

BusinessObjects™ XI

RELEASE 2

Enterprise Sizing Guide

Overview

Enterprise system sizing involves the process of determining how many resources are required to support a given workload.

As input in the sizing process we often have:

- The number of users
- A description or understanding of types and percentage of usage types (how many users are doing what)
- What frequency or how often users are interacting

As output in the sizing process we want to understand:

- The number of processors (CPUs) required to support a load
- The amount of memory required to support a load
- The appropriate configuration for a load type (number of services and where to put them)

The general assumption through this document is that each main service or component can be assessed independently through its capacity to respond to simultaneous requests from users in a time period with a certain amount of hardware resources (CPU and Memory).

OVERVIEW	1
INTRODUCTION	3
HOW TO USE THE SIZING GUIDE FOR WINDOWS AND UNIX ENVIRONMENTS.....	4
STEP 1: LOAD	5
ESTIMATING POTENTIAL USERS	5
ESTIMATING CONCURRENT ACTIVE USERS	5
ESTIMATING SIMULTANEOUS REQUESTS.....	5
STEP 2: SERVICES	7
INTELLIGENCE TIER.....	9
<i>Central Management Server</i>	9
<i>Crystal Reports Cache Server</i>	11
<i>File Repository Servers (FRS)</i>	12
<i>Search Server (for “Content Search” only)</i>	13
<i>Event Server</i>	15
PROCESSING TIER.....	15
<i>Desktop Intelligence</i>	15
<i>Performance Manager</i>	17
<i>Web Intelligence Report Server</i>	18
<i>Web Intelligence Job Server</i>	19
<i>Crystal Reports Page Server</i>	20
<i>Crystal Reports Job Server</i>	24
<i>Report Application Server</i>	25
<i>List of Values Job Server</i>	26
<i>Connection Server</i>	27
APPLICATION TIER	27
<i>Web Application Server</i>	27
<i>OLAP Intelligence functionality within the Application Tier</i>	28
<i>Business Objects Voyager OLAP server</i>	30
<i>Query as a Web Service (QaaWS) special settings</i>	32
<i>Live Office special settings</i>	32
BASIC ONE MACHINE CONFIGURATION (VERTICAL SCALING).....	34
MULTIPLE MACHINE CONFIGURATIONS TYPES (HORIZONTAL SCALING).....	34
DISK I/O AND NETWORK BANDWIDTH AND LATENCY	35
STEP 4: SYSTEM DATABASE TUNING	36
SYSTEM DATABASE BACKEND PERFORMANCE	36
CONNECTION BETWEEN ENTERPRISE XI AND THE DATABASE SYSTEM	36
PERFORMANCE CRITERIA	36
CALCULATING THE DATABASE FILE SIZE.....	38
USEFUL RESOURCES.....	39
FINDING MORE INFORMATION	39

Introduction

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 is critical to recognize the set of factors that can influence these three aspects of scalability. This document will focus on the major aspects that influence sizing decisions including information on actual hardware and configuration of a Business Objects Enterprise XI™ 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.

A step-based methodology for sizing is used in this guide. From start to finish the following four steps will help to plan and estimate the size of a deployment:

STEP 1: Load

In Step 1 of the sizing process, the load is determined. Load defines the amount and types of user/client activity - transactions that will interact with the Enterprise XI System. The results obtained in this step are needed to perform the necessary calculations in Step 2.

STEP 2: Service Thresholds, CPU and Memory

In Step 2 of the sizing process, an analysis is completed looking at the thresholds of each required service in the system. Services are the Enterprise XI software components that are required to provide and support the anticipated load and functionality. Also in Step 2, the number of CPUs and amount of memory required is determined. CPUs and memory address the supporting infrastructure that must support the services and load.

STEP 3: Configuration & Infrastructure Characteristics

In Step 3 of the sizing process, the appropriate configuration of machines is determined. For example, two Dual boxes as opposed to one Quad or the configuration that achieves the correct balance of reliability and performance. Additionally, it is important to consider minimum disk I/O speeds and network latency measures for various components of the infrastructure.

STEP 4: CMS System Database Tuning

In Step 4 of the sizing process, the underlying database server tuning is addressed. Tuning the database server and network to take full advantage of Enterprise XI will help to provide optimal performance and reliability.

How to use the Sizing Guide for Windows and UNIX environments

This Sizing Guide is a general guide. Enterprise XI is a highly flexible system and there are many possible variables that can impact how an optimal configuration might look. This guide offers conceptual information and measurements that reflect observed Enterprise XI system and component behavior. Sizing formulas in Step 2 can assist in understanding the nature of relationships between user interactions and service functionality as well as how this relates to CPU utilization, memory and/or disk consumption. For the purposes of this guide, most system and performance testing was conducted on machines averaging at 2.5 GHz clock speed with 2GB of memory per CPU.

Use this guide to better understand how each applicable Enterprise XI Service (Server) behaves, including relative capacities and thresholds within the overall Enterprise XI system.

This guide can be used for both Windows and UNIX environments. Acknowledging the myriad of variables (inherent CPU differences, hyper threading, operating system, network, report design, functionality, SDK, and so on) that impact sizing and performance - and within the degree of accuracy possible to size a complete Business Intelligence system - the concepts, formulas, configurations and observations can be used universally.

This Sizing Guide is a living document that will be adapted and changed as new and valuable sizing/configuration information becomes available.

Step 1: Load

What is Load?

Load defines the amount and types of use and activity that will interact with the Enterprise XI System. Load can be broken down to various types of user interactions and user types:

- **Potential users (Named users)** - is the number of users that are able to log on to the system.
- **Concurrent active users** - is an estimate of the number of users who are concurrently logged on to the system and are actively interacting with the system (clicking folders, viewing reports, scheduling, and so on). There is a distinction between *concurrent active users* and *concurrent users*. Concurrent active users are actively interacting with the system at various rates, rather than being logged on but inactive.
- **Simultaneous requests** - is an estimate of the total number of simultaneous requests that are being made by concurrent active users to the system. These requests include logging, clicking folders, viewing a report page (from the cache or not), opening a WebIntelligence document, scheduling, refreshing, and so on. The number of simultaneous requests is generally dependent of the usage type and rate.

Concurrent active users and simultaneous requests are the two load types that will most determine the required resources and the appropriate configuration to support a high-performance and highly-reliable Enterprise XI system. The number of potential users is useful in determining these three measurements.

Estimating Potential Users

This is the easiest number to calculate as this is the total population of users who have the ability to access the Enterprise XI environment

Estimating Concurrent Active Users

When calculating the size and configuration of a deployment, it is important to determine the expected concurrent system usage.

In our experience, many customers find that their concurrency ratios are on average from 10% to 20% of their total potential user base (1000 total potential users = 100 to 200 concurrent active 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 active users: concurrent active users = anywhere from 10% to 20% of total potential users

@ 10%: 1000 potential users = 100 estimated concurrent active users

Estimating Simultaneous Requests

The quickest method for estimating the number of Simultaneous Requests is to calculate 10% of Concurrent Active Users

1000 concurrent active users x 10% = 100 Simultaneous Requests

The following process is an example of one methodology that might be used to estimate this number in more detail. 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. The percentages used in this example can be adjusted to match the profile of a particular environment.

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

- **Heavy Users** – users who are constantly logged on to the system viewing reports nearly continuously.
- **Active Users** – users who are frequently logged on to the system, averaging one request every 4 seconds.
- **Moderate Users** – users who are logged on to the system from time to time, averaging one request every 8 seconds.
- **Light Users** – users who log on to the system infrequently, view a couple of reports and log out, having an estimate of one request every 16 seconds.

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

Example:

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

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

Now that we have determined the percentage breakdown for each user set we can estimate the total simultaneous requests based on the estimated user type request frequency (in other words, how we define what a "Heavy concurrent active user" means in terms of request frequency or rate).

The following percentages are assumptions that have been made relating to the rate of simultaneous use based on user group type:

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

Formula to calculate simultaneous user requests:

$$(((\text{Concurrent active users} \times \% \text{ of Heavy Users}) / 100) * (1)) + (((\text{Concurrent active users} \times \% \text{ of Active Users}) / 100) * (0.25)) + (((\text{Concurrent active users} \times \% \text{ of Moderate Users}) / 100) * (0.12)) + (((\text{Concurrent active users} \times \% \text{ of Light Users}) / 100) * (0.06)) = \text{Calculated Simultaneous Users (rounded up)}$$

- or -

$$(((100 \times 15) / 100) * (1)) + (((100 \times 45) / 100) * (0.25)) + (((100 \times 25) / 100) * (0.12)) + (((100 \times 15) / 100) * (0.06)) = 31$$

Based on the assumption of 100 concurrent active 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.

Step 2: Services

Step 2 assists with estimating the number of Enterprise XI Services required in supporting the user activity (Load) estimated in Step 1.

Through intensive mission critical subsystem testing during the development and quality assurance process as well as through large scale end-to-end user benchmarks executed at our hardware partner labs, specific thresholds and capacities as well as the individual services' ability to scale are determined for each service. This section of the sizing and configuration guide outlines the specific nature of each Enterprise XI Service as determined through this process.

The Enterprise XI suite consists of many core services, some of which are essential to system operation and others that are optional. It is during this step that it is determined which services are required and how many are needed to provide the desired functionality and optimal performance.

