White Paper

Sizing and Deployment Guide Crystal Enterprise 10

Authors: Keith Moon and Kevin Crook

Contributors: James Anderson, Scott Cameron, Cortney Claiborne, Davythe Dicochea, Erin O'Malley, Robert Rukavina, James Thomas

Contents

Overview

There are three different aspects to consider when designing a scalable enterprise reporting application: the application itself, the security and system data, and the actual hardware and configuration. Large deployments are often quite complex and it's critical to recognize the set of factors that can influence these three aspects of scalability. This document will focus on the actual hardware and configuration of a Crystal Enterprise™ deployment. This document can be used to estimate requirements for large deployments; however, we do recommend engaging Business Objects Consulting Services to help with planning in these situations

Crystal Enterprise

Standard Configurations

Crystal Enterprise 10 (CE 10) is a reliable, web-based platform for reporting, analysis, and information delivery. It allows users to choose the deployment that makes the best use of resources to offer the best performance. The Crystal Enterprise 10 Administrators Guide includes details on the architecture. In particular, the scalability chapter examines some standard deployment scenarios. Though not the only ways to deploy Crystal Enterprise, they offer a good starting point.

Concurrent Users

When calculating the size and configuration of a deployment, it is important to determine the expected concurrent system usage. Regarding concurrency, it is also important to determine the acceptable response time for a report request. To improve response time, the system should be designed to have adequate processor and memory availability on the report servers to ensure that there is an available Page Server thread for each concurrent user. If this is not possible then the system will queue the requests and manage its back-end processes as efficiently as possible using a combination of cache and job swapping on the Page Servers.Therefore, ensuring adequate processor power and memory amount is important in planning your deployment.

In our experience, many customers find that their concurrency ratios are on average from 10% to 20% of their total user base (e.g., 1000 total users = 100 to 200 concurrent users). This can vary significantly depending on the nature and breadth of the deployment, but is a reasonable rule of thumb for planning purposes. If your estimated concurrency rate is higher or lower, you can use this value to complete the rest of the calculations in this document. A guideline for estimating concurrent users: concurrent users = anywhere from 10% to 20% of total users @ 10%: 1000 users = 100 estimated concurrent users

Reporting Services

(Cache Server–Page Server–Job Server–Report Application Server)

Crystal Enterprise can process reports using two methods. These methods are "on demand"[1] and "scheduled." When a report is run "on demand," it is processed through the Page Server or Report Application Server and served up immediately. When a report is scheduled, it is processed through the Job Server and saved to the file repository server (FRS).

When a report is viewed (either "on demand" or "scheduled") using the Interactive DHTML Viewer, the Report Application Server is used to view the reports. For all other viewers, both the Cache Server and Page Server are used, whether the report is "on demand" or "scheduled."

Cache Server

(For a full architecture diagram please refer to the CE 10 Administrators Guide.)

The Cache Server stores report pages generated by the Page Server. By storing report pages in cache, the database server does not need to be accessed each time the report is requested.

[1] "On demand" reports allow end users to access and retrieve data directly from the database server when a report is viewed. For optimal performance and a superior end-user experience, it is recommended that "on demand" reports be designed to retrieve smaller sets of data (amount varies depending on environment). Reports that retrieve a larger set of data are best suited to being "scheduled."

When a report is requested for viewing, the system will first check the Cache Server to see if there are any current available cache pages generated for that report. If there are pages, the Cache Server will send available report pages to the Web Application Server or Web Component Server. If there are no cached pages, the Cache Server will request a Page Server to generate these pages.

There are two thresholds that impact the performance of the Cache Server. First is the “maximum simultaneous processing threads” per Cache Server service, and second is the “maximum simultaneous processing threads” per processor. A guideline for the maximum simultaneous processing threads per Cache Server service is 400, and a guideline for the maximum simultaneous processing threads per processor is 100.

- 400 “maximum simultaneous processing threads” per Cache Server service
- 100 “maximum simultaneous processing threads” per processor

Example 1

If an environment must service 200 simultaneous Cache Server requests based on the 400 maximum per Cache Server service, one would require only one Cache Server service. And based on the 100 “maximum simultaneous processing threads” per processor, one would require two processors for optimal performance.

