crystal decisions™

Crystal Enterprise 8.5 Firewall Support

Understanding Crystal Enterprise and Firewall Integration

Overview

This document will provide you with an understanding of how to integrate Crystal Enterprise (CE) version 8.5 into a secure network environment.

Topics covered include:

- Supported firewalls
- Overview of CE's architecture from a firewall administrator's perspective.
- Details on firewall sensitive CE components and instructions on how to configure them.
- Several common firewall network configurations, samples and references.

You should have a good understanding of firewalls and CE prior to reviewing this document.

Contents

Introduction

From the outset, CE has been designed to provide pertinent information via a number of services to the web client. Due to the web client's connectivity to the Internet, there is great interest by network administrators to protect the resources to which web applications provide access. In a CE scenario, these resources are commonly corporate databases that consist of sensitive information. As a result, there is a need for firewalls and their related mechanisms to protect these databases. In many cases the network security and related expectations have already been put in place, thus providing rigid application requirements. CE caters to these requirements by providing components specifically designed for the security savvy network administrator while at the same time providing architecture such that these requirements do not inhibit the power and performance of a CE system. To explain CE's approach to network security this document will discuss the following topics:

- Firewalls: Common firewalls and their related mechanisms, including which firewalls have been tested with CE.
- TCP/IP: How CE uses TCP/IP for communication and what you need to know to accommodate this communication.
- CE architecture and network configuration: An overview of the CE architecture and how it integrates with firewalls, followed by a break down of the security sensitive components. Finally, it will walk through how to set up the CE components and the concerns of which to be aware.

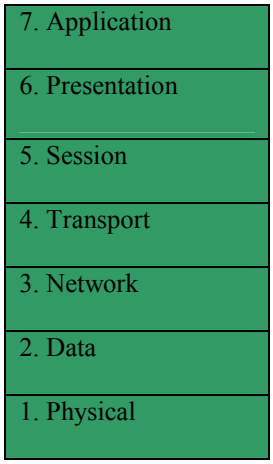
Firewall Overview

The following section will provide a brief overview of the architecture surrounding firewalls.

Network Architecture Layers

To understand how firewalls work it helps to understand how the different layers of a network interact. Network architecture is designed around a seven-layer model. Each layer has its own set of responsibilities, and handles them in a well-defined manner.

The OSI Model



Firewalls operate at different layers to use different criteria to restrict traffic. The lowest layer at which a firewall can work is layer 3. In the OSI Model, this is the Network layer. This layer is concerned with routing packets to their destination. At this layer, a firewall can determine whether a packet is from a trusted source, but cannot be concerned with what it contains or with what other packets it is associated. Firewalls that operate at the Transport layer know a little more about a packet, and are able to grant or deny access depending on more sophisticated criteria. At the Application layer, firewalls know a great deal about what is going on and can be very selective in granting access.

NOTE	There is another common network stack model called the TCP/IP Model. It breaks down the functionality in different layers but ultimately provides the same functionality as mentioned above.

Firewalls

The following section will provide a brief overview of the types of firewalls CE has been designed work with and the specific firewalls that CE has been tested with.

Types of Firewalls

Packet Filter (IP Filter) Firewalls

Packet Filter type firewalls operate on layer 3; they restrict network traffic based on the IP address and port number of each packet.

Stateful Inspection Firewalls

Some firewalls use a more sophisticated IP filtering technique called Stateful Inspection. Check Point FireWall-1 is an example of this type of firewall. Operating on layers 3 through 7, this is where the connection is remembered and packets are allowed to travel back and forth for this connection. The reason for this enhancement is to cater to the behavior that most TCP/IP applications exhibit. This is the common request/reply model where a client initiates a request to a server and the server replies back on the same connection.

NOTE	For the remainder of this document it will be assumed that all Packet Filter firewalls are using Stateful inspection.

Stateful Inspection Firewalls with Network Address Translation (NAT)

Many common firewalls provide the ability to hide the internal IP addresses from the outside world by masking them. The firewall provides a mapping from the external address to the internal address and vice-versa if desired. This operation is referred to as Network Address Translation or commonly abbreviated as NAT. NAT can cause problems for some applications including CE. These issues will be addressed later in this document.

Circuit-Level Gateway Firewalls (Proxies)

Operating on layer 5 of the OSI Model, proxies monitor TCP handshaking between packets to determine whether a requested session is legitimate. Information passed to a remote computer through a circuit-level gateway appears to have originated from the gateway. This is useful for hiding information about protected networks.

Different types of proxy servers exist for different protocols. Socks is a generic application proxy that works with most TCP-based applications. CE has been designed to provide direct support for Socks proxy servers.

Differences between IP Filter and Proxies

In comparing IP Filter and Proxies, one major difference between them is that IP Filter firewalls are transparent to the host and there are no settings or additional software required on the hosts, whereas a proxy server is not transparent.

Supported Firewall Types

The types of firewalls CE has been designed to work with are:

- IP Filtering-based firewalls
- IP Filtering-based firewalls with NAT
- Socks proxy servers
- Dual home Socks servers

Tested Firewalls

The firewall products that CE is compatible with are:

- Check Point FireWall-1 V4.1
- Microsoft ISA Server
- NEC e-Border 1.2

NOTE	The above firewalls have been used for the scenarios mentioned in this document. CE may work with other firewalls or systems (i.e. hardware NAT systems). Use of other firewalls and other scenarios not mentioned here are at the discretion of the customer implementing them.

CE Internal Communication

The CE system is an extensible framework. It is based on a simple infrastructure that provides two main functions:

- Servers that can publish the services they provide.
- Clients that can discover these services and then use them.

