Sun Java™ System

Calendar Server 6 Deployment Planning Guide

2004Q2

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 816-6709-10

Contents

List of Tables

List of Figures

Preface

The *Sun Java System Calendar Server 6 2004Q2 Deployment Planning Guide* contains the information you need to deploy Sun Java™ System Calendar Server 6 2004Q2. This guide helps you through the process of understanding Calendar Server, evaluating and analyzing your site, and designing the kind of deployment architecture that meets your organization's needs.

Who Should Read This Guide

This guide is intended for site planners, system administrators, and support specialists who are responsible for planning and deploying Calendar Server at their site.

What You Need to Know

This guide is for individuals who are responsible for assessing and deploying Calendar Server at your site, including:

- Evaluators

- Architects
- System administrators

This guide assumes you are familiar with the following:

- How to install enterprise-level software products
- DWP, WCAP, and LDAP protocols
- Solaris system administration and networking

Conventions

The following table describes the typeface conventions used in this guide.

Table 1 Typeface Conventions

Typeface	Meaning	Examples
`AaBbCc123` (Monospace)	API and language elements, HTML tags, web site URLs, command names, file names, directory path names, on-screen computer output, sample code.	Edit your `.login` file. Use `ls -a` to list all files. `% You have mail.`
`AaBbCc123` (Monospace bold)	What you type, as contrasted with on-screen computer output.	`% su` `Password:`
AaBbCc123 (Italic)	Book titles.	Read Chapter 6 in the *User's Guide*.
	New words or terms.	These are called *class* options.
	Words to be emphasized.	You *must* be superuser to do this.
	Command-line variables to be replaced by real names or values.	The file is located in the *cs_svr_base*`/cal/sbin` directory.

The following table describes placeholder conventions used in this guide.

Table 2 Placeholder Conventions

Item	Meaning	Examples
product_base	Placeholder for the directory where the product is installed.	The *cs_svr_base*`/bin` directory might be `/opt/SUNWics5/cal/sbin`.

The following table describes the symbol conventions used in this book.

Table 3 Symbol Conventions

Symbol	Meaning	Notation	Example
[ ]	Contain optional command options.	0[*n*]	04, 0
{ }	Contain a set of choices for a required command option.	d{y|n}	dy
|	Separates command option choices.		
+	Joins simultaneous keystrokes in keyboard shortcuts that are used in a graphical user interface.		Ctrl+A
-	Joins consecutive keystrokes in keyboard shortcuts that are used in a graphical user interface.		Esc-S
>	Indicates menu selection in a graphical user interface.		File > New File > New > Templates

Resources on the Web

In addition to this guide, you will want to refer to Sun Java™ System Calendar Server 6 documentation. Use the following URLs to see this documentation:

http://docs.sun.com/db/coll/CalendarServer_04q2

Third-party URLs are included in this document to provide additional, related information.

NOTE Sun is not responsible for the availability of third-party Web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused by or in connection with the use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

How to Report Problems

If you have problems with Calendar Server, contact Sun customer support using one of the following mechanisms:

- Sun Software Support services online at

  http://www.sun.com/service/sunone/software

  This site has links to the Knowledge Base, Online Support Center, and ProductTracker, as well as to maintenance programs and support contact numbers.

- The telephone dispatch number associated with your maintenance contract

So that we can best assist you in resolving problems, please have the following information available when you contact support:

- Description of the problem, including the situation where the problem occurs and its impact on your operation
- Machine type, operating system version, and product version, including any patches and other software that might be affecting the problem
- Detailed steps on the methods you have used to reproduce the problem
- Any error logs or core dumps

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. Use the web-based form to provide feedback to Sun:

http://www.sun.com/hwdocs/feedback/

Please provide the full document title and part number in the appropriate fields. The part number is a seven-digit or nine-digit number that can be found on the title page of the book or at the top of the document. For example, the part number of this *Calendar Server Deployment Planning Guide* is 816-6709-10.

Chapter 1

Understanding Calendar Server

This chapter provides an overview of Sun Java™ System Calendar Server 6 2004Q2, the business reasoning behind deploying Calendar Server, and the deployment process itself.

This chapter contains the following sections:

Calendar Server Overview

Sun Java System Calendar Server 6 2004Q2 (formerly Sun™ ONE Calendar Server) is a high- performance, Internet standards-based calendar server designed with the scalability to meet the needs of customers ranging from medium-and large-sized enterprises to even the largest Internet, telecommunication, and enterprise service provider. Through a native Web browser interface or connectors to other calendar clients, including Microsoft Outlook, Calendar Server provides group scheduling and personal calendaring to consumers at home or at work, while enabling them to share calendar information with others over the Internet. The user interface (UI) can be customized to include Web links for e-commerce, banner ads, logo, or brand of the calendar server customer, and more.

Calendar Server provides one of the industry's most open, interoperable, and high-performance time and resource management solutions. Through its scalability, performance, and reliability, it provides the features you require at a lower total cost of ownership than alternative solutions. Native support for iCalendar standards enables users to schedule events in a format that is easily shared across the Internet. The Java System Calendar Server employs standards and protocols such as:

- Internet Calendaring (iCalendar)
- iCalendar Transport-Independent Interoperability Protocol (iTIP)
- iCalendar Message-based Interoperability Protocol (iMIP)
- eXtensible Markup Language (XML)
- Lightweight Directory Access Protocol (LDAP)
- HyperText Transport Protocol (HTTP)

The Calendar Server architecture is flexible, extensible, and scalable both vertically (by increasing the number of CPUs per system) and horizontally (by introducing additional servers into the network). As a result, Calendar Server can be thought of as a system of servers that can be configured to fit a variety of needs. It can remain in isolation as a standalone calendar server, or it can be configured with many instances, having the various services duplicated or split between them.

Calendar Server makes use of plugins to obtain external services. It also supports both LDAP-and identity-based deployments, and integrates with the Sun Java™ System Identity Server (formerly Sun™ ONE Identity Server), Sun Java™ System Portal Server (formerly Sun™ ONE Portal Server), and Sun Java™ System Instant Messaging (formerly Sun™ ONE Instant Messaging) to provide additional functionality.

Calendar Server provides the following benefits:

- Web-based group scheduling, free-busy lookup, and corporate directory lookup
- Web-based resource scheduling for conference rooms, projectors, and other resources
- XML-based customization (color, login, user interface, logo, branding, and so on)
- Support for standards-based event and calendar data feeds published in XML or iCalendar formats, which improves communication and can offer new revenue opportunities through commerce links and banner ads

- Native LDAP support, with an API for other directory services
- Connectors to additional calendar clients including Microsoft Outlook and Evolution, enabling these clients to perform scheduling on Calendar Server
- Support for hosted domains
- Simplified system management, online backup and restore, and entire database backup and restore
- Support for multiple calendars, such as work, family, friends, and more
- Support for public and private calendars, as well as public, private, and confidential individual events
- Support for layered calendar views, which enables users to combine two or more calendars into a single view, providing improved communication and productivity
- Automatic e-mail notification of appointments, invitations, and reminders sent to selected recipients; integrates with Java System Instant Messaging to provide automatic pop-up reminders
- Support for multiple owners of each calendar, to facilitate communication and productivity with project teams and community groups
- Ability to delegate calendar ownership to others who may act on behalf of the primary owner
- Daily, weekly, monthly, yearly, and comparison views
- Supports hundreds of thousands of users through a scalable, networked, server-to-server, client- server architecture
- Supports Secure Sockets Layer (SSL) encryption, LDAP authentication, authentication plugins, and identity-enabled single sign-on (SSO) through the Java System Identity Server

For more information on Calendar Server concepts, see the *Sun Java System Calendar Server Administration Guide*:

http://docs.sun.com/doc/817-5697

How Calendar Server Satisfies Business Needs

Calendar Server provides one of the industry's most open, interoperable, and high-performance time and resource management solutions. Calendar Server provides the features you need at a lower total cost of ownership than alternative solutions. Through its flexible and extensible architecture, Calendar Server scales both vertically (by increasing the number of CPUs per system) and horizontally (by adding more servers to the network).

The following table summarizes the benefits to the enterprise provided by Calendar Server.

Table 1-1 How Calendar Server Benefit the Enterprise

Key Feature	Benefit to the Enterprise
High performance and scalability	Enables efficient communications and improves quality of service for both enterprises and ISPs.
Extensive security features	Protects the integrity of communications and data and the privacy of employees, customers and partners, and enables compliance with industry regulations.
Scalable, robust and extensible components	Enables deployment of unified communication services, bringing together telephone services with email notification, faxing, paging, and other technologies.
Extensible collaboration platform for scheduling events, managing tasks and resources	Calendar Server improves time and resource management, and enhances user productivity.
Group scheduling for meetings and events	Calendar Server improves team collaboration and communication across the enterprise.
Information sharing through hyperlinks in events or tasks	Calendar Server facilitates collaboration through exchange of information relevant to tasks or events.
Open, modular, and standards-based architecture	Enables customers to deploy customized and personalized solutions.

Making the Calendar Server Deployment Highly Available

Calendar Server provides high-availability options that supports the Sun™ Cluster services. With this option, a secondary Calendar Server host provides services to users if the primary system is taken offline for maintenance or is down due to a problem.

Additionally, you can deploy Calendar Server in a highly available configuration through use of redundant components. This kind of deployment gives services a high level of uptime. A highly available deployment of this sort requires the redundancy of every component in the service architecture. These components include a duplicate data store server, duplicate network interface cards, and duplicate system storage.