Figure 1 - BusinessObjects Enterprise XI Services

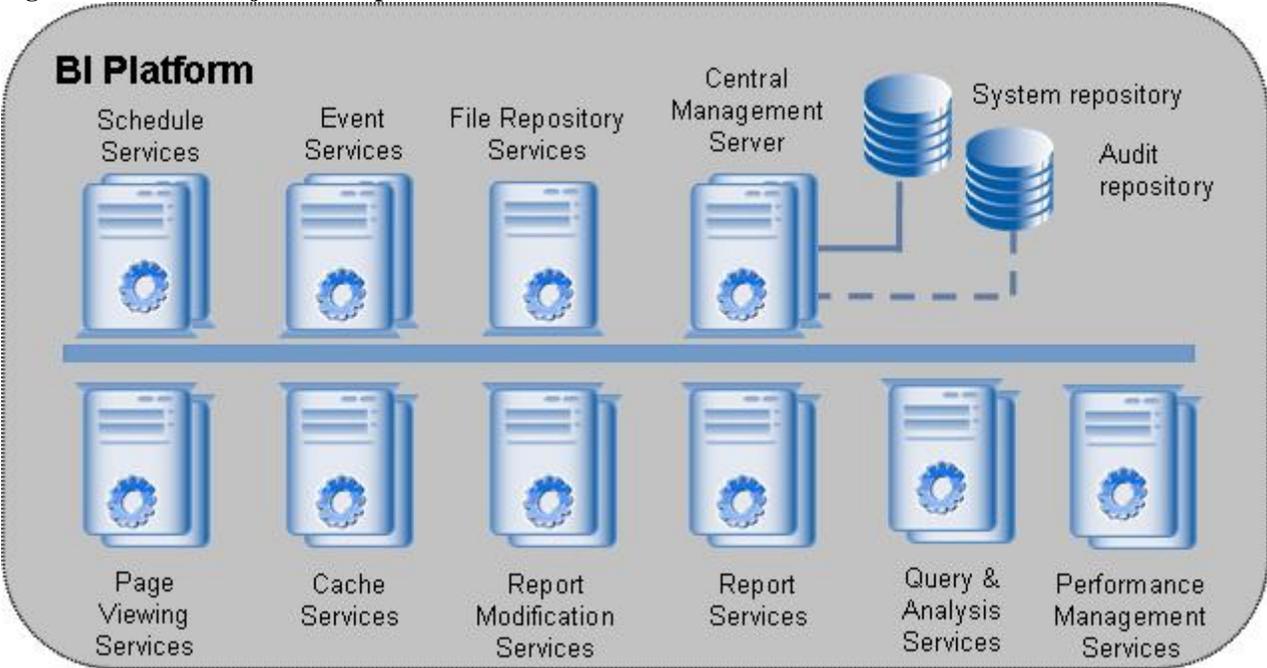


Figure 2 - Server Services as seen in the Central Configuration Manager

Display Name	Version	Status	Description
World Wide Web Publishing Service	5.2.3790.0	Running	Provides Web connectivity and administration through the Internet Information Services Manager
WinHTTP Web Proxy Auto-Discovery ...	5.2.3790.0	Running	
Web Intelligence Report Server	11.5.0.313	Running	Processes Web Intelligence Documents
Web Intelligence Job Server	11.5.0.313	Running	Handles off-loaded processing of Web Intelligence Documents.
Report Application Server	11.5.0.313	Running	Provides service for building and customizing of Crystal Reports documents.
Program Job Server	11.5.0.313	Running	Processes scheduled program objects.
Output File Repository Server	11.5.0.313	Running	Provides object storage and management services.
List of Values Job Server	11.5.0.313	Running	Processes scheduled List of Values objects
Input File Repository Server	11.5.0.313	Running	Provides object storage and management services.
Event Server	11.5.0.313	Running	Monitors events
Destination Job Server	11.5.0.313	Running	Processes destination objects
Desktop Intelligence Report Server	11.5.0.313	Running	Desktop Intelligence Report Server
Desktop Intelligence Job Server	11.5.0.313	Running	Handles off-loaded processing of Desktop Intelligence documents.
Desktop Intelligence Cache Server	11.5.0.313	Running	Desktop Intelligence Cache Server
Crystal Reports Page Server	11.5.0.313	Running	Generates report pages requested by report viewers.
Crystal Reports Job Server	11.5.0.313	Running	Handles off-loaded processing of Crystal Reports documents.
Crystal Reports Cache Server	11.5.0.313	Running	Stores report pages frequently requested by report viewers.
Connection Server	11.5.0.313	Running	Connection Server
Central Management Server	11.5.0.313	Running	Provides scheduling, security, and system management services.
Apache Tomcat 5.0.27	1.0.0.0	Running	Tomcat Application Server
AA Statistical Process Control Server		Running	Provides process control analytic services
AA Set Analyzer Server		Running	Provides set analytic services
AA Repository Management Server		Running	Provides AF repository services
AA Predictive Analytic Server		Running	Provides predictive analytics services
AA Metric Aggregation Server		Running	Provides metrics services
AA Individual Profiler Server		Running	Provides individual analytic services
AA Dashboard Server		Running	Provides dashboard management services
AA Analytics Server		Running	Provides analytic processing, rendering and transformation services
AA Alert & Notification Server		Running	Provides Alerting services

Services:

Intelligence Tier

- [Central Management Server \(CMS\)](#)
- [Crystal Reports Cache Server](#)
- [Input/Output File Repository](#)
- [Search Server](#)
- [Event Server](#)

Processing Tier

- [Desktop Intelligence Report Server](#)
- [Desktop Intelligence Cache Server](#)
- [Desktop Intelligence Job Server](#)
- [Enterprise Performance Manager Services](#)
 - [Dashboard Manager](#)
 - [Analytics](#)
- [Web Intelligence Report Server](#)
- [Web Intelligence Job Server](#)

- [Crystal Reports Page Server](#)
- [Crystal Reports Job Server](#)
- [Report Application Server](#)
- [List of Values Job Server](#)

Application Tier

- [Web Application Server](#)
- [OLAP Intelligence Functionality](#)
- [Business Objects Voyager OLAP server](#)
- [Query as a Web Service](#)
- [Live-Office](#)

Intelligence Tier

Central Management Server

The principal component of the Enterprise XI system is the Central Management Server (CMS). 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 (such as successful or failed instances)
- Management of system objects (such as Info Objects)
- License management
- System database management
- Name service
- Cluster management

Calculate Number of CMS Services Required

The number of CMS services required in a system depends on:

- The number of concurrent active users
- The number of simultaneous user requests involving viewing or querying CMS objects (such as viewing documents)
- If there is a high volume of batch scheduling
- If software fault tolerance is required (clustering)

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

The clustering of CMS Services may be advisable under the following conditions:

- The number of potential concurrent users or scheduling requests exceeds acceptable performance capacity of a single CMS Service.
- A high volume of batch scheduling requests are being made and requests exceed acceptable performance capacity of a single CMS Service.
- Required Fault Tolerance – if the one CMS Service shuts down, another CMS Service is available to seamlessly take over workload.

A common guideline used to help determine when to consider creating a clustered environment (adding additional CMS services) is when the number of concurrent active 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.

CMS guidelines for Clustering:

- For every additional 600 concurrent active users
- If software fault tolerance is desired

Example:

- If estimating 1,400 Concurrent Active Users, consider including a cluster of two or three CMS services in the configuration.
- Include at least two CMS services for Required Fault Tolerance – if one CMS service is shut down, another CMS service takes over the workload.

CMS Clustering across Subnets

A cluster that has two or more CMS cluster members on different subnets is technically possible and has been QA tested. This configuration is supported by Business Objects, strictly providing that no significant additional network latency is created as a result of an additional subnet.

The most important factor to ensure efficient CMS clustering performance is to eliminate excessive latency between CMS services and the CMS Database.

Example:

CMS1 and the CMS system database are located in the same datacenter in New York. CMS2 is a member of the same cluster as CMS1 but is located in China and must communicate with the CMS database in New York. Excessive network latency of CMS2 in China to the CMS database in New York would be problematic.

Ensure that all CMS members of a cluster have uniform communication speeds to the system database. For best performance, run each CMS cluster member on a machine that has the same type of CPU. For more detailed clustering information refer to the online [Enterprise XI Administrator's guide](#).

Processor Requirements

The number of CPUs required to support CMS services will be highly dependent on the type of CMS activity. For example, large updates to the CMS system database (such as adding or deleting a large number of users; viewing or querying a large number of objects) will use a large amount of CPU time. For increased CMS throughput and response times, allocate additional CPU resources.

Rule of Thumb:

- 1 CPU for every 500 concurrent active users.
- 1 CMS service for every 600 - 700 concurrent active users.

- or -

- 1 CPU for every estimated 100 user simultaneous requests. This number may greatly vary depending on the type of action made.

Example:

Question: What would be the estimate of how many CMS services and CPUs are required to support 4000 concurrent active users in a highly-active system?

- $4000 \text{ concurrent active users} / 500 \text{ concurrent active users per 1 CPU} = 8 \text{ CPUs}$
- $4000 \text{ concurrent active users} / 700 \text{ concurrent active users per 1 CMS service} = 5.71 = 6 \text{ CMS services}$

Answer: An estimate of 6 CMS services installed across 8 available CPUs.

This rule of thumb should be treated as an initial guideline since capacity numbers are highly dependent on other factors such as CPU speed, network, database connectivity, and so on.

Memory Requirements

For best performance, run each CMS cluster member on a machine that has the same amount of memory. Memory usage is controlled (approximately) by the number of objects stored in the object cache. This is controlled by the windows registry key called "MaximumObjectsToKeepInMemory." Specify the maximum number of objects that the CMS stores in its memory cache. Increasing the number of objects reduces the number of database calls required and greatly improves CMS performance. However, placing too many objects in memory may result in the CMS having too little memory remaining to process queries. The upper limit is 100,000 and the default setting is 10,000.

For CMS System Database memory requirements refer to [Step 4: System Database Tuning](#).

Crystal Reports Cache Server

The Cache Server stores report pages generated by the Page Server. By storing report pages in cache, the Page Server and/or database server does not need to be accessed each time the report is requested. The Cache Server is used for viewing Crystal reports only, when the viewer's preference is set to any viewer other than Interactive DHTML.

