.PaperReplay**







Paper Replay Function	Image	Area Id
Record Button		2
Pause Button		3
Stop Button		4
Add Bookmark		5
Previous Bookmark		17
Next Bookmark		18
Slow Play Button		9
Reset SpeedPlay Button		10
Speed Play Button		11
Position Bar Button		6
Jump Back Button		7
Jump Forward Button		8

## Smartpen Settings

You should consider adding these buttons only when it is appropriate to provide a link to the Settings application. When a user performs a penDown event on any of the Setting active regions, the system will de-active the current application and activate Settings.

To include functionality that is similar to Settings in your own penlet, you must code it yourself. It is not provided by inclusion of these areas in an AFD. Settings functionality is available in the `com.livescribe.configuration` Package.

Class name: **com.livescribe.settings.Settings**

Settings Function	Image	Area Id
Date Display		96
Time Display		95
Battery Display		21
Storage Display		22
Audio Display		98
Software Version Display		103

## System-Level Controls









You can include the following Area Ids on your paper product to provide system-level functionality for menuing and volume control. These controls are called *syslets*.

Unlike penlets, the currently active penlet does not lose focus when a smartpen-down occurs on these Area Ids. That is, when a user taps on a syslet control, like on the Nav Plus, the currently-running applications continue to run. For example, if you are running your application, it will continue to run, even when you tap on the Nav Plus.

Class name: **com.livescribe.runtime.SystemPenlet**

Pulse System Setting	Image	Area Id	Behavior Notes
----------------------	-------	---------	----------------

## Developing Paper Products

Nav Plus Left		1	System generates MenuEvent
Nav Plus Right		2	System generates MenuEvent
Nav Plus Up		3	System generates MenuEvent
Nav Plus Down		4	System generates MenuEvent
Nav Plus Center		5	System generates MenuEvent
Volume Increase		52	Volume UI Displayed, Display returned to App after a few seconds
Volume Decrease		50	Volume UI Displayed, Display returned to App after a few seconds
Volume Mute		51	Volume UI Displayed, Display returned to App after a few seconds

## Common Keyboard and Mouse Shortcuts

Action	Shortcut
Delete Region	Delete (inside main editing window)
Delete Polygon Vertex	Minus or Backspace while pressing the vertex handle
Zooming the view	Ctrl and Mouse wheel or Ctrl and + or Ctrl and -
Copy	Ctrl + c

Cut	Ctrl + x
Paste	Ctrl + v
Select all	Ctrl + a
Undo	Ctrl + z
Redo	Ctrl + y
Enables quick find using keyboard of a specific open document	Ctrl + e
Back and forward over document history	Ctrl + left and right key
Save	Ctrl + s
Close	Ctrl + w
Print	Ctrl + p
New project	Ctrl + n
Minimize UI	Ctrl + m
Either left-click on the bounding box or use ALT + left-click to cycle through all objects underneath the mouse pointer.	Selecting objects

## Tools for Image Creation and Printing

This section discusses tools and processes for creating images and printing paper products. Topics include:

- [Layout and creation of images](#) – These tools can take EPS and other file formats, lay them out, and export as EPS.
- [Conversion of images to EPS](#) - These tools can convert from another format to Encapsulated Postscript (EPS).
- [Conversion to PDF for printing](#) – These tools can convert the EPS output from the Paper Designer to PDF.

## Layout and Creation of Images

This section describes software you can use to create graphical content for your paper product.

Adobe has several products (such as Indesign and Illustrator) that are excellent for creating images and exporting them in EPS format. A great free alternative is a program called Inkscape.

- Illustrator - <http://www.adobe.com/products/illustrator/>
- Inkscape - <http://www.inkscape.org/>

The Paper Designer itself is actually a layout program since it allows collation of multiple images on a page. You may choose to do all your image content creation in an external program and import the graphics. Alternatively, you may choose to do image collation in the Paper Designer.

## Conversion of Images to EPS

The Livescribe Paper Designer requires all images used in a paper product to be in EPS format. There are many layout and composition programs available today that can import bitmap and vector based formats and export them as EPS. If you own one of these programs, you can easily prepare your images for the Paper Designer.

If do not own one of these program, but own a copy of Adobe Acrobat Professional, you can use it to export EPS files from PDF. Therefore, if you can convert an image into PDF, then you can use Acrobat to finish the conversion to EPS. Acrobats EPS generation is often technically perfect.

- Acrobat - <http://www.adobe.com/products/acrobat/>

There are some less expensive alternatives to Adobe Acrobat Professional. Instead, you can use ImageMagick and Ghostscript together to convert from many formats into EPS. The files generated, however, might not always be 100% correct. For example, the WYSIWYG may be incorrect and the output EPS files might be larger than those generated by Acrobat.

- Image Magick - <http://www.imagemagick.org/>
- Ghostscript - <http://pages.cs.wisc.edu/~ghost/>
- Ghostview - <http://pages.cs.wisc.edu/~ghost/>

## Conversion from EPS to PDF for Printing

If your print process or vendor requires or prefers to use PDF, you will need to convert the Paper Designer output from EPS to PDF.

The best option is to use Adobe Acrobat Distiller. This is standard with Acrobat Professional. A common problem with Distiller is that it does utilize the page bounding box when it determines the output size of each page. The page size setting in Distiller will set the page size for all the pages. This is fine when one page size is used in the entire document. It will fail if the paper product has different page sizes, or uses mixed portrait and landscape layouts.