NOTE This guide does not discuss the details of using Sun Cluster in highly available deployments for Calendar Server. See the Sun Cluster and Calendar Server documentation for more information on this topic.

Using Portal Server with Calendar Server

You can install Calendar Server with Portal Server to provide access to a calendar portlet in a portal page. This portlet provides a calendar schedules and address book information. The integration of Portal Server includes single sign-on capabilities between Portal Server and Calendar Express web client (as well as other Messenger Express and Communications Express clients).

The following two components of Portal Server provide additional functionality to a basic Calendar Server deployment:

- **Portal Server Desktop.** Calendar Server installed on Portal Server enables users to launch Calendar Express.
- **Sun Java™ System Portal Server 6, Secure Remote Access.** Enables remote end users to securely connect to Secure Remote Access organization's network and its services over the Internet. End users access Secure Remote Access by logging in to the web-based Portal Server Desktop through the Secure Remote Access gateway. The authentication module configured for Portal Server authenticates the end user. The end-user session is established with Portal Server and the access is enabled to the end user's Portal Server Desktop.

NOTE This guide does not discuss portal deploying Calendar Server in a portal environment. See the Portal Server documentation for more information.

Understanding the Deployment Process

The Calendar Server deployment process consists of the following general phases:

- Deployment design
- Development
- Prototype testing
- Production rollout

The deployment phases are not rigid: the deployment process is iterative in nature. Nevertheless, the following subsections discuss each of the deployment phases independently.

Designing the Deployment and Architecture

In general, during the deployment design phase, you construct a deployment architecture based on the deployment scenario specified in the requirements analysis phase. The objective is to map the logical building blocks (the logical architecture) to a physical environment (a physical topology) in a way that meets the system requirements specified in the deployment scenario.

One aspect of this design is sizing the physical environment to meet load, availability, and performance requirements. The deployment architecture takes into account details of the physical topology, such as the capabilities of different computing nodes and network bandwidth, in assigning system servers and application components to the computing nodes in the environment.

Objectives of Your Deployment

Before you begin your deployment planning, a good question to ask is:

Why is my organization deploying Calendar Server?

Several reasons to consider are:

- **Cost savings.** The total cost of ownership per user is lower than using other calendar products on the market.
- **Increased productivity.** Your calendar users can manage their events and tasks as well as schedule meetings and appointments with others in the organization. Your users can also manage calendar groups and resources such as meeting rooms and equipment. They can also synchronize their calendars with mobile devices (PDAs) and Microsoft Outlook.
- **Improved scalabilty and availability.** Calendar Server scales both horizontally and vertically. If your organization grows, you can easily upgrade your configuration by upgrading a server or add more servers.
- **Improved security.** If you deploy Calendar Server on a Solaris system, your organization can avoid many viruses and other security threats that are common in Windows environments.
- **High availability (HA) configuration.** Integration with clustering software such as the Sun Cluster enables you to configure Calendar Server as a high availability service. If you experience a software or hardware failure, Calendar Server fails over to a secondary server.

Calendar Server Deployment Team

Deploying Calendar Server usually involves a number of people, each with different roles and responsibilities. In a small organization, one person might perform several roles. Some of the roles to consider are:

- Program Manager oversees the overall Calendar Server deployment and is responsible for its success or failure.
- Calendar Server Administrator performs day-today administrative tasks to manage Calendar Server and might also be responsible for installing and upgrading Calendar Server.
- Performance Engineer test and monitors the Calendar Server performance for the trial and production deployments to see if the deployment criteria is met.
- Development Engineering writes Calendar Server applications or plugins, or customizes the Calendar Server user interface (UI), if required.
- Documentation Specialist writes any customized documentation for administrators and end users.
- Education / Training develops training classes and material.

Support Specialists support both the trial and production deployments.

Calendar Server End Users

End users can connect to Calendar Server by using the Calendar Express Web client, Communications Express web client, or Sun Java™ System Connector for Microsoft Outlook.

Questions about end users at your site include:

- How many total Calendar Server end users will your site have?
- How will your end users connect to Calendar Server? Calendar Express? Microsoft Outlook? A combination of clients?
- How many geographic locations are involved? Are your end users all in the same time zone or are they in different time zones?
- Will your end users log into Calendar Server at the same time each day?
- How many active end users will your deployment have during peak use?
- How fast will your end user base grow?
- What are your specific performance requirements for the Calendar Server end users?
- What are your single sign-on (SSO) requirements?
- Are any of your users migrating from Netscape Calendar 4.x?
- Are your end users planning to use Sun™ ONE Synchronization?
- Are you planning to customize the UI for your end users?
- Does your site plan to use a proxy server?
- Does your site plan to use load balancing?

Expected End User Performance

What are your specific performance requirements for your end users? For example:

- What end user response times are acceptable?
- Can you tolerate a possible degradation in performance during peak load times?

What configuration do you plan to use for your deployment? Calendar Server configuration scenarios include:

- Single Calendar Server instance
- Single front end with single back-end database server

- Multiple front-end servers with multiple back-end database servers using LDAP CLD plug-in
- Multiple front-end / back-end servers using LDAP CLD plug-in
- High Availability (HA) configuration

If you plan to configure multiple front-end servers, how do you plan to distribute your end users?

If you plan to configure multiple back-end database servers, how do you plan to distribute your database? For example, geographically.

What plans to you have for growth? For both front-end and back-end servers?

Development and Customization

The logical architecture specified in the requirements analysis stage of the life cycle determines the scope of the development work needed to implement a solution.

Additional work might be necessary, either in extending services through the use of APIs, or in customizing look and feel, for example, introducing a corporate branding.

For some solutions, development and customization might be quite extensive, requiring you to develop new business and presentation services. In other cases, it might be sufficient to customize existing graphical user interfaces, such as the Portal Server desktop, to achieve the functionality required.

For more information on using product APIs and customizing product functionality, see the appropriate component product documentation, including:

- *Sun Java System Communications Services Event Notification Service Guide*
- *Sun Java System Calendar Server Developer's Guide*

Prototyping and Testing

In the prototyping phase, you prototype your deployment design by implementing the deployment architecture in a test environment. You use new application logic and server customizations from the development effort, as described above (see "Development and Customization"), to perform proof-of-concept deployment testing. This phase involves installing, configuring, and starting up distributed applications and any required infrastructure services in your test environment.

If prototype testing reveals shortcomings in your deployment architecture, you modify the architecture, prototype again, and test again. This iterative process should eventually result in a deployment architecture that is ready for deployment in a production environment.

Your trial deployment should include a rollback plan, in case the deployment fails or runs into serious problems. As part of this plan, consider:

- A backup procedure and schedule for your calendar data. For example, you might want to backup your calendar data every night and save the data for a week. See the *Sun Java System Calendar Server Administration Guide* for more information:

  `http://docs.sun.com/doc/817-5697`

- Criteria for initiating the rollback plan. How severe must your problems be before you initiate your rollback plan? For example, if your database becomes corrupted and you cannot recover it.

Rolling Out the Production System

In the production rollout phase, you implement your deployment architecture in a production environment. This phase involves installing, configuring, and starting up distributed applications and any required infrastructure services in a production environment. You normally start with a limited deployment and move to organization-wide implementation. In this process, you perform trial runs, in which you apply increasing loads and stress test the system.

As part of the rollout phase you might need to perform administrative tasks such as provisioning users, implementing single sign-on, and tuning the system to meet performance objectives. Verifying the deployment and performing capacity planning are also part of this phase. Capacity planning, of which monitoring the system plays an important role, is necessary for meeting the long-term needs of system growth.

Chapter 2

Analyzing Your Requirements

Planning your Calendar Server deployment requires that you first analyze your organization's business and technical requirements. This chapter helps you to gather and assess your requirements, which you then use to determine your Calendar Server architecture.

This chapter contains the following sections:

Identifying Deployment Goals

Before you purchase or deploy Calendar Server software or hardware, you need to identify your deployment goals. Deployment requirements can come from various sources within an organization. In many cases, requirements are expressed in vague terms, requiring you to clarify them towards determining a specific goal.

The outcome of your requirements analysis should be a clear, succinct, and measurable set of goals by which to gauge the deployment's success. Proceeding without clear goals accepted by the stake holders of the project is precarious at best.

Some of the requirements you need to examine before you can plan your deployment include:

- Business requirements
- Technical requirements
- Financial requirements

- Service Level Agreements (SLAs)

Business Requirements

Your business objectives affect deployment decisions. Specifically, you need to understand your users' behavior, your site distribution, and the potential political issues that could affect your deployment. If you do not understand these business requirements, you can easily make wrong assumptions that affect the accuracy of your deployment design.

Operational Requirements

Express operational requirements as a set of functional requirements with straightforward goals. Typically, you might come across informal specifications for:

- End-user functionality
- End-user response times
- Availability / uptime
- Information archival and retention

For example, translate a requirement for "adequate end-user response time" into measurable terms such that all stake holders understand what is "adequate" and how the response time is measured.

Culture and Politics

A deployment needs to take into account your corporate culture and politics. Demands can arise from areas that end up representing a business requirement. For example:

- Some sites might require their own management of the deployed solution. Such demands can raise the project's training costs, complexities, and so forth.
- Given that the LDAP directory contains personnel data, the Human Resources department might want to own and control the directory.

Technical Requirements

Technical requirements (or functional requirements) are the details of your organization's system needs.

Supporting Existing Usage Patterns

Express existing usage patterns as clearly measurable goals for the deployment to achieve. The following questions will help you determine such goals.

