

Results Cache Manager

Scott Auge
sauge@amduus.com
<http://www.amduus.com>

ABSTRACT

Often in web applications we will need to draw up a large set of result data stored in a table. This often takes a lot of time to pull out of the main tables, collate, and join together fields. This can become expensive on pages only meant to display a portion of the data at a time (the concept of “paging” the data so the web page is manageable to read.) By using a cache manager, it will be easier to store the result set to disk and use some procedures to pull back “pages” of data that we desire. This manager can be used in Web 1.0 as well as in AJAX type web applications.

DISCUSSION

The cache manager is a persistent procedure. One can have many caches in a given program. This is achieved by merely creating new persistent procedures to store different result sets to be cached and manipulated.

We also need the cache manager to be flexible enough to handle multiple types of tables or temp-tables. This code can handle both types though temp-tables will likely be most useful.

The cache manager will be able to store multiple types of fields and different collections of fields.

Creating A Cache Manager

The steps to creating a cache manager are as follows:

1. Create a temp table to store your data in.
2. Populate your temp-table with data you wish to cache.
3. Create an instance of the cachemgr.p program and remember it's handle.
4. Delete the instance of the persistent procedure before finishing your web hit.

Here is some example code:

```
DEFINE VARIABLE hCache AS HANDLE NO-UNDO.  
DEFINE VARIABLE cCacheName AS CHARACTER NO-UNDO.  
  
RUN cachemgr.p PERSISTENT SET hCache.  
  
DELETE OBJECT hCache.
```

Saving A Result Set With The Cache Manager

This is done when you first calculate the result set. One doesn't want to have to repeatedly do this so on the first hit populate your temp-table and then call WriteDataToDisk. WriteDataToDisk will return a name for the cache. This name should be remembered for subsequent hits by either the WebSession table or by a NVP traveling in the web pages between hits.

Here is some example code:

```
DEFINE VARIABLE hCache AS HANDLE NO-UNDO.
DEFINE VARIABLE cCacheName AS CHARACTER NO-UNDO.

DEFINE TEMP-TABLE Results ...

... Populate your table ...

RUN cachemgr.p PERSISTENT SET hCache.

RUN WriteDataToDisk IN hCache
(INPUT TEMP-TABLE Results:HANDLE,
 OUTPUT cFileName).

... Store cCacheName someplace for the session ...

DELETE OBJECT hCache.
```

Loading A Result Set With The Cache Manager

On second and $n + 1$ hits that is paging information from the cache, one will need to load the cache from disk into the instance of the cachemgr.p procedure on that hit.

To do so, all you need to do is pull back the name of the cache from the NVP or WebSession record and call:

```
RUN ReadDataFromDisk IN hCache
(INPUT cFileName,
 INPUT TEMP-TABLE Results:HANDLE).
```

Determining The Number Of Pages With The Cache Manager

Once you have loaded a result set from the cache, you can determine the number of pages contained therein. This can help you determine if you need a “next” link or not as well a list of page numbers that can be clicked on to load that specific page.

Here is some sample code:

```
RUN ReadDataFromDisk IN hCache
(INPUT cFileName,
 INPUT TEMP-TABLE Results:HANDLE).
```

```
RUN DetermineNumberOfPages IN hCache
(INPUT 10, /* Rows Per Page */
OUTPUT PageCount).
```

Retrieving A Specific Page Of Rows From The Cache Manager

Once one has read in a cache from disk, one can easily retrieve into a temp-table of the same type put into the cache the select page of data. Then one can simply do a FOR EACH of the data in the temp-table.

```
DEFINE TEMP-TABLE Results
  FIELD FirstName AS CHARACTER
  FIELD LastName AS CHARACTER
  FIELD Order AS INTEGER.

DEFINE TEMP-TABLE PagedResults LIKE Results.

...

RUN ReadDataFromDisk IN hCache
(INPUT cFileName,
INPUT TEMP-TABLE Results:HANDLE).

...

RUN DetermineNumberPages IN hCache
(INPUT 4,
OUTPUT iNumberOfPages).

...

EMPTY TEMP-TABLE PagedResults.

RUN ReturnPageResults IN hCache
(INPUT 1, /* Page number */
INPUT 4, /* Rows per page */
TEMP-TABLE PagedResults:HANDLE).

FOR EACH PagedResults NO-LOCK:
  {&DISPLAY} PagedResults.
END.
```

Clearing The Cache

If you have the opportunity to, one should clear the cache when done. This can be done by:

```
RUN ClearData IN hCache
(INPUT cCacheName).
```

Often a user will likely leave without clearing any caches. We should simply set up a cron job to clear temp cache files one or two days old.

Caveats To Remember

All temp-tables sent into the cachemgr.p must be the same temp-table definition as was used to create the cache.

