

Anyframe JasperReports Plugin



Version 1.0.1

저작권 © 2007-2011 삼성SDS

본 문서의 저작권은 삼성SDS에 있으며 Anyframe 오픈소스 커뮤니티 활동의 목적하에서 자유로운 이용이 가능합니다. 본 문서를 복제, 배포할 경우에는 저작권자를 명시하여 주시기 바라며 본 문서를 변경하실 경우에는 원문과 변경된 내용을 표시하여 주시기 바랍니다. 원문과 변경된 문서에 대한 상업적 용도의 활용은 허용되지 않습니다. 본 문서에 오류가 있다고 판단될 경우 이슈로 등록해 주시면 적절한 조치를 취하도록 하겠습니다.

I. Introduction	1
II. JasperReports Integration	2
1. Configuration	3
1.1. web.xml 작성하기	3
1.2. jasperreports-servlet.xml 작성하기	4
2. Controller	5
2.1. HTML Reporting	5
III. JasperReports Designer	7
3. Installation	8
3.1. 다운로드	8
3.2. 설치 환경	8
3.3. Report Designer 설치	8
3.3.1. Configuration	9
4. Design Report	11
4.1. 목표 결과물	11
4.2. 디자인 파일(JRXML) 작성	11
4.2.1. Step 1 : Open JasperAssistant Perspective	12
4.2.2. Step 2 : Create a new Report	12
4.2.3. Step 3 : Design a report using Palette	16
4.2.4. Step 4 : Preview Report	22

I.Introduction

jasperreports plugin은 Spring MVC와 JasperReports (<http://jasperforge.org/projects/jasperreports>) [<http://jasperforge.org/projects/jasperreports>]를 연계하여 Reporting 기능을 수행하는 방법을 가이드하기 위한 샘플 코드와 이 오픈소스들을 활용하는데 필요한 참조 라이브러리들로 구성되어 있다.

Installation

Command 창에서 다음과 같이 명령어를 입력하여 jasperreports plugin을 설치한다.

```
mvn anyframe:install -Dname=jasperreports
```

installed(mvn anyframe:installed) 혹은 jetty:run(mvn clean jetty:run) command를 이용하여 설치 결과를 확인해볼 수 있다.

Dependent Plugins

Plugin Name	Version Range
core [http://dev.anyframejava.org/docs/anyframe/plugin/essential/core/1.0.3/reference/htmlsingle/core.html]	2.0.0 > *

II. JasperReports Integration



는 오픈 소스로 다양한 콘텐츠를 PDF, HTML, XLS, CSV 파일 등으로 출력하는 리포팅 툴이다. 전체적으로 자바로 쓰여졌으며, 다양한 어플리케이션에서 사용되어 다양한 포맷으로 콘텐츠를 생성할 수 있다. XML 포맷으로 리포트 디자인을 작성하여 컴파일 한 후 RDBMS와 JDBC를 통해 데이터를 바인딩함으로써 최종적으로 다양한 리포트를 생성할 수 있다.

JasperReports 특징을 살펴보면 다음과 같다. :

- 다양한 Reporting Output 형태(PDF, HTML, XLS, CSV 등) 제공
- 다양한 Data Source를 이용하여 Reporting 데이터를 구성할 수 있음
- 다양한 레이아웃으로 Reporting이 가능하며 Chart 및 Graphical한 화면 요소 제공
- Graphical한 Report Designer 제공(여러 종류의 유/무료 툴 사용 가능)

아래의 그림은 XML 포맷의 리포트 디자인 과정부터 최종적으로 JasperReporting Engine으로부터 최종 결과 리포트를 생성하는 과정을 한 눈에 볼 수 있도록 표현하고 있다.



Jasper XML(jrxml) 파일을 생성하는 툴을 제공함으로써 개발자는 비즈니스 컴포넌트 개발 후 쉽게 리포팅 기능을 구현할 수 있다. 또한 이 매뉴얼에서 설명하는 JasperReports (<http://jasperforge.org/projects/jasperreports>) [<http://jasperforge.org/projects/jasperreports>] 기능은 Spring MVC와 통합된 형태로 예제가 설명되어 있다. Anyframe 에서 제공하는 Reporting 기능은 기본적으로 JasperReports 기능을 모두 제공하나 Spring MVC와 통합된 형태로 기능이 제공되고 있기 때문에 Spring MVC를 통해서 제공되지 않은 JasperReports 일부 기능이 있을 수 있다. 이 경우 필요하다면 Spring MVC를 확장하여 구현 가능하다.

예를 들어 보면 다음과 같은 경우들이 포함된다. :

- **HTML 파일로 Reporting** 하는 경우 - 이미지 파일을 html 내에서 디스플레이 해주기 위해서 Spring MVC에서 제공하는 뷰 클래스를 확장하였다.

Anyframe 를 설치하여 Reporting 기능을 이용할 경우에는 이 매뉴얼에서 가이드 하는 방법대로 사용하면 위의 기능을 추가 코딩 없이 바로 사용할 수 있다.

아래 내용을 보고 JasperReports를 설치한 후, 직접 JRXML 파일을 디자인하고 샘플 어플리케이션을 실행시켜보도록 하자.

1. Configuration

JasperReports와 Spring MVC를 연계하기 위해서는 다음과 같이 2가지 XML 파일에 대해서 설정해야 한다.

- web.xml 작성하기
- jasperreports-servlet.xml 작성하기

이 매뉴얼상에 언급된 소스 코드들은 모두 Eclipse에 Import한 anyframe-jasperreports-pi 프로젝트에 예제로 들어있다.

1.1. web.xml 작성하기

anyframe-jasperreports-pi/src/main/webapp/WEB-INF 폴더 하위의 web.xml에 아래 내용을 추가해 넣는다. 기존의 Anyframe 관련 설정은 그대로 두고 JasperReports 관련한 설정만 더 추가한 것이다.

jasperaction 서블릿의 클래스인 DispatcherServlet은 Spring MVC 클래스로 url이 /reports/ pattern으로 요청되면 모두 jasperaction 서블릿을 통해 서비스 되도록 설정하고 있다. **image** 서블릿은 HTML 형태로 Reporting 하는 경우 image 파일을 처리해 주기 위해서 필요한 서블릿이다. HTML 형태로 Reporting 하지 않는다면 설정할 필요가 없다.

```
<servlet>
  <servlet-name>action</servlet-name>
  <servlet-class>
    org.springframework.web.servlet.DispatcherServlet
  </servlet-class>
  <init-param>
    <param-name>contextConfigLocation</param-name>
    <!-- in classpath, jasper-servlet.xml exists -->
    <param-value>classpath:/spring/*-servlet.xml</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>

<servlet-mapping>
  <servlet-name>action</servlet-name>
  <url-pattern>*.do</url-pattern>
</servlet-mapping>

<!-- jasperreports-configuration-START-->
<servlet>
  <servlet-name>image</servlet-name>
  <servlet-class>net.sf.jasperreports.j2ee.servlets.ImageServlet</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>action</servlet-name>
  <url-pattern>/reports/*</url-pattern>
</servlet-mapping>

<servlet-mapping>
  <servlet-name>image</servlet-name>
  <url-pattern>/reports/image</url-pattern>
</servlet-mapping>
<!-- jasperreports-configuration-END-->
```

1.2.jasperreports-servlet.xml 작성하기

action 서블릿의 클래스인 DispatcherServlet이 사용하는 빈 설정파일에 jasperreports-servlet.xml 파일을 등록해주도록 한다.

