

プログラマリファレンス

iPlanet Calendar Server

リリース5.0

プログラマリファレンス
2001年2月

Copyright © 2000 Sun Microsystems, Inc. Some preexisting portions Copyright © 2000 Netscape Communications Corporation. All rights reserved.

Sun、Sun Microsystems、Sun のロゴ、iPlanet、iPlanet のロゴは、Sun Microsystems, Inc. の米国およびその他の国における商標または登録商標です。Netscape と Netscape N のロゴは、米国およびその他の国における Netscape Communications Corporation の登録商標です。その他の Netscape のロゴ、製品名、およびサービス名は、Netscape Communications Corporation の商標であり、その他の国でも登録されている場合があります。

米国政府による本製品の取得については、「Commercial Software-Government Users Subject to Standard License Terms and Conditions」の条項が適用されます。

本書に記載の製品は、使用法、コピー、配布、逆コンパイルを制限した使用許諾規定の下に配布されます。製品または本書の全部または一部を、Sun-Netscape Alliance およびそのライセンス交付者の書面による許可がない場合、いかなる手段によっても複製することが禁じられています。

本書は、「現状のみ」で提供されています。また、法律的にこれらの否認が無効であると見なされる範囲を除き、商品性、特定の目的への適合性、または不侵害性に対する黙示的保証を含む、すべての明示的、暗示的な条件、表現、保証に対して責任を負わないものとします。

目次

このマニュアルについて	11
対象読者	11
お読みになる前に	12
内容の紹介	12
表記上の規則	13
関連マニュアル	14
オンラインマニュアル	14
第1章 アーキテクチャの概要	17
iPlanet Calendar Server の概要	18
機能の要約	20
バージョン 5.0 の新機能	21
カレンダーサーバサービス	22
csadmin	23
cshttpd	23
csnotifyd	23
csdwpd	23
enpd	24
起動順	24
グループスケジューリング	24
ディレクトリサーバサービス	25
水平方向のスケラビリティ	25
構成	26
単純な単一インスタンス	27
ネットワークフロントエンド、データベースバックエンド	27
複数のフロントエンド、複数のバックエンド	28
新しいデフォルトクライアント UI: SHTML	29
アーキテクチャの基本	30

SHTML と WCAP	31
コア	31
データベース	33
カレンダーデータ	33
カレンダーデータの形式	33
グループ	33
イベントフィード	34
カレンダーデータの交換	34
カレンダーのユーザ設定	35
カレンダーアクセスコントロール	35
サポートされるコーディング方式	35
カレンダーサーバ API (CSAPI)	36
イベント通知サービス (ENS)	37
プロキシ認証 SDK (authSDK)	37
シングルサインオン (SSO)	38
Web カレンダーアクセスプロトコル (WCAP)	38

第 2 章 Calendar Server API (CSAPI) の概要 41

CSAPI のアーキテクチャ	42
スレッドに対する安全の要件	44
依存性	44
CSAPI の使用法	44
CSAPI モジュールの読み込み	45
プラグインインタフェース	46
クライアントおよびサーバの API	46
サーバに対する照会の例	47
CSAPI のサンプル	48

第 3 章 CSAPI リファレンス 49

csIAccessControl	50
CheckAccess	50
Init	52
csIAuthentication	52
ChangePassword	53
Init	54
Logon	55
Logout	55
VerifyUserExists	56
csICalendarLookup	57
Init	57
QualifyCalid	58
FreeCalid	58

QueryType	59
FreeType	60
csIDataTranslator	60
GetSupportedContentTypes	61
Init	62
Translate	63
csIPlugin	64
GetDescription	64
GetVendorName	65
GetVersion	65
Init	66
csIQualifiedCalidLookup	66
FindCalid	67
Init	68
csIUserAttributes	68
FreeAttribute	69
GetAttribute	69
Init	70
SetAttribute	71
csICalendarServer	72
GetVersion	72
Init	73
csIMalloc	73
Calloc	74
Free	74
FreeIf	75
Init	75
Malloc	76
Realloc	76
第 4 章 イベント通知サービス (ENS) の概要	79
ENS 用語集	80
イベント	81
イベント参照	81
例	82
イベント通知サービス	82
通知する	82
購読する	83
購読を中止する	83
Calendar Server と ENS との連携動作	84
アラームキュー	85
デーモン	85
アラーム転送の信頼性	86

例	86
API の概要	88
パブリッシャ API 関数	89
サブスクリバ API 関数	89
公開および購読ディスパッチャー API 関数	90
カスタムアプリケーションの作成と実行	91
コーディング例の場所	91
インクルードファイルの場所	91
動的にリンクまたは共有されるライブラリ	91
実行時ライブラリパス変数	92

第 5 章 イベント通知サービス API

リファレンス	93
パブリッシャ API の関数リスト	93
サブスクリバ API の関数リスト	94
公開および購読ディスパッチャーの関数リスト	94
パブリッシャ API	95
publisher_t	95
publisher_cb_t	96
publisher_new_a	96
publisher_new_s	97
publish_a	98
publish_s	99
publisher_delete	100
publisher_get_subscriber	100
renl_create_publisher	101
renl_cancel_publisher	102
サブスクリバ API	102
subscriber_t	103
subscription_t	103
subscriber_cb_t	103
subscriber_notify_cb_t	104
subscriber_new_a	104
subscriber_new_s	105
subscribe_a	106
unsubscribe_a	107
subscriber_delete	108
subscriber_get_publisher	108
renl_create_subscriber	109
renl_cancel_subscriber	110
公開および購読ディスパッチャー API	110
pas_dispatcher_t	111
pas_dispatcher_new	111

pas_dispatcher_delete	111
pas_dispatch	112
pas_shutdown	112
コーディング例	113
簡単なパブリッシャとサブスクリバ	113
パブリッシャのコードサンプル	113
サブスクリバのコードサンプル	116
信頼性の高いパブリッシャとサブスクリバ	119
信頼性の高いパブリッシャのサンプル	119
信頼性の高いサブスクリバのサンプル	122
第 6 章 プロキシ認証 SDK の概要	125
SDK を使用するユーザ	125
authSDK とは	125
アーキテクチャ	126
初期化	126
検索	126
クリーンアップ	126
関数の概要	127
第 7 章 プロキシ認証 SDK リファレンス	129
プロキシ認証 SDK 関数のリスト	129
プロキシ認証 SDK 関数	130
CEXP_GenerateLoginURL	130
CEXP_GetVersion	131
CEXP_Init	131
CEXP_SetHttpPort	132
CEXP_Shutdown	132
authSDK の使い方	133
その他のヒント	134
第 8 章 シングルサインオン認証	135
シングルサインオンとは	135
シングルサインオンの制限	136
処理の流れ	136
実装の要件	138
cookie 情報	138
その他の推奨設定	139
信頼されたアプリケーションレコード	139
シングルサインオフパラメータ	139
接頭辞の文字列	139
シングルサインオンの例	139

例	140
この例の構成パラメータ	141
留意点	142
セキュリティ	142
管理	142
スケーラビリティ	143
パフォーマンス	143
第9章 Web カレンダーアクセスプロトコル (WCAP) の概要	145
はじめに	145
このバージョンの新機能	146
コマンドの概要	148
セッション ID	149
コマンドの形式	150
クライアントの要求形式	150
URI 形式	151
HTML フォーム	151
クライアント側のイベント通知	151
サーバの応答形式	151
第10章 WCAP コマンド	153
共通のトピック	153
コマンド	153
共通のトピック	156
アクセスコントロールエントリ (ACE)	156
ほかの言語または文字セットの選択	159
繰り返しコンポーネントの削除	160
符号化文字	161
エラー処理	161
エラー文字列	161
レイヤーエラー番号配列	162
レイヤーカウント配列	162
エラーコード	162
時刻、文字列、パラメータなどの形式	165
予定なし / 予定ありに限定したアクセス	165
iPlanet Calendar Server 5.0 の新しいコマンドおよびパラメータ	166
新しいコマンド	167
新しいパラメータ	167
出力形式	169
簡略出力	170
繰り返し処理	170
rules	171

rdates	171
exrules	172
exdates	172
rid	173
mod	173
rchange	173
コマンド	174
addlink	175
change_password	177
check_id	178
createcalendar	179
deletecalendar	181
deletecomponents_by_range	182
deleteevents_by_id	184
deleteevents_by_range	187
deletetodos_by_id	188
deletetodos_by_range	191
export	193
fetchcomponents_by_alarmrange	196
fetchcomponents_by_attendee_error	199
fetchcomponents_by_lastmod	201
fetchcomponents_by_range	204
fetchevents_by_id	214
fetchtodos_by_id	216
get_all_timezones	218
get_calprops	220
get_freebusy	224
get_guids	228
get_userprefs	230
import	232
login	234
logout	237
ping	237
search_calprops	238
set_calprops	241
set_userprefs	245
storeevents	247
storetodos	257
upload_file	264
version	266
write_file	267

このマニュアルについて

このマニュアルでは、iPlanet Calendar Server 5.0 のアーキテクチャと、サーバのカスタマイズに使用できる次の 2 つの API と 1 つのプロトコルの使い方について詳細に説明します。

- **Calendar Server アプリケーションプログラムインタフェース (Calendar Server Application Program Interface、CSAPI)**。サーバの機能を変更する
- **イベント通知サーバ (Event Notification Server、ENS)** アプリケーションプログラムインタフェース。公開および予約購読機能を変更する
- **プロキシ認証 SDK (Proxy Authentication SDK、authSDK)**。ポータル認証サービスを使用するための外部プラグイン
- **Web Calendar Access Protocol (WCAP)**。カレンダーサービスにアクセスする

また、iPlanet Calendar Server の単一ドメインインスタンスで使用する、シングルサインオン (Single Sign-on、SSO) 代替認証スキーマについても説明します。

この章は、次の項目について説明します。

- 対象読者
- お読みになる前に
- 内容の紹介
- 表記上の規則
- 関連マニュアル
- オンラインマニュアル

対象読者

このマニュアルは、iPlanet Calendar Server 5.0 を実装するときに、アプリケーションをカスタマイズするプログラマを対象にしています。

お読みになる前に

このマニュアルは、C および C++ の知識を持ち、以下について概要を理解しているプログラマを対象にしています。

- インターネットと World Wide Web
- カレンダー操作の概念
- LDAP
- RFC2445、RFC2446、RFC2447

これらの RFC には、(特に指定のない場合に)WCAP コマンドで使用される時間、文字列、パラメータなどの形式と定義について詳細に記述されています。

RFC は次の IETF Web サイトで参照できます。

- <http://www.ietf.org/rfc/rfc2445.txt>
- <http://www.ietf.org/rfc/rfc2446.txt>
- <http://www.ietf.org/rfc/rfc2447.txt>

内容の紹介

このマニュアルでは、iPlanet Calendar Server 内部の 3 つの API、SDK、および 1 つの プロトコルについて製品の全般的なアーキテクチャとともに説明します。インタフェースごとに、概要を説明する章、リファレンスの章の順に構成されています。概要だけの場合もあります。

章の構成は以下のとおりです。

- このマニュアルについて (この章)
- 第 1 章「アーキテクチャの概要」

この章は、iPlanet Calendar Server 全体の概要と、このマニュアルに記載されている各 API やプロトコルの簡単な説明で構成されます。

- 第 2 章「Calendar Server API (CSAPI) の概要」

プログラマはこの API を使って、サーバのアクセス制御、認証、カレンダー検索、データ形式の変換、およびユーザ属性のアクセスをカスタマイズできます。

- 第 3 章「CSAPI リファレンス」

この章では、CSAPI インタフェースとそのメソッドについて説明します。インタフェースには、クライアント用とサーバ用があります。

- 第4章「イベント通知サービス (ENS) の概要」

ENS は、公開および予約購読サービスで、イベントと仕事のディスパッチャーとして機能します。この API を使って、デフォルトのメカニズムをカスタマイズまたは置き換えることができます。
- 第5章「イベント通知サービス API リファレンス」

この章では、ENS API を構成する関数とそのパラメータについて説明します。
- 第6章「プロキシ認証 SDK の概要」

この章では、サーバに付属する3つの認証スキーマの1つについて説明します。この API を使えば、ユーザのポータルサービスを iPlanet Calendar Server と統合できます。
- 第7章「プロキシ認証 SDK リファレンス」

この章では、SDK を構成する5つの関数について説明します。
- 第8章「シングルサインオン認証」

この章では、この製品で提供される3つの認証メカニズムの1つについて説明します。シングルサインオンは、単一ドメイン設定で機能し、ユーザが1回サインオンするだけで、認証サークルに含まれるすべてのアプリケーションを使用できるようにします。
- 第9章「Web カレンダーアクセスプロトコル (WCAP) の概要」

この章では、WCAP プロトコルについて紹介します。WCAP は、カレンダーデータを送るためのコマンドベースのシステムです。
- 第10章「WCAP コマンド」

この章は、個々のコマンドの詳細情報といくつかの共通トピックから構成されます。

表記上の規則

クーリエ (courier) フォント – コンピュータの画面に表示されるテキスト、または入力するテキストを示す場合に使用されます。また、ファイル名、識別名、関数、および例にも使用されます。

斜体クーリエ (italicized-courier) フォント – インストール固有の情報に置き換えるテキストを表します。これは、その場に応じたサーバのディレクトリパス、ディレクトリ名、アカウント ID を入力する必要がある場合に使用されます。たとえば、このマニュアルでは、ディレクトリパスが次の形式で記載されています。

```
server-root/opt/cal/bin
```

この場合、`server-root` は、サーバをインストールしたディレクトリを表します。このディレクトリは、使用するオペレーティングシステムによって次のように異なります。

- Solaris の場合、デフォルトのインストールでは、ソフトウェアファイルの `server-root` として `/opt/SUNWics5/cal` というパスが使用されます。このため、デフォルトのインストールディレクトリを選択する場合、`server-root/cal` は、次のディレクトリを意味します。

```
/opt/SUNWics5/cal
```

- Solaris 以外の UNIX システムの場合、デフォルトのインストールディレクトリは、`/opt/iPlanet/CalendarServer5/cal` です。この場合、`server-root/cal` は次のディレクトリを意味します。

```
/opt/iPlanet/CalendarServer5/cal
```

- Windows NT の場合、デフォルトのインストールディレクトリは、`c:\Program Files\iPlanet\CalendarServer5\cal` です。この場合、`server-root/cal` は次のディレクトリを意味します。

```
c:\Program Files\iPlanet\CalendarServer5\cal
```

注 このマニュアル中のパス名は、Solaris がインストールされていることを想定して記述されています。iPlanet Calendar Server をほかの UNIX オペレーティングシステムや Windows NT で実行する場合は、Solaris のファイルパスを、使用するオペレーティングシステムに対応するファイルパスに読み換えてください。

関連マニュアル

このマニュアル以外に、次のマニュアルも利用できます。

- 『iPlanet Calendar Server 管理ガイド』
- 『iPlanet Calendar Server インストールガイド』

オンラインマニュアル

オンラインの『iPlanet Calendar Server プログラマリファレンス』が PDF 形式および HTML 形式で用意されています。このマニュアルは、次の Web サイトにあります。

<http://docs.iplanet.com/docs/>

Calendar Server に関するドキュメントはすべて、次の Web サイトにあります。

<http://docs.ipplanet.com/docs/>

アーキテクチャの概要

iPlanet Calendar Server 5.0 は、公開されているインターネット標準を使用した、強力で柔軟性のあるクロスプラットフォームソリューションです。このソリューションを利用して、あらゆる規模のサービスプロバイダは、顧客の個人用およびグループ用のスケジュールカレンダーを管理できます。

この章では、次の項目について説明します。

- iPlanet Calendar Server の概要
- バージョン 5.0 の新機能
- カレンダーサーバサービス
- グループスケジューリング
- 水平方向のスケラビリティ
- 新しいデフォルトクライアント UI: SHTML
- アーキテクチャの基本
- カレンダーデータ
- カレンダーサーバ API (CSAPI)
- イベント通知サービス (ENS)
- プロキシ認証 SDK (authSDK)
- シングルサインオン (SSO)
- Web カレンダーアクセスプロトコル (WCAP)

iPlanet Calendar Server の概要

iPlanet Calendar Server は、展開が簡単な LDAP ベースのソリューションです。このソリューションを利用すれば、インターネットサービスプロバイダおよび通信事業者は、加入者に対して、個人用イベントカレンダー管理機能だけでなくグループスケジューリング機能を提供できます。

iPlanet Calendar Server 5.0 は、さまざまなニーズに応じて構成できるサーバシステムです。このサーバシステムは、スタンドアロンのカレンダーサーバとして独立して構築したり、以降で説明するように複数のインスタンスで構成し、さまざまなサービスをインスタンス間で複製または分割したりできます。25 ページの「水平方向のスケーラビリティ」を参照してください。iPlanet Calendar Server 5.0 では、カレンダー、カレンダーのプロパティ、アクセスコントロール情報、イベント、仕事、およびアラームが格納および管理されます。ユーザ情報の記憶領域は管理されません。システムには、ディレクトリサービスが必要です。外部のサービスを利用する場合は、プラグインを使用します。ディレクトリサービスは、認証、ユーザ設定の格納と検索などを操作するときに使用されます。iPlanet Calendar Server には、LDAP サービスを使用してディレクトリサービスを実行するためのプラグインが付属しています。LDAP 以外のディレクトリサービスをサポートするために、独自のプラグインを使用することもできます。

図 1-1 は、最小構成の iPlanet Calendar Server (iCS) 5.0 システムです。単一の iPlanet Calendar Server インスタンス、ディレクトリサービス、およびイベント通知のサポートから構成されています。

図 1-1 最小構成の iPlanet Calendar Server システム

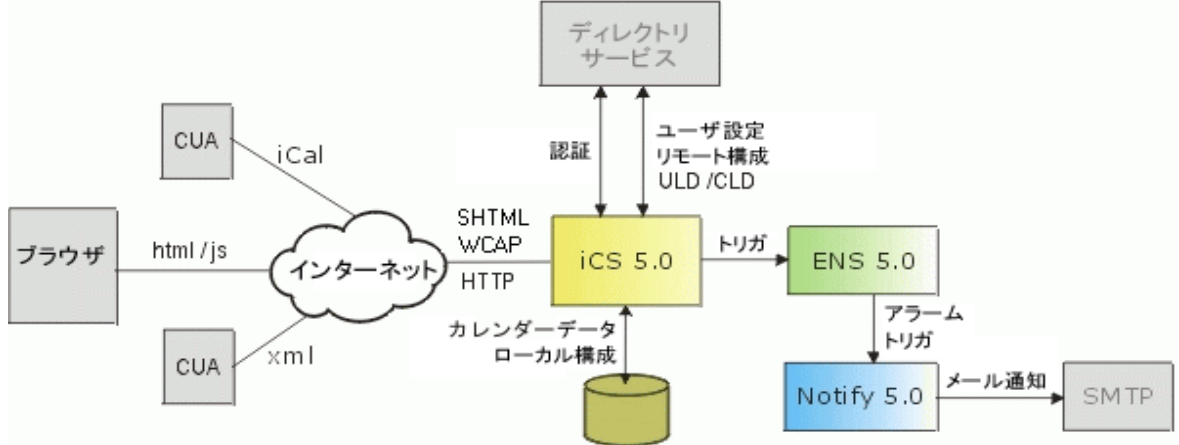


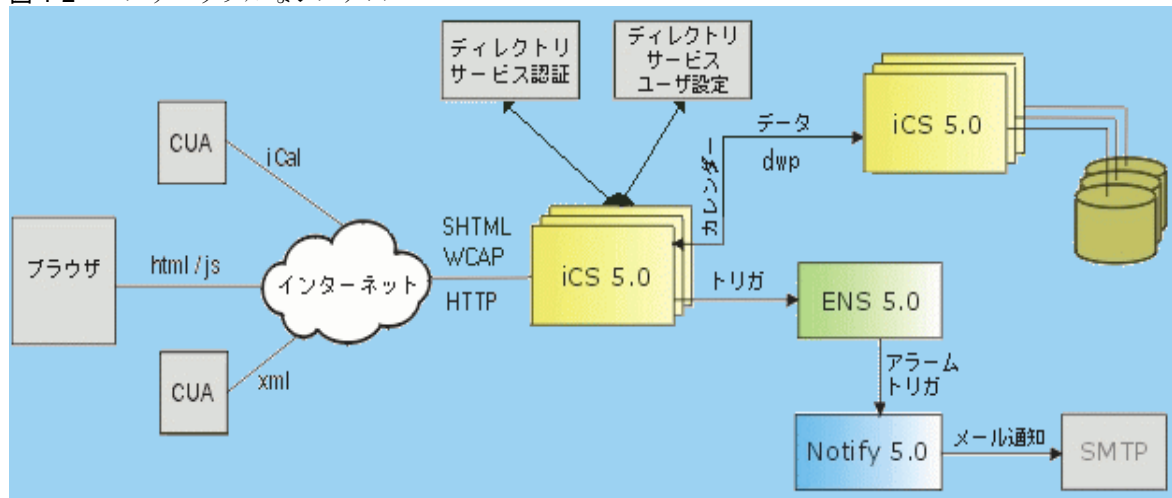
表 1-1 に、図 1-1 に出てくる省略名の説明を示します。

表 1-1 iPlanet Calendar Server システムの用語の説明

CLD	カレンダー検索データベース (Calendar Lookup Database)
CUA	カレンダーユーザエージェント (Calendar User Agent)
ENS	イベント通知サーバ (Event Notification Server)
iCal	iCalendar-RFC 2445

iPlanet Calendar Server は、垂直方向および水平方向に拡張できます。単一マシン上のマルチプロセッサ構成で動作させたり、複数のマシンに分割して動作させたりできます。図 1-2 では、3 つの iPlanet Calendar Server が HTTP 「フロントエンド」 サービスとして使用され、ほかの 3 つの iPlanet Calendar Server が「データベース」サービスとして使用されています。これら 6 つのインスタンスは、別々のマシンで実行できます。この図の中の省略名「dwp」は、データベースワイヤプロトコル (Database Wire Protocol) を表しています。水平方向のスケールビリティによって実現できるさまざまな構成についての詳細は、25 ページの「水平方向のスケールビリティ」を参照してください。

図 1-2 スケーラブルなシステム



機能の要約

iPlanet Calendar Server 5.0 は、一連の新しいインターネットカレンダー標準を完全にサポートするように構築されており、次の長所を持っています。

- **グループスケジューリング** - イベントを作成し、参加者を招集します。出席予定者は、出席するか欠席するかを回答します。参加者がカレンダーサーバに存在しない場合は、グループスケジューリングエンジンからスケジューリングメッセージが IMIP メッセージとして電子メールで送信します。IMIP メッセージについては、RFC2447 に規定されています。
- **インターネット上でのカレンダー操作とスケジューリング** - iCalendar カレンダー操作標準がサポートされているため、イベントがインターネットを介して簡単に共有できる形式になっています。
- **低い所有コスト** - LDAP のサポートにより、サービスプロバイダは、すべての加入者を単一ユーザディレクトリで簡単に集中管理できるうえ、サーバの管理コストを最小限に抑えることができます。また、プロバイダはさらに高度な機能を持つサービスを提案できるプラットフォームを顧客に提供できます。
- **強力なスケーラビリティ** - iPlanet Calendar Server 5.0 は、水平方向にも垂直方向にも拡張できます。

垂直方向の拡張では、最大規模のサービスプロバイダの要件に対応することができ、数百万ユーザの個人用イベントカレンダーを運用するための環境をサポートします。単一マシンをマルチプロセッサ構成で動作させて、マシンの処理能力を最大限活用できます。

水平方向の拡張では、さまざまな組み合わせで複数のコンピュータに分割して動作させることもできます。たとえば、別々の 2 台のマシンで、3 つのカレンダーサーバ HTTP サービスと 3 つのカレンダーサーバデータベースサービスを動作させることもできます。ここでは、HTTP サービスはデータベースサービスを有効に利用します。この新しい機能の詳細は、25 ページの「水平方向のスケーラビリティ」を参照してください。

バージョン 5.0 の新機能

iPlanet Calendar Server 5.0 には、次の新機能が追加されています。

- **グループスケジューリング** - 個人用カレンダー情報を管理するだけでなく、ほかのカレンダーユーザを会議に招集し、出席または欠席の回答を受け取ることができるようになりました。この新しい機能の詳細は、24 ページの「グループスケジューリング」を参照してください。
- **水平方向のスケーラビリティ** - サーバを単一マシンで実行したり、構成方法が豊富なのでサーバプロセスを複数のマシンへ分散したりすることができます。この新しい機能の詳細は、25 ページの「水平方向のスケーラビリティ」を参照してください。
- **デフォルトで提供される新しいクライアントユーザインタフェース** - Calendar Express は、カレンダークライアント UI として同梱されていますが、SHTML を採用することにより、ブラウザの描画と応答時間が高速化され、さらにサイトを速く簡単にカスタマイズできるようになりました。
- **iPlanet Calendar Server 2.x からの移行** - 同梱されているユーティリティを使用して、既存の iPlanet Calendar Server 2.x からデータをインポートできます。バージョン 5.0 を正常にインストールした後で、個別のプログラムを実行すれば、データの移行が完了します。詳細については、『iPlanet Calendar Server インストールガイド』を参照してください。
- **同期化** - iPlanet Synchronization 1.0 ソフトウェアは、ダウンロードして入手できるオプションのアプリケーションです。iPlanet Calendar Server 5.0 のインストールパッケージには組み込まれていません。このソフトウェアを使用して、オンラインカレンダーデータを Palm OS デバイス、Windows CE デバイス、ACT!、Outlook、および Palm Desktop アプリケーションと同期させることができます。

iPlanet Calendar Server 5.0 は、iPlanet Calendar Server 2.x のユーザの視点に基づいて構築されており、複数のマシンによる水平方向のスケーラビリティが導入されています。この新しいアーキテクチャによって、通知機能を使用したグループスケジューリング機能が提供されます。この通知機能を実現するために、iPlanet Calendar Server にはイベント通知サービス (ENS) が付属しています。このアーキテクチャの特長は、さまざまなコンポーネントをカスタマイズして、顧客のニーズに基づいた独自のアプリケーションを開発できることです。

