Report Status

This section covers the following topics:

- Report States
- Report Definition
- Container Files and Database Files

Report States

Entire Output Management creates active reports according to master report definitions. An active report may be in one of four states:

Status	Explanation
Current	This means that the active report has not yet reached its expiry date. It can be viewed online, but is not yet archived.
Current/Archived	This means that the report has not yet reached its expiry date, can be viewed online and has already been copied to an archive (so if the spool queue gets deleted, the report is not lost).
Archived	An active report exists but has passed its expiry date. The only copy of it is in an archive. The report cannot be viewed online.
Revived	An active report exists, has passed its expiry date and been archived and has subsequently been revived from the archive so that it is again available for online viewing until its revive expiry date passes.

Additionally, an active report may have an interim state. For example, "to be archived". An active report would be "to be archived" between its expiry and next run of the archive job. Such a report can still be viewed online until the archive job has processed it. (Bear in mind that reports which are on the printout queue waiting to be printed or in a still open active bundle will not be archived, even if they have expired).

Report Definition

Active report handling is governed by various settings in the report definition (or in System Defaults, if the report definition does not specify them). These settings are explained below.

1. Archive directly - Whether or not an archive copy is to be taken immediately. May be set to:

Setting	Explanation
Y	As soon as the active report is created it is marked as "to be archived". The next time the archive job runs, the active report will be archived and will no longer be viewable online. If Y is specified, the report retention information is ignored.
I	As soon as the active report is created it is marked as "to be archived and retained online". The next time the archive job runs, the active report is archived, but it is also retained for online viewing until the expiry of the report retention period.
N	The active report is created and is viewable online until the expiry of its report retention.

2. Retention (Report) These fields specify how long a report is retained for online viewing. You specify a number, for example 5, a unit (A - absolute days, W - working days (can specify a calendar which defines which days are working and which not), V - weeks, M - months, G - generations, that is instances of the same active report), a calendar and an action. The action is ignored if Archive directly is Y or I. Otherwise, action may be A to archive the active report or P to delete it.

3. Retention (Archive) Specifies how long the active report is to be retained on archive. Again you provide a number and a unit. The unit can be D(ays), W(eeks), M(onths) or Y(ears). As working days is not an option, there is no calendar for archive retention. When this retention period expires, the active report is completely deleted and cannot be recovered.

4. Retention (Revive) How long a revived active report is to be retained for viewing. Specify number, unit and calendar, as for report retention. When this period expires, the active report reverts to an archived state and can no longer be viewed online.

Report Retention Example

Say a report is defined with a report retention period of 5 A and action A.

An active report is created on March 1st, 2008. Its expiry date will be on March 6th, 2008. So the monitor will mark the report as "to be archived" on March 7th, 2008. The next time the archive job is run, the active report contents will be copied to an archive dataset, and if anyone needs to view the report after that, it has to be revived.

Container Files and Database Files

Entire Output Management can copy report contents from their original location (for example, JES spool) into either a container file or the Entire Output Management system file (or both).

Copying into the system file is independent of any container file usage and will only be done for reports that are defined with Store in NOM DB set to Y.

On z/OS systems, reports should only be copied into the database, if absolutely necessary (for example, to avoid accidental loss through spool deletion) because it is a big overhead to store large reports in the database.

Usage of Container Files

- Defining Container Files
- Compression and Blocking
- Direct Access

Entire Output Management will copy report sources into a container file under the following circumstances:

1. Report is from CMA-SPOOL, Natural Advanced Facilities, SAP or the 3GL interface.
2. In BS2000/OSD, copy files is set in monitor defaults.
3. In JES2/JES3/POWER, if a spool file is processed with a DEST that matches one of the destinations defined in monitor defaults/copy files.

Defining Container Files

On the Monitor Defaults screen, press PF7 and the Copy to DB Files window opens, here you define your container files, together with the spool destination that will be copied into the associated container file.

CMA-SPOOL, NAF, SAP and 3GL Interface all have their own sections in monitor defaults where you can define container files.

Compression and Blocking

Entire Output Management stores output in a multiple-value field of 11 * 250 = 2761 bytes, which when this feature was implemented, was at that time the greatest common block size for disk storage devices supported by Adabas.

Giving this multiple-value field to Adabas as it is would mean, that every occurrence would normally become shorter by the Adabas compression. Meaning also, that each Adabas record is not filled to our maximum of 2761 bytes and thus resulting in more Adabas calls when storing and retrieving the data. This may have a high impact on performance when dealing with very large output.

For this reason Entire Output Management does the compression itself.

Direct Access

Separating, browsing and printing in parts usually results in the need for direct access to record ranges of the output. This, however, is not given in mainframe spooling systems and disk datasets with a variable record length. Access to the Entire Output Management container file is extremely fast.

Entire Output Management does this storage into the container file before processing any report definitions. The original output is copied as a whole, all following separation processing, browsing and printing in parts will be very fast.

Entire Output Management keeps the whole original output stored in the database as long as there is at least one active report with location S (S stands for *source*, not *spool*) pointing to it. This could mean that the container file is filled with very large output, even though only a fraction is actually needed.

It makes sense to store in a container file whenever intensive separation processing, browsing or printing in parts is necessary.

If you have intensive separation processing, but the resulting reports out of the whole original is only a fraction, set the option Store in NOM DB in the report definition(s). The resulting reports will be copied from the original output residing in the container file into the Entire Output Management system file, and the location of the report will become D. When the next cleanup is done and there are no reports with location S pointing to the original source, it will be deleted from the container file.

Be aware though, that for the lifetime of the original output in the continer file, reports created with Store in NOM DB in the report definition(s) mean that those parts of the output are indeed stored again in the database, whereas with location S Entire Output Management would keep only pointers to the container file.

In summary:

- Use container files for heavy processing.
- In addition, use Store in NOM DB when only small parts of the original are needed or when the resulting reports have very different expiration dates.
- Remember that the report with the highest expiration date will determine the lifetime of the whole original output in the container file.