- **reportViewResolver** - views.properties 형태로 URL이 요청된 경우 해당 Controller 클래스의 viewReport 메소드를 수행시킨다.
- **reportView** - 가장 핵심이 되는 빈으로 Controller로부터 뷰 이름을 알아낸 결과 reportView인 경우, ExtendedJasperReportsMultiFormatView 클래스를 실행시킨다. reportView 빈의 Property 설명은 다음과 같다.

Property	Value	설명
url	/WEB-INF/jsp/jasperreports/report/movieReport.jasper	jrxml 파일을 JasperAssistant Preview를 이용하여 컴파일한 결과 파일
jdbcDataSource	dataSource	현재 HSQL DB를 사용하는 dataSource 빈 사용
exporterParameters - ...IMAGES_URI	image?image=	HTML Reporting 시 이미지 파일을 출력하기 위해서 web.xml에 정의한 image 서블릿을 사용하도록 설정
exporterParameters - ...CHARACTER_ENCODING	euc-kr	Reporting 시 한글 깨짐을 방지하기 위해서 설정

```
<bean id="reportViewResolver"
class="org.springframework.web.servlet.view.ResourceBundleViewResolver">
  <property name="order" value="1"/>
  <property name="basename" value="jasperreports.views"/>
</bean>

<bean id="reportView" class="org.anyframe.jasperreports.view.ReportsMultiFormatView">
  <property name="url" value="/WEB-INF/jsp/jasperreports/report/movieReport.jasper"/>
  <property name="jdbcDataSource" ref="dataSource"/>
  <property name="exporterParameters">
    <map>
      <entry
key="net.sf.jasperreports.engine.export.JRHtmlExporterParameter.IMAGES_URI" value="image?
image="/>
      <entry key="net.sf.jasperreports.engine.JRExporterParameter.CHARACTER_ENCODING"
value="euc-kr"/>
    </map>
  </property>
</bean>
```

2.Controller

Anyframe 에서 제공하는 Reporting 기능은 기본적으로 JasperReports 기능을 모두 제공하나 Spring MVC 와 통합된 형태로 기능이 제공되고 있기 때문에 Spring MVC를 통해서 제공되지 않은 JasperReports 일부 기능이 있다. 이 때 개발자는 뷰 클래스를 확장하거나, Controller 클래스에 부가 작업을 해줘야 하는데 아래의 경우의 경우 Anyframe 에서 이미 확장하여 제공하므로 개발자의 추가 작업 부담을 줄일 수 있다.

- HTML Reporting- 개발자 추가 작업 일부 필요

2.1.HTML Reporting

이미지 파일을 html 내에서 디스플레이 해주기 위해서 Spring MVC에서 제공해주는 뷰 클래스를 확장하고, Controller 클래스에 부가 작업을 해줘야 한다. Spring MVC에서 제공해주는 뷰 클래스를 확장한 것은 Anyframe 을 통해 제공될 것이므로, 개발자는 신경쓸 필요가 없다. 그러나 Controller 클래스에 부가 작업을 해야 하는 것은 개발자가 개발할 Controller내에 request 정보를 Model 객체 내에 저장시켜주는 작업으로 아래에 보여지는 코드와 같이 작성하도록 한다.

- ReportsMultiFormatView 확장 - renderReport 메소드 내에서 아래와 같이 request로부터 세션 객체를 얻어낸 후, 객체 내 어트리뷰트로 populatedReport 값을 저장한다. 이는 후에 HTML Reporting 시 이미지 파일을 디스플레이하기 위해 필요하다.(내용을 참고만 한다. 실제 개발자가 할 일은 없다.)

```
protected void renderReport(JasperPrint populatedReport, Map model,
    HttpServletResponse response) throws Exception {

    // Check for request object and set image servlet
    if (model.containsKey("requestObject")) {
        HttpServletRequest request = (HttpServletRequest) model
            .get("requestObject");
        request.getSession().setAttribute(
            ImageServlet.DEFAULT_JASPER_PRINT_SESSION_ATTRIBUTE,
            populatedReport);
    }
}
```

- **Controller** 추가 작업 - 위에서와 같이 model 객체 내에서 "requestObject" 값을 얻어오기 위해서는 Controller 클래스의 Model 객체에 request 객체를 저장해주는 작업이 필요하다. 장르별 영화 등록 현황 Reporting을 위한 ReportController 클래스의 예는 다음과 같다.

```
@Controller("reportController")
@RequestMapping("/movieReport.*")
public class ReportController{

    @RequestMapping
    public String viewReport(HttpServletRequest request, Model model) throws Exception {

        String uri = request.getRequestURI();
        String format = "html";
        try {
            format = uri.substring(uri.lastIndexOf(".") + 1);
        } catch (IndexOutOfBoundsException e) {
            // ignore - html format will be used
        }

        model.addAttribute("format", format);
        model.addAttribute("requestObject", request);

        return "reportView";
    }
}
```

```
}
```

- **JSP** 추가 작업 - JSP 상에서 Report 를 호출하기 위한 예는 다음과 같다. movieReport.html 또는 movieReport.pdf 형태로 호출한다.

```
<a href="{ctx}/reports/movieReport.html"><font size="2">HTML Report</font></a><br/>  
<a href="{ctx}/reports/movieReport.pdf"><font size="2">PDF Report</font></a><br/>
```

III. JasperReports Designer

JasperAssistant는 JasperReports를 위한 Visual Report Designer 중 하나인 상용 툴이다. Eclipse Plugin 형태로 제공되며 직관적인 Graphical한 인터페이스로 jrxml 파일 작성을 용이하게 도와준다.

3. Installation

다음은 JasperReports 설치 순서로서 다운로드 와 설치환경 그리고 Report Designer 설치 3가지 영역으로 구분하여 설명을 진행한다. Spring MVC와 통합된 형태로 JasperReports 기능을 사용하므로 Anyframe 의 Spring과 Spring MVC가 이미 설치되었다는 가정 하에서 아래 설치 과정을 진행하도록 하겠다. 그러므로 Anyframe 을 설치한 이후에 JasperReports를 설치하도록 한다.

3.1. 다운로드

다음은 JasperReports 기능을 사용하기 위해서 다운로드 받아야 하는 기본적인 파일들이다.

파일명	설명
JasperAssistant_3.1.1_Eclipse3.x.zip [http://www.jasperassistant.com/download.html]	[필수] JasperReports Report Designer Eclipse Plugin 설치 파일(Download Evaluation Version)
AdbeRdr920_ko_KR.exe [http:// get.adobe.com/kr/reader/]	[필수] PDF 파일 디스플레이를 위한 Adobe Reader 9.2 (한국어 버전) 설치 파일

3.2. 설치 환경

JasperReports를 활용한 Sample Application은 아래와 같은 환경에서 테스트되었다. 아래의 SW에 대한 설치 방법은 이 매뉴얼에서는 생략한다. 이 다음 단계를 진행하기 전에 아래 3가지는 모두 개발 환경에 설치되어야 한다. 이 매뉴얼 내에서는 JDK 1.5 버전 기준으로 설명하고 있다.