- How are current services utilized?
- Can your users be categorized (for example, as sporadic, frequent, or heavy users)?
- What size messages do users commonly send?
- How many invites are usually on calendar appointments?
- How many messages do users send?
- How many calendar events and tasks do users typically create per day or per hour?
- To which sites in your company do your users send messages?

Study the users who will access your services. Factors such as when they will use existing services are keys to identifying your deployment requirements and therefore goals. If your organization's experience cannot provide these patterns, study the experience of other organizations to estimate your own.

Regions in organizations that have heavy usage might need their own servers. Generally, if your users are far away from the actual servers, they will experience slower response times. Consider whether the response times will be acceptable.

Site Distribution

Use these questions to understand how site distribution impacts your deployment goals:

- How are your sites geographically distributed?
- What is the bandwidth between the sites? Centralized approaches will require greater bandwidth than de-centralized. Mission critical sites might need their own servers.

Network

The following questions help you understand your network requirements:

- Do you want to obfuscate internal network information?
- Do you want to provide redundancy of network services?
- Do you want to limit available data on access layer hosts?

- Do you want to simplify end-user settings, for example, have end users enter a single mail host that does not have to change?
- Do you want to reduce network HTTP traffic?

NOTE Answering yes to these questions suggests a two-tier architecture.

Existing Infrastructure

You might be able to centralize servers if you have more reliable and higher available bandwidth.

- Will the existing infrastructure and facilities prove adequate to enable this deployment?
- Can the DNS server cope with the extra load? Directory server? Network? Routers? Switches? Firewall?

Support Personnel

24-hour, seven-day-a-week (24 x 7) support might only be available at certain sites. A simpler architecture with fewer servers will be easier to support.

- Is there sufficient capacity in operations and technical support groups to facilitate this deployment?
- Can operations and technical support groups cope with the increased load during deployment phase?

Financial Requirements

Financial restrictions impact how you construct your deployment. Financial requirements tend to be clearly defined from an overall perspective providing a limit or target of the deployment.

Beyond the obvious hardware, software, and maintenance costs, a number of other costs can impact the overall project cost, including:

- Training
- Upgrade of other services and facilities, for example, network bandwidth or routers
- Deployment costs, such as personnel and resources required to prove the deployment concept

- Operational costs, such as personnel to administer the deployed solution

You can avoid financial issues associated with the project by applying sufficient attention and analysis to the many factors associated with the project requirements.

Service Level Agreements (SLAs)

You should develop SLAs for your deployment around such areas as uptime, response time, message delivery time, and disaster recovery. An SLA itself should account for such items as an overview of the system, the roles and responsibilities of support organizations, response times, how to measure service levels, change requests, and so forth.

Identifying your organization's expectations around system availability is key in determining the scope of your SLAs. System availability is often expressed as a percentage of the system uptime. A basic equation to calculate system availability is:

```
Availability = uptime / (uptime + downtime) * 100
```

For instance, a service level agreement uptime of four nines (99.99 percent) means that in a month the system can be unavailable for about four minutes.

Furthermore, system downtime is the total time the system is not available for use. This total includes not only unplanned downtime, such as hardware failures and network outages, but also planned downtime, preventive maintenance, software upgrade, patches, and so on. If the system is supposed to be available 7x24 (seven days a week, 24 hours a day), the architecture needs to include redundancy to avoid planned and unplanned downtime to ensure high availability.

Determining Project Goals

Your investigation and analysis should reveal your project's requirements. Next, you should be able to determine a clearly measurable set of goals. Specify these goals in such a manner that personnel not directly associated with the project can understand the goals and how to measure the project against them.

Stake holders need to accept the project goals. The projects goals need to be measured in a post-implementation review to determine the success of the project.

Planning for Growth

In addition to determining what capacity you need today, assess what capacity you need in the future, within a timeframe that you can plan for. Typically, a growth timeline is in the range of six to twelve months. Growth expectations and changes in usage characteristics are factors that you need to take into account to accommodate growth.

As the number of users and messages increase, you should outline successful guidelines for capacity planning. You need to plan for increases in message traffic for the various servers, a larger volume of users, larger mailbox sizes, and so forth. As growth occurs in the user population, usage characteristics change over time. Your deployment goals (and therefore deployment design) must respond accordingly to be viable into the future.

Ideally, you should design your architecture to easily accommodate future growth. Monitoring the deployment, once it enters its production phase, is also crucial to being able to understand when and by how much a deployment needs to grow.

Understanding Total Cost of Ownership

Total Cost of Ownership (TCO) is another factor that affects capacity planning. This includes choosing the hardware upon which to deploy your Calendar Server. Table 2-1 on page 31 presents some factors to consider as to whether to deploy more smaller hardware systems or fewer larger hardware systems.

Table 2-1 Considerations for Total Cost of Ownership

Hardware Choice	Pros	Cons
More, smaller hardware systems	• Smaller hardware systems generally cost less. • More, smaller hardware systems can be deployed across many locations to support a distributed business environment. • More, smaller hardware systems can mean less down time for system maintenance, upgrade, and migration because traffic can be routed to other servers that are still online while others are being maintained.	• Smaller hardware systems have a more limited capacity, so more of them are needed. Management, administration, and maintenance costs go up as the number of hardware systems goes up. • More, smaller hardware systems require more system maintenance because there are more of them to maintain.
Fewer, larger hardware systems	• Fewer hardware systems means fewer fixed management costs per server. If your management costs are a recurring monthly bill, whether internal or from an ISP, costs will be lower, because you have fewer hardware systems to manage. • Fewer hardware systems can also mean easier system maintenance, upgrade, and migration because there are fewer systems to maintain.	• Larger hardware systems generally cost more initially. • Fewer hardware systems can also mean a greater system down-time for maintenance, upgrade and migration. • Require specially trained system administrators.

Chapter 3

Determining Your Network Infrastructure Needs

Your network infrastructure is the underlying foundation of the system. It forms the services that create the operating makeup of your network. In a Calendar Server deployment, determining your network infrastructure from the project goals ensures that you will have an architecture that can scale and grow.

This chapter contains the following sections:

Understanding Your Existing Network

You need to understand your existing network infrastructure to determine how well it can meet the needs of your deployment goals. By examining your existing infrastructure, you identify if you need to upgrade existing network components or purchase new network components. You should build up a complete map of the existing network by covering these areas:

1. Physical communication links, such as cable length, grade, and so forth
2. Communication links, such as analog, ISDN, VPN, T3, and so forth, and available bandwidth and latency between sites
3. Server information, including:
    - Host names
    - IP addresses

- Domain Name System (DNS) server for domain membership

4. Locations of devices on your network, including:
   - Hubs
   - Switches
   - Modems
   - Routers and bridges
   - Proxy servers
5. Number of users at each site, including mobile users

After completing this inventory, you need to review that information in conjunction with your project goals to determine what changes are required so that you can successfully deliver the deployment.

Understanding Network Infrastructure Components

The following common network infrastructure components have a direct impact upon the success of your deployment:

- Routers and switches
- Firewalls
- Load balancers
- Storage Area Network (SAN)
- DNS

Routers and Switches

Routers connect networks of your infrastructure, enabling systems to communicate. You need to ensure that the routers have spare capacity after the deployment to cope with projected growth and usage.

In a similar vein, switches connect systems within a network.

Routers or switches running at capacity tend to induce escalating bottlenecks, which result in significantly longer times for clients to submit messages to servers on different networks. In such cases, the lack of foresight or expenditure to upgrade the router or switch could have a personnel productivity impact far greater than the cost.

Firewalls

Firewalls sit between a router and application servers to provide access control. Firewalls were originally used to protect a trusted network (yours) from the untrusted network (the Internet). These days, it is becoming more common to protect application servers on their own (trusted, isolated) network from the untrusted networks (your network and the Internet).

Router configurations add to the collective firewall capability by screening the data presented to the firewall. Router configurations can potentially block undesired services (such as NFS, NIS, and so forth) and use packet-level filtering to block traffic from untrusted hosts or networks.

In addition, when installing a Sun server in an environment that is exposed to the Internet, or any untrusted network, reduce the Solaris installation to the minimum number of packages necessary to support the applications to be hosted. Achieving minimization in services, libraries, and applications helps increase security by reducing the number of subsystems that must be maintained. The Solaris™ Security Toolkit provides a flexible and extensible mechanism to minimize, harden, and secure Solaris Operating Environment systems.

Your Site Security Policy should provide direction on such issues.

Load Balancers

Use load balancers to distribute overall load on your Web or application servers, or to distribute demand according to the kind of task to be performed. If, for example, you have a variety of dedicated applications and hence different application servers, you might use load balancers according to the kind of application the user requests.

If you have multiple data centers, you should consider geographic load balancing. Geographic load balancing distributes load according to demand, site capacity, and closest location to the user. If one center should go down, the geographic load balancer provides failover ability.

For load balancers on Web farms, place the hardware load balancers in front of the servers and behind routers because they direct routed traffic to appropriate servers. Software load balancing solutions reside on the Web servers themselves. With software solutions, one of the servers typically acts a traffic scheduler.

A load balancing solution is able to read headers and contents of incoming packets. This enables you to balance load by the kind of information within the packet, including the user and the type of request. A load balancing solution that reads packet headers enables you to identify privileged users and to direct requests to servers handling specific tasks.

You need to investigate how dynamically the load balancer communicates with all the servers it caters to. Does the scheduler ping each server or create “live” agents that reside on the servers to ascertain load data? You should also examine how the load balancer parses TCP packets. Pay attention to how quickly the load balancer can process a packet. Some load balancers will be more efficient than others. Load balancer efficiency is typically measured in throughput.

