

# Concepts and Facilities

This document describes what Entire Output Management can do and how it works. It covers the following topics:

- What Can Entire Output Management Do?
  - How Does Entire Output Management Work?
  - Overview of Entire Output Management's Functions
- 

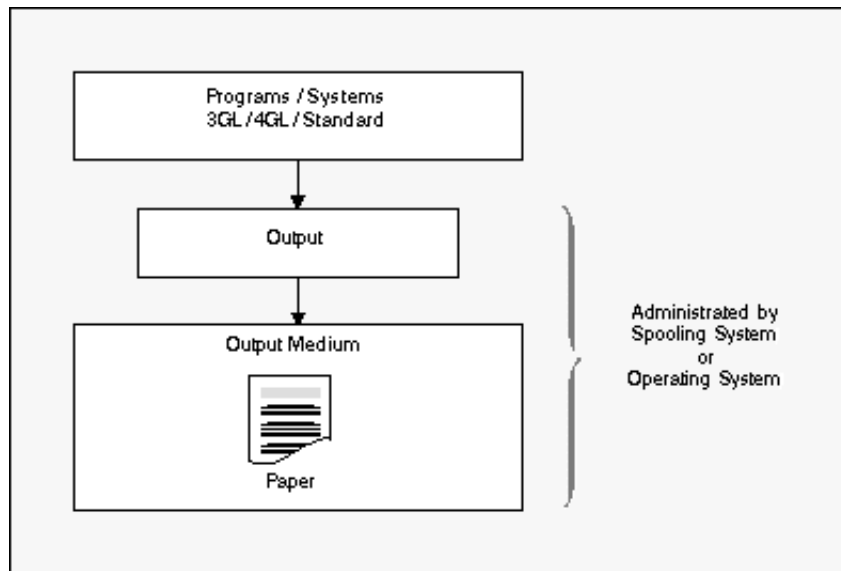
## What Can Entire Output Management Do?

Entire Output Management is a tool that can process every kind of print data in heterogeneous client/server environments, rule-based and automatically in a user-friendly format, without changing the applications or programs that created the data.

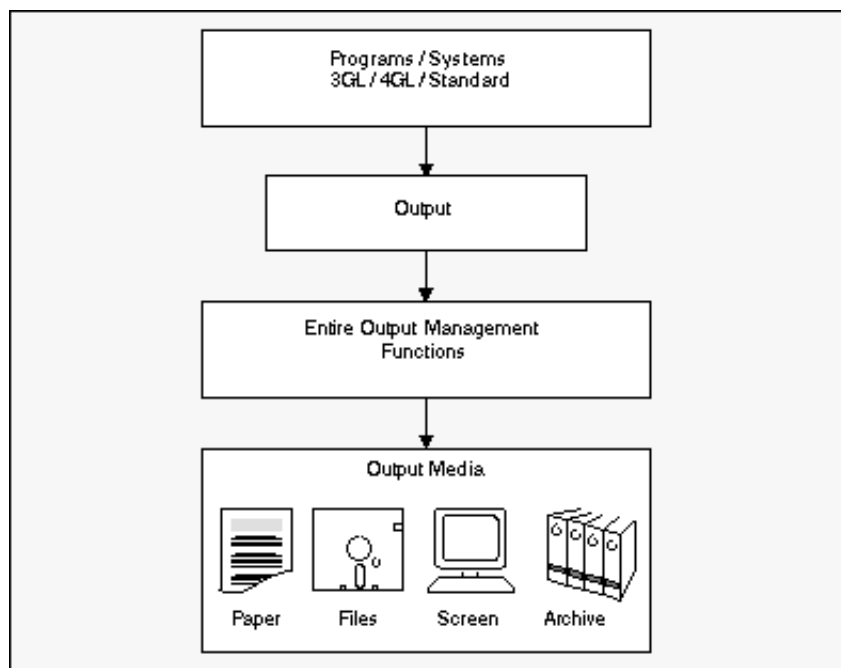
For this reason, a data center can use Entire Output Management to automate the distribution of print data. Individual departments can also use Entire Output Management for presenting their data on screen, so they can be used immediately for decision-making, without having to resort to hard copy. One of the main goals of Entire Output Management is to save on printing costs.