This internal framework has been designed to meet the complex needs of distributed applications and the multi-platform support required for enterprise environments today.

CE Framework

At the core of CE is the intelligence-providing layer called the CE Framework. The CE Framework is made up of a collection of services, which provide a series of "Business Intelligence" related functions implemented by one or more CE servers. When servers first connect to CE, they connect to the CE Framework. Once a server is connected, clients may connect to the framework to discover the server. Most CE client and server components communicate via the framework. With the noted exception being the Web Connector as it is excluded for security purposes.

Core of the CE Framework Communication

Transmission Control Protocol/Internet Protocol (TCP/IP) is at the core of the CE Framework communications. The use of TCP/IP is internal and hidden from CE administrators because it is so seamlessly integrated that it is not relevant.

TCP/IP was chosen for reliability, functionality, and cross-platform support. By using TCP/IP, CE's distributed components can communicate across many computers and platforms as if they were in one homogenous environment. Most firewalls are capable of filtering TCP requests; this is all that is needed to provide support for CE. However, for the purpose of using firewalls, there is one concept about the way that CE uses TCP/IP that needs to be understood, this is the method used to connect to servers.

Connecting to Servers

For each service that a server exposes, there is a unique identifier that contains information about the server and service. When a client or server wishes to connect to a service they will use the unique identifier of the service to connect to it. The information contained in the unique identifier for a service will be the computer name (hostname) and IP address of the server computer and the port on which to connect to that service. The information contained within the unique identifier is particularly critical when working with firewalls.

Client-Resource Discovery and Connection

When a client wishes to find a CE service, it attempts discovery by first looking for the server hosting the service on the CE Framework. This is achieved by the client issuing a request to a Directory Listing service hosted by the APS. The Directory Listing service contains an entry for each service available online at that time. Each entry contains the unique identifier of the service including the IP address, hostname and port number of the server as taken from the server. The client retrieves this information and then attempts to connect directly to the server hosting the service. If the client is separated from the APS and other servers via a firewall then communication across this boundary must be addressed. It should be noted that a client in this case refers to a CE Framework aware client. For example, the CE SDK and even CE servers (a server can be a client to another server) are clients to the CE Framework. Web clients using plug-ins such as the Crystal Reports Viewers are not direct CE Framework clients and thus are not directly affected by this concern.

Web Connector-Web Component Server (WCS) Discovery and Connection

As the Web Connector does not directly connect to the CE Framework, it must be manually configured to the location of the Web Component Server (WCS). Web Connector communication will be addressed in greater detail later in this document.

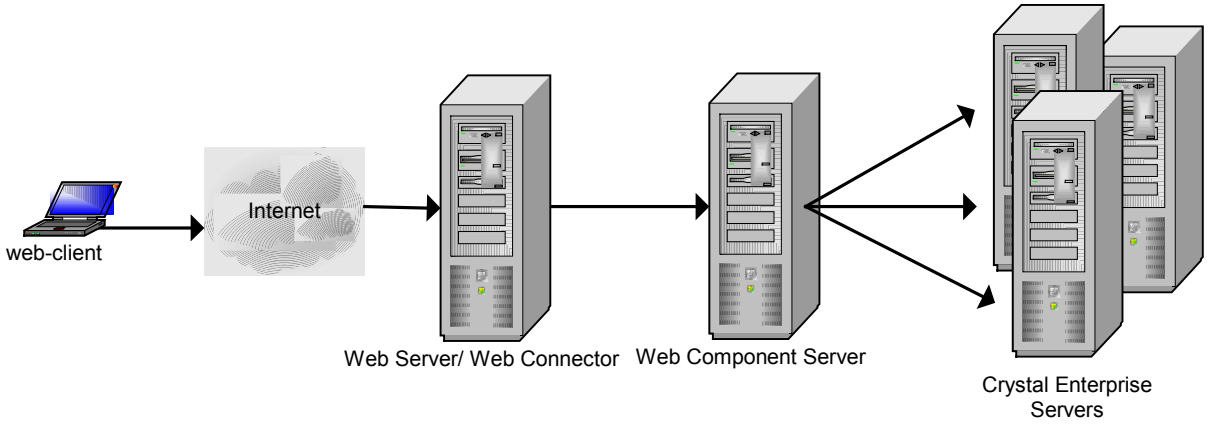
CE Secure Network Architecture

The following sections will present the CE web-based architecture, the components specifically designed to work with web-based network

configuration, and finally, how CE must be configured to work with different network configurations.

Architectural Overview

From the outset, CE was designed to provide an interface such that web applications could be built to meet the many needs of web-clients. Currently, any application providing content to a web-client is required to integrate with some form of network security. This comes in many forms, including authentication, authorization, and firewalls. To meet the demands of security savvy network administrators, CE has a web tier front end that allows for separation of web server, script processing, task processing and data access. For the web case, the architectural model is broken out as reflected in the following diagram.



The barrier that a secure system provides is commonly broken down into distinct layers. Each layer is identified by the communication from one network to another network via a firewall. To parallel this layering, the following section will first introduce the CE components involved and then address the requirements for components to communicate with each other through a firewall.

The Components

This section introduces the main components used in CE. CE Servers will be the terminology used for describing all servers in CE other than the WCS. To parallel the separation of layers due to a secure network environment this section discusses the key CE components.