この新しい機能が追加されたため、このマニュアルで説明するインタフェースとツールの一部に根本的な変更が必要になりました。カスタマイズ用の新しいインタフェースが、このマニュアルに追加されています。今回追加されたインタフェースは、プロキシ認証 SDK です。

互換性を保つために、WCAP は引き続き使用できるため、iPlanet Calendar Server 2.x のデータ形式をすべて使用できます。WCAP の詳細は、第 9 章「Web カレンダーアクセスプロトコル (WCAP) の概要」および第 10 章「WCAP コマンド」を参照してください。

カレンダーサーバサービス

iPlanet Calendar Server 5.0 システムは、単一マシン上で実行したり、複数のマシン上に分割して実行したりできる複数のカレンダーサーバデーモンなど、いくつかの実行プロセスで構成されています。実行インスタンスのモジュールの組み合わせは、ics.conf ファイルに格納されるサーバ構成情報で定義されます。このファイルはコマンドライン管理ツールを使用して変更できます。この管理方法で、単一マシン上の単一サービスや複数のマシン上の複数サービスを管理します。管理者は、カレンダーサービスを実行しているマシン以外のマシンから、コマンドをリモートで実行することもできます。表 1-2 に、iPlanet Calendar Server 5.0 の 5 つのデーモンを示します。

表 1-2 iPlanet Calendar Server のデーモン

csadmin	管理サービス。Group Scheduling Engine (GSE) を含み、アラームをモニタする
csdwpd	プロセス間データベースサービス
cshttpd	HTTP サービス。SHTML 要求および WCAP 要求をサポートする
csnotifyd	通知サービス。データベースが存在する各インスタンス内に必要
enpd	イベント通知サービス

csadmind

このサービスは、アラームの通知、グループスケジューリングの要求、データベースのチェックポイントの設定とデッドロックの検出、およびディスク使用状況とサーバレスポンスのモニタを行います。

cshttpd

iPlanet Calendar Server 5.0 では、主要な通信手段として HTTP を使用します。このサービスは、HTTP コマンドを受け取り、データを取得して呼び出し元に返します。5.0 の新しいユーザインタフェースでは、そのコマンドがデフォルトの `.shtml` 拡張子が付いたコマンドを受信し、HTML でフォーマットされたデータを返します。一方、`.wcap` 拡張子が付いた要求を受信した場合は、標準 RFC2445 iCalendar 形式の生の (raw) カレンダーデータ、XML、または HTML 内に組み込まれた JavaScript としてデータをフォーマットして直します。

csnotifyd

`csnotifyd` は通知サービスで、カレンダーに基づいてイベントと仕事の通知を送信し、イベントブローカとしてイベント通知サーバ (Event Notification Server、ENS) を活用します。通知サービスは、アラームイベントを購読します。アラームイベントが発生すると、通知サービスは SMTP アラームメッセージを受信者に送信します。詳細は、37 ページの「イベント通知サービス (ENS)」、第 4 章「イベント通知サービス (ENS) の概要」および第 5 章「イベント通知サービス API リファレンス」を参照してください。

csdwpd

このサービスは、同じシステム内の複数のマシンを取りまとめて、1 つの分散されたカレンダーストアを作成します。このサービスは、iPlanet Calendar Server 5.0 がインストールされている任意のマシン上でバックグラウンドで実行できます。`csdwpd` は、データベースワイヤプロトコル (DWP) に準拠した、カレンダー操作情報への要求を受け付けるサービスとして動作します。

このサービスは、次の条件を満たすサーバで実行してください。

- ローカルにカレンダーのデータを保管する機能があること

- ほかの iPlanet Calendar Server からこのサーバのカレンダーデータにネットワーク経由でアクセスできること

注 このサービスは、必ず高速ネットワーク上で実行してください。データベース間のパイプ (ネットワーク) が低速の場合は、システムの全体的なパフォーマンスが著しく低下することがあります。

enpd

このデーモンは、イベント通知サービスのもう一方の半分にあたり、イベントアラームのブローカとして動作します。csadminデーモンからアラームの通知を受け取り、このイベントの購読依頼を確認し、購読先へのアラーム通知をcsnotifydに渡し、イベントの購読者に通知します。また、csnotifydから購読および購読の取り消し (購読の中止) を受信して格納します。

起動順

iPlanet Calendar Server 5.0 のデーモンは、次に示す特定の順序で起動する必要があります。

1. enpd - ほかの iPlanet サーバによって共有されるイベント登録および通知の汎用サービス。
2. csnotifyd - カレンダーサーバ通知デーモン
3. csadmin - カレンダーサーバ管理デーモン (各サーバマシンにインストールが必要)
4. csdwpd - カレンダーサーバデータベースデーモン (リモートデータベース構成の場合のみ起動される)
5. cshttpd - カレンダーサーバデーモン (少なくとも 1 つ必要)

グループスケジューリング

バージョン 2.x では、独自のカレンダーにイベントと仕事のスケジュールを設定し、そのカレンダーをほかのユーザと共有できましたが、iPlanet Calendar Server 5.0 では、ほかのカレンダーに対してスケジュール設定を要求できます。イベントのスケジュールを設定して参加者を招集し、出席予定者はその要求を受諾または拒否します。出席予定者が受諾または拒否すると、イベントの開催者とすべての出席予定者のカレンダーがサーバによって更新されます。インストールされているデータベースに登録されていない参加者には、電子メールで通知されます。

ディレクトリサーバサービス

デフォルトでは iPlanet Calendar Server は、Netscape Directory Server など、LDAP ディレクトリ内で定義および維持されているユーザをサポートします。また、iPlanet Calendar Server は、CSAPI プラグインの使用もサポートします。このプラグインで、LDAP 以外のディレクトリ内で定義されたユーザにもアクセス権限を設定できます。たとえば、標準の UNIX 認証形式で格納されたユーザ、または Windows NT ユーザ管理データベースに格納されたユーザにアクセス権限を設定できます。

ユーザが LDAP ディレクトリにすでに格納されている場合、ディレクトリサーバを Netscape Directory Server 4.12 にアップグレードするのが、iPlanet Calendar Server を配備するもっとも簡単な方法です。そうすれば、iPlanet Calendar Server システムがそのあとの処理を実行してくれます。また、ディレクトリスキーマを手作業で変更して、iPlanet Calendar Server のデータへのアクセス権限を設定することもできます。iPlanet Calendar Server のディレクトリスキーマを変更する方法についての詳細は、『iPlanet Calendar Server インストールガイド』の「LDAP サーバのインストールと設定」を参照してください。

水平方向のスケラビリティ

iPlanet Calendar Server を複数のマシンにまたがってインストールすることにより、水平方向のスケラビリティを得ることができます。iPlanet Calendar Server は、cshttpd、csadmind、csdwpd、csnotifyd、および enpd というデーモンで構成されており、これらのデーモンをさまざまな構成で実行できるため、強力な柔軟性とスケラビリティが提供されます。

iPlanet Calendar Server 5.0 では、水平方向のスケラビリティを実現するために、データベースワイヤプロトコル (DWP) の csdwpd が内部で使用されています。iPlanet Calendar Server 5.0 ではデフォルトのバークレー DB を使用しており、これはネットワークデータベースでないため、このプロトコルの実装が必要です。DWP プロトコルは、HTTP が基盤になっています。DWP プロトコルは、HTTP の POST または GET コマンドと、直列化されたバイナリデータベース情報を含んだ単一バイナリ MIME 部で構成されます。

今後のリリースでは、カレンダーデータベース API が公開される予定です。そしてデータベースを任意のデータベース技術を使用して実装できます。DWP はネットワークデータベースをサポートしている実装では不要です。

構成

水平方向のスケラビリティを実現するには、各マシンに iPlanet Calendar Server 5.0 のさまざまな組み合わせのインスタンスをインストールします。各システムの基本要件は次のとおりです。

- 各インスタンスに `csadmin` が組み込まれていること
- ほかのすべてのデーモンは、1 つ以上のインスタンスに少なくとも 1 回はインストールされていること

ただし、ローカルデータベース接続を使用して、単一インスタンスをインストールしている場合は例外です。このような単純な構成では、`csdwpd` デーモンは必要ありません。

表 1-3 は、3 つの構成例と、各インスタンスのインストールに必要なサービスの一覧です。図 1-3 に、これらの構成を示します。これ以外の構成にすることも可能です。どの組み合わせが適しているかは、自分自身で判断してください。

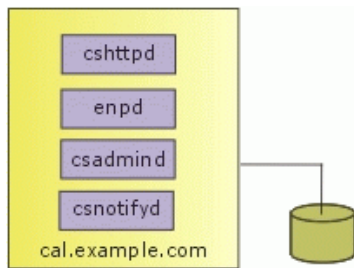
表 1-3 必要なサービスのインスタンス構成

インスタンス	必要なサービス				
	<code>csadmin</code>	<code>cshttpd</code>	<code>csnotifyd</code>	<code>csdwp</code>	<code>enpd</code>
単純な単一インスタンスのインストール。 (ローカルデータベース) 次の図 1-3 を参照	要	要	要	不要	要
HTTP サービスだけのインスタンス (データベースサービスだけのインスタンスと組み合わせて使用) 例: 図 1-4 の「ネットワークフロントエンド、データベースバックエンド」、および図 1-5 の「複数のフロントエンド、複数のバックエンド」の左側のボックスを参照	要	要	不要	不要	不要
データベースサービスだけのインスタンス (HTTP サービスだけのインスタンスと組み合わせて使用) 例: 図 1-4 の「ネットワークフロントエンド、データベースバックエンド」、および図 1-5 の「複数のフロントエンド、複数のバックエンド」の右側のボックスを参照	要	不要	要	要	要

単純な単一インスタンス

iPlanet Calendar Server 5.0 を実行できるもっとも単純な構成です。図 1-3 に示すように、着信した SHTML 要求と WCAP 要求を処理する `cshttpd`、イベントを通知する `enpd` と `csnotifyd`、および必須の `csadmin` から構成されています。データベース全体はローカルです。

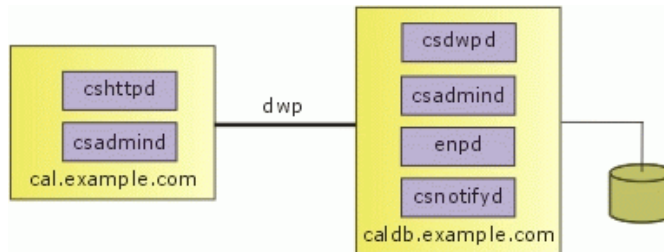
図 1-3 単純な単一インスタンス構成



ネットワークフロントエンド、データベースバックエンド

この構成では、図 1-4 に示すように、HTTP サービスフロントエンドで、ブラウザおよびほかのクライアントがカレンダーサーバに接続します。フロントエンドの名前は、この例では `cal.example.com` です。カレンダーデータに対するすべての要求は、この例では、`caldb.example.com` という名前のデータベースサービスにルーティングされます。以下のことに注意してください。フロントエンドでは、データベースを処理しないため、`cshttpd` と `csadmin` 以外は必要ありません。一方、バックエンドでは、`cshttpd` は必要ありませんが、`csdwpd`、`enpd`、および `csnotifyd` が必要です。

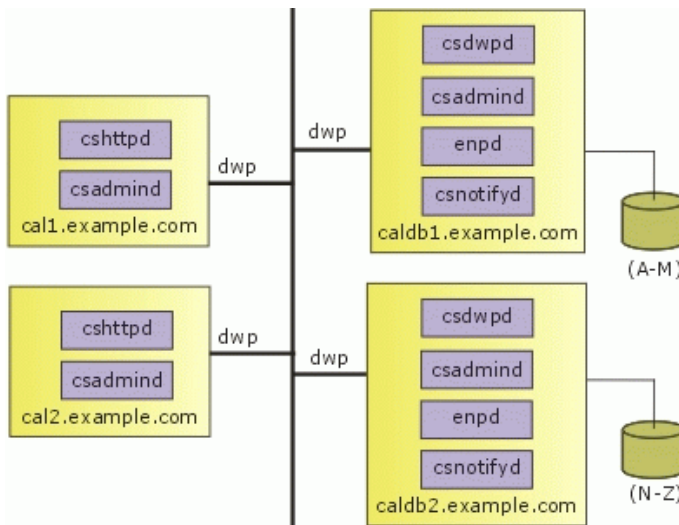
図 1-4 ネットワークフロントエンド、データベースバックエンド



複数のフロントエンド、複数のバックエンド

この構成では、図 1-5 に示すように、外部の何らかのメカニズムによって、クライアントはフロントエンドの HTTP サービスのどれかにルーティングされます。ログインしたときに返されるセッション ID は、ログインしたホストのみで有効です。このセッション ID に対するすべての要求は、同じホストにルーティングされる必要があります。異なるホストにルーティングされた場合、ユーザは再度ログインする必要があります。この例では、データベースが分割されており、カレンダー A-M は caldb1.example.com サーバにルーティングされ、カレンダー N-Z は caldb2.example.com サーバにルーティングされます。CSAPI プラグインによって、カレンダー ID とカレンダーが存在するサーバの名前が対応付けられます。iPlanet Calendar Server 5.0 に実装されているデフォルトの CSAPI では、カレンダー ID をサーバ名に関連付けるアルゴリズムが使用されています。

図 1-5 複数のフロントエンド、複数のバックエンド



新しいデフォルトクライアント UI: SHTML

iPlanet Calendar Server 5.0 では、バージョン 2.x で実装されていた WCAP プロトコルを使用するデフォルトクライアント UI (ユーザインタフェース) は実装されていません。WCAP プロトコルでは、HTML と JavaScript の組み合わせが生成され、クライアントに渡されて処理されていました。新しい SHTML コマンドを使用することによって、サーバですべての処理が行われてから、HTML 形式だけの出力が生成され、クライアントに渡されます。新しい SHTML コマンドでは、XML プロトタイプ定義と XSL スタイルシートテンプレートを使用して、HTML が生成されます。ユーザインタフェース内に表示されるビューおよびダイアログごとに、1 つ以上の対応するテキストファイルのペアがあります。各ペアは、.xml ファイルと .xsl ファイルから構成されます。プログラマは、これらのファイルの一方または両方を変更または置換して、ユーザインタフェースをカスタマイズできます。

バージョン 2.x 用カスタムユーザインタフェースをすでに開発している場合は、変更しないでそのまま使用できます。iPlanet Calendar Server 5.0 は、iPlanet Calendar Server 2.x との下位互換性が保証されています。

WCAP は、未処理のカレンダーデータを取り出す唯一の方法です。書式設定されていない生のカレンダー情報が必要な場合、クライアントは WCAP 形式で要求を送信する必要があります。

アーキテクチャの基本

iPlanet Calendar Server 5.0 は、共有ライブラリの集合として実装されています。これらの共有ライブラリは、さまざまな組み合わせで結合されて、実行可能デーモンである `cshttpd`、`csdwpd`、`csadmind`、および `csnotifyd` が動作します。

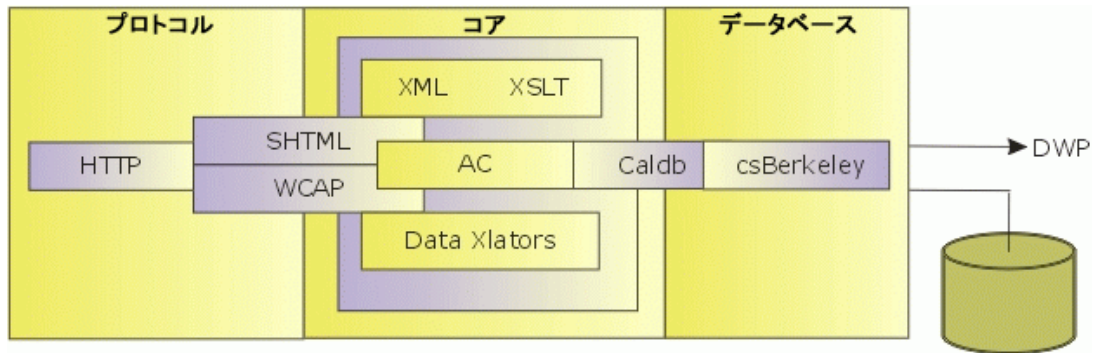
イベント通知サービス (ENS) のデーモンである `enpd` サービスは、iPlanet Calendar Server 5.0 に付属しており、個別にインストールされます。ENS についての詳細は、この章の 37 ページの「イベント通知サービス (ENS)」と、このマニュアルの第 4 章および第 5 章を参照してください。

共有ライブラリは、サブシステムと呼ばれる次の 3 つの主要カテゴリに分類されます。

- プロトコル
- コア
- データベース

図 1-6 は、これらのサブシステム間の論理的な流れを示しています。この図の後で、各サブシステムについて説明します。

図 1-6 サーバのアーキテクチャ



SHTML と WCAP

SHTML と WCAP は、HTTP を基盤にしています。要求は、HTTP プロトコルレイヤー経由で受信されます。これは、カレンダー要求をサポートするために必要な最小構成の HTTP サーバです。クライアントは、SHTML または WCAP のどちらかを使用して要求を送信します。WCAP は、一部の管理コマンドを除くすべてのサーバコマンドを実行できる、公開プロトコルです。WCAP は、書式設定されていない生のカレンダー情報を必要とするクライアントでも使用できます。また、JavaScript 対応のユーザインタフェースを入手するときにも使用できます。このユーザインタフェースは、iPlanet Calendar Server 2.x で使用できる唯一のインタフェースでしたが、バージョン 5.0 では、新しい SHTML 対応のユーザインタフェースに置き換えられました。この新しいユーザインタフェースは、XML および XSLT 仕様に基づいて設計されており、コマンドに応答してユーザインタフェースを生成します。UI ジェネレータは、要求を受信すると、アクセスコントロールに応じて、カレンダーとユーザデータが含まれる XML 仕様のドキュメントツリーを作成します。次に、XSLT 仕様が、ドキュメントデータツリーを走査し、HTML を作成します。この設計の利点の 1 つは、クライアントとサーバ間の対話が少なくなり、送信される総ビット数も少なくなることです。以前の iPlanet Calendar Server 2.x では、組み込みの JavaScript を返し、この JavaScript がクライアント側で HTML を生成していました。XML/XSLT 方式では、ブラウザ側で速く描画が行える出力を生成します。

以前のバージョンとの互換性を保つために、iPlanet Calendar Server 2.x のデザインは引き続き機能します。つまり、デフォルトで、WCAP 要求に対して HTML テキストと JavaScript の組み合わせを返します。iPlanet Calendar Server 2.x と同様に、.wcap 拡張子を使用するコマンドは、HTML にくるまれた XML、または HTML にくるまれた iCalendar として出力を要求します。

WCAP プロトコルについては、第 9 章「Web カレンダーアクセスプロトコル (WCAP) の概要」を参照してください。

コア

コアサブシステムは、アクセスコントロールサブシステム、UI ジェネレータサブシステム (XML と XSLT を使用する SHTML、またはデータ変換機能を使用する WCAP)、および Caldb サブシステムで構成され、それぞれ別の区分に配置されています。CSAPI プラグインは、コアサブシステム内にあります。

コマンドは、プロトコルによって処理されてコアサブシステムに送信され、コアサブシステムで実行されます。コマンドには、次の 4 つの種類があります。

- **カレンダー操作** - データベースの処理を必要とするカレンダー操作の場合、要求はプロトコルによってコアのアクセスコントロールサブシステムに送信されます。アクセスコントロールサブシステムは、汎用の Caldb サブシステムを使用して、カレンダーの読み込みおよび書き込み操作を行います。

Caldb サブシステムは、データベース技術固有のサブシステムを使用して、操作を行います。このバージョンの iPlanet Calendar Server では、バークレーデータベースが実装されているため、Caldb サブシステムは、ローカルデータベースファイルを処理したり、適切なマシンに対するデータベースワイヤプロトコル (DWP) 要求を作成したりする機能を持ちます。

このデータベースサブシステムは、低レベルの形式でデータを返します。コアの UI ジェネレータ (SHTML または WCAP) は、この低レベルデータを必要な出力に変換します。.shtml 拡張子付きのコマンドの場合は、デフォルトで HTML 形式を生成します。.wcap 拡張子付きのコマンドの場合は、要求された出力形式を返します。WCAP は、このサーバでサポートされる唯一の公開プロトコルです。書式設定されていないカレンダーデータを取り出すには、WCAP を使用します。

注 HTML/js が iPlanet Calendar Server 2.x のデフォルト UI 出力でしたが、iPlanet Calendar Server 5.0 では、.shtml に置き換まりました。

- **ユーザの属性** - ユーザ属性に関する要求は、コアのディレクトリサービスに送信されます。iPlanet Calendar Server 5.0 では、ディレクトリサービスなどの外部サービスを取得するときにプラグインが使用できます。このバージョンには、LDAP サービスのプラグインが付属しています。インストールをカスタマイズするには、LDAP 以外のディレクトリサービスをサポートするプラグインを別に作成します。
- **認証** - 認証に関する要求は、コアのデフォルト LDAP ディレクトリサービスに送信されます。iPlanet Calendar Server 5.0 には、次の 3 つの認証オプションがあります。
 - CSAPI 認証。第 2 章「Calendar Server API (CSAPI) の概要」を参照。内部の認証
 - プロキシ認証 SDK。第 6 章「プロキシ認証 SDK の概要」を参照。外部プラグイン認証
 - シングルサインオン認証。第 8 章「シングルサインオン認証」を参照。単一ドメイン内で、アプリケーション間の信頼サークルが形成されるCSAPI 認証およびプロキシ認証 SDK には、カスタマイズ可能な API があります。
- **その他** - その他のサービスに関する要求は、ほかのコアサブシステムに送信されません。

データベース

iPlanet Calendar Server 5.0 では、Sleepycat のバークレー DB を使用した csBerkeley サブシステムが使用されます。このデータベースの API は公開されていません。

カレンダーデータ

この節では、カレンダーデータについて説明します。

- カレンダーデータの形式
- グループ
- イベントフィード
- カレンダーデータの交換
- カレンダーのユーザ設定
- カレンダーアクセスコントロール
- サポートされるコーディング方式

カレンダーデータの形式

カレンダーのデータ形式は、IETF iCalendar の標準である RFC-2445 に準じて設計されています。アクセスコントロールレイヤによって、iCalendar コンポーネントの集合であるカレンダーが管理されます。iCalendar コンポーネントは、イベント、仕事、およびアラームで構成されます。カレンダーごとに、1 人の主な所有者が割り当てられます。第 1 所有者以外に、複数の所有者を割り当てることもできます。ユーザ属性は、外部メカニズムによって管理されます。デフォルトのメカニズムは、LDAP です。

グループ

グループとは、名前付きカレンダーリストのことです。ユーザは、カレンダーを購読します。ユーザは、アクセスコントロールに応じて、購読したカレンダーを表示および変更できます。また、ユーザはグループを作成できます。カレンダーグループを使用すれば、カレンダーを並べて表示したりカレンダーの所有者をイベントに招集したりするたびに、カレンダーリストを再指定または選択する必要がなくなります。

グループを使用すれば、複数のカレンダーソースを単一カレンダーに集約して表示することができます。たとえば、所有しているカレンダー、部門のカレンダー、祝日カレンダー、「最新アクションビデオのリリース」カレンダーから構成される、デフォルトのカレンダー表示を作成できます。それらをまとめて「カレンダーグループ」と呼びます。

イベントフィード

このインフラストラクチャによって、iCalendar または XML などのインポート形式で、イベントデータをデータベースにリアルタイムに送信できます。イベントデータは、カレンダーから送信されます。たとえば、アメリカンフットボールチームのシーズンスケジュール、コンベンションセンターのスケジュール、コンサートのスケジュールなど、興味を持つイベントのスケジュールのカレンダーが使われます。

iPlanet Calendar Server には、イベントフィードおよびデータベースからカレンダーデータを取り出すツールが用意されています。ユーザは、このツールを利用して、イベント情報を表示し、取り出すことができます。これらのイベントをカレンダー表示上で階層化すれば、ほかの会社で管理されるさまざまなイベント情報を豊富に入手できます。

カレンダーデータの交換

すべてのカレンダーとイベントは、URL として参照できます。これらの URL リンクは、電子メールメッセージと Web ページ内に組み込むことができます。リンク位置をクリックすれば、月次カレンダー表示、週次カレンダー表示、日次カレンダー表示、または特定のイベントを表示できます。カレンダーが公開されている場合は、ユーザはログインする必要はありません。

iPlanet Calendar Server では、サーバ側の電子メールアラームがサポートされます。このアラームは、受取人リストに送信できます。電子メールメッセージの書式は、任意に設定できます。メッセージの書式は、ユーザまたはカレンダーの属性としてではなく、サーバの属性として管理されます。iPlanet Calendar Server 5.0 では、ITIP/IMIP 標準 (RFC-2446、RFC-2447) が制限付きでサポートされます。つまり、イベントに対して、ITIP メソッドの PUBLISH、REQUEST、REPLY、および CANCEL がサポートされます。

カレンダーのユーザ設定

iPlanet Calendar Server は、ユーザ設定と呼ばれる属性に従って、個々のユーザのカレンダー操作情報の表示を生成します。ユーザ設定は、カレンダー設定とは異なり、情報のユーザインタフェース表示方法を示しています。ユーザ設定には、電子メールのアドレス、ユーザ名、カレンダー情報を描画するときを使用する色などがあります。ユーザ設定の完全なリストについては、第 10 章「WCAP コマンド」の `get_userprefs` および `set_userprefs` コマンドの説明を参照してください。