Entire Output Management functions like a logical administration system for data: it has clearly defined interfaces to existing spooling systems and the functionally equivalent parts of an operating system.

### Administration of Program Output without Entire Output Management



### Administration of Program Output with Entire Output Management



This means that the programs producing the data remain unchanged: output is produced in the usual way and is passed to the spooling system for administration. Instead of printing these data automatically, Entire Output Management reads them and uses predetermined rules on them.

Among its numerous processing options, Entire Output Management, then allows you to return all or part of the data to the spooling system, which in turn controls the physical printing process.

## How Does Entire Output Management Work?

- General
- Reports
- Bundles
- Distribution Lists
- Data Distribution
- Printing
- TCP/IP Direct Printing
- Processing of Binary Data
- Folders
- Archiving/Reviving
- Condensing (Compression)

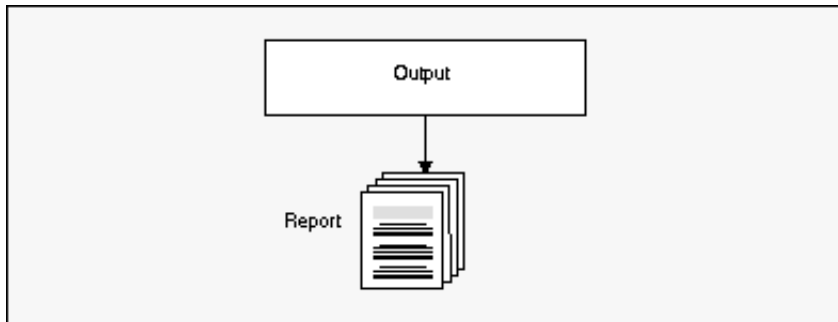
- Cleanup

## General

Entire Output Management allows you to adapt your present data distribution system step by step. You can also specify characteristics to identify the processes or programs whose output Entire Output Management will handle.

In addition, you can determine which parts of the output from the identified programs is to be further processed by Entire Output Management. This is called *separation* and the result is a *report*.

## Reports

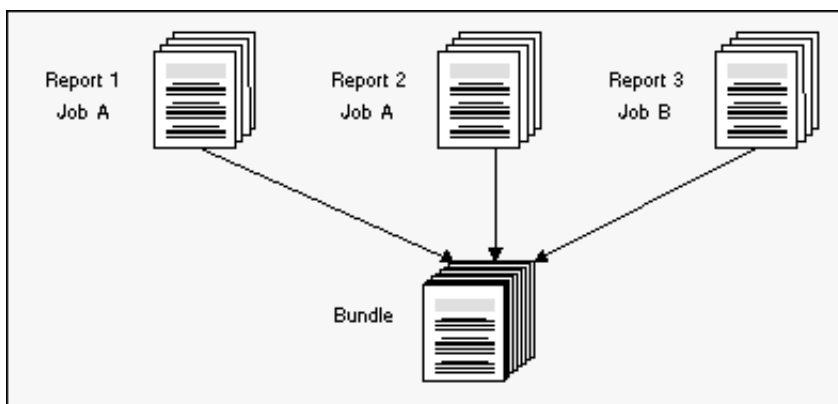


A report contains those parts of the output that are important for the recipient. Entire Output Management's rules make it easy for you to determine what is important.

Entire Output Management by itself allows you to create and process reports.

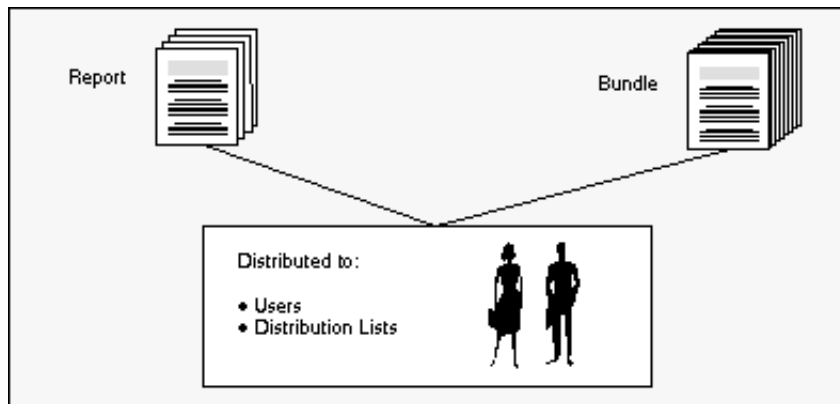
In addition, reports can be combined to form larger packets, even if the reports come from different data sources (e.g. jobs). This option is called bundling and the result is called a *bundle*.