To fix this problem, check the box **Fix page size** in the printing options of the Livescribe Paper Designer. This will hard code the desired page sizes in the output postscript, and will dictate to Distiller what the page size for each page should be.

Do not use this setting as a default. Turn this on whenever your intent is to generate PDF from your PS and then turn it off.

As an alternative to Acrobat Professional, you can use Ghostscript. It is a command line tool and requires some computer skill to use properly.

## Printer Settings

Currently there are two options for printing your paper products:

- directly print from Paper Designer
- generate a postscript file (an EPS) and print using a program that can print it

Printing dot pattern with good quality is difficult. A standard page (A4/Letter) has about 700,000 dots, and each of those needs to be rendered by the device with pinpoint accuracy, by the mechanical and electrical print engine, and by the RIP (the software and/or hardware that converts from a vector format to the bitmap to be printed by the physical device).

Livescribe pattern is a digital code. Accuracy of the pattern is vital. Unfortunately, most devices are optimized to print text and photos well. Since device manufacturers optimize their devices for the common case, newer machines have advanced image processing algorithms that actually destroy dot pattern. In addition, software conversion through various formats can introduce severe scaling and a minimal scaling, translation, or processing of the content that ruins the dot pattern entirely.

To have the highest probability of printing dot pattern on a device, Livescribe recommends that you turn off all image processing, image scaling, image centering, and print using Raw output postscript files without any conversions.

Given the large number of dots on the page, the performance per page can go down. In certain high-speed environments, this can cause problems. Printing postscript is

normally the best option since the postscript code is highly optimized to produce printed pattern fast. Conversion to PDF might slow down printing by up to 100 times depending on the conversion method and format.

In addition, it is important that the pattern is clearly visible to the smartpen infrared camera, and that the graphics color is handled correctly. Even the application order of the colors might affect the pattern quality since later or earlier iterations on the same paper can destroy the pattern contrast and location.

When printing or generating postscript files there are many different options available, often more than what is displayed in the UI. Most of these options are required to support as many devices as possible. If care-free printing with very high performance and quality is required, Livescribe recommends using one of the Anoto tested printers (see [Recommended Printers](#)). On tested devices, you can leave all the settings as default values and produce perfect pattern when printing directly from the Paper Designer.

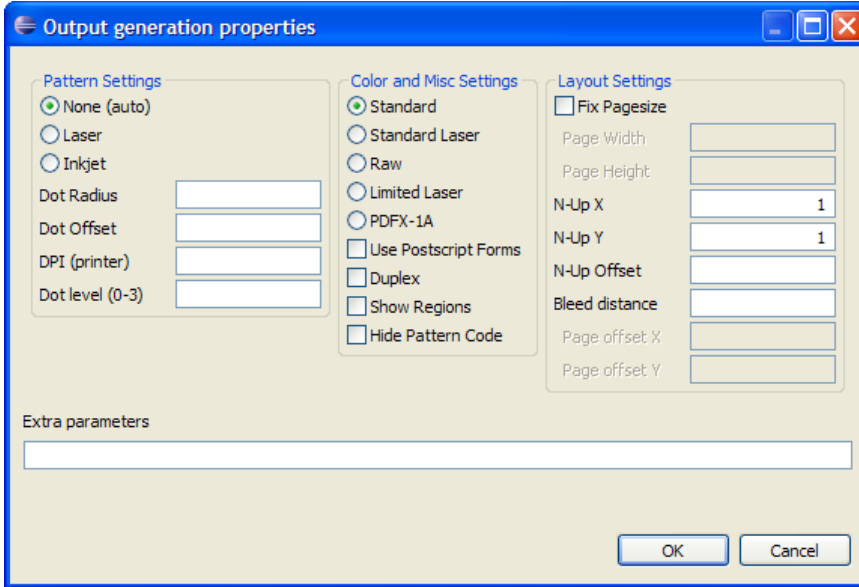
## Setting Output Generation Properties

This section describes available output generation properties and how they work. There are several categories of properties to set:

- [Pattern Settings](#)
- [Color Settings](#)
- [Miscellaneous Settings](#)
- [Layout Settings](#)

**Note:** In the Paper Design, most of the settings that refer to a distance or size accept multiple types of units. For example, it works correctly to write 12um, 33mm or 1337au.

## Developing Paper Products



### Pattern Settings

None (auto)	This means that the dot is optimized for offset printing but will attempt to adapt if target is a local printer. This means dot offset 50 um, dot radius 30 um and DPI 0 which means infinite.
Laser	This is the setting to use for a laser device; it can work well on certain inkjet devices as well. This means in practice 600 DPI and dot radius 40 um
Inkjet	This is the setting to use for an inkjet device. It means in practice 600 DPI and a specific special switch that lower the dot radius. Note that Laser with a manually set dot radius to a low value is not identical to using this setting, in addition to this certain inkjets have very exotic DPIs and they could benefit greatly by specifying this manually also into the DPI box.
Dot radius	This is the radius of the dot on the paper. This setting overrides whatever mode above is selected but retains the other settings of that mode
Dot offset	This is the offset from the grid center position. This is a rarely used setting.
DPI	This is the target DPI of the target device it will cause the pattern to be optimized to the pixel grid on a device with this particular DPI. Specifying laser or inkjet will automatically set it to 600. Any integer multiple of 600 DPI will give exactly the same result as 600 DPI so there is no quality gain by specifying 1200 DPI for a 1200 DPI device. Typically this field is only used for exotic DPIs. Since most normal devices are multiples of 600 DPI. Another thing that the user must be aware of is that since the pattern will be scaled differently depending on what the DPI is set to the entire page including the graphics (to preserve dot vs. graphics relationships) will be scaled accordingly. When using 600 DPI setting the page will be scaled down roughly 1% or for you that want the exact formula $25.4 \text{ (mm/inch)} / 600 \text{ (dots/inch)} * 7 \text{ (dots)} / 0.3 \text{ mm} = 0.98777$ . This causes some problems for