Web Connector

The Web Connector is a web server plug-in, residing on the same computer as the web server. The web server loads the Web Connector to satisfy CE requests. To provide multiple web server and multi-platform support, several Web Connectors are provided. For example, ISAPI Extension for Microsoft Internet Information Server, NSAPI for Netscape's HTTP servers and CGI. Each type of

Web Connector provides the same functionality, the only difference being due to the web server environment in which they run.

The Web Connector has been specifically created to decouple the web server and CE; optionally giving network administrators another degree of separation in a secure network configuration. Its functionality is best described as being analogous to a one-way request router. That is, it forwards requests from the web server to the WCS. The WCS then communicates with any of the other CE servers required to satisfy the request. This Web Connector/WCS communication is all that is needed to provide access to a number of services offered by CE.

The WCS does not trust the Web Connector itself. Thus, it does not become a weak point in the security system; if the Web Connector is compromised it will not compromise the rest of the system.

Web Component Server (WCS)

The Web Component Server (WCS) is the heart of the web processing intelligence. It runs server side scripts, like CSP (Crystal Server Pages) to service client requests, handles state for the clients and can communicate with any of the servers in the CE environment.

The WCS is the component where trust by the rest of the CE system is established. This trust is only established when a user logs on. That is, when the WCS receives a user's logon credentials from the Web Connector it then logs on to the APS, which then establishes a trusted connection between the WCS and the APS for the current user's session.

CE Servers

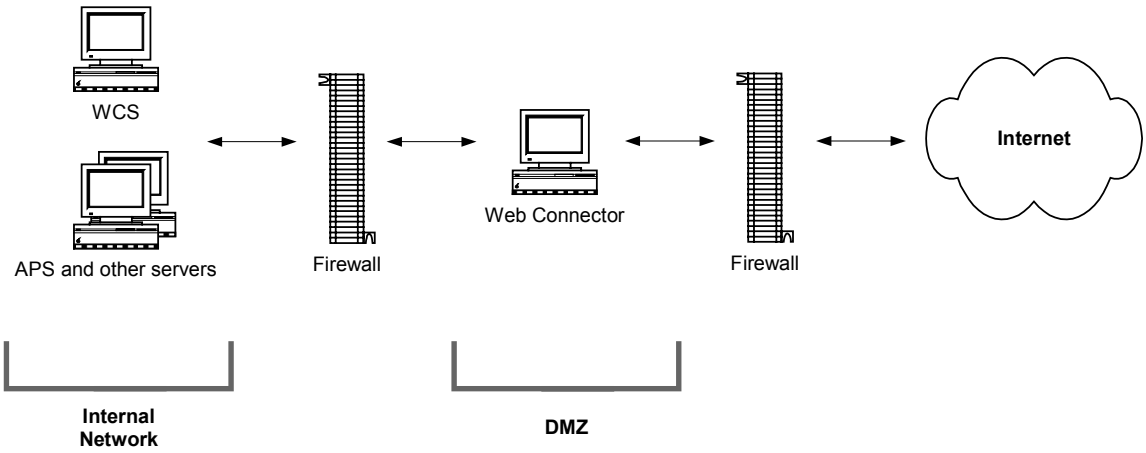
There is a collection of CE servers that service requests from the WCS. These servers may be the Page Server, Cache Server, Report Application Server, the Crystal Analysis Professional server, etc. Each server imposes a different set of network communication requirements on the system. That is, communication from the WCS to any of the CE servers must be addressed on a per server basis. In general, it is suggested that these servers be granted direct access to the WCS.

Web Connector & WCS Communication

This section will discuss the specific communication and network/firewall configurations allowed for Web Connector and WCS communication.

CE has been designed to specifically work with the web and the related security required for it. In most cases in which clients access protected information, the access is made through a web server running in a Demilitarized Zone (DMZ). This common situation is addressed here with several proposed solutions.

The only CE component that needs to provide direct service to external clients (browsers) is the Web Connector. Thus, the most logical and secure way to position the servers is to put the Web Connector in the DMZ and all other servers in the internal network.



This section assumes that the Web Connector and WCS reside on separate computers. If they do reside on the same computer then their communication is uninterrupted by firewalls and thus no specific firewall configuration is required for communication between these components. However, if this is the case, then it is more than likely that the Web Connector and WCS are both in a DMZ meaning that a firewall exists between the WCS and the rest of the CE servers. Communication between the WCS and the other CE servers via a firewall requires attention and is addressed in the next section.

The above diagram assumes that the WCS and the rest of the CE servers are running in a protected environment and that there is no interface between their communications. Similarly, requests from the Internet are directed only to the web server and thus do not interfere with CE.

The design of any secure system involves the balance of security and access. That is, access that is too great can leave private information vulnerable, while on the other hand if an environment is too secure it can limit performance. CE strives to maximize both by separating the architecture into a web-client communication tier and a data processing tier.

NOTE	There can be many WCSs for each Web Connector and many Web Connectors for each WCS. For simplicity, this document will only consider the relationship between one Web Connector and one WCS. To allow more of each, apply the same rules as explained here on a per Web Connector to WCS connection basis.

Web-Client Communication Tier

Communication from the web-client to the CE servers is serial until it reaches the WCS where network administrators are allowed to lock down on the web-client/WCS communication. At the WCS, data is ready to be presented in final form to the client, thus making it a natural location at which to begin securing the data and communication.

Data Processing Tier

The data processing tier involves one or more of the following: data collection and consolidation, processing and formatting. These actions are separated in CE and can be placed in a secure environment such that they may occur freely without compromising the overall security of the system.