- 1 Dual machine w/one Cache Server service: Each Cache Server set to 200 maximum simultaneous processing threads

Example 2

If an environment must service 600 simultaneous Cache Server requests based on the 400 maximum per Cache Server service, one would require at least two Cache Server services. And based on the 100 “maximum simultaneous processing threads” per processor, one would require six processors for optimal performance.

- 1 Quad machine w/two Cache Server service: Each Cache Server set to 200 maximum simultaneous processing threads
- 1 Dual machine w/one Cache Server service: Each Cache Server set to 200 maximum simultaneous processing threads

A "round robin" method is used when selecting Cache Servers, so for proper load balancing, make sure that each Cache Server service is set to the same number of "maximum simultaneous processing threads". One can regulate the set number of “maximum simultaneous processing threads” for the Cache Server service through the Crystal Management Console (CMC).

NOTE:	As the Cache Server service has a direct relationship with the Page Server service, it is recommended that the Maximum Simultaneous Processing Threads for the Cache Server service be set equal to the Maximum Simultaneous Processing Threads of the Page Server service (e.g., If Page Server Maximum Simultaneous Processing Threads is set to 50, then Cache Server Maximum Simultaneous Processing Threads should be set to 50.).

Viewing Scheduled Instances (Page Server)

The Page Server is primarily responsible for responding to page requests by processing reports and generating Encapsulated Page Format (EPF) pages. The EPF pages contain formatting information that defines the layout of the report. The Page Server retrieves data for the report from the latest instance or directly from the database (depending on the user's request and user's security level).

Specifically, the Page Server responds to page requests made by the Cache Server. The Page Server and Cache Server interact closely, so cached EPF pages are reused as frequently as possible, and new pages are generated as soon as they are required. ePortfolio takes advantage of this behavior by ensuring that the majority of report-viewing requests are made to the Cache Server and Page Server. However, if a user's default viewer is the Interactive DHTML viewer, the Report Application Server processes the report.

Default settings for the Page Server allow for 75 simultaneously processing threads; however, for optimal performance the Page Server should be configured for 50. When you are running your reports in "on demand" mode, each Page Server should be allocated one processor.

On Demand Mode (Page Server)

1 Processor = 1 Page Server service = 50 Maximum Simultaneous Processing Threads (Optimal Performance)

When a user views a scheduled instance, the Page Server will load the report and generate the Page on Demand files (.epf) for the user. In this mode run two Page Servers per processor, as this mode is not as processor or memory intensive as viewing "on demand" reports.

Viewing Scheduled Instances (Page Server)

1 Processor = 2 Page Server services (50 Maximum Simultaneous Processing Threads per Page Server) = 100 Simultaneous Processing Threads

Report Application Server

The Report Application Server (RAS) is very similar to the Page Server. It, too, is primarily responsible for responding to page requests by processing reports and generating EPF pages. However, the RAS uses an internal caching mechanism that involves no interaction with the Cache Server. Specifically, the RAS processes reports that ePortfolio users view with the Interactive DHTML viewer. The RAS also provides the reporting capabilities that allow ePortfolio users to create and modify reports over the web.

Because the RAS functions similarly to the Page Server, it should be configured similarly.

1 Processor = 1 Report Application Server service = 50 Maximum Simultaneous Processing Threads (Optimal Performance)

Job Server

The Job Server processes report files (.rpt) as requested by the Crystal Management Server (CMS) and generates report instances (versions of the report that contain saved data). To generate a report instance, the Job Server communicates with the database to retrieve the current data. Multiple Job Servers can facilitate large-scale deployments.

New to Crystal Enterprise 10 is the ability of the Job Server to process Program objects and Packaged objects. Program objects can include executables, batch files, scripts (Java, VB, etc.), or Java programs. Object packages are simply collections of CE objects (e.g., report files) that can be grouped together and managed by the CE system as a single object.

When you are scheduling reports, the reports are run through the Job Server. The Job Server service launches a Job Server child process so each report is run in its own process. Each Job Server service can reliably manage five simultaneous job child processes. The complexity and size of reports can cause this number to vary, but on average five simultaneous jobs can handle any kind of report. You can definitely change this number and testing on individual systems will demonstrate what a reasonable number is. You can reliably run one Job Server service per processor. However, there are some efficiencies when running a Job Server on multiple processor machines. See below.

Schedule Mode (Job Server)

1 Processor = 1 Job Server service = 5 Maximum Jobs (job child) Allowed (Optimal Performance).

If you have a machine with one to four processors, it has been found to be more efficient to have one Job Server on the machine rather than installing multiple Job Servers on a multi-processor machine.

