

An Oracle White Paper
September 2011

Oracle Communications Calendar Server

Introduction	1
Oracle Communications Calendar Server Overview	3
What is the Oracle Communications Calendar Server?	3
Key Functionality	3
Calendar Server Interactions	4
Calendar Server Architectural Overview	7
Calendar Server Components	7
Protocol Subsystem	8
Core Subsystem	8
Database Subsystem	8
Distributed Horizontally Scalable Architecture	8
Calendar Server APIs	9
Java Message Service APIs	9
Proxy Authentication	10
Web Calendar Access Protocol (WCAP)	10
Advantages for Enterprises and Service Providers	10
Multiclient Access	11
Convergence	12
Outlook Connector	15
Other Clients	16
Security at Multiple Levels	16
High Performance, High Availability	17
Database Availability	18
Java Messaging Service (JMS)	19
Hosted Domains	20
Calendar Server Administration	20
The Command Line Interface	21
Administrative Functions	21
Delegated Administrator Functions	21
Migration	22
Conclusion	22
High-Performance Server	22
Based on Standards and Proven Technology	22
Flexibility and Extensibility	23
Feature-rich Server and Client	23
Appendix A	24
Abbreviation/Acronym List	24

Introduction

Organizations today require a dependable, scalable, cost-effective calendaring solution that integrates with other popular calendaring solutions as well as other communications components, such as messaging and address book. These organizations need a secure solution that is adaptable to their individual environments and highly available to ensure that they always have access to information about important appointments.

The Oracle Communications Calendar Server satisfies these requirements. It 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, telecommunications, and enterprise service provider. Through a rich integrated AJAX Web interface, standards-based clients, and connectors to other calendar clients (including Microsoft Outlook), the Oracle Communications Calendar Server provides group scheduling and personal calendaring to consumers at home or at work while also 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. This paper explores the product's architecture, design, deployment features, and benefits.

The Oracle Communications 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 can provide customers with the features they require at a lower total cost of ownership than alternative solutions. Native support for iCalendar standards allows users to schedule events in a format that is easily shared across the Internet, and support for CalDAV enables the use of multiple standards-based clients including Apple iCal, Apple iPhone, Mozilla Thunderbird/Lightning, and others. Oracle continues to lead in defining and implementing industry standards across its entire communications product line. The Oracle Communications Calendar Server employs standards and protocols such as:

- Calendaring Extensions to WebDAV (CalDAV)
- CalDAV Scheduling Extensions to WebDAV
- Collection Synchronization for WebDAV
- Internet Calendaring (iCalendar)
- iCalendar Transport-Independent Interoperability Protocol (iTIP)
- Calendar Message-based Interoperability Protocol (iMIP)
- Internet Calendar Scheduling Protocol (iSchedule)
- eXtensible Markup Language (XML)
- Lightweight Directory Access Protocol (LDAP)
- HyperText Transport Protocol (HTTP)

The Oracle Communications Calendar Server is a component of the Oracle Communications Unified Communications Suite, an open and integrated infrastructure software system that delivers industry-leading e-mail, calendaring, and real-time collaboration functionality for service providers and large organizations worldwide. The Oracle Communications 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). The Oracle Communications Calendar Server architecture is built upon robust, highly scalable components such as the Oracle Glassfish Server and the MySQL database, and thus is able to leverage the HA capabilities of the Application Server and the availability of standard tools for database backup, restore, and high availability. Simply put, the Oracle Communications Calendar Server provides a robust, scalable, and highly available platform that provides vast interoperability with a multitude of calendar clients

Oracle Communications Calendar Server Overview

What is the Oracle Communications Calendar Server?

The Oracle Communications Calendar Server is a high-performance, standards-based solution for time and resource management. It is designed with the scalability to meet the needs of Communication service providers (CSPs), telecommunications providers, and enterprise service providers as well as large enterprises. It provides group scheduling and personal calendaring to consumers at home or at work, and lets them share calendar information with others over the Internet. The Oracle Communications Calendar Server can provide customers with lower total cost of ownership because of its scalability, performance, and reliability.

Key Functionality

The Oracle Communications Calendar Server provides the following high-level functionality:

- A modular, pluggable architecture where components and services can be run on a single system or spread over multiple systems. The architecture is scalable vertically by CPU and horizontally by system, and provides client protocols for third-party client integration.
- Group scheduling as well as personal calendaring. Group scheduling includes free-busy lookup to find the next available free slot; individual and group attendees, invitations, and responses; and conflict handling in cases where an attendee's time may be double-booked. The Calendar Server supports implicit scheduling which processes incoming invitations and delivers them to the recipients' default calendars, for internal recipients with no extra client interaction. The Oracle Communications Calendar Server can also be used to schedule resources such as conference rooms and audio-visual equipment.
- Interoperability with a multitude of calendar clients including Thunderbird/Lightning, Apple iCal, Apple iPhone, and Microsoft Outlook, thereby enabling users of these products to perform calendaring and group scheduling with Oracle Communications Calendar Server. Interoperability with Outlook occurs via a connector. This is extremely useful for Outlook enterprise users in search of a lower cost calendaring solution that integrates into their existing environments.
- A rich and flexible calendar access control mechanism, and delegated calendar ownership. Each calendar has a primary owner, and can delegate to others who can act on behalf of the primary owner. These delegates can invite others to meetings or reply to invitations on behalf of the primary owner. A user can create different calendars, much like mail users can create different folders. Users can be secured for availability and schedule access while calendars can be secured for read, modify, or delete access. As a result, calendars can be made publicly accessible to individuals, groups, or everyone, or kept private solely for the owners. Privacy can also be designated at the event level. Individual events can be assigned public, private, or confidential access.
- Multiple calendars per user, allowing clients to do the layering of calendars onto other calendars to create composite calendar views, and daily, weekly, or monthly views.
- Includes the Convergence, a highly configurable, extensible front-end UI based on Asynchronous

JavaScript™ and XML (AJAX) technology. This AJAX Web client combines e-mail, calendaring, address book and instant messaging into a rich integrated Web experience. Single sign-on and tight integration enable users to utilize calendaring functionality from within mail and address book applications. Customization enables customers to specify UI Themes, to incorporate special branding, or to integrate third-party functionality (mashups).