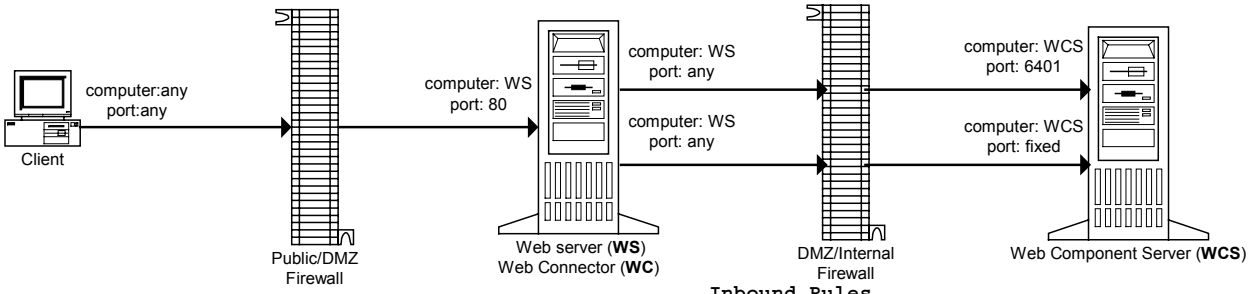
Setup and Configuration

There are currently two methods for setting up the network to configure the Web Connector/WCS communication. The first, and more common, is using a firewall, while the second involves the use of a Socks server with a firewall.

Basic Setup and Configuration

The simplest and most common secure network configuration for Web Connector/WCS communications is as follows:

Suggested Crystal Enterprise Network Configuration



Inbound Rules

Source Computer	Port	Destination Computer	Port
any	any	WS	80

Outbound Rules

Source Computer	Port	Destination Computer	Port
---------- none ----------			

Inbound Rules

Source Computer	Port	Destination Computer	Port
WS	any	WCS	6401
WS	any	WCS	fixed

Outbound Rules

Source Computer	Port	Destination Computer	Port
---------- none ----------			

NOTE	The Public/DMZ (outside) firewall rules are provided for comparison purposes only. CE web clients communicate via the web server and http requests. The Web Connector, WCS, and CE servers only communicate from the web server to the internal network. Thus, as long as external web requests are allowed to the web server, configuration rules on the outside firewall have no effect on CE.

High Level Description of Web Connector/WCS Communication

Below is a general description of the Web Connector/WCS communication:

1. First, the Web Connector makes a discovery request to the WCS.
2. The WCS replies back to the Web Connector with an IP address, hostname, and port that the Web Connector can use for sending web requests to.
3. The Web Connector then makes all new requests to the WCS on the specified IP address (or hostname) and port.

Thus, the initial connection to the WCS is a two-stage process. Once the Web Connector has discovered the WCS, all further communication for web-based traffic will go to the location that the WCS provided using the hostname/IP address and port. To better understand the firewall implications, the following describes this same process in more detail.

Detailed Description of Web Connector/WCS Communication

Below is a detailed description of the Web Connector/WCS communication:

1. The Web Connector makes a TCP connection to the WCS, on port 6401.
2. Once a TCP connection has been established, IP packets are sent back and forth on this channel.
3. In this TCP connection, the Web Connector receives the unique identifier of the WCS service from the WCS.
4. The unique identifier of the WCS service contains the IP address or name of the WCS, as well as the port number specified in the *requestPort* option (if the option is not specified, a free port is picked at random).
5. After the Web Connector receives the unique identifier of the WCS service, the TCP connection to the WCS on port 6401 is closed.
6. The Web Connector will then make a new TCP connection to the WCS, on the request port.
7. Once this TCP connection is made, IP packets will be sent back and forth on this channel.

The key point to note is that only the Web Connector makes a TCP connection to the WCS and not the other way around. However, at the IP level, packets are sent in both directions. As an analogy, a web browser makes a connection to a web server, sends an http request, and receives an http response back followed by the connection being closed. The web server never 'calls back' to the web browser, it merely sends data back on the established connection. This is important to note as Stateful firewalls only need inbound access rules to allow the Web Connector and WCS to communicate. Outbound access rules are not needed.

NOTE	The Web Connector may open multiple connections to the WCS or other WCSs for the purpose of distributing request load.

Network Address Translation (NAT)

There are several areas where Network Address Translation (NAT) can be employed on a network. NAT employed on the outer firewall does not create issues for CE. However, if NAT is employed on the inner firewall between the Web Connector and the WCS, then the components need to be configured to work with this scenario. This is because the Web Connector gets the IP address and port number of the WCS from the unique identifier, which, by default, contains the internal IP address and TCP port number rather than the translated ones. As this internal IP address may not be routable from the web server, the WCS can be configured to return a resolvable host name to the Web Connector. This configuration is setup through the switches explained below.

The port switch

CE 8.5 gives the administrator the ability to set routing specific switches on a command line. Let's take a look at an example of a WCS command line that allows for communication between the connector and the WCS on a NAT firewall deployment.

-port jackpot.crystaldecisions.com:6401 **–requestport** 3366

The –port switch has a fully qualified domain name, or IP address, and a port number. In this example, the WCS will listen for requests on port 6401. This information will be sent from the WCS to the connector after the initial connection or handshake is made between the connector and the WCS. The connector will then attempt to resolve the WCS as hostname jackpot (the externally routable hostname of the WCS) for the remainder of the session. Setting a value for this switch ensures that the WCS will *not* tell the connector to use an internal IP address or non-routable machine name after the initial handshake/connection.