Comparison of Job Server and Page Server

The Page Server is designed to process a large set of smaller reports whereas the Job Server is designed to process a smaller set of very large reports. We can think of the Page Server as the lean, fast thoroughbred racehorse and the Job Server as the strong, robust workhorse.

Smaller reports are less complex and contain a smaller set of data. They are suitable for a large group of users to view as “on demand” reports. The Page Server jobs are run as “threads” rather than “processes,” requiring lesser resources and allowing for a larger number of quick simultaneous jobs.

Larger complex reports that must retrieve and process a very large set of data should be scheduled. The Job Server jobs are run as independent “processes” rather than “threads,” requiring more resources for each job but allowing for efficient processing of a smaller set of large, heavy-processing jobs.

Allocation and Balancing Processors for Reporting

1) If you are running a system where the reports are scheduled at night and then viewed during the day, then you should calculate the maximum number of processors needed for either the Job Server or Page Server functions and allocate that number of processors for report processing.

Jobs Scheduled at Night [Schedule Mode (Job Server)]

100 reports taking 10 minutes each on average to process that must be processed in a four-hour (240 min) time window:

- (# of reports) X (Average Report Process Time)/(Time Window) = # of simultaneous jobs required
- 100 reports X 10 minutes/240 minutes=4.16 (round up) = 5 simultaneous jobs required
- # of simultaneous jobs required/Maximum Jobs Allowed = # of processors
  5 simultaneous jobs required/5 Maximum Jobs (Optimal for Schedule Mode) = 1 processor

Reports Viewed During the Day [Viewing Scheduled Instances (Page Server)]

Estimated 170 users will come in Monday morning to view report instances (simultaneously).

- (# of Concurrent Users)/(# of Page Server Threads per 1 processor) = # of processors required
- 170 users/100 simultaneous processing threads (2 Page Servers) = 1.70 (round up)=2 processors

In the above example, we have calculated that the highest number of processors required at any one time is two and therefore a dual processor machine could be adequate. We would schedule the jobs at night (using the Job Server) and our report instances would be viewed during the day where we estimate that potentially 170 users could view these reports at any one time. To satisfy scheduled jobs at night, we would need to run a minimum of one Job Server; however, as many as two Job Servers could be optimally run on this machine. To satisfy report viewing during the day, two processors would be required with two Page Servers running on each processor for a total of four Page Servers.

2) If you are planning a combination of viewing through “on demand” mode during the day as well as “viewing scheduled instances,” you should allocate processors as if all of the requests were “on demand” requests because you cannot be sure that every request will not be an “on demand” request.

Assume All Reports Might Be Run "On Demand"

Estimated 170 users will come in Monday morning to view “on demand” reports (simultaneously)

- (# of Concurrent Users)/(# of Page Server Threads per 1 processor) = # of processors required
- 170 users/50 simultaneous processing threads (1 Page Server) = 3.40 (round up) = 4 processors

In the above example, we have calculated that the highest number of processors required at any one time is 4, and therefore either two dual processor machines would be required or 1 Quad Processor machine. In this example each of the four processors would be running one Page Server for a total of four available Page Servers. (See Rule of Thumb section below for help on deciding between a Quad or Dual machine.)

3) If you plan to run a system where reports might be scheduled and run on demand concurrently, you should allocate enough processors that meet the "on demand" requirements and the number of processors that will be needed to be serviced by Job Servers

Jobs Schedule During Day

At maximum 10 simultaneous scheduled reports taking 5 minutes on average to process

- (# of reports) X (Average Report Process Time) / (Time Window) = # of simultaneous jobs required
  (10 reports) X (5 minutes) / (5 minute) = 10 (round up) simultaneous jobs required
- # of simultaneous jobs required/ Maximum Jobs Allowed = # of processors
  10 simultaneous jobs required / 5 Maximum Jobs (Optimal for Schedule Mode) = 2 processor

Assume All Reports Might Be Run "On Demand"

Estimated 170 users will come in Monday morning to view "on demand" reports (simultaneously)

- (# of Concurrent Users) / (# of Page Server Threads per 1 processor) = # of processors required
  170 users / 50 simultaneous processing threads (1 Page Server) = 3.40 (round up) = 4 processors

Since, in this scenario, scheduling and report viewing can happen simultaneously, it is important to allocate enough processing power to cover the event where 100% of simultaneous users requests might be scheduling, or where 100% of simultaneous user requests might be viewing on demand. In this example, we have calculated that the total number of processors required would be five, therefore a combination of quad, dual, and or a single processor machines would be required. We would optimally run a minimum of one Job Server and a minimum of four Page Servers to cover all potential requests.