カレンダーアクセスコントロール

iPlanet Calendar Server 5.0 では、グループスケジューリングを使用するときにカレンダーにアクセスできるユーザが、アクセスコントロールで決定されます。アクセスコントロールエントリ (Access Control Entry、ACE) の文字列を使用して、ユーザに許可されるカレンダーへのアクセス特権の種類を指定します。ACE の文字列は、アクセスコントロールのカレンダープロパティである `acl` (アクセスコントロールリスト) に格納され、カレンダーのアクセスコントロールを一元管理します。カレンダープロパティへの書き込み権限を持つユーザだけが、これらの文字列を変更できます。デフォルトでは、Calendar Server の管理者とカレンダーの主な所有者だけが、プロパティの書き込み権限を持ちます。iPlanet Calendar Server のアクセスコントロールモデルでは、ほかのユーザの代理としてふるまう機能もサポートされます。たとえば、管理者のアシスタントに対して、ほかのユーザの代理としてふるまう権限を設定すると、管理者のアシスタントの代理としてイベントの招集、取り消し、応答などを行うことができます。

アクセスコントロールについては、次の節で詳しく説明します。

- 「アクセスコントロールエントリ (ACE)」の詳細は、第 10 章「WCAP コマンド」を参照
- 構成の設定についての詳細は、「iPlanet Calendar Server 管理ガイド」を参照

サポートされるコーディング方式

データの格納のために、コアサブシステムは、入力を Caldb サブシステムで使用されるバイナリ形式に変換します。iPlanet Calendar Server では、次のコーディング方式をサポートしています。

- HTML (デフォルト)
- HTML/JavaScript
- XML
- iCalendar

UI のビューとダイアログ用に独自の XSL 変換を開発して、ほかの形式を追加できます。また、CSAPI を使用して、WCAP プロトコルの変換 DLL または共有ライブラリを開発できます。(第 2 章「Calendar Server API (CSAPI) の概要」を参照)

カレンダーサーバ API (CSAPI)

CSAPI は、COM と似たインタフェースです。CSAPI を使用すれば、サーバの一部をカスタマイズして実装できます。次の機能を変更するには、CSAPI を使用します。

- アクセスコントロール (50 ページの「csIAccessControl」を参照)
- 認証 (52 ページの「csIAuthentication」を参照)
- カレンダー検索 (57 ページの「csICalendarLookup」および 66 ページの「csIQualifiedCalidLookup」を参照)
- データ形式の変換 (60 ページの「csIDataTranslator」を参照)
- ユーザ属性へのアクセス (50 ページの「csIAccessControl」を参照)

たとえば、デフォルトでは、サーバの認証メカニズムには LDAP が使用され、ユーザ設定はデフォルトで LDAP に保存されています。LDAP に基づいていない、ユーザ認証やユーザ設定の保存のための既存のインフラストラクチャがある場合は、CSAPI を使用してデフォルトのメカニズムを置き換え、既存の認証およびディレクトリサービスを使用します。加えて、次の 3 つのカスタマイズ可能なインタフェースがあります。

- csIPlugin。起動時に、プラグインモジュールに関する情報をサーバに提供します
- csICalendarServer。実行中のサーバインスタンスのバージョン情報が提供されます
- csIMalloc。メモリ割り当てスキーマ

CSAPI については、第 2 章「Calendar Server API (CSAPI) の概要」および第 3 章「CSAPI リファレンス」で詳しく説明します。

イベント通知サービス (ENS)

iPlanet Calendar Server では、イベント通知サービス (ENS) は主としてアラームディスプレイパッチャーとして使用されます。アラームディスプレイパッチャーによってアラームキュー上のイベントが検出され、イベントの通知が購読者に送信されます。ENS API を使用すれば、iPlanet Calendar Server 5.0 で使用される公開および購読に関する次の機能を変更できます。

- イベントを購読する
- イベントの購読を中止する
- イベントの購読者に通知する

ENS の概要については、第 4 章「イベント通知サービス (ENS) の概要」を参照してください。

各分野の関数の詳細については、第 5 章「イベント通知サービス API リファレンス」を参照してください。

プロキシ認証 SDK (authSDK)

authSDK は、iPlanet Calendar Server 5.0 に用意されている 3 つのユーザ認証ツールの 1 つです。authSDK を使用すれば、既存のポータルサービスを iPlanet Calendar Server 5.0 に統合できるため、ユーザがアプリケーションにアクセスするたびに再認証する必要がなくなります。authSDK は、DLL/共有オブジェクトライブラリの libicsexp10 内にパッケージ化されている 5 つの関数と、ヘッダーファイルの expapi.h から構成されています。これらの関数によって、次の 3 つの単純な処理が行われます。

- 初期化
- 検索
- クリーンアップ

また、標準以外のポートを使用可能にする関数と、トラブルシューティング用の authSDK バージョン番号を取得する関数があります。

注 iPlanet Calendar Server と authSDK との間で接続が確立されると、そこで設定された関係が信頼の 1 つになります。したがって、ユーザがログインして authSDK で認証されると、カレンダーサーバは、プロキシによって生成された証明書を受け入れます。この証明書がすべてのアプリケーションで使用されます。

authSDK の詳細については、第 6 章「プロキシ認証 SDK の概要」および第 7 章「プロキシ認証 SDK リファレンス」を参照してください。

シングルサインオン (SSO)

シングルサインオン (Single Sign-on, SSO) は、iPlanet Calendar Server 5.0 に用意されている 3 つの認証メカニズムの 1 つです。シングルサインオンを使用するには、クライアントのブラウザで cookie がサポートされ、サーバで HTTP がサポートされている必要があります。シングルサインオンは、その他の認証メカニズム、セッション管理、およびリソースアクセスコントロールから独立しています。

シングルサインオンを使用すると、アプリケーション間の信頼サークルが形成され、信頼サークル内で cookie が共有され、ユーザ認証が交換されます。必要に応じて、各アプリケーションに独自の検証インタフェースを設定できます。ただし、各検証機能は、ほかのアプリケーションの検証機能のルーチンで認識される cookie を格納します。

このメカニズムには、いくつかの制限事項があります。まず、アプリケーションごとに、検証プロトコルが実装されている必要があります。また、信頼されているすべてのアプリケーションは、同じドメインに配置されている必要があります。さらに、異なる識別情報に切り替えるときは、ユーザがブラウザを再起動する必要があります。これは、各ブラウザセッションで複数のユーザ ID がサポートされないためです。

シングルサインオンの詳細は、第 8 章「シングルサインオン認証」を参照してください。

Web カレンダーアクセスプロトコル (WCAP)

WCAP はコマンドベースのシステムで、カレンダー操作データを転送するクライアントの要求とサーバの応答で構成されます。WCAP 2.0 では、HTTP 経由でカレンダー操作データが返されます。WCAP コマンドを使用すると、カレンダーコンポーネント、ユーザ設定、カレンダーのプロパティ、およびタイムゾーンなどのカレンダー情報を取得、削除、および変更できます。時刻、文字列、パラメータなどは、ほかの方法で指定されていない限り、RFC2445、RFC2446、および RFC2447 の仕様に準拠します。

WCAP は、iPlanet Calendar Server 5.0 に用意されている 3 つのユーザ認証メカニズムの 1 つです。デフォルトでは、プレーンテキストのパスワードとユーザ名を使用します。WCAP で使用する認証メカニズムは、置換または拡張できます。第 3 章「CSAPI リファレンス」の 52 ページの「csIAuthentication」の節を参照してください。

WCAP では、クライアントの要求とサーバの応答で、次のデータ形式がサポートされます。

- プレーンテキスト形式のカレンダーデータ (HTML のみ)。これは、UI の新しいデフォルトの形式です
- text/calendar 形式のカレンダーデータ (iCalendar)
- text/xml 形式のカレンダーデータ。XML スタイルの iCalendar
- text/js として、JavaScript オブジェクトが組み込まれたカレンダーデータ。これは、iPlanet Calendar Server2.x ユーザインタフェースのデフォルトです

Calendar Server API (CSAPI) の概要

この章では、Calendar Server API (CSAPI) の概要を説明します。CSAPI は、iPlanet Calendar Server 5.0 の機能を変更または強化できる高性能のプログラム可能インタフェースのセットです。CSAPI により、速度、メモリ使用量、および負荷の面で、システムの実行可能ファイルもどの言語のスクリプトも上回る、非常に高速な実行時共有オブジェクトを作成できます。これらの要素はすべて、ハイエンドシステムのスケラビリティに貢献します。

この章は、以下の節で構成されています。

- CSAPI のアーキテクチャ
 - スレッドに対する安全の要件
 - 依存性
- CSAPI の使用法
 - CSAPI モジュールの読み込み
 - プラグインインタフェース
 - クライアントおよびサーバの API
- CSAPI のサンプル

CSAPI のアーキテクチャ

CSAPI は、Calendar Server の機能への共有オブジェクト実行時インタフェースの集合です。プラグイン CSAPI モジュールを使用して、着信した要求や応答のためのサーバデータを操作できます。このアーキテクチャにより、サーバは未知のデータに対する単なるゲートウェイとして動作できます。また、このアーキテクチャにより、動的なログ記録や統計情報の追跡、外部認証スキーマ、ユーザ属性の操作、およびその他の機能も実装できます。

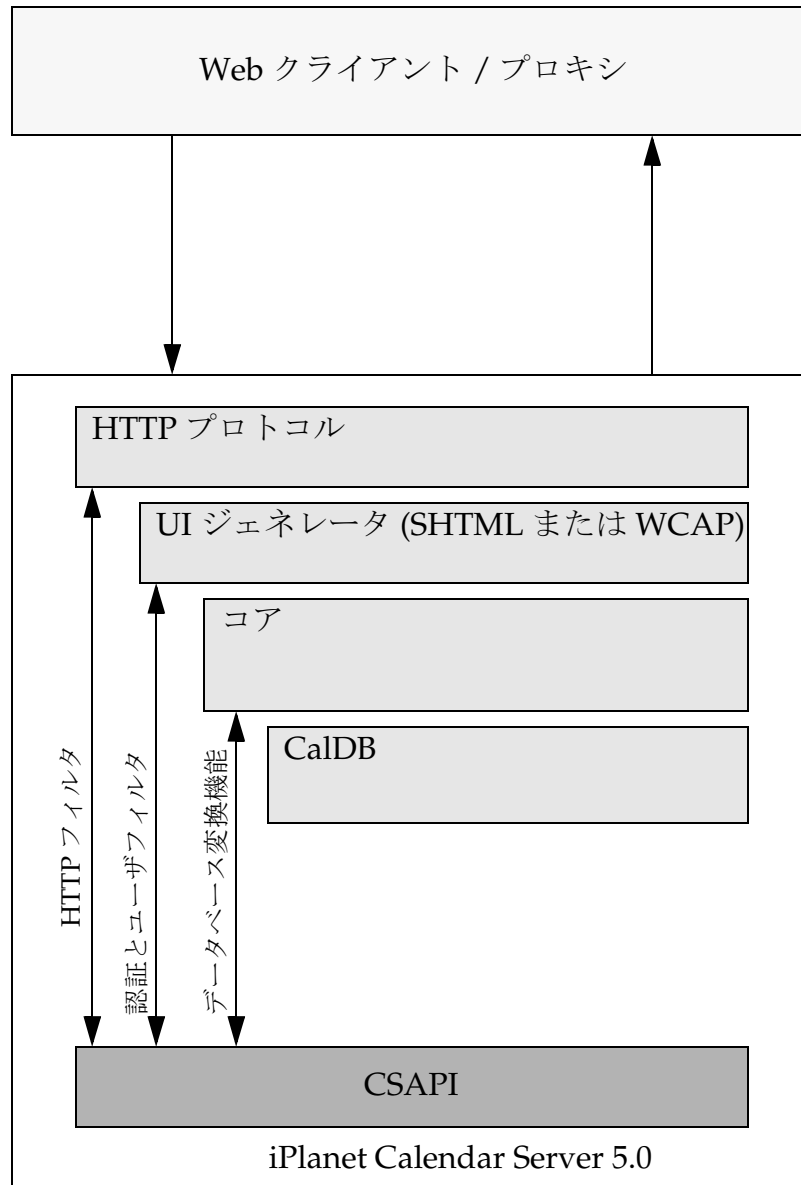
次の図 2-1 に、iPlanet Calendar Server 内での CSAPI モジュールとほかのサブシステムとの関係を示します。CSAPI モジュールは、どの機能グループをサポートするかによって、データのフォーマット、認証、ディレクトリサービスなどの iPlanet Calendar Server の 1 つ以上の領域と連携して動作します。

モジュールとは、UNIX 上では共有オブジェクト (.so ファイル) を、Windows NT ではダイナミックリンクライブラリ (.dll ファイル) を指します。プログラマが提供する各モジュールは、このマニュアルで定義されている 1 つ以上の CSAPI インタフェース (または純粋な仮想基底クラス) を実装します。第 3 章「CSAPI リファレンス」を参照してください。クライアント側の各インタフェースが、iPlanet Calendar Server の 1 つの機能領域に対応します。モジュールに実装するものによって、iPlanet Calendar Server とその領域のネイティブな機能を拡張したり置き換えたりすることができます。

サーバ側の API セットにより、CSAPI モジュールは、サーバのバージョン情報を取得したり、サーバの高速メモリ割り当てメカニズムを使用したりできます。

このシステムには、各 CSAPI インタフェースのプラグインがデフォルトのコードとともに含まれています (このプラグインをテンプレートとして使用して、独自のプラグインを作成できます)。また、サポートされているすべてのライブラリとヘッダーも装備されています。

図 2-1 CSAPI とほかのサブシステムとの関係



スレッドに対する安全の要件

数千ものスレッドが随時 1 つのモジュールにアクセスできるので、CSAPI モジュールのプラグインはスレッドに対して安全 (thread safe) である必要があります。スレッドに関する問題を認識できない (thread-aware でない) プラグインの場合は、プラグイン内の関数呼び出しレベルで単純な情報収集 (モニタ) のみを行なってください。NSPR スレッドの詳細は、mozilla.org の NSPR リファレンスマニュアルを参照してください。URL については、この章の「依存性」の節を参照してください。

依存性

CSAPI は、UNIX システムおよび Windows NT システム用の C および C++ のインタフェースです。CSAPI は、NSPR (Netscape Portable Runtime) と、インタフェースディスプレイパッチ用の XPCOM を使用します。NSPR は、オペレーティングシステムサービスへのプラットフォームに依存しない API である mozilla.org ソースコードの一部です。

NSPR のマニュアルについては、次の mozilla の技術マニュアルのサイトを参照してください。

<http://www.mozilla.org/docs/refList/refNSPR>

XPCOM のマニュアルについては、次のサイトを参照してください。

<http://www.mozilla.org/projects/xpcom>

異なるプラットフォーム上で実行する必要があるシステムでは、プラットフォームに依存しない C データ型と実行時関数に NSPR を使用する必要があります。iPlanet Calendar Server は、XPCOM C++ API (QueryInterface) を使用して、特定のモジュールが実装しているインタフェースを見つけます。

CSAPI の使用法

次の節では、プログラマが提供するプラグインをシステムがどのように読み込み、使用するか説明します。システムには、デフォルトのプラグインが付属しています。デフォルトのプラグインはその一部、またはすべてを拡張したり置き換えたりすることができます。

CSAPI モジュールの読み込み

iPlanet Calendar Server は、起動時に CSAPI モジュールを `cal/bin/plugins` ディレクトリから読み込み、サーバの停止時に読み込み解除します。すべてのプラグインモジュールは、このディレクトリに存在し、ファイル名の先頭に `cs_` を持つ必要があります。

サーバは、サーバ起動時に動的に読み込まれるモジュールを決めるために、`ics.conf` を検査します。`csapi.plugin.loadall` の設定の値が `y` の場合、サーバは `cal/bin/plugins` ディレクトリ内の名前が接頭辞 `cs_` で始まるすべての共有オブジェクトを読み込みます。値が `n` の場合、設定はプラグインによって異なります。`ics.conf` の設定の詳細は、『iPlanet Calendar Server 管理ガイド』を参照してください。

特定のプラグインの読み込みを指定するには、`csapi.plugin.loadall` を `n` に設定する必要があります。このほかに2つの設定を使用する必要があります。それは、値を `y` に設定した `csapi.plugin.plugin name` と、値にプラグインの名前を設定した `csapi.plugin.plugin name.name` です。たとえば、`my_lookup` という名前のカレンダー検索プラグインの場合、設定は次のとおりです。

```
csapi.plugin.loadall = "n"
csapi.plugin.calendarlookup = "y"
csapi.plugin.calendarlookup.name = "my_lookup"
```

設定の `plugin name` の部分は、両方の設定で一致している必要がありますが、`.name` 設定の値と同じである必要はありません。したがって、`cs_myown_plugin` というプラグインを作成する場合、設定を `csapi.plugin.anyname` および `csapi.plugin.anyname.name` と呼ぶことができます。`csapi.plugin.anyname.name` の値は、「`cs_myown_plugin`」になります。

iPlanet Calendar Server は、NSPR 関数 `PR_LoadLibrary()` を使用して起動時に共有オブジェクトを読み込み、関数 `PR_UnloadLibrary` を使用して停止時に共有イメージを読み込み解除します。共有オブジェクトがメモリに読み込まれると、iPlanet Calendar Server は関数 `PR_FindSymbol` を使用してサーバが把握している API の実装へのエントリポイントを探します。

プラグインインタフェース

すべての CSAPI プラグインは、XPCOM 仕様で要求されている、エクスポートされる唯一のシンボル `NSGetFactory` をサポートします。このエン트리ポイントから、Calendar Server は XPCOM メソッドの `QueryInterface` を呼び出して、`csIPlugin` インタフェースを実装しているオブジェクトを見つけます。これにより、サーバはプラグインのバージョン、説明、およびベンダー情報を照会できます。このインタフェースはオプションです。ただし、サーバがバージョン管理できるようにこのインタフェースの実装をお勧めします。

クライアントおよびサーバの API

CSAPI は、クライアント API とサーバ API の 2 つに分けられます。

表 2-1 に、1 つ以上のプラグインで実装できる CSAPI クライアントインタフェースを一覧表示します。

表 2-1 CSAPI クライアントの API

CSAPI モジュールインタフェース	説明
<code>csIAccessControl</code>	デフォルトのアクセスコントロールメカニズムを拡張または置き換える
<code>csIAuthentication</code>	ログイン認証メカニズムを拡張または置き換える
<code>csICalendarLookup</code>	デフォルトのカレンダー検索メカニズムを拡張または置き換える
<code>csIDataTranslator</code>	着信データおよび発信データの形式変換を拡張または置き換える
<code>csIPlugin</code>	モジュールのバージョン管理情報と内容に関する情報を提供する
<code>csIUserAttributes</code>	ユーザ属性の格納および検索メカニズムを拡張または置き換える
<code>csIQualifiedCalidLookup</code>	指定された完全修飾の URL のカレンダー ID を検索する

これらのインタフェースの詳細は、第 3 章「CSAPI リファレンス」を参照してください。

すべてのインタフェースには、実装する必要がある次の初期化メソッドがあります。

```
Init (nsISupports * aServer);
```

サーバは、新しく読み込まれたモジュールにインタフェースを登録した直後にこのメソッドを呼び出します。モジュール内では、サーバが返すパラメータ `aServer` をバインドし、このパラメータを使用してサーバインスタンスを参照できます。カスタムプラグインでは、`QueryInterface` メソッドを使用して、表 2-2 に一覧表示したサーバインタフェースを取得できます。

表 2-2 CSAPI サーバの API

サーバインタフェース	説明
<code>csICalendarServer</code>	バージョン番号などの全般的なサーバ情報を提供する
<code>csIMalloc</code>	サーバのメモリ割り当てメカニズムへアクセスする

サーバに対する照会の例

次の例では、`Calendar Server` のバージョンを確認します。この例では、以下の処理を行う方法を示します。

- `Init` メソッドから戻された参照をバインドする
- サーバのインタフェースを照会する
- そのインタフェースでサーバメソッドを呼び出す
- サーバへの参照を解放する

```
NS_IMETHODIMP csDataTranslator ::Init(nsISupports * aServer)
{
    nsresult res = NS_COMFALSE ;
    PRUint32 min, maj;
    csICalendarServer * cs;
    /* CalendarServer の QueryInterface。呼び出しが成功したら、サーバは
    リファレンスカウントを増分します */
    if (aServer)
        res = aServer->QueryInterface(kICalendarServerIID, (void**) &cs);
    /* サーバへの参照の取得に成功したら、バージョンを確認する */
    if (NS_SUCCEEDED(res)) {
        cs->GetVersion(maj, min);
        if (min > 0 && maj >= 1)
            res = NS_OK;
        else
            res = NS_COMFALSE;
    }
    /* サーバインスタンスに対するこの参照を解放する */
    cs->Release();
}
return res;
}
```

CSAPI のサンプル

csapi/samples ディレクトリには、3つの CSAPI インタフェースのコーディング例が含まれています。独自の CSAPI モジュールを作成する際、それらのファイルをテンプレートとして使用できます。

以下のサンプルモジュールが用意されています。

表 2-3 CSAPI インタフェースのサンプル

CSAPI モジュールのサンプル	説明
authentication	<p>このサンプルでは、デフォルトのログイン認証メカニズムを置き換え、ローカルの認証を使用してユーザを確認する。このサンプルは、Solaris と Window NT で実行できる</p> <p>Solaris では、このサンプルは pam ライブラリを使用してローカルの /etc/passwd ファイルまたは NIS に対して認証を行う</p> <p>Windows NT では、このサンプルは、Microsoft クライアントに対して認証を行う WIN32 API LogonUser を使用する。このサンプルを NT で正常に実行できるようにするには、管理者はバッチジョブを介したユーザのログオン権を有効にする必要がある。これは、ユーザー マネージャ管理ツールの「原則 / ユーザーの権利」セクションから実行できる</p>
dataTranslator	<p>このサンプルは、着信データおよび発信データのデフォルトの形式変換を拡張または置き換える。このサンプルは、icalendar データを Microsoft Outlook の CSV 形式に変換する方法を示している。CSV 形式は単純な行指向のファイルで、各項目にそれぞれの行があり、プロパティはコンマで区切られている</p>
userAttributes	<p>このサンプルは、ユーザ属性の格納および検索用のデフォルトのメカニズムを拡張または置き換える。このサンプルは、ローカルのユーザ設定を格納するパークレーデータベースの使い方を示している。このコードは、サポートされているすべてのプラットフォームで実行できる。データベースは、単純な表形式のキーと値のペアで構成される。ユーザ設定を格納するためのキーは、\$user.\$pref として格納された文字列である。キーは一意である必要がある</p>

CSAPI リファレンス

ここでは、それぞれが API である 9 つの CSAPI インタフェースについて詳しく説明します。API は、クライアント側とサーバ側に分けられます。

表 3-1 および表 3-2 に一覧表示した API を使用して、iPlanet Calendar Server のデフォルトの動作を拡張または置き換えます。

表 3-1 クライアント API

<code>csIAccessControl</code>	デフォルトのアクセスコントロールメカニズムを拡張または置き換える
<code>csIAuthentication</code>	ログイン認証メカニズムを拡張または置き換える
<code>csICalendarLookup</code>	デフォルトのカレンダー検索メカニズムを拡張または置き換える
<code>csIDataTranslator</code>	着信データおよび発信データの形式変換を拡張または置き換える
<code>csIPlugin</code>	モジュールのバージョン管理情報と内容に関する情報を提供する
<code>csIUserAttributes</code>	ユーザ属性の格納および検索メカニズムを拡張または置き換える
<code>csIQualifiedCalidLookup</code>	指定された完全修飾の URL のカレンダー ID を検索する

表 3-2 サーバ API

<code>csICalendarServer</code>	バージョン番号などの全般的なサーバ情報を提供する
<code>csIMalloc</code>	サーバのメモリ割り当てメカニズムへアクセスする

csIAccessControl

iPlanet Calendar Server のデフォルトのアクセスコントロール動作を拡張または置き換えるには、このインタフェースのメソッドを実装します。

メソッド

csIAccessControl インタフェースは、次の2つのメソッドを実装します。

CheckAccess	ユーザに対するアクセスコントロール基準を設定する
Init	インタフェースが見つかって登録されたことを確認する

説明

許可されるアクセスのタイプを定義します。デフォルトのアクセスコントロールを使用するか、デフォルトを置き換えるかを指定するリターンコードを設定する必要があります。

CheckAccess

目的

ユーザのカレンダーアクセスを設定します。

構文

```
PRUint32 CheckAccess (char * aUser,  
                     char * aCalid,  
                     PRInt32 * aAccessRequest,  
                     PRInt32 * aAccessAllowed,  
                     PRInt32 * aReturnCode) = 0;
```

パラメータ

このメソッドには、次の5つのパラメータがあります。

aUser	要求を発行する認証済みユーザ。匿名 (anonymous) アクセスの場合、pUserID は「anonymous」
aCalid	アクセス対象のカレンダーの calid
aAccessRequest	要求されたアクセスタイプを表すビットフラグのセット
aAccessAllowed	出力パラメータ。許可したアクセスを表すビットフラグのセット。このメソッドは、aAccessRequest で指定されたビットだけをチェックする

aReturnCode	リターン時に、サーバがデフォルトの認証手順を引き続き行うべきかどうかを決める定数を含める 次の定数のどちらか
	<ul style="list-style-type: none"> • NS_CONTINUE_DEFAULT_PROCESSING • NS_OVERRIDE_DEFAULT_PROCESSING