## Bundles



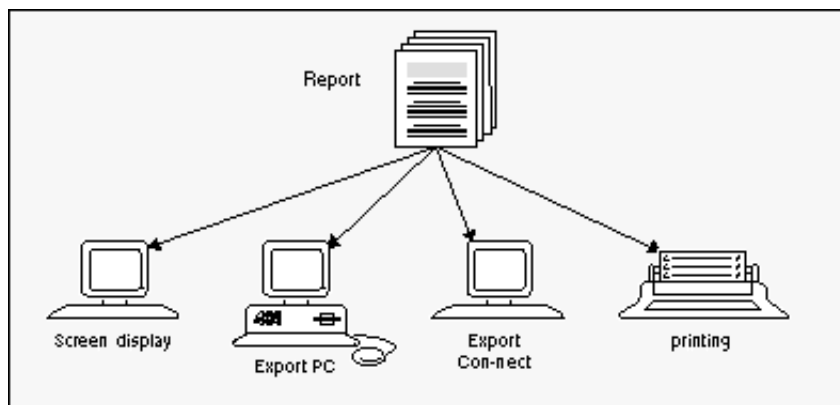
Distribution lists, which contain the names of individual users or the names of other distribution lists or both, allow you to distribute reports and bundles to selected end-users.

## Distribution Lists



Only the members of the distribution list of a report or bundle have access to the report or bundle contents (multi-client facility) and they can display these data on the screen.

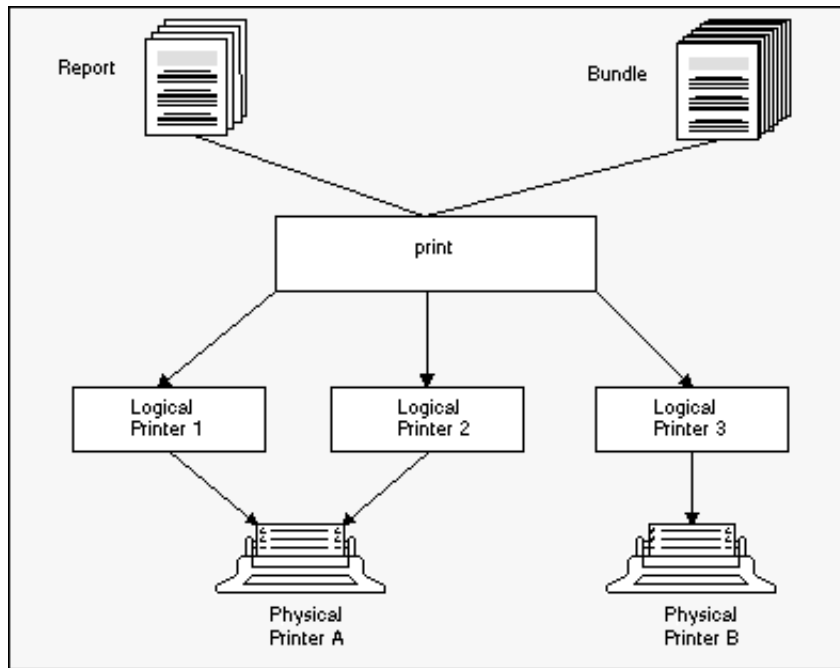
## Data Distribution



Up to now, the user had to view data on paper after an involved manual distribution process and in a form determined by the application that produced the data. But now Entire Output Management enables the user to display the important data directly at his or her terminal. And all or part of the data can be exported for example to Con-nect, Software AG's office communication system, or to a PC for further local processing.

You can, of course, also print reports and bundles. This can be done either automatically with a rule-based process or manually, when requested by a user viewing the data at his or her terminal. Printing in Entire Output Management means the selected data (bundles, reports, subsections of reports) can be formatted for a particular printer. They are then passed to a logical printer.