Rule of Thumb

If you are running all components on one machine, Crystal Enterprise can support 50 simultaneous user requests viewing on demand reports per processor, and 100 simultaneous user requests if they are viewing instances. In addition to providing more capabilities to service simultaneous user requests, the system will perform better as you scale the system vertically (by adding processors and memory).

NOTE:	To optimize system performance on the Page Server or Job Server, network settings on Microsoft NT Server or Windows 2000 Server should be set to "Maximize Throughput for Network Applications." This will give a higher priority to applications running on the server. Microsoft recommends this setting when you are running their server applications (like SQL Server and, particularly, Exchange Server). We recommend this setting for your Crystal Enterprise Job, Page, and Report Application Server installation.

Minimum Hardware Requirements for Crystal Enterprise 10

Windows

- P3 700MHz
- 512 Megs RAM
- 2 Gigs available hard drive space
- CD Rom

Solaris

- 500 MHz single CPU
- 1 Gig RAM
- 2 Gigs available hard drive space
- CD Rom

AIX

- 500 MHz single CPU
- 1 Gig RAM
- 2 Gigs available hard drive space
- CD Rom

Memory

Page and Job Server memory usage is affected by the size and complexity of a report as well as the amount of data being processed. As a guideline, for a production environment, it is recommended that a minimum of 512MB RAM be allocated per processor.

Disk Space

For the Page and Job Server services, sufficient hard drive disk space should be available in the temp directory for the creation of temp files. Temporary files are created during report processing. The data from the database server is stored in these files until it can be saved and compressed in the report. Correspondingly, the hard drive access speed to the temp directory will have an impact on the speed at which a report processes. When Job Server, Page Server, or Report Designer opens a report, the saved records are expanded. The compression ratio is, on average, a 15:1 compression ratio.

When the Page Server is called upon when viewing a scheduled instance, the Page Server checks out a copy of the report from the Output File Repository Server. The copy is temporarily stored in the MachineName.pageserver\Temp\procReportTemp\mgrTemp directory. The copy is kept for as long as specified in the Crystal Management Console under Minutes before an Idle Job is Closed: (Servers|Page Server|Properties). For this reason, there must be a sufficient amount of hard drive disk space available to store these temporary report file copies.

Similarly, when the Report Application Server is used to view an instance, it obtains a copy of the report from the Output File Repository Server. The copy is temporarily stored in the location specified by the TEMP system environment variable. It will remain there until the time specified in Crystal Management Console under "User Inactivity Timeout" (Servers|Report Application Server|Server).

For the Cache Server service, sufficient hard drive disk space should be available for the generation of cache files as well as in the temp directory for the creation of temp files. The location of Cache Server cache files can be specified from the Crystal Management Console. The highest volume of cache files will likely be the ".epf." A single .epf file represents one page of a report and the size of the individual file is variable dependent on the complexity of the report page (e.g., a single .epf file from the sample "Statement of Account" report is 88kb in size). A sufficient amount of disk space should be made available to support all potential cache pages generated at a given period.

NOTE:	To further increase performance, it is also recommended, if possible, to assure that different temp directories are set for each Page, Job, and Cache Server service (location of temp files can be specified in the Crystal Management Console).

Web Application Server

Depending on how the system is being utilized, the Web Application Server (WAS) can manage differing number of concurrent user sessions and simultaneous requests. The main functions of the Web Application Server are:

1. Processing the ASP/JSP script
2. Translating the Encapsulated Page Files (page on demand) to DHTML pages.
3. Communicating with Cache Server for report view requests
4. Managing session state information for the users

Processor

1 Web Application Server can manage approximately 400 concurrent user sessions (user session = 1 logged on user) per processor. And generally, a service can efficiently manage 100 simultaneous requests (request = e.g., a user clicking on a folder). Under normal circumstances it is improbable that all concurrent users would make a request simultaneously, therefore the following numbers allow for and differentiate between "concurrent user sessions" and "simultaneous requests." Because the service deals with two thresholds (Maximum number of Concurrent User Sessions and Maximum number of Simultaneous Requests), it is important to consider both when determining the required hardware.