- JDK 1.5
- Web Container - Tomcat 6.0
- DataBase - Hsqldb 1.8.0.10
- Eclipse 3.5.0

본 문서에서는 기본적으로 Tomcat6.0 [<http://tomcat.apache.org/>] 및 eclipse 3.5.0 (WTP 포함) [<http://www.eclipse.org/>] 을 기준으로 설치 가이드를 진행한다.

이하 문서에서 [Eclipse Home] 이라함은 Anyframe에서 제공한 eclipse 기반 툴셋의 루트를 지칭한다. Tomcat6.0 서버는 적절한 위치에 설치한 후 Eclipse Server Runtime 으로 등록하여야 한다. 위의 다운로드 테이블의 내용 중 [필수]라고 표시된 파일들을 모두 다운로드 받았다면, AdbeRdr920_ko_KR.exe 파일을 실행시켜서 설치하도록 한다.

3.3. Report Designer 설치

JasperReports 보고서 XML 파일 작성을 용이하게 하기 위해서 여러 종류의 툴이 제공된다. 무료 툴 (iReport)과 유료 툴(JasperAssistant) 등 여러가지가 있으므로 원하는 툴을 선택하여 작성할 수 있는데, 여기서는 **JasperAssistant**를 사용 하는 것으로 한다.

[참고]

* **iReport** : <http://jasperforge.org/projects/iReport> (free)

* **JasperAssistant** : <http://www.jasperassistant.com> [<http://www.jasperassistant.com/>] (미 구매 시 21일 evaluation version 사용 가능)

현재 JasperAssistant Eclipse plugin 3.1.1 버전을 사용하고 있다. JasperAssistant의 버전 업데이트는 빈번하게 발생하고 있으므로 최신 버전을 다운로드 받아서 직접 설치를 수행하고자 한다면 아래 사이트를 참고한다.

* Homepage : <http://www.jasperassistant.com> [<http://www.jasperassistant.com/>]

* Download : <http://www.jasperassistant.com/download.html>

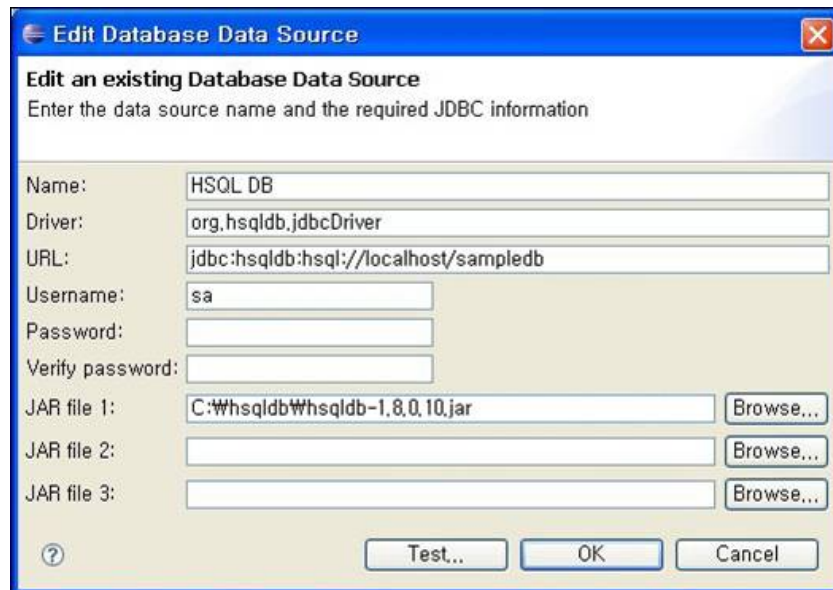
JasperAssistant를 설치하기 위해서는 2가지 방법이 사용될 수 있다. 하나는 Eclipse Update Site 기능을 이용하는 방법이고, 나머지 하나는 직접 Eclipse Plugin 형태의 배포 파일을 다운로드 받아서 Eclipse에 복사해 넣는 방법이다. 여기서는 배포 파일을 다운로드 받아서 Eclipse에 복사해 넣는 방법으로 설치하는 방법을 설명하고 있다.

[Eclipse Home] 폴더에 JasperAssistant_3.1.1_Eclipse3.x.zip 압축 파일을 풀어서 설치하도록 한다. Eclipse를 사용 중이었다면 종료시킨 후 재기동 시키도록 한다.

3.3.1. Configuration

Eclipse Plugin인 JasperAssistant를 설치한 후, 사용하기 위해서는 아래와 같이 몇 가지 Configuration 작업을 해야 한다. Configuration 창을 열기 위해서는 **Window -> Preferences -> JasperAssistant** 메뉴를 선택한다.

- License Information - 상용틀인 JasperAssistant를 구매하였다면, 이 화면에서 License Key를 입력해 넣도록 한다.
- Console - [Default 사용] JasperAssistant console 창 설정 변경
- Designer - [Default 사용] Designer Editor 설정 변경
- Data Sources - 필수 설정 으로 반드시 Report Designer를 사용하기 전에 설정해야 한다. Report 대상 데이터 소스에 접근하기 위한 정보를 입력한다. 이때, 총 4가지 타입의 데이터 소스(Empty Data Source, Database Data Source, XML Data Source, Custom Data Source) 형태를 제공하는데 이 메뉴 열에서는 Database Data Source 설정 방법에 대해서만 언급한다. HSQLDB에 접근하기 위해서 아래와 같이 Driver, URL, hsqldb jar 파일 위치 등의 정보를 입력하도록 한다.



- Export - [Default 사용] Export 설정 변경(File > Report Export.. 기능 실행 시 사용됨)으로 Report Designer 사용 시 반드시 사용되지는 않지만, Export 기능을 사용하게 된다면 PDF, Excel, HTML 파일 등의 실행 파일의 위치를 명시해줘야 한다. 각각의 Preview 항목에 C:\Program Files\Adobe\Reader 9.0\Reader\AcroRd32.exe, C:\Program Files\Internet Explorer\iexplore.exe 으로 작성한다.

JasperAssistant의 설치는 완료되었다. JasperAssistant를 활용하여 Report 파일을 만드는 방법은 Report Designer 매뉴얼 부분을 참고하도록 한다.