戻り値

成功した場合は、NS_OK。失敗した場合は、ゼロ以外のエラーコード。

説明

ユーザのアクセスタイプを要求するには、このメソッドを使用します。ユーザの名前を aUser パラメータに、要求するアクセスタイプを aAccessRequest パラメータ (ビットマスク) に指定して送ります。システムは、aAccessRequest ビットマスクに指定されたアクセスタイプだけをチェックします。返されるビットマスク aAccessAllowed は、要求したタイプに対してユーザに許可されたアクセスを表します。

anonymous アクセスの場合、ユーザ ID は「anonymous」です。

次に、利用できるアクセスタイプを定義する ICS_ACCESTYPE 定数 (ビットマップ) を示します。

表 3-3 ICS_ACCESTYPE 定数

アクセスタイプ	ビットマップ
ICS_ACCESTYPE_NONE	0x00000000
ICS_ACCESTYPE_READCOMPONENT	0x00000001
ICS_ACCESTYPE_WRITECOMPONENT	0x00000002
ICS_ACCESTYPE_CREATECOMPONENT	0x00000008
ICS_ACCESTYPE_DELETECOMPONENT	0x00000010
ICS_ACCESTYPE_READCALENDAR	0x00000020
ICS_ACCESTYPE_WRITECALENDAR	0x00000040
ICS_ACCESTYPE_CREATECALENDAR	0x00000080
ICS_ACCESTYPE_DELETECALENDAR	0x00000100
ICS_ACCESTYPE_SCHEDULE	0x00000200
ICS_ACCESTYPE_FREEBUSY	0x00000400
ICS_ACCESTYPE_SELF_ADMIN	0x00000800

表 3-3 ICS_ACCESTYPE 定数

アクセスタイプ	ビットマップ
ICS_ACCESTYPE_ALL	0xFFFFFFFF

独自のアクセスコントロール手順を設定するには、このメソッドを使用します。元からある (ネイティブの) アクセスコントロールメカニズムを拡張して独自の処理を先に実行してからデフォルトの処理を続けて実行するか、あるいは、ネイティブのアクセスコントロールメカニズムを完全に置き換えることができます。

Init

目的

インタフェースが登録されていることを確認し、サーバへの参照を提供します。

構文

```
PRUint32 Init (nsISupports * aServer) = 0;
```

パラメータ

このメソッドには、次のパラメータがあります。

aServer	リターン時に、このモジュールが登録されたサーバへの参照を含む
---------	--------------------------------

戻り値

成功した場合は、NS_OK。失敗した場合は、ゼロ以外のエラーコード。

説明

サーバは、モジュールの読み込み時にインタフェースを見つけて登録したあと、このメソッドを呼び出して、その操作が成功したことを確認します。aServer で渡されたポインタを使用して、サーバへの呼び出しを行うことができます。

csIAuthentication

Calendar Server のデフォルトの認証動作を拡張または置き換えるすべてのプラグインは、このインタフェースを実装する必要があります。

メソッド

csIAuthentication インタフェースは、次の 5 つのメソッドを実装します。

ChangePassword	ユーザのパスワードを変更する
Init	インタフェースが見つかって登録されたことを確認する
Logon	ユーザをログインする
Logout	ユーザをログアウトする
VerifyUserExists	ユーザの存在を確認する

説明

選択した認証手法を実装しているログオン、ログオフ、確認、およびパスワードのメソッドを定義できます。1 つのメソッドを置き換えたあとも、その他のメソッドに対してデフォルトの動作を引き続き使用できます。各メソッドは、リターンコードのパラメータ (aReturnCode) を使用して、自分自身 (メソッド) の実行後にデフォルトのアクセスコントロール処理を続けるかどうかをサーバに指示します。リターンコードの値は、次の定数のどちらかである必要があります。

NS_CONTINUE_DEFAULT_PROCESSING	サーバがデフォルトのアクセスコントロール処理を続けることを示す
NS_OVERRIDE_DEFAULT_PROCESSING	このメソッドがサーバのネイティブのアクセスコントロールメカニズムを置き換えることを示す

ChangePassword

目的

指定されたユーザのパスワードを変更します。

構文

```
PRUint32 ChangePassword (char * aUser,  
                        char * aOldPassword,  
                        char * aNewPassword,  
                        PRInt32 * aReturnCode) = 0;
```

パラメータ

このメソッドには、次の 4 つのパラメータがあります。

aUser	ユーザの名前
-------	--------

<code>aOldPassword</code>	古いパスワード
<code>aNewPassword</code>	新しいパスワード
<code>aReturnCode</code>	リターン時に、サーバがデフォルトの認証手順を引き続き行うべきかどうかを決める定数を含める 次の定数のどちらかを含める <ul style="list-style-type: none">• <code>NS_CONTINUE_DEFAULT_PROCESSING</code>• <code>NS_OVERRIDE_DEFAULT_PROCESSING</code>

戻り値

成功した場合は、`NS_AUTHENTICATION_CHANGEPASSWORD_SUCCESS`。失敗した場合は、`NS_AUTHENTICATION_CHANGEPASSWORD_FAILURE`。

説明

指定されたユーザのパスワードを変更します。

Init

目的

インタフェースが登録されていることを確認し、サーバへの参照を提供します。

構文

```
PRUint32 Init (nsISupports * aServer) = 0;
```

パラメータ

このメソッドには、次のパラメータがあります。

<code>aServer</code>	リターン時に、このモジュールが登録されたサーバへの参照を含む
----------------------	--------------------------------

戻り値

成功した場合は、`NS_OK`。失敗した場合は、ゼロ以外のコード。

説明

サーバは、モジュールの読み込み時にインタフェースを見つけて登録したあと、このメソッドを呼び出して、その操作が成功したことを確認します。`aServer` で渡されたポインタを使用して、サーバへの呼び出しを行うことができます。

Logon

目的

プレーンテキストによるログインのための認証手順を拡張または置き換えます。

構文

```
PRUint32 Logon (char * aUser,  
               char * aPassword,  
               PRInt32 * aReturnCode) = 0;
```

パラメータ

このメソッドには、次の3つのパラメータがあります。

aUser	ユーザのログイン名
aPassword	プレーンテキストのパスワード
aReturnCode	リターン時に、サーバがデフォルトの認証手順を引き続き行うべきかどうかを決める定数を含める 次の定数のどちらかを含める <ul style="list-style-type: none">• NS_CONTINUE_DEFAULT_PROCESSING• NS_OVERRIDE_DEFAULT_PROCESSING

戻り値

成功した場合は、NS_AUTHENTICATION_LOGON_SUCCESS。失敗した場合は、NS_AUTHENTICATION_LOGON_FAILURE。

説明

iPlanet Calendar Server へのログインに関する独自の認証手順を指定するために、このメソッドを使用します。ネイティブの認証メカニズムを拡張して独自の処理を先に実行してからデフォルトの処理を続けて実行するか、ネイティブの認証メカニズムを完全に置き換えることができます。

Logout

目的

ユーザをログアウトします。

構文

```
PRUint32 Logout (char * aUser, PRInt32 * aReturnCode) = 0;
```

パラメータ

このメソッドには、次の 2 つのパラメータがあります。

aUser	ログアウトするユーザのユーザ ID
aReturnCode	リターン時に、サーバがデフォルトの認証手順を引き続き行うべきかどうかを決める定数を含める 次の定数のどちらかを含める
	<ul style="list-style-type: none">• NS_CONTINUE_DEFAULT_PROCESSING• NS_OVERRIDE_DEFAULT_PROCESSING

戻り値

成功した場合は、NS_OK。失敗した場合は、ゼロ以外のエラーコード。

説明

VerifyUserExists

目的

ユーザ ID が LDAP ディレクトリに存在することを確認します。

構文

```
PRUint32 VerifyUserExists (char * aUser, PRInt32 * aReturnCode) = 0;
```

パラメータ

このメソッドには、次の 2 つのパラメータがあります。

aUser	ユーザの名前
aReturnCode	リターン時に、サーバがデフォルトの認証手順を引き続き行うべきかどうかを決める定数を含める 次の定数のどちらかを含める
	<ul style="list-style-type: none">• NS_CONTINUE_DEFAULT_PROCESSING• NS_OVERRIDE_DEFAULT_PROCESSING

戻り値

成功した場合は、NS_OK。失敗した場合は、ゼロ以外のエラーコード。

説明

csICalendarLookup

デフォルトのカレンダー検索 (LDAP) を拡張または置き換えるには、このインタフェースのメソッドを実装します。

メソッド

csICalendarLookup は、次の 5 つのメソッドを実装します。

Init	インタフェースが見つかって登録されたことを確認する
QualifyCalid	相対 calid を修飾する
FreeCalid	前に割り当てられた完全修飾 calid のメモリ領域を解放する
QueryType	プラグインのタイプを照会する
FreeType	前に割り当てられたタイプのメモリ領域を解放する

説明

1 つ以上のメソッドを実装することにより、カレンダー検索を制御できます。

Init

目的

インタフェースが登録されていることを確認し、サーバへの参照を提供します。

構文

```
PRUint32 Init (nsISupports * aServer) = 0;
```

パラメータ

このメソッドには、次のパラメータがあります。

aServer	リターン時に、このモジュールが登録されたサーバへの参照を含む
---------	--------------------------------

戻り値

成功した場合は NS_OK、失敗した場合はゼロ以外のエラーコード。

説明

サーバは、モジュールの読み込み時にインタフェースを見つけて登録したあと、このメソッドを呼び出して、その操作が成功したことを確認します。aServer で渡されたポインタを使用して、サーバへの呼び出しを行うことができます。

QualifyCalid

目的

相対 calid を修飾します。

構文

```
PRUint32 QualifyCalid (char * aRelativeCalid,  
                      char ** aQualifiedCalid,  
                      PRInt32 * aReturnCode) = 0;
```

パラメータ

このメソッドには、次のパラメータがあります。

aRelativeCalid	修飾対象の相対 calid
aQualifiedCalid	リターン時には、修飾された calid の URL が含まれる
aReturnCode	成功した場合は、NS_OK。失敗した場合は、通常の処理を中断する

戻り値

成功した場合は NS_OK、失敗した場合はゼロ以外のエラーコード。

説明

FreeCalid

目的

前に割り当てられた完全修飾カレンダー ID (calid) のメモリ領域を解放します。

構文

```
PRUint32 FreeCalid (char ** aQualifiedCalid, PRInt32 * aReturnCode) =  
0;
```

パラメータ

このメソッドには、次のパラメータがあります。

aQualifiedCalid	解放する calid
aReturnCode	成功した場合は、NS_OK。失敗した場合は、通常の処理を中断する

戻り値

成功した場合は NS_OK、失敗した場合はゼロ以外のエラーコード。

説明

QueryType

目的

データベースのプラグインのタイプを照会します。

構文

```
PRUint32 QueryType (char * aType, PRInt32 * aReturnCode) = 0;
```

パラメータ

このメソッドには、次のパラメータがあります。

aType	CLD (Calendar Lookup Database、カレンダー検索データベース) のタイプ
aReturnCode	成功した場合は、NS_OK。失敗した場合は、通常の処理を中断する

戻り値

成功した場合は NS_OK、失敗した場合はゼロ以外のエラーコード。

説明

この関数は、プラグインが実装している CLD のタイプを表す文字列を返します。

サポートされている唯一のタイプは「algorithmic」で、これは正規表現をサポートします。

FreeType

目的

前に割り当てられたデータベースのプラグインのタイプのメモリ領域を解放します。

構文

```
PRUint32 FreeType (char * aType, PRInt32 * aReturnCode) = 0;
```

パラメータ

このメソッドには、次のパラメータがあります。

aType	解放するデータベースのプラグインのタイプ
aReturnCode	成功した場合は、NS_OK。失敗した場合は、通常の処理を中断する

戻り値

成功した場合は NS_OK、失敗した場合はゼロ以外のエラーコード。

説明

QueryType メソッドで割り当てられた文字列のメモリ領域を解放します。

csIDataTranslator

これは、データ変換機能のプラグインのインタフェースです。すべてのパラメータのメモリ領域は、プラグインを使用して割り当ててください。

メソッド

csIDataTranslator インタフェースは、次の3つのメソッドを実装します。

GetSupportedContentTypes	このデータベース変換機能がサポートするコンテンツタイプをサーバに通知する
Init	インタフェースが見つかって登録されたことを確認する
Translate	カレンダーデータを指定された MIME 形式に変換する

説明

このインタフェースを使って、データベースとの間で、またはさまざまなデータ変換機能間でやり取りされるカレンダーデータの HTML 本文の内容を操作したり変更したりできます。データ変換機能は、WCAP 応答の出力形式 (fmt-out) のコンポーネントを操作します。

iPlanet Calendar Server では、カレンダーデータの変換用として次の MIME タイプをサポートします。

表 3-4 MIME タイプ

MIME タイプ	説明
text/calendar	iCalendar
text/xml	XML 形式の iCalendar
text/js	ネイティブの JavaScript

CSAPI データ変換モジュールは、`GetSupportedContentType` メソッドを使用して特定の MIME タイプをサーバに登録します。データ変換機能は、サポートされている任意の MIME タイプで受け取るデータを要求できます。

着信データの MIME タイプがモジュールによって受け入れられる場合、サーバはそのデータをモジュールの `Translate` メソッドに渡します。データ変換機能はそのデータをサポートされている MIME タイプに変換し、サーバへ戻し、サーバはデータベースを更新します。

GetSupportedContentTypes

目的

このデータベース変換機能がサポートしているコンテンツタイプ (Content-Types) を取得します。

構文

```
PRUint32 GetSupportedContentTypes (char ** aSupportedInContentTypes,  
char ** aSupportedOutContentType,  
char ** aPreferredInContentType,  
PRInt32 * aReturnCode) = 0;
```

パラメータ

このメソッドには、次のパラメータがあります。

<code>aSupportedInContentTypes</code>	サーバが変換機能に入力データとして送ることのできるコンテンツタイプのリスト。NULL で終わっている文字列の配列
<code>aSupportedOutContentType</code>	このプラグインによるデータ変換後のコンテンツタイプ
<code>aPreferredInContentType</code>	このプラグインが優先的に受け取るコンテンツタイプ。最初のパラメータで渡された、サポートしているコンテンツタイプの1つを指定する
<code>aReturnCode</code>	リターン時に、サーバがデフォルトの認証手順を引き続き行うべきかどうかを決める定数を含める 次の定数のどちらかを含める <ul style="list-style-type: none">• <code>NS_CONTINUE_DEFAULT_PROCESSING</code>• <code>NS_OVERRIDE_DEFAULT_PROCESSING</code>

戻り値

成功した場合は、`NS_OK`。失敗した場合は、ゼロ以外のコード。

説明

このデータベース変換機能がサポートするコンテンツタイプを取得します。

Init

目的

インタフェースが登録されていることを確認し、サーバへの参照を取得します。

構文

```
PRUint32 Init (nsISupports * aServer) = 0;
```

パラメータ

このメソッドには、次のパラメータがあります。

<code>aServer</code>	リターン時に、このモジュールが登録されたサーバへの参照を含む
----------------------	--------------------------------

戻り値

成功した場合は `NS_OK`、失敗した場合はゼロ以外のエラーコード。

説明

サーバは、モジュールの読み込み時にインタフェースを見つけて登録したあと、このメソッドを呼び出して、その操作が成功したことを確認します。aServer で渡されたポインタを使用して、サーバへの呼び出しを行うことができます。

Translate

目的

あるコンテンツタイプから別のコンテンツタイプへ変換する機能を実装します。

構文

```
PRUint32 Translate (char * aInContentType,
                  char * aOutContentType,
                  char ** aInBuffer,
                  char ** aOutBuffer,
                  PRInt32 * aInSize,
                  PRInt32 * aOutsize,
                  PRInt32 * aReturncode) = 0;
```

パラメータ

このメソッドには、次のパラメータがあります。

aInContentType	着信するコンテンツタイプ
aOutContentType	発信するコンテンツタイプ
aInBuffer	入力データバッファ
aOutBuffer	出力データバッファ
aInSize	入力バッファのサイズ
aOutSize	出力バッファのサイズ
aReturnCode	リターン時に、サーバがデフォルトの認証手順を引き続き行うべきかどうかを決める定数を含める 次の定数のどちらかを含める <ul style="list-style-type: none">• NS_CONTINUE_DEFAULT_PROCESSING• NS_OVERRIDE_DEFAULT_PROCESSING

戻り値

成功した場合は NS_OK、失敗した場合はゼロ以外のエラーコード。

説明

このメソッドは、指定されたバッファから指定された入力形式の内容を取り出して、内容を元の形式から出力形式に変換し、変換した内容を指定された出力バッファに書き込んで、出力バッファのサイズを指定された場所へ書き込みます。

csIPlugin

起動時にプラグインモジュールの情報をサーバに提供するには、このインタフェースのメソッドを実装する必要があります。

メソッド

csIPlugin インタフェースは、次の4つのメソッドを実装します。

GetDescription	プラグインの機能の説明文を取得する
GetVendorName	このプラグインを提供しているベンダーの説明文を取得する
GetVersion	このプラグインのメジャーおよびマイナーバージョンを取得する。この値は、1.0以上である必要がある
Init	インタフェースが見つかって登録されたことを確認する

説明

このインタフェースは必須ではありません。しかし、サーバがモジュールを読み込むときにバージョン情報を取得できるように、各モジュールでこのインタフェースを実装することを強くお勧めします。これらのメソッドは、プラグインの内容に関する情報をサーバに返します。

GetDescription

目的

モジュールの説明文を返します。

構文

```
PRUint32 GetDescription (nsString& aDescription) = 0;
```

パラメータ

このメソッドには、次のパラメータがあります。

aDescription	リターン時に、モジュールの説明文を含む
--------------	---------------------

戻り値

成功した場合は `NS_OK`、失敗した場合はゼロ以外のエラーコード。

説明

このメソッドを使用して、モジュールの説明文を提供します。

GetVendorName

目的

モジュールを提供しているベンダーの説明文を返します。

構文

```
PRInt32 GetVendorName (NSString& aVendorName) = 0;
```

パラメータ

このメソッドには、次のパラメータがあります。

<code>aVendorName</code>	リターン時に、ベンダーの説明文を含む
--------------------------	--------------------

戻り値

成功した場合は `NS_OK`、失敗した場合はゼロ以外のエラーコード。

説明

モジュールの提供元を特定します。

GetVersion

目的

起動時にサーバにバージョン情報を提供します。

構文

```
PRUint32 GetVersion (PRUint32& aMajorValue,  
                    PRUint32& aMinorValue) = 0;
```

パラメータ

このメソッドには、次の2つのパラメータがあります。

<code>aMajorValue</code>	リターン時に、メジャーバージョン番号を含む
<code>aMinorValue</code>	リターン時に、マイナーバージョン番号を含む

戻り値

成功した場合は NS_OK、失敗した場合はゼロ以外のエラーコード。

説明

このメソッドを使用して、モジュールのメジャーバージョン番号およびマイナーバージョン番号を特定します。この番号は、1.0 以上である必要があります。

Init

目的

インタフェースが登録されていることを確認し、サーバへの参照を取得します。

構文

```
PRUint32 Init (nsISupports * aServer) = 0;
```

パラメータ

このメソッドには、次のパラメータがあります。

aServer	リターン時に、このモジュールが登録されたサーバへの参照を含む
---------	--------------------------------

戻り値

成功した場合は NS_OK、失敗した場合はゼロ以外のエラーコード。

説明

サーバは、モジュールの読み込み時にインタフェースを見つけて登録したあと、このメソッドを呼び出して、その操作が成功したことを確認します。aServer で渡されたポインタを使用して、サーバへの呼び出しを行うことができます。

csIQualifiedCalidLookup

渡された修飾 URL のカレンダー ID を取得するデフォルトのメソッドを拡張または置き換えるには、このインタフェースのメソッドを実装します。

メソッド

csICalendarLookup は、次の 2 つのメソッドを実装します。

FindCalid	修飾 URL のカレンダー ID を返す
Init	インタフェースが見つかって登録されたことを確認する

説明

渡された修飾 URL のカレンダー ID を取得します。calid が見つからない場合、このコマンドはエラーを返します。

FindCalid

目的

指定された URL のカレンダー ID を検索します。

構文

```
PRUint32 FindCalid (char * pQualifiedURL,  
                   char ** ppCalidOut,  
                   PRInt32 * piCalidSize,  
                   PRInt32 * aReturnCode) = 0;
```

パラメータ

このメソッドには、次のパラメータがあります。

pQualifiedURL	検索対象の URL
ppCalidOut	見つかった calid のアドレスのポインタ
piCalidSize	返された calid のサイズ
aReturnCode	リターン時に、サーバがデフォルトの認証手順を引き続き行うべきかどうかを決める定数を含める 次の定数のどちらかを含める <ul style="list-style-type: none">• NS_CONTINUE_DEFAULT_PROCESSING• NS_OVERRIDE_DEFAULT_PROCESSING

戻り値

渡された修飾 URL の calid。

説明

渡された完全修飾の URL を使用して、カレンダー ID データベースの検索を実行します。一致した場合、calid のアドレスのポインタと calid のサイズを示す整数を返します。

Init

目的

インタフェースが登録されていることを確認し、サーバへの参照を提供します。

構文

```
PRUint32 Init (nsISupports * aServer) = 0;
```

パラメータ

このメソッドには、次のパラメータがあります。

aServer	リターン時に、このモジュールが登録されたサーバへの参照を含む
---------	--------------------------------

戻り値

成功した場合は NS_OK、失敗した場合はゼロ以外のエラーコード。

説明

サーバは、モジュールの読み込み時にインタフェースを見つけて登録したあと、このメソッドを呼び出して、その操作が成功したことを確認します。aServer で渡されたポインタを使用して、サーバへの呼び出しを行うことができます。

csUserAttributes

ユーザ属性の設定または取得の手順を置き換えるには、このインタフェースのメソッドを実装します。

メソッド

csUserAttributes インタフェースは、次の4つのメソッドを実装します。

FreeAttribute	取得した属性を保存するために使用されたメモリを解放する
GetAttribute	ユーザの属性値を取得する
Init	インタフェースが見つかって登録されたことを確認する
SetAttribute	ユーザの属性値を設定する

説明

このユーザ属性インタフェースによって、CSAPI モジュールは、受け取ったすべての要求を保持または操作してユーザ属性値を設定および取得できるようになります。属性を取得および設定するメソッドは、好みの手法を使用して、自分自身で提供してください。

FreeAttribute

目的

ローカルの属性記憶域に関連付けられたメモリを解放します。

構文

```
PRInt32 FreeAttribute (char * aValue, PRInt32 * aReturnCode) = 0;
```

パラメータ

このメソッドには、次の 2 つのパラメータがあります。

aValue	取得した属性値を保存するために割り当てた場所
aReturnCode	リターン時に、サーバがデフォルトの認証手順を引き続き行うべきかどうかを決める定数を含める 次の定数のどちらかを含める
	<ul style="list-style-type: none">• NS_CONTINUE_DEFAULT_PROCESSING• NS_OVERRIDE_DEFAULT_PROCESSING

戻り値

成功した場合は NS_OK、失敗した場合はゼロ以外のエラーコード。

説明

GetAttribute メソッドを使用して属性の値を取得すると、その値は、割り当てた場所を選択したメモリ管理手法を使って保存されます。メモリが不要になったときに同じメモリ管理手法でそのメモリを解放するには、FreeAttribute メソッドを使用してください (csIMalloc を参照)。

GetAttribute

目的

ユーザの属性値を検索します。

構文

```
PRUint32 GetAttribute (char * aUser,  
                      char * aKey,  
                      char ** aValue,  
                      PRInt32 * aReturnCode) = 0;
```

パラメータ

このメソッドには、次の4つのパラメータがあります。

aUser	ユーザの名前
aKey	属性キー
aValue	リターン時に、取得された属性値のポインタがここに含まれる
aReturnCode	リターン時に、サーバがデフォルトの認証手順を引き続き行うべきかどうかを決める定数を含める 次の定数のどちらかを含める <ul style="list-style-type: none">• NS_CONTINUE_DEFAULT_PROCESSING• NS_OVERRIDE_DEFAULT_PROCESSING

戻り値

成功した場合は NS_OK、失敗した場合はゼロ以外のエラーコード。

説明

指定されたユーザの指定された属性の値を取得し、その値を aValue が指す場所に保存します。返された属性の保存場所の割り当てと、不要になったときの解放 (FreeAttribute メソッドを使用) は、開発者の責任で行う必要があります。

Init

目的

インタフェースが登録されていることを確認し、サーバへの参照を取得します。

構文

```
PRUint32 Init (nsISupports * aServer) = 0;
```

パラメータ

このメソッドには、次のパラメータがあります。

aServer	リターン時に、このモジュールが登録されたサーバへの参照を含む
---------	--------------------------------

戻り値

成功した場合は NS_OK、失敗した場合はゼロ以外のエラーコード。

説明

サーバは、モジュールの読み込み時にインタフェースを見つけて登録したあと、このメソッドを呼び出して、その操作が成功したことを確認します。aServer で渡されたポインタを使用して、サーバへの呼び出しを行うことができます。

SetAttribute

目的

ユーザの属性値を設定します。

構文

```
RUInt32 SetAttribute (char * aUser,  
                    char * aKey,  
                    char * aValue,  
                    PRInt32 * aReturnCode) = 0;
```

パラメータ

このメソッドには、次のパラメータがあります。

aUser	ユーザの名前
aKey	属性キー
aValue	値
aReturnCode	リターン時に、サーバがデフォルトの認証手順を引き続き行うべきかどうかを決める定数を含める 次の定数のどちらかを含める <ul style="list-style-type: none">• NS_CONTINUE_DEFAULT_PROCESSING• NS_OVERRIDE_DEFAULT_PROCESSING