Storage Area Networks (SANs)

Understanding the data requirements of the storage system is necessary for a successful deployment. Increasingly, SANs are being deployed so that the storage is independent of the servers used in conjunction with it. Deploying SANs can represent a decrease in the time to recover from a non-functional server as the machine can be replaced without having to relocate the storage drives.

Use these questions to evaluate if your deployment storage requirements would be best served through a SAN:

- Are reads or writes more prevalent?
- Do you need high I/O rate storage? Is striping the best option?
- Do you need high uptime? Is mirroring the best option?
- How is the data to be backed up? When is it going to be backed up?

DNS

Servers which make heavy usage of DNS queries should be equipped with a local caching DNS server to reduce lookup latency as well as network traffic.

When determining your requirements, consider allocating host names for functions such as mailstore, mail-relay-in, mail-relay-out, and so forth. You should consider this policy even if the host names all are currently hosted on one machine. With services configured in such a way, relocation of the services to alternate hardware significantly reduces the impacts of the change.

Planning Your Network Infrastructure Layout

In deriving your infrastructure topology, you need to consider the following perspectives:

- DMZ
- Intranet
- Internal Network
- Proxies

Demilitarized Zone (DMZ)

These days, most company networks are configured for a DMZ. The DMZ separates the corporate network from the Internet. The DMZ is a tightly secured area into which you place servers providing Internet services and facilities (for example, web servers). These machines are hardened to withstand the attacks they might face. To limit exposure in case of a security breach from such attacks, these servers typically contain no information about the internal network. For example, the nameserver facilities only include the server and the routers to the Internet.

Progressively, DMZ implementations have moved the segment behind the firewall as firewall security and facilities have increased in robustness. However, the DMZ still remains segmented from the internal networks. You should continue to locate all machines hosting Web servers, FTP servers, mail servers, and external DNS on a DMZ segment.

A simpler network design might only define separate DMZ segments for Internet services, VPN access, and remote access. However, security issues exist with VPN and remote access traffic. You need to separate appropriate connections of these types from the rest of the network.

The firewall providing the DMZ segmentation should allow only inbound packets destined to the corresponding service ports and hosts offering the services within the DMZ. Also, limit outbound initiated traffic to the Internet to those machines requiring access to the Internet to carry out the service they are providing (for example, DNS and mail). You might want to segment an inbound-only DMZ and an outbound-only DMZ, with respect to the type of connection requests. However, given the potential of a denial-of-service attack interrupting DNS or email, consider creating separate inbound and outbound servers to provide these services. Should an email-based Trojan horse or worm get out of control and overrun your outbound mail server, inbound email can still be received. Apply the same approach to DNS servers.

Intranet

The DMZ provides a network segment for hosts that offer services to the Internet. This design protects your internal hosts, as they do not reside on the same segment as hosts that could be compromised by an external attack. Internally, you also have similar services to offer (Web, mail, file serving, internal DNS, and so on) that are meant solely for internal users. Just as the Internet services are segmented, so too, are the internal services. Separation of services in this manner also permits tighter controls to be placed on the router filtering.

Just as you separate the Internet-facing services into the DMZ for security, your private internal services should reside in their own internal DMZ.

Just as multiple DMZs can be beneficial—depending on your services and your network's size—multiple intranets might also be helpful.

The firewall rules providing the segmentation should be configured similarly to the rules used for the DMZ's firewall. Inbound traffic should come solely from machines relaying information from the DMZ (such as inbound email being passed to internal mail servers) and machines residing on the internal network.

Internal Network

The segments that remain make up your internal network segments. These segments house users' machines or departmental workstations. These machines request information from hosts residing on the intranet. Development, lab, and test network segments are also included in this list. Use a firewall between each internal network segment to filter traffic to provide additional security between departments. Identify the type of internal network traffic and services used on each of these segments to determine if an internal firewall would be beneficial.

These machines should not communicate directly with machines on the Internet. Preferably, these machines avoid direct communication with machines in the DMZ. Ultimately, the services they require should reside on hosts in the intranet. A host on the intranet can in turn communicate with a host in the DMZ to complete a service (such as outbound email or DNS). This indirect communication is acceptable.

Proxies

Only the machines directly communicating with machines on the Internet should reside in the DMZ. If users require Internet access, though, this creates a problem based on your previous topology decisions. In this situation, proxies become helpful. Place a proxy on an internal network segment, or, better yet, an intranet segment. A machine requiring access to the Internet can pass its request onto the proxy, which in turn makes the request on the machine's behalf. This relay out to the Internet helps shield the machine from any potential danger it might encounter.

Because the proxy communicates directly with machines on the Internet, it should reside in the DMZ. However, this conflicts with the desire to prevent internal machines from directly communicating with DMZ machines. To keep this communication indirect, use a double proxy system. A second proxy residing in the intranet passes connection requests of the internal machines to the proxy in the DMZ, which in turn makes the actual connection out on the Internet.

Firewall Configuration

In addition to the typical packet-filtering features, most firewalls provide features to prevent IP spoofing. Use IP-spoofing protection whenever possible.

For instance, if there is only one entry point into your network from the Internet and a packet is received from the Internet with a source address of one of your internal machines, it was likely spoofed. Based on your network's topology, the only packets containing a source IP address from your internal machines should come from within the network itself, not from the Internet. By preventing IP spoofing, this possibility is eliminated, and the potential for bypassing IP address-based authorization and the other firewall-filtering rules is reduced. Use the same IP-spoofing protection on any internal firewall as well.

Mobile Users

When you have remote or mobile users, pay attention to how you will provide them access to the facilities. Will there be any facilities they cannot access? What kind of security policies do you need to address? Will you require SSL for authentication? Also, examine whether your mobile user population is stable or is expected to increase over time.

Chapter 4

Planning Your Calendar Server Configuration

This chapter describes the three basic Calendar Server configurations, which can vary depending on your site's specific requirements.

This chapter contains the following sections:

Calendar Server Considerations

Calendar Server consists of five major services:

- HTTP Service (`cshttpd`) listens for HTTP requests. It receives user requests and returns data to the caller.
- Administration Service (`csadmind`) is required for each instance of Calendar Server. It provides a single point of authentication and administration for the Calendar Servers and provides most of the administration tools.
- Notification Service (`csnotify`) sends notifications of events and to-dos using either email or the Event Notification Service.
- Event Notification Service (`enpd`) acts as the broker for event alarms.
- Distributed Database Service (`csdwpd`) links multiple database servers together within the same Calendar Server system to form a distributed calendar store.

In a scalable Calendar Server deployment, you deploy an instance of HTTP Service and Administration Service together as a Calendar front-end system. (You would deploy one instance of the `cshttpd` per machine. On each machine, you would configure one `cshttpd` process per CPU on that machine.) An instance of Notification Service, Event Notification Service, Distributed Database Service and Administration Service are deployed together as the Calendar back-end system.

Authentication and XML / XSLT transformation are two Calendar Service activities that generate heavy load. Additional CPUs can be added to meet quality of service requirements.

Calendar back-end services usually require half the number of CPUs sized for the Calendar front-end services. To support quality of service by the Calendar front-end system, the Calendar back-end system should use around two-thirds of the front-end CPUs.

You will want to consider early on in a deployment separating the Calendar Service into front-end and back-end services.

The Calendar Server HTTP process that is typically a component of the front-end services is a dominant user of CPU time. This suggests care should be taken in accounting for peak calendar usage and choosing sufficient front-end processing power to accommodate the expected peak HTTP sessions. Typically, you would make the Calendar Server front end more available through redundancy, that is, by deploying multiple front-end hosts. As the front-end systems do not maintain any persistent calendar data, they are not good candidates for HA solutions like Sun Cluster. Moreover, the additional hardware and administrative overhead of such solutions make deploying HA for Calendar Server front ends both expensive and time-consuming.

NOTE The only configuration for Calendar front ends that might warrant a true HA solution is where you have deployed the Calendar front end on the same host that contains a Messaging Server MTA router. Even in this configuration, however, the overhead of such a solution should be carefully weighed against the slight benefit.

A good choice of hardware for the Calendar Server front ends is a single or dual processor SPARC or Intel server. You would deploy one instance of the Calendar Server `cshttpd` process per machine. Such a deployment affords a cost-effective solution, enabling you to start with some level of initial client concurrency capability and add client session capacity as you discover peak usage levels on your existing configuration.

When you deploy multiple front ends, a load balancer (with sticky / persistent connections) is necessary to distribute the load across the front-end services.

The Calendar Server back-end services are well balanced in resource consumption and show no evidence of bottleneck formation either in CPU or I / O (disk or network). Thus, a good choice of hardware for the back end would be a SPARC server with a single striped volume. Such a machine presents considerable capacity for large-peak calendar loads.

If your requirements include high availability, it makes sense to deploy the Calendar Server back end with Sun Cluster, as the back end does contain persistent data.