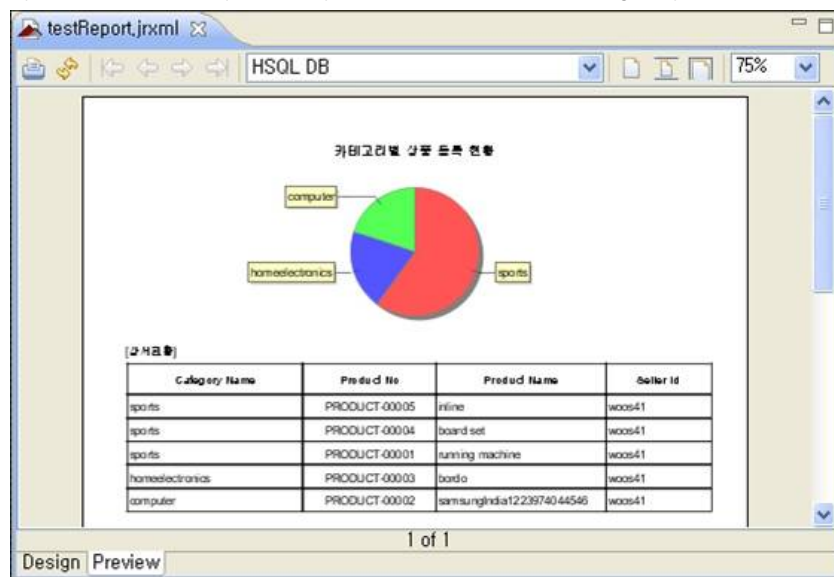
4.Design Report

여기서는 예제를 어떻게 작성하였는지에 대해서 설명할 것이다. 예제내에 카테고리별 상품 목록을 조회하는 부분을 JasperReports를 이용하여 HTML, PDF, Excel, CSV 등 다양한 형태로 Reporting할 수 있다. JasperAssistant를 이용하여 카테고리별 상품 등록 현황을 조회하는 Reporting 부분을 jrxml로 작성하고, JasperReports 엔진을 통해서 다양한 UI 형태로 보여주고 있는 것이므로, 먼저 JasperAssistant를 사용하여 jrxml 파일을 작성하는 방법부터 살펴보도록 한다. 만약 JasperAssistant 사용 방법에 대해서 이 예제를 통한 방법이 아닌, 전반적인 사용 방법이 궁금하다면 JasperAssistant 사이트에 올려진 온라인 매뉴얼 문서 [<http://www.jasperassistant.com/docs/guide/index.html>] 나 압축된 PDF 매뉴얼 문서를 참고하도록 한다.

또한 Flash로 만들어진 JasperAssistant 사용 방법 데모를 보려면 여기(Demo) [<http://www.jasperassistant.com/demo.html>] 를 참고하도록 한다. 간단한 Report 작성하는 방법을 화면과 설명을 통해 쉽게 알려주고 있으므로 처음 JasperAssistant를 접하게 되는 사용자들에게 도움이 된다.

4.1.목표 결과물

다음은 이 매뉴얼을 통해 작성된 카테고리별 상품 등록 현황 디자인 파일을 JasperAssistant의 Preview 기능을 사용하여 디자인(jrxml) 작성 결과를 확인해 본 모습이다. 이처럼, 카테고리별 상품 등록 현황을 테이블(Table) 형태와 파이 차트(Pie Chart) 형태로 보여지도록 디자인(jrxml) 파일을 작성해보자.



4.2.디자인 파일(JRXML) 작성

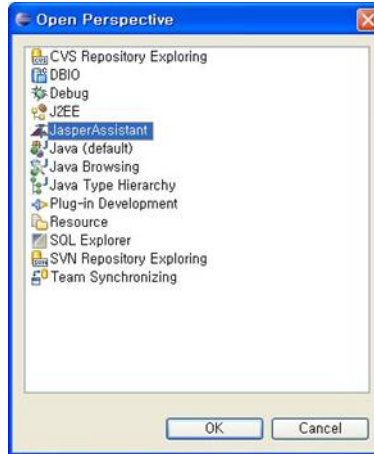
위와 같은 목표 결과물을 만들어 내기 위해서는 Database에 저장된 카테고리별 상품의 목록을 조회하여 다양한 UI로 Reporting 할 수 있도록 jrxml 파일을 작성해야 한다.

여기서 작성된 예제는 아래와 같은 환경에서 테스트되었다. 이 다음 단계를 진행하기 전에 아래 항목들은 모두 개발 환경에 설치되어야 한다.

- Eclipse 3.5.0
- JasperAssistant 3.1.1
- DataBase - Hsqldb 1.8.0.10

4.2.1.Step 1 : Open JasperAssistant Perspective

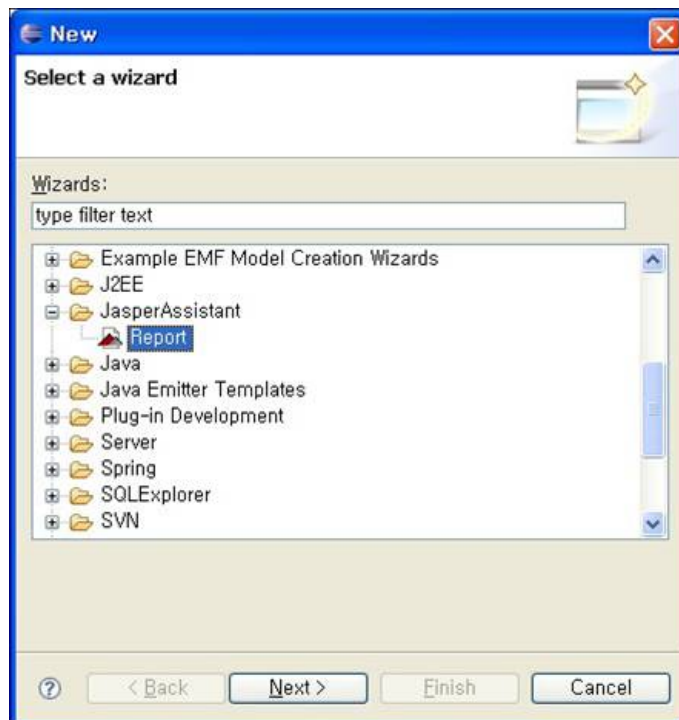
JasperAssistant는 Report 작성을 위해서 Eclipse Perspective를 제공한다. 이 Perspective은 Report 작성 시 최적의 Eclipse Workbench 레이아웃 및 필요한 모든 뷰를 제공한다. JasperAssistant Perspective를 열기 위해서는 **Window ->Open Perspective ->Other... ->JasperAssistant** 메뉴를 선택한다.