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

The threshold that determines the number of Cache Server services required is the "maximum simultaneous processing threads" for each Cache Server service. The number of "simultaneous processing threads" is equal to the number of [simultaneous requests](#). A guideline for the maximum simultaneous processing threads per Cache Server service is 400

- 400 "maximum simultaneous processing threads" per Cache Server service

Processor Requirements

One CPU is required for every estimated 200 "maximum simultaneous processing threads".

Example 1:

If an environment must service 200 simultaneous Cache Server requests, based on the 400 maximum for each Cache Server service, you would require only one Cache Server service. Also, based on the 200 "maximum simultaneous processing threads" for each processor, you would require a single processor.

- One CPU machine with one Cache Server service

Example 2:

If an environment must service 800 simultaneous Cache Server requests, based on the 400 maximum for each Cache Server service you would require at least two Cache Server services. Also, based on the 200 "maximum simultaneous processing threads" for each processor, you would require at least four available processors.

- One Quad machine with two Cache Server services

Memory Requirements

Estimate 1MB for each simultaneous processing thread plus an additional 17MB base.

Disk Requirements

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 temporary files (C:\Program Files\Business Objects\BusinessObjects Enterprise 11\Data\<machinename>.cacheserver\). The location of Cache Server cache files can be specified from the Central Management Console (CMC). The highest volume of cache files will likely be the "encapsulated page file" or ".epf." A single .epf file represents one page of a Crystal report and the size of the individual file is variable depending on the complexity of the report page (for example, 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 in a given period.

File Repository Servers (FRS)

There is an Input and an Output File Repository Server in every Enterprise XI implementation.

The Input File Repository Server manages objects (Crystal reports, OLAP Intelligence reports, Web Intelligence reports, Desktop Intelligence Reports, program objects, Microsoft Excel files, Microsoft Word files, Microsoft PowerPoint files, Adobe Acrobat PDFs, rich text format files, text files, hyperlinks and object packages) that have been published to the system by administrators or end users. These can be published with the Publishing Wizard, the Central Management Console, or the Import Wizard. They can also be published with a Business Objects designer component such as Crystal Reports or the Web Intelligence Java or HTML Report Panels.

The Output FRS maintains all the instances that have been produced from reports (Crystal, Web Intelligence or Desktop Intelligence), programs, and object packages that have been scheduled.

Repository location

You may have multiple FRS (input) and FRS (output) services on one or several different machines to support a high-availability environment, however the FRS services will behave in an active/passive manner where the first available FRS will be active and all other FRS services will remain passive unless the active FRS becomes unavailable.

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 Windows 2000 Server can be set to "Maximize Throughput for File Sharing". This will give a higher priority to file sharing applications.

Calculate Number of File Repository Servers (FRS) Required

One of each Input and Output File Repository Servers are required. In larger deployments, there may be multiple Input and Output File Repository Servers, for redundancy. In this case, all Input File Repository Servers must share the same directory. Likewise, all Output File Repository Servers must share a directory.

Processor Requirements

The File Repository Servers require higher I/O resources (faster disk, network) and fewer CPU resources. When estimating the number of CPUs in the Enterprise XI system, the File Repository Servers are not considered.

Memory Requirements

The File Repository Servers will have little impact on system memory.

Disk Requirements

Enough disk space must be available to store files. Typically, the Output FRS will require more disk space than the Input FRS. The Output FRS maintains all the instances (with Saved Data) that have been produced from reports (Crystal, Web Intelligence, or Desktop Intelligence), programs, and object packages that have been scheduled, and as such will require proportionately more disk space. For both the Input and Output FRS, the amount of space required will vary from system to system, however, knowing the average file size and multiplying this by the number of projected instances will assist in estimating total disk needs.

Search Server (for “Content Search” only)

Content Search is a new search feature that can be found in the XI Productivity Pack. It can be implemented in every Enterprise XI deployment.

Content Search is visible in the CMS as a service (Search Service) and is manageable via the CMC. In the CMC “Search” is an engine that is running in background for:

- Crawling and indexing the XI repository and file system automatically or on demand
- Generating and populating the search results using the search box

Content search is a new feature that completes the “regular” search, made on document title. It does not replace the “regular” search.

Particular settings and prerequisites

Search Server Max Heap Size

The Search Server Max Heap size must be increased from the default 512MB to 1024MB.

- Maximum Memory Pool = 1024

Processor Requirements

There are three profiles in this family of users:

Passive User

This profile is the simplest “power search” profile. It consists of a regular user that knows exactly what document he is looking for, but does not want to browse the categories and folders. He uses the Power Search capabilities to simplify the find:

- Maximum Passive user = 32 per CPU
- Optimal Passive user = 8 per CPU

Analyst User

This profile is the medium “power search” profile. It consists of a regular user that does not know which document she is looking for, but she does know that one document (or more) is present in the repository. She knows a word or a concept that will return these documents. She types these words and makes her request more focused with the “search server” specific features.

- Maximum Analyst user = 32 per CPU
- Optimal Passive user = 8 per CPU

Power User

This profile is the higher “power search” profile. It consists of a regular user that does not know which document he is looking for. He does not know if a document exists in the repository. He knows a word or a concept that he wants to include in a document. He wants to build a request with “power search” help. He types this word and makes his request more focused with the “search server” specific feature: create query.

- Maximum Power user = 32 per CPU
- Optimal Passive user = 16 per CPU

Memory Requirements

Depending on the design of a query, the memory requirements will vary but will not be more than the regular WIReportServer demanding resources.

Disk Requirements

Enough disk space must be available to store indexes. That is greatly dependent on the number of objects that can be indexed. Also, for faster processing, the directory that contains indexes should be installed on the fastest possible disks.

- Optimize disk performance on: C:\Program Files\Business Objects\BusinessObjects Enterprise 11.5\Data\search or the equivalent directory path on a UNIX system.

Event Server

The Event Server manages file-based events. When you set up a file-based event within Enterprise XI, the Event Server monitors the directory that you specified. When the appropriate file appears in the monitored directory, the Event Server triggers your file-based event. That is, the Event Server notifies the CMS that the file-based event has occurred. The CMS then starts any jobs that are dependent upon your file-based event.

Processor and Memory Requirements

The Event Server, under normal enterprise usage, is not a processing or memory intensive server and as such will not be weighted in the sizing process. If Event Server functionality is required, allocate this service into the system but do not estimate any additional CPUs for this service.

Processing Tier

Desktop Intelligence

The Desktop Intelligence Report Server and the Desktop Intelligence Cache Server (in conjunction with a Desktop Intelligence Job Scheduling Server) are collectively referred to as Desktop Intelligence. It handles documents of extension-type “rep”, corresponding to document types that were formerly known as Full-Client Documents.

Desktop Intelligence Report Servers

The Desktop Intelligence Report Server is used to view and analyze Desktop Intelligence documents (formerly known as Full-Client Documents).

Desktop Intelligence Cache Servers

The principal benefit of the Desktop Intelligence Cache Server is to minimize document processing. This is achieved by means of intelligent caching of documents, and by sharing cached documents between various users, in accordance with a preprogrammed logic. This logic takes into account such elements as user access rights, document access properties, caching parameters (such as the specified duration of validity of a cached document), whether a request to explicitly refresh a document should be met by cached date or in an immediate update from the database, and so on.

Desktop Intelligence Job Server

The Desktop Intelligence Job Server processes scheduling requests it receives from the CMS for Desktop Intelligence documents and generates the instance of the Desktop Intelligence document.

Processor Requirements

Both the Desktop Intelligence Report server and the Desktop Intelligence Cache server have to be considered for processing Desktop Intelligence documents.

Every user request involving a Desktop Intelligence document will be treated through the Desktop Intelligence cache server (if preference is set to HTML viewer). If the information is already located in the cache, the result will be returned back directly to the user without any further processing. This has been designed in order to improve the performances of the system.

If the information is not already in the cache, the request will be forwarded to the Desktop Intelligence Report server. For high reliability, the Desktop Intelligence Report server creates sub-processes on a per-document basis (instead of handling multiple documents in the same executable). The management of sub-processes is automatically done by the report server according to the users' requests and the compatibility with the currently allocated sub-processes. The maximum number of sub-processes can be adjusted through configuration settings, in order to avoid system saturation. Similarly, the way sub-processes can be recycled to process different documents can also be adjusted through configuration settings by specifying the inactivity time-out after which the sub-process can be released.

Depending on the size and complexity of the documents and the type of action being performed (view as opposed to refresh), processor requirement will vary, however; the following rule is recommended:

For sizing estimates for the **Desktop Intelligence Cache server**, it is recommended to plan for 50 maximum simultaneous user requests for each CPU. It will be used mainly for viewing requests.

- 1 Desktop Intelligence Cache Server service installed for each 200 – 400 simultaneous requests
- and -
- 1 CPU to support every 50 simultaneous user requests

For sizing estimates for the **Desktop Intelligence Report server**, it is recommended to use a range of 8 to 12 maximum simultaneous user request for each CPU. It will be used mainly for populating the cache and refreshing documents.

- 1 Desktop Intelligence Report Server service installed for each machine
- and -
- 1 CPU to support every 8 – 12 maximum simultaneous user requests

Memory Requirements

The amount of main memory required specifically for Desktop Intelligence will be a function of the number of Desktop Intelligence report users, the volume and size of the documents they use and the actions performed on the documents.

Some points to consider for the Desktop Intelligence Cache Server:

- The maximum value of the Desktop Intelligence Cache Server size can be set in the Central Management Console. By default it is 100 MB. This value sets the maximum amount of physical memory for the cache within the fccache process. However, the total size of the physical memory footprint of the Desktop Intelligence Cache Server process (in other words, the RSS value seen with the UNIX ps command) will be almost twice that amount.
- If the cache size set in the CMC is excessively large, the Desktop Intelligence Cache Server will continue to use the space until it reaches the maximum. It will then launch the cache clean-up mechanism. After this clean-up has finished, the cache size (and so naturally the process size) remains the same. In other words, the clean-up mechanism will not reduce the size of the memory footprint.