戻り値

成功した場合は NS_OK、失敗した場合はゼロ以外のエラーコード。

説明

指定されたユーザの指定された属性を指定された値に設定します。

csICalendarServer

サーバのバージョン情報をプラグインモジュールに提供します。

メソッド

csICalendarServer インタフェースは、次の2つのメソッドを実装します。

GetVersion	カレンダーサーバのバージョン情報を取得する
Init	インタフェースが見つかって登録されたことを確認する

説明

このプラグインモジュールを使用すると、csICalendarServer インタフェースにクエリーを送信して iPlanet Calendar Server の実行中のインスタンスのバージョン情報を取得することができます。このインスタンスはクライアントが存在する間は有効なので、Init はサーバへの参照を返しません。

GetVersion

目的

プラグインモジュールにサーバのバージョン情報を提供します。

構文

```
PRUint32 GetVersion (PRUint32& aMajorValue,  
                    PRUint32& aMinorValue) = 0;
```

パラメータ

このメソッドには、次の2つのパラメータがあります。

aMajorValue	リターン時に、メジャーバージョン番号を含む
aMinorValue	リターン時に、マイナーバージョン番号を含む

戻り値

成功した場合は NS_OK、失敗した場合はゼロ以外のエラーコード。

説明

このメソッドを使用して、サーバのメジャー番号およびマイナーバージョン番号を特定します。この番号は、常に 1.0 以上です。

Init

目的

インタフェースが登録されたことを確認します。

構文

```
PRUint32 Init () = 0;
```

パラメータ

このメソッドには、パラメータはありません。

戻り値

成功した場合は `NS_OK`、失敗した場合はゼロ以外のエラーコード。

説明

サーバは、このメソッドを呼び出して、インタフェースが検出されて正常に登録されたことを確認します。

csIMalloc

メモリの割り当てと解放を行います。

メソッド

`csIMalloc` インタフェースは、次の6つのメソッドを実装します。

<code>Calloc</code>	オブジェクト用のメモリを割り当て、初期化する
<code>Free</code>	使われなくなったメモリを解放する
<code>FreeIf</code>	メモリを解放する。NULL ポインタも受け付ける
<code>Init</code>	インタフェースが見つかって登録されたことを確認する
<code>Malloc</code>	メモリを割り当てる
<code>Realloc</code>	前に割り当てたメモリを割り当て直す

説明

プラグインモジュールは、このオブジェクトを使用して、サーバの効率的なメモリ割り当て手法を利用できます。このオブジェクトはクライアントが存在する間有効なので、`Init` は参照を返しません。

Calloc

目的

複数のオブジェクトにメモリを割り当て、ゼロに初期化します。

構文

```
void* Calloc (PRUint32 aSize, PPRUint32 aNum) = 0;
```

パラメータ

このメソッドには、次の2つのパラメータがあります。

aSize	各オブジェクトのバイト単位のサイズ
nNum	オブジェクトの数

戻り値

成功した場合は割り当てたメモリのポインタ、失敗した場合は NULL。

説明

このメソッドは、指定されたサイズを持つ指定された数のオブジェクトに必要なメモリを割り当て、そのメモリをゼロに初期化します。

Free

目的

前に Malloc メソッドで割り当てられたメモリを解放します。

構文

```
PRUint32 Free (void * aPtr) = 0;
```

パラメータ

このメソッドには、次のパラメータがあります。

aPtr	解放するメモリへのポインタ
------	---------------

戻り値

成功した場合は NS_OK、失敗した場合はゼロ以外のエラーコード。

説明

C および C++ の対応するメソッドと同じ方法でこのメソッドを使用して、前に割り当てられたメモリを解放します。

FreeIf

目的

前に Malloc メソッドで割り当てられたメモリを解放します。NULL ポインタも受け付けます。

構文

```
PRUint32 FreeIf (void * aPtr) = 0;
```

パラメータ

このメソッドには、次のパラメータがあります。

aPtr	解放するメモリのポインタ、または NULL
------	-----------------------

戻り値

成功した場合は NS_OK、失敗した場合はゼロ以外のエラーコード。

説明

aPtr が NULL でない場合は、指定された場所のメモリを解放します。

Init

目的

インタフェースが登録されていることを確認し、サーバへの参照を提供します。

構文

```
PRUint32 Init (nsISupports * aServer) = 0;
```

パラメータ

このメソッドには、次のパラメータがあります。

aServer	リターン時に、このモジュールが登録されたサーバへの参照を含む
---------	--------------------------------

戻り値

成功した場合は、NS_OK。失敗した場合は、ゼロ以外のエラーコード。

説明

サーバは、モジュールの読み込み時にインタフェースを見つけて登録したあと、このメソッドを呼び出して、その操作が成功したことを確認します。aServer で渡されたポインタを使用して、サーバへの呼び出しを行うことができます。

Malloc

目的

指定された量のメモリを割り当てます。

構文

```
void* Malloc (PRUint32 nBytes) = 0;
```

パラメータ

このメソッドには、次のパラメータがあります。

nBytes	割り当てるメモリのバイト単位のサイズ
--------	--------------------

戻り値

成功した場合は割り当てられたメモリのポインタ、失敗した場合は NULL。

説明

C および C++ の対応するメソッドと同じ方法でこのメソッドを使用します。

Realloc

目的

前に割り当てたメモリを割り当て直します。

構文

```
void* Realloc (void * aPtr, PRUint32 nBytes) = 0;
```

パラメータ

このメソッドには、次のパラメータがあります。

aPtr	前に割り当てたメモリのポインタ
nBytes	新たに割り当てるメモリのバイト単位のサイズ

戻り値

成功した場合は割り当てられたメモリのポインタ、失敗した場合は NULL。

説明

C および C++ の対応するメソッドと同じ方法でこのメソッドを使用して、前に割り当てられたメモリを割り当て直します。

イベント通知サービス (ENS) の概要

イベント通知サービス (Event Notification Service、ENS) は、iPlanet Calendar Server 5.0 に付属する汎用の公開および購読サービスです。ENS は、iPlanet アプリケーションによって使用されるディスパッチャーとして機能します。ENS は、iPlanet アプリケーションが扱う特定のタイプのイベントの収集を集中管理します。特に、分類できるイベントのレポートを受け取り、そのカテゴリのイベントにアプリケーションが扱う処理として登録されている場合は、それらのアプリケーションに通知します。

要約すると、ENS は次の目的に使われるインターネットサービスです。

- 情報を必要としているユーザにイベントを通知する
- 新たに生じるイベントを購読する (新たに生じるイベントの通知を要求する)
- 既存の購読をキャンセルする (購読を中止する)

注 iPlanet では、汎用メッセージ指向ミドルウェアシステムを提供しています。ENS とは異なり、Java Message Queue (JMQ) と呼ばれます。JMQ は、Java Message Service (JMS) の実装です。

現時点では、ENS サブスクリバ API のみが Calendar Server イベントにアクセスするのに使用できます。特に、JMS を使って Calendar Server イベントにアクセスしないでください。将来、iPlanet メッセージングミドルウェアの API は JMS に統合される予定です。つまり、JMS が Calendar Server イベントにアクセスする主要な手段になります。

この章には、次のトピックがあります。

- ENS 用語集
- イベント
- イベント参照

- イベント通知サービス
- Calendar Server と ENS との連携動作
- API の概要
- カスタムアプリケーションの作成と実行

ENS 用語集

イベント	リソースの変化。たとえば、ユーザが新しい会議をカレンダー（リソース）に追加した場合など。ENS の場合、イベントとはアラームキューの状態の変化のこと
イベントカスタマ	「イベントサブスクライバ」の同義語
イベントプロデューサ	「イベントパブリッシャ」の同義語
イベントパブリッシャ	ほかのアプリケーションにイベントを通知するアプリケーション
イベント参照	ENS によって処理されるイベントを識別する。RFC 2396 で定義されている URI 構文に準拠する
イベントサブスクライバ	イベントを消費するアプリケーション
通知	イベントの発生を示すメッセージ。イベントパブリッシャによって送信され、イベントへの参照とイベントカスタマによって使用されるオプションデータを含む。このメッセージは、通知サービスには不透明
通知サービス	ほかのサーバから購読依頼と通知を受け取り、通知をサブスクライバに中継する
通知サーバ	通知サービスは、1 つ以上のサーバインスタンスから構成される。各インスタンスは、異なるホストで実行される
通知する	「公開する」の同義語
公開する	通知を送信する。イベントパブリッシャは、通知サービスにイベントを提供する
RENL (Reliable Event Notification Link)	RENL は、パブリッシャ、サブスクライバ、および一意の識別子を持ち、これらによって応答が必要な通知を識別する
リソース	IP ネットワークからアクセスされるデータ。たとえば、カレンダーなどが該当する
リソースの状態	リソースを表す属性の値。たとえば、会議の時刻など

購読する	購読依頼を送信する。イベントサブスクライバが、通知を必要とするイベントについて通知サービスに登録すること
購読依頼	イベントサブスクライバが送信するメッセージ。イベント参照、クライアント側の要求識別子、およびオプションのアクセスコントロール規則が含まれる
購読を中止する	購読をキャンセルする。イベントサブスクライバが、特定のイベントについて通知の中継の停止をイベント通知サービスに登録すること

イベント

イベントとは、リソースの1つ以上のプロパティ値が変化することです。URI でイベントを表現します。イベントがいつ発生したかという情報を必要とするアプリケーションが、ENS に登録されます。その登録情報でイベントを順番に識別し、通知と購読を対応付けます。

イベント参照

イベント参照は、ENS によって処理されるイベントを識別します。イベント参照は、RFC 2396 で定義された URI 構文を使います。

イベント参照の URI 構文を次に示します。

イベント参照 := *scheme* ":"*authority resource* ["?" *param* "=" *value* *("&" *param* "=" *value*)]

この構文の各変数は次のとおりです。

- *scheme* は、http、imap、ftp、wcap などのアクセス手段です。
iPlanet Calendar Server 5.0 の場合、ENS スキーマは *enp* になります。
- *authority* は、リソースへのアクセスを制御する DNS ドメインまたはホスト名です。
- *resource* は、*authority* コンテキスト内のリソースへ到達するためのパスです。複数のパス要素を指定するには、「/」で区切ります。
- *param* は、リソースの状態を表すパラメータの名前です。
- *value* は、パラメータの値です。パラメータと値の組み合わせは複数個指定できません。指定しなくてもかまいません。

すべての iPlanet Calendar Server イベントの URI スキーマは、次のようになります。

`enp://domain.com/calendar/ics/v50/instance`

例

たとえば、iPlanet Calendar Server 5.0 ユーザが特定の一意の識別子ですべてのイベントアラームを購読する場合は、URI は次のようになります。

```
enp://domain.com/calendar/ics/v50/ics-hostname/alarms?type=events
&uid=XXXX
```

イベント通知サービス

イベント通知サービスは、アプリケーションが次の目的で使用するインターネットサービスです。

- 通知する
- 購読する
- 購読を中止する

ENS は、enpd デーモンとして動作します。ほかの iPlanet Calendar Server 5.0 デーモンとともに、さまざまなカレンダーサーバ構成で動作します。ENS および iPlanet Calendar Server 5.0 の統合についての詳細は、この章の 84 ページの「Calendar Server と ENS との連携動作」と第 1 章「アーキテクチャの概要」の 25 ページの「水平方向のスケラビリティ」を参照してください。

通知する

ENS はサブスクライバにイベントを通知するために、イベント参照、オプションのアクセスコントロール規則、およびオプションのアプリケーション固有の (ENS には不透明) データで構成される通知を送信します。「公開する」とも呼びます。95 ページの「パブリッシャ API」を参照してください。

通知には、次の 2 種類があります。

- **非請求通知。** イベントパブリッシャから通知サーバに送信される通知。イベントサブスクライバが存在するかどうか、サブスクライバが通知を受け取ったかどうかをパブリッシャが知らないまたはそれに注意を払わない場合には、この要求は応答を必要としません。しかしながら、パブリッシャとサブスクライバが互いに認識できる場合は、RENL (reliable event notification link) を設定することもあります。その場合には、サブスクライバはパブリッシャからの通知を処理すると、応答をパブリッシャに返します。
- **サーバ通知。** 購読依頼に基づいて、サーバからサブスクライバへ送信される通知。このタイプの通知には、応答を返す必要があります。サーバ通知には、自発通知と同じ属性が含まれます。

購読する

ENS は、イベントの通知要求を受け取ります。イベントサブスクライバが送信する要求が購読依頼です。購読依頼は、セッションの存続期間中、または購読が中止されるまで有効です。

購読依頼は、イベント名、クライアント側の要求 ID、およびオプションのアクセスコントロール規則を含んでいます。102 ページの「サブスクライバ API」を参照してください。

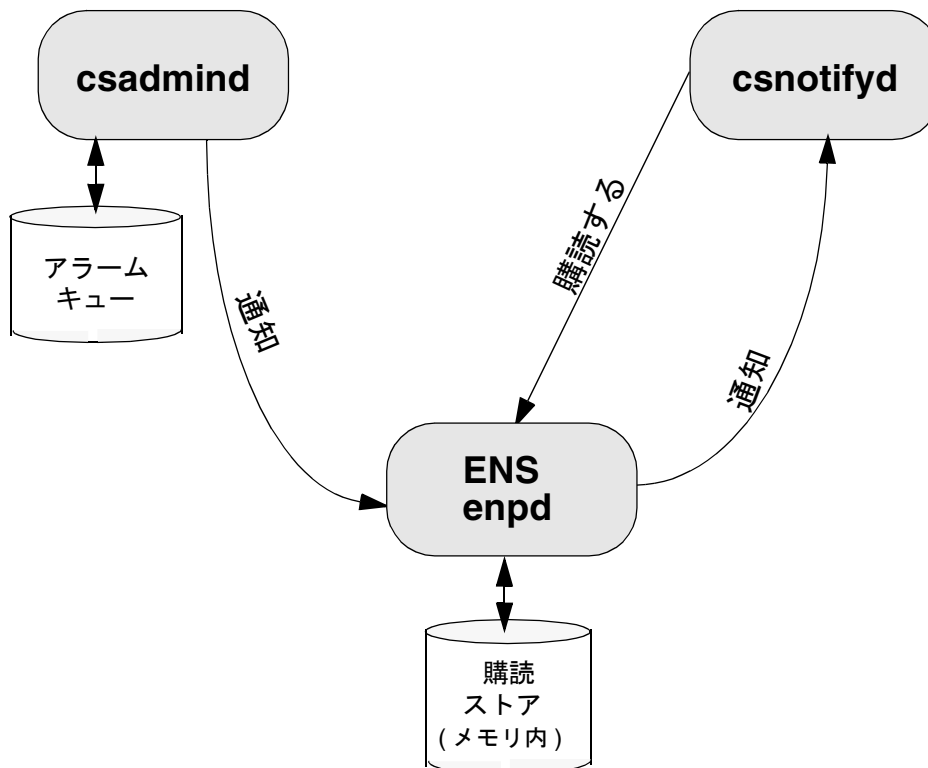
購読を中止する

ENS は、購読のキャンセル要求を受け取ります。クライアント側の要求 ID とセッション ID が購読依頼を識別します。102 ページの「サブスクライバ API」を参照してください。

Calendar Server と ENS との連携動作

ENS は、アラームキューおよび2つのデーモン `csadmin` と `csnotifyd` を介して、Calendar Server と連携動作します。図 4-1 は、2つの Calendar Server デーモンと ENS デーモンとの連携動作を示しています。

図 4-1 ENS の概要



アラームキュー

ENS は、アラームディスパッチャーです。このディスパッチャーによって、アラームの配信と生成が分離され、また、電子メールや無線通信などのいくつかの配信方法を使用できます。csadmin は、アラームキューの状態の変化に基づいてイベントを検出します。アラームキューの状態は、アラームがキューに格納されるたびに変化します。アラームは、イベントプロデューサーがアラームを生成したときにキューに格納されます。以下の URI は、これらの種類のイベントを表します。

イベントの場合

```
enp:///ics/eventalarm?calid=calid&uid=uid&rid=rid&aid=aid
```

仕事の場合

```
enp:///ics/todoalarm?calid=calid&uid=uid&rid=rid&aid=aid
```

この構文の各変数は次のとおりです。

- *calid* は、カレンダーの ID です。
- *uid* は、カレンダー内のイベント / 仕事 ID です。
- *rid* は、イベント / 仕事が繰り返し発生する場合の繰り返し ID です。
- *aid* は、イベント / 仕事内のアラーム ID です。アラームが複数ある場合は、*aid* でアラームを識別します。

csadmin は、アラームをキューから削除して通知を enpd へ送信します。次に、enpd は、このタイプのイベントの購読者がいるかどうかを調べ、検出した購読に対する通知を csnotifyd に送信します。この 3 つのデーモン間の対話により、イベント通知が行われます。

デーモン

iPlanet Calendar Server には、ENS デーモンの enpd と対話する次の 2 つのデーモンが組み込まれています。

- csadmin

csadmin は、通知サービスに通知を送信するパブリッシャーであり、イベントアラームを ENS へ送信します。csadmin は、iPlanet Calendar Server 5.0 のアラームキューを管理します。また、スケジューラを実装し、アラームが生成されるタイミングを知らせます。アラームの生成を知らされた時点で、csadmin はイベントを公開します。ENS は、イベント通知の受信およびディスパッチを行います。

アラームを確実に送信するために、`csadmin` は、特定のイベントまたは特定のイベントタイプに対する応答を要求します。この章の 86 ページの「アラーム転送の信頼性」を参照してください。`csadmin` は、RENL (Reliable Event Notification Links) を使用してイベント通知が確実に送信されたかどうかを確認します。RENL の詳細については、この節の「API の概要」の「パブリッシャの API」を参照してください。

- `csnotifyd`

`csnotifyd` は、特定のイベントに関心を持つ、つまり特定のイベントを購読しているサブスクライバです。購読しているイベントについての通知をイベント通知サービスから受け取り、これらのイベントと仕事についてのお知らせを電子メールでクライアントに送信します。

ENS アーキテクチャには購読を中止する機能がありますが、`csnotifyd` では次の 2 つの理由によりイベントの購読を中止しません。まず、通常動作において、購読の中止または再購読を行う必要はないためです。次に、購読は一時的に保存される (メモリに保存される) だけなので、ENS への接続が切断されるとすべての購読は暗黙に購読が中止されるためです。

`csnotifyd` は、`ics://hostname/alarm/pop` を購読するため、ホスト `hostname` に対するすべてのアラーム通知を受け取ります。`csnotifyd` は、アラーム通知を受け取ると電子メールメッセージを生成します。

アラーム転送の信頼性

アラームの紛失を防ぐために、`csadmin` と `csnotifyd` は特定の種類のアラームに対して ENS の RENL 機能を使用します。`csadmin` は、それらのアラームに対して終端間の応答を要求します。また、`csnotifyd` は通知を正常に処理したあと、受け取った各 RENL アラーム通知に対する応答を生成します。

RENL アラームの場合、ネットワーク、ENS デーモン、または `csnotifyd` が通知の処理に失敗した場合、`csadmin` は応答を受け取れないため、アラームキューからアラームを削除しません。そのため、タイムアウトのあとでアラームが再度公開されます。

例

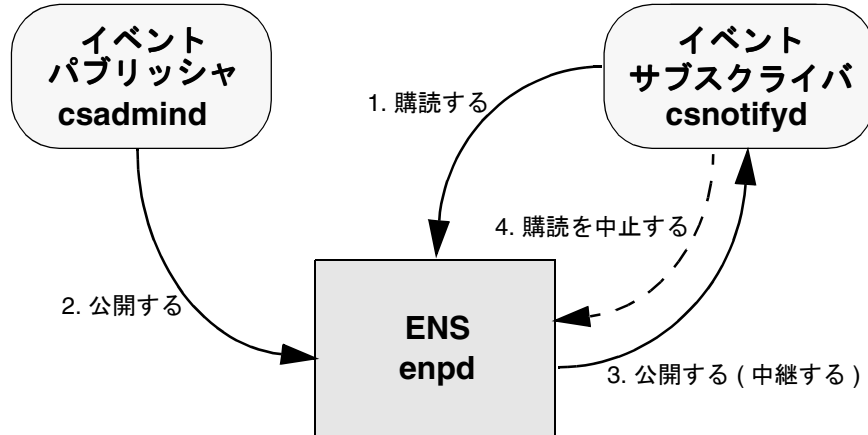
ENS の公開と購読の一般的な流れを示します。

1. イベントサブスクライバの `csnotifyd` が、特定のイベントに対して関心があることを表明します (購読する)。
2. イベントパブリッシャの `csadmin` は、イベントを検出して通知を送信します (公開する)。
3. ENS がイベントをサブスクライバに公開します。

- イベントサブスクライバは、イベントに対する関心をキャンセルします (購読の中止)。このステップは、ENS への接続が切断されると暗黙に行われます。

図 4-2 は、この流れを示しています。また、表 4-1 は、この図の説明です。

図 4-2 イベント通知サービスの公開と購読の流れの例



API の概要

表 4-1 イベント通知サービスの公開と購読の流れの例

動作	ENS の応答
1. csnotifyd が購読要求を ENS に送信する	ENS が購読依頼を購読データベースに保存する
2. csadmind が通知要求を ENS に送信する	ENS は、通知に一致する購読依頼を購読データベースから検索する
3. csnotifyd が ENS から通知を受け取る	購読依頼条件に一致している個々の通知に対して、アクセス権が付与され、その依頼ごとに通知要求を配信キューに追加する csnotifyd は、配信キューの要求を非同期に削除して通知を送信する
4. このバージョンの csnotifyd は、ENS にキャンセル要求を送信しない	購読依頼は、データベースではなくメモリだけに保存され、ENS への接続が切断されると、すべての購読が暗黙に中止されるため

ENS には、次の 3 つの API があります。

- パブリッシャ API

パブリッシャは、購読依頼イベントに対する通知をサブスクライバに送信します。アプリケーションは、通知を受け取ったことを確認する送達確認を要求することもできます。これを行うには、RENL (Reliable Event Notification Link) が必要です。RENL は、送達確認が必要な通知を識別するためのパブリッシャ、サブスクライバ、および一意の識別子を持ちます。パブリッシャは、送達確認を受け取ったことをアプリケーションに通知するために、end2end_ack コールバックを起動し、publish_a に渡します。

- サブスクライバ API

サブスクライバとは、特定のイベントに関心を持つ、通知サービスのクライアントのことです。通知サービスは、パブリッシャから特定のイベントについての通知を受け取ると、その通知をサブスクライバに中継します。

サブスクライバは、購読を中止することもできます。購読を中止すると、有効な購読依頼がキャンセルされます。

RENL を有効にするために、サブスクライバは自身の存在を ENS に対して宣言します。その結果、サブスクライバアプリケーションの代わりに、通知に対する送達確認が透過的に生成されます。サブスクライバは、任意のタイミングで RENL を無効にできます。

- 公開および購読ディスパッチャー API

非同期パブリッシャを使用する場合は、ENS はコールバックを起動するために、スレッドプールからスレッドを借りる必要があります。その場合、アプリケーションが独自のスレッドプールを作成して ENS に渡すか、ENS にスレッドプールを作成および管理させるか、選択できます。どちらの場合も、ENS はディスパッチャーオブジェクトを作成および使用して、使用されるディスパッチャー (`pas_dispatcher_t`) をインスタンス化します。

GDisp (`libasync`) が、iPlanet Calendar Server 5.0 でサポートされているディスパッチャーです。

パブリッシャ API 関数

パブリッシャの API は、表 4-2 に示す 1 つの定義と 9 つの関数から構成されます。

表 4-2 ENS パブリッシャ API の関数

定義 / 関数	説明
<code>publisher_t</code>	パブリッシャの定義
<code>publisher_cb_t</code>	非同期呼び出しの結果を確認する汎用コールバック関数
<code>publisher_new_a</code>	新規の非同期パブリッシャを作成する
<code>publisher_new_s</code>	新規の同期パブリッシャを作成する
<code>publish_a</code>	非同期通知を通知サービスに送信する
<code>publish_s</code>	同期通知を通知サービスに送信する
<code>publisher_delete</code>	パブリッシャのセッションを終了する
<code>publisher_get_subscriber</code>	パブリッシャの証明書を使用してサブスクライバを作成する
<code>renl_create_publisher</code>	RENL を作成する。end2end_ack の呼び出しが有効になる
<code>renl_cancel_publisher</code>	RENL をキャンセルする

サブスクライバ API 関数

サブスクライバ API には、表 4-3 に示す 2 つの定義と 10 個の関数があります。

表 4-3 ENS サブスクライバ API の関数

定義 / 関数	説明
<code>subscriber_t</code>	サブスクライバの定義

表 4-3 ENS サブスクリイバ API の関数 (続き)

定義 / 関数	説明
subscription_t	購読の定義
subscriber_cb_t	非同期呼び出しの結果を確認する汎用コールバック関数
subscriber_notify_cb_t	同期コールバック。通知を受け取ったときに呼び出される
subscriber_new_a	新規の非同期サブスクリイバを作成する
subscriber_new_s	新規の同期サブスクリイバを作成する
subscribe_a	非同期購読を確立する
unsubscribe_a	非同期購読をキャンセルする
subscriber_delete	サブスクリイバを終了する
subscriber_get_publisher	サブスクリイバの証明書を使用してパブリッシャを作成する
renl_create_subscriber	RENL のサブスクリイバ側を作成する
renl_cancel_subscriber	RENL をキャンセルする

公開および購読ディスパッチャー API 関数

公開および購読ディスパッチャーの API には、表 4-4 に示す 1 つの定義と 4 つの関数があります。

表 4-4 ENS の公開および購読ディスパッチャー API の関数

定義 / 関数	説明
pas_dispatcher_t	公開および購読ディスパッチャーの定義
pas_dispatcher_new	ディスパッチャーを作成する
pas_dispatcher_delete	pas_dispatcher_new で作成したディスパッチャーを破棄する
pas_dispatch	イベント通知環境のディスパッチループを開始する
pas_shutdown	pas_dispatch で開始したイベント通知環境のディスパッチループを停止する