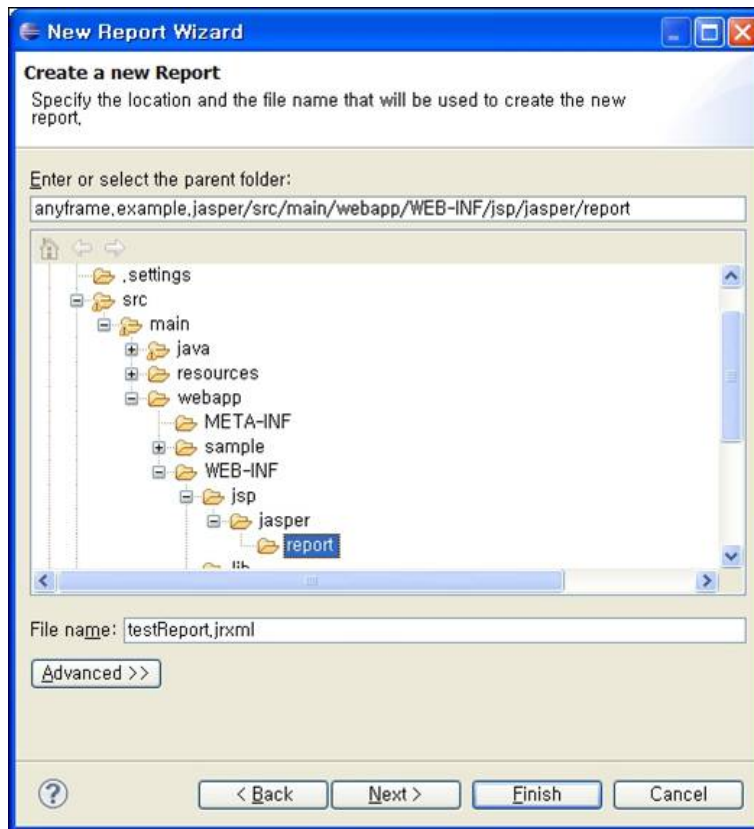
4.2.2.Step 2 : Create a new Report

이제 새로운 Report를 작성해보도록 하자. 우선 src/webapp/WEB-INF/reports 폴더에 신규 Report 파일 (jrxml)을 생성한다. 아래 화면들은 카테고리별 상품 등록 현황 Report 파일을 위해서 작업한 내용들이다. 만약 추가적으로 새로운 Report 파일을 작성하고자 한다면 아래 내용을 참고하여 작성하면 된다.

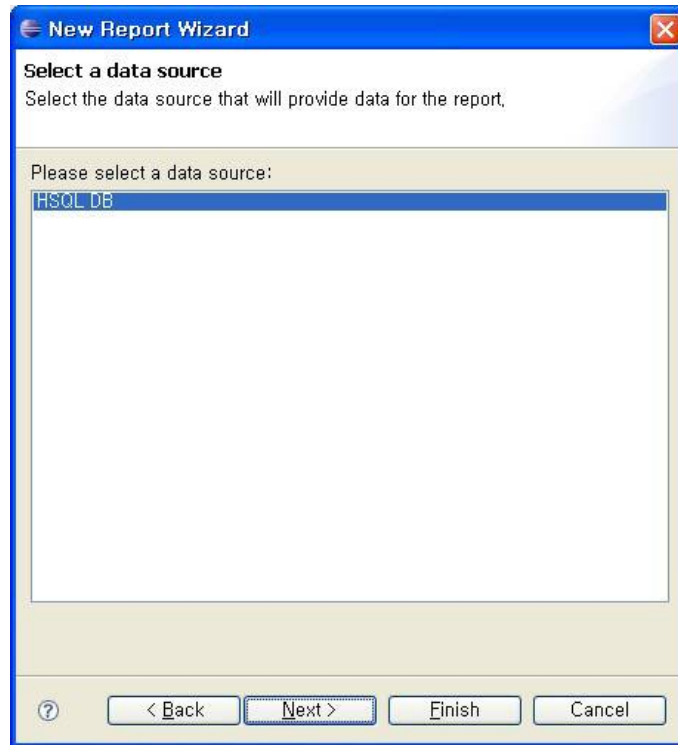
- **Select a wizard** - Report wizard를 선택한다. 이때 JasperAssistant Perspective 화면이라면 좌측에 Navigator 뷰가 나타날 것이다. src/main/webapp/WEB-INF/jsp/jasper/report 폴더를 마우스로 선택한 상태에서 File > New 메뉴를 선택하면 Report 라는 하위 메뉴가 보이게 된다. 이 메뉴를 이용하면 바로 아래 단계로 이동하게 된다. 혹은 File > New > Other... 메뉴를 선택하면 아래 그림과 같은 화면이 나오게 되고, 이때 JasperAssistant > Report wizard를 선택하도록 한다.



- **New Report Wizard : Create a new report** - 신규 Report 파일을 생성한다. 이때 파일 확장자를 jrxml
으로 작성하도록 한다. 카테고리별 상품 등록 현황 Report 파일명은 testReport.jrxml로 하도록 하고,
anyframe-jasperreports-pi/src/main/webapp/WEB-INF/jsp/jasper/report 폴더 하위에 Report 파일이
저장되고 있음을 확인한다.



- **New Report Wizard : Select a data source**- Report에 사용될 Data가 저장된 data source를 선택한다.
Preferences 항목에 정의된 Data Source들이 목록으로 보여진다. 제공되는 샘플은 HSQL DB를 사용하
므로 목록 중에 HSQL DB를 선택하도록 한다. Data Source를 추가하고자 하면, Window ->Preferences
->JasperAssistant ->Data Source 메뉴에서 추가하면 된다.

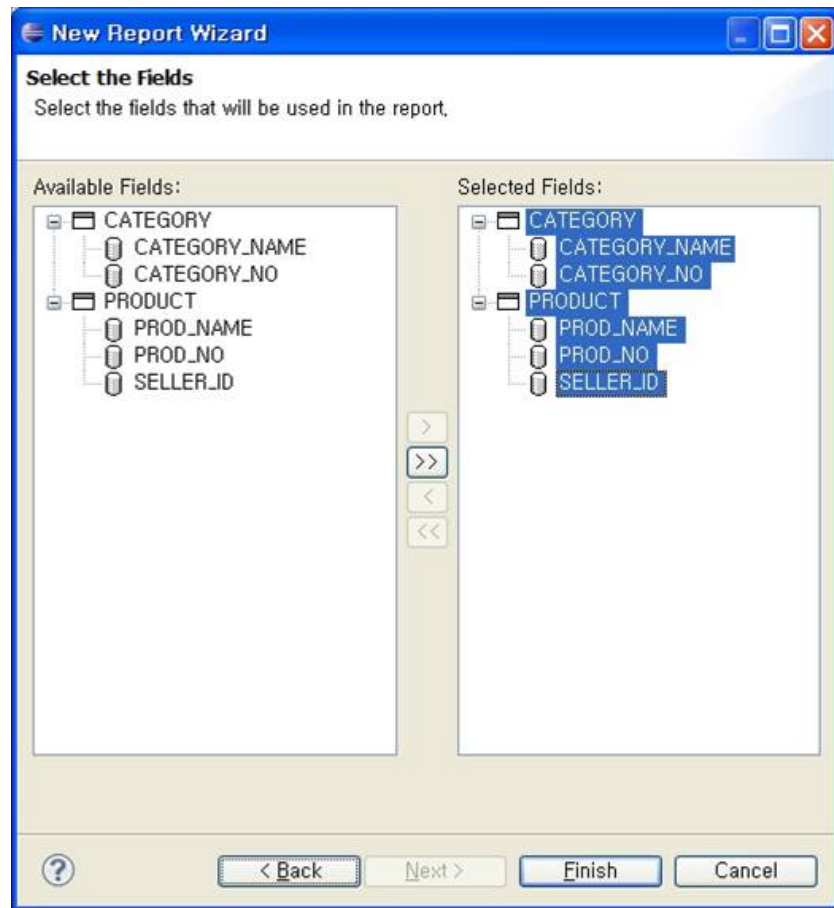


- **New Report Wizard : Specify the SQL query** - Report에 사용될 Data를 얻어오는 SQL 쿼리를 작성한다. 카테고리별 상품 등록 목록을 조회하여 보여줄 Report를 작성할 것이므로 SQL 쿼리에 다음과 같이 입력하도록 한다.

```
select category.CATEGORY_NO, category.CATEGORY_NAME, product.PROD_NO, product.PROD_NAME,
       product.SELLER_ID
from PRODUCT product, CATEGORY category
where product.CATEGORY_NO = category.CATEGORY_NO
order by category.CATEGORY_NO
```