## Printing



A *logical printer* is a specific set of attributes linked to a *physical printer*, which determine the printing format for the physical printer.

Reports and bundles have an individual lifetime, during which they are available online. Each member of a distribution list can authorize other users to access those reports for which he or she is authorized. Such a report appears in the inbasket of the recipient, from which he or she can file it in individually defined folders.

## TCP/IP Direct Printing

Entire Output Management provides methods for accessing printers: It is able to directly print to TCP/IP printers. This print method is intended to access printers that have either got a dedicated IP address or can be reached using printer queue names of printer servers.

This print method is intended for direct printing of shorter documents directly from online applications. However, it is executed in a Natural subtask and no online space (such as CICS memory) is affected. No spooling system is required, and no extra address space will be used, not even an ESY server. This makes this print method independent from batch jobs, Broker, CA-SPOOL, ESY, JES, and Natural Advanced Facilities.

Performance enhancements are achieved, and 4 MB of space in an Entire Output Management printer subtask per 1000 pages of text output are needed as soon as the printer (server) acknowledges the reception of the printout.

TCP/IP direct printing relieves the former printer type UNIXLP which worked with servers across broker communication.

At print time, the printout will be initiated online or by the monitor and executed in one of the defined print tasks of Entire Output Management. At execution time, the printout will be kept in virtual memory and then sent to TCP/IP directly (via socket programming) using the LPR/LPD protocol (RFC1179).

## Processing of Binary Data

Entire Output Management also allows you to process binary data. This means that any kind of file or printout can be kept in Entire Output Management, archived, printed, distributed, and passed to a destination system.

Basically there are two ways to get binary data into Entire Output Management:

- You can use the direct-input interface together with the Open Print Option (OPO). A Windows client passes output of the Windows systems to Entire Output Management. This can be the output of a Windows printer driver or any program that can forward data to OPO using the pipe mechanism of Windows (see the section *Installing the Open Print Option* in the *Installation* documentation for details). The output is redirected to Entire Output Management using the Remote Procedure Call (RPC) facility of EntireX.
- You can define a report that receives data from a UNIX or Windows directory. A parameter "binary reading" can be filled with "B" to indicate that the file is to be processed in binary format.

The report is passed through the Entire Output Management trigger queue, similar to the NOMPUT interface. Therefore the trigger queue must be activated (NOM API and User-Exit Defaults). The opened active report receives the type "binary" (a special CC type).

It cannot be browsed with the mainframe character interface. However, the Entire Output Management GUI Client is able to receive the data, store it on the PC (please note that the whole file has to be transferred) and browse (view) it with the Windows default application of the specific file type.

Binary active reports can be printed with UNIXLP printers or on BS2000/OSD with a SYSPRBS2 printer type. Under UNIX, a printer type NATUNIX is available which can pass the data to the "lpr" utility (in which case the option "-l" has to be used to ensure that the data are forwarded in a transparent mode) or to a file in a UNIX directory.

### Note:

A binary report that is an output of a Windows printer driver will be bound to the hardware it was created for. The decision where to print the data Entire Output Management may have received a long time ago is made at creation time. Consider to use file formats like PDF if you want to keep binary data for a long time.