- An integrated corporate address book that is used to search for potential attendees. The user can enter any part of an attendee's name and the address book searches through LDAP for appropriate matches. The user can then select the correct name and invite the user or group to the meeting.
- Flexible event notification and subscription service that enables processes to register and subscribe to events of interest and perform appropriate actions. An API enables developers to customize this notification capability to any type of notification, including Short Message Service (SMS) or pager notifications. There is also a server preference to enable or disable this feature.
- Event feeds can be imported and offered as a service to consumers. These event feeds can be produced by individuals or content providers. Consumers may subscribe to event feeds of interest, such as horoscopes or symphony calendars, and even purchase tickets.
- Supports the use of hosted domains. In a hosted-domain installation, each domain shares the same instance of the Oracle Communications Calendar Server, allowing a number of domains to exist on a single server.
- Command line administration enables administrators to create simple scripts to perform routine tasks such as calendar or entire database backups; creating, modifying, and deleting calendars; creating, modifying, and deleting users; and more. Administrators may also choose to use the Delegated Administrator graphical user interface to provision calendar users and resources.
- Standards-based design to enable interoperability with other standards-based calendar systems. This design provides iMIP interoperability with external calendaring systems.
- Employs an LDAP directory for user provisioning and storing user preferences. The Oracle Communications Calendar Server also supports the use of LDAP groups for both event invitations and calendar access control settings. This enables users to invite groups to events and to specify which groups can access a user's calendars.
- Natively supports SSL for secure communications between the client and server. The Calendar Server can also be configured to communicate with the Directory Server over SSL.
- Supports high availability through an architecture built upon the highly available Oracle Glassfish Server and the MySQL database.
- A set of migration tools are available for migrating the previous version of the Sun Calendar Server to Oracle Communications Calendar Server 7.
- Wireless access via the native CalDAV protocol and third-party solutions.

Calendar Server Interactions

In order for online calendar information to be useful, it must be readily available. Though users may have access to a Web browser much of the time, there will be times when the information must be available on

other devices such as a cell phone, pager, personal digital assistant (PDA), desktop client, and so on. Figure 1 shows the various devices and applications with which the Oracle Communications Calendar Server may interact.

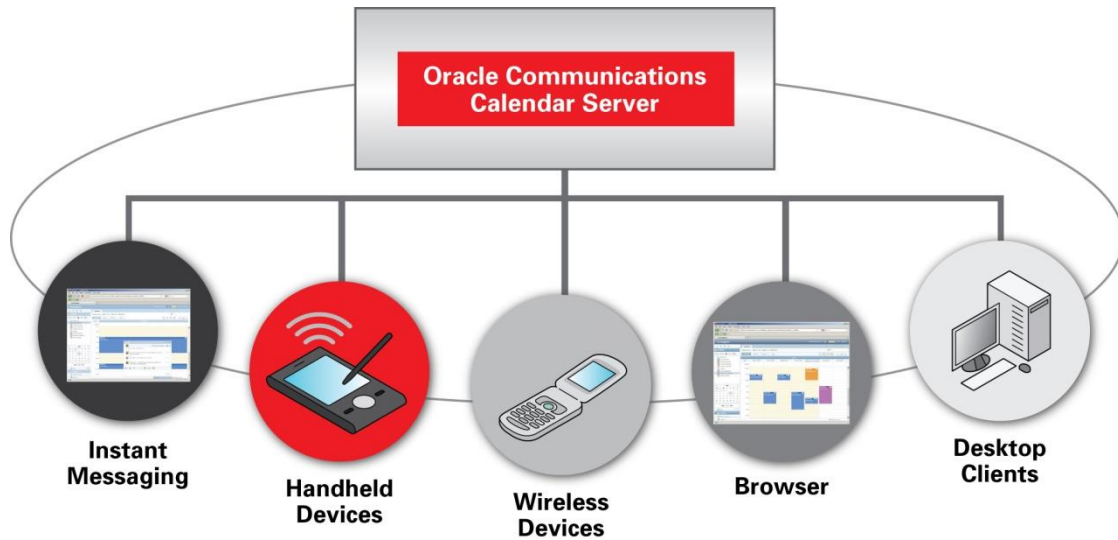


Figure 1. Oracle Communications Calendar Server Interactions

A calendar server must provide high performance and be highly scalable. It must be highly available so that customers' data is available when they need it. Ideally, it should integrate easily into an infrastructure of other services such as mail and directory.

A CSP or enterprise may wish to add value to services they already offer by adding calendaring features. A calendar server should facilitate development of custom calendar-based applications. The data supplied by the calendar server should be in a standard format.

Figure 2 shows an example of how the Oracle Communications Calendar Server can be incorporated with other servers that make up both CSP and enterprise infrastructures.

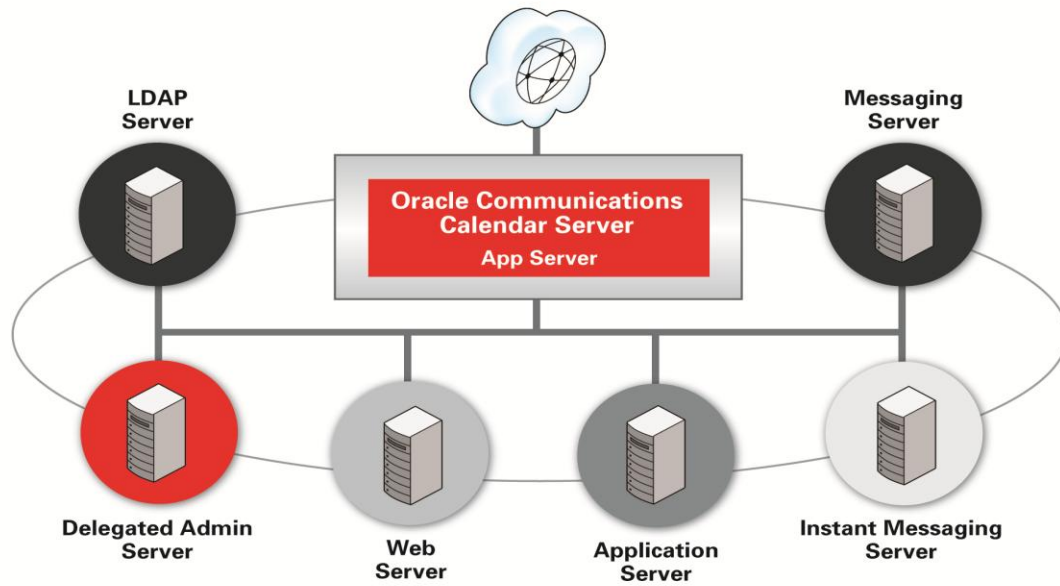


Figure 2. Oracle Communications Calendar Server relationship with other servers

The Oracle Communications Calendar Server requires external services for authentication and user preference storage. By default, these services are provided by LDAP.

The Oracle Communications Calendar Server is built from the ground up on standards. Its stored data format is based on iCalendar technology, and it supports iMIP Publish, Request, Reply, and Cancel messages for events and tasks. Native support for iCalendar standards enables users to schedule events in a format that is easily shared across the Internet. The Calendar Server's support for the CalDAV and WebDAV set of standards enables it to interoperate with multiple standards-compliant clients and to perform scheduling across servers and domains.