The requestport switch

In our sample WCS command line we also have a requestport switch.
By default, the WCS will dynamically choose a port on which the Web Connector will communicate *after* the initial handshake has been made. Setting the requestport switch will tell the connector to communicate with the WCS on a specific port, (in this case it's **3366**) after the initial handshake/connection. At this point the administrator will need to open two inbound ports on the firewall:
6401 (the WCS listens for requests from the connector on this port)
3366 (the connector communicates with the WCS on this port after the connection has been established on **6401**)

A note about switches and CE servers

It is important to note that the -requestport switch works slightly differently for different servers.

APS and WCS

The –requestport switch specifies the secondary port that the APS uses for identifying other servers and for registering with itself and/or a cluster. It specifies the port on which the WCS listens for replies from the APS and the other CE servers. In both cases, this switch does not control the ports on which the APS and WCS listen for initial communication from other servers. It only controls what ports they listen on and communicate on after initial contact is made.

In the above example with the WCS, if the –port switch is not used to indicate a port, the default port for initial communication is 6401 (APS is 6400). If the initial communication port needs to be changed, the –port switch must be used.

-port jackpot.crystaldecisions.com:12500 **–requestport** 3366

The WCS will listen for initial communication on port 12500 and continue communication on port 3366 once initial contact has been made on port 12500.

Other Servers

-requestport specifies the port on which the server listens for CE requests. That is, it determines both the port the server is listening on and the port it communicates on.

–requestport 14000

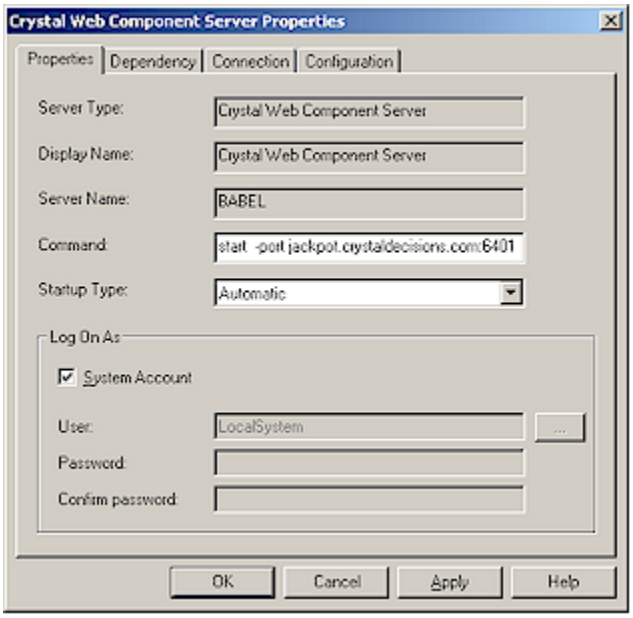
If the above switch is added to the FRS command line, when the FRS registers with the APS it tells the APS that it is registered and listening on port 14000. When a server wants to initiate communication with the FRS, it talks to the APS. The APS tells it that the FRS is listening on port 14000. The other server then initiates communication with the FRS on port 14000. When the FRS gets the request, it tells the other server to go to port 14000 to continue communication - the same port it is listening on for initial communication from other servers.

Setting the WCS command line switches

On the WCS machine, open the Crystal Configuration Manager from the Start menu.

1. Stop the WCS service.

2. Double click on the WCS service and you will see a dialog box similar to the one below.



3. In the command line of the WCS, enter switches for the hostname, portname and request port.

4. Click OK and then start the WCS

Configuration of the –requestport and –port switches in a Unix environment

In a Unix environment, configuration of the –requestport and –port switches is made in the ccm.config file. The default location of this file is the CE install directory. For example, /export/home/crystal.
NOTE: The service you are reconfiguring should be stopped before the changes are made. In the example below, we have made changes to the WCS configuration.

wcsLAUNCH='"/crystal/user3/crystal/enterprise/generic/crystalrestart.sh" -protect "/crystal/user3/crystal/enterprise/solaris_sparc/wcs/bin/crystalwcsd" -name jackpot.wcs -ns tstest67:6430
-port 6431 -port jackpot.crystaldecisions.com:6431
-loggingPath "/crystal/user3/crystal/logging" -pidFile "/crystal/user3/crystal/serverpids/user3_wcs.pid" -restart -fg'

Firewall Configuration

The firewall will require two rules to grant Web Connector/WCS communication. That is two inbound rules, the first for the discovery request from the Web Connector and the second for normal requests. No outbound traffic is required and thus can be restricted.

Inbound Firewall Rules

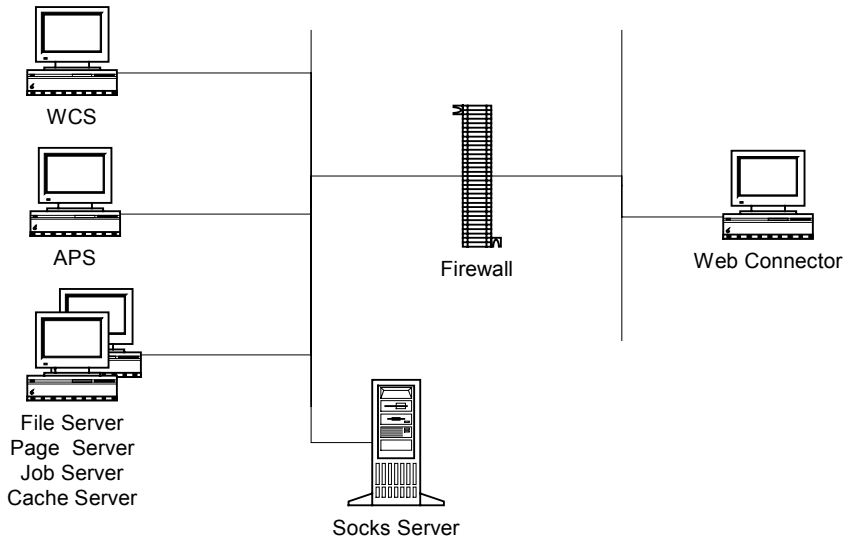
Source		Destination	
Computer	Port	Computer	Port
Web Connector computer	Any	WCS computer	6401
Web Connector computer	Any	WCS computer	Fixed or any