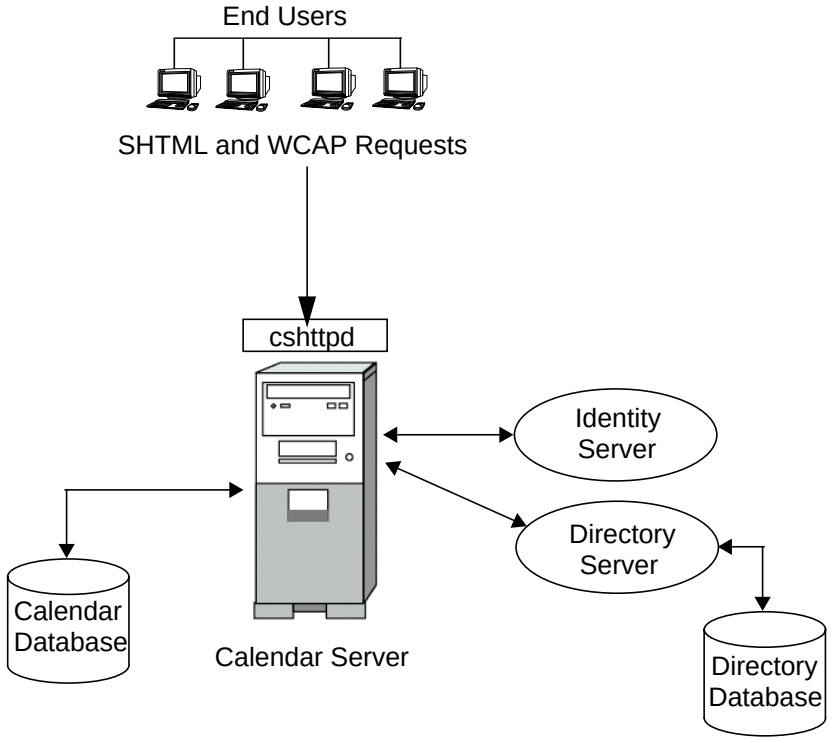
NOTE In a configuration with both front-end and back-end Calender Server hosts, all hosts must be running:

- The same operating system
- The same releases of Calendar Server, including patch or hotfix releases.

Single-Server Minimal Configuration

In a single-server minimal configuration (shown in Figure 4-1 on page 44), all Calendar Server services (processes) run on the same server, either in the same CPU (processor) or across multiple CPUs. The directory server and Sun Java System Identity Server processes can run on the same server or on different servers.

Figure 4-1 Single-Server Minimal Calendar Server Configuration



A Calendar Server instance on a single server includes the following services:

- Administration service (`csadmind` process) provides support for administration functions such as commands to start or stop Calendar Server, create or delete calendar users or resources, or fetch and store calendars.
- HTTP service (`cshttpd` process) handles incoming SHTML and WCAP requests.

For a description of Calendar Server services, see the *Sun Java System Calendar Server Administration Guide*.

The Database Wire Protocol (DWP) service (`csdwpd` process), which provides networking capability when the calendar database is on another server, is not required for a minimal configuration because the database is on the same server.

Calendar Server requires a directory server to authenticate users and to store user preferences. Usually, the directory server is an LDAP directory server, such as Sun Java System Directory Server. However, if you prefer, you can use the Calendar Server API (CSAPI) to write a plug-in to use a non-LDAP directory server. This API is described in the *Sun Java System Calendar Server Developer's Guide*.

The directory server can run on the same server where Calendar Server is running or on a remote server.

Sun Java System Identity Server (release 2003Q4 (6.1) or later) provides the following functionality:

- **`commadmin` utility.** Use this CLI utility to provision and manage hosted (virtual) domains, users, groups, organizations, resources, and roles for Sun Java System communications servers, including Calendar Server.

  For information about the `commadmin` utility, see the *Sun Java System Communications Services User Management Utility Administration Guide*.

- **Single Sign-on (SSO).** You can implement SSO for Sun Java Enterprise System servers, including Calendar Server and Messaging Server, using Identity Server, or through trusted circle technology. Identity Server serves as the SSO gateway for the Sun Java Enterprise System servers. Users log in to Identity Server and then can access other the servers, as long as all servers are configured properly for SSO.

- **Sun Java System LDAP Schema 2.** Identity Server (release 2003Q4 or later) is required if you want to use this version of the schema.

For more information on the above topics, see the *Sun Java System Calendar Server Administration Guide*.

Identity Server can run on the same server where Calendar Server is running or on a remote server.

End users connect to Calendar Server from client machines by using one of the two Web user interfaces (UIs), that is, either Sun Java System Calendar Express, or Sun Java System Communications Express. For information on either user interface, refer to the respective interface's online Help. Also, see the *Sun Java System Communications Express Administration Guide*:

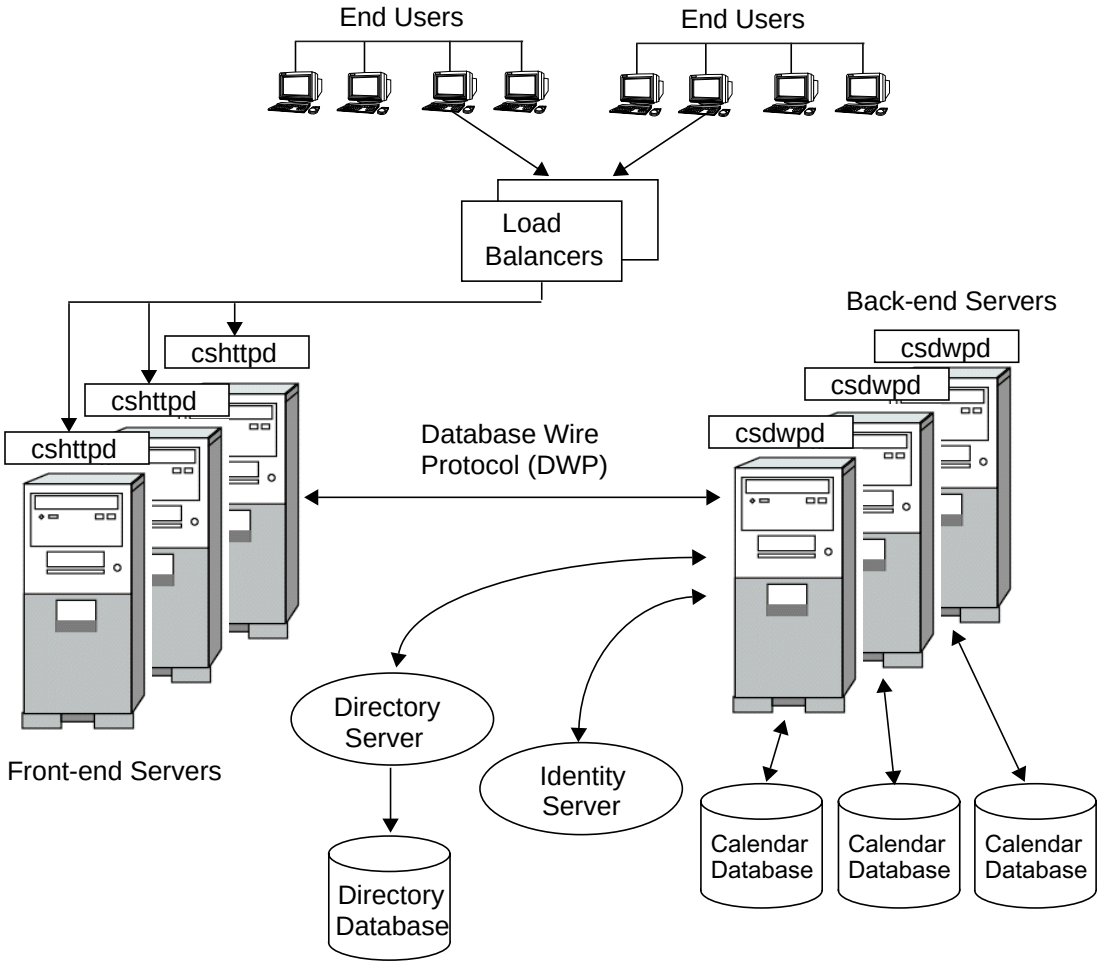
`http://docs.sun.com/doc/817-5416`

Network Front-end/Database Back-end Server Configuration

Calendar Server supports scalability by distributing a configuration over multiple front-end and back-end servers. On each server, Calendar Server services (processes or daemons) can also be distributed across multiple CPUs (or processors).

In network front-end / database back-end configuration (shown in Figure 4-2 on page 47), users log in to a front-end server and connect to a back-end server using the Database Wire Protocol (DWP) service (`csdwpd` process). The calendar database is connected only to the back-end servers.

Figure 4-2 Network Front-end / Database Back-end Server Configuration



Calendar Server processes run on both the front-end and back-end servers as follows:

- Users are directed by load balancers to a front-end server, where they log in. Each front-end server requires these services:
  - Administration Service (`csadmind` process)
  - HTTP Service (`cshttpd` process)

- Each back-end server is connected to a calendar database, so each back-end server requires these services:
  - Administration Service (`csadmind` process)
  - Event Notification Service (`enpd` and `csnotifyd` processes)
  - Database Wire Protocol (DWP) Service (`csdwpd` process) to provide networking capability to the front-end servers for the calendar database

In this configuration, users do not log in to the back-end servers, so the HTTP service (`cshttpd` process) is not required.

For a description of Calendar Server services, see the *Sun Java System Calendar Server Administration Guide*.

A scalable Calendar Server configuration requires a directory server to authenticate users and to store user preferences.

You can use Identity Server (release 6.1 (release 6 2003Q4) or later) to implement Single Sign-on (SSO), to use Sun Java Enterprise System LDAP Schema 2, or to provision and manage hosted (virtual) domains, users, groups, organizations, resources, and roles.