## *Developing Paper Products*

	content that needs to be exactly a specific size when printed on a 600DPI device since it will come out of the printer 1% smaller than the original content. The solution for this problem is to upscale the original content with the inverse value and after this scaling occurs it will come out the correct size.
Dot level	This is a way of compressing the size of the output postscript. The default value is 1 and the other allowed values are 0, 2, and 3. The value 2 and 3 reduce the pattern size to 50% and 33% respectively. The value zero removes the pattern entirely from the output and is useful for proofing and testing purposes. Setting this value to 3 always is not necessarily a good idea because of two things. First it might reduce the pattern decoding performance, which might lead to lower printing speed. Secondly some older devices might not be 100% compatible with this and might cause malfunction. If the output will be converted to PDF the value here should be 1 since the output PDF size will be the same not matter what the setting is.

## Color Settings

The color settings affect how color of graphics and bitmaps are handled by the print engine. Knowing how this works is crucial to get good color fidelity and performance. It is also required to ensure that the pattern is visible through the graphics on the page. Ghostscript is required for several of the settings.

Standard	This is the default mode, which takes the input content and converts it through Ghostscript but does no additional processing on the content. The reason for this being the standard mode is not that it is the mode to be used; rather it is default is because it is the most robust and it allows malformed EPS files to be used.
Standard Laser	This mode works through Ghostscript as the standard mode does but it also automatically process any input color to be invisible to the pen on majority of devices. In practice what it does is that it converts everything to CMYK and moves the K component in the graphics to the three other components. It is not fool proof since some devices do automatic undercolor removal in driver or hardware and for those types of devices the Limited Laser mode is the last resort, explanation of that mode is further down this document.
Raw Mode	This is what Livescribe use for our printed content and it should be used any time the requirements on performance, size, quality and color fidelity are high. It is the recommended mode to use when printing on offset since as long as 5 color plates are used the color can remain untouched and still work. This mode does not require Ghostscript since the input is embedded in clear text in the output stream and any input will be completely untouched by the tool, which will guarantee quality. With great power comes great responsibility however, if the input is malformed the output will also be malformed and various other problems can occur. In addition,

## Developing Paper Products

	any special color processing like getting rid of the K component from the graphics for digital devices is solely up to the person creating the original content.
Limited Laser	This mode is targeted at devices that have automatic undercolor removal that cannot be turned off. It will transform the colors in a way that will prevent the automatic undercolor removal to generate any K in the output. It completely destroys most colors and is unappealing to look at but it gets the job done and lets the pen read the pattern everywhere.
PDFX1A	This mode works like standard mode but will convert any RGB images to CMYK images since PDFX1A standard requires all images to CMYK. It is a rarely used mode and if requirements are so specific that they require PDFX1A compliance it is usually solved at content creation and using the raw mode.

## Miscellaneous Settings

These settings do not belong in any other category.

Use Postscript forms	This uses a feature of the postscript language to cause repeated graphics only be defined once in the postscript files. If your device supports postscript forms it can dramatically reduce size of the output especially when the content contains many repeated graphics such as notebooks.
Duplex	Duplex specifies to the printer to print on both sides of the paper.
Show Regions	This will add extra graphics that show the active regions on the printed page. It is useful for checking alignment between graphics and areas and to be able to design test paper before the final graphical content exists.
Hide Pattern Code	This removes the Pattern Print Code from the output. The Pattern Print Code is a tiny text on the lower left corner that shows useful information for debugging paper in the field.

## Layout Settings

The layout settings specify how the content is positioned on the pages. Many programs and RIPs also provide settings for some or all of these features. Be very careful if you use those features in your RIP because it does not account for the required scale and placement of dot pattern and might ruin the pattern quality.

Fix Page Size	This will force the target device (RIP) to provide output using each pages page size. This is very useful when converting to PDF and to print landscape pages correctly. The drawback of setting this is that if the target device cannot accommodate the exact size of each page the printing will typically fail.
Page Width & Page	These to options are only allowed if Fix Page Size is set to true. This setting will

## Developing Paper Products

Height	override the page size that is enforced to the specified value for all pages.
N-Up X & N-Up Y	N-up will cause multiple pages in the document to be printed on the same output page laid out side by side. For example specifying N-Up X = 2 and N-Up Y = 3 then up to 6 pages will be printed on the same page.
N-Up offset	This changes the default offset added between each of the tiled pages when printing using N-up
Bleed distance	This is the distance to bleed the pattern outside the crop marks. The value depends on what the print vendor specifies. Typical values are between 4 – 9 mm. Specifying any value here will also automatically turn on crop marks. The crop marks aids cutting the paper correctly. When crop marks are added to the page color swatches and a large printed label with the page address is added in the crop region.
Page Offset X & Page Offset Y	When using bleed a bleed setting and crop marks these offsets will move the printed content into the page a specified value. These values together with the N-Up settings allow alignment of the graphics with the printed content on the rear side of the content if duplex printing is used.