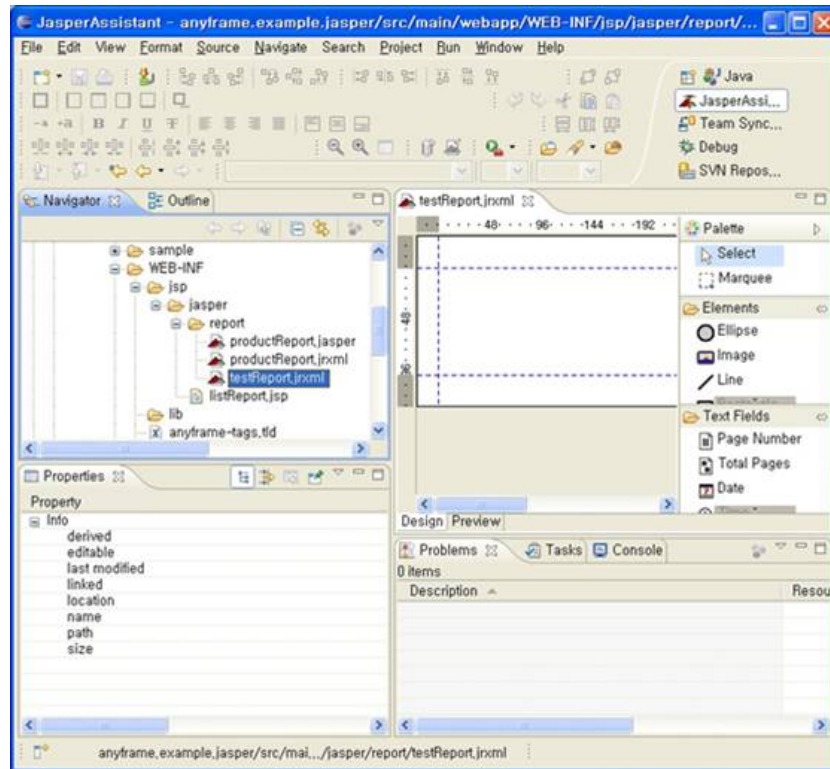
- **New Report Wizard : Select the Fields** - Report에 사용될 Field를 선택한다. DB로부터 얻어온 Table의 컬럼을 모두 (CATEGORY_NAME, CATEGORY_NO, PROD_NAME, PROD_NO, SELLER_ID) 선택하도록 한다.



신규 Report 파일(jrxml) 생성 작업이 완료되었다. 아래 단계를 통해서 디자인 작업을 계속 진행하도록 한다.

4.2.3.Step 3 : Design a report using Palette

Step 2에서 생성한 Report 파일을 JasperAssistant Editor를 이용하여 Open하면, 우측에 Design/Preview Tab과 Palette가 나타난다.



이 Palette 상의 Elements와 Text Fields들을 이용하여 Report 파일을 디자인한다. 우선 Report XML 파일의 구조를 살펴보도록 한다. Report 파일은 section으로 구성되어 있는데 Section이란 height 별로 각각에 고유한 이름을 가지고 일정한 영역을 가지는 Report 파일의 화면을 나누는 개념이다. Section은 Band라는 이름으로 불리우는데 현재 JasperReports 사용 시 혼용해서 사용되고 있다. 현재 JasperAssistant Editor 메뉴 상에서 사용되는 이름은 Band이므로 앞으로는 Band라고 통일해서 부르도록 한다. Band는 가로방향으로 나뉘어 각각에 고유의 이름이 명명된 일정한 영역을 뜻한다. 아래 그림은 Band 구조를 보여주고있다. 기본적으로 Title, Last Page Footer, Summary Band를 제외한 나머지 Band들은 Detail Band를 기준으로 상하로(Header와 Footer) 구분되어 짝을 이루고 있고, 각각의 Band는 고유의 목적(기능)이 있다.

	title	
	pageHeader	
	columnHeader	
	detail	
	columnFooter	
	pageFooter	
	lastPageFooter	
	summary	

Band 명	설명
Background band	보고서의 배경 설정을 할 수 있다.
Title band	가장 먼저 보여주는 band로 보고서 전체 페이지중 단 한번만 출력된다. 주로 보고서의 Title을 기재하는데 사용된다.
Page header band	한번 정의되면 보고서 전체 페이지의 헤더 부분에 똑같은 위치와 크기로 페이지마다 반복 출력된다.
Page footer band	한번 정의되면 보고서 전체 페이지의 하단 부분에 똑같은 위치와 크기로 페이지마다 반복 출력 된다.
Column header band	각 Detail band의 column 항목의 상단 영역으로 사용된다. 이 영역 역시 페이지마다 출력 된다.
Column footer band	각 Detail band의 column 항목의 하단 영역으로 사용된다. 이 영역 역시 페이지마다 출력 된다.
Group header band	사용자가 임의로 만든 band로 0개 이상을 만들수 있다. Detail band 상단에 위치한다.
Group footer band	사용자가 임의로 만든 band로 0개 이상을 만들수 있다. Detail band 하단에 위치한다.
Detail band	가장 핵심이 되는 band이다. Detail Band는 주로 실제 Query 를 이용해 수집된 데이터들을 나열해 출력할 수 있도록 하는 Band로 모든 Band들이 Detail Band를 중심으로 구성되고 설계된다고 해도 과언이 아니다. 여기서 Query를 이용해 수집된 데이터는 Detail Band 뿐만 아니라 다른 Band 에서도 사용할 수 있지만, Detail Band와 틀린 점은 Detail Band에서는 수집된 한 개 이상의 모든 Rows를 출력할 수 있지만 Detail Band를 제외한 다른 Band 에서는 첫 번째 Row에 있는 Column 값들만 출력할 수 있다.
Last Page Footer	다른 일반 Page footer와 달리 보고서의 맨 마지막 페이지에 특별한 결과를 나타내고자 할 때 사용하는 Band로 Title Band처럼 보고서 전체 페이지중 딱 한번만 출력할 수 있다.
Summary band	Report footer Band라고도 하며 보고서 전체 페이지중 맨 마지막 페이지에 출력되는 Band로 주로 총 합계 등을 나타내고자 할 때 사용한다.

샘플 예제인 카테고리별 상품 등록 현황 Report 파일은 Title, Page Header, Column Header, Detail 을 사용하여 작성되었다. testReport.jrxml파일을 신규 생성하면, Default로 Detail band만 존재한다. 그러므로 추가되는 band는 Add Band 메뉴를 이용하여 추가시키도록 한다.