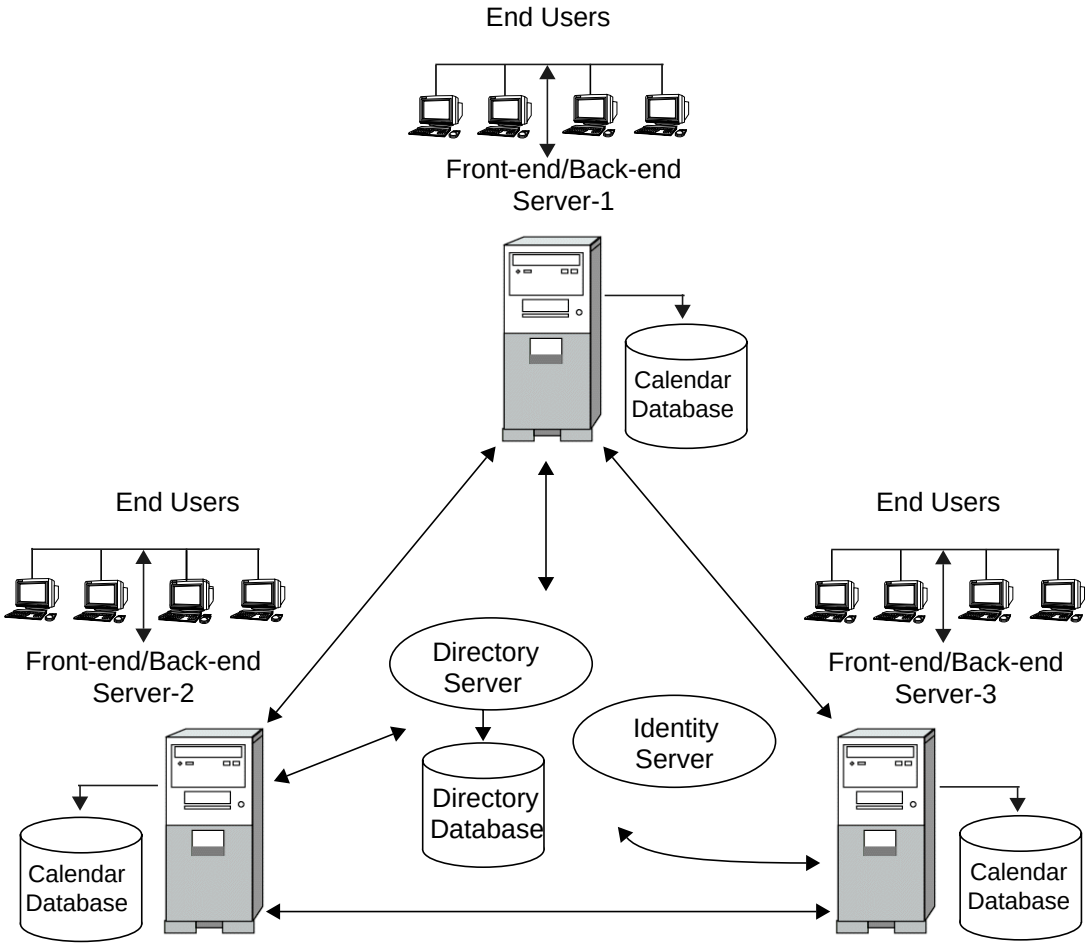
End users connect to Calendar Server from client machines by using one of the two Web user interfaces (UIs), that is, either Sun Java System Calendar Express, or Sun Java System Communications Express. For information on either user interface, refer to the respective interface's online Help. Also, see the *Sun Java System Communications Express Administration Guide*:

`http://docs.sun.com/doc/817-5416`

Multiple Front-end/Back-end Server Configuration

In a multiple front-end / back-end server configuration (shown in Figure 4-3 on page 49), users log in to a specific server, and each server is connected to a calendar database. This configuration allows calendars to be geographically distributed, with each calendar residing on the server where its owner logs in to Calendar Server.

Figure 4-3 Multiple Front-end / Back-end Server Configuration



Each front end and back end requires all Calendar Server services: Administration Service (`csadmind` process), HTTP Service (`cshttpd` process), Event Notification Service (`enpd` and `csnotifyd` processes), and Database Wire Protocol (DWP) Service (`csdwpd` process).

For a description of Calendar Server services, see the *Sun Java System Calendar Server Administration Guide*.

A multiple front-end / back-end server configuration requires a directory server to authenticate users and to store user preferences.

You can use Identity Server (release 6.1 (release 6 2003Q4) or later) to implement Single Sign-on (SSO), to use Sun Java Enterprise System LDAP Schema 2, or to provision and manage hosted (virtual) domains, users, groups, organizations, resources, and roles.

End users connect to Calendar Server from client machines by using one of the two Web user interfaces (UIs), that is, either Sun Java System Calendar Express, or Sun Java System Communications Express. For information on either user interface, refer to the respective interface's online Help. Also, see the *Sun Java System Communications Express Administration Guide*:

http://docs.sun.com/doc/817-5416

Chapter 5

Understanding Calendar Server Schema and Provisioning Options

This chapter describes the schema and provisioning options for Calendar Server. Because of the complexity in provisioning Calendar Server, you need to understand your options before installing the product.

This chapter contains the following sections:

Understanding Calendar Schema Choices

This section describes the two schema options that are available and supported with Calendar Server, and how to decide which to use.

NOTE Refer to the *Sun Java System Communications Services Schema Migration Guide* for information on how to migrate from Sun Java System LDAP Schema version 1 to Sun Java System LDAP Schema version 2.

Support for installation and provisioning of Schema 1 will be deprecated and removed from future releases. However, customers with their own provisioning tools may continue to use LDAP Schema 1.

Deciding Which Schema to Use

Choosing the schema that's right for your installation depends on your provisioning needs:

- Are you integrating Calendar Server with other Java Enterprise System component products, such as Sun Java System Portal Server or Sun Java System Identity Server, which provides single sign-on capabilities?

  If you answer Yes, then you must use Schema 2 Native Mode

- Are you installing Calendar Server for the first time or are you upgrading from an older version?

  If you are installing Calendar Server for the first time, use Schema 2 Native Mode

- Do you plan to use either Sun Java System Identity Server CLI utilities for provisioning or single sign-on?

  If you answer Yes, use Schema 2 Native or Compatibility Mode.

- If you are upgrading from an older version of Calendar Server, you can either use Schema 1 or Schema 2 Native or Compatibility Mode

- Do you want to use the Calendar Server `csdomain` utility for provisioning domains?

  If you answer Yes, use Schema 2 Native or Compatibility Mode. If you don't plan to use Identity Server 6.1 features or integrate Calendar Server with other Java Enterprise System products, use Schema 1.

- If you don't want to use either Sun Java System Identity Server or Calendar Server CLI utilities for provisioning, use can use either Schema 2 Native Mode for new installations, or Schema 1 or Schema 2 Compatibility Mode for existing Calendar Server installations.

LDAP Schema 1

LDAP Schema 1 is a provisioning schema that consists of both an Organization Tree and a DC Tree. This set of schema (at the time, it was simply called "schema") was supported in previous Calendar Server 5.x versions.

When Calendar Server searches for user or group entries, it looks at the user's or group's domain node in the DC Tree and extracts the value of the `inetDomainBaseDN` attribute. This attribute holds a DN reference to the organization subtree containing the actual user or group entry.

Only sites that have installed previous versions of Calendar Server should use Schema 1.

NOTE Migrating to Schema 2 is imperative if you plan to install Calendar Server with other Sun Java System products in the future.

Supported Provisioning Tools

Schema 1 supports LDAP provisioning tools. For more information, see "Understanding Calendar Server Provisioning Tools" on page 54.

Schema 2 (Native Mode)

Schema 2 is a newly defined set of provisioning definitions that describes the types of information that can be stored as entries by using the Directory Server LDAP.

The native mode uses search templates to search the LDAP directory server. Once the domain is found by using the domain search template, the user or group search templates are used to find a specific user or group.

You should use native mode if you are installing Calendar Server for the first time and you do not have other applications on your machine that are dependent on a two-tree provisioning model. You should also use this mode if you want to install other products in the Java Enterprise System product suite.

If you have an existing Calendar Server 5.x installation that uses Schema 1, and you want to integrate Calendar Server with other Java Enterprise Server products, you should migrate your directory to Schema 2 after you upgrade to Calendar Server 6. Refer to the *Sun Java System Communications Services Schema Migration Guide* for information on how to migrate from LDAP Schema version 1 to LDAP Schema version 2.

NOTE Schema 2 Native Mode is the recommended provisioning model for all Sun Java System products in the Java Enterprise System product suite.

Supported Provisioning Tools

Schema 2 supports Sun Java System Communications Services User Management Utility. For more information, see "Understanding Calendar Server Provisioning Tools" on page 54.

Schema 2 Compatibility Mode

Schema 2 compatibility mode is an interim mode between Schema 1 and Schema 2 native mode. Schema 2 compatibility mode supports both schemas and enables you to retain the existing two-tree design you already have. Schema 2 compatibility mode also assumes that you have installed Identity Server prior to installing Messaging Server.

Use Schema 2 Compatibility if you have existing applications that require Schema 1, but you also need functionality that requires Schema 2, for example, Identity Server, single sign-on, and so forth.

NOTE Schema 2 compatibility mode is provided as a convenience in migrating to the Schema 2 Native mode. Do not use Schema 2 compatibility mode as your final schema choice. The migration process from Schema 1 to Schema 2 compatibility mode and then finally to Schema 2 native mode is more complex that simply migrating from Schema 1 to Schema 2 native mode. See the *Sun Java System Communications Services Schema Migration Guide* for more information.