This can be illustrated in the following examples:

Example 1
1 Single processor machine (1 processor) with 1 Web Application Server service running could efficiently service 400 concurrent user sessions and can handle 100 simultaneous user requests.

Example 2
1 Dual processor machine (2 processors) with 2 Web Application Server services running could efficiently service 800 concurrent user sessions and can handle 200 simultaneous user requests.

Example 3
1 Quad processor machine (4 processors) with 4 Web Application Server services running could efficiently service 1600 concurrent user sessions and can handle 400 simultaneous user requests.

Please refer to Appendix A for help on estimating how many simultaneous user requests can be expected in a particular environment.

The number of simultaneous requests per processor (100) is a good guideline number; however, it is important to note that items 1 and 2 from the point list above (Processing the ASP or JSP script, Translating the Encapsulated Page Files to DHTML pages) can impact this number.

1) The complexity and amount of data that is requested by the script can impact the number of concurrent users per processor. Please see the performance section in the Query Language Reference in the Web Developer guide for a detailed reference on how to optimize your application. Generally, the complexity and amount of data requested will not limit the number of concurrent users the Application Server can handle, but it will impact user experience through response time of the system.

2) When DHTML is being used for viewing of reports, the WAS can manage a maximum of 75 simultaneous user requests per processor, as the WAS will have added processor load of exporting the EPF files to DHTML. However, in benchmark testing, the optimal number will be 50 simultaneous user requests per processor. A factor that can affect this number is the "readability" of the reports or how users utilize or view the reports. If you expect users to navigate to a page or section of a report and then spend an amount of time looking at or reading that page or section, the maximum number of simultaneous requests per processor on the Web Application Server can increase.[2]

It is recommended that you run one Web Application service per 100 simultaneous users if viewing with the Active x or Java Viewers, and one Web Application service per 50 simultaneous user requests if using DHTML.

Viewing in ActiveX or Java Viewers

400 concurrent users = 100 simultaneous user requests = 1 Web Application Server service = 1 Processor

Viewing in DHTML Viewer

Maximum = 400 concurrent users = 75 simultaneous user requests = 1 Web Application Server (WAS) service = 1 Processor

Optimal = 400 concurrent users = 50 simultaneous user requests = 1 Web Application Server (WAS) service = 1 Processor

NOTE:	If you are using automated testing scripts remember these test scripts conduct tests of simultaneous transactions. When calculating concurrent users, plan for users not all clicking at the same time. Generally, they are staggered.

Supporting Other Services on the Web Application Server

If deployments are also using the Crystal Analysis® Professional component or the Report Application Services, then these services also take advantage of the WAS and will impact the amount of users who can view reports if the WAS needs to simultaneously support these other services.

[2] Benchmark numbers actually reference simultaneous requests, not users. Simultaneous requests assume that a user never pauses to read a page they are presented and that every user makes a request at the exact same instant. Numbers in real-world scenarios may be higher.

CMS Repository and Scheduling Services

Crystal Management Server

The principle component of the Crystal Enterprise system is the Automated Process Server. Its primary responsibilities include (but are not limited) to the following:

- Management of users and user groups
- Management of object security
- Central system configuration
- Management of scheduled tasks
- Management of historical tasks (e.g., success or failed instances)
- Management of system objects (i.e., Info Objects)
- License management
- System database management
- Name service
- Cluster management

Processor

Large amounts of batch scheduling may dramatically increase the workload of the CMS. Additionally, any large updates to the CMS system database (i.e., adding or deleting a large number of users; viewing or querying a large number of objects) will use intensive CPU time.

Memory

The CMS can manage an unlimited number of users and groups; however, the existence of users does create an overhead on the CMS, as a private folder is created for every user added to the system.

CMS Default Startup Memory Usage = 27MB

CMS Default DB size = 2.13MB

CMS Default Total Objects = 105

1 User Object = 5.9 KB loaded into (CMS) CrystalMS.exe memory

The CMS is smart about memory usage and it will unload any users and objects that are not currently logged on to the system. This is a particularly important consideration for SDK developers when using session handling within the Crystal Enterprise environment. Anytime a session is released (for example, releasing and restarting sessions between page requests), this is effectively logging the user off of the CMS system temporarily (releasing a license). At the point that the session is released, the CMS could unload any related objects to this user. When the session is restarted upon new page request, the CMS would once again have to reload these objects into memory.