Web Connector Configuration

The Crystal Configuration Manager (CCM) on the computer on which the Web Connector resides (web server) must be used to configure the Web Connector. For the purpose of load balancing and fault tolerance, the Web Connector is capable of communicating with many WCSs. To configure the list of WCSs, add each WCS computer and port in the CCM. For explicit instructions on how to configure the Web Connector on a Windows NT or 2000 machine, refer to the Administrator's Guide in the \Doc directory of the CE CD.

Setup and Configuration with a Socks Server

As CE provides direct support for Socks, if the basic firewall setup is not desired, provides limitations, or use of a proxy server is desired, the following configuration can be used:



Firewall Configuration

As per the requirements of the Proxy server.

Web Connector Configuration

On the Web Connector computer, specify the Socks server on the WCS' Configuration tab.

The discovery request from the Web Connector to WCS port 6401 is routed via the Socks server. If the Web Connector is configured with the IP address of the WCS, then the Socks server will route the request directly. However, if the host name is used the Socks server will need to resolve it. (For example, if the Web Connector and WCS use the NetBIOS name while the Socks server is installed on a UNIX server that does not support NetBIOS names, then the Socks server must be able to resolve the name as specified by the Web Connector, e.g. by using the local hosts file).

WCS Configuration

The WCS configuration occurs via the APS. On the APS computer, specify the Socks server on the APS' Connection tab.

Even though the Web Connector connects to the WCS, the WCS' Socks server information is configured on the APS rather than the WCS. This is because the WCS will obtain the Socks setting from the APS. As an option, the Socks setting may be added on the WCS, but it is still necessary to configure Socks on the APS. Otherwise, the APS will make a connection to the WCS through the Socks server, which is undesirable.

Socks Configuration

Access control rules on the Socks server should be set to something similar to the following:

Source		Destination	
Computer	Port	Computer	Port
Web Connector computer	Any	WCS computer	6401
Web Connector computer	Any	WCS computer	Any

For Socks Configuration in a Unix environment please see the **Crystal Enterprise 8.5 Administrator's Guide**.

Placement of Web Files (Virtual Path Mapping)

If the Web Connector and WCS are not running on the same computer then an issue may exist regarding where to install web content files and how to configure the components for this. In addition, if running a Web Connector on UNIX, issues will exist regarding where the files are located. For an explanation of the issues involved and how to set up and configure components, please refer to the technical brief, "Virtual Directories and Path Mapping with CE", which can be downloaded from

http://support.crystaldecisions.com/docs

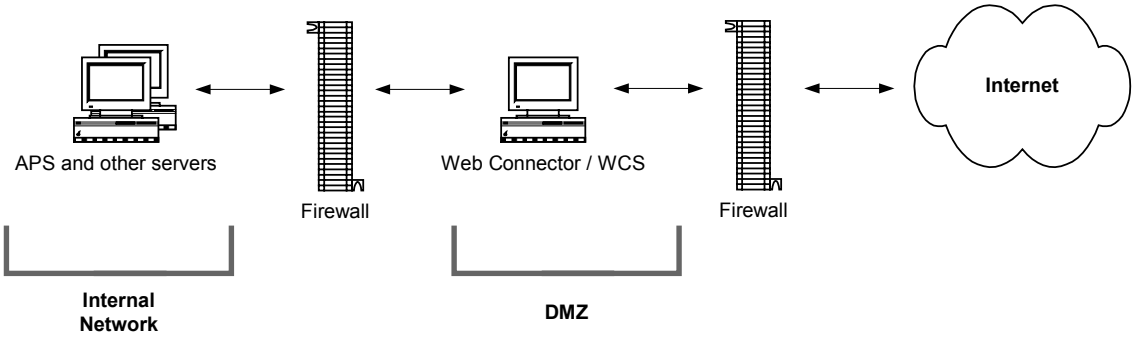
by searching for the filename ce8_virtual_dir_path_mapping.pdf .

WCS and CE Server Communication

The next level of communication is between the WCS and the other CE servers. This level of communication differs from the Web Connector/WCS communication in that many CE servers can connect to the WCS. Discovery of each server is done via the APS; requests can be made back and forth between the components. The communication requirements are much greater than the Web Connector/WCS communication and thus the network configuration is more complex.

A possible configuration is to put both the Web Connector and WCS in the DMZ and all other CE servers in the internal network. This option may be of choice if the WCS is hosting some custom application that may need to provide service to external clients (since the WCS is effectively a web application server). The following scenario illustrates one possible configuration.

As the firewall-related concerns with the Web Connector/WCS communication are independent of the WCS/CE server communication, please refer to the previous sections for it.