- If the cache size set in the CMC is too small, then the Desktop Intelligence Cache Server process will be forced to use temporary files of the disk as part of the cache. The resulting disk I/O will have a negative impact on performance.
- The "amount of cache to keep when document cache is full" (low water mark) parameter is calibrated for a good balance between maintaining enough copies of the documents in the cache and releasing enough space for new ones. However it can be tuned for specific needs if necessary.

For the Desktop Intelligence Report Server, the initial memory size is 22Mb. Sub-processes have an initial memory size of 70Mb; the size will grow according to the size and complexity of the processed document.

Performance Manager

Performance Manager (PM) uses nine server processes:

- AA Alert & Notification Server: alerting services
- **AA Analytics Server:** analytic processing, rendering, and transformation services
- **AA Dashboard Server:** dashboard management services
- AA Individual Profiler Server: individual analytic services
- AA Metric Aggregation server: metrics services
- AA Predictive Analytic Server: predictive analytic services
- AA Repository Management Server: AF repository services
- AA Set Analyzer Server: set analytic services
- AA Statistical Process Control Server: process control analytic services

The **AA Dashboard** and **AA Analytic** servers are the principal servers required for heavy loads. They can have multiple servers running at the same time, as they are enabled for load-balancing.

The other servers can also be deployed on several machines, however only one instance can be active at a time. The other instances function as backup.

While analytics can be based on universe queries, performance management analytics interact with metrics to render their information. Because they work with a much smaller amount of data and with significantly less-complex queries, they can run more quickly. In addition, analytics are managed through a process called AA Analytic, which is a multi-threaded object that can handle the processing required of hundreds of concurrent analytic requests.

Processor Requirements

Although AA Analytic and AA Dashboard servers may expand to several CPUs, the best throughput will be obtained by running one service (either AA Analytic or AA Dashboard) for each CPU.

For sizing estimates on AA Analytics, it is recommended to use a range of 18 to 40 Simultaneous Requests for each available CPU. This can be highly dependent on analytics complexity.

For sizing estimates on AA Dashboards, it is recommended to use a maximum of 40 Simultaneous Requests for each available CPU. This can be highly dependent on the number of analytics displayed on the dashboard and the type of action.

Example Configuration:

If there are an expected 160 concurrent active users accessing a performance management dashboard, and assuming conservatively 40 maximum simultaneous requests per processor, the potential configuration is:

- One QUAD machine with two AA Analytics and two AA Dashboard services where each service is configured to support 40 concurrent active users.

Memory Requirements

Depending on the design of a report, the types of actions being performed (viewing, modifying, refreshing), the memory requirements will vary.

A general guideline for sizing the AA Analytic server is 200Mb for each service.

A general guideline for sizing the AA Dashboard server is a base value of 120Mb for each service. In a manner similar to report processing elements, the AA Dashboard server memory footprint will be augmented each time an analytic is shown in the dashboard, so it is mostly dependent on the number and complexity of the analytics presented.

Scalability

The scalability of PM is dependent on both the power of the machine on which PM server processes are deployed and the power of the database machine, especially when the metrics use live data.

Web Intelligence Report Server

The Web Intelligence Report Server is used to create, edit, view, and analyze Web Intelligence documents (stored in the Input/Output FRS). It also processes scheduled Web Intelligence documents and generates new instances of the document, which it stores on the Output File Repository Server (OFRS). Depending on the user's access rights and the refresh options of the document, the Web Intelligence Report Server will use cached information, or it will refresh the data in the document and then cache the new information.

Processor Requirements

Although a Web Intelligence report server may expand to several CPUs, the best throughput will be obtained by running one Web Intelligence report server service for each CPU.

- Run one Web Intelligence Server service for every available CPU.

Example Configuration:

If there is an expected 100 concurrent active users viewing or modifying a Web Intelligence document, potential configuration is:

- One QUAD machine with 4 Web Intelligence Report Server services each.

Memory Requirements

Depending on the design of a report, the types of actions being performed (viewing, modifying, refreshing), the memory requirements will vary. A “refresh” request demands the greatest amount of memory for a Web Intelligence document as the database is queried and the entire dataset will be transferred to the Web Intelligence server.

NOTE	When using very large documents, it may be necessary to increase the number of Web Intelligence report server services to more than one for each CPU, in order to avoid reaching the 2GB user process address space limit.
-------------	--

Physical Address Extension support (PAE)

QA testing of Enterprise XI has used the /PAE switch to increase memory access. The /PAE switch changes the addressing mode to allow the operating system to access more than 4GB of RAM. By using the /PAE

switch each process is still limited to 2GB of user addressable space, but now the system can have more of these large process running at once.

Cache Clean

This kind of workflow corresponds to opening a Web Intelligence document from the cache and does not require a regeneration of the document. In other words, it does not contain non-cacheable data (“current time” or “current user” for example).

The performance and scalability in this case depends mainly on disk I/O performance when accessing the folder:

- Optimize disk performance on: C:\Program Files\Business Objects\BusinessObjects Enterprise 11.5\Data\<cms name>\storage or the equivalent directory path on a UNIX system.

Cache Dirty

This kind of workflow corresponds to opening a Web Intelligence document that requires a regeneration of the document without accessing the database. In other words, it contains some non-cacheable data.

The performance and scalability in this case depends on CPU power and memory size of the middle-tier.

The memory requirement can be calculated as follows:

- X = the initial memory size of the Web Intelligence Report Server process
- Y = the memory size of Web Intelligence Report Server after opening a typical Web Intelligence document
- Y – X = the memory size necessary to open a typical document
- $Z = 1.5 \text{ Gb} / (Y - X)$ = the number of typical documents that can be opened simultaneously by one Web Intelligence Report Server

The number of concurrent users is higher than Z as in reality there are many more concurrent users than the number of documents opened simultaneously.

Refresh

This kind of workflow corresponds to opening a Web Intelligence document and refreshing it. This workflow requires an access to the database in order to get the data and then regenerate the document.

The performance and scalability of this workflow depends on:

- The number of simultaneous connections that can be made to the database
- The database power (memory and CPU)
- The memory and CPU of the middle-tier (same calculation as for Cache Dirty workflows)

The performance of the database can be highly optimized if the database system is configured so it caches the SQL queries (to not recalculate the same query). This way, if a user refreshes a report for the first time then another user refreshes the same report, the Database system will not calculate the same query for the second user.

Web Intelligence Job Server

The Web Intelligence Job Server processes scheduling requests that it receives from the CMS for Web Intelligence documents. It forwards these requests to the Web Intelligence Report Server, which will generate the instance of the Web Intelligence document. The Web Intelligence Job Server does not actually generate object instances.

Processor Requirements

The Web Intelligence Job Server has a comparable function to the Crystal Reports Job Server in that it is responsible for handling Scheduled Jobs. However, the Web Intelligence Job Server does not actually “process” reports, but only acts as a scheduling manager or “router”, sending jobs to be processed by the Web Intelligence Report Server.

NOTE	One parameter to be aware of is -requestTimeout N. Where N is in milliseconds, the default is 600,000 and lowest allowed value is 30,000. If it is expected that reports will run longer than 10 minutes when scheduled, then this setting should be increased.
-------------	---

- One available CPU can optimally support five maximum jobs (processes) – higher or lower depending on report complexity and size.

Disk Requirements

For the Web Intelligence Job Server service, sufficient hard drive disk space should be available in the Temp directory for the creation of temporary files during report processing. The data from the database server is stored in these files until it can be saved and compressed in the report.

Hard drive access speed to the Temp directory may have an impact on the speed at which a report processes.

- Optimize disk performances on: C:\Program Files\Business Objects\BusinessObjects Enterprise 11.5\Data\procSched\

Crystal Reports Page Server

The Page Server is primarily responsible for responding to page requests by processing Crystal reports and generating Encapsulated Page Format (EPF) pages. A single .epf file represents one page of a Crystal 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 (encapsulated page files) pages are reused as frequently as possible, and new pages are generated as soon as they are required. The InfoView portal 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.

The Crystal Reports Page Server creates Page Server Sub Processes. Each Sub Process loads CRPE and then instantiates threads or Print Jobs as needed. With the Page Server, if an individual Print Job were to fail for any reason, only those threads contained in the Page Server Sub-Process would be affected. All other Sub Processes within the Page Server service would be unaffected. In addition, individual Sub Processes are shut down after so many requests and a new Sub Process is started, if required, so as to maximize resource management.

Definitions used in this section:

- **Page Server Service** – Service that manages sub-processes
- **Page Server Service Sub Process** – Process responsible for managing Report Jobs

- **Report Job** – Thread responsible for generating report pages requested by report viewers
- **Maximum Simultaneous Report Jobs** – The total number of Report Jobs that can be contained in a Page Server Service

For the Page Server service, the number of Page Server sub-processes and the total Maximum Simultaneous Report Jobs are determined by the following default algorithm:

Max Simultaneous Report Jobs (threads) = (# CPU's) * (25) [minimum of 50 on a single CPU]

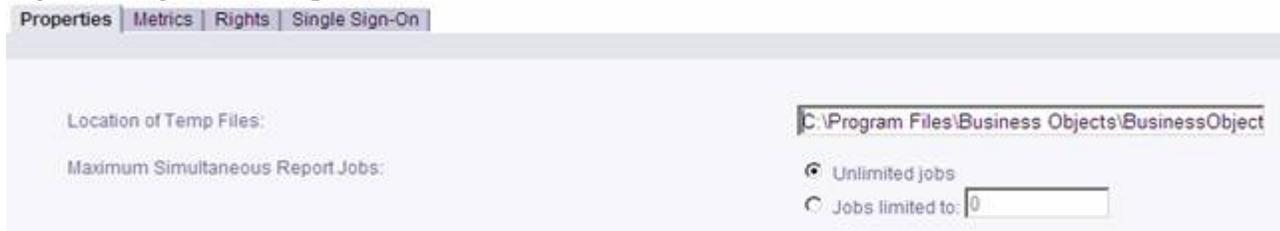
Max # of Sub Processes = (Max Simultaneous Report Jobs) / 10 report jobs per sub-process (rounded up)

Max # of Processes = (Max # of Sub Processes) + 1 Parent Process

(The maximum report jobs per sub-process is set at 10)

By default, the above algorithm is used to determine the Maximum Number of Simultaneous Report Jobs on a particular machine. The algorithm has been purposely tuned conservatively to favor reliability (lower number of Simultaneous Report Jobs for each CPU) so as to work optimally in most reporting environments and configurations. The default can easily be overridden in the CMC (under Page Server properties) by changing the radio button from **Unlimited jobs** to **Jobs limited to**.