Also, the more individual accounts and objects that need to be loaded by the CMS, the more memory it will use. In addition, more objects contained in a particular level under a user will create more CMS overhead when attempting to access or view that level. The CMS will work more efficiently, and user response times for viewing will be faster if objects are divided up into a hierarchy rather than all objects being contained at one level.

1 Folder Object = approx 20KB loaded into CrystalMS.exe memory

1 Report Object = approx 10KB loaded into CrystalMS.exe memory

1 Report Instance = approx 10KB loaded into CrystalMS.exe memory

There are many factors that can increase the amount of memory required. For example, report instances that contain numerous prompts will be larger than average as every string prompt requires 256 bytes. The CMS will, in normal operation, only keep the most recently accessed objects in memory and stay within its memory limits as defined below. However, during periods of rapid object access (like batch reporting), the CMS may exceed the specified amount of memory. To determine the amount of memory to put on a CMS machine the person deploying the system must determine how many objects and instances they expect to have in the system. Performing the following calculation should help in estimating the amount of memory that the CMS will need to operate optimally.

Recommended Megabytes of CMS Memory = (10 * total Objects)/1000 + (10 * total Instances)/1000 + (20 * Total Folders)/1000 + 27

Once the maximum amount of memory required by the CMS is calculated, you can subsequently determine how much system memory you want to reserve for the CMS. The following registry key on the CMS determines how much memory the CMS will use.

HKEY_LOCAL_MACHINE\Software\Crystal Decisions\10.0\CMS\Instances\CMSInstanceName\MaxMemoryUsagePercent

The default setting is 75% of existing system memory. It is recommended that this generally be between 50-80% of the total system memory. The smaller the disparity between the calculated memory requirements and the allocated system memory, the better Crystal Enterprise will perform.

A command line setting also controls the number of objects in memory. By default, the maximum number of objects in memory is 10,000. However, if a CMS has sufficient memory, it can be beneficial to increase this to 100,000. To do this, modify the command line startup for the CMS to:

\\machinename\C$\Program Files\Crystal Decisions\Enterprise 10\win32_x86\CrystalMS.exe" -service -name COHO.cms -restart -threads 50 –maxobjectsincache 10000

Crystal Enterprise System Database

A Microsoft Data Engine (MSDE) database is created by default (Note: CMS clustering is automatically supported by the default MSDE database). MSDE is a client/server data engine that provides local data storage and is compatible with Microsoft SQL Server 7.0. If you already have MSDE or SQL Server 7.0 installed, the installation program uses it to create the CMS database.

Because MSDE uses SQL Server 7.0, if the Crystal Enterprise 10.0 installation program detects that you have SQL Server 6.5 installed it will not install a CMS on that machine. You need to upgrade to SQL Server 7 in order to install the Crystal Enterprise system database on that particular machine.

For production environments, it is recommended that the CMS database not be placed on:

1) The database server against which the reports are running
2) Any one of the Job Server or Page Server machines
3) One of the CMS machines.

The reason for this is that if the Database Server, Job Server, or Page Server is busy processing the data for reports, the retrieval time for the CMS to query the same server will be negatively impacted. Please refer to the Crystal Enterprise Installation guide for information on supported CMS databases.

The largest table in a CMS system database is the APS_InfoObjects table. This table contains all the string, numeric, and binary fields that represent the Crystal Enterprise environment. The CMS database should be allocated with enough space to house all the objects and instances as calculated above. For example, the APS_InfoObjects table stores the report instance information and each record ranges in size from 6 to 12 KB (with an average of 10 KB). Note that the same factors that affect the size of a report instance in memory (see Memory section above) also affect the size of records in the APS_InfoObjects table.

If you are using Microsoft SQL Server or any of Sybase's SQL databases, you must also consider the transaction logs. The largest tables in the database will frequently be written to when jobs are being processed; hence the resulting transaction log can get quite large. You should consider setting aside an amount equal to two thirds of the size of the CMS system database for this transaction log. If problems persist, consider scheduling a stored procedure to periodically dump the transaction log.