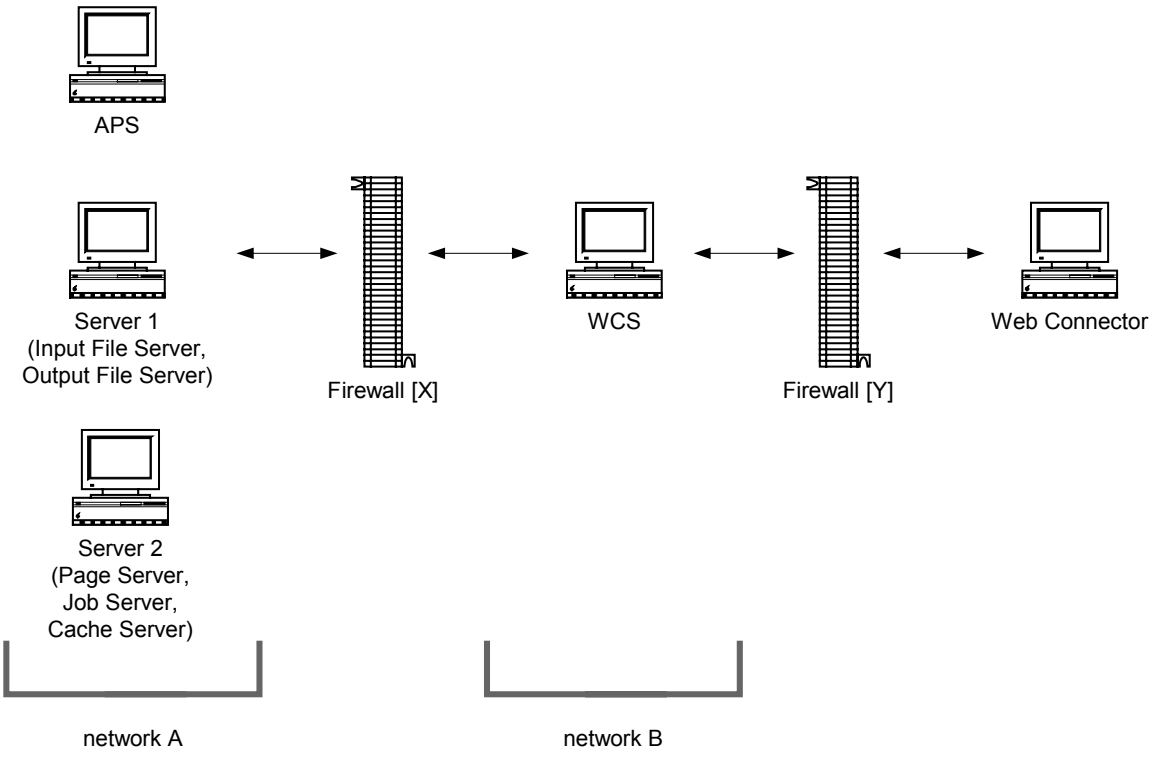


Setup and Configuration

There are currently many network setups for configuring WCS/CE server communication. The first, and more common, is using a firewall, while the others involve the use of Socks server and firewalls.

Basic Setup and Configuration

The simplest and most common secure network configuration for WCS/CE server communication is as follows:



The Web Connector can be installed on the same computer as the WCS in the DMZ, but that decision will not affect WCS/CE server communication.

The WCS will need to communicate with the CE servers. For each server the communication channels need to be open such that they can freely communicate. These channels will be addressed on a per server basis.

The communication discovery is similar to the Web Connector except instead of communicating with each server for the request port, this information is retrieved via the Directory Listing service.

High Level Description of WCS/CE server Communication

Below is a general description of the WCS/CE server communication:

1. Discovering the Directory Listing service

   a) First, the WCS will make a discovery request to the APS.

   b) The APS replies back to the WCS with the name/IP address and port of the Directory Listing service hosted by the APS.

   c) The WCS can now make all requests to the Directory Listing service on the specified name/IP address and port.

2. Discovery and Connecting to Servers

   a) The WCS will make a request to the Directory Listing service on the APS requesting the server it wants.

   b) The Directory Listing service will reply with the name/IP address and port number of the requested server.

   c) The WCS will now make all requests to that server via the specified name/IP address and port.

   d) The WCS may register a listener with some of the servers. In this case, the server will make a connection request back to the WCS.

   e) The WCS will repeat the server discovery steps for other servers with which it wishes to communicate.

The initial connection to the Directory Listing service is a two-step process, followed by each server discovery and connection being another two-step process. The WCS does a Directory Listing discovery at startup and will only do it again to reestablish a connection if it is broken. To help provide understanding of the firewall implications above, the following is a more detailed explanation.

Detailed Description of WCS/CE server Communication

Below is a detailed description of the WCS/CE server communication:

1. Discovering the Directory Listing service

   a) The WCS makes a TCP connection request to the APS on port 6400.

   b) Once a TCP connection has been established, IP packets are sent back and forth on this channel.

   c) In this TCP connection, the WCS will receive the unique identifier of the Directory Listing service.

   d) The unique identifier contains the IP address and hostname of the APS computer, (specified by the –port switch) as well as the port number specified in the request port switch for the APS (if the option is not specified, a free port is picked at random)

   e) After the WCS receives the unique identifier of the Directory Listing service, the TCP connection to the APS on port 6400 is closed.

f) The WCS will make a TCP connection to the Directory Listing service, on the IP address, hostname, and the request port that were provided.

g) Once this TCP connection is made, IP packets will be sent back and forth on this channel.

