

DownloadFactory.java - ExternalFactory.java

```

1/*****
2 * Copyright (c) 2003, 2007 IBM Corporation and others.
3 * All rights reserved. This program and the accompanying materials
4 * are made available under the terms of the Eclipse Public License v1.0
5 * which accompanies this distribution, and is available at
6 * http://www.eclipse.org/legal/epl-v10.html
7 *
8 * Contributors:
9 *   IBM Corporation - initial API and implementation
10 *****/
11package org.eclipse.swt.browser;
12
13import org.eclipse.swt.internal.C;
14import org.eclipse.swt.internal.mozilla.*;
15
16class DownloadFactory {
17    XPCOMObject supports;
18    XPCOMObject factory;
19    int refCount = 0;
20
21DownloadFactory () {
22    createCOMInterfaces ();
23}
24
25int AddRef () {
26    refCount++;
27    return refCount;
28}
29
30void createCOMInterfaces () {
31    /* Create each of the interfaces that this object implements */
32    supports = new XPCOMObject (new int[] {2, 0, 0}) {
33        public int /*long*/ method0 (int /*long*/[] args) {return QueryInterface
34            (args[0], args[1]);}
35        public int /*long*/ method1 (int /*long*/[] args) {return AddRef ();}
36        public int /*long*/ method2 (int /*long*/[] args) {return Release ();}
37    };
38    factory = new XPCOMObject (new int[] {2, 0, 0, 3, 1}) {
39        public int /*long*/ method0 (int /*long*/[] args) {return QueryInterface
40            (args[0], args[1]);}
41        public int /*long*/ method1 (int /*long*/[] args) {return AddRef ();}
42        public int /*long*/ method2 (int /*long*/[] args) {return Release ();}
43        public int /*long*/ method3 (int /*long*/[] args) {return CreateInstance
44            (args[0], args[1], args[2]);}
45        public int /*long*/ method4 (int /*long*/[] args) {return LockFactory
46            ((int)/*64*/args[0]);}
47    };
48}
49
50void disposeCOMInterfaces () {
51    if (supports != null) {
52        supports.dispose ();
53        supports = null;
54    }
55    if (factory != null) {

```

```

1/*****
2 * Copyright (c) 2008 IBM Corporation and others.
3 * All rights reserved. This program and the accompanying materials
4 * are made available under the terms of the Eclipse Public License v1.0
5 * which accompanies this distribution, and is available at
6 * http://www.eclipse.org/legal/epl-v10.html
7 *
8 * Contributors:
9 *   IBM Corporation - initial API and implementation
10 *****/
11package org.eclipse.swt.browser;
12
13import org.eclipse.swt.internal.*;
14import org.eclipse.swt.internal.mozilla.*;
15
16class ExternalFactory {
17    XPCOMObject supports;
18    XPCOMObject factory;
19    int refCount = 0;
20
21ExternalFactory () {
22    createCOMInterfaces ();
23}
24
25int AddRef () {
26    refCount++;
27    return refCount;
28}
29
30void createCOMInterfaces () {
31    /* Create each of the interfaces that this object implements */
32    supports = new XPCOMObject (new int[] {2, 0, 0}) {
33        public int /*long*/ method0 (int /*long*/[] args) {return QueryInterface
34            (args[0], args[1]);}
35        public int /*long*/ method1 (int /*long*/[] args) {return AddRef ();}
36        public int /*long*/ method2 (int /*long*/[] args) {return Release ();}
37    };
38    factory = new XPCOMObject (new int[] {2, 0, 0, 3, 1}) {
39        public int /*long*/ method0 (int /*long*/[] args) {return QueryInterface
40            (args[0], args[1]);}
41        public int /*long*/ method1 (int /*long*/[] args) {return AddRef ();}
42        public int /*long*/ method2 (int /*long*/[] args) {return Release ();}
43        public int /*long*/ method3 (int /*long*/[] args) {return CreateInstance
44            (args[0], args[1], args[2]);}
45        public int /*long*/ method4 (int /*long*/[] args) {return LockFactory
46            ((int)/*64*/args[0]);}
47    };
48}
49
50void disposeCOMInterfaces () {
51    if (supports != null) {
52        supports.dispose ();
53        supports = null;
54    }
55    if (factory != null) {