The Oracle Communications Calendar Server supports the use of a rich integrated AJAX Web client — Convergence — that combines calendaring, address book, e-mail, presence, and instant messaging into an integrated Web experience. Because Convergence is AJAX technology-based, it can be customized for a particular enterprise or CSP. The Oracle Communications Calendar Server also supports a number of CalDAV-compliant clients, including Apple iCal, Apple iPhone, Mozilla Lightning, and Notifylink. Additionally, the Oracle Communications Calendar Server provides a connector to Microsoft Outlook, thereby enabling Outlook to communicate with the Oracle Communications Calendar Server.

The Oracle Communications Calendar Server communicates natively using CalDAV but also supports a non-standard, more lightweight Web Calendar Access Protocol (WCAP) for interoperability with certain clients that do not communicate using CalDAV. Earlier versions of the Calendar Server communicated to clients exclusively over WCAP. In order to continue support for some of these clients, a certain amount of

WCAP support is provided with the Oracle Communications Calendar Server. WCAP 7.0 is based on the earlier Sun WCAP versions. WCAP is extremely useful in cases where calendar information must be provided to clients and devices that do not yet support CalDAV.

Calendar Server Architectural Overview

Figure 3 illustrates the Oracle Communications Calendar Server internal architecture.

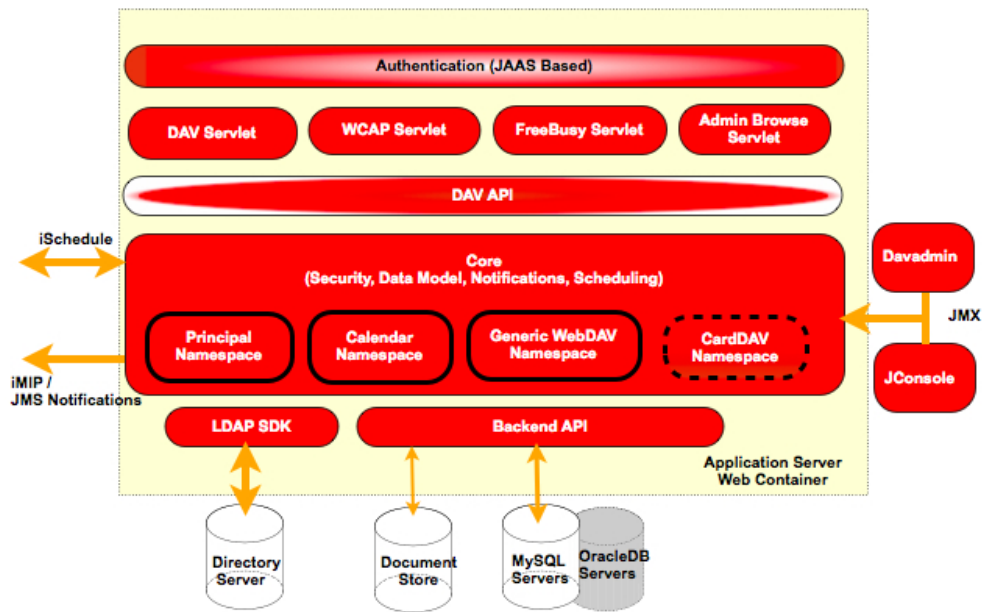


Figure 3. Oracle Communications Calendar Server Architecture

Calendar Server Components

The Oracle Communications Calendar Server is built, in part, on proven components from the Oracle Communications Messaging Server (formerly the Sun Java System Messaging Server) and Oracle Directory Server Enterprise Edition (formerly Sun Java System Directory Server), Oracle Glassfish Server and MySQL. The design is highly modular. The Oracle Communications Calendar Server is implemented by way of a collection of servlets deployed within a Web Container. In its basic state, the Calendar Server essentially consists of two components: a stateless Java EE-based front end, and a database backend. Both components can be located on the same host or the components may be separated onto multiple hosts. The Java EE-based front end is responsible for the primary functionality of the Calendar Server and performs such functions as freebusy lookup, group scheduling, and processing CalDAV commands. The functions of the front end fall into three main categories or subsystems: protocol(s) support, core, and database access.

Protocol Subsystem

The protocol subsystem consists of the servlets that handle communication to a variety of clients over various HTTP-based protocols. The DAV servlet is responsible for communicating and processing commands from CalDAV clients. The protocol subsystem also provides the capability to communicate with WCAP-based clients through the WCAP servlet. The WCAP component is responsible for processing WCAP commands from the Convergence server and the Outlook Connector. WCAP is based on HTTP and is an open protocol that can perform all server commands (except for certain administrative commands). It can be used by lightweight clients that need calendar information in JSON format or need most computation to be done on the server end. The protocol subsystem also exposes a Freebusy service and an administrative browse interface, both of which are based on HTTP.

Core Subsystem

The core subsystem performs a variety of functions such as data validation, scheduling, JMS event generation, and access control list (ACL) enforcement.

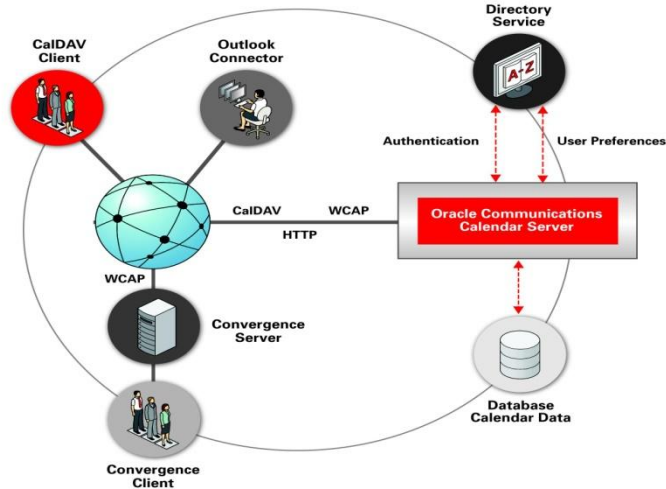
Database Subsystem

The database subsystem is responsible for storing data to and retrieving data from the database. The architecture supports horizontal scalability since there can be as many Java EE front-ends and database back-ends as required to support the deployment. The database storage is completely separated from the Core server. The architecture uses a standard MySQL or Oracle database backend and as a result, no additional tools are required to do database administration. The administrator may use the standard feature set available for each database, to administer the server. However, Calendar Server provides a specialized utility for database backup and restore that acts as a wrapper to the DB utilities.

Distributed Horizontally Scalable Architecture

The Calendar Server may be configured as a single system with connections to a directory server, the database, and the Internet as depicted in Figure 4.