Figure 3 - Page Server Properties



The **Jobs limited to:** option gives the administrator the ability to increase or decrease the Maximum Number of Simultaneous Report Jobs that can run on a single Page Server service (parent process). Based on internal testing, broad guidelines for this setting are outlined below within this section.

IMPORTANT	<p>In Enterprise XI only one Page Server service is required to run on a single machine. The Enterprise XI Page Server creates and stops new Page Server sub-processes on an as-needed basis. The Page Server detects the number of processors on the machine and will scale accordingly.</p> <p>For this reason, it is important to not install, and is not necessary to install more than one Page Server on a single machine if the default setting of Unlimited jobs is kept.</p>
------------------	--

Dedicated Page Server machine

The newly-designed Page Server improves performance, reliability and manageability particularly when a machine is fully dedicated to Page Server processing. Again, there is only need to install one Page Server and this Page Server will dynamically adapt to different loads (by creating and stopping Page Server sub-processes as needed) within its available resources. In the case of a dedicated Page Server machine, it is advisable to use the default setting of **Unlimited jobs** for **Maximum Simultaneous Report Jobs**.

When using the default setting of **Unlimited jobs**, the Crystal Reports Page Server's **Maximum Number of Simultaneous Report Jobs** (equal to the amount of user [simultaneous requests](#)) maximum will be calculated as 25 x Number of CPUs, with a minimum of 50 (25 by calculation, but minimum is always 50).

One CPU - Maximum Simultaneous Report Jobs = 50

Two CPUs - Maximum Simultaneous Report Jobs = 50

Four CPUs - Maximum Simultaneous Report Jobs = 100

Eight CPUs - Maximum Simultaneous Report Jobs = 200

Shared Page Server machine

There may be limited cases where it is advisable to change the default setting from unlimited so as to throttle back the maximum number of Page Server Jobs created on a single machine. This might be advisable in the case where the Page Server is sharing resources with other services (such as the CMS, Job Server, Cache Server, and so on) as would be the case where the complete Enterprise XI stack is installed on a stand alone machine.

The Page Server service can be changed from the default setting of **Unlimited jobs** to a recommended range of 25 - 75 **Maximum Simultaneous Report Jobs** for each available CPU. The setting of 25 - 75 can be adjusted higher or lower depending on the environment (like report complexity, size, and so on). Values below 25 for each CPU may be appropriate if other processes (such as the CMS, Job Server, Web Intelligence, and so on) are present on the machine and are sharing the same CPUs.

On-Demand (Live Data) compared to Saved Data Viewing (Pre-Scheduled Instance)

Live Data

On-demand reporting gives users real-time access to live data, straight from the database server. Use live data to keep users up-to-date on constantly changing data, so they can access information that's accurate to the second. For instance, if the managers of a large distribution center need to keep track of inventory shipped on a continual basis, then live reporting is the way to give them the information they need.

Before providing live data for all your reports however, consider whether or not you want all of your users accessing data from the database server on a continual basis. If the data isn't rapidly or constantly changing, then all those requests to the database do little more than increase network traffic and consume server resources. In such cases, you may prefer to schedule reports on a recurrent basis so that users can always view recent data (report instances) without connecting to the database server.

Saved Data

To reduce the amount of network traffic and the number of hits on your database servers, you can schedule reports to be run at specified times. When the report has been run, users can view that report instance as needed, without triggering additional connections to the database.

Report instances are useful for dealing with data that is not continually updated. When users navigate through report instances, and drill down for details on columns or charts, they do not access the database server directly; instead, they access the saved data. Consequently, reports with saved data not only minimize data transfer over the network, but also lighten the database server's workload.

For example, if your sales database is updated once a day, you can run the report on a similar schedule. Sales representatives then always have access to current sales data, but they are not accessing the database every time they open a report.

CPU utilization and memory consumption will be relatively comparable between Live Data viewing and Saved data viewing, however; viewing Saved Data reports on average will decrease viewing response times and increase throughput and system efficiency.

Crystal Page Server Data Sharing

The **Oldest On-Demand Data Given To a Client (in minutes)** setting controls how long the Page Server uses previously processed data to meet requests. If the Page Server receives a request that can be met using data that was generated to meet a previous request and the time elapsed since that data was generated is less than the value set here, then the Page Server will reuse this data to meet the subsequent request.

Reusing data in this way significantly improves system performance when multiple users need the same information. When setting the value of the oldest processed data given to a client, consider how important it is that your users receive up-to-date data. If it is very important that all users receive fresh data (perhaps because important data changes very frequently) you may need to disallow this kind of data reuse by setting the value to 0. The default is always set to 0 meaning that all users will, by default, receive fresh data.

If Data Sharing can be used in a system, this can decrease the number of CPUs required to process (view) a report.

When to use Crystal Reports Job Server vs. Crystal Reports Page Server

The Crystal Report Page Server is designed to process a large set of smaller reports whereas the Crystal Reports Job Server is designed to process a smaller set of very large reports. 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 (Live Data). Larger complex reports that must retrieve and process a very large set of data should be scheduled (Saved Data)

Processor Requirements

For sizing estimates based on number of simultaneous jobs for each CPU it is recommended to use a range starting from the default 25 to a recommended maximum of 75.

- 1 Processor = 25 - 75 Maximum Simultaneous Report Jobs

Memory Requirements

Depending on the design of a report and the number of records retrieved from the database, memory requirements may vary. When a report is viewed and loaded into memory the report is decompressed and expanded up to as much as 40 times the original report file size (with saved data/retrieved records).

Example (minimum memory requirements on Page Server machine):

- 500KB Report File Size (contains saved data) = 500KB * 40 (decompression ratio) = 20MB
- 25 Reports * 20MB = 500MB of minimum memory required

It is not recommended that the /3GB boot.ini switch be used to increase the size of the user process address space from 2 GB to 3GB (therefore reducing the size of the system space from 2GB to 1GB) on a Page Server machine. When the Page Server is rendering reports a high degree of GDI work (Graphical Device Interface, a Windows standard for representing graphical objects and transmitting them to output devices) is being performed and most GDI memory is allocated in the kernel address space, which is at a real premium with /3GB.

Physical Address Extension support (PAE)

QA testing of Enterprise XI included the use of the /PAE switch to increase memory access. The /PAE switch changes the addressing mode to allow the operating system to access more than 4GB of RAM. By using the /PAE switch each process is still limited to 2GB of user addressable space, but now the system can have more of these large process running at once.

For more advanced configuration/sizing estimates that may include Viewing and Scheduling concurrently see [Step 3: Configuration](#)

Disk Requirements

For the Crystal Reports Page Server service, sufficient hard drive disk space should be available in the Temp directory for the creation of temporary files during report processing (C:\Program Files\Business Objects\BusinessObjects Enterprise 11\Data\<machinename>.pageserver\). The data from the database server is stored in these files until it can be saved and compressed in the report.

Hard drive access speed to the temp directory may have an impact on the speed at which a report processes.

- Optimize disk performances on: C:\Program Files\Business Objects\BusinessObjects Enterprise 11.5\Data\procSched\<machinename>.pageserver or the equivalent directory path on a UNIX system.

Crystal Reports Job Server

The Crystal Reports Job Server processes report files (.rpt) as requested by the 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.

The Job Server has the ability to process both Report files and Packaged objects. Object packages are simply collections of Enterprise XI objects (such as report files) that can be grouped together and managed by the Enterprise XI 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. A single Job Server service with one available CPU can optimally support five maximum jobs (processes). Also, a single Job Server service can be used to support up to 20 jobs across four CPUs. The default settings for maximum jobs will vary based on the complexity and size of reports, but the conservative setting of five simultaneous jobs per available CPU should handle any kind of report.

Processor Requirements

- One available CPU can optimally support five maximum jobs (processes) – higher or lower depending on report complexity and size

For schedule batch reporting, the number of CPUs required to support the Crystal Reports Job Server service is dependent on the number of concurrent jobs processing, the complexity and size of reports, and the time window available to execute reports.

$$\frac{(\# \text{ of reports}) \times (\text{Average Report Process Time})}{(\text{Time Window}) \times (\# \text{ of simultaneous jobs per CPU})} = \text{number of required CPUs}$$

Example:

How many CPUs are required to process 1000 reports taking 10 minutes each on average, that must be processed in a four-hour (240 min) time window?

$$\frac{(1000 \text{ reports}) \times (10 \text{ minutes})}{(240 \text{ minutes}) \times (5 \text{ simultaneous jobs per CPU})} = 8.4 \text{ CPUs (round up to nine CPUs, such as two Quads and one Single)}$$

Memory Requirements

Depending on the design of a report and the number of records retrieved from the database, memory requirements may vary. When a report is executing and loaded into memory the report is decompressed and expanded up to as much as 40 times the original report file size (with saved data/retrieved records).