2. Discovery and Connecting to Servers

a) The WCS will make a request to the Directory Listing service on the existing TCP connection for the service it wants.

b) The Directory Listing service returns the unique identifier for the service requested.

c) The unique identifier contains the name/IP address of the server, as well as the port number specified in the request port option of that server (if the option is not specified, a free port is picked at random).

d) The WCS will make a TCP connection to the server, on the name/IP address and request port that were provided.

e) Once this TCP connection is made, IP packets will be sent back and forth on this channel.

f) The WCS may register a listener with some of the servers to receive events and messages.

g) The WCS will send the server the unique identifier of the WCS' listener interface. The IP is the WCS' computer's name/IP and the port is picked at random unless otherwise specified in the WCS' command line.

h) The server will make a new TCP connection to the WCS listener interface.

i) The WCS will repeat the server discovery steps for other servers with which it wishes to communicate.

It is important to note that unlike the Web Connector case, it is necessary to receive communication requests back from the servers. Thus, the firewall must allow inbound and outbound communication.

Firewall Configuration

The firewall requires two rules for the Directory Listing service and a rule for each server to which it wishes to connect. For the servers to communicate back to the WCS, a single outbound rule is required. It is assumed all other traffic is rejected.

Inbound Firewall Rules

Source		Destination	
Computer	Port	Computer	Port
WCS	Any	APS	6400
WCS	Any	APS	Fixed or any
WCS	Any	Server1(Input FRS)	Fixed or any
WCS	Any	Server1(Output FRS)	Fixed or any
WCS	Any	Server2(Page Server)	Fixed or any
WCS	Any	Server2(Job Server)	Fixed or any
WCS	Any	Server2(Cache Server)	Fixed or any

Outbound Firewall Rules

Source		Destination	
Computer	Port	Computer	Port
Network A	Any	Any	Any

WCS Configuration

There is no configuration of the WCS required for the current discussion.

CE Server Configuration

The servers, by default, will dynamically choose a port on which the WCS will initiate new requests. To set this to a fixed port, add the "–requestPort <port number>" command line option to each server via the CMC.

NOTE: In a Unix deployment of CE, these changes would be made in the ccm.config file after stopping the appropriate server. This file is usually located in the Crystal install root.

To stop the page server:
$ ccm.sh -stop pageserver

Then, edit the ccm.config file to add the request port switch as shown below. The pageserver can be restarted when you have finished editing the file.

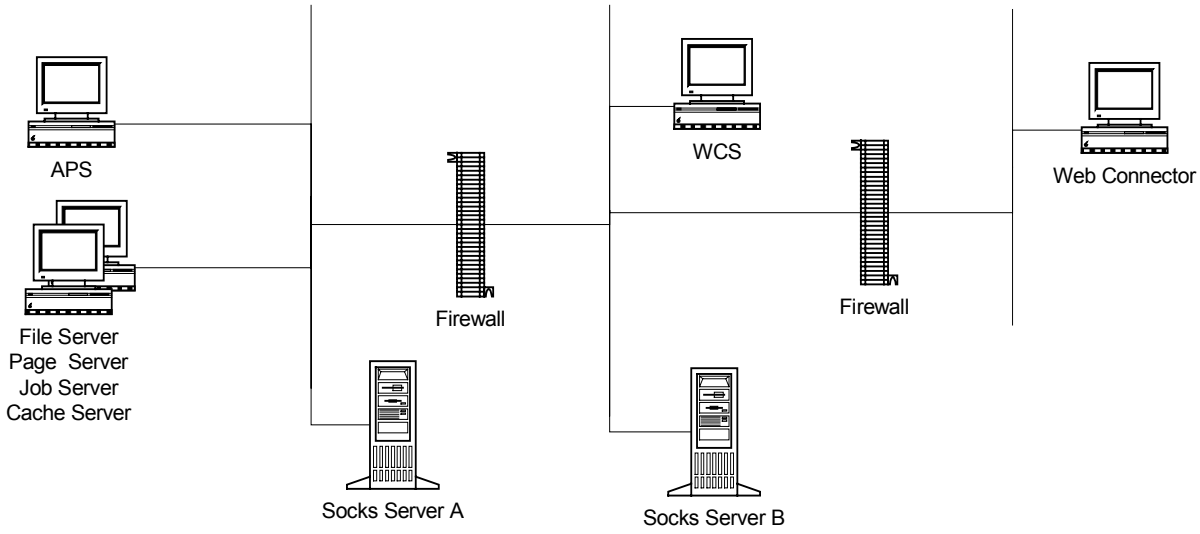
pageserverLAUNCH="'/crystal/user1/crystal/enterprise/generic/crystalrestart.sh "-protect "/crystal/user1/crystal/enterprise/aix_rs6000/crystalpagesd" -name jackpot.pageserver -ns jackpot.ts.crystaldecisions.com:6410 **– requestPort 7500** -loggingPath "/crystal/user1/crystal/logging"
- pidFile"/crystal/user1/crystal/serverpids/user1_pageserver.pid" -restart -fg'

To start the pageserver:
$ ccm.sh -start pageserver

Setup and Configuration with a Socks Server

As CE provides direct support for Socks, if the basic firewall setup is not desired, provides limitations, or the use of a proxy is desired, there are many configurations available.

The following is a sample:



There are many other configurations possible including multiple firewalls with multiple WCSs and multiple Web Connectors. Multiple Socks servers can be used to support such a scenario. More in-depth explanation of WCS/CE server communication will be covered in future revisions of this document. Please refer to the **Crystal Enterprise 8.5 Administrator's Guide** for information regarding how to set up and configure for these scenarios.

Contacting Crystal Decisions for Technical Support

We recommend that you refer to the product documentation and that you visit our Technical Support web site for more resources.

Self-serve Support:

http://support.crystaldecisions.com/

Email Support:

http://support.crystaldecisions.com/support/answers.asp

Telephone Support:

http://www.crystaldecisions.com/contact/support.asp