カスタムアプリケーションの作成と実行

ユーザ独自のパブリッシャアプリケーションおよびサブスクライバアプリケーションの作成を支援するために、この製品にはコーディング例が付属しています。コーディング例のリストについては、第5章「イベント通知サービス API リファレンス」の 113 ページの「コーディング例」を参照してください。ここでは、コーディング例、API のインクルード(ヘッダ)ファイル、およびカスタムプログラムの作成と実行に必要なライブラリの場所を示します。

- コーディング例の場所
- インクルードファイルの場所
- 動的にリンクまたは共有されるライブラリ
- 実行時ライブラリパス変数

コーディング例の場所

この製品には、この製品を初めて使うユーザを支援するために、4つの簡単なサンプルプログラムが付属しています。コーディング例は、次のディレクトリにあります。

```
/opt/SUNWics5/cal/csapi/samples/ens
```

インクルードファイルの場所

パブリッシャ API およびサブスクライバ API には、次のインクルード(ヘッダ)ファイルがあります。publisher.h、subscriber.h、および pasdisp.h(公開および購読ディスパッチャー)。これらのファイルは、CSAPI include ディレクトリにあります。デフォルトの include パスは、次のとおりです。

```
/opt/SUNWics5/cal/csapi/include
```

動的にリンクまたは共有されるライブラリ

カスタムコードは、動的にリンクされる libens ライブラリとリンクする必要があります。このライブラリには、パブリッシャおよびサブスクライバの API が実装されています。一部のプラットフォームでは、libens に依存するすべてのライブラリを、リンク指令に含める必要があります。次にそれらの従属ライブラリを順番に示します。

1. libgap
2. libcyrus

3. libyasr
4. libasync
5. libnspr3
6. libplsd4
7. libplc3

上記のライブラリは、iPlanet Calendar Server で使用されるため、サーバの bin ディレクトリに配置されています。デフォルトの libens のパスは、次のとおりです。

```
/opt/SUNWics5/cal/bin
```

注 NT の場合、パブリッシャアプリケーションおよびサブスクライバアプリケーションを作成するには、上記のすべてのライブラリに対応するアーカイブファイル (.lib ファイル) も必要になります。アーカイブファイルは、CSAPI のライブラリディレクトリである lib にあります。デフォルトの lib のパスは、次のとおりです。

```
drive:\Program Files\iPlanet\cal\csapi\lib
```

実行時ライブラリパス変数

カスタムプログラムが /opt/SUNWics5/cal/bin ディレクトリ内の必要な実行時ライブラリを検索できるように、ユーザ環境の実行時ライブラリのパス変数にこのディレクトリを含める必要があります。この変数の名前は、プラットフォームによって異なります。

- SunOS および Linux の場合は、LD_LIBRARY_PATH
- NT の場合は、PATH
- HPUNIX の場合は、SHLIB_PATH

イベント通知サービス API リファレンス

この章では、ENS API について説明します。この章は次の 4 つの節に分かれています。

- パブリッシャ API
- サブスクライバ API
- 公開および購読ディスプレイャー API
- コーディング例

パブリッシャ API の関数リスト

パブリッシャ関数について、次の表 5-1 で説明します。

表 5-1 ENS パブリッシャ API の関数リスト

定義 / 関数	説明
<code>publisher_t</code>	パブリッシャの定義
<code>publisher_cb_t</code>	非同期呼び出しの結果を確認する汎用コールバック関数
<code>publisher_new_a</code>	新規の非同期パブリッシャを作成する
<code>publisher_new_s</code>	新規の同期パブリッシャを作成する
<code>publish_a</code>	非同期通知を通知サービスに送信する
<code>publish_s</code>	同期通知を通知サービスに送信する
<code>publisher_delete</code>	パブリッシャのセッションを終了する
<code>publisher_get_subscriber</code>	パブリッシャの証明書を使用してサブスクライバを作成する
<code>renl_create_publisher</code>	RENL を作成する。 <code>end2end_ack</code> の呼び出しが有効になる

表 5-1 ENS パブリッシャ API の関数リスト (続き)

renl_cancel_publisher	RENL をキャンセルする
-----------------------	---------------

サブスクライバ API の関数リスト

サブスクライバ関数について、次の表 5-2 で説明します。

表 5-2 ENS サブスクライバ API の関数リスト

定義 / 関数	説明
subscriber_t	サブスクライバの定義
subscription_t	購読の定義
subscriber_cb_t	非同期呼び出しの結果を確認する汎用コールバック関数
subscriber_notify_cb_t	通知の受信時に呼び出される同期コールバック
subscriber_new_a	新規非同期サブスクライバを作成する
subscriber_new_s	新規同期サブスクライバを作成する
subscribe_a	非同期購読を確立する
unsubscribe_a	非同期購読をキャンセルする
subscriber_delete	サブスクライバを終了する
subscriber_get_publisher	サブスクライバの証明書を使用してパブリッシャを作成する
renl_create_subscriber	RENL のサブスクライバ側を作成する
renl_cancel_subscriber	RENL をキャンセルする

公開および購読ディスパッチャーの関数リスト

公開および購読ディスパッチャー関数について、次の表 5-3 で説明します。

表 5-3 ENS 公開および購読ディスパッチャーの関数リスト

定義 / 関数	説明
pas_dispatcher_t	公開および購読ディスパッチャーの定義
pas_dispatcher_new	ディスパッチャーを作成する
pas_dispatcher_delete	pas_dispatcher_new で作成したディスパッチャーを削除する
pas_dispatch	イベント通知環境のディスパッチループを開始する

表 5-3 ENS 公開および購読ディスパッチャーの関数リスト (続き)

pas_shutdown	pas_dispatch で開始したイベント通知環境のディスパッチループを停止する
--------------	---

パブリッシャ API

パブリッシャ API は、次の 1 つの定義と 9 つの関数から構成されています。

- publisher_t
- publisher_cb_t
- publisher_new_a
- publisher_new_s
- publish_a
- publish_s
- publisher_delete
- publisher_get_subscriber
- renl_create_publisher
- renl_cancel_publisher

publisher_t

目的

パブリッシャ

構文

```
typedef struct enc_struct publisher_t;
```

パラメータ

なし

戻り値

なし

publisher_cb_t

目的

非同期呼び出しの結果を確認するため、ENS によって呼び出される汎用コールバック関数

構文

```
typedef void (*publisher_cb_t) (void *arg, int rc, void *data);
```

パラメータ

arg	呼び出し元によって渡されるコンテキスト変数
rc	リターンコード
data	パブリッシャの作成の場合、新しく作成されたコンテキストが含まれる

戻り値

なし

publisher_new_a

目的

新規非同期パブリッシャを作成します。

構文

```
void publisher_new_a (pas_dispatcher_t *disp,  
                    void *worker,  
                    const char *host,  
                    unsigned short port,  
                    publisher_cb_t cbdone,  
                    void *cbarg);
```


パラメータ

disp	pas_dispatcher_new によって返される P&S スレッドプールコンテキスト
worker	アプリケーションワーカー。NULL 以外の場合、ENS によって作成された既存のワーカーにグループ化され、現在のパブリッシャセッションで使用される。これは、複数のスレッドによってパブリッシャデータが同時にアクセスされるのを防ぐために使用される
host	通知サーバのホスト名
port	通知サーバのポート番号
cbdone	パブリッシャが正常に作成されたとき、または作成できなかったときに呼び出されるコールバック cbdone には次の 3 つのパラメータがある <ul style="list-style-type: none"> • cbarg 最初の引数 • 状態コード ゼロ以外の場合は、パブリッシャが作成できなかったことを示し、値は失敗の原因を表す • 新規アクティブパブリッシャ
cbarg	cbdone の最初の引数

戻り値

なし。cbdone コールバックの 3 番目の引数として、新規アクティブパブリッシャが渡されます。

publisher_new_s

目的

新規同期パブリッシャを作成します。

構文

```
publisher_t *publisher_new_s (pas_dispatcher_t *disp,
                             void *worker,
                             const char *host,
                             unsigned short port);
```

パラメータ

disp	pas_dispatcher_new によって返される P&S スレッドプールコンテキスト
worker	アプリケーションワーカー。NULL 以外の場合、ENS によって作成された既存のワーカーにグループ化され、現在のパブリッシャセッションで使用される。また、複数のスレッドによってパブリッシャデータが同時にアクセスされることを防ぐために使用する
host	通知サーバのホスト名
port	通知サーバのポート

戻り値

新規アクティブパブリッシャ (publisher_t)

publish_a

目的

非同期通知を通知サービスに送信します。

構文

```
void publish_a (publisher_t *publisher,  
               const char *event_ref,  
               const char *data,  
               unsigned int datalen,  
               publisher_cb_t cbdone,  
               publisher_cb_t end2end_ack,  
               void *cbarg,  
               unsigned long timeout);
```

パラメータ

<code>publisher_t</code>	アクティブなパブリッシャ
<code>event_ref</code>	イベント参照。変更されたリソースを識別する URI
<code>data</code>	イベントデータ。通知メッセージの本文。MIME オブジェクトであることが必要。通知サービスに対して不透明で、通知サービスはイベントのサブスクリバへの中継だけを行う
<code>datalen</code>	バイト単位のデータ長
<code>cbdone</code>	通知サービスによってデータが受け入れられた場合、または受け入れられなかったと見なされた場合に呼び出されるコールバック。通知が受け入れられるかどうかは、使用するプロトコルによって左右される。プロトコルには、トランスポート送達確認 (TCP) や独自の送達確認メカニズムを選択できる
<code>end2end_ack</code>	RENL 内のサブスクリバピアから送達確認を受信したあとに呼び出されるコールバック関数。RENL のコンテキスト内でだけ使用される
<code>cbarg</code>	<code>cbdone</code> または <code>end2end_ack</code> が呼び出されるときの最初の引数
<code>timeout</code>	RENL の完了を待機する時間の長さ

戻り値
なし

publish_s

目的

同期通知を通知サービスに送信します。

構文

```
int publish_s (publisher_t *publisher,
              const char *event_ref,
              const char *data,
              unsigned int datalen);
```

パラメータ

<code>publisher</code>	アクティブなパブリッシャ
<code>event_ref</code>	イベント参照。変更されたリソースを識別する URI
<code>data</code>	イベントデータ。通知メッセージの本文。MIME オブジェクトであることが必要。通知サービスに対して不透明で、通知サービスはイベントのサブスクライバへの中継だけを行う
<code>datalen</code>	バイト単位のデータ長

戻り値

成功した場合はゼロ、失敗した場合は障害コード。RENL の場合、サブスクライバが通知を完全に処理し、送達確認が完了するまで、呼び出しは返されません。

publisher_delete

目的

パブリッシャのセッションを終了します。

構文

```
void publisher_delete (publisher_t *publisher);
```

パラメータ

<code>publisher</code>	削除対象のパブリッシャ
------------------------	-------------

戻り値

なし

publisher_get_subscriber

目的

パブリッシャの証明書を使ってサブスクライバを作成します。

構文

```
struct subscriber_struct * publisher_get_subscriber(publisher_t  
*publisher);
```

パラメータ

<code>publisher</code>	サブスクライバを作成するときに使用される証明書を持つパブリッシャ
------------------------	----------------------------------

戻り値

作成に成功した場合はそのサブスクライバ、失敗した場合は `NULL`。作成に失敗した場合は、`subscriber_new` を使用してサブスクライバを作成します。

`renl_create_publisher`

目的

`RENL` を宣言します。これにより `end2end_ack` の呼び出しが有効になります。この呼び出しが返されたあとは、指定されたパブリッシャおよびサブスクライバに対応する送達確認が受信されたときに、`end2end_ack` 引数が呼び出されます。

構文

```
void renl_create_publisher (publisher_t *publisher,
                           const char *renl_id,
                           const char *subscriber,
                           publisher_cb_t cbdone,
                           void *cbarg);
```

パラメータ

<code>publisher</code>	アクティブなパブリッシャ
<code>renl_id</code>	一意の <code>RENL</code> 識別子。これにより、2つのピア間に複数の <code>RENL</code> を設定できる
<code>subscriber</code>	認証されたピアの ID
<code>cbdone</code>	<code>RENL</code> が確立されたときに呼び出されるコールバック
<code>cbarg</code>	<code>cbdone</code> が呼び出されるときの <code>cbdone</code> の最初の引数

戻り値

なし

renl_cancel_publisher

目的

RENL をキャンセルします。キャンセルしても、通知は送信されます。ただし、クライアントの送達確認が着信しても、パブリッシャの `end2end_ack` 引数は呼び出されません。パブリッシャを削除すると、RENL がすべて自動的に削除されます。このため、パブリッシャを削除する前に、この関数を呼び出して RENL 関連のメモリを解放する必要があります。

構文

```
void renl_cancel_publisher (renl_t *renl);
```

パラメータ

renl	キャンセル対象の RENL
------	---------------

戻り値

なし

サブスクライバ API

サブスクライバ API は、次の 2 つの定義と 10 の関数から構成されています。

- `subscriber_t`
- `subscription_t`
- `subscriber_cb_t`
- `subscriber_notify_cb_t`
- `subscriber_new_a`
- `subscriber_new_s`
- `subscribe_a`
- `unsubscribe_a`
- `subscriber_delete`
- `subscriber_get_publisher`
- `renl_create_subscriber`
- `renl_cancel_subscriber`

subscriber_t

目的

サブスクライバ

構文

```
typedef struct enc_struct subscriber_t;
```

パラメータ

なし

戻り値

なし

subscription_t

目的

購読

構文

```
typedef struct subscription_struct subscription_t;
```

パラメータ

なし

戻り値

なし

subscriber_cb_t

目的

非同期呼び出しの結果を確認するために、ENS によって呼び出される汎用コールバック関数

構文

```
typedef void (*subscriber_cb_t) (void *arg,  
                                int rc,  
                                void *data);
```

パラメータ

arg	呼び出し元によって渡されるコンテキスト変数
rc	リターンコード
data	サブスクリイバの作成の場合、新しく作成されたコンテキストが含まれる

戻り値

なし

subscriber_notify_cb_t

目的

通知の受信時に呼び出されるサブスクリイバコールバック

構文

```
typedef void (*subscriber_notify_cb_t) (void *arg,  
                                       char *event,  
                                       char *data,  
                                       int datalen);
```

パラメータ

arg	サブスクリイバに渡されるコンテキストポインタ (notify_arg)
event	イベント参照 (URI)。この通知イベント参照は購読と対応する。ただし、uidなどのイベント属性と呼ばれる情報が追加されている場合もある
data	通知の本文。MIME オブジェクト
datalen	データの長さ

戻り値

成功した場合はゼロ、失敗した場合はゼロ以外の値

subscriber_new_a

目的

新規非同期サブスクリイバを作成する

構文

```
void subscriber_new_a (pas_dispatcher_t *disp,
                      void *worker,
                      const char *host,
                      unsigned short port,
                      subscriber_cb_t cbdone,
                      void *cbarg);
```

パラメータ

disp	pas_dispatcher_new によって返されるスレッドディスパッチャーコンテキスト
worker	アプリケーションワーカー。NULL 以外の場合、ENS によって作成された既存のワーカーにグループ化され、現在のサブスクリバセッションで使用される。また、複数のスレッドによってサブスクリバデータが同時にアクセスされるのを防ぐために使用する。呼び出し元が GDisp コンテキストを作成して振り分ける場合にだけ使用できる
host	通知サーバのホスト名または IP アドレス
port	購読サービスのポート番号
cbdone	サブスクリバセッションがアクティブになり、購読依頼を発行できるようになったときに呼び出されるコールバック cbdone には次の 3 つのパラメータがあります。 <ul style="list-style-type: none"> • cbarg 最初の引数 • 状態コード ゼロ以外の場合、サブスクリバを作成できなかったことを示し、値は失敗の原因を示す • 新規アクティブサブスクリバ(subscriber_t)
cbarg	cbdone の最初の引数

戻り値

なし。cbdone コールバックの 3 番目の引数として、新規アクティブサブスクリバが渡されます。

subscriber_new_s**目的**

新規同期サブスクリバを作成します。

構文

```
subscriber_t *subscriber_new_s (pas_dispatcher_t *disp,  
                                const char *host,  
                                unsigned short port);
```

パラメータ

disp	pas_dispatcher_new によって返される公開および購読ディスパッチャ
worker	アプリケーションワーカー。NULL 以外の場合、ENS によって作成された既存のワーカーにグループ化され、現在のパブリッシャセッションで使用される。また、複数のスレッドによってパブリッシャデータが同時にアクセスされるのを防ぐために使用する。呼び出し元が GDisp コンテキストを作成して振り分ける場合にだけ使用できます
host	通知サーバのホスト名または IP アドレス
port	購読サービスのポート番号

戻り値

新規アクティブサブスクライバ (subscriber_t)

subscribe_a

目的

非同期購読を確立します。

構文

```
void subscribe_a (subscriber_t *subscriber,  
                 const char *event_ref,  
                 subscriber_notify_cb_t notify_cb,  
                 void *notify_arg,  
                 subscriber_cb_t cbdone,  
                 void *cbarg):
```

パラメータ

subscriber	サブスクライバ
event_ref	イベント参照。イベントのソースを識別する URI
notify_cb	この購読に対応する通知を受信したときに呼び出されるコールバック
notify_arg	notify_arg の最初の引数。サブスクライバはアクティブな状態のまま、任意のスレッドから任意のタイミングに呼び出される場合がある
cbdone	購読依頼が完了したときに呼び出される。次の 3 つのパラメータを持つ <ul style="list-style-type: none"> • cbarg (以下の説明を参照) • 状態コード • 不透明な購読オブジェクトへのポインタ
cbarg	cbdone の最初の引数

戻り値

なし

unsubscribe_a

目的

非同期購読をキャンセルします。

構文

```
void unsubscribe_a (subscriber_t *subscriber,
                  subscription_t *subscription,
                  subscriber_cb_t cdone,
                  void *cbarg);
```

パラメータ

subscriber	削除対象のサブスクリイバ
subscription	キャンセル対象の購読
cbdone	購読の中止が完了したときに呼び出される。次の3つのパラメータを持つ <ul style="list-style-type: none">• cbarg (以下の説明を参照)• 状態コード• 不透明購読オブジェクトへのポインタ
cbarg	cbdone の最初の引数

戻り値

なし

subscriber_delete

目的

サブスクリイバを削除します。

構文

```
void subscriber_delete (subscriber_t *subscriber);
```

パラメータ

subscriber	削除対象のサブスクリイバ
------------	--------------

戻り値

なし

subscriber_get_publisher

目的

サブスクリイバの証明書を使ってパブリッシャを作成します。

構文

```
struct publisher_struct *subscriber_get_publisher (subscriber_t
*subscriber);
```

パラメータ

subscriber	パブリッシャを作成するときに使用される証明書を持つサブスクライバ
------------	----------------------------------

戻り値

作成に成功した場合はそのパブリッシャ、失敗した場合は NULL。作成に失敗した場合は、publisher_new を使用してサブスクライバを作成します。

renl_create_subscriber**目的**

RENL のサブスクライバ側を作成します。

構文

```
renl_t *renl_create_subscriber (subscription_t *subscription,
                                const char *renl_id,
                                const char *publisher);
```

パラメータ

subscription	購読
renl_id	一意の RENL 識別子。これにより、2つのピア間に複数の RENL を設定できる
publisher	認証されたピアの ID

戻り値

不透明 RENL オブジェクト

renl_cancel_subscriber

目的

RENL をキャンセルします。ただし、購読はキャンセルされません。この購読に対して以降の送信確認が行われないことを ENS に知らせます。この関数により、RENL オブジェクトが破棄されるため、アプリケーションはこの RENL をそれ以降使用できなくなります。購読をキャンセルすると、RENL はすべて自動的に削除されます。このため、サブスクライバを削除する前に、この関数を呼び出して RENL 関連のメモリを解放する必要はありません。

構文

```
void renl_cancel_subscriber (renl_t *renl);
```

パラメータ

renl キャンセル対象の RENL

戻り値

なし

公開および購読ディスパッチャー API

公開および購読ディスパッチャー API は、次の 1 つの定義と 4 つの関数から構成されています。

- pas_dispatcher_t
- pas_dispatcher_new
- pas_dispatcher_delete
- pas_dispatch
- pas_shutdown

注 サポートされているスレッドディスパッチャーは、GDisp (libasync) だけです。

pas_dispatcher_t

目的

公開および購読ディスパッチャー

構文

```
typedef struct pas_dispatcher_struct pas_dispatcher_t;
```

パラメータ

なし

戻り値

なし

pas_dispatcher_new

目的

ディスパッチャーを作成または通知します。

構文

```
pas_dispatcher_t *pas_dispatcher_new (void *disp);
```

パラメータ

disp	ディスパッチャーコンテキスト。NULL の場合、通知のディスパッチを開始するには、アプリケーションから pas_dispatch を呼び出す必要がある NULL 以外の場合、libasync ディスパッチャーが使われる
------	--

戻り値

パブリッシャまたはサブスクライバを作成するときに使用するディスパッチャー (pas_dispatcher_t)

pas_dispatcher_delete

目的

pas_dispatcher_new で作成したディスパッチャーを削除します。

構文

```
void pas_dispatcher_delete (pas_dispatcher_t *disp);
```

パラメータ

disp	イベント通知クライアント環境
------	----------------

戻り値

なし

pas_dispatch

目的

イベント通知環境のディスパッチループを開始します。アプリケーションで独自のスレッドプールを使用している場合は何も起きません。

構文

```
void pas_dispatch (pas_dispatcher_t *disp);
```

パラメータ

disp	新規ディスパッチャ
------	-----------

戻り値

なし

pas_shutdown

目的

pas_dispatch で開始したイベント通知環境のディスパッチループを停止します。アプリケーションが提供したディスパッチャーを pas_dispatcher_new に渡していた場合には何も起きません。

構文

```
void pas_shutdown (pas_dispatcher_t *disp);
```


パラメータ

disp 停止対象のディスパッチャーコンテキスト

戻り値

なし

コーディング例

iPlanet Calendar Server には完全な ENS 実装が付属しています。必要であれば、これらの API を使用して ENS をカスタマイズできます。次の 4 つのコードサンプル (簡単なパブリッシャとサブスクライバ、および信頼性の高いパブリッシャとサブスクライバ) は、API の使い方を示しています。コーディング例は、この製品の次のディレクトリにあります。

```
/opt/SUNwics5/cal/csapi/samples/ens
```

簡単なパブリッシャとサブスクライバ

これらのコーディング例では、簡単な連携動作する非同期パブリッシャおよびサブスクライバを確立します。

パブリッシャのコードサンプル

```
/*
 * Copyright 2000 by Sun Microsystems, Inc.
 * All rights reserved
 *
 * apub : 簡単な連携動作する非同期パブリッシャ
 *
 * 構文 :
 *   apub ホスト名 ポート番号
 */

#include <stdlib.h>
#include <stdio.h>

#include "pasdisp.h"
#include "publisher.h"
```

```
static pas_dispatcher_t *disp = NULL;
static publisher_t *_publisher = NULL;
static int _shutdown = 0;

static void _read_stdin();

static void _exit_usage()
{
    printf("\n 用法 : \napub ホスト名 ポート番号 \n");
    exit(5);
}

static void _exit_error(const char *msg)
{
    printf("%s\n", msg);
    exit(1);
}

static void _call_shutdown()
{
    _shutdown = 1;
    pas_shutdown(disp);
}

static void _open_ack(void *arg, int rc, void *enc)
{
    _publisher = (publisher_t *)enc;
    (void *)arg;

    if (!_publisher)
    {
        printf(" 状態 %d のため、パブリッシャの作成に失敗しました \n", rc);
        _call_shutdown();
        return;
    }

    _read_stdin();

    return;
}

static void _publish_ack(void *arg, int rc, void *ignored)
{
    (void *)ignored;

    free(arg);

    if (rc != 0)
    {
        printf(" 状態 %d のため、公開に失敗しました \n", rc);
        _call_shutdown();
    }
}
```

```

        return;
    }
    _read_stdin();
    return;
}

static void _read_stdin()
{
    static char input[1024];
    printf("apub> ");
    fflush(stdout);
    while (!_shutdown)
    {
        if ( !fgets(input, sizeof(input), stdin) )
        {
            continue;
        } else {
            char *message;
            unsigned int message_len;

            input[strlen(input) - 1] = 0; /* Strip off the \n */
            if (*input == '.' && input[1] == 0)
            {
                publisher_delete(_publisher);
                _call_shutdown();
                break;
            }

            message = strdup(input);
            message_len = strlen(message);
            publish(_publisher, "enp://yoyo.com/xyz", message,
                message_len,
                _publish_ack, NULL, (void *)message, 0);
            return;
        }
    }
    return;
}

main(int argc, char **argv)
{
    unsigned short port = 7997;
    char host[256];

    if (argc < 2) _exit_usage();

```

```

if (*(argv[1]) == '0')
{
    strcpy(host, "127.0.0.1");
} else {
    strcpy(host, argv[1]);
}

if (argc > 2)
{
    port = (unsigned short)atoi(argv[2]);
}

disp = pas_dispatcher_new(NULL);
if (disp == NULL) _exit_error(" デイスパッチャーを作成できません ");
publisher_new_a(disp, NULL, host, port, _open_ack, disp);
pas_dispatch(disp);

_shutdown = 1;
pas_dispatcher_delete(disp);
exit(0);
}

```

サブスクリイバのコードサンプル

```

/*
 * Copyright 1997 by Sun Microsystems, Inc.
 * All rights reserved
 *
 * asub : 非同期サブスクリイバの例
 *
 * 構文 :
 *   asub ホスト名 ポート番号
 */

#include <stdlib.h>
#include <stdio.h>

#include "pasdisp.h"
#include "subscriber.h"

static pas_dispatcher_t *disp = NULL;
static subscriber_t *_subscriber = NULL;
static subscription_t *_subscription = NULL;
static renl_t *_renl = NULL;