Sample Test Program

This is a test program to check if the routine will work:

```
DEFINE VARIABLE hCache AS HANDLE NO-UNDO.
DEFINE VARIABLE hTbl AS HANDLE NO-UNDO.
DEFINE VARIABLE cFileName AS CHARACTER NO-UNDO.
DEFINE VARIABLE cError AS CHARACTER NO-UNDO.
DEFINE VARIABLE iNumberOfPages AS INTEGER NO-UNDO.
```

```
DEFINE TEMP-TABLE Results
  FIELD FirstName AS CHARACTER
  FIELD LastName AS CHARACTER
  FIELD Order AS INTEGER.
```

```
DEFINE TEMP-TABLE PagedResults LIKE Results.
```

```
CREATE Results.
```

```
ASSIGN
Results.FirstName = "1f"
Results.LastName = "1l"
Results.Order = 1.
```

```
CREATE Results.
```

```
ASSIGN
Results.FirstName = "2f"
Results.LastName = "2l"
Results.Order = 2.
```

```
CREATE Results.
```

```
ASSIGN
Results.FirstName = "3f"
Results.LastName = "3l"
```

```
Results.Order = 3.

CREATE Results.

ASSIGN
Results.FirstName = "4f"
Results.LastName = "4l"
Results.Order = 4.

CREATE Results.

ASSIGN
Results.FirstName = "5f"
Results.LastName = "5l"
Results.Order = 5.

CREATE Results.

ASSIGN
Results.FirstName = "6f"
Results.LastName = "6l"
Results.Order = 6.

CREATE Results.

ASSIGN
Results.FirstName = "7f"
Results.LastName = "7l"
Results.Order = 7.

CREATE Results.

ASSIGN
Results.FirstName = "8f"
Results.LastName = "8l"
Results.Order = 8.

RUN cachemgr.p PERSISTENT SET hCache.

RUN WriteDataToDisk IN hCache
(INPUT TEMP-TABLE Results:HANDLE,
OUTPUT cFileName).

RUN GetError IN hCache (OUTPUT cError).

{%OUT} "WriteDataToDisk<BR>".
{%OUT} "Error " cError "<BR>".
```

```

{%OUT} "File Name " cFileName "<BR>".

EMPTY TEMP-TABLE Results.

RUN ReadDataFromDisk IN hCache
(INPUT cFileName,
 INPUT TEMP-TABLE Results:HANDLE).

RUN GetError IN hCache (OUTPUT cError).

{%OUT} "ReadDataFromDisk<BR>".
{%OUT} "Error " cError "<BR>".
{%OUT} "File Name " cFileName "<BR>".

RUN DetermineNumberPages IN hCache
(INPUT 4,
 OUTPUT iNumberOfPages).

RUN GetError IN hCache (OUTPUT cError).

{%OUT} "ReadDataFromDisk<BR>".
{%OUT} "Error " cError "<BR>".
{%OUT} "iNumberOfPages " iNumberOfPages "<BR>".

/* Test nice mod of entries per page */

/* Page 1 */

EMPTY TEMP-TABLE PagedResults.

RUN ReturnPageResults IN hCache
(INPUT 1,
 INPUT 4,
 TEMP-TABLE PagedResults:HANDLE).

RUN GetError IN hCache (OUTPUT cError).

{%OUT} "ReturnPageResults<BR>".
{%OUT} "Error " cError "<BR>".

FOR EACH PagedResults NO-LOCK:
  {%DISPLAY} PagedResults.
END.

```

```
/* Page 2 */

EMPTY TEMP-TABLE PagedResults.

RUN ReturnPageResults IN hCache
(INPUT 2,
 INPUT 4,
 TEMP-TABLE PagedResults:HANDLE).

RUN GetError IN hCache (OUTPUT cError).

{%OUT} "ReturnPageResults<BR>".
{%OUT} "Error " cError "<BR>".

FOR EACH PagedResults NO-LOCK:
  {%DISPLAY} PagedResults.
END.
```

```
/* Test nasty mod of entries per page */
```

```
/* Page 1 */

EMPTY TEMP-TABLE PagedResults.

RUN ReturnPageResults IN hCache
(INPUT 1,
 INPUT 3,
 TEMP-TABLE PagedResults:HANDLE).

RUN GetError IN hCache (OUTPUT cError).

{%OUT} "ReturnPageResults<BR>".
{%OUT} "Error " cError "<BR>".

FOR EACH PagedResults NO-LOCK:
  {%DISPLAY} PagedResults.
END.
```

```
/* Page 2 */
```

```
EMPTY TEMP-TABLE PagedResults.
```

```
RUN ReturnPageResults IN hCache  
(INPUT 2,  
  INPUT 3,  
  TEMP-TABLE PagedResults:HANDLE).
```

```
RUN GetError IN hCache (OUTPUT cError).
```

```
{&OUT} "ReturnPageResults<BR>".  
{&OUT} "Error " cError "<BR>".
```

```
FOR EACH PagedResults NO-LOCK:  
  {&DISPLAY} PagedResults.  
END.
```

```
/* Page 3 */
```

```
EMPTY TEMP-TABLE PagedResults.
```

```
RUN ReturnPageResults IN hCache  
(INPUT 3,  
  INPUT 3,  
  TEMP-TABLE PagedResults:HANDLE).
```

```
RUN GetError IN hCache (OUTPUT cError).
```

```
{&OUT} "ReturnPageResults<BR>".  
{&OUT} "Error " cError "<BR>".
```

```
FOR EACH PagedResults NO-LOCK:  
  {&DISPLAY} PagedResults.  
END.
```

```
DELETE OBJECT hCache.
```

The results of this are:

WriteDataToDisk
Error NONE
File Name /tmp/cachemgr_p_BrhpRvZIic4892174
ReadDataFromDisk
Error NONE
File Name /tmp/cachemgr_p_BrhpRvZIic4892174
ReadDataFromDisk
Error NONE
iNumberOfPages 2
ReturnPageResults
Error NONE

FirstName	LastName	Order
-----	-----	-----
1f	11	1
2f	21	2
3f	31	3
4f	41	4

ReturnPageResults
Error NONE

FirstName	LastName	Order
-----	-----	-----
5f	51	5
6f	61	6
7f	71	7
8f	81	8

ReturnPageResults
Error NONE

FirstName	LastName	Order
-----	-----	-----
1f	11	1
2f	21	2

3f 3l 3

ReturnPageResults

Error NONE

FirstName	LastName	Order
-----	-----	-----

4f	4l	4
----	----	---

5f	5l	5
----	----	---

6f	6l	6
----	----	---

ReturnPageResults

Error NONE

FirstName	LastName	Order
-----	-----	-----

7f	7l	7
----	----	---

8f	8l	8
----	----	---