

Chapter 2

The Mechanism behind Extensions



How “Extensions” are implemented



**How is it
different from
plugins**



Plugin

is a small application that interacts with the main application to provide a certain function.



Firefox Extension is like
dynamically applied
software patch



Patch?



```

Index: layout/xul/base/src/nsNativeScrollbarFrame.cpp
=====
RCS file: /cvsroot/mozilla/layout/xul/base/src/nsNativeScrollbarFrame.cpp,v
retrieving revision 1.22
diff -u -6 -p -r1.22 nsNativeScrollbarFrame.cpp
--- layout/xul/base/src/nsNativeScrollbarFrame.cpp31 Dec 2004 01:13:26 -0000    1.22
+++ layout/xul/base/src/nsNativeScrollbarFrame.cpp28 Jul 2005 22:06:05 -0000
@@ -342,6 +342,68 @@ nsNativeScrollbarFrame::Hookup()
     if (!curpos || error)
         return;

     scrollbar->SetPosition(curpos);
 }

+NS_IMETHODIMP
+nsNativeScrollbarFrame::Paint(nsPresContext*      aPresContext,
+                               nsIRenderingContext& aRenderingContext,
+                               const nsRect&        aDirtyRect,
+                               nsFramePaintLayer    aWhichLayer,
+                               PRUint32             aFlags)
+{
+    if (NS_FRAME_IS_UNFLOWABLE & mState) {
+        return NS_OK;
+    }
+
+    if (NS_FRAME_PAINT_LAYER_BACKGROUND == aWhichLayer) {
+        PaintSelf(aPresContext, aRenderingContext, aDirtyRect);
+    }
+
+    return nsBoxFrame::Paint(aPresContext, aRenderingContext, aDirtyRect,
+                             aWhichLayer);
+}

```



**Applied
dynamically?**



Actually,



Key technologies used in Firefox



XUL
CSS
JavaScript
XPCOM



**Can all be
overwritten and
appended after they're
built and installed**





css



Cascading

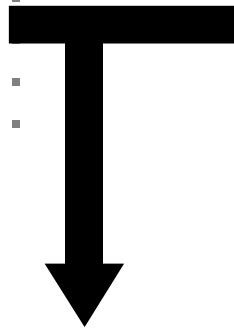


**When there are different rules
matching the same element, the
highest priority rule is assigned
(The rule is overwritten)**



```
box {  
  height: 100px;  
  border: 2px solid white;  
}
```

```
box#content {  
  height: 200px;  
  border-width: 3px;  
}
```



```
{  
  height: 200px;  
  border-style: solid;  
  border-color: white;  
  border-width: 3px;  
}
```



Rules are applied in the following order

Firefox Internal Stylesheet



Extension Stylesheet



User Stylesheet



**Extensions can
overwrite Firefox
stylesheets**



**Extension containing
only stylesheets**



Theme





XUL



A subset of XML



**Cascading feature
like in CSS**



Doesn't exist in XML



A Unique feature of XUL



Overlay



**This feature
merges multiple
XUL documents
into one**

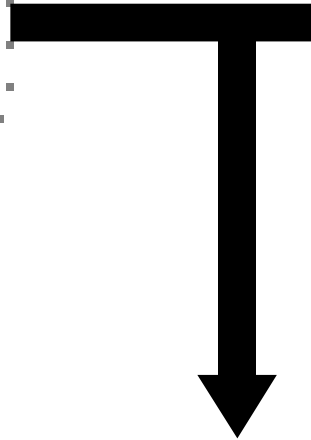


Original XUL document

```
<?xml version="1.0"?>
<window xmlns="...">
  <vbox id="containerBox"/>
</window>
```

Overlay XUL document

```
<?xml version="1.0"?>
<overlay xmlns="...">
  <vbox id="containerBox">
    <checkbox label="Imported Checkbox1"/>
    <checkbox label="Imported Checkbox2"/>
  </vbox>
</overlay>
```



New XUL document after merge

```
<?xml version="1.0"?>
<window xmlns="...">
  <vbox id="containerBox">
    <checkbox label="Imported Checkbox1"/>
    <checkbox label="Imported Checkbox2"/>
  </vbox>
</window>
```