Understanding Calendar Server Provisioning Tools

Through supported Calendar Server provisioning tools, you can query, modify, add, or delete user, group, and domain entry information in your LDAP directory. This section examines these Calendar Server provisioning tools.

In addition to the questions asked in "Deciding Which Schema to Use" on page 52, you should use Table 5-1 on page 56 to evaluate your schema and provisioning tool options.

NOTE Prior to installing and configuring Calendar Server, you need to decide upon a schema model and tool or tools for provisioning your Calendar Server entries.

The following sections provide high-level information about the supported provisioning tools:

- LDAP Provisioning Tools

LDAP Provisioning Tools

Schema 1 users and groups can be provisioned using the LDAP Directory tools (Schema 2 is not supported). Unlike the Delegated Administrator graphical and command-line interfaces, you can directly provision users and groups by adding, removing, and modifying the LDIF records through LDAP without having to use a user interface.

User Management Utility

Sun Java System Identity Server uses Schema 2. Because the Sun Java System component products in the Java Enterprise System product suite use Schema 2, use the Communications Services 6 User Management Utility. This should particularly be the case if you are using more than one Java Enterprise System product, or if you are performing a brand new installation of Calendar Server.

NOTE Even though you install Identity Server, there is no graphical user interface compatibility with Calendar Server. Therefore, to provision users and groups with an interface, you can only use the user management utility.

See the *Sun Java System Communications Services User Management Utility Administration Guide* for installation details.

Comparing Your Provisioning Tool Options

Table 5-1 on page 56 shows the various supported schema, provisioning tools, provisioning limitations, and recommended documentation for additional information.

Table 5-1 Calendar Server Provisioning Mechanisms

Supported Provisioning Tool	Provisioning Tool Functionality	Provisioning Tool Limitations	For Further Information
LDAP Provisioning Tools Uses: Schema 1	Provides tools to directly modify LDAP entries or for creating custom provisioning tools.	• Incompatible with Sun Schema 2 and with other Java Enterprise System products.	Read the *Sun ONE Calendar Server 5.2 Provisioning Guide* and *Sun ONE Messaging and Collaboration Schema Reference Manual*. Describes the Sun LDAP Schema 1 provisioning model. In addition, these guides explain how to use LDAP provisioning tools and the usage of specific attributes and object classes.
User Management Utility Uses: Schema 2	Provides a command-line interface for administrators to manage users, groups, domains, and mailing lists. Compatible with other Java Enterprise System products.	• Not backwardly compatible with Sun Schema 1. • No GUI provisioning tool to use with Sun Java System Identity Server • Sun Java System Identity Server must be installed to enable this command-line interface.	Read the *Sun Java System Communications Services User Management Utility Administration Guide*. Provides syntax and usage for the command-line utility.

Chapter 6

Designing a Secure Calendar Server

This chapter provides an overview of security methods, describes common security threats, and outlines the steps in analyzing your security needs.

This chapter contains the following sections:

Creating a Security Strategy

Creating a security strategy is one of the most important steps in planning your deployment. Your strategy should meet your organization's security needs and provide a secure calendar environment without being overbearing to your users.

In addition, your security strategy needs to be simple enough to administer. A complex security strategy can lead to mistakes that prevent users from accessing their mail, or it can allow users and unauthorized intruders to modify or retrieve information that you don't want them to access.

RFC 2196, the *Site Security Handbook*, lists five steps to developing a security strategy:

1. Identify what you are trying to protect.

   For example, your list might include hardware, software, data, people, documentation, network infrastructure, or your organization's reputation.

2. Determine what you are trying to protect it from.

   For example: unauthorized access to calendars, events or tasks.

3. Estimate how likely threats are to your system.

   If you are a large Service Provider, your chances of security threats could be greater than a small organization. In addition, the nature of your organization could provoke security threats.

4. Implement measures that will protect your assets in a cost-effective manner.

   For example, the extra overhead in setting up an SSL connection can put a performance burden on your Calendar deployment. In designing your security strategy, you need to balance security needs against server capacity.

5. Continuously review your strategy and make improvements each time a weakness is found.

   Conduct regular audits to verify the efficiency of your overall security policy. You can do this by examining Calendar Server log files. For more information, refer to the *Sun Java System Calendar Server Administration Guide*.

   `http://docs.sun.com/doc/817-5697`

Your security strategy should also plan for:

- Physical Security
- Server Security

- Network Security
- Calendar Security Overview

Physical Security

Limit physical access to important parts of your infrastructure. For example, place physical limits on routers, servers, wiring closets, server rooms, or data centers to prevent theft, tampering, or other misuse. Network and server security become a moot point if any unauthorized person can walk into your server room an unplug your routers.

Server Security

Consider using security products (for example Solaris™ Security Toolkit) to harden the operating system. Also, remove services and facilities that you do not use (for example, telnet, ftp, and rlogin access to a server).

In a multi-tiered environment, configure back-end servers such that they will only provide the appropriate services to known front-end servers.

Limiting access to important operating system accounts and data is also part of any security strategy. Protection is achieved through the authentication and access control mechanisms available in the operating system.

In addition, you should install the most recent operating environment security patches and set up procedures to update the patches once every few months and in response to security alerts from the vendor.

Network Security

Limiting access to your network is an important part of your security strategy. Normally, overall access to networks is limited through the use of firewalls. However, email must be made available outside your site. SMTP is one such service.

To secure your network, you should:

- Turn off all operating system-provided services that listen on ports that you do not use.
- Replace `telnet` with `sshd`, if possible.

- Place your server behind a packet filter, which drops external packets with an internal source IP address. A packet filter forbids all connections from the outside except for those ports that you explicitly specify.

Calendar Security Overview

Security plays a key role in the day-to-day operations of today's enterprise. Breaches in security can not only compromise trade secrets, but can also result in downtime, data corruption, and increased operation costs. Calendar Server provides a number of security levels to protect users against eavesdropping, unsanctioned usage, or external attack. The basic level of security is through authentication. Calendar Server uses LDAP authentication by default, but also supports the use of an authentication plugin for cases where an alternate means of authentication is desired. Furthermore, integration with Identity Server enables Calendar Server to take advantage of its single sign-on capability.

Security involves not only ensuring the integrity of users. It also means ensuring the confidentiality of data. To this end, Calendar Server supports the use of SSL encryption for login, or both login and data. In other words, only the login may be encrypted, or the entire session including the login may be encrypted, from the Web client to the server.

Integration with the Java System Portal Server, Secure Remote Access product also provides SSL encryption, but through a proxy gateway. In addition, integration with the portal gateway provides a URL rewriting capability to further insulate Calendar Server from external entities. Calendar Server can be deployed with the portal gateway such that there is no direct connection to the Calendar Server without going through the gateway. In this case, every URL is rewritten, thus obfuscating the true URL of the Calendar Server. Even though a user is authenticated, that does not mean that the user should have access to other calendar users' data.

Within a calendar domain exist other layers of security to prevent authenticated users from unauthorized access to other authenticated users' calendar data. One security measure is through the Calendar Server access control entries. Access control enables calendar users to specify who can see their calendars, who can schedule events into their calendars, who can modify their calendars, and who can delete events from their calendars. Access control also enables a user to select who can act on his behalf to respond to invitations, schedule or modify events, and delete events. Finally, access control can be used to span domains of users, thus preventing (or enabling) users in one domain from scheduling events with users of another domain.

However, in addition to access control, Calendar Server provides an additional level of security at the database protocol level for deployments that separate the calendar front end from the database back end. This level of security is referred to as Database Wire Protocol (DWP) authentication, and utilizes a user name / password pair to authenticate a DWP connection. The userid / password paris on both the front end and database back end must be identical for a DWP connection to be authenticated.

Monitoring Your Security Strategy

Monitoring your server is an important part of your security strategy. To identify attacks on your system, monitor message queue size, CPU utilization, disk availability, and network utilization. Unusual growth in the message queue size or reduced server response time can identify some of these attacks. Also, investigate unusual system load patterns and unusual connections. Review logs on a daily basis for any unusual activity.

Planning User Authentication

User authentication enables your users to log in through their calendar clients to retrieve their calendar information. Methods for user authentication include:

- Plain Text and Encrypted Password Login
- Certificate-based Authentication with Secure Sockets Layer (SSL)

Plain Text and Encrypted Password Login

User IDs and passwords are stored in your LDAP directory. Password security criteria, such as minimum length, are determined by directory policy requirements. Password security criteria is not part of Calendar Server administration. To understand directory server password policies, see the *Sun Java System Directory Server Deployment Planning Guide*:

`http://docs.sun.com/doc/817-5218`

Both plain text and encrypted password login can be used.

Certificate-based Authentication with Secure Sockets Layer (SSL)

Calendar Server uses the SSL protocol for encrypted communications and for certificate-based authentication of clients and servers. This section describes certificate-based SSL authentication.

SSL is based on the concepts of public-key cryptography. Although TLS (Transport Layer Security) is functionally a superset of SSL, the names are used interchangeably.

At a high-level, a server which supports SSL needs to have a certificate, a public key, a private key, certificate, key, and security databases. This helps assure message authentication, privacy, and integrity.

To authenticate with SSL, the calendar client establishes an SSL session with the server and submits the user's certificate to the server. The server then evaluates if the submitted certificate is genuine. If the certificate is validated, the user is considered authenticated.

If you use SSL for authentication, you need to obtain a server certificate for your Calendar Server. The certificate identifies your server to clients and to other servers. Your server can have more than one server certificate with which it identifies itself. Your server can also have any number of certificates of trusted Certification Authorities (CAs) that it uses for client authentication.