NOTE:	By default, the CMS uses two threads to communicate with the system database. This should be increased to 10 threads where the number of concurrent, active users is greater than 100. To do this, go into the Crystal Configuration Manager, modify the properties of the CMS, and in the command line add the switch "-ndbqthreads 10" (without the quotation marks).

Clustering

The CMS supports clustering capabilities. This capability allows two or more CMS machines to share Crystal Enterprise information. This results in an increased number of potential concurrent users and scheduling requests and provides high availability through machine failover support.

So the clustering of CMS machines may be advisable under the following conditions:

1) Required Fault Tolerance: If the one CMS machine shuts down, another CMS machine seamlessly takes over workload.
2) The number of potential concurrent users or scheduling requests exceeds acceptable performance capacity of a single CMS machine.

In respect to point two above, a good point to start considering a clustered environment would be when the number of concurrent users logging on to a CMS approaches or exceeds 600. This number is only a guideline and as such can be affected by the types of tasks that are most usually performed. Tasks where users log in and log off of the CMS, users navigate through folders, or when there are a large number of scheduling requests being made are the most taxing on the CMS. In contrast, the CMS has less contention when users are logged on to the system and stay within a particular folder (e.g., personal or group folder) and stay there to view a report or set of reports.

File Repository Servers

There is at least one input and one output File Repository Server (FRS) in every Crystal Enterprise implementation. The input FRS manages all the report objects that have been added by the administrator or end user via the Crystal Publishing Wizard, the Crystal Management Console, or the Crystal Import Wizard. The output FRS maintains all the report instances (Crystal Report® files with saved data) generated by the Job Server. The File Repository Servers are not processor intensive executables. You may have multiple FRS (input) and FRS (output) services on several different machines to support a high-availability environment.

Input File Repository Service

The location of where the input File Repository Server stores the input repository must have a sufficient enough disk space to store all report objects (templates) that have been added by the administrator or end user using the Crystal Publishing Wizard, the Crystal Management Console, or the Crystal Import Wizard. The size of an object varies depending on the complexity of a report. The input FRS only holds the report template so typically the reports are not that large in size. The sample reports that ship with CE are on average 250 KB without data.

Output File Repository Service

The location of where the output File Repository Server stores the output repository must have sufficient enough disk space to store all instances (report files with saved data) generated by the Job Server.

Repository Location

The input and output Repository do not have to reside on the same machine. The location of the FRS repositories is managed through the CMC in the Servers section under the Properties tab.

NOTE:	To optimize system performance on the File Repository Servers, network settings on Microsoft NT Server or Windows 2000 Server could be set to “Maximize Throughput for File Sharing.” This will give a higher priority to file sharing applications.

Appendix A–Questions for Server Sizing

The following questions are helpful in estimating the minimum number of machines and processors, and the configuration and number of Crystal Enterprise server components. The answers to these questions are applicable when making use of this sizing guide as well as useful information to Crystal Enterprise consulting experts for Crystal Enterprise deployments.

General User Numbers	
How many users will you have?	
How many users will be on the system concurrently (in %)?	
Concurrency Breakdowns	
What % of concurrent users will be heavy users? (Users constantly logged onto the system viewing reports almost continuously. Heavy users request a URL every second.)	
What % of concurrent users will be active users? (Users logged onto the system frequently throughout the day. Active users request a URL every 10 seconds.)	
What % of concurrent users will be moderate users? (Users logged onto the system from time to time throughout the day. Moderate average one URL request every 15 seconds.)	
What % of concurrent users will be light users? (Users logged onto the system infrequently. Light users request a URL every 20 seconds.)	
Web Requirements	
Do you intend to primarily use the DHTML viewer?	
What % of users will be running reports on demand?	
Scheduling Requirements	
Will jobs be scheduled?	
Will jobs be scheduled primarily during off hours?	
Is there a reporting time window?	
How long is the reporting window in minutes?	
How many report instances need to be created?	
How long do you estimate each report will take (min)?	
System Requirements and Complexity	
Is fault tolerance required?	
How many report templates do you expect to have?	

Appendix B–Calculating the Number of Simultaneous User Requests

The number of simultaneous user requests is difficult to estimate; however, if we divide the users into sets of types of users, based on how they use the system, we can more accurately assume a number.