## Recommended Printers

Livescribe licenses its dot pattern technology from Anoto Group AB. Anoto maintains a list of tested printers that accurately print dot pattern. Please refer to their website for the official list of printers: <http://www.anoto.com/which-printers-can-i-use-with-anoto-pendocuments-4.aspx?cid=printer-requirements>

As of this printing, the tested laser printers are:

Okidata Laser Printer Models:

- OKI C5450
- OKI C5900
- OKI C5950
- OKI C6100
- OKI C6150
- OKI C710
- OKI C9600
- OKI C8800
- OKI C9650

## Other Printer Models

If you have a color laser printer that prints 600 dpi or higher, it may be capable of printing Livescribe dot paper as well. Some other laser jet printers also may work.

To test if your printer can print Livescribe Dot Paper, use the Print Your Own Notepad feature in Livescribe Desktop for Windows. Print and validate the test page for this feature. If the test page is valid, then you can use that printer to print your custom paper product. Refer to the *Livescribe Desktop for Windows* user guide for more information.

## The Livescribe Pattern Server

To create published paper products, a dot pattern license needs to be applied to the AFD. The Livescribe dot pattern contains millions of pages and the central entity that tracks the pattern allocation is called the Livescribe Pattern Server (LPS). The LPS is a web service hosted by Livescribe. The server keeps track of pattern space used by Livescribe and provided to developers. The LPS ensures that all paper products use unique page addresses (PA).

## Pattern License

A pattern license is a file that contains what pages the license spans, meta data (like pattern type), and some DRM. A pattern license can either be added to a document directly from the license server or through a file in the file system. A document's licenses can be exported to disk, or removed and viewed through the Paper Designer. As a rule, do not apply the same license to multiple paper products. Instead, get a new license for each paper product.

## User Accounts

To get pattern from the LPS, you need the server address, an account name, and a password. You can get an account by emailing Livescribe at: [patternlicense@livescribe.com](mailto:patternlicense@livescribe.com).

There are some properties connected to an account, such as a contact email and a user level. The user level for most users is USER. In addition to the other properties, each user has a number of Pattern Credits (PC). The user account information is private confidential information and should be treated as such and never give away your account password to anyone. Livescribe will never ask for your password.

## Pattern Credits

A pattern credit is the virtual currency used to buy pattern. Each user account has a certain number of credits, much like a bank account. A letter or A4 size page is worth roughly 10 PC. A smaller page is worth less and a bigger page is worth more with cost proportional to the page size area. When a license is acquired the cost of the license (number of pages \* cost per page) is deducted from the account balance of the account. If the license costs more than the available credits on the account, an error is reported and the operation fails. When this happens new credits needs to be added to the account. Contact [patternlicense@livescribe.com](mailto:patternlicense@livescribe.com) to get more pattern credits.

## Additional Pattern Server Information

The LPS has a standard public JAX-RPC web service API and user created tools can use the pattern server to acquire pattern in their applications for example in proprietary paper design tools or print on demand pattern printing. Before accessing the JAX-RPC API directly, please email [patternlicense@livescribe.com](mailto:patternlicense@livescribe.com) to get an authorization to run requests against the server not originating from the Paper Designer.

The license model is hierarchical, which means that a license can be split into smaller pieces multiple times. For corporate and other types of institutions, it is possible to run a private pattern server to serve the organizations pattern needs. A private server would get very large licenses from the LPS (thousands of pages) and then subdivide them further once clients contact the private server. Hosting a private pattern server is only relevant if very high requirements on availability, privacy, throughput and control is required. For most generic use the main LPS is sufficient. The pattern server is currently not an off the shelf product but interested customers can contact Livescribe sales for questions.

## AFD Document Format

Information about each Livescribe paper product is stored in a proprietary document format known as an AFD (Anoto Functionality Document). An AFD is a multipage, viewable, printable, and editable document format similar to PDF, PS, and XPS. An AFD file contains information about which applications the paper product is linked to, the number of pages, the sizes of those pages, and the regions on the pages. The AFD format is what enables all parts of the Livescribe platform to work with and process data associated with paper products. Components of the Livescribe platform use AFDs to perform various tasks:

## Developing Paper Products

- **Paper Designer:** Creates, views, and prints paper products.
- **Livescribe Desktop:** Views, manages, prints, uploads, and archives paper products.
- **Pulse smartpen:** Reads and adds strokes and other data
- **Livescribe Online:** Creates pencasts.
- **Penlets:** Can get information of the current page and trigger on areas, in the future, it can also read other information in the AFD.
- **Desktop applications:** Once the Livescribe Desktop SDK is released, an application on the desktop will be able to read, write, create, and print AFDs.

## AFD Technical Details

The AFD format is XML and other file formats in a container (zip). You can explore an AFD using a zip reader and a text editor. However, be very careful not to modify the files because even small errors or slight zip incompatibilities might render the file unusable for the Livescribe platform.

Note: With the future release of the Desktop SDK, tools will be available to make programmatic changes to the AFD in a controlled manner.