```

DownloadFactory.java - ExternalFactory.java

```

53     factory.dispose ();
54     factory = null;
55 }
56
57
58 int /*long*/ getAddress () {
59     return factory.getAddress ();
60 }
61
62 int QueryInterface (int /*long*/ riid, int /*long*/ ppvObject) {
63     if (riid == 0 || ppvObject == 0) return XPCOM.NS_ERROR_NO_INTERFACE;
64     nsID guid = new nsID ();
65     XPCOM.memmove (guid, riid, nsID.sizeof);
66
67     if (guid.Equals (nsISupports.NS_ISUPPORTS_IID)) {
68         XPCOM.memmove (ppvObject, new int /*long*/[] {supports.getAddress ()},
69 C.PTR_SIZEOF);
69         AddRef ();
70         return XPCOM.NS_OK;
71     }
72     if (guid.Equals (nsIFactory.NS_IFACTORY_IID)) {
73         XPCOM.memmove (ppvObject, new int /*long*/[] {factory.getAddress ()},
74 C.PTR_SIZEOF);
74         AddRef ();
75         return XPCOM.NS_OK;
76     }
77
78     XPCOM.memmove (ppvObject, new int /*long*/[] {0}, C.PTR_SIZEOF);
79     return XPCOM.NS_ERROR_NO_INTERFACE;
80 }
81
82 int Release () {
83     refCount--;
84     if (refCount == 0) disposeCOMInterfaces ();
85     return refCount;
86 }
87
88 /* nsIFactory */
89
90 int CreateInstance (int /*long*/ aOuter, int /*long*/ iid, int /*long*/ result) {
91     Download download = new Download ();
92     download.AddRef ();
93     XPCOM.memmove (result, new int /*long*/[] {download.getAddress ()},
94 C.PTR_SIZEOF);
94     return XPCOM.NS_OK;
95 }
96
97 int LockFactory (int lock) {
98     return XPCOM.NS_OK;
99 }
100
101

```

```

53     factory.dispose ();
54     factory = null;
55 }
56
57
58 int /*long*/ getAddress () {
59     return factory.getAddress ();
60 }
61
62 int QueryInterface (int /*long*/ riid, int /*long*/ ppvObject) {
63     if (riid == 0 || ppvObject == 0) return XPCOM.NS_ERROR_NO_INTERFACE;
64     nsID guid = new nsID ();
65     XPCOM.memmove (guid, riid, nsID.sizeof);
66
67     if (guid.Equals (nsISupports.NS_ISUPPORTS_IID)) {
68         XPCOM.memmove (ppvObject, new int /*long*/[] {supports.getAddress ()},
69 C.PTR_SIZEOF);
69         AddRef ();
70         return XPCOM.NS_OK;
71     }
72     if (guid.Equals (nsIFactory.NS_IFACTORY_IID)) {
73         XPCOM.memmove (ppvObject, new int /*long*/[] {factory.getAddress ()},
74 C.PTR_SIZEOF);
74         AddRef ();
75         return XPCOM.NS_OK;
76     }
77
78     XPCOM.memmove (ppvObject, new int /*long*/[] {0}, C.PTR_SIZEOF);
79     return XPCOM.NS_ERROR_NO_INTERFACE;
80 }
81
82 int Release () {
83     refCount--;
84     if (refCount == 0) disposeCOMInterfaces ();
85     return refCount;
86 }
87
88 /* nsIFactory */
89
90 int CreateInstance (int /*long*/ aOuter, int /*long*/ iid, int /*long*/ result) {
91     External external = new External ();
92     external.AddRef ();
93     XPCOM.memmove (result, new int /*long*/[] {external.getAddress ()},
94 C.PTR_SIZEOF);
94     return XPCOM.NS_OK;
95 }
96
97 int LockFactory (int lock) {
98     return XPCOM.NS_OK;
99 }
100
101

```