For the purposes of this calculation, we will divide users into four types:

- Heavy Users: users who will be constantly logged onto the system viewing reports nearly continuously.
- Active Users: users who are logged into the system frequently throughout the day averaging one request every 10 seconds.
- Moderate Users: users who are logged into the system from time to time throughout the day averaging one request every 20 seconds.
- Light Users: users who will log into the system infrequently and will view a couple of reports and logout.

Divide concurrent users into these four defined categories and then calculate the percentage of each type.

1000 users = 100 concurrent users (estimating at 10%)

Heavy Users	15 concurrent users	15%
Active Users	45 concurrent users	45%
Moderate Users	25 concurrent users	25%
Light Users	15 concurrent users	15%
Total	100 concurrent users	100%

Now that we have determined the percentage of each type of user, we can calculate the final number. Assumptions have been made relating to the rate of simultaneous use based on user group type:

- For every 100 heavy concurrent users, we will assume 100 simultaneous requests could be made or a 100% rate.
- For every 100 active concurrent users, we will assume that 25 simultaneous request could be made or a 25% rate.
- For every 100 moderate concurrent users, we will assume that 12 simultaneous requests could be made or a 12% rate.
- For every 100 light concurrent users, we will assume that six simultaneous requests could be made or a 6% rate.

Formula:

(Concurrent Users X % of Heavy Users / 100) + (Concurrent Users X % of Active Users / 100 * 0.25) +(Concurrent Users X % of Moderate Users / 100 * 0.12) + (Concurrent Users X % of Light Users / 100 * 0.06) = Calculated Simultaneous Users (rounded up)

–or–

(100 X 15 / 100) + (100 X 45 / 100 * 0.25) + (100 X 25 / 100 * 0.12) + (100 X 15 / 100 * 0.06) = 31

Based on the assumption of 100 concurrent users and the types of activities each user is likely to perform, we have calculated that there will be an average of 31 simultaneous user requests.

Appendix C–The Four Basic Steps to Sizing Crystal Enterprise

1) Determine total number of potential users, concurrent users, and simultaneous user requests.

Potential Users (# of users that are able to access system)	
Concurrent Users (e.g., sessions, connections)	
Simultaneous User Requests (e.g., viewing a report, opening a folder, logging on)	

2) Determine # of services required for CMS, WAS, CS, JS.

Number of CMS Services	
Number of WAS Services	
Number of CS Services	
Number of JS Services	

3) Fine-tune services and determine # of Page Server services.

Each Cache Server “Maximum Simultaneous Processing Threads”	
Number of PS (Page Server service\s)	
Each Page Server “Maximum Simultaneous Processing Threads”	
Each Job Server “Maximum Jobs Allowed”	

4) Determine number of processors and machines.

Number of CPUs for CMS	
Number of CPUs for WAS	
Number of CPUs for CS	
Number of CPUs for PS	
Number of CPUs for JS	

Appendix D

The size of the Active X Viewer is: 932 KB (This is a one-time download)

The size of the Java Viewer is: 368 KB/649 KB (IE/Netscape—Download once per browser session)

Getting More Information

For more information, look at the following documentation or contact technical support.

Product Documentation

Documentation is available in printed documentation or in electronic format on the Crystal Enterprise/Crystal Reports® CD and on the online store (https://secure.businessobjects.com).

- Crystal Installation Guide
- Crystal Enterprise Administrator's Guide
- Crystal Enterprise Disaster Recovery Planning (DRP) (http://support.businessobjects.com)

Technical Support

Technical support web site:
http://support.businessobjects.com

Answers by email support:
http://support.businessobjects.com/support/answers.asp

Phone support:
Tel: (604) 669-8379

Americas
Business Objects Americas
3030 Orchard Parkway
San Jose, California 95134
USA
Tel: +1 408 953 6000
+1 800 877 2340

Asia-Pacific
Business Objects Asia Pacific Pte Ltd
350 Orchard Road
#20-04/06 Shaw House
238868
Singapore
Tel: +65 6887 4228

Europe, Middle East, Africa
Business Objects SA
157-159 rue Anatole France
92309 Levallois-Perret Cedex
France
Tel: +33 1 41 25 21 21

Japan
Business Objects Japan K.K.
Head Office
Yebisu Garden Place Tower 28F
4-20-3 Ebisu, Shibuya-ku
Tokyo 150-6028
Tel: +81 3 5447 3900

For a complete listing of our sales offices, please visit our web site.

▶ **www.businessobjects.com**

PT# WP2085-A

Printed in France and the United States – July 2004.