• Title band 작성

testReport.jrxml 파일이 Open되었을때 좌측에 Outline 뷰에서 루트 항목(기본적으로 Unnamed라고 표기됨)을 선택 후 마우스 우측 버튼을 클릭하여 Add Band > Add Title band 메뉴를 선택하도록 한다. 또는 Editor에서 점(dot)가 찍혀있는 얇은 상하좌우 공란을 선택한 후 마우스 우측 버튼을 클릭하면 Add Band 메뉴를 사용할 수 있다. 우측 Editor 화면에 Title band가 생성되고 Palette의 Elements 항목 중 Static Text를 드래그앤드롭 하여 Title band 위에 놓는다. Static Text Element의 Properties 항목 중 Static Text > Text Property의 Value에 카테고리별 상품 등록 현황 이라고 입력해 넣는다. 또한 Reporting 시 보기 좋게 하기 위해서 아래와 같이 Property의 Value를 설정하도록 한다.

Property	Value	설명
Static Text > Text	카테고리별 상품 등록 현황	타이틀 명
Text Font > Font Size	12	타이틀 글자 크기 설정
Text Font > PDF Embedded	true	PDF 형태로 Reporting 하는 경우 한글 지원하기 위해 설정
Text Font > PDF Encoding	UniKS-UCS2-H	PDF 형태로 Reporting 하는 경우 한글 지원하기 위해 설정하는데

Property	Value	설명
		글자 배열 방식에 따라 UniKS-UCS2-H(horizontal) 혹은 UniKS-UCS2-V(vertical)를 선택
Text Font > PDF Font Name	HYGoThic-Medium	PDF 형태로 Reporting 하는 경우 한글 지원하기 위해 설정하며, 이때 다른 글자체 선택 가능함

Editor 기능을 이용하여 Static Text Box의 크기를 크게 하고 글자를 가운데 정렬시켜 놓는다.



• **Column Header band 작성**

testReport.jrxml파일이 Open되었을때 좌측에 Outline 뷰에서 루트 항목(기본적으로 Unnamed라고 표기됨)을 선택 후 마우스 우측 버튼을 클릭하여 Add Band > Column Header band 메뉴를 선택하도록 한다. 그런 다음 Palette의 Elements 중 Static Text 항목을 드래그앤드롭하여 [상세현황], Category Name, Product No, Product Name, Seller Id 이라는 헤더 정보를 작성한다. 이 하위에 실제 카테고리 별 상품 등록 현황이 해당 필드에 맞게 Reporting 될 것이다. Editor의 툴바에서 제공하는 기능을 이용하여 굵은 글씨로 변경시키고, 테두리 선(Toggle Border 기능 사용)을 만들도록 한다. 작성하고 나면 다음과 같이 페이지 헤더 부분이 완성된 것을 확인할 수 있다. (PDF 출력을 위하여 Title band 작성을 참고하여 속성을 부여한다.)



• **Detail band 작성**

Default로 생성되는 band이므로 새로 band를 추가할 필요 없다. testReport.jrxml파일이 Open되었을 때 좌측에 Outline 뷰에서 루트 항목(기본적으로 Unnamed라고 표기됨) 하위의 Fields 항목을 열어보면 위 Step 3에서 선택했던 Field들이 나온다. 이 Field들을 드래그앤드롭하여 Detail band 영역으로

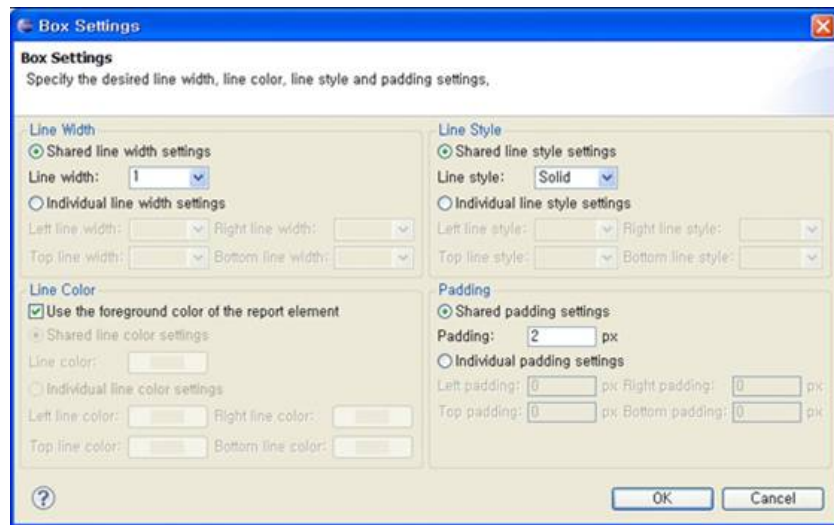
가지고 오면 아래와 같이 $\{F\}$ 형태로 보이게 된다. 실제 Data Source로부터 데이터를 받아와서 이 위치에 Reporting 하게 된다. 테두리 선 등 필요한 작업을 Editor의 툴바 기능을 이용해서 수행한다.(PDF 출력을 위하여 Title band 작성을 참고하여 속성을 부여한다.)



Text Field 입력 란 박스의 Padding 값을 설정하기 위한 Property 설명과 해당 화면이다.

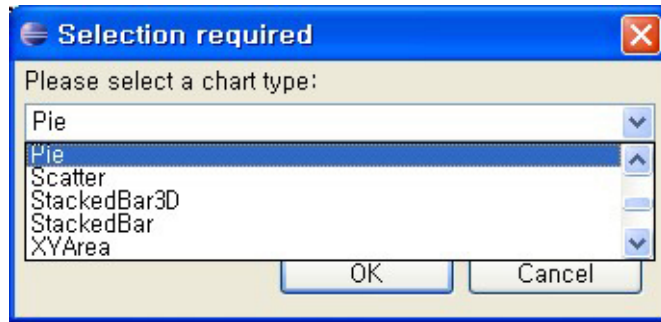
Property	Value	설명
Text > Box	(2,2,2,2) : 직접 입력 불가능	Properties 탭에서 Text > Box 우측의 [...] 버튼을 클릭하면 아래와 같이 설정할 수 있는 팝업창이 나온다.

Box Settings 팝업창에서 Padding 항목을 수정해주면 된다. 여기서는 Shared padding settings 항목을 선택하고 Padding 값을 2 px로 주었다.



• Page Header band 작성

testReport.jrxml파일이 Open되었을때 좌측에 Outline 뷰에서 루트 항목(기본적으로 Unnamed라고 표기됨)을 선택 후 마우스 우측 버튼을 클릭하여 Add Band > Page Header band 메뉴를 선택하도록 한다. 이 Page Header band 영역에는 파이 차트를 추가해넣는다. 우측에서 Palette의 Elements 중 Chart를 클릭한 후 band 영역을 다시 클릭한다. 그러면 다음과 같이 어떤 타입의 차트를 선택할 지 묻는 팝업창이 나온다. 여기서 Pie를 선택하도록 한다. 만약 클릭으로 Chart를 선택하지 않고 위에서처럼 드래그앤드롭 형태로 Chart를 갖고 오면 Default로 Area 타입의 차트가 적용된다. 그러므로 차트 타입을 선택하기 위해서는 클릭으로 Chart Element를 선택하도록 한다.



