```
1    /* -*- Mode: C++; tab-width: 2; indent-tabs-mode: nil; c-basic-offset: 2 -*-
2     *
3     * The contents of this file are subject to the Netscape Public License
4     * Version 1.0 (the "NPL"); you may not use this file except in
5     * compliance with the NPL.  You may obtain a copy of the NPL at
6     * http://www.mozilla.org/NPL/
7     *
8     * Software distributed under the NPL is distributed on an "AS IS" basis,
9     * WITHOUT WARRANTY OF ANY KIND, either express or implied. See the NPL
10    * for the specific language governing rights and limitations under the
11    * NPL.
12    *
13    * The Initial Developer of this code under the NPL is Netscape
14    * Communications Corporation.  Portions created by Netscape are
15    * Copyright (C) 1998 Netscape Communications Corporation.  All Rights
16    * Reserved.
17    */
18
19   /*
20    * nsMemModule. The memory based cache module.
21    *
22    * Gagan Saksena
23    * 02/03/98
24    *
25    */
26
27   #ifndef _nsMemModule_H_
28   #define _nsMemModule_H_
29
30   #include "nsCacheModule.h"
31   #include "nsMemCacheObject.h"
32   #include "nsCachePref.h"
33
34   //#include "nsHash.h" // TODO - replace with nsHashtable when the XPCOM_BRANCH merges
35
36   //#include "nsHashtable.h"
37   /*
38   // {5D51B250-E6C2-11d1-AFE5-006097BFC036}
39   static const NS_MEMMODULE_IID =
40   { 0x5d51b250, 0xe6c2, 0x11d1, { 0xaf, 0xe5, 0x0, 0x60, 0x97, 0xbf, 0xc0, 0x36 } };
41   */
42
43   class nsMemModule : public nsCacheModule
44   {
45
46   public:
47       nsMemModule(const PRUint32 size=nsCachePref::GetInstance()->MemCacheSize());
48       ~nsMemModule();
49
50   /*
51       NS_IMETHOD                QueryInterface(const nsIID& aIID,
52                                                void** aInstancePtr);
```

```
55
56    */
57        PRBool          AddObject(nsCacheObject* io_pObject);
58
59        PRBool          Contains(nsCacheObject* io_pObject) const;
60        PRBool          Contains(const char* i_url) const;
61
62        void            GarbageCollect(void);
63
64        nsCacheObject*  GetObject(const PRUint32 i_index) const;
65        nsCacheObject*  GetObject(const char* i_url) const;
66
67        nsStream*       GetStreamFor(const nsCacheObject* i_pObject);
68
69        PRBool          ReduceSizeTo(const PRUint32 i_NewSize);
70
71        PRBool          Remove(const char* i_url);
72        PRBool          Remove(const PRUint32 i_index);
73
74    //     PRBool          RemoveAll(void);
75
76        PRBool          Revalidate(void);
77
78
79        // Start of nsMemModule specific stuff...
80        // Here is a sample implementation using linked list
81    protected:
82        nsMemCacheObject* LastObject(void) const;
83
84    private:
85        nsMemCacheObject* m_pFirstObject;
86
87        //nsHash m_ht; //TODO replace with nsHashtable
88        //Optimization
89        nsMemCacheObject* m_pLastObject;
90
91        nsMemModule(const nsMemModule& mm);
92        nsMemModule& operator=(const nsMemModule& mm);
93
94    /*
95        class nsMemKey : public nsHashKey
96        {
97        public:
98                      nsMemKey();
99                      ~nsMemKey();
100       PRUint32      HashValue();
101       PRBool        Equals(nsHashKey *aKey);
102       nsHashKey*    Clone();
103       };
104    */
105    };
106
```

```
109    {
110          /* Mem module elements are never revalidated */
111          return PR_FALSE;
112    }
113
114    #endif
115
```