Figure 4. Minimal Oracle Communications Calendar Server system



The directory server provides a user repository but it is also used for authentication. It is not necessary to do both on the same directory server. Although this diagram depicts the calendar server as a single system, portions of the server can be distributed over several systems.

For example, Web server processes can be run on one machine (the front-end server) and the database can be run on a separate machine (the back-end server). Furthermore, there can be multiple front- and back-end servers, creating the effect of distributing calendars over multiple calendar servers. For example, an administrator may wish to distribute all the calendar users over four machines. Machine 1 may serve all calendars in the U.S., the second machine may serve all calendars in Europe, and so forth. This effectively distributes the Calendar Server DAV repository across multiple DAV server instances by installing multiple instances of the database. There does not need to be a one-to-one correspondence between front and back-ends. One can just as easily deploy four CalDAV front-ends and three database back-ends. All front-ends will communicate with all back-ends.

Calendar Server APIs

In addition to providing a modular, highly scalable architecture, the Oracle Communications Calendar Server provides the following application programming interfaces (APIs) and software development kits (SDKs) to extend the functionality of the product.

Java Message Service APIs

The JMS API enables developers to handle event notifications. The Calendar Server utilizes JMS to emit notifications for events that are of interest to event subscribers. These events could be the creation of an

appointment, an alarm when a given appointment is about to occur, or another significant event. JMS provides a standard, flexible, and reliable approach to event notification, and developers can use the JMS API to create their own subscribers, to trigger an action on an event subscriber, or to extend the list of interesting events beyond what the Oracle Communications Calendar Server notification consumer provides.

Proxy Authentication

Proxy authentication enables a calendar administrator to log onto the Oracle Communications Calendar Server on behalf of a calendar user, which is useful for integrating additional servers or services with the Calendar Server. For example, a portal system may authenticate the user, and proxy authentication allows the user access to the Oracle Communications Calendar Server without reauthentication. The Calendar Server provides the ability to perform proxy authentication through both CalDAV and WCAP protocols.

Web Calendar Access Protocol (WCAP)

The WCAP mechanism enables a variety of clients to retrieve calendar data from the server. WCAP 7.0 (WCAPbis) may be used to query the Oracle Communications Calendar Server for data or modify data on it. WCAP returns data in one of three data formats: iCalendar, JSON or XML/iCalendar. This is extremely useful, and enables a wide range of devices to communicate with the Oracle Communications Calendar Server.

Advantages for Enterprises and Service Providers

In order for a calendaring solution to meet the needs of today's enterprises and service providers, it must use open protocols, enabling accessibility to real-time calendar data from the Web and popular desktop clients. It must support online and offline modes, and provide access to mobile devices. The interface should be customizable to enable unique branding. Furthermore, the calendaring solution must provide secure access and a means to ensure the privacy of the data. It must be scalable and adaptable to individual customer environments and highly available to ensure that the customer always has access to information about important appointments.

The Oracle Communications Calendar Server provides a wealth of features and benefits, including:

- Group scheduling, free-busy lookup, and corporate directory lookup
- Integrated personal address book and e-mail functionality through Convergence
- Resource scheduling for conference rooms, projectors, and other resources
- Global and domain-level customization (color, login, user interface, logo, branding, and so on) in Convergence
- Interoperability with additional communications clients including Microsoft Outlook, Mozilla Thunderbird and Apple iCal, enabling these clients to perform scheduling on the Oracle Communications Calendar Server

- APIs to support calendar server interoperability with other clients
- 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, enabling users to combine two or more calendars into a single view for improved communication and productivity in the Convergence client
- Convergence UI supports drag and drop of calendar events, context sensitive menus, event creation using natural grammar, and interactive hover pop-ups
- Flexible JMS-based event notification system. Automatic e-mail notification of appointments, invitations, and reminders sent to selected recipients
- Flexible access control mechanism that provides support for multiple owners of each calendar to facilitate communication and productivity with project teams and community groups. Provides the ability to delegate calendar ownership to others who may act on behalf of the primary owner
- Daily, weekly, and monthly views in the Convergence UI
- Agenda, invitation, and task views in the Convergence UI
- Support for hundreds of thousands of users through a scalable, networked, server-to-server, client-server architecture
- Support for Secure Sockets Layer (SSL) encryption, LDAP authentication, pluggable authentication and authorization service
- Event and task categories for identifying the type of event or task
- Text search for locating events based on key words or type of event
- Support for standards-based event and calendar data feeds published in iCalendar format, which improves communication and can offer new revenue opportunities through commerce links and banner ads
- Support for hosted domains including command line and GUI tools to manage these domains
- Simplified system management, simplified database management including backup and restore tools
- Wireless access via native CalDAV support and third party solutions
- Support for attachments in events and tasks
- Support for LDAP groups within invitations and calendar access permissions
- Lower total cost of ownership (TCO)

Multiclient Access

Service providers and enterprises today require accessibility to calendar data from a variety of clients, devices, and applications. The more accessible this data becomes, the easier it is for users to collaborate effectively. The Oracle Communications Calendar Server supports the use of a wide assortment of clients, including a rich AJAX technology-based Web user interface, a variety of CalDAV compliant clients like Apple iCal and Mozilla Thunderbird/Lightning, a connector to Microsoft Outlook, as well as mobile access.

Convergence

Convergence is a full-featured AJAX client that supports Oracle Communications Calendar Server. This client is a component of the Oracle Communications Unified Communications Suite and integrates personal address book, e-mail, and instant messaging functionality with advanced calendar functionality. It is a single client that presents a unified user interface (UI) to the back-end calendar, messaging, and instant messaging servers. It also provides access to a common address book that is shared by the mail, calendar, and instant messaging functions.

Convergence uses Asynchronous JavaScript and XML (AJAX) for its mail, calendar, address book, instant messaging and global options interfaces. It is deployed as a Web application using the Oracle Glassfish Server. Consequently, it can be deployed on a machine that is physically separate from the machine running the Oracle Communications Calendar Server Java EE front-end services, or it may be deployed on the same machine running the Calendar Server, and in fact within the same Web container.

Convergence supports calendar events, tasks, and reminders; displays calendars in standard views such as day, week, and month; and also supports object oriented views such as the Agenda, Invitations, and Tasks views that enable the user to view a specific type of calendar data.

The Convergence interface enables users to create and manage multiple calendars. For example, a user can have one calendar for work, a second calendar for a group of coworkers, a third calendar for company holidays, and a fourth calendar for personal activities. One of the most compelling features of Convergence is its support for combining several calendars and displaying them in a single composite view. A user can define any number of calendars within a single view. For example, a user may display a view consisting of a Work calendar, Child's School calendar, Sports calendar, and Family Event calendar. The user may also display a virtual team calendar, in which the user combines a personal calendar along with those coworkers with whom he or she works most closely. This allows the user to quickly see what everyone else is doing as well as when they can all meet.