Methods other than overlay



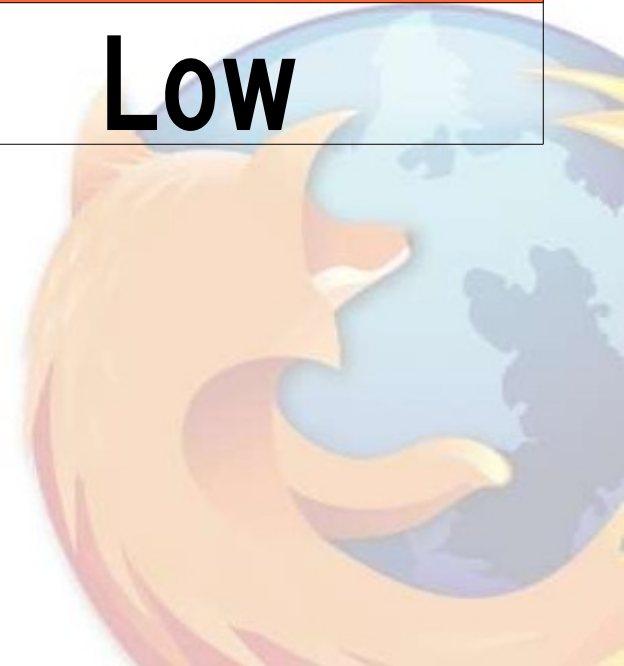
**Control XUL
using**

JavaScript and DOM



Comparing Overlay and JavaScript+DOM

| | Flexibility | Maintainability |
|----------------|-------------|-----------------|
| Overlay | Low | High |
| JavaScript+DOM | High | Low |



**Use each one
when appropriate**





JavaScript



2 important traits



1

**Strings, numbers, and
functions are all treated
as objects**



2

**No need for class definitions like in Java
(Can add, remove, and overwrite properties dynamically)**



Example

```
: var obj = new Object();  
: obj.objectProperty = new Object();  
: obj.objectProperty.newProperty = 'string'  
: obj.objectProperty.parent      = obj;  
: obj.myself = obj;
```



**Turn a function object
into a property**



Becomes a method



Example

```
var obj = new Object();  
obj.name = 'Firefox';  
  
function sayName() {  
    alert(this.name);  
}  
  
obj.sayMyName = sayName;  
obj.sayMyName(); // sais "Firefox"
```



Sample



Modification

Override `window.open()` so that
any attempt to open a Google
gets blocked
(*No ulterior motives)



```
· window.__original__open = window.open;
· window.open = function(aURI, aName, aFlag) {
·   if (aURI.indexOf('.google.') > -1)
·     return null;
·   return this.__original__open(aURI, aName, aFlag);
· };
· window.open('http://www.google.co.jp/'); // will be blocked
```



**Define a new method in
the JavaScript extension**



Feature Appended



**Replace existing method
with JavaScript extension**



Functionality Change





XPCOM



**Existing XPCOM
component features**



**Basically cannot
be configured**



Reason 1

Components built using
C/C++ is in binary,
hence they cannot be
configured



Reason 2

Components built using JavaScript need to be designated as configurable, otherwise they cannot be configured



But



**Existing XPCOM
components can be
replaced**



Keywords for XPCCOM



- **Component**
- **Interface**
- **Contract ID**



Component



**Small parts of
applications
(Libraries)**



Interface



**Input/Output standard
for components**



Contract ID



Name to identify components



**Create a component
with**

Same Interface

Same Name

**as an existing
component**



Then



**You can
replace
the existing XPCOM
component with the
user created one**





Summary

| | |
|-------------------|--------------------------|
| XUL | Overlay+DOM |
| CSS | Cascading |
| JavaScript | Method overload |
| XPCOM | Replace component |



**Different concept
from plugins**



| | Effect | Format |
|------------------|--|-----------------------------|
| Plugin | Performs tasks requested by Firefox | External application |
| Extension | Modifies Firefox source code | Patch |



**But it is
no ordinary
software patch**



**Because the
patch application
is dynamic**



XUL
CSS
JavaScript

**Are all interpretive, so they
don't require compilers**



**Firefox can be
configured and
modified in every way
after it is
built and installed**

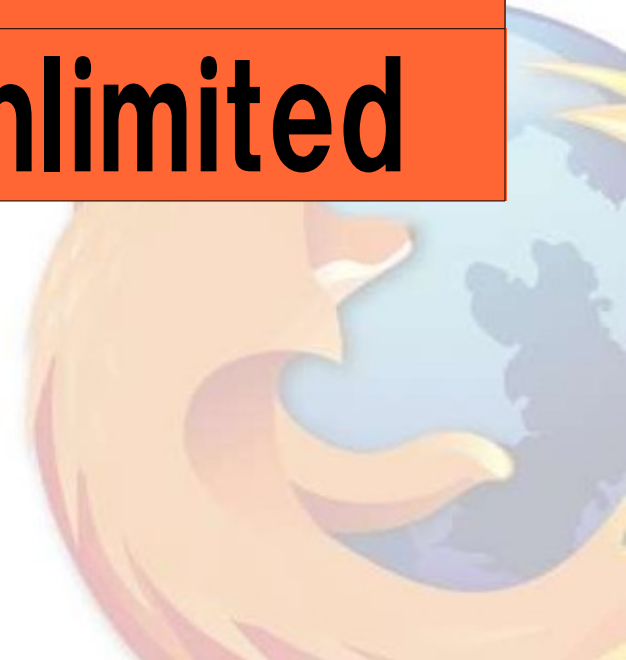




**Extension
traits**



| | Add/Remove | Possibilities |
|---------------------------|-------------------|----------------------|
| Plugin | Easy | Limited |
| Conventional patch | Difficult | Unlimited |
| Extension | Easy | Unlimited |



Conclusion



Extensions are very powerful customization tools when compared to plugins and conventional patches