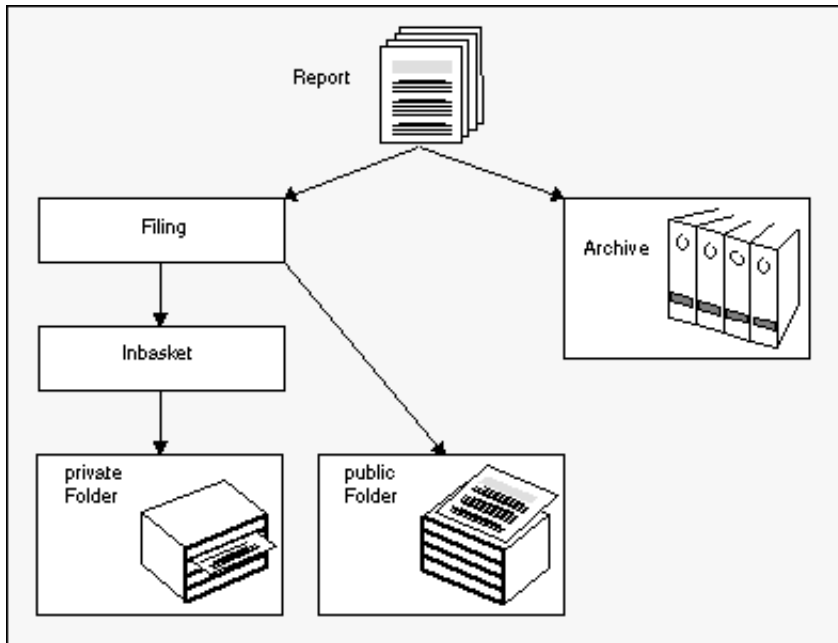
Entire Output Management converts binary data to the BASE64 format. This makes it possible to pass the data across platforms and code pages, because BASE64 consists of printable characters only. However, this also means that the data are larger than the original binary data stream. When active reports are output to destinations, the BASE64 data stream is decoded to binary format again.

The output of printer drivers often result in large data streams, because a printer driver creates print pages with all graphical statements of the appropriate printer language.

In theory, separation exits can be used. However, binary data cannot be separated with standard exits, as they are not readable.

Any report Entire Output Management receives via the OPO can be enriched with user-defined meta data. The meta data can be viewed via PF2 on the spooling attributes screen of an active report, or in any user exit that uses the field SPOOL-ATTRIBUTES-EXTENDED. Meta data are passed to Entire Output Management via an XML file that can be defined in an OPO printer definition. See the section *Installing the Open Print Option* in the *Installation* documentation for details.

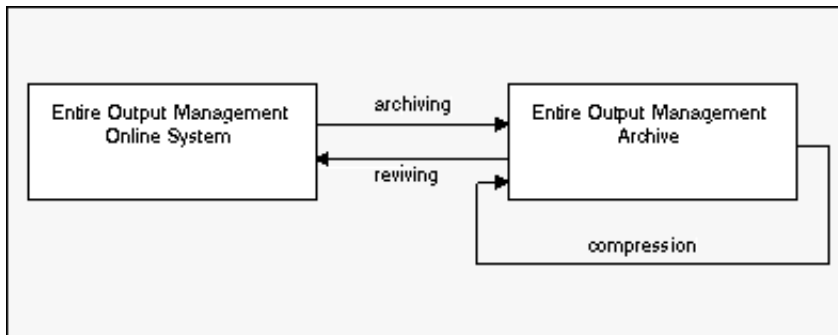
## Folders



There are also public folders, to which each Entire Output Management user has access. The procedure described above is, of course, not necessary for the public Folders.

When the lifetime of a report has expired, the report can be archived on storage media.

## Archiving/Reviving



Entire Output Management keeps an archive account and allows you to revive archived reports if you need them online again. Archive storage media which are no longer needed are freed during compression.

## Condensing (Compression)

"Condense" jobs read archives and delete all data whose retention period has been exceeded and which are no longer needed. This helps to reduce space for archive data (which are sequential files) to a minimum. With the function Automatic Archiving Defaults you can define a threshold for the number of reports in an archive dataset that will trigger its being marked for condensation.

## Cleanup

This section provides an overview of the various cleanup functions provided by Entire Output Management. It covers the following topics:

- Cleanup Types
- Daily Cleanup
- Source Cleanup
- Report Cleanup
- Daily Cleanup – Online or in Batch Mode?

### Cleanup Types

There are three types of cleanup:

- daily cleanup,
- source cleanup,
- report cleanup.

The primary cleanup type is daily cleanup. Entire Output Management performs its daily cleanup once and once only per day on its next activation after the time specified in the system defaults (option 8.1.1). If no time is specified there, the daily cleanup will be done on the first activation after midnight.

Source cleanup and report cleanup are optional and can be activated in option 8.1.7. Each can be activated individually and a schedule defined for it. They may be scheduled to run whenever you wish, using a calendar to specify on which days of the week or month they are to run, whether they should run before or after non-working days, the time window during which they are to run and how often. For example, you could define that source/report cleanup is to run every 2 hours between 08:00 and 20:00 on Mondays, Wednesdays and Fridays.