In fact, one of the primary features of Convergence is its support for group scheduling. This encompasses a wide range of activities, from inviting members to a meeting and replying to invitations to tracking attendee acknowledgments and providing a free-busy lookup (which enables the user to quickly see when invitees are available for a meeting).

Convergence also provides an integrated address book for inviting individuals to meetings. One can invite users from a personal address book or a corporate directory. Generally, an attendee is invited to a meeting by entering a user's name (it is actually the email address that is invited, but Convergence maps the user name to the email address). However, through the address book integration, the user needs to enter only a portion of the attendee's name and auto completion is performed. Convergence presents a list of all name matches. The user can select the appropriate names from the list and invite them to the meeting. The corporate directory can also be used to schedule resources such as conference rooms.

Convergence supports configurable access control, enabling users to create public calendars that are readable and writable by others as well as private calendars that are accessible only to their owners. Permissions are categorized into two types: *access* permissions and *scheduling* permissions. Access permissions define how calendars are to be shared and hence define the access allowed on a given user's calendar, such as read permission and write permission. Scheduling permissions impact a user's privacy and hence define the access a user can grant to another user irrespective of the actual calendar, such as invite permission or check availability permission. Users can define a separate set of access permissions for each calendar that they own. The owners can also specify a list of users including LDAP groups and define a set of access permissions or scheduling permissions for each. Allowable access permissions include:

- Read access, which enables a user to subscribe to the calendar and view events in it
- Read + Write access, which enables a user to view and modify events in the calendar
- Owner access, which enables a user to view, modify, or delete events in the calendar

Allowable scheduling permissions include:

- Availability access, which enables a user to view the free/busy availability of the user
- Invite access, which enables a user to schedule events with another user

Access control for calendars is supported at the individual calendar level as well as the global and system administrator levels. The type of access granted can be differentiated across individuals. For example, a user may wish to grant certain users read-only access while others receive availability and invite-only access. Availability and invite-only access allow a user to schedule an event with another user, but the user cannot read appointments in the user's calendar. As a result, User1 knows when User2 is available for a meeting (and can schedule a new meeting into User2's calendar), but cannot see individual items on User2's calendar. In contrast, when a calendar is configured for public read, no access check is made for read requests. This means that a URL to a public calendar can be put into a static Web page, an e-mail message, or a newsgroup message; when the user clicks it, the calendar is displayed without requiring the user to log into the server.

Convergence also provides access control at the individual event level through the use of public, private, and confidential events. Individual event access control enables users to create private events within an otherwise public calendar. This is especially useful for users who would like to keep their default calendar public, but want to control the privacy of certain events within that calendar. Events can be designated public, private, or time and date only.

Designed with worldwide calendar sharing in mind, Convergence supports multiple time zones. The time scale for calendar views can be in any time zone. In fact, multiple time zones can be shown, and a specific time zone can be applied to a calendar.

Recurring events and tasks are supported with a full range of recurrence options, including repeat for a specific number of instances and repeat until a specific date. Events and tasks can also have reminders that may be e-mailed to one or more addresses. Printer-friendly views are also supported. Convergence can import and export calendar information in iCalendar format.

Convergence not only provides superior calendar functionality, but it also provides an integrated solution that is more functionally rich than the individual calendar or messenger user interfaces. Consequently, the Convergence user interface is ideal for those individuals who want a fully integrated rich Web client — not just a standalone calendar client. However, it is not necessary to install the Messaging Server or Instant Messaging to take advantage of Convergence. Convergence can still be used as an alternative interface to traditional interfaces even if only the Oracle Communications Calendar Server is installed. The Personal Address Book integration would be present with or without the Messaging Server or Instant Messaging. Convergence introduces calendar-specific, value-added features such as color-coded calendars and the ability to use natural grammar to schedule events. Convergence enables users to:

- Invite a contact to an event, or e-mail a contact from within a personal address book
- View an invitee's instant messaging presence while creating a calendar event
- Change a calendar event's date or time using Drag and Drop
- Search for users in a personal address book or remote address book such as the corporate address book from within calendar
- Quickly create an event or a contact using instant event and contact shortcuts from within the calendar or address book
- Generate a list of all outstanding events, invitations, or tasks
- View the email sender or recipient's calendar availability from within the messaging application
- Easily traverse between the calendar view, the mail view, and the address book view as seamlessly as clicking links on a Web page

The Convergence UI is highly customizable. Because it is based on AJAX, the calendar client can be customized to suit the needs of the deployment or the domain. Customization examples include customizing the skin or theme, the login screen, the icons used in the calendar component, the text within Convergence, the application banner and application bar, the branding, the toolbars, the calendar views, the calendar pop-ups, and so on. Customization also includes the integration of 3rd party applications into the user interface not only to enable a new service but to integrate with the existing services as well. To the end user, the application appears in the UI as another component, just like mail or calendar.

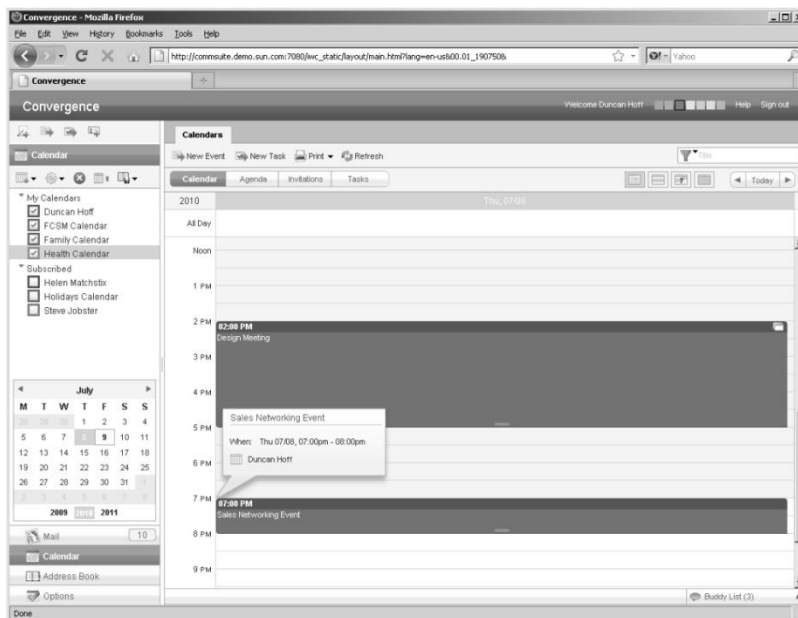


Figure 5. Convergence interface

Outlook Connector

In addition to providing a native Web client, the Oracle Communications Calendar Server supports the use of a connector to Microsoft Outlook. The Connector for Microsoft Outlook enables direct connection from Outlook to the Oracle Communications Calendar Server, Oracle Communications Messaging Server, and Oracle Directory Server Enterprise Edition. Therefore, it eliminates the need for a Microsoft Exchange Server for messaging and calendaring. The Connector for Microsoft Outlook for Oracle Communications Unified Communications Suite is a Messaging API (MAPI) provider that is installed on the Outlook desktop machine. It uses Simple Mail Transfer Protocol (SMTP) and Internet Message Access Protocol 4 (IMAP4) protocols to communicate with the Messaging Server, LDAP to communicate with the Directory Server, WABP (Web Address Book Protocol) to communicate with the personal address book, and WCAP to communicate with the Oracle Communications Calendar Server.