좌측에 Outline 뷰에서 루트 항목(기본적으로 Unnamed라고 표기됨)하위의 Summary > Chart_숫자 > Pie Dataset 항목을 선택하면 Properties 뷰에서 Dataset에 대한 설정을 할 수 있다.

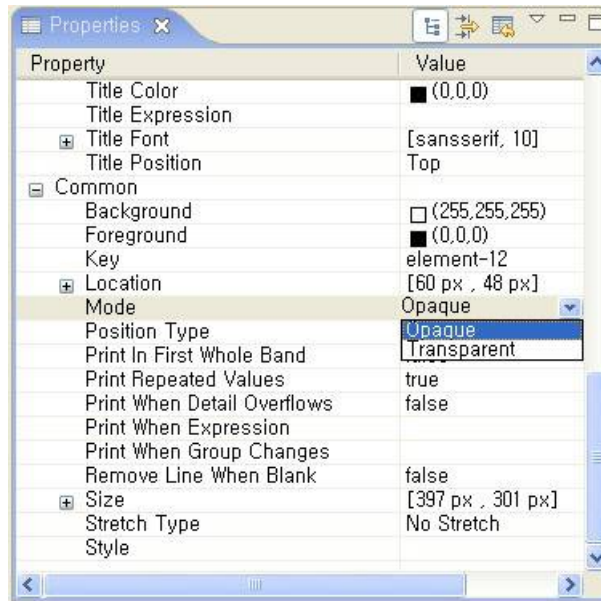
Property	Value	설명
Pie Dataset > Key Expression	`\${CATEGORY_NAME}`	파이 차트 작성 시 키가 되는 필드 정보를 입력
Pie Dataset > Value Expression	`\${CategoryGroup_COUNT}`	Number 형 데이터로 변경 가능한 데이터 필드 정보만 입력 가능함
Pie Dataset > Label Expression		공란으로 두면 기본적으로 Key = Value 형태로 디스플레이됨

좌측에 Outline 뷰에서 루트 항목(기본적으로 Unnamed라고 표기됨)하위의 Group 항목을 선택한 후 마우스 우측 버튼을 클릭하여 Add Group 을 선택하여 다음과 같이 Group 을 추가한다.

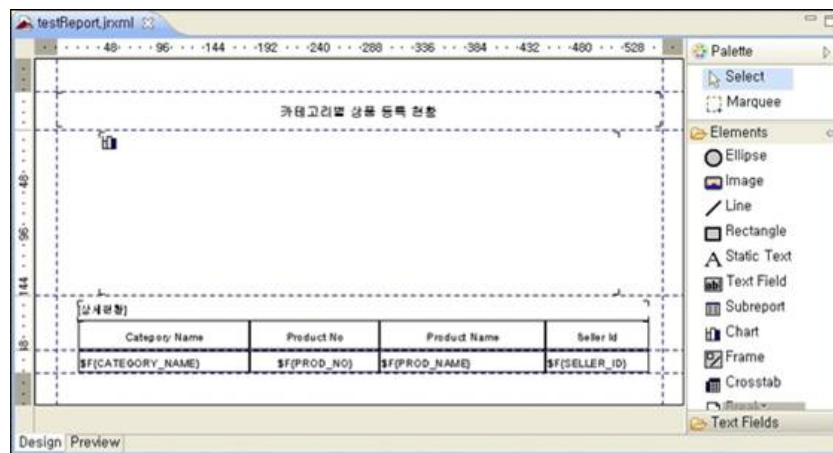
Property	Value	설명
Group > Expression	Expression `\${CATEGORY_NO}`	그룹 조건을 입력
Group > Name	CategoryGroup	그룹 조건명을 입력 (Pie Dataset > Value Expression 과 일치)

좌측에 Outline 뷰에서 루트 항목(기본적으로 Unnamed라고 표기됨)하위의 Summary > Chart_숫자 항목을 선택한 후 Properties 뷰에서 Mode를 변경시켜 준다.

Property	Value	설명
Common > Mode	Opaque	파이 차트를 HTML로 Reporting 시 Mode를 Transparent로 하면 차트 이미지 주위로 불필요한 바탕색이 생기므로 이를 없애기 위해서 Opaque를 선택함 PDF, Excel 등 타 Reporting 형태일 경우에는 상관없음

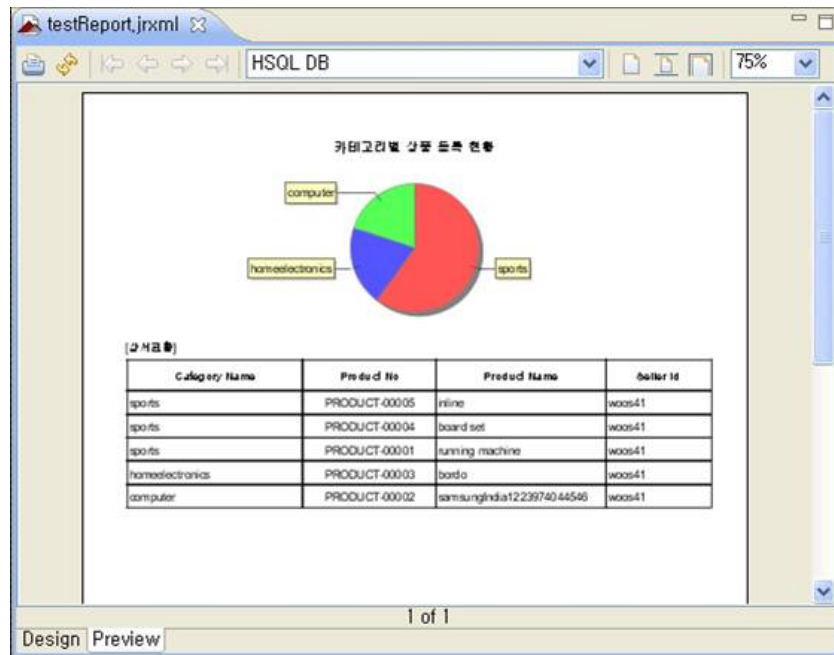


파이 차트 작성까지 모두 끝내고 나서 Editor 상의 Design Tab을 통해 보면 다음과 같은 모습으로 보인다. Preview Tab으로 이동하지 않으면 파이 차트 상에 데이터가 들어가 있는 형태를 볼 수 없으며, 에러 발생 시에도 확인이 불가능하므로 Design Tab에서의 작업이 모두 끝나면 다음 단계인 Preview를 진행시킨다.



4.2.4.Step 4 : Preview Report

디자인 Report 파일 작성이 모두 완료되었다면 JasperAssistant에서 제공되는 Preview 기능을 사용하여 제대로 작성되었는지를 확인해본다. 만약 Report 파일에 에러가 존재한다면 Preview 기능을 수행시킬 때 확인해볼 수 있을 것이다. Preview 결과가 이 매뉴얼 상단의 목표 결과물과 동일한지 확인해본다.



이로써 JasperAssistant를 활용한 jrxml 파일 작성이 완료되었다. 이제 샘플 웹 어플리케이션에서 해당 Reporting 파일을 어떻게 호출해서 보여주는지 확인해본다.