```

```
static void _exit_usage()
{
    printf("\n  使用法 : \nasub ホスト名 ポート番号 \n");
    exit(5);
}

static void _exit_error(const char *msg)
{
    printf("%s\n", msg);
    exit(1);
}

static void _subscribe_ack(void *arg, int rc, void *subscription)
{
    (void) arg;

    if (!rc)
    {
        _subscription = subscription;
        printf("購読依頼に成功しました \n");
    } else {
        printf("購読依頼に失敗しました - 状態 %d\n", rc);
        pas_shutdown(dispatch);
    }
}

static void _unsubscribe_ack(void *arg, int rc, void *ignored)
{
    (void *) ignored;
    (void *) arg;

    if (rc != 0)
    {
        printf("購読の中止に失敗しました - 状態 %d\n", rc);
    }

    subscriber_delete(_subscriber);
    pas_shutdown(dispatch);
}

static int _handle_notify(void *arg, char *url, char *str, int len)
{
    (void *) arg;
    printf("[%s] %.*s\n", url, len, (str) ? str : "(null)");
    return 0;
}

static void _open_ack(void *arg, int rc, void *enc)
{
    _subscriber = (subscriber_t *) enc;
}
```

```
(void *)arg;
if (rc)
{
    printf(" 状態 %d のため、サブスクリイバの作成に失敗しました \n", rc);
    pas_shutdown(dispatch);
    return;
}

subscribe(_subscriber, "enp://yoyo.com/xyz",
          _handle_notify, NULL,
          _subscribe_ack, NULL);

return;
}

static void _unsubscribe(int sig)
{
    (int)sig;
    unsubscribe(_subscriber, _subscription, _unsubscribe_ack, NULL);
}

main(int argc, char **argv)
{
    unsigned short port = 7997;
    char host[256];

    if (argc < 2) _exit_usage();
    if (*(argv[1]) == '0')
    {
        strcpy(host, "127.0.0.1");
    } else {
        strcpy(host, argv[1]);
    }
    if (argc > 2)
    {
        port = (unsigned short)atoi(argv[2]);
    }

    dispatch = pas_dispatcher_new(NULL);
    if (dispatch == NULL) _exit_error(" デイスペッチャーを作成できません ");
    subscriber_new_a(dispatch, NULL, host, port, _open_ack, NULL);
    pas_dispatch(dispatch);
    pas_dispatcher_delete(dispatch);
    exit(0);
}
```

信頼性の高いパブリッシャとサブスクライバ

これらのコーディング例では、信頼性の高い非同期パブリッシャおよびサブスクライバを確立します。

信頼性の高いパブリッシャのサンプル

```

/*
 * Copyright 2000 by Sun Microsystems, Inc.
 * All rights reserved
 *
 * rpub : 簡単で *信頼性の高い* 連携動作する非同期パブリッシャ
 * rsub (信頼性の高いサブスクライバ) と組み合わせて
 * 使用するよう設計されています。
 *
 * 構文 :
 *   rpub ホスト名 ポート番号
 */

#include <stdlib.h>
#include <stdio.h>

#include "pasdisp.h"
#include "publisher.h"

static pas_dispatcher_t *disp = NULL;
static publisher_t *_publisher = NULL;
static int _shutdown = 0;
static renl_t *_renl;

static void _read_stdin();

static void _exit_usage()
{
    printf("\n 用法 : \nrpub ホスト名 ポート番号 \n");
    exit(5);
}

static void _exit_error(const char *msg)
{
    printf("%s\n", msg);
    exit(1);
}

static void _call_shutdown()
{
    _shutdown = 1;
    pas_shutdown(disp);
}

```

```
static void _renl_create_cb(void *arg, int rc, void *ignored)
{
    (void *)arg;
    (void *)ignored;

    if (!_publisher)
    {
        printf("REN1 の作成に失敗しました - 状態 %d\n", rc);
        _call_shutdown();
        return;
    }

    _read_stdin();
    return;
}

static void _publisher_new_cb(void *arg, int rc, void *enc)
{
    _publisher = (publisher_t *)enc;
    (void *)arg;

    if (!_publisher)
    {
        printf("パブリッシャの作成に失敗しました - 状態 %d\n", rc);
        _call_shutdown();
        return;
    }

    renl_create_publisher(_publisher, "renl_id", NULL,
                          _renl_create_cb, NULL);

    return;
}

static void _recv_ack(void *arg, int rc, void *ignored)
{
    (void *)ignored;

    if (rc < 0)
    {
        printf("応答のタイムアウト\n");
    } else if (rc == 0) {
        printf("応答を受信しました\n");
    }
    fflush(stdout);

    _read_stdin();

    free(arg);
}
```



```

    return;
}
static void _read_stdin()
{
    static char input[1024];
    printf("rpub> ");
    fflush(stdout);
    while (!_shutdown)
    {
        if (!fgets(input, sizeof(input), stdin) )
        {
            continue;
        } else {
            char *message;
            unsigned int message_len;

            input[strlen(input) - 1] = 0; /*\n を削除する*/
            if (*input == '.' && input[1] == 0)
            {
                publisher_delete(_publisher);
                _call_shutdown();
                break;
            }

            message = strdup(input);
            message_len = strlen(message);

            /* タイムアウト時間 5 秒 */
            publish(_publisher, "enp://yoyo.com/xyz",
                message, message_len,
                NULL, _recv_ack, message, 5000);

            return;
        }
    }
    return;
}
main(int argc, char **argv)
{
    unsigned short port = 7997;
    char host[256];

    if (argc < 2) _exit_usage();
    if (*(argv[1]) == '0')
    {
        strcpy(host, "127.0.0.1");
    } else {

```

```

        strcpy(host, argv[1]);
    }
    if (argc > 2)
    {
        port = (unsigned short)atoi(argv[2]);
    }

    disp = pas_dispatcher_new(NULL);
    if (disp == NULL) _exit_error(" デイスパッチャーを作成できません ");
    publisher_new_a(disp, NULL, host, port, _publisher_new_cb,
                   NULL);

    pas_dispatch(disp);
    _shutdown = 1;
    pas_dispatcher_delete(disp);
    exit(0);
}

```

信頼性の高いサブスクリイバのサンプル

```

/*
 * Copyright 1997 by Sun Microsystems, Inc.
 * All rights reserved
 *
 * rsub : 非同期サブスクリイバの例
 *
 * 構文 :
 * rsub ホスト名 ポート番号
 */

#include <stdlib.h>
#include <stdio.h>

#include "pasdisp.h"
#include "subscriber.h"

static pas_dispatcher_t *disp = NULL;
static subscriber_t *_subscriber = NULL;
static subscription_t *_subscription = NULL;
static renl_t *_renl = NULL;

static void _exit_usage()
{
    printf("\n 用法 : \nrsub ホスト名 ポート番号 \n");
    exit(5);
}

```

```
static void _exit_error(const char *msg)
{
    printf("%s\n", msg);
    exit(1);
}

static void _subscribe_ack(void *arg, int rc, void *subscription)
{
    (void) arg;

    if (!rc)
    {
        _subscription = subscription;
        printf("購読依頼に成功しました\n");
        _renl = renl_create_subscriber(_subscription, "renl_id",
NULL);
    } else {
        printf("購読依頼に失敗しました - 状態 %d\n", rc);
        pas_shutdown(dispatch);
    }
}

static void _unsubscribe_ack(void *arg, int rc, void *ignored)
{
    (void *) ignored;
    (void *) arg;

    if (rc != 0)
    {
        printf("購読の中止に失敗しました - 状態 %d\n", rc);
    }

    subscriber_delete(_subscriber);
    pas_shutdown(dispatch);
}

static int _handle_notify(void *arg, char *url, char *str, int len)
{
    (void *) arg;
    printf("[%s] %.*s\n", url, len, (str) ? str : "(null)");
    return 0;
}

static void _open_ack(void *arg, int rc, void *enc)
{
    _subscriber = (subscriber_t *) enc;
```

```

(void *)arg;
if (rc)
{
    printf(" 状態 %d のため、サブスクリイバの作成に失敗しました \n", rc);
    pas_shutdown(dispatch);
    return;
}

subscribe(_subscriber, "enp://yoyo.com/xyz", _handle_notify,
          NULL, _subscribe_ack, NULL);

return;
}

static void _unsubscribe(int sig)
{
    (int) sig;
    unsubscribe(_subscriber, _subscription, _unsubscribe_ack, NULL);
}

main(int argc, char **argv)
{
    unsigned short port = 7997;
    char host[256];

    if (argc < 2) _exit_usage();
    if (*(argv[1]) == '0')
    {
        strcpy(host, "127.0.0.1");
    } else {
        strcpy(host, argv[1]);
    }
    if (argc > 2)
    {
        port = (unsigned short)atoi(argv[2]);
    }

    dispatch = pas_dispatcher_new(NULL);

    if (dispatch == NULL) _exit_error(" デイスペッチャーを作成できません ");
    subscriber_new_a(dispatch, NULL, host, port, _open_ack, NULL);
    pas_dispatch(dispatch);
    pas_dispatcher_delete(dispatch);
    exit(0);
}

```

プロキシ認証 SDK の概要

この章では、iPlanet Calendar Server 5.0 プロキシ認証 SDK (authSDK) について説明します。この章には、次のトピックが含まれます。

- SDK を使用するユーザ
- authSDK とは
- アーキテクチャ
- 関数の概要

SDK を使用するユーザ

インストールした iPlanet Calendar Server 5.0 にポータルサービスが含まれている場合は、authSDK を使ってそのポータルを Calendar Server に統合できます。ポータルシステムでユーザが認証されると、authSDK 関数によって Calendar Server に通知され、再度認証することなく、さまざまなサービスへのアクセスをユーザに許可します。

authSDK とは

authSDK は、5 つの関数をエクスポートする DLL/ 共有オブジェクトで構成されています。

インストールパッケージには、次のファイルが含まれています。これらは *server root/bin/authsdk* にあります。

- libicsexp10.so または LIBICSEXP10.DLL。SDK ライブラリ
- expapi.h。API ユーザ用のヘッダーファイル

アーキテクチャ

authSDK は、非常にシンプルです。初期化、検索、およびクリーンアップで構成されています。また、関数 `CEXP_SetHttpPort` により、authSDK で標準以外のポート番号の使用が可能になり、関数 `CEXP_GetVersion` により、顧客またはテクニカルサポートに連絡する際に必要な authSDK バージョン番号を取得できます。API 関数の詳細は、第7章「プロキシ認証 SDK リファレンス」を参照してください。

初期化

初期化するには、`CEXP_Init` を呼び出します。この関数に LDAP 情報を渡すと、ユーザが属するカレンダーサーバの検出に検索フェーズで使用される LDAP 接続が初期化されます。この接続は、スレッド環境用に設定されます。すべてのスレッドはこの接続を共有しますが、ロックは LDAP で行われるため、ほとんどの環境で十分な速度が得られます。この接続はオープン状態で維持されるため、検索が行われるたびにセットアップや終了によるコストはかかりません。

初期化によって設定されるその他の項目には、LDAP クエリーでの検索ユーザ名の照合に使用する、プログラマが指定する属性が含まれます。

検索

ユーザの新しいセッションを生成するには、検索用の関数 `CEXP_GenerateLoginURL` を使用します。検索では認証は行われません。ユーザ名と IP アドレスを使用して、セッションテーブルにそのユーザのエントリを作成し、そのセッションに関連する URL を返します。カレンダーサーバのホスト名が渡された場合、この関数はそのサーバに接続してセッションを生成します。ホスト名が渡さなかった場合は、LDAP サーバに照会してホスト名を決定します。検索で実際にユーザ認証なしでセッションを生成するには、プロキシ管理者の ID とパスワードを用意する必要があります。

クリーンアップ

スレッド環境で、メモリ、オープンしている LDAP 接続などのリソースをクリーンアップしたい場合は、`CEXP_Shutdown` を使用します。

関数の概要

表 6-1 に示すように、SDK には 5 つの関数があります。

表 6-1 プロキシ認証 SDK 関数

関数	説明
CEXP_GenerateLoginURL	有効なセッション ID を含む URL を生成する
CEXP_GetVersion	バージョン ID 文字列を生成する
CEXP_Init	SDK を初期化する
CEXP_SetHttpPort	カレンダーサーバとの接続に使用するポート番号を指定する
CEXP_Shutdown	メモリの解放、接続の解除を含む、すべてのシャットダウン手順を実行する

プロキシ認証 SDK リファレンス

この章では、iPlanet Calendar Server 5.0 のプロキシ認証 SDK (authSDK) API について説明します。この章は、プロキシ認証 SDK 関数と authSDK の使い方の 2 つの部分に分かれています。

プロキシ認証 SDK 関数のリスト

authSDK には、次の 5 つの関数があります。

- `CEXP_GenerateLoginURL`
有効なセッション ID を含む URL を生成します。
- `CEXP_GetVersion`
バージョン ID 文字列を生成します。
- `CEXP_Init`
SDK を初期化します。
- `CEXP_SetHttpPort`
カレンダーサーバとの接続に使用するポート番号を指定します。
- `CEXP_Shutdown`
メモリの解放、接続の解除を含む、すべてのシャットダウン手順を実行します。

プロキシ認証 SDK 関数

authSDK は、5 つの関数で構成されています。次のリストは、使用する順序ではなくアルファベット順に関数を並べたものです。関数の使用方法については、この章の「authSDK の使い方」を参照してください。

- CEXP_GenerateLoginURL
- CEXP_GetVersion
- CEXP_Init
- CEXP_SetHttpPort
- CEXP_Shutdown

CEXP_GenerateLoginURL

目的

特定のユーザの有効なセッション ID を含むログイン URL を返します。

構文

```
int CEXP_GenerateLoginURL (char * pszUser,  
                           char * pszClientAddress,  
                           char * pszCalendarHost,  
                           char * pszURL);
```

パラメータ

次の 4 つのパラメータがあります。

pszUser	ユーザ名の文字列
pszClientAddress	クライアントホスト IP アドレスの文字列
pszCalendarHost	カレンダーサーバのホスト名 (IP アドレスではない) の文字列
pszURL	URL を格納するバッファのポインタ

戻り値

成功した場合は 0、失敗した場合は -1。成功した場合は、pszURL バッファに有効な URL 文字列が格納されます。

CEXP_GetVersion

目的

バージョン ID 文字列を取得します。

構文

```
char * CEXP_GetVersion(void);
```

パラメータ

パラメータはありません。

戻り値

バージョン ID 文字列への参照。

CEXP_Init

目的

SDK を初期化します。

構文

```
int CEXP_Init (char * pszLdapHost,
               char * pszLdapMatchAttrib,
               char * pszLdapDN,
               unsigned int iLdapPort,
               char * pszLdapBindUser,
               char * pszLdapBindPass,
               char * pszAdminUser,
               char * pszAdminPassword);
```

パラメータ

次の 8 つのパラメータがあります。

pszLdapHost	ディレクトリサーバのホスト名の文字列
pszLdapMatchAttrib	属性名の文字列。ユーザ名の照合に使用される
pszLdapDN	ユーザレコードの検索に使用するベース DN の文字列。「DN」(Distinguished Name、識別名) は、LDAP ディレクトリエントリの名前と場所の文字列表現
iLdapPort	ディレクトリサーバのポート番号を指定する整数
pszLdapBindUser	バインド DN の文字列
pszLdapBindPass	バインド DN のパスワードの文字列

<code>pszAdminUser</code>	Calendar Server 管理者の LDAP ユーザ ID の文字列
<code>pszAdminPassword</code>	Calendar Server 管理者のパスワードの文字列

戻り値

成功した場合は 0、失敗した場合は -1。

コメント

バインド DN とパスワードが NULL の場合は、匿名 (anonymous) 検索が行われます。

CEXP_SetHttpPort

目的

カレンダーサーバとの接続に使用する HTTP ポート番号を設定します。

構文

```
void CEXP_SetHttpPort (int iHttpPort);
```

パラメータ

次の 1 つのパラメータがあります。

<code>iHttpPort</code>	ポート番号を指定する整数
------------------------	--------------

戻り値

なし

CEXP_Shutdown

目的

SDK の使用后、すべてのグローバルメモリを整理し、接続を解除し、ほかのクリーンアップ関数を実行します。

構文

```
int CEXP_Shutdown (void);
```

パラメータ

パラメータはありません。

戻り値

成功した場合は 0、失敗した場合は -1。

コメント

SDK を使用しているすべてのスレッドが完了するまではこの関数を呼び出さないでください。

authSDK の使い方

authSDK をインストールに実装するには、次の手順に従います。

1. authSDK をコードにリンクします。

authSDK を既存のコードに統合するには、呼び出し元のコードに `expapi.h` ヘッダーファイルをインクルードし、DLL/ 共有オブジェクトとリンクします。プラットフォームによっては、ほかのシステムライブラリとのリンクも必要になる場合があります。

2. ポータルの認証プログラムを使ってユーザを認証します。
3. `CEXP_Init` を呼び出します。

この関数は、authSDK 設定情報を初期化します。この呼び出しは、ほかの authSDK 関数を呼び出す前に行う必要があります。

4. オプションで、`CEXP_SetHttpPort` を呼び出します。

デフォルトでは、authSDK は標準の HTTP ポート 80 を使用します。セッション生成の接続の際に、authSDK が標準以外のポートを使用するように指定する場合は、この関数を使用します。

警告

この関数は、スレッドに対して安全ではなく、グローバル値を設定します。この関数をスレッド環境で使用する場合は、この呼び出しと `CEXP_GenerateLoginURL` の呼び出しをロックする必要があります。

5. `CEXP_GenerateLoginURL` を呼び出します。

この関数は、ユーザおよびクライアントの IP アドレスに対するセッションハンドルを生成し、Calendar Server への接続時に使用されるログイン URL を含む文字列を、割り当てられたバッファに返します。この文字列は、識別情報としての一種のトークンです。このトークンは、HTTP ヘッダーまたは JavaScript を通じて cookie または URL の形式でクライアントに提供されます。クライアントは、識別情報としてこのトークンを提示して、Calendar Server に接続します。

6. オプションで、`CEXP_Shutdown` を呼び出します。

authSDK で使用されているリソースのクリーンアップおよびシャットダウンを行う場合は、この関数を呼び出します。簡易 CGI ログインなどの一部の環境では、この関数を呼び出す必要はありませんが、API を使用するプラグインでは、リソースを再生して実行を続ける場合があります。

その他のヒント

AuthSDK を確実に使用するには、次の設定が必要です。

- `ics.conf` ファイルの `service.http.allowadminproxy` の値を「yes」にする必要があります。
- `init` メソッドに渡すパラメータ `caladmin` を、`ics.conf` ファイルの `service.admin.calmaster.userid` と同じ値にする必要があります。
- `init` メソッドに渡すパラメータ `calpass` を、`ics.conf` ファイルの `service.admin.calmaster.cred` と同じ値にする必要があります。
- 2 つのパラメータ `caladmin` および `calpass` は、ディレクトリサービスで定義されている必要があります。
- カレンダーサーバがデフォルトのポート 80 で待機していない場合は、`SetHttpPort` メソッドで適切なポート番号を指定する必要があります。

シングルサインオン認証

この章では、iPlanet Calendar Server 5.0 に組み込まれているシングルサインオン認証メカニズムについて説明します。この方式は、ほかの認証メカニズム、セッション管理、およびリソースアクセスコントロールから独立しています。この認証メカニズムを使用するには、クライアントで **cookie** をサポートし、サーバで **HTTP** をサポートしている必要があります。シングルサインオンは、アプリケーションのパフォーマンスに影響を与えません。この方式では、セッションの一元管理は必要ありません。アプリケーションは独自のセッションを管理するので、別個のタイムアウトおよび取り消しポリシーを使用できます。

この章には、次のトピックが含まれます。

- シングルサインオンとは
- シングルサインオンの制限
- 処理の流れ
- 実装の要件
- シングルサインオンの例
- 留意点

シングルサインオンとは

シングルサインオンでは、ユーザは一度署名すると複数のアプリケーションを使用できます。これらのアプリケーション間には、それぞれに設定した **cookie** を互いに認証として使用できる共通の検証システムがあるため、ユーザは各アプリケーションで別々に署名する必要はありません。

各アプリケーションでは、必要に応じて独自の照合インタフェースを使用できます。ただし、各照合機関には、ほかのアプリケーションの照合機関ルーチンが認識できる **cookie** を保存しておく必要があります。この章の「**cookie 情報**」を参照してください。

シングルサインオンの制限

シングルサインオンには前述した利点がありますが、使用に関して次のようないくつかの制限があります。

- アプリケーションに照合プロトコルを実装する必要があります。この章の「留意点」を参照してください。
- この方式は、マシンを共有する状況には適しません。
- 各アプリケーションは、同じドメインに属し、相互のシングルサインオン照合 URL にアクセスできる必要があります。
- 別のユーザに代わる場合は、ブラウザを再起動する必要があります。
これは、各ブラウザセッションで1つのユーザ ID しかサポートできないためです。
- クライアントが **cookie** をサポートしている必要があります。

処理の流れ

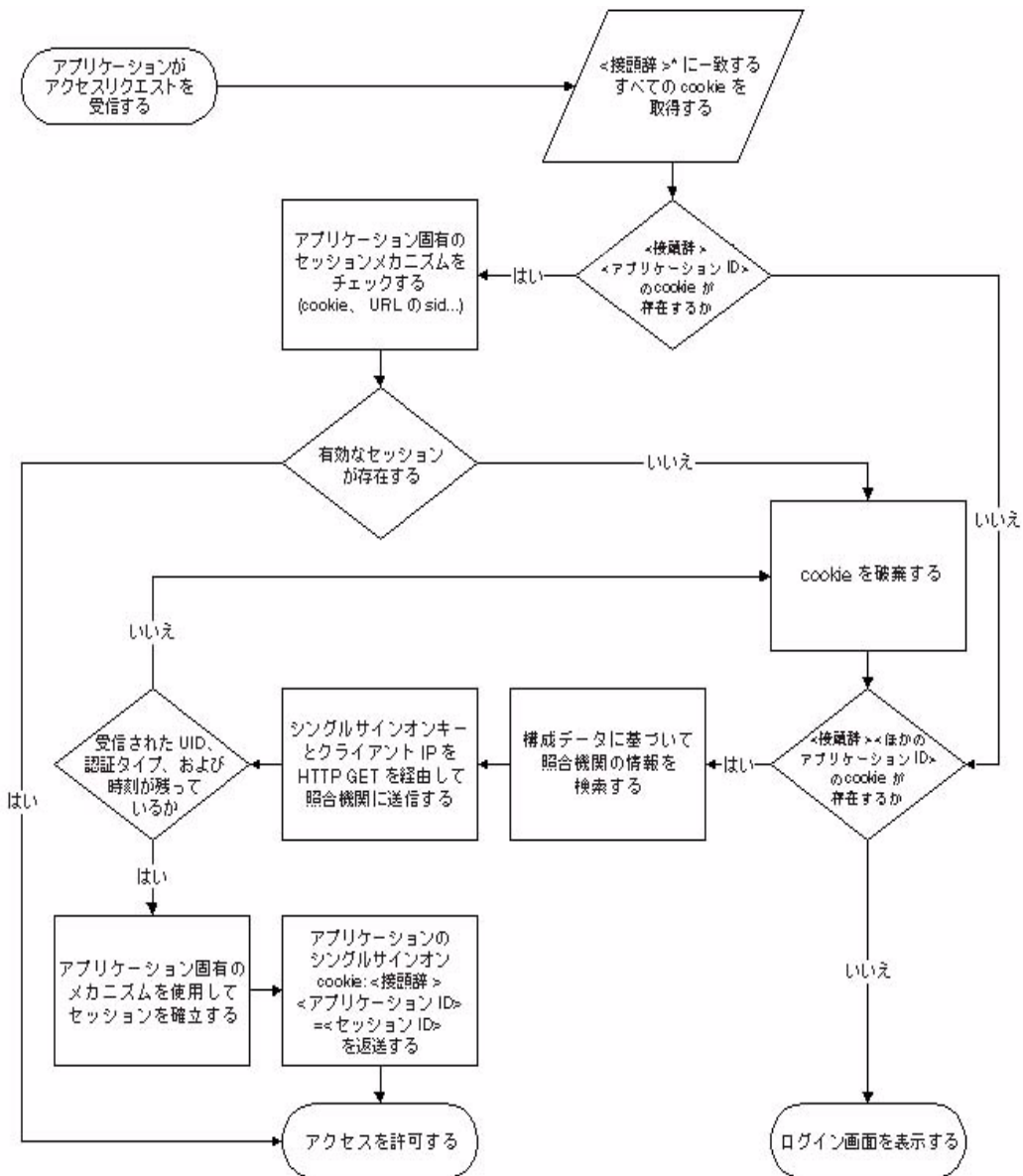
次の図に示すように、アプリケーションがアクセス要求を受信した時点で処理の流れが開始されます。共通の検証システムを持つ信頼された一連のアプリケーションは、同じ接頭辞を共有するため、プログラムは接頭辞が一致するすべての **cookie** を取り込みます。

そのアプリケーションに固有の **cookie** が見つからない場合、アプリケーションは、ほかの信頼されたアプリケーションに利用できる **cookie** があるかどうかを確認します。有効な **cookie** が見つかり、プログラムはこれらの **cookie** の情報を使用して、認証を照合し、セッションを確立します。アクセスを許可した場合は、アプリケーション固有の **cookie** を返送します。