The Connector for Microsoft Outlook maintains a persistent connection with the Oracle Communications Calendar Server, thereby enabling real-time access to calendar data from Outlook. Because Outlook provides a database or Personal Folder Storage (PST) file for use in offline mode, users of the Connector for Microsoft Outlook can update their schedules while on the road and later synchronize their offline store with their calendar server. Additionally, the Oracle Communications Calendar Server supports a mixed environment of clients such that Convergence users can easily schedule appointments with Outlook users and vice versa.

Key features of the Connector for Microsoft Outlook for Oracle Communications Unified Communications

Suite include:

- Access to mail, calendar, contacts, and tasks
- Support for Outlook corporate/workgroup mode
- Send and receive meeting requests and responses via e-mail
- View free/busy times of invitees
- Calendar sharing and delegation
- Shared folders (mail, calendar, contacts, and tasks)
- Outlook to Convergence interoperability
- Offline access
- Automated install and configuration
- Seamless upgrade from a previous version
- Desktop data migration and conversion
- Supported platforms: Outlook 2003 and Outlook 2007
- Localized in seven languages

Other Clients

Because the Oracle Communications Calendar Server supports CalDAV, it interoperates with virtually any CalDAV-compliant client. This includes the popular iCal desktop client from Apple as well as the iPhone mobile client. Another popular desktop client is the Lightning calendar extension to the Mozilla Thunderbird messaging client. Lightning is essentially an open source calendar component for the Thunderbird client that provides an integrated messaging and calendaring solution. The Oracle Communications Calendar Server has a simple, open protocol, the Web Calendar Access Protocol (WCAP), which sits on top of HTTP that can be used by new and custom applications. With WCAP 7.0, any application that can open a socket and speak HTTP can be a calendar-enabled application.

Finally, Oracle Communications Calendar Server integrates with several third party solutions that extend calendaring functionality to other clients such as SyncML devices, Blackberry devices, Windows Mobile devices, Palm devices, and other clients.

Security at Multiple Levels

Security plays a key role in the day-to-day operations of today's enterprise. Breaches in security cannot only compromise trade secrets, but may also result in downtime, data corruption, and increased operation costs. The Oracle Communications 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. The Oracle Communications Calendar Server uses LDAP authentication by default, but the authentication and authorization service is pluggable.

Security involves not only ensuring the integrity of users; it also means ensuring the confidentiality of data. To this end, the Oracle Communications Calendar Server supports the use of SSL encryption for

both login and data. In other words, the entire session including the login may be encrypted, from the Web client to the server. The Calendar Server also supports the use of SSL to the LDAP directory 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, there exists other layers of security to prevent authenticated users from unauthorized access to other authenticated users' calendar data. One security measure is through the Oracle Communications 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 or her 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.

In addition the Calendar server offers the ability to scan all data for viruses. The virus scan can be done online as data comes in, or offline to selected data in the database.

Another security related feature is the ability to block off certain rogue clients to prevent them from inundating the server with bogus requests and resulting in a denial of service attack.

High Performance, High Availability

The Oracle Communications Calendar Server utilizes several features and techniques to create a high-performance calendaring solution. The very nature of its distributed architecture enables the Oracle Communications Calendar Server to be deployed with several front-end calendar Web servers communicating to one or more database back-ends. The AJAX Web client utilizes proxy servers to send requests to the Calendar Server. As client requirements increase, scalability can be enhanced by increasing the number of intermediate proxy and/or front-end calendar Web servers.

Oracle Communications Calendar Server provides an LDAP-based solution to locate calendars in a multi-database environment. An LDAP attribute identifies the location of a given user's calendars. The attribute could specify a calendar database that is consistent with a user's geographic location. The Oracle Communications Calendar Server also provides a number of tunable parameters that enable the administrator to improve performance. Some of these parameters involve LDAP indexing for certain Oracle Communications Calendar Server attributes, specifying the size of the LDAP cache, and more.

To ensure reliability, the Oracle Communications Calendar Server can be configured to be highly available by using Application Server clustering technology. A cluster is a loosely coupled collection of Application Server instances that work together as one logical entity. A cluster provides a runtime environment for one or more Java EE applications. A **highly available cluster** integrates a state replication service with clusters and a load balancer. The Oracle Glassfish Server provides homogeneous clusters and enables an administrator to manage and modify each cluster as though it were a single entity. Usually that means

that the server instances reference the same server configuration, can access the same physical resources, and have the same applications deployed to them. Homogeneity assures that before and after failures, the load balancer always distributes load evenly across the active instances in the cluster. It should be noted that the cluster not only provides high availability; it also enhances the scalability since increasing the number of instances naturally increases the capacity of the system. The Oracle Glassfish Server also supports two means of maintaining session-state information. One can use the High Availability Database (HADB) technology, or one can use the lighter weight memory replication alternative. Memory replication relies on instances within the cluster to store HTTP session and stateful session information for one another in memory, not in a database. In the Oracle GlassFish Server, cluster instances are organized in a ring topology. Each member in the ring sends memory state data to the next member in the ring, its replica partner, and receives state data from the previous member. As state data is updated in any member, it is replicated around the ring. Normally, an administrator would place replica partner instances on different physical machines in order to maximize availability. In a typical cluster topology, multiple application server instances may run on a single node or host machine, and there may be several nodes in a cluster. If a machine suffers a catastrophic failure, all the data is preserved on the other machine, either in its original form or as replicants of the instances on the failed machine. If a single instance of the cluster fails, a working instance may already have the stored replication data from the session and can take immediate ownership of the session, or this instance can broadcast a request for the replication data and that instance would then obtain the stored session information it needs from another instance. In either case, this capability provides automatic failover when a system shutdown or failure occurs. When an instance within the cluster fails, the surviving instances upstream and downstream of the failed replica partner reconfigure themselves to select each other as new replica partners. In this way, the cluster topology dynamically adjusts as instances are brought offline or online. In-memory replication should be considered when 99.999% or five nines availability is not required. It is very easy to set up and administer, and requires less overhead than does HADB.