An AFD file contains the following components:

Document info	Contains basic information of the file such as the Author, Title etc
Document	Contains the number of pages, page sizes and a link to the page specification
Page Specification(s)	Contains graphics and regions for the page
Pattern	Contains a specification of what pattern addresses are mapped to the document and how
Licenses	Contains the licenses for the pattern
Resources	Contains the graphics resources for printing and viewing (EPS files)
User data	Contains any addition to the document that is not covered by the base AFD format. Some of the additions are from the Livescribe platform itself and in the future a developer will be able to add his own data to the AFD and it can tunnel through the architecture together with the rest of the paper data
Strokes	Contains the stroke files from the smartpen
Application mapping	Provides the link between a paper area and a penlet

Icons	The icons for display in LD or other desktop applications
-------	---

## Templates and Stride

When creating, for example, a notebook where there are many exact identical pages in a specified pattern it would be cumbersome to create and edit many copies of identical pages. To help this standard flow there is something called a page template. A page template is an abstract representation of a page, and document is built from specifying a sequence of page templates.

When creating a new page in the Paper Designer the user actually creates a page template. This page template can then be edited as the user wishes. For a typical notebook two templates are created, one for the left page and one for the right page. Once these are ready, you can create a notebook with every other page left page, and every other page is right. This is done by editing the page allocation. The syntax for this might be confusing at first, but is actually simple.

In the example case, the proper way of expressing the notebook is to write 1:2:99 on Page 1 line and 2:2:100 on Page 2 line. This will cause the left page to start on first output page and go on every other page ( :2: ) up to page 99, corresponding for Page 2 is that It should start on page 2 and exist on every other page up to page 100. The step size between each copy of a page template is called *stride*. The syntax enables much more complex expressions than this simple base example.

There are two significant advantages of using the page template system:

- workload reduction since only two pages needs to be actually created for a notebook
- the smartpen performance will be higher the fewer page templates that exists in the document

It does not take along time to open an AFD in the smartpen and the smartpen will do it when the smartpen tip touches a new paper product and since the time to feedback (sound/text) is the AFD open time it is crucial that this is very low.

## Pattern Licenses

A pattern license is acquired from the Livescribe Pattern Server (LPS) and it is unique from all other licenses. It is important to have a central entity that distributes licenses since otherwise there is a risk of two paper products sharing the same pattern, which would cause double exposure and terrible effects in the system.

A pattern license contains a number of pages, the size of the pages, some info about the purpose and owner of the license and some DRM. A license is a file and can be stored to disk, emailed etc. To produce any real content using the Paper Designer an account on the LPS is needed. See [The Livescribe Pattern Server](#) for more information.

## AFD Copies

AFD copies are also a slightly complex subject. One AFD copy is all the printed pages specified in the document. A document can contain multiple copies and multiple copies can be printed.

Each copy looks identical to any other copy for the naked eye since the number of pages and the graphics on each of the pages are identical. However, the pattern for each copy is different.

Paper product created by Livescribe (like Livescribe notebooks and journals) require archiving when a new notebook of the same type needs to be used. The reason for this is that notebooks of the same type are all the same copy (that is, use the same pattern. If they were all different copies (different pattern) archiving would not be necessary, and a user could have any number of the same notebook in use at the same time.

Livescribe printed the notebooks using the same copy for cost reasons. Since the notebooks are printed in very high volume, digital printing was not feasible. Further, the offset printing requires each copy to be identical, which rules out changing the pattern per page. Therefore, for many types of paper products, it is not feasible to use copies. For any content that is produced using digital printing, however, it is beneficial to use copies.

Adding copies to a document is very simple. Adding more pages to a license will add more copies. A document can contain any number of pattern licenses and they can be added long after paper creation to add more copies.

For example if a 2000 page license is added to a 100-page document, 20 new copies will be added to the document. Tracking which copies are used, and whether or not a document has run out of free copies is a complex topic that will be documented with the Livescribe Desktop SDK.

## Locked Property

An AFD has a state that specifies if it is locked or unlocked. A document normally remains unlocked during content creation and while doing test printing. Once a real



production license has been applied, it is locked. Locking prevents changing the pages size, number of pages, areas and graphics on the pages. The reason to lock the document is that any changes to the document would render the physical printed copies of the document out of sync with the AFD file and this need to be avoided at all costs so the lock is there for a good reason. It is possible to unlock a document in the document properties section in Paper Designer, but it is discouraged. One reason to do this might be to remove all licenses and create a different document using the old one as the base.

## AFD for the Smartpen

Currently to save some space there is a specific AFD for the smartpen where the EPS graphics is stripped from the file. This file is automatically created by the Paper Designer in the background. In the future to enable documents to be printed after “transport” in the smartpen the AFDs might not be stripped of the graphics.

## Expanded AFDs

There is an option to create something called an expanded AFD. What this does is that it takes each page in the document and creates a unique page template for that page inheriting all properties but also adding an extra image to each page that is unique for that page. The purpose for this is to support very complex print flows where it is required to have a custom crop image per page and where the built in crop mark generation does not suffice. Most customers will probably not use this feature, but it is important to be aware that the standard AFD is the base document. Even if the output expanded AFD can be edited and printed, it should not be used for anything but printing since it will have lost many of the good properties of the base AFD with few page templates.

