

리눅스 데스크탑 솔루션 개발지원 Firefox 테마 패키징 및 스킨 구현

한국소프트웨어진흥원
공개SW기술지원센터

<Revision 정보>

| 일자 | VERSION | 변경내역 | 작성자 |
|--------------|---------|-------|-----|
| 2007. 08. 13 | 0.1 | 초기 작성 | 박준완 |
| | | | |

목 차

| | |
|-----------------------------------|----|
| 1. 문서 개요 | 4 |
| 가. 문서의 목적 | 4 |
| 나. 본 문서의 사용방법 | 4 |
| 다. 참고 사항 | 4 |
| 2. Firefox 테마 패키징 및 스킨 만들기 | 5 |
| 가. 테마 패키징 하기 | 5 |
| 가. 테마 파일의 구조 | 5 |
| 나. Install.rdf | 5 |
| 다. Install.rdf에 들어가야 하는 속성들 | 6 |
| 라. 샘플 Install.rdf 파일 | 6 |
| 나. Firefox의 스킨 만들기 | 8 |
| 가. 기본 설정 사항 | 8 |
| 나. 테마 압축 풀기 | 8 |
| 다. Classic.jar의 위치 찾기 | 8 |
| 라. Classic.jar 파일의 내부 | 8 |
| 마. 새로운 테마 설치하기 | 9 |
| 바. 필요한 파일 복사하기 | 9 |
| 사. 설치 파일 만들기 | 9 |
| 아. JAR 파일로 리패키징 하기 | 10 |
| 자. 웹에서 바로 설치할 수 있도록 만들기 | 11 |

< 그림 차례 >

| | |
|-------------------------------|----|
| 1. JAR 파일 구조 | 5 |
| 2. Install.rdf manifest | 6 |
| 3. Install.rdf Sample | 7 |
| 4. targetApplication 속성 | 8 |
| 5. 변경 사항 | 9 |
| 6. Install.rdf | 10 |
| 7. Firefox의 버전 범위 | 10 |
| 8. My_Theme 내용 | 11 |
| 9. 설치 스크립트 | 11 |

1. 문서 개요

본 문서는 KIPA 공개S/W 기술지원 센터에서의 리눅스 데스크탑 솔루션 개발지원 결과를 보고하기 위해 제작되었다.

가. 문서의 목적

다음과 같은 세부적인 목적을 달성하기 위하여 작성되었다.

0 Firefox 테마 패키징 방법

0 Firefox 스킨 구현 방법

나. 본 문서의 사용방법

다음과 같은 방법으로 사용할 수 있다.

0 개인이 만든 테마를 Firefox 테마에 맞게 패키징 할 수 있다.

0 개발 지원 결과 내역을 참조한다.

다. 참고사항

0 개발지원시 시스템 관련사항, 기술적 배경 등 참고사항들을 기술한다.

2. Firefox 테마 패키징 및 스킨 만들기

파이어폭스나 쉐더버드의 테마를 만들 때에는 Cascading Sheets(CSS) 에 대한 지식이 필수적이며, XBL에 대한 지식도 필요할 수 있다. 또한, 그래픽 디자인 및 미적 감각도 필요하다. 이 문서는 디자인 및 미적 감각을 배제하고 사용자가 쉽게 자신의 테마를 만들어 패키징 하고 스킨을 만들 수 있는 방법에 대해 기술한다.

가. 테마 패키징 하기

가. 테마 파일의 구조

파이어폭스/쉐더버드의 테마는 다음과 같은 구조의 JAR 파일로 포장된다.

```
theme.jar:
  install.rdf
  contents.rdf
  preview.png
  icon.png
  browser/files
  global/files
  mozapps/files
  communicator/files
  ...
```

<그림 1> JAR 파일 구조

- * 테마에 사용되는 크롬을 등록하기 위해 chrome.manifest (Firefox/Thunderbird 1.5의 경우) 나 contents.rdf 파일을 디렉토리 최상위단에 만들어 주어야 한다. 또한 테마 윈도우에 나타낼 메타데이터를 정의하기 위한 install.rdf manifest 파일도 함께 만들어야 한다.
- * preview.png 는 테마 윈도우의 스크롤 가능한 미리보기 영역에 보여질 이미지이다. 이미지의 사이즈는 어떤 크기로도 가능하다.
- * icon.png 는 32x32 크기의 PNG (알파 투명도 값을 가질 수 있는)파일이다. 이는 테마 윈도우의 테마 목록에서 보여진다.

나. Install.rdf

install.rdf manifest 는 다음과 같이 만들 수 있다:

```
<?xml version="1.0"?>
<RDF xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:em="http://www.mozilla.org/2004/em-rdf#">
  <Description about="urn:mozilla:install-manifest">
    <em:type>4</em:type>
    more properties
  </Description>
</RDF>
```

<그림 2> Install.rdf manifest

다. install.rdf 에 들어가야 하는 속성들

install.rdf 파일에는 다음과 같은 속성들이 들어가야 한다.