### Daily Cleanup

The daily cleanup processes all expired entities:

- Active reports that have expired are purged or marked for archiving, as appropriate.
- Active bundles, printouts and log records that have expired are purged.
- Revived reports are “unrevived”, that is, the contents are deleted if the report was revived to the database and the active report is reset to its archived state.
- Archived active reports are deleted.
- Active report sources (spool files or container file entries) are checked and, if no longer needed, deleted.



## Source Cleanup

Source cleanup checks Entire Output Management sources (for example, JES or Power spool files in Entire Output Management's temporary classes or entries in the container files) to see if they are still needed - and deletes them if they are not needed. They are no longer needed when there is no active report with location "S" referring to them. Container file entries which are no longer needed are deleted even if Spool Cleanup is set to "N".

## Report Cleanup

Report cleanup checks all active reports with location "S" to establish whether or not the source file is still available. If not, the active report is deleted.

## Daily Cleanup – Online or in Batch Mode?

The daily cleanup process may entail a lot of work (typically it can run between 30 and 60 minutes in a production environment), and during this time the monitor can do nothing else; this means that no reports, bundles or printouts can be processed until the daily cleanup has finished. This is because the monitor is a single task.

To avoid this problem, the daily cleanup can run in an asynchronous batch job. To do this, you execute the program NOMCLEAN in a standard batch Natural (with the necessary parameters set, such as LFILE 206). This batch job can then be activated by a job scheduler to run a couple of hours before the Entire Output Management monitor is scheduled to perform the daily cleanup. When the monitor is about to perform the daily cleanup, it can ascertain that NOMCLEAN has ran and so it will not have to do anything.

## Communication

NOMCLEAN and the monitor keep each other informed of the current status by using two control records in the Entire Output Management system file. These control records are accessed using the view NRM-ARCHIVE-TASK and the descriptor M-MONITOR-ID = 'CLEANUP' or M-MONITOR-ID = SYSNOM (actually the library Entire Output Management is running from).

The communication paths are as follows:

- NOMCLEAN at its start sets SYSNOM's CLEANUP-IDENTIFIER to 'CLEAN', CLEANUP-ACTIVE-NOW to TRUE and CLEANUP-RESCHEDULE to FALSE to show that it is active.
- RMONITOR (monitor main loop) only deletes no longer needed sources if SYSNOM's CLEANUP-IDENTIFIER is not CLEAN or CLEANUP-ACTIVE-NOW is not TRUE (that is, only if NOMCLEAN is not currently active). In addition, to avoid concurrency problems while creating active reports, RMONITOR sets CLEANUP's REPORT-PROCESSING-NOW to TRUE while creating active reports, and then to FALSE when it has finished. This is necessary to prevent NOMCLEAN from deleting reports or sources that are being worked on by the monitor.
- RMARCSCH (monitor scheduled functions) only performs daily cleanup if SYSNOM's CLEANUP-RESCHEDULE and CLEANUP-ACTIVE-NOW are both FALSE (that is, only if NOMCLEAN is not currently active and has not already run on that day). If CLEANUP-RESCHEDULE is TRUE, this means that NOMCLEAN has already run so RMARCSCH resets CLEANUP-RESCHEDULE to FALSE and updates the next scheduled daily cleanup to the same time on the next day. RMARCSCH only performs source/report cleanup if

SYSNOM's CLEANUP-ACTIVE-NOW is FALSE. This is to avoid collision with NOMCLEAN which may also be cleaning up sources.

- RMSRCK (source/report cleanup) may be invoked by the monitor or NOMCLEAN. When invoked by NOMCLEAN, it needs to avoid the concurrency problem described above. It does this by checking REPORT-PROCESSING-NOW and, if it is TRUE, uses the NPR EVENTING view to make itself dormant (repeatedly, for 5 seconds at a time) until REPORT-PROCESSING-NOW is FALSE, which is when RMSRCK resumes processing where it left off.

If multi-tasking is active (more than one monitor task is defined), the monitor cleanup functions will be performed by the main task, task 1. This will allow cleanup to be separated from other monitor functions like active report and printout creation.

## Overview of Entire Output Management's Functions