## Document GUID

The GUID (globally unique identifier) is a 64 bit number that normally is unique for each AFD. A document GUID is used to identify a document uniquely and can be used to look up a document in a database or to specify what documents has changed in a smartpen since the last time of docking. In addition, to prevent theft of the pattern licenses in a document the licenses can be GUID locked to prevent them from being used in any other document. The document GUID is generated when a license is added to a document and the document is locked. Changing the document after generating the GUID will render the document unreadable and this is another reason for the locked state of the document.

## Publish Document

Publishing a document is the final stage of the production process. When this action is carried out, all the licenses in the document are GUID-locked and the document can no longer be changed, at this point it is also safe to distribute the AFD to non-trusted parties since the licenses are protected.

## Printing Basics

Paper creation and printing are rather complex to grasp without prior experience. Software engineers with a background in graphics often seriously overestimate how much of their ability translates into the printing field. Most things actually do translate properly, but the terminology differs, and a lot of the terminology originates from the printing industry with its thousands of years of history. So, even if you as the reader feel well informed with respect to software engineering in general please study the information provided in this section.

## Printing Terminology

This is a list of common printer terminology terms and definitions. LS indicates Livescribe-specific terms.

Duplex	Printing on both sides of the paper
Trapping	Adding additional color in "holes" so that even if the plates are misaligned slightly the white color will not shine through
Postscript Forms	A method that enables a complex piece of graphics to be represented once in the file and then used over and over without specifying the content again. This is used to optimize the file size
Process Colors	The built in standard set of colors in the device
Spot Colors	Optional colors used when printing
RGB	Red Green and Blue color typically used on computers and with digital photography
CMYK	Cyan Magenta Yellow Black color typically used on printing devices

## Developing Paper Products

Page Bounding Box	A rectangle encompassing the page
Crop Marks	Lines outside the Crop Box that shows printing personnel how to cut the page into the finished product
Bleed	Extra graphics added on the edges of the page that “bleeds” outside the crop box, the purpose is that even if the paper is cut slightly wrong that the page will not have any white lines on the edges of the paper
PPC (LS)	Paper Print Code a small text in the lower left corner of each printed page that shows the page address and what pattern settings has been used
N-UP	Tiling multiple images on one sheet of paper
DPI	Dots Per Inch (the resolution of the printer)
RIP	Is the software and/or hardware that converts from a vector format to the bitmap to be printed by the physical device. Commonly a desktop printer has a built in RIP in the printer or the driver and the more complex the device gets the more likely there is that there are more different options in the rip and very high end devices have interchangeable RIPs that does its processing in both software and hardware and they can be very expensive. The word “RIP” is constantly both used as a noun or a verb.
Color Swatch	A color patch used to measure color quality, it is typically printed in the crop region
Crop Region	The page rectangle that represents the viewable or finished paper product size, this is smaller than or equal to the Page Bounding Box

## Print Processes

There are three primary ways to create paper for consumer use. There are many variations, hybrids, and these three are broad generalizations.

## Offset Printing

Offset printing is the oldest way of producing paper. The printing is carried out in multiple iterations on the paper surface; iteration consists of creating a relief in a metal plate of the graphics on the page, applying ink on the relief and pressing it onto the paper.

Each of these iterations adds a color to the page. The relief is monochromatic since the relief can either touch the paper on an area or not touch it at all. Creating 50% of the color requires a *halftone screen* (compare to dithering) this halftone screen has 50% of the surface area with color and 50% without the resolution of this screen affects how constant the color appears. These reliefs are called *plates* (since they are metal plates). There is one plate per color used on the page. Typically, when printing four different colors are used and these are called the *process colors*. The names of these are Cyan, Magenta, Yellow and Black, normally these are referred to as CMYK. Printing standard four color printing requires four plates one for each of the process colors. These plates are applied after each other and for example to make the color, red requires to print using M and Y since these together creates red. On a printer, you typically start with a white paper. Adding the maximum of all colors will result in black. The fourth color (black) is used to save ink and make grayscales more defined and the color in general less grainy.

When printing using an offset press due to its iterative nature any number of color plates can be used, even less than four. Since the initial cost to print something is largely dependant on the cost of creating the physical plates, many paper products use one or two plates. For certain content other colors than the process colors can be used, these colors are called *spot colors* and can be any color, either an off the shelf standard mixture or something very specific and proprietary an example of a special color is the ultra violet visible ink used on certain banknotes. Many combinations are possible for example, it is possible to print using four plates, one extra spot color, and only using black process color (K) and one extra spot color for a company logotype for instance.

The merits of using offset are high quality output and low cost per copy for large number of copies. The negatives are that it is not really "print on demand" and the initial cost (or the cost for a few copies) is very high.

## Toner Based Electrostatic Printing (Laser)

Electrostatic printing devices (or Laser printing) are based on electrostatically charging the paper where the color should end up and attracting opposite charged toner particles to the paper. Once the toner particles are attached to the paper they are fused using heat or similar to stick to the paper after charge is gone. The toner

itself is small “rocks” of color and often sports various very high tech coating to enhance the properties. The quality of the print is both related to the mechanical properties of the device and on the toner particle size, toner coating, and particle size variation.

An electrostatic device typically also uses CMYK as its native color space. It is very rare that it allows for spot colors to be used. If a printed document contains a spot color, it is approximately converted to a color using a combination of process colors.