Example (minimum memory requirements on Crystal Report Job Server machine):

- 2MB Report File Size (contains saved data) = 2MB * 40 (decompression ratio) = 80MB
- 5 Reports * 80MB = 400MB minimum memory required

It is not recommended that the /3GB boot.ini switch be used to increase the size of the user process address space from 2 GB to 3GB (reducing the size of the system space from 2GB to 1GB) on a Crystal Report Job Server machine. When the Crystal Report Job Server is rendering reports, a high degree of GDI work (Graphical Device Interface, a Windows standard for representing graphical objects and transmitting them to output devices) is being performed and most GDI memory is allocated in the kernel address space, which is at a real premium with /3GB.

Physical Address Extension support (PAE)

QA testing of Enterprise XI included the use of the /PAE switch to increase memory access. The /PAE switch changes the addressing mode to allow the operating system to access more than 4GB of RAM. By using the /PAE switch each process is still limited to 2GB of user addressable space, but now the system can have more of these large process running at once.

Disk Requirements

For the Crystal Reports Job Server service, sufficient hard drive disk space should be available in the Temp directory for the creation of temporary files during report processing (C:\Program Files\Business Objects\BusinessObjects Enterprise 11\Data\procSched\

Hard drive access speed to the temp directory may have an impact on the speed at which a report processes.

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, RAS uses an internal caching mechanism that involves no interaction with the Cache Server. Specifically, RAS processes reports that InfoView users view with the Interactive DHTML viewer. RAS also provides the reporting capabilities that allow InfoView users to create and modify Crystal reports over the web. Additionally, Report Application Server is used at the time of viewing or submitting schedule requests for reports containing dynamic prompts and cascading lists of values.

The default **Maximum Simultaneous Report Jobs** is set at 75 for each RAS service. This value, as with the Crystal Reports Page Server, may be adjusted according to anticipated load and available hardware resources. The guideline of 25 to 75 (default) Simultaneous Report Jobs per CPU is recommended. The ideal setting for your reporting environment, however, is highly dependent upon your hardware configuration, your database software, and your reporting requirements. Additionally, a recommended guideline is to run one Report Application Server per CPU.

Example:

- For each available CPU run one RAS service with a setting of 25 to 75 Maximum Simultaneous Report Jobs (use 25 for Optimal Performance).

Processor Requirements

- One processor = 25 to 75 Maximum Simultaneous Processing Jobs (use 25 for Optimal Performance)

Memory Requirements

Depending on the design of a report and the number of records retrieved from the database, memory requirements may vary. When a report is viewed and loaded into memory, the report is decompressed and expanded up to as much as 40 times the original report file size (with saved data/retrieved records).

Example (minimum memory requirements on RAS machine):

- 500KB Report File Size (contains saved data) = 500KB * 40 (decompression ratio) = 20MB
- 25 Reports * 20MB = 500MB minimum memory required

It is not recommended that the /3GB boot.ini switch be used to increase the size of the user process address space from 2 GB to 3GB (reducing the size of the system space from 2GB to 1GB) on a RAS machine. When RAS is rendering reports a high degree of GDI work (Graphical Device Interface, a Windows standard for representing graphical objects and transmitting them to output devices) is being performed and most GDI memory is allocated in the kernel address space, which is at a real premium with /3GB.

Physical Address Extension support (PAE)

QA testing of Enterprise XI included the use of the /PAE switch to increase memory access. The /PAE switch changes the addressing mode to allow the O/S to access more than 4GB of RAM. By using the /PAE switch each process is still limited to 2GB of user addressable space, but now the system can have more of these large process running at once.

List of Values Job Server

The List of Values (LOV) Job Server processes scheduled list of values objects. These are objects that contain the values of specific fields in a Business View. Lists of values are used to implement dynamic prompts and cascading lists of values within Crystal Reports. List of values objects do not appear in CMC or InfoView. For more information, see the [Business Views Administrator's Guide](#).

The List of Values Job Server behaves similarly to the Report Job Server in that it retrieves the scheduled objects from the Input File Repository Server (FRS) and saves the instance it generates to the Output FRS. There is never more than one instance of a list of values object. On-demand list of value objects are processed by the Report Application Server.

Processor Requirements

The number of CPUs required to support the LOV service is dependent on the number of concurrent jobs processing. By default, the **Maximum Jobs Allowed** option is set to 5. For most environments this setting will not have to be changed and a single CPU is sufficient.

Memory Requirements

These LOV objects are "Group by" Crystal Reports. The file size of the LOV objects (stored in the output FRS) can be used to calculate memory requirements. When a report is processed and loaded into memory

the report is decompressed and expanded up to as much as 40 times the original report file size (with saved data/retrieved records).

Example (minimum memory requirements LOV machine):

- 100KB Report File Size (contains saved data) = 100KB * 40 (decompression ratio) = 4MB
- 5 LOV Reports * 4MB = 20MB minimum memory required

Connection Server

Sizing information for the Connection Server is currently being collected.

Application Tier

Web Application Server

Depending on how the system is being utilized, the Web Application Server (IIS, Apache Tomcat, BEA WebLogic, IBM WebSphere) can manage differing numbers of concurrent user sessions and simultaneous requests. The main functions of the Web Application Server within the Enterprise XI system are:

1. Processing the .NET/Java 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
5. Facilitating OLAP Intelligence view requests
6. Communicating with Web Intelligence Report Server for view requests

Processor Requirements

NOTE	Processor Requirements guidance is generalized and relative requirements may change based on the individual characteristics of different Web Application Server types. To better understand specific Web Application Server characteristics, please consult with the specific vendor.
-------------	---

Based on internal performance testing, one Web Application Server can manage approximately 400 concurrent user sessions (user session = one logged on user) per processor. Generally, a service can efficiently manage 100 simultaneous requests (request = a user clicking on a folder, for example). 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 with the following examples:

Example 1:

- One single available processor (one processor) with one Web Application Server service running could efficiently service 400 concurrent user sessions and can handle 100 simultaneous user requests.

Example 2:

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

Example 3:

- One Quad processor machine (four processors) with four Web Application Server services running could efficiently service 1600 concurrent user sessions and can handle 400 simultaneous user requests.

Rule of Thumb:

- One Web Application Server = 400 concurrent user sessions (user session = one logged on user).
- One Web Application Server = 100 simultaneous requests per processor (request = a user clicking on a folder, for example)

As a base guideline, it is recommended to estimate one Web Application Server per 100 simultaneous requests, however each viewer type will have its own characteristics that will impact the capacity for concurrent users or simultaneous requests and as such this number can be higher or lower.

Viewing in ActiveX or Java Viewers

- One Web Application Server = 100 simultaneous requests per processor (request = a user interacting with the report, for example).

Viewing in HTML Viewer (Crystal report, Web Intelligence report, Desktop Intelligence report)

- One Web Application Server = 50 simultaneous requests per processor (request = a user interacting with the report, for example).

OLAP Intelligence functionality within the Application Tier

OLAP Intelligence allows for native access and analysis of OLAP servers from Microsoft, Hyperion, IBM, and SAP. It lets you slice, dice, and drill directly on multidimensional sources and deliver interactive workbooks with prebuilt guided analysis and workflows via the Web, Windows, and Microsoft Excel. It is implemented through the Web Component Adapter (WCA) that in turn invokes OLAP intelligence processing (OLAPSESSIONS.EXE).

Processor Requirements

When the OLAP Intelligence DHTML viewer is being used for viewing reports, the Web Application Server (WAS) can manage a maximum of 5 - 8 simultaneous user requests per processor which equals approximately 40 – 60 concurrent active OLAP users.

However, in benchmark testing, the optimal number will be five simultaneous user requests per processor. A factor that can affect this number is the “complexity” of the reports or how users utilize and interact with 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 WAS can increase.

For OLAP Intelligence viewing, it is recommended that you run one Web Application service for each five simultaneous user requests if using DHTML on a single processor machine.

Viewing in DHTML Viewer (.NET InfoView)

- Maximum = 60 concurrent users = 8 simultaneous user requests = 1 Web Application Server (WAS) service = 1 Processor
- Optimal = 40 concurrent users = 5 simultaneous user requests = 1 Web Application Server (WAS) service = 1 Processor

Scaling up the number of CPUs on a single WAS:

Configuration:

- WAS: See the table below
- OLAP Server (data source): dedicated dual CPU machine

Type of CPU(s)	# CPUs on the WAS	CPU Speed (GHz)	(CPU speed) multiplied by (# CPUs)	Amount of RAM on WAS (GB)	Optimal Simultaneous Processing Threads	Optimal # concurrent users
Intel Pentium 4	1 (single)	1.7	1.7	1	5	40
Intel Xeon	2 (dual)	3.06	6.12	2	15	120
Intel Xeon	4 (quad)	1.4	5.6	3.48	Just less than 15	Just less than 120

OLAP Intelligence and scaling out (multiple machines) the number of WAS

In addition to scaling up the number of processors on your WAS, OLAP Intelligence can also scale out to several physical WAS machines to support clustering (also known as web farms). To achieve clustering, OLAP Intelligence can be installed on multiple WAS machines. During the OLAP Intelligence install, select the **Web Components** option on the WAS machines. When logging on to Enterprise XI, specify the centralized CMS server to log on against, such that all OLAP Intelligence reports stored in Enterprise will be accessible on all WAS machines.

The same calculations and data as in the above table is used when determining the number of WAS machines to scale horizontally (multiple machines).