- * em:id
- * em:version
- * em:type
- * em:targetApplication
- * em:name
- * em:internalName

install.rdf 에 선택적으로 들어갈 수 있는 속성들

- * em:description
- * em:creator
- * em:contributor
- * em:homepageURL
- * em:updateURL

주의 : 사용자가 만든 테마가 <http://addons.mozilla.org> 웹사이트에 게시된다면 updateURL 속성은 포함시키면 안된다.

라. 샘플 install.rdf 파일

```
<?xml version="1.0"?>

<RDF xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:em="http://www.mozilla.org/2004/em-rdf#">

  <Description about="urn:mozilla:install-manifest">
    <em:id>{18b64b56-d42f-428d-a88c-baa413bc413f}</em:id>
    <em:version>1.0</em:version>
    <em:type>4</em:type>

    <!-- Target Application this extension can install into,
         with minimum and maximum supported versions. -->
    <em:targetApplication>
      <Description>
        <em:id>{ec8030f7-c20a-464f-9b0e-13a3a9e97384}</em:id>
        <em:minVersion>0.8</em:minVersion>
        <em:maxVersion>0.9</em:maxVersion>
      </Description>
    </em:targetApplication>

    <!-- Front End MetaData -->
    <em:name>New Theme 1</em:name>
    <em:description>A test theme for Firefox</em:description>
    <em:creator>Ben Goodger</em:creator>
    <em:contributor>John Doe</em:contributor>
    <em:homepageURL>http://www.bengoodger.com/</em:homepageURL>

    <!-- Front End Integration Hooks (used by Theme Manager)-->
    <em:internalName>newtheme1</em:internalName>
  </Description>
</RDF>
```

<그림 3> Install.rdf Sample

아래의 것은 통상적인 타겟 어플리케이션의 GUID이다. 개발자는 GUID를 자신이 만든 targetApplication 속성에 정의할 수 있다:

| | |
|-------------|--|
| Firefox | {ec8030f7-c20a-464f-9b0e-13a3a9e97384} |
| Thunderbird | {3550f703-e582-4d05-9a08-453d09bdfdc6} |
| Sunbird | {718e30fb-e89b-41dd-9da7-e25a45638b28} |

<그림 4> targetApplication 속성

나. Firefox의 스킨 만들기

가. 기본 설정 사항

최신 버전의 파이어폭스와 DOM Inspector를 다운로드하여 설치한다.

나. 테마 압축 풀기

이미 기존에 파이어폭스에서 사용할 수 있도록 만들어진 어떤 테마를 살펴보다도, 파이어폭스의 기본 테마인 Winstripe와의 일관성을 발견할 수 있다. 이 테마는 파이어폭스의 설치 디렉토리에 존재하는 classic.jar 파일 안에 들어 있다. jar 파일은 zip과 확장자만 다르지 그 압축 구조는 같기 때문에 일반적인 zip 압축 프로그램을 사용하여 풀어볼 수 있다. 만약에 압축 프로그램에서 jar를 지원하지 않는다면 classic.zip과 같이 파일 이름을 바꾸고 압축을 해제할 수 있다.

다. Classic.jar 의 위치 찾기

Linux: /usr/lib/MozillaFirefox/chrome/classic.jar

라. Classic.jar 파일의 내부

classic.jar의 압축을 풀면 skin 디렉토리, preview.png, 그리고 icon.png 파일이 들어 있다.

skin

이 디렉토리 안에는 스킨을 이용하는 모든 훌륭한 재료들을 포함하는 classic 디렉토리가 있다.

classic

classic 디렉토리는 다음과 같은 디렉토리들을 포함한다.

browser

browser 디렉토리에는 툴바 아이콘들이 들어 있다. 북마크 관리자나 설정창에 사용되는 아이콘도 여기에 들어 있다.

global

global 디렉토리는 테마에서 특별히 중요한 부분으로, 브라우저의 외관을 설정하는 중요한 CSS 파일을 담고 있다.

mozapps

mozapps 디렉토리에는 브라우저와 확장 관리자, 업데이트 마법사를 위한 스타일 및 아이콘이 들어 있다.

help

이 디렉토리에는 헬프 윈도우에 필요한 테마 파일들이 들어 있다.

communicator

이 디렉토리의 내용은 사용되지 않는다.

마. 새로운 테마 설치하기

라이브 에디트를 설정하는 것에 부분부분 어려운 점이 있기 때문에, 일단은 직접 파이어폭스 테마를 만들며 변화하는 모습을 지켜보기 전에 기존의 테마를 설치할 수 있는 상태로 리패키징하는 법을 먼저 배우는 것이 좋다.