If 99.999% availability is required, the Oracle Glassfish Server provides the High Availability Database for storing HTTP session and stateful session bean data. Generally, HADB must be configured and managed independently of the Application Server, but this also means that it can be independently scaled for optimum performance. It requires additional hardware resources and is more complex to administer but provides higher availability for applications that require it. For further information regarding application server high availability, please consult the Sun Java System Application Server High Availability Administration Guide.

Database Availability

Database availability is a key component of high availability. The Oracle Communications Calendar Server database is either MySQL or OracleDB and hence leverages a wealth of tools and technology already present in the DBs themselves or through third party solutions to provide a highly available database. This includes replication and clustering solutions as well as mechanisms to ensure database consistency and rapid database recovery should there be a database corruption. In terms of database

backup and recovery, the Oracle Communications Calendar Server supports a number of techniques to satisfy a variety of environments. One technique is the use of file system snapshots or ZFS Snapshots to produce periodic snapshots of the file system including the database. These snapshots can then be backed up using ZFS or third party software.

MySQL also provides mechanisms to replicate the database or back it up. With database replication, data from one MySQL server is replicated onto one or more slave servers in an asynchronous fashion. This means that slaves do not require a permanent connection to the network and that updates from the master to the slave can occur over long distances or intermittent connections. In addition to database replication, MySQL provides tools such as *mysqldump* to backup a database or collection of databases. In addition to the MySQL tools, the Oracle Communications Calendar Server provides a native backup and restore utility for specifically backing up the calendar database. Finally, administrators can use commercial backup products from third parties such as Innobase. The advantage in using MySQL as the database is the accessibility of a wide assortment of database management tools, resulting in reduced cost and deployment cycles.

Java Messaging Service (JMS)

The Oracle Communications Calendar Server uses the built-in Java Messaging Service (JMS) in the Oracle Glassfish Server as its default notification mechanism to provide alarm notifications and server notifications. An alarm notification could be a reminder that a calendar appointment is about to happen or a task is due. A server notification could be a Glassfish Message Queue or Java Message Queue (JMQ) notification that a calendar has been created or an event has been modified. There are two parts to the format of a notification: The event reference, which is the URL identifying the event, and the payload, which is the iCalendar data describing the event. The Message Queue service implements the Java Messaging Service (JMS) specification, providing a message broker, interfaces to create clients that produce or consume messages, and administrative services and control. JMS provides a standard, flexible, and reliable approach to event notification.

Other benefits of Glassfish Message Queue include:

- Users can produce messages to a topic or a queue or to both of these delivery methods. Anyone subscribing to the topic will receive the message. If no one subscribes to the topic, the message is discarded. Messages delivered to a queue remain in the queue until either the message times out or a consumer retrieves the message. Only one subscriber can receive the message even if there are multiple consumers waiting for messages in the queue. (Calendar Server notifications use Topics -- not queues. Multiple consumers can subscribe and make use of the Topic notifications.)
- Glassfish Message Queue offers enhanced load balancing during message distribution, especially when messages are produced to a queue.
- Glassfish Message Queue supports multiple notification plug-ins that can produce messages to a topic, a queue, and so on.

- Glassfish Message Queue provides a reliable notification delivery mechanism. The queue can be configured to be persistent such that if a server goes down, the messages can be retrieved and made available to the appropriate consumer.

Calendar Server Notification Services use a publish/subscribe model. The Notification module consists of two major components, the **Notification Service** and **Notification Consumer**. The Notification Service component is part of the Calendar server, and is the publisher that posts messages to a pre-configured JMS topic managed by the JMS provider. The Notification Consumer component is the subscriber or the message consumer of that JMS topic. The default consumer bundled with the Calendar Server listens for important events and sends a notice via email when the event occurs. Developers can choose to write their own customized Notification Consumer programs. The advantage of such an approach to event notification is that developers can extend the notification capabilities of the Calendar Server beyond what is provided out of the box.

Hosted Domains

Some businesses may choose to outsource their messaging and calendaring needs to large service providers capable of supplying many enterprises with these services. For the service provider utilizing this business model, the ability to host many user domains is critical to the success of the business. The Oracle Communications Calendar Server provides support for hosted domains, with each domain sharing the same instance of the software. This means that many domains can exist on a single server.

The Oracle Communications Calendar Server supports both LDAP Schema v.1 and LDAP Schema v.2. In LDAP Schema v.1, the LDAP directory organization for a hosted-domain installation includes two trees for domain management: A DC tree and an Organization (OSI) tree. LDAP Schema v.1 can only be used in LDAP-based calendar deployments where the Sun Access Manager is not managing the LDAP directory server. In contrast, LDAP Schema v.2 can be used in both identity-based deployments and LDAP-based deployments, but is required for identity-based deployments. Because the Oracle Communications Calendar Server supports both versions of LDAP Schema, hosted-domain support is available for both LDAP- and identity-based calendar deployments.

The nature of hosted domains is that the end user does not know about users on another domain. The information contained within the domain is not shared with anyone outside the domain unless there is an agreement in place to do so. By default, users may search only within their own domains to invite other users and groups to meetings.

Calendar Server Administration

The Oracle Communications Calendar Server provides a flexible administration mechanism for provisioning and/or managing users, resources, calendars, the database and more. The Calendar Server uses the Java Management Extensions (JMX) to perform administration and configuration. The primary tool for managing the calendar server is a command line utility that performs a variety of functions such

as creating calendars, creating events, and backing up the calendar database. This tool runs over JMX and an administrator can also use any of the supported JMX consoles such as JConsole to monitor the deployment. The Oracle Communications Calendar Server may be provisioned through the Delegated Administrator graphical user interface or the Delegated Administrator command line interface.