The benefits of using a laser printer are the low latency, no additional physical cost for unique data per page. The negatives are that an office-level device cannot compete with the color quality and resolution of an offset press. In addition, the cost for high volumes are typically higher than for offset printed content.

### Inkjet Printing

Inkjet printing process works by shooting tiny droplets of color onto the paper. The droplets are typically created by a matrix of either piezoelectric or thermal droplet cannons (jets). Once the droplets end up on the paper, it naturally is absorbed by the *substrate* (aka paper). The performance of these printers are based on the accuracy of the inkjets, the precision in dot size and the graininess of the perceived color for lighter shades of color is based on the minimal size of the droplet that the jet can produce.

The inks in an inkjet are very expensive complex mixtures but the most important difference between them is that some of them are *pigment* based and others are *dye* based. It is very important to know if the printer has dye based K component. If that is the case, Livescribe pattern printed using this device will not be readable with the smartpen. Unfortunately, if the K ink is dye based or pigment based is not something you can read off the carton or spec sheet for the device, only testing of the ink or the printed output can check if this is the case.

A typical inkjet has CMYK as its process colors but some of them also have additional lighter variants of the process colors (often denoted cmyk), this is to help remedy the droplet size/graininess problem for lighter hues. From an external perspective, the color conversion between CMYK -> CMYK cmyk space is not something of concern because it is all handled in the printer.

A customer level inkjet does not support spot colors but high end industrial level printers might. The merits of using an inkjet vs. an offset process is the same as for laser vs. offset, lower latency, lower initial cost etc. It is however interesting to discuss the differences between laser and inkjet when it comes to printing pattern. Inkjet printers are very good at reproducing high quality photographs, but since the

variation of the dot size is large, the accuracy of dot placement is relatively poor and since the inkjets occasionally jam up this poses a problem when printing pattern. The dots can have variations in size, location and even certain dots can go missing altogether. Printing dots using inkjets can be feasible in certain cases but for devices in the 500\$-1000\$ range laser printers produce far higher quality pattern.

## Vector Based Formats

The historical way of representing graphics to be drawn onto a computer screen is by using bitmap graphics. A bitmap is an array of color values (pixels) that forms a two dimensional image. Typical bitmap formats are GIF, JPG and PNG. A bitmap format is the typical output from a camera since the camera captures the image in the same sort of two dimensional pixel grid. The problem with bitmap graphics is that they do not tend to scale well since the actual underlying geometry is not represented in the format, only intensity values per pixel. In addition to this problem, bitmap images size increase as the resolution squared i.e. a two times as high resolution image consumes four times the space.

A vector based format is a graphical representation of graphics that scales properly to any size; a circle remains a perfect circle for example. A vector format typically stores all the different operations that were used to create them. Vector formats are invariant to target resolution since they do not have a native resolution. Most vector based formats supports embedding bitmaps, for example if you layout a page you might want to add a photograph into it and for photographs a bitmap is the most appropriate storage format since as discussed previously a camera creates the photograph using pixels.

Historically when printing, vector based formats has been used to represent graphics. Vector based graphics has recently started to become the primary format as the resolution of the computer screens has come closer to a printers resolution and that the PC processing power has increased. Using vector based graphics on the desktop has the same advantages like scalability and lower size. The transition to vector based formats for the UI on the desktop was done in Vista/OSX.

There are multiple vector based formats just as there are multiple bitmap formats. The primary formats are from Adobe: Portable Document format (PDF), Postscript (PS), Encapsulated Postscript (EPS) and from Microsoft XML Paper Specification (XPS).

PDF is a modern standard that is the de-facto standard for the printing Industry today.

## Developing Paper Products

EPS and PS are the predecessors to PDF and they are actually the same format. The difference between EPS and PS is loosely that an EPS represents a single image or page but a PS can be a collection of pages.

XPS is Microsoft's PDF alternative that was introduced in Windows Vista. It has yet to gain any foothold in the very conservative printing industry and since it is a Windows-only format, it is not used in our printing path.

The primary format used by the Paper Designer as input is the EPS format since it represents graphics without any page information. The primary output of the Paper Designer is PS since it represents a printable multi page document.

The tool will gradually get PDF support added to it over time but for the first version, there will be none. See [Conversion from EPS to PDF for Printing](#).

## Color Theory and Smartpen Vision

The smartpen camera takes images in the infrared wavelengths; this means that the camera view is not the same as you see with the naked eye. This has several advantages since certain inks are invisible in this spectrum but visible to the human eye. Livescribe pattern itself needs to be visible in the infrared spectra but since the black ink/toner typically contains carbon particles it is highly absorbent in all spectra. The other process colors CMY are typically invisible and if only these are used for printing the graphics the graphics will not interfere with the smartpen reading the dots.

The theoretical conversion between RGB and CMY given all values are normalized to be between 0 and 1.

$$C' = 1 - R$$

$$M' = 1 - G$$

$$Y' = 1 - B$$

These formulas make great sense if you consider the discussion earlier about RGB being additive and CMY being subtractive. A quick test using RGB = 1, 1, 1 (white) gives us CMY = 0, 0, 0 which means no color, and on a white paper that should be white!