바. 필요한 파일 복사하기

먼저 첫번째 과정은 필요한 모든 파일을 올바른 디렉토리 구조로 옮겨넣는 것이다. My_Theme라는 이름의 디렉토리를 하나 만든 후, 이 디렉토리로 들어가 browser, global, communicator, help, 그리고 mozapps와 같은 이름의 디렉토리들을 만들어 classic.jar 파일을 열어 보았을 때와 같은 구조로 구성한다. 여기에 icon.png 파일과 preview.png 파일을 만들어 넣는다.

사. 설치 파일 만들기

Contents.rdf

contents.rdf 파일은 스킨에 대해 기술해 놓은 작은 XML 데이터베이스로 이루어져 있다. 이 파일에는 "My_Theme"의 구성 요소들을 찾아볼 수 있고, 이 내용을 새로 만드는 테마에 맞추어 이름을 바꿀 수도 있다.

패키지 섹션은 변경하는 브라우저의 컴포넌트를 목록으로 나타낸다. 모질라에서도 이 스킨을 사용하려면 모질라에 맞추기 위해 몇 줄을 추가하거나 바꾸어야 한다. 하지만 지금의 목록은 변경한 모든 것을 반영하고 있으므로, 이 섹션에서 이전 것을 사용하였던 부분에 대해서만 여기서 굵은 글씨로 표시된 부분에 이름과 버전이 맞게 변경해주면 된다.

```
<RDF:li resource="urn:mozilla:skin:My_Theme:communicator"/>
<RDF:li resource="urn:mozilla:skin:My_Theme:editor"/>
<RDF:li resource="urn:mozilla:skin:My_Theme:global"/>
<RDF:li resource="urn:mozilla:skin:My_Theme:messenger"/>
<RDF:li resource="urn:mozilla:skin:My_Theme:navigator"/>
```

<그림 5> 변경 사항

install.rdf

install.rdf를 복사하여 My_Theme 디렉토리 안에 집어넣는다. 이 파일은 스킨에 대해 기술해 놓은 작은 XML 데이터베이스 파일이다.

```
<Description about="urn:mozilla:install-manifest">
<em:id>{Themes_UUID}</em:id>
<em:version>Themes_Version</em:version>
```

<그림 6> Install.rdf

첫 섹션은 테마를 위해 UUID를 확립하고 버전을 부여하는 부분을 포함한다. 그리고 이 스킨을 사용할 수 있는 파이어폭스의 버전 범위를 명기해야 한다. 아래의 예를 볼 수 있다.

```
<em:targetApplication>
<Description>
<!-- Firefox's UUID -->
<em:id>{ec8030f7-c20a-464f-9b0e-13a3a9e97384}</em:id>
<em:minVersion>Min_FF_Version</em:minVersion>
<em:maxVersion>Max_FF_Version</em:maxVersion>
</Description>
</em:targetApplication>
```

<그림 7> Firefox의 버전 범위

위와 같은 문제들을 해결하였다면 버전으로 인하여 테마가 깨지거나 엉망이 되는 문제는 피할 수 있다.

CSS 파일

디렉토리에 들어있는 CSS 파일은 브라우저의 어느 부분에 그림을 넣고, 버튼이나 컨트롤의 테두리나 두께 등을 정의하여 어떻게 보일지를 지정하는 데 사용된다. 표준 버튼을 바꾸려면 global 디렉토리로 들어가 button.css 파일을 열어 'button {'으로 시작되는 섹션을 찾는다. 이 섹션은 기본 상태, 그러니까 마우스를 위로 올리거나 선택하지 않았고, 그렇다고 사용중지 상태도 아닌 일반적인 버튼의 모양을 정의하는 섹션이다. background-color: 부분에 DarkBlue를 입력하고, color: 부분에 White 라고 입력한 뒤 파일을 저장한다.

아. JAR 파일로 리패키징하기

아래와 같은 디렉토리 구조를 JAR 파일로 리패키징하면 된다.

```
/browser/*
/communicator/*
/global/*
/help/*
/mozapps/*
/contents.rdf
/install.rdf
/icon.png
/preview.png
```

<그림 8> My_Theme 내용

이 구조가 들어있는 부모 디렉토리인 My_Theme째 묶어야 설치할 때 에러 메시지가 생기지 않는다.

자. 웹에서 바로 설치할 수 있도록 만들기

JAR 파일을 웹에 올리고, 클릭하면 바로 설치할 수 있도록 만들기 위해 자바 스크립트를 구성한다.

```
<a href='javascript:InstallTrigger.installChrome(InstallTrigger.SKIN,
"myskin.jar", "My Skin Theme")'>누르면 스킨이 설치됩니다</a>
```

<그림 9> 설치 스크립트

하드 디스크에 JAR 형태로 존재하는 스킨 파일을 설치하려면 웹 인스톨러를 사용하여 해당 파일을 업로드하고 바로 설치할 수 있다. 혹은 모질라에서 테마 윈도우를 띄우고 jar 파일을 끌어당겨 놓을 수도 있다.