For more information on SSL, see the *Sun Java System Calendar Server Administration Guide*:

`http://docs.sun.com/doc/817-5697`

Understanding Security Misconceptions

This section describes common messaging misconceptions that are counterproductive to the security needs of your deployment.

- **Hiding Product Names and Versions**

  At best, hiding product names and versions hinders casual attackers. At worst, it gives a false sense of security that might cause your administrators to become less diligent about tracking real security problems.

In fact, removing product information and version numbers makes it more difficult for the vendor support organization to validate software problems as being that of their software or that of other software.

A determined individual can use other protocol behaviors to determine the vendor name and version regardless of any attempt to hide it.

- **Hiding Names of Internal Machines**

  Hiding internal IP addresses and machine names will make it more difficult to:

  - Trace abuse or spam
  - Diagnose mail system configuration errors
  - Diagnose DNS configuration errors

  A determined attacker will have no problem discovering the machine names and IP addresses of machines once they find a way to compromise a network.

- **Network Address Translation (NAT)**

  If you use NAT to provide a type of firewall, you do not have an end-to-end connection between your systems. Instead, you have a third node which stands in the middle. This NAT system acts as a middleman, causing a potential security hole.

- **Calendar Server's Processes**

  It is standard practice to not run Calendar Server's processes as `root`.

Other Security Resources

For more information on designing a secure Messaging deployment, review the Computer Emergency Response Team (CERT) Coordination Center site:

`http://www.cert.org`

Chapter 7

Pre-installation Considerations

This chapter describes considerations you need to think about before installing Calendar Server.

This chapter contains the following sections:

Calendar Server Installation

The installation and configuration of Calendar Server has significantly changed from earlier Calendar Server releases (pre-2003Q4 versions). There is no longer a standalone installer for Calendar Server.

If you do not already have Calendar Server 2003Q4 (6.0) installed, you must use the Sun Java Enterprise System installer to get the 2004Q2 version. With this installer, you can also install other Sun component products and packages. For information about the Java Enterprise System installer, refer to the *Sun Java Enterprise System 2004Q2 Installation Guide*.

If you want to upgrade from Calendar Server 6 2003Q4 to Calendar Server 6 2004Q2, the upgrade process is described in “Upgrading from Java Enterprise System 2003Q4” in the *Sun Java Enterprise System 2004Q2 Installation Guide*.

For information about migrating from older versions of Calendar Server, up through version 5.x, see Migrations Utility chapter in the Calendar Server Administration Guide (covers up to 5.x). For migrating from versions later than 5.x, contact your Sun support representative.

Which Calendar Server Components to Configure?

When you install Calendar Server software, the Java Enterprise System installer installs all the Calendar Server packages. You then configure the appropriate Calendar Server component on a Calendar host through the Calendar Server configurator program.

The following table shows which components you need to configure for each type of Calendar host.

Table 7-1 Which Calendar Server Components to Configure?

Type of Calendar Host Being Configured	Needs These Components Selected in the Configurator Program
Front end	HTTP service(s) and Administration service
Back end	Notification Service, Event Notification Service, Distributed Database Service and Administration Service

The Distributed Database Service (`csdwpd`) is required only on back-end servers, that is, a server that has a calendar database, but does not provide user access services (`cshttpd`). It is not required on front-end servers that do not have a calendar database. The `csdwpd` service enables you to link front-end and back-end servers within the same Calendar Server configuration to form a distributed calendar store.

Planning for Calendar Server Administrators

Administrators for Calendar Server include:

- Calendar Server Administrator (calmaster)
- Calendar Server User and Group
- Superuser (root)

Calendar Server Administrator (calmaster)

The Calendar Server administrator is a specific user name with its associated password that can manage Calendar Server. For example, a Calendar Server administrator can start and stop Calendar Server services, add and delete users, create and delete calendars, and so on. This user has administrator privileges for Calendar Server but not necessarily for the directory server.

The default user ID for the Calendar Server administrator is `calmaster`, but you can specify a different user during Calendar Server configuration, if you prefer. After installation you can also specify a different user in the `service.admin.calmaster.userid` parameter in the `ics.conf` file.

The user ID you specify for the Calendar Server administrator must be a valid user account in your directory server. If the Calendar Server administrator user account does not exist in the directory server during configuration, the configuration program can create it for you.

See the CS AG for the complete list of Calendar Server administrator configuration parameters in the `ics.conf` file.

Calendar Server User and Group

On Solaris Operating Systems, these special accounts are the user ID and group ID under which Calendar Server runs. Sun recommends that you use the default values, `icsuser` and `icsgroup`, which are automatically created by the configuration program, if they do not exist. If you prefer, however, you can specify values other than `icsuser` and `icsgroup` when you run the Calendar Server configuration program. These values are stored in the `local.serveruid` and `local.servergid` parameters, respectively, in the `ics.conf` file.

Superuser (root)

On machines running the Solaris™ Operating System, you must log in as or become `superuser` (`root`) to install Calendar Server. You can also run as `superuser` to manage Calendar Server using the command-line utilities. For some tasks, however, you should run as `icsuser` and `icsgroup` (or the values you have selected) rather than `superuser` to avoid access problems for Calendar Server files.

Planning for Hosted Domains

Calendar Server supports hosted (or virtual) domains. In a hosted domain installation, each domain shares the same instance of Calendar Server, which enables multiple domains to exist on a single server. Each domain defines a name space within which all users, groups, and resources are unique. Each domain also has a set of attributes and preferences that you specifically set.

When installing and configuring hosted domains, use Schema 2 only.

Installing and configuring hosted domains on a server involves these high-level steps:

1. Installing and configuring Directory Server
2. Installing and configuring Web Server 6
3. Installing and configuring Identity Server

   The User Management Utility is installed with Identity Server.
4. Installing Calendar Server
5. Running the `comm_dssetup.pl` script

   For instructions on running this script, see Chapter 2 in the *Sun Java System Calendar Server Administration Guide* at:

   `http://docs.sun.com/coll/CalendarServer_04q2`
6. Configuring Communications Services User Management Utility

   For instructions on configuring and using `commadmin`, see the *Sun Java System Communications Services User Management Utility Administration Guide.*
7. Creating default domain and site administrator (`calmaster`)

   Note that the default domain is created when `commadmin` is configured, but the domain entry must be modified to add Calendar (or Mail) services. And, the site calendar administator (`calmaster`) must be set up. For instructions on how to perform these two tasks, see “Post Configuration Tasks” in the *Sun Java System Calendar Server Administration Guide.*
8. Configuring Calendar Server

   For instructions on running the `csconfiguratior.sh` program, see Chapter 3 in the *Sun Java System Calendar Serve Administration Guide.*

9. Setting hosted domain configuration parameters for Calendar Server

   For a list of the configuration parameters and their values, see "Hosted Domain Configuration Parameters" in the *Sun Java System Calendar Server Administration Guide*.

10. Creating the hosted domains for your site using `commadmin`.

11. Populating your hosted domains with users and resources using `commadmin`.

12. Starting Calendar Server services

    For instructions, see the *Sun Java System Calendar Server Administration Guide*.

See the appropriate core and component product documentation for more information on the preceding steps.

NOTE Always perform your provisioning for Schema 2 with the Communications Services User Management Utility interface.

Schema 1 provisioning tools do not support hosted domains.

Post-Installation Configuration

After you install Calendar Server 6 2004Q2, you must configure it. This step was previously performed as part of the installation process, but has now been separated out of the installer.

After you install Calendar Server, you must configure Calendar Server as follows:

1. Run the Directory Server Setup script (`comm_dssetup.pl`) to configure Sun Java System Directory Server 5.x.
2. Run the Calendar Server configuration program (`csconfigurator.sh`) to configure your site's specific requirements and to create a new `ics.conf` configuration file. For a description of the parameters in the `ics.conf` file, see the *Sun Java System Calendar Server Administration Guide*:

   `http://docs.sun.com/doc/817-5697`

Both `comm_dssetup.pl` and `csconfigurator.sh` are located in the `/opt/SUNWics5/cal/sbin` directory.

There are some configuration settings and changes that the Java Enterprise System installer and Calendar Server configuration utility (`csconfigurator.sh`) do not make. You must manually make changes to the following items:

- **DWP and CLD configurations.** Edit the `ics.conf` file so that the CLD cache option is enabled. This cache stores the DWP host server information for calendar users and thus reduces calls to the LDAP directory server.
- **Default Timezone.** If your default timezone is not Americas / New York, change it by editing the `ics.conf` file. You also need to change it in the `/opt/SUNWics5/cal/bin/html/default_user_prefs.xml` file so that it is in sync with the `ics.conf` file.
- **Client-side Rendering.** Calendar Server performs client-side rendering by downloading the XSLT processing to the end user's browser, which in turn reduces the processing that must be done by Calendar Server. Calendar Server downloads the XSLT processing only if the browser is capable of rendering the XSLT processing. In the current release, this applies only to Internet Explorer 6.0. Edit the `ics.conf` file to make this performance improvement to client-side rendering.
- **Setting for `tmpfs`.** Edit the `tmpfs` setting for performance enhancement.

For more information on these changes, see the *Sun Java System Calendar Server Administration Guide*:

`http://docs.sun.com/doc/817-5697`

Glossary

Refer to the Java Enterprise System Glossary (http://docs.sun.com/doc/816-6873) for a complete list of terms that are used in this guide.