The black is generated from these derived color using a process called *black generation* and *undercolor removal*.

$$K = \min(C', M', Y') \quad (\text{black generation})$$

## Developing Paper Products

$C = C' - K$  (undercolor removal)

$M = M' - K$  (undercolor removal)

$Y = Y' - K$  (undercolor removal)

The CMYK color system used by printing devices is over determined this means that in the three dimensional color space there are four base axes (the different colors) since only three axes are needed to create any color (compare to RGB) if they are linearly independent. In nonprofessional terms only CMY is actually needed to create any color and black is optional. This is in theory, in practice the black is a great addition since as previously stated, it saves ink and it makes grays and blacks more defined and higher contrast.

Problems arise since most tools that create EPS files will create black color using the K component and this will disturb the pens reading of the dots. There are two ways to solve this problem. The first solution is to print using offset using one extra special spot color for the pattern. The black color used for printing the regular process K is used to print the pattern (as the spot color Anoto Pattern) and the regular process color K is changed to a color mixture that looks black in the visible spectra but is invisible in the infra red. The secondary solution is required on devices where there is no possibility of adding spot colors, the over determination of the CMYK color system now comes to the rescue. Since we know (from the previous formula) that  $K=X$  is equal color wise to  $CMY=X, X, X$  we can just manually convert the K color to the other colors when creating the content. To do this it is important to work in a tool that has a CMYK mode and that the document creator knows how this works.

There is also a built in color conversion method in the tool that automatically convert the incoming graphical CMYK to CMY only. This is selectable in the printing properties dialog and it is called "standard laser".

## FAQ and Troubleshooting

Question or Issue	Answer
When I print, the graphics do not align properly with the output.	There is a known issue with content bounding boxes when printing using other mode than Raw. There are two solutions to the problem. Either use the Raw mode or make sure that the actual content extends to the edges of the bounding boxes.
Why are the AFD files so large?	In the postscript, rendering preferences the DPI is set to a specific value. The default DPI that is used is too small for small images and too large for



## Developing Paper Products

	<p>large images. Try adjusting this value in the preferences prior to adding images. Also the rasterization DPI is in regards to original content size it is not related to the scaled size on the page and always try to use approximately the same physical size for the base content as the target size on the page i.e. always create the content roughly the same size as the target printed size. In future versions of the tool, an option to set the DPI dynamically depending on the image size will be available and that will fix most of these problems.</p>
<p>I have an AFD created by a co-worker. How do I open it up in the Paper Designer?</p>	<p>The simplest way is to create a new project, delete that AFD and then drag and drop the actual AFD file into the src directory in the project explorer. Importing the file from the file system is another alternative.</p>
<p>My document has become corrupt. It will not open.</p>	<p>Try to copy your files in the src folder to a backup location then delete them from the src folder. Move back the .wcopy file into the src folder and rename it to yourfilename.AFD</p>
<p>When I print a page with an image, two pages are ejected, one with the image and one with the pattern.</p>	<p>This problem occurs when using the Raw mode and when the input EPS has a showpage command somewhere in the file (typically very close to end). The Raw mode requires well formed EPS files and those should never include the showpage command since it actually instructs the printing device to eject the page. The solution is either to use not use Raw mode or to manually remove the showpage command from the eps file prior to adding it to the page.</p>
<p>I am running Ghostscript as my rasterizer but sometimes it crashes when I add an image.</p>	<p>Ghostscript can sometimes randomly crash using specific DPIs. Try changing your render DPI and add the image again.</p>
<p>My system or Eclipse crashed, and now when I start Eclipse it says my workspace is locked.</p>	<p>Try killing the javaw.exe process using the task manager and after that go into the .metadata directory in your workspace root directory (My Documents/My Paper Projects) and deleting the file .lock after this restart the SDK</p>
<p>I have prior experience with the Anoto design tools. How do this relate to them and are they compatible? Can I use the Livescribe Paper Designer to create paper products for Anoto pens and vice versa?</p>	<p>Currently the two solutions use completely different parts of the pattern, and a paper product that is designed by one of the solutions does not work with the other.</p>
<p>I want to cut pattern but I cannot find how to do it in the tool.</p>	<p>Cutting pattern is not currently supported.</p>

## Developing Paper Products

<p>I have a custom paper product and I want a mute button on it. Can I use the functionality for mute already in the Pulse, and if yes how do I do it?</p>	<p>Yes. You can use existing Livescribe controls. See <a href="#">Adding Standard Livescribe Controls</a>.</p>
<p>I want to have a Region with a hole in it.</p>	<p>Add your hole as a Region on top of the Region, make sure the new region has lower Z-order and the occlusive bit set to true. Assign a new Region Id to the new shape and do not perform any functionality when this new Region is activated.</p>
<p>What is the purpose of 1 dimensional shapes such as Line, Polyline? They have no area, how can a user press them?</p>	<p>The intersection in Pulse is based on intersecting the actual stroke with the shape so a Polyline will generate AreaEnter and AreaExit events when the smartpen is drawn over it. A PenDown or PenUp event cannot be triggered by these shapes.</p>
<p>What about a Point as a shape?</p>	<p>The Point shape is there for completeness and it can potentially be used for special queries like check for intersection with dynamic shapes and soon on.</p>
<p>I have tried to copy and paste multiple Regions across pages and documents. It sort of works, but it would be nice if the application link and Region Id were copied with the areas.</p>	<p>The properties are copied with the copied regions, but you still have to add the application to the application list, and verify that the controls are linked to the correct penlet.</p>