Memory Requirements

OLAPSESSIONS.EXE (OLAPWorker.exe) Memory Usage

Majority of memory used by OLAPSESSIONS.EXE (specific to .NET) and OLAPWorker.exe (specific to Java) are allocated when loading a report. The amount of memory allocated when loading a report depends on the following factors:

- **Size of the Report** - The bigger the report in terms of pages, the more memory is allocated.
- **Complexity of Report Pages** - The more complex the content of a report, the more memory is allocated.
- **Number of Concurrent Sessions** - When viewing reports, each OLAPSESSIONS.EXE serves up to 10 concurrent sessions. Each additional session served requires additional memory allocation and the amount depends on the report requested. Except the first of the 10 sessions, the memory allocated for each additional session is approximately the same.

Concurrent users	Simple 1 page (KB)	Complex 1 page (KB)	Complex 15 pages (KB)
1	19,616	20,552	21,720
2	25,288 (+5,671)	26,376 (+5,824)	28,252 (+6,532)
3	30,260 (+4,972)	32,036 (+5,660)	35,736 (+7,484)
4	35,156 (+4,896)	37,552 (+5,516)	42,344 (+6,508)
5	40,148 (+4,992)	43,216 (+5,664)	49,008 (+6,664)
6	45,144 (+4,996)	48,984 (+5,768)	55,472 (+6,464)
7	50,148 (+5,004)	54,676 (+5,692)	62,020 (+6,548)
8	55,088 (+4,940)	60,356 (+5,680)	68,780 (+6,760)
9	60,040 (+4,952)	65,948 (+5,592)	75,284 (+6,504)
10	64,728 (+4,688)	72,612 (+6,664)	81,816 (+6,532)
11*	19,600	20,568	21,696

* The 11th concurrent user is served by a new OLAPSESSIONS.EXE, separated from the one serving users 1 to 10.

Temp Space Consumption

OLAP Intelligence makes minimal use of temporary space on the file system. Temporary files are not used when opening OLAP Intelligence reports. Temporary .car files are created under the C:\windows\temp directory when either the **Save View** or **Save As** operations are performed from the toolbar within the OLAP Intelligence DHTML Viewer. These files are cleaned up right after the operation has been completed.

Compression

Temporary files are not created during report processing. None of the data from the OLAP Server is stored within the report (.car) files and as such there is no need for compression.

Business Objects Voyager OLAP server

Voyager allows for native access and analysis of OLAP servers from Microsoft (including Yukon cubes), Hyperion, IBM, and SAP. It is implemented in a backend tier (MDAS.exe) and in the application server tier.

Processor Requirements

Users are differentiated here between “consumer” users, and “power” users:

- **Power user** - Does in-depth OLAP analysis, drills, sorts, drag-and-drop, performs calculations, modifies the chart, possibly opens multiple connections, uses highlights and exceptions
- **Consumer user** - Displays reports created by a power user, does simple operations (sorts) and prints

The impact on processor requirements being very different, it is required to differentiate these 2 communities of users.

Heavy users

Heavy users use the system continuously, with no think time. This is the extreme case. Benchmarks have shown that consumer users uses the CPU more intensively than power users, since power users’ load is concentrated on the OLAP cube. The recommendation is:

- Heavy power users: 8 per CPU
- Heavy consumer users: 4 per CPU

Other users

If we add think time in the scenario, the capacity of Voyager will be extended considerably, especially considering that the OLAP analysis requires many operations and much thinking around these. The following calculation, based on internal benchmarking can be done to give an estimate of the system capacity:

- One heavy power users does 41 operations in two minutes. If we suppose that the user reasonably takes five minutes in average to do this workflow, there will be $300 - 129 = 171$ seconds spent as think time. That means a concurrent user will utilize really 43% (129 out of 300) of the full service time. The capacity of the system could be extended to 57%. We might be able to extrapolate the optimum load to $16 / 0.57 = 28$ users.
- Optimal = 28 concurrent users = One MDAS service = One processor

Vertical scalability

The best ratio that is observed when doubling the number of CPUs is 1.5. Thus when adding more and more CPUs, the asymptotic ratio will be three. It is not advised to deploy Voyager on systems with many CPUs.

Horizontal scalability

The best ratio that is observed when doubling the number of systems is 1.7, which is much better than the vertical scalability and should then be the preferred solution.

When disk write is required by an OLAP middleware driver (for instance Essbase) rather than having a powerful server with many CPUs, it is recommended to use multiple clustered machines with fewer CPUs in each.

Tier balance

The measured balance in our test cases is 70% (for back-end MDAS) and 30% (for web application tier), when the two are on the same machine. This can be used as a guideline to define the number of MDAS servers.

For instance, on an eight CPU server where the CMS and Application server are activated, we would activate five MDAS (served by five CPUs) and leave the three CPUs (30% of eight CPUs) to the CMS and Tomcat

Memory Requirements

As the MDAS' memory management is based on JVM Garbage Collection, the real memory usage cannot be linearly approximated by the number of concurrent users. The query size and the type of OLAP middleware (Microsoft Yukon, Essbase, and so on) have a great impact on the memory footprint in the JVM.

Each MDAS instance requires from 250MB to 1GB (peak value).

- Minimum: 512MB per MDAS
- Optimal: 1GB RAM available per MDAS

Disk Requirements

When using Essbase as the OLAP server, temporary files are created by the Essbase driver in Windows temporary folder (C:\windows\temp).

The disk contention, essentially in write operations, may occur in the following conditions:

- High user concurrency
- CPU power - the more powerful the CPUs (speed and number), the more the disks are loaded
- High number of MDAS instances
- Important query size: the temporary files contain data retrieved from OLAP server

Recommendations:

- Deploy MDAS instances in separate clusters
- Limit the number of MDAS instances for each machine
- Maximize disk performance by using the operating system's available features

Query as a Web Service (QaaWS) special settings

Query as a Web Service is a desktop product that can be found in the XI Productivity Pack. It can be implemented in every Enterprise XI deployment.

There is no particular process for QaaWS in the CMS and this service is not manageable via the CMC.

Queries are created inside the desktop product with tools like Xcelsius and are pushed inside the application server as web queries.

These web queries are directly available for end users.

Particular settings

WIReportServer

As WIReportServer is the process used repeatedly by QaaWS on an Enterprise XI deployment, the **Maximum Simultaneous Connections** for the process should be set from the default "50" to the desired number of users.

Tomcat

- Tomcat Connection Timeout should be disabled - set from the default of "20000" to "0"
- Tomcat Max Threads should be increased from "150" to "600"

Processor Requirements

For sizing estimates based on the number of simultaneous jobs for each CPU, it is recommended to use no more than 300 Maximum Simultaneous Connections for each available CPU (this setting can be highly dependent on query complexity). Although a Web Intelligence report server may expand to several CPUs, the best throughput will be obtained by running one Web Intelligence report server service for each CPU.

Maximum:

One Web Intelligence Server service with one available CPU can support up to 300 Maximum Simultaneous Connections.

- 300 concurrent users = one WIReportServer = one processor

Optimal:

One Web Intelligence Server service with one available CPU, optimally serves between 64 and 128 simultaneous connections.

- 64 - 128 concurrent users = one WIReportServer = one processor

NOTE	Default Web Intelligence Server service "Maximum Simultaneous Connections" will need to be adjusted to equal or more than the maximum desired number of simultaneous connections.
-------------	---

Memory Requirements

Depending on the design of a query, the memory requirements will vary but will not be more than the regular WIReportServer demanding resources.

Live Office special settings

Depending on the usage that is made by Live Office, the settings of the given services should be applied as described in the previous sections. For instance:

- If Live-Office is used with Crystal Reports, follow the section on RAS.
- If Live-Office is used with WebIntelligence, follow the WIReportServer tuning.
- If Live-Office is used as a query panel, pay attention to the WIReportServer tuning, on the refresh aspect, (as the query panel is similar to that of WebIntelligence, when the refresh of a document is requested).

Recommendations

- Add one ReportApplicationServer instance for every 50 to 75 Live Office Crystal users.
- The ReportApplicationServer connection timeout should be optimally adjusted in order to not keep unused sessions left for too long a time. When the timeout value is too large, the maximum connection may be overflowed. The timeout value, however, should be long enough to prevent active sessions from being interrupted.
- In a cluster environment, ReportApplicationServer and WIReportServer should be deployed on separate servers in order to reduce the concurrency on disk I/O.
- The WIReportServer maximum simultaneous connections value can be increased from 300 to 600 to support a large number of Live Office Web Intelligence users. Rather than multiplying the WIReportServer instances and keeping a smaller value, setting a larger number of connections in one instance is more resource-efficient and reduces disk contention.
- The number of QueryPanel users should be restricted in order to avoid too much database access.
- In a Live Office document, it is more efficient to use a filter rather than an interactive prompt (less interactions with the database).
- As a rule of thumb, use as many or more CMS instances than Report Application Server instances for Live Office Crystal users. This implies better user session handling.
- In a cluster environment, Report Application Servers deployed on the primary node give better performance than those deployed on secondary nodes. Secondary nodes can be used for WiReportServers without performance issues.
- When deploying over several systems of several CPUs, horizontal scalability is more efficient than vertical scalability..
- Faster hard disks on the Live Office server is better for performance and scalability. For example, SCSI, striped formatting disk, SAN and so on.

Step 3: Configuration and Infrastructure Characteristics

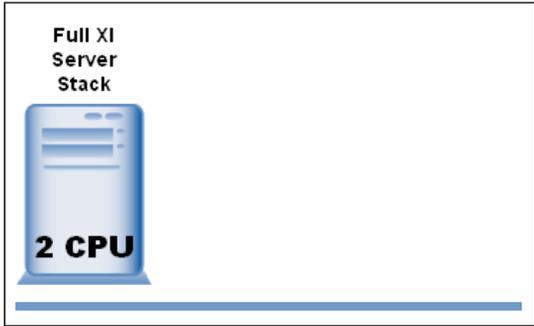
Step 3 deals with configuration choices and core characteristics of the infrastructure pertaining to disk I/O and Network Bandwidth and Latency. Content in this document outlines the most basic configuration concepts and helps to expand on the suggested configurations found in the Enterprise XI Administrator's Guide. For more specific configuration and tuning information, refer to the section called "Scaling Your System/General scalability considerations" in the [Enterprise XI Administrator's Guide](#).