The Command Line Interface

The command line interface for performing calendar-specific administrative tasks enables an administrator to develop simple scripts to automatically perform administrative functions. The primary function of this interface is to configure the calendar server, manage calendars, and manage the database. There are a number of command line utilities available, all running under the umbrella command of *davadmin*. One or more users can be assigned as calendar server administrators who typically perform the following functions:

- List, delete, or modify accounts
- Perform, list, or set configuration options
- Create, modify, and delete calendars (also known as calendar collections)
- List, delete, or import calendar components (also known as calendar resources)
- Back up and restore the calendar database, or list the contents of a backup file
- Migrate users and data from Calendar Server 6.3 to Oracle Communications Calendar Server 7

Administrative Functions

The following are examples of administrative functions that can be performed through the command line interface:

- List all configuration parameters
- Set a particular log level for logging error messages
- Back up a user's calendars, or the entire calendar database
- Perform an incremental backup or full backup for a particular domain
- Restore calendars or database from a backup file
- Manage calendars and their properties
- Manage events and tasks in a calendar
- Manage calendar databases
- Manage calendar users (includes modifying user accounts, changing a user's email address, deleting users, or listing the properties of an account)

Delegated Administrator Functions

The Delegated Administrator for Oracle Communications Unified Communications Suite provides an intuitive graphical user interface for provisioning domains, calendar users and resources. An administrator can perform the following functions using this tool:

- Create, modify, and delete organizations
- Create, modify, and delete users
- Create, modify, and delete calendar resources such as conference rooms

- Create, modify, and delete user groups
- Manage service packages that define the type of service applied to a user
- Assign organization administrator roles to users and distribute administrative capabilities to these organization administrators

In addition, the Delegated Administrator provides a command line utility for performing most of the functions performed by the graphical user interface.

Migration

Migration from one calendar system to another can be an expensive ordeal and, unless the organization has the right tools and procedures to perform a smooth migration, the cost of migration could seriously impact business and offset any previously perceived benefits. Some service providers and enterprises may not have a calendaring solution deployed and may choose to deploy the Oracle Communications Calendar Server. Other businesses may already have an earlier version of the Sun Calendar Server deployed, but would like to migrate data to the new version.

To support these scenarios, the Oracle Communications Calendar Server provides a migration tool for migrating users and data from the Calendar Server 6.3 version to Calendar Server 7. Users who are on earlier versions of the Calendar Server can use a variety of tools shipped with Calendar Server 6.3 to get them to that particular version. After migrating to version 6.3, one may use the Calendar Server 7 migration utility to perform the final migration.

Conclusion

The Oracle Communications Calendar Server is a compelling solution for any business seeking to provide calendar access to its employees, partners, or customers. It offers several key features that make it an exciting component of the Oracle Communications Unified Communications Suite.

High-Performance Server

The Oracle Communications Calendar Server is built on an extensible, distributed, and scalable architecture. The server scales both horizontally across machines and vertically across CPUs, and can be enhanced to support an ever-growing number of end users. The distributed nature of the architecture enhances the overall scalability of the calendaring solution, since various calendar server components and services can be distributed throughout the network and run on dedicated hardware if necessary. It is built on scalable and highly available components like the Oracle Glassfish Server and the MySQL database.

Based on Standards and Proven Technology

The Oracle Communications Calendar Server is built on standards such as iCalendar, CalDAV, WebDAV, XML, HTTP, iMIP, iTIP, LDAP, and SMTP. As a result, services and applications that speak these

protocols can be easily integrated into the Oracle Communications Calendar Server environment. Furthermore, support for open standards like CalDAV enables the calendar server to interoperate with a growing number of CalDAV-compliant clients.

The Oracle Communications Calendar Server integrates seamlessly with the proven, highly scalable Oracle Directory Server Enterprise Edition, Oracle Communications Messaging Server, Oracle Communications Instant Messaging Server, and Convergence. The integration with other products produces an overall communications solution that is greater than the sum of its parts. Integration with the Messaging Server and Convergence produces a tightly integrated communications solution that includes calendar, messaging, address book, chat, and presence.

Flexibility and Extensibility

The Oracle Communications Calendar Server supports multiple devices and clients through native CalDAV support as well as a connector for Microsoft Outlook. Oracle Communications Calendar Server enables organizations to consolidate to a single calendar server and still use a wide variety of clients such as Thunderbird, Outlook, Apple iCal, Convergence, and a wide assortment of devices including Blackberry and iPhone devices. The more flexible the calendar solution, the more likely that solution will be able to adapt to individual customer environments and changing requirements.

Feature-rich Server and Client

The Oracle Communications Calendar Server provides a solution that is feature-rich on the server and on the client. It addresses the needs of today's organizations that require online and offline support, mobility, client customization, secure access, and high availability. Finally, the Oracle Communications Calendar Server provides all of this with a lower total cost of ownership than similar solutions available today.

Appendix A

Standards and References

This section provides a list of relevant standards (under the technology/function categories as indicated) that are supported by the Oracle Communications Calendar Server.

Calendaring

- CalDAV (RFC 4791) — Calendaring Extensions to WebDAV
- iCalendar (RFC 5545) — Describes a standard schema for calendar objects on the Internet
- iTIP (RFC 5546) — iCalendar Transport-Independent Interoperability Protocol
- iMIP (RFC 6047) — iCalendar Message-based Interoperability Protocol
- CalDAV Scheduling Extensions to WebDAV — Defines how CalDAV scheduling occurs
- Collection Synchronization for WebDAV — Defines WebDAV collections
- XML — eXtensible Markup Language (REC-XML)
- WCAP 7.0 — Web Calendar Access Protocol 7.0

Messaging

- IMAP (RFC 2060) — Internet Message Access Protocol
- SMTP (RFC 821) — Simple Mail Transfer Protocol

Directory Services

- LDAP (RFC 2251) — Lightweight Directory Access Protocol (RFC 2251)

HyperText Transfer Protocol (HTTP)

- HTTP 1.1 — HyperText Transfer Protocol 1.1 (RFC 2068 and RFC 2616)

Abbreviation/Acronym List

AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
authSDK	Proxy Authentication SDK
CUA	Calendar User Agent
CSP	Communication Service Provider
DB	Database
GUI	Graphical User Interface
HTML	HyperText Markup Language
HTTP	HyperText Transport Protocol
ICAL	Internet Calendaring (also iCalendar)
IETF	Internet Engineering Task Force
JMS	Java Messaging Service
JSON	JavaScript Object Notation
IMAP	Internet Messaging Access Protocol
iMIP	iCalendar Message-Based Interoperability Protocol
IP	Internet Protocol
ISP	Internet Service Provider
iTIP	iCalendar Transport-Independent Interoperability Protocol
LDAP	Lightweight Directory Access Protocol
MAPI	Messaging Application Programming Interface
PAB	Personal Address Book
PIM	Personal Information Manager
PST	Personal Folder Storage
SDK	Software Development Kit

SMS	Short Message Service
SMTP	Simple Mail Transfer Protocol
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
UI	User Interface
URL	Uniform Resource Locator
WCAP	Web Calendar Access Protocol
XML	eXtensible Markup Language
XHTML	eXtensible HyperText Markup Language



Oracle Communications Calendar Server
September 2011

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com



| Oracle is committed to developing practices and products that help protect the environment

Copyright © 2009, 2010, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. UNIX is a registered trademark licensed through X/Open Company, Ltd. 0510