NOTE	The following are conceptual illustrations of basic configuration types and are only examples. The number of CPUs on a box is not limited to a specific number of CPUs. The distribution of Enterprise XI Services is highly flexible and can be adjusted to suit any environment.
-------------	--

Basic One Machine Configuration (Vertical Scaling)

The most basic configuration separates the BusinessObjects Enterprise servers from the rest of your reporting environment and from your web server, and installs all BusinessObjects Enterprise servers on a single machine. This grants the BusinessObjects Enterprise servers their own set of processing resources, which they do not have to share with database and web server processes.

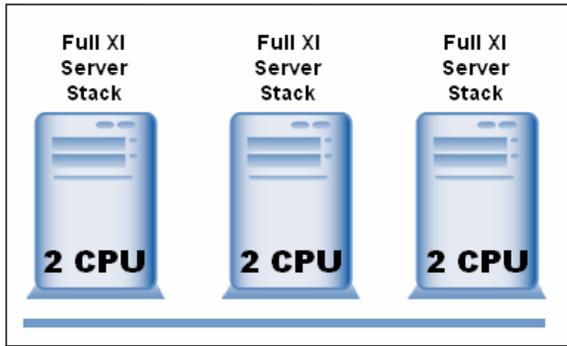
Figure 4 – One machine Enterprise XI configuration



Multiple Machine Configurations Types (Horizontal Scaling)

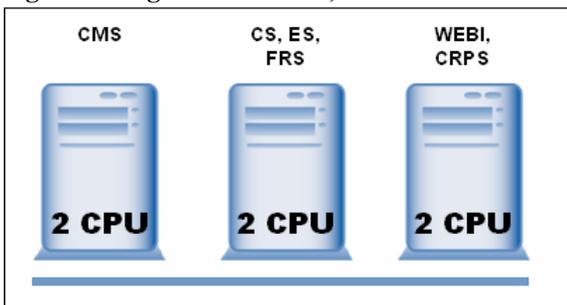
High Redundancy and Fault Tolerance - Configured for software and hardware redundancy where the complete Enterprise XI stack of servers is installed on each of multiple machines (such as a Datacenter Configuration).

Figure 5 - High Redundancy and Fault Tolerance



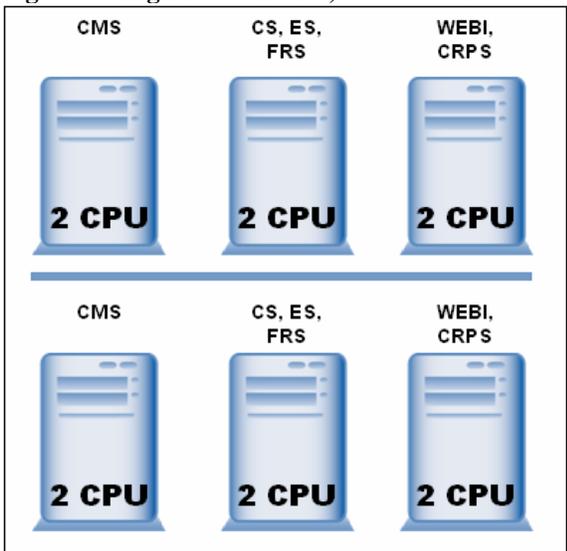
High Performance, Software Redundancy and Fault-Tolerance - Configured so that the environment divides the BusinessObjects Enterprise processing load in a logical manner, based on the types of work performed by each server. This is the most common multiple machine configuration.

Figure 6 - High Performance, Software Redundancy and Fault-Tolerance



High Performance, Software/Hardware Redundancy and Fault-Tolerance - Configured so that the environment divides the BusinessObjects Enterprise processing load in a logical manner, based on the types of work performed by each server, but then duplicates each of these machines to provide for redundancy and fault-tolerance.

Figure 7 - High Performance, Software/Hardware Redundancy and Fault-Tolerance



Disk I/O and Network Bandwidth and Latency

This section will be updated in a future release of this guide

Step 4: System Database Tuning

System Database Backend Performance

To provide optimal overall performance, the network and the database system need to be adequately sized and configured. Tuning a database system for peak performance is a complex task and involves hundreds of database parameters. Different tuning methodologies have been developed and comprehensive performance tuning guides are available.

Connection between Enterprise XI and the database system

For data storage and retrieval, Enterprise XI exchanges messages with the database system over the network. Enterprise's data caches ensure minimal network traffic between Enterprise XI and the database system. Nevertheless, good network performance is critical. For example, some operations cannot use Enterprise XI's data caches.

The database client software uses the underlying network system. There is no tuning required for the database client software.

NOTE	Make sure that network response times are adequate and network throughput is sufficient.
-------------	--

Performance Criteria

Meeting the following five criteria helps to avoid the most common performance bottlenecks.

- The database system's cache hit rates are over 90%.
- The optimizer statistics are not older than 24 hours.
- Lock granularity is row locking.
- There are no lock escalations.
- There are no log write waits (the average disk write queue length to the disk drives which contain the database log files is smaller than five).

90% or higher cache hit rates

Give us:

- relatively low number of physical disk reads and writes
- relatively low number of SQL compiler executions

Are important to avoid:

- the disk subsystem is unable to keep up with the level of I/O that the database server requests
- long disk read or disk write queues
- unnecessary SQL statement compilations or long SQL statement compilation times

Are achieved by:

- providing sufficient physical memory.
- configuring sufficient cache sizes.

Up-to-date optimizer statistics

Give us:

- The Query Optimizer chooses the Query Execution Plan with the minimal cost.
- Good selectivity estimates of predicates in SQL expressions specially for expressions including columns with unequal or skewed distribution of column values.

Are important to avoid:

- Long response times and low throughput
- Long execution times for queries, especially for queries predicate has high selectivity
- Full table scans
- Lock escalations
- Lock waits or deadlocks
- Significant differences between the estimated number of rows for each operator in the query plan from the actual number of rows

Are achieved by:

- implementing a procedure to periodically update the statistics or to update the statistics after frequent changes to the database.
- creating all statistics (table, column, index) and the statistics are not stale (underlying object has not been modified significantly ... more than 10% of the rows).
- setting the sample size (percentage of data that got analyzed to gather statistical information) to a value which is sufficient for the skewed data distribution.
- setting the number of histogram buckets to a value which is sufficient for the skewed data distribution.

Lock granularity is row locking

Gives us:

- increased concurrent execution of transactions.
- reduced transaction processing time due to less lock waits.

Is important to avoid:

- Lock waits or deadlocks

Is achieved by:

- setting the lock granularity of your database to row level.

No lock escalations

Give us:

- increased concurrent execution of transactions.

Is important to avoid:

- lock waits or deadlocks

Are achieved by:

- configuration of a sufficient size of the lock list.
- setting the lock escalation threshold to an appropriate value.
- keeping the optimizer statistics up-to-date.

No log write waits

Give us:

- increased concurrent execution of transactions.
- reduced transaction processing time due to no I/O waits.

Is important to avoid:

- lock waits or deadlocks

Is achieved by:

- providing an I/O subsystem with sufficient throughput.
- providing a Disk Controller Cache and setting the Disk Controller cache to 100% write cache.

Calculating the Database File Size

Estimate

Each row stores one Enterprise XI object. The average size of an Enterprise XI object is 1024 Bytes.

$$\begin{aligned} & (\text{Size Data} + \text{Size Indexes}) * (1 + (1 - \text{page fill factor})) \\ & = ((\text{Number Rows} * 1024) + \text{Number Indexes} * (32 * \text{number of rows} * 2)) * 1.3 \\ & = 1.3 * \text{number rows} * (1024 + \text{number indexes} * 64) \\ & = 1.3 * \text{number rows} * 2624 \end{aligned}$$

Number of Objects	Database Size	Unit
10,000	33,312	KB
100,000	333,125	KB
1,000,000	3,331,250	KB
10,000,000	33,312,500	KB

Sybase (for each one job one data page is allocated)

$$\begin{aligned} & = ((\text{number rows} * 844) + (\text{number rows} * \text{page size}) + \text{number indexes} * (32 * \text{number of rows} * 2)) * 1.3 \\ & = 1.3 * \text{number rows} * (2624 + \text{page size}) \end{aligned}$$

Number of Objects	Database Size (page size 4K)	Unit
10,000	87,360	KB
100,000	873,600	KB
1,000,000	8,736,000	KB
10,000,000	87,360,000	KB

Useful Resources

- IBM DB2 8.1 Manuals “Administration Guide: Performance”
- IBM Redbook “DB2 UDB ESE V8 non-DPF Performance Guide for High Performance OLTP and BI”
- Oracle Manual “Performance Tuning Guide and Reference”
- Oracle Manual “Supplied PL/SQL Packages and Types Reference”

Finding More Information

For more information and resources, refer to the product documentation and visit the support area of the web site at: <http://www.businessobjects.com/>

► www.businessobjects.com

© 2007 Business Objects. All rights reserved. Business Objects owns the following U.S. patents, which may cover products that are offered and licensed by Business Objects: 5,555,403; 6,247,008; 6,289,352; 6,490,593; 6,578,027; 6,768,986; 6,772,409; 6,831,668; 6,882,998; 7,139,766; 7,181,435; 7,181,440 and 7,194,465. Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Xcelsius, Crystal Decisions, Intelligent Question, Desktop Intelligence, Crystal Enterprise, Crystal Analysis, Web Intelligence, RapidMarts, and BusinessQuery are trademarks or registered trademarks of Business Objects in the United States and/or other countries. All other names mentioned herein may be trademarks of their respective owners.