信頼できるアプリケーションの接頭辞を持つ **cookie** がない場合、または利用可能な **cookie** が無効な場合は、ログイン画面を表示します。

図 8-1 に、シングルサインオンの処理の流れを示します。

図 8-1 シングルサインオンの処理の流れ



実装の要件

このシングルサインオンスキーマを使用するには、すべてのアプリケーションが次の要件を満たしている必要があります。

- cookie の読み込みおよび書き込みができること
- 次の追加構成パラメータをサポートしていること
 - 信頼されたアプリケーションのリスト
 - シングルサインオン
 - 接頭辞の文字列
- HTTP を経由したシングルサインオン照合プロトコルを実装していること
 - 要求：

要求には、ヘッダーにシングルサインオン cookie が含まれ、URL のパラメータとしてクライアント IP アドレスが指定されます。

```
GET verificationurl client=clientIPHTTP/1.0
```
 - キーが有効な場合の応答：

各キーと値のペアが別々の行に示された text/plain 形式の ASCII 応答。timeremaining は、オプションです。

```
fquid= ユーザ ID@ 完全修飾ドメイン名
authtype=plaintext | cert | ...
timeremaining=[ このセッションのタイムアウトまでの残り時間 ]
```
 - キーが無効な場合の応答：

text/plain 形式の ASCII メッセージ: エラー : ユーザのセッションが無効です。
- 構成ファイル `ics.conf` を編集して、一連の信頼されたアプリケーションの構成データを設定すること。構成ファイルの詳細は、『iPlanet Calendar Server 管理ガイド』を参照してください。シングルサインオン構成パラメータは、接頭辞 `sso` で始まります。

cookie 情報

cookie の形式は、次のとおりです。

構成可能な接頭辞-アプリケーションインストールの一意の識別子=シングルサインオンキー

その他の推奨設定

- ドメインを `.domain name` に設定する
- パスを `/` に設定する
- `expires` フィールドを設定しない

信頼されたアプリケーションレコード

追加構成パラメータの記憶域および形式はアプリケーション固有ですが、各レコードには次の内容が含まれている必要があります。

- この特定の信頼されたアプリケーションのインストールに一意の識別子
- アプリケーションのホストになるマシンの完全修飾ドメイン名または IP アドレス
- シングルサインオン照合プロトコルをサポートする URL

例 :`http://domain.com/VerifySSO?`

シングルサインオフパラメータ

シングルサインオフパラメータは、このアプリケーションがシングルサインオフを実行するかどうかを示すブール値です。デフォルトは、`true` です。

- `true` の場合、アプリケーションはユーザがログアウトした時点で、一連の信頼されたアプリケーションの接頭辞に一致するすべてのシングルサインオン `cookie` を削除します。
- `false` の場合、アプリケーションはユーザがログアウトした時点で、独自のシングルサインオン `cookie` のみを削除します。

接頭辞の文字列

これは、一連の信頼されたアプリケーションのすべてのアプリケーションによって共有される共通の文字列です。別の接頭辞を選択することで、ドメイン内で複数の一連の信頼されたアプリケーションを作成することができます。

シングルサインオンの例

この例は、シングルサインオンのサイクルが達成される手順を示しています。

例

参加するアプリケーションは、この一連の信頼されたアプリケーションの接頭辞として `ssogrp1` を使用します。このサークルの信頼されたアプリケーションは、WebMail、WebCal、および HRapp です。

1. John はメールをチェックします。

WebMail は、208.12.60.3 上で作業している `jsmith` からログイン要求を受信します。WebMail は、名前の一部に `ssogrp1` が含まれる `cookie` を検索しますが、該当する `cookie` は存在しません。WebMail はユーザに認証を求めるプロンプトを表示し、接頭辞文字列 `key-`、この WebMail インストールの一意の識別子 (`3fr7d`)、およびシングルサインオンキー (`13khj513k91dh`) を含む `cookie` を返送します。これらの接頭辞文字列、一意の識別子、およびシングルサインオンキーが、セッション ID となります。

返される `cookie` は、`ssogrp13fr7d=13khj513k91dh` です。

2. John は、次にカレンダーをチェックします。
 - WebCal は、208.12.60.3 上で作業している `jsmith` からログイン要求を受信します。WebCal で見つかった `ssogrp1* cookie` は、`ssogrp13fr7d=13khj513k91dh` だけでした。WebCal は、構成ファイルで次のエントリを見つけました。

```
3fr7d.verificationurl=http://webmail host/VerifySSO?
```
 - WebCal は、次の HTTP GET を発行します。

```
GET http://webmail host/VerifySSO?client=208.12.60.3 HTTP/1.0
Cookie: ssogrp13fr7d=13khj513k91dh
```
 - WebMail は、これが有効なキーであることを確認して、次の応答を返します。

```
fquid=jsmith@example.com
authtype=plaintext
timeremaining=1000
```
 - WebCal は、John 用のシングルサインオンキー `a97ads64` を生成し、`cookie` として `ssogrp11kj87f=a97ads64` を返送します。
3. John は、しばらくの間ネットサーフィンをしたあとで、もう一度カレンダーをチェックします。
 - WebCal は、208.12.60.3 上で作業している `jsmith` からログイン要求を受信します。
 - WebCal は、いくつかの `ssogrp1 cookie` を見つけました。そのうちの 1 つが、WebCal の一意のアプリケーション ID (`1kj87f`) に一致しました。
WebCal は、`cookie` として `ssogrp11kj87f=a97ads64` を返送します。

- WebCal は、セッション ID をシングルサインオンキーとして使用しているため、自身のセッションデータベースからセッション a97ads64 を得て、セッションが依然として有効であることを確認して、ユーザにアクセスを許可します。
4. John は、会社の HR アプリケーションにアクセスする必要があります。このアプリケーションは、証明書による認証を必要とします。
- HR アプリケーションは、社内で開発されたもので、照合プロトコルをサポートするように変更されています。
- HRapp は、208.12.60.3 上で作業している jsmith からログイン要求を受信します。
このアプリケーションは証明書による認証を必要としているため、クライアントに証明書の送信を求めます。
 - HRapp は、ssogrp1* cookie の確認は行いません。
 - HRapp は、John 用のセッション ID B53P997KD を生成し、cookie として ssogrp1adf38=1ka79jy5d313r を返送します。
この cookie により、ほかの参加アプリケーションは HRapp を照合機関として使用できるようになります。
5. John はもう一度メールをチェックして、ログアウトすることにしました。
- WebMail は、208.12.60.3 上で作業している jsmith からログアウト要求を受信します。
 - WebMail は、John の WebMail セッションを無効にすると同時に、すべての ssogrp1* cookie を削除します。
6. この時点では、John がアプリケーションにアクセスする場合は、もう一度ログインする必要があります。

この例の構成パラメータ

- 接頭辞:sso.appprefix="ssogrp1"
- シングルサインオフのブール値:sso.singlesignoff="true"
- アプリケーション情報:
 - WebMail:


```
appid=3fr7d
3fr7d.ip=198.93.96.111
3fr7d.verificationurl=http://Webmail ホスト/VerifySSO?
```
 - WebCal:


```
lkj87f.ip=198.93.78.103
```

```
lkj87f.verificationurl=http://Webcal ホスト /VerifySSO?
```

○ HRapp:

```
adf38.ip=198.93.70.8
```

```
adf38.verificationurl=http://HR ホスト /VerifySSO?
```

留意点

この節では、次の4つの領域に関する留意点、仮定、および要件について説明します。

- セキュリティ
- 管理
- スケーラビリティ
- パフォーマンス

セキュリティ

- シングルサインオンキーとクライアント IP アドレスの正しい組み合わせを推測するのは非常に困難であると仮定しているため、照合プロトコルに対して認証を要求する必要はありません。
- プロキシを使用してクライアント接続を確立する場合、シングルサインオンキーは唯一の防衛手段になります。そのため、キーの予測が非常に困難であることはきわめて重要です。
- セキュリティ保護されたシングルサインオンキーは、アプリケーションで生成する必要があります。
- 各シングルサインオンキーが長時間にわたって有効にならないように、アプリケーションでセッションタイムアウトを短時間に設定することも役に立ちます。
- HTTPS は、通信を保護する目的で使用できます。アプリケーションでクライアントの証明書による認証が必要な場合は、常にクライアントに対して証明書を要求する必要があります。とはいうものの、このアプリケーションは依然としてほかのアプリケーションのための照合機関の役目をすることができます。

管理

- すべてのアプリケーションは、信頼されたアプリケーションのリストを保持する必要があります。各アプリケーションは、異なるメカニズムを使用してそのリストの保管および管理を行うことができます。

- シングルサインオンキーを生成したアプリケーションだけがそのキーを無効にすることができます。管理者は、すべてのアプリケーションに対するユーザのアクセス権を簡単に無効にすることはできません。

スケーラビリティ

cookie の数の制限：

- 合計で 300 の cookie
- サーバまたはドメインごとに 20 の cookie。完全指定のホストおよびドメインは別のエントリとして計算され、それぞれの cookie の数は 20 に制限されます

パフォーマンス

- ほかのアプリケーションに対する照合要求は、一度しか行われません。アプリケーションのセッションがいったん確立されると、照合プロトコルは行われません。
- ブラウザに古い cookie (無効なシングルサインオンキーを持つ cookie) がたくさん記録されていると、新しいアプリケーションにログインするときに時間がかかることがあります。

ただし、これは次の理由により、めったに起きません。

- 各アプリケーションは、ユーザがログアウトする時点で自身の cookie を削除します (アプリケーションがシングルサインオフを実行する場合は、該当する接頭辞を含むシングルサインオン cookie をすべて削除します)。
- アプリケーションは、cookie が複数のブラウザセッションにわたって持続しないように expires フィールドを空白のままにします。

留意点

Web カレンダーアクセスプロトコル (WCAP) の概要

この章では、Web カレンダーアクセスプロトコル (WCAP) 2.0 について説明します。WCAP は、クライアントが iPlanet Calendar Server 5.0 との通信に使用する高度なコマンドベースのプロトコルです。この章は、以下の節で構成されています。

- はじめに
- このバージョンの新機能
- コマンドの概要
- コマンドの形式

はじめに

iPlanet Calendar Server 5.0 では、デフォルトのクライアント UI プロトコルに SHTML が採用されています (新しいプロトコルの詳細は、『iPlanet Calendar Express カスタマイズガイド』を参照)。ただし、以前にカスタマイズした Calendar Express クライアントとの互換性のために WCAP を使用することもできます。WCAP は、引き続き iPlanet Calendar Server 2.0 プロトコルをサポートします。

WCAP はコマンドベースのシステムで、カレンダー操作データを転送するクライアントの要求とサーバの応答で構成されます。WCAP 2.0 では、HTTP 経由でカレンダー操作データが返されます。ほとんどの場合、iPlanet Calendar Server 5.0 は URL 符号化 (URL の中にエンコードされた) 引数を使ってデータを受け取ります。

WCAP 2.0 では、HTTP メッセージの中に出力を含めて返します。返される HTTP メッセージの本文には、次の形式のカレンダーデータが含まれます。

- text/calendar 形式 (iCalendar)
- text/xml 形式

この XML 形式は iCalendar DTD に従います。

- JavaScript オブジェクトが埋め込まれた `text/js`

この形式は、iPlanet Calendar Server 2.0 ユーザインタフェースのデフォルトでした。`.shtml` アプローチの採用のため、iPlanet Calendar Server 5.0 ではこの形式は廃止されています。

WCAP 2.0 コマンドは、次の 4 つのカテゴリに分類されます。

- ユーザ設定情報
- Web カレンダー操作データ
- グループスケジューリングに関するメッセージ送信
- その他のコマンド

このバージョンの新機能

このバージョンには、次の 4 つのコマンドが追加されています。

- `check_id`
管理者はこのコマンドを使用して、セッションがまだ有効かどうかを確認できます。
- `fetchcomponents_by_alarmrange`
指定した期間内にアラームが配信されるコンポーネントを照会します。
- `fetchcomponents_by_attendee_error`
グループスケジューリングメッセージの送信中にエラーが発生したコンポーネントを照会します。
- `get_freebusy`
カレンダーの予定なし / 予定あり時間帯の情報を返します。

既存のコマンドに次の 4 つのパラメータが追加されています。

- `acl`
`set_calprops` に追加されたパラメータ。アクセスコントロールの ACE 文字列を登録します。アクセスコントロールエントリの詳細は、第 10 章「WCAP コマンド」を参照してください。
- `alarmAudio`
`storeevents` および `storetodos` に追加されたパラメータ。オーディオアラームを使用可能にします。

- `alarmFlashing`
storeevents および storetodos に追加されたパラメータ。フラッシュアラームを有効にします。
- `alarmPopup`
storeevents および storetodos に追加されたパラメータ。ポップアップダイアログアラームを有効にします。
- `compstate`
すべての `fetch` コマンドで、`compstate` を使用して、コンポーネントを状態別に取得できます。
- `userid`
`get_userprefs` および `set_userprefs` に追加されたパラメータ。管理者はこのパラメータを使用して、任意のユーザのプリファレンスを検索および設定できます。
- `convertCalid`
`set_userprefs` で、カレンダーの予約購読リスト (`icsSubscribed`)、およびグループの予約購読リスト (`icsSet`) の設定を可能にします。
- `method`
グループスケジューリングのために、storeevents および storetodos コマンドに追加されたパラメータ。ITIP メソッドを指定します。

次のコマンドは、機能が根本的に変更されています。

- `set_calprops` は、iPlanet Calendar Server 2.x では置換モードで機能しましたが、新しいバージョンでは、更新モードで機能します。つまり、コマンドですべてのパラメータを指定する必要はなくなりました。変更対象のパラメータだけを指定します。その他のパラメータは、更新するまで変更されません。

既存のパラメータ `tzid` は、次の既存のコマンドに追加されています。

- `deleteevents_by_id`
- `deletetodos_by_id`
- `fetchcomponents_by_alarmrange`
- `fetchcomponents_by_lastmod`
- `fetchcomponents_by_range`
- `fetchevents_by_id`
- `fetchtodos_by_id`

既存のパラメータ `tzid` は、次の 2 つの既存のコマンドで変更されています。

- storeevents
- storetodos

第 10 章「WCAP コマンド」の 166 ページの「iPlanet Calendar Server 5.0 の新しいコマンドおよびパラメータ」も参照してください。

コマンドの概要

表 9-1 に、WCAP 2.0 でサポートされているコマンドの概要のリストを示します。各コマンドの詳細は、第 10 章「WCAP コマンド」を参照してください。

表 9-1 WCAP コマンドの概要

WCAP コマンド	説明
addlink	あるカレンダーから別のカレンダーへのイベントリンクを追加する
change_password	ユーザのパスワードを変更する
createcalendar	新しいカレンダーを作成する
deletecalendar	既存のカレンダーを削除する
deletecomponents_by_range	指定した期間のカレンダー内のイベントおよび仕事を削除する
deleteevents_by_id	指定した calid および指定した uid と繰り返し ID のペアのイベントを削除する
deleteevents_by_range	指定した期間のカレンダー内のイベントを削除する
deletetodos_by_id	指定した calid および指定した uid と繰り返し ID のペアの仕事を削除する
deletetodos_by_range	指定した期間のカレンダー内の仕事を削除する
export	カレンダーをファイルにエクスポートする
fetchcomponents_by_alarmrange	指定した期間内にアラームが配信されるコンポーネントを照会する
fetchcomponents_by_attendee_error	グループスケジューリングメッセージの送信中にエラーが発生したコンポーネントを照会する
fetchcomponents_by_lastmod	変更されたコンポーネントを照会する (時間範囲を指定)。
fetchcomponents_by_range	指定した期間内のコンポーネントを照会する (属性によりフィルタ)。
fetchevents_by_id	一意の識別子 (uid、繰り返し ID、修飾子) を使用して、1 つまたは複数のイベントを照会する

表 9-1 WCAP コマンドの概要 (続き)

WCAP コマンド	説明
fetchtodos_by_id	一意の識別子 (uid、繰り返し ID、修飾子) を使用して、1 つまたは複数の仕事を照会する
get_all_timezones	サーバがサポートしているすべてのタイムゾーンを返す
get_calprops	カレンダープロパティを返す
get_freebusy	カレンダーの予定なし / 予定あり時間帯の情報を返す
get_guids	ランダムな uid のセットを返す
get_userprefs	ユーザ設定および特定のサーバ設定を返す
import	ファイルからユーザのカレンダーにカレンダーをインポートする
login	ユーザを認証して、最初の HTML ビューにリダイレクトする
logout	現在のユーザのセッションを終了してログイン画面に戻る
ping	管理者専用。カレンダーサーバの ping
search_calprops	指定したパラメータ値でカレンダーを検索する
set_calprops	カレンダープロパティを設定する
set_userprefs	ユーザ設定を設定する
storeevents	アプリケーションまたは URL 符号化方式で指定されたイベントを保存する。このコマンドは、URL 内のプロパティを渡してイベントを保存します
storetodos	アプリケーション / URL 符号化方式で指定されている仕事を保存する
upload_file	ファイルをサーバにアップロードする
version	サーバでサポートされている WCAP バージョンを返す
write_file	ファイルをデータベースに書き込む

セッション ID

多くの WCAP コマンドでは、login コマンドによって返されるセッション ID (id) を指定する必要があります。セッション ID を使うことによって、必要なレベルの権限または所有権を持つ、認証されたユーザだけがデータにアクセスできるようにします。

システムにログインする場合、ユーザは認証に使う識別情報を提示します。デフォルトの認証メカニズムでは、プレーンテキストのパスワードとユーザ名が使用されます。iPlanet Calendar Server は、認証に成功した場合だけセッション ID を生成します。セッション ID は、以後のカレンダー操作で認証の証明の役割を果たします。

認証メカニズムをカスタマイズして、ローカルまたは外部の認証方式を使用することができます。第 3 章「CSAPI リファレンス」の 52 ページの「csIAuthentication」を参照してください。

コマンドの形式

プラグインアーキテクチャにより、iPlanet Calendar Server は複数のコマンド形式をサポートできます。クライアントとサーバの両方でさまざまなデータ形式を使用して、さまざまな ISP のニーズを満たすことができます。

コマンドプロトコルは HTTP を使用し、W3C URL 仕様で定義されている規格に準拠しています。

iPlanet Calendar Server の WCAP は、XML または iCalendar 形式の JavaScript オブジェクトで構成され、クライアントサイドとサーバサイドの両方で HTTP を通じて HTML ドキュメントとして送信されます。クライアントとしては、Internet Explorer 4.0 以上および Navigator 4.05 以上をサポートしています。

クライアントの要求形式

クライアントは、コマンド要求を URI (Universal Resource Identifier) データ形式で、または HTML フォームを使って iPlanet Calendar Server に送信します。

コマンドの形式	説明
URI	標準 URI 構文を使ってクライアントから送信される要求
HTML フォーム - urlencoded	ネイティブ JavaScript オブジェクトを使ってクライアントから送信される要求
HTML フォーム - text/xml	XML 形式にフォーマットした JavaScript オブジェクトを使ってクライアントから送信される要求
HTML フォーム - text/calendar	iCalendar 形式にフォーマットした JavaScript オブジェクトを使ってクライアントから送信される要求

URI 形式

次の形式を使用して、URI 要求を送信します。

```
http://webcalendarserver/COMMAND?PARAM=VAL&PARAM=VAL...
```

複数のアイテムは、セミコロンで区切ります。文字列にセミコロン文字が含まれている場合、セミコロンは符号化文字 %3B に置き換えられます。たとえば、ID のリストの中の文字列「gh;j」を表すには、次の形式を使用します。

```
http://webcalendarserver/fetchcomponents_by_range.wcap?uid=abc;def;gh%3bi;jkl
```

注 これは、iPlanet Calendar Server 2.0 で使用されていた WCAP 1.0 から変更されています。以前のバージョンでは、セミコロンの符号化文字は、=3B でした。

第 10 章「WCAP コマンド」の 161 ページの「符号化文字」も参照してください。

HTML フォーム

method=[GET|POST] および action=command の形式でフォームを送信します (command は、実行するコマンド)。パラメータには、符号化された形式を使用する必要があります。

クライアント側のイベント通知

応答ページの親フレームのクライアント側の JavaScript コードはすべて、CalcommandCallback() というメソッドを実装する必要があります (command は、要求するコマンドの名前)。このコールバックは、HTML 応答の読み込みが完了すると呼び出されます。

このコマンドでデータを取得する場合は、HTTP GET を追加指定します。

このコマンドでデータを変更する (作成と削除を含む) には、HTTP POST を追加指定します。

サーバの応答形式

iPlanet Calendar Server は、クライアントの要求に対して、JavaScript オブジェクトを含む HTML で応答します。サーバ、ユーザ、または個々の要求のそれぞれに対して応答形式を設定することができます。

クライアントは、要求する出力の形式を次の 3 つの中から選択できます。

応答形式	説明
JavaScript オブジェクトを含む HTML - text/js	HTML に JavaScript オブジェクトが含まれた応答。これは、すべての WCAP コマンドに対するデフォルトの出力形式です
JavaScript オブジェクトを含む HTML - text/calendar	HTML に iCalendar 形式の JavaScript オブジェクトが含まれた応答
JavaScript オブジェクトを含む HTML - text/xml	HTML に XML 形式の JavaScript オブジェクトが含まれた応答

WCAP コマンド

この章は、各 WCAP コマンドに共通のトピックと個々の WCAP コマンドに関するトピックに分かれています。

共通のトピック

- アクセスコントロールエントリ (ACE)
- ほかの言語または文字セットの選択
- 繰り返しコンポーネントの削除
- 符号化文字
- エラー処理
- 時刻、文字列、パラメータなどの形式
- 予定なし / 予定ありに限定したアクセス
- iPlanet Calendar Server 5.0 の新しいコマンドおよびパラメータ
- 出力形式
- 繰り返し処理

コマンド

表 10-1 は、WCAP コマンドのアルファベット順のリストです。

表 10-1 WCAP コマンド

addlink	あるカレンダーから別のカレンダーへのイベントリンクを追加する
change_password	ユーザのパスワードを変更する
check_id	セッションが有効かどうかを確認する

表 10-1 WCAP コマンド (続き)

createcalendar	新しいカレンダーを作成する
deletecalendar	既存のカレンダーを削除する
deletecomponents_by_range	指定した期間のカレンダー内のイベントおよび仕事を削除する
deleteevents_by_id	指定した calid および指定した uid と繰り返し ID のペアのイベントを削除する
deleteevents_by_range	指定した期間のカレンダー内のイベントを削除する
deletetodos_by_id	指定した calid および指定した uid と繰り返し ID のペアの仕事削除する
deletetodos_by_range	指定した期間のカレンダー内の仕事を削除する
export	カレンダーをファイルにエクスポートする
fetchcomponents_by_alarmrange	指定した期間内にアラームが配信されるコンポーネントを照会する
fetchcomponents_by_attendee_error	グループスケジューリングメッセージの送信中にエラーが発生したコンポーネントを照会する
fetchcomponents_by_lastmod	変更されたコンポーネントを照会する (時間範囲を指定)。
fetchcomponents_by_range	指定した期間内のコンポーネントを照会する (属性によりフィルタ)。
fetchevents_by_id	一意の識別子 (uid、繰り返し ID、修飾子) を使用して、1 つまたは複数のイベントを照会する
fetchtodos_by_id	一意の識別子 (uid、繰り返し ID、修飾子) を使用して、1 つまたは複数の仕事を照会する
get_all_timezones	サーバがサポートしているすべてのタイムゾーンを返す
get_calprops	カレンダープロパティを返す
get_freebusy	カレンダーの予定なし / 予定あり時間を返す
get_guids	ランダムな uid のセットを返す
get_userprefs	ユーザ設定および特定のサーバ設定を返す
import	ファイルからユーザのカレンダーにカレンダーをインポートする
login	ユーザを認証して、最初の HTML ビューにリダイレクトする
logout	現在のユーザのセッションを終了してログイン画面に戻る
ping	管理者専用。カレンダーサーバの ping

表 10-1 WCAP コマンド (続き)

search_calprops	指定したパラメータ値でカレンダーを検索する
set_calprops	カレンダープロパティを設定する
set_userprefs	ユーザ設定を設定する
storeevents	アプリケーションまたは URL 符号化方式で指定されたイベントを保存する。このコマンドは、URL 内のプロパティを渡してイベントを保存します。
storetodos	アプリケーションまたは URL 符号化方式で指定された仕事を保存する
upload_file	ファイルをサーバにアップロードする
version	サーバでサポートされている WCAP バージョンを返す
write_file	ファイルをデータベースに書き込む

共通のトピック

この節では、1つまたは複数のコマンドに適用される共通のトピックについて説明します。トピックはアルファベット順に並んでいます。

アクセスコントロールエントリ (ACE)

アクセスコントロールエントリ (ACE 文字列) によって、カレンダーのアクセスコントロールが決まります。カレンダーには、複数の ACE 文字列を適用できます。適用されるすべての ACE 文字列は、まとめてアクセスコントロールリスト (ACL) と呼ばれます。システムが ACL リストを調べる際には、アクセスを許可するか拒否するかにかかわらず、最初に見つかった ACE が使用されます。このため、ACL の順序は重要な意味を持っています。ACE 文字列は、適用範囲の狭い ACE から、適用範囲の広い ACE の順序に配置しておく必要があります。

一部のアクセス権は初めから組み込まれています。たとえば、主な所有者は所有するカレンダーのすべての機能にアクセスできます。このため、システムは主な所有者が自身のカレンダーにアクセスする際にアクセスコントロールチェックを行う必要はありません。

`set_calprops` コマンドでは、カレンダーへの ACE 文字列の登録を容易にするために、`acl` パラメータが使用されます。`acl` パラメータは、セミコロンで区切られた ACE 文字列のリストです。デフォルトの `acl` は `ics.conf` ファイルで設定できます。`calstore.calendar.default.acl` の設定を変更するか、`cscal` コマンドラインユーティリティを使用します。構成の設定に関する詳細は、『iPlanet Calendar Server 管理ガイド』を参照してください。

ACE 文字列の例を次に示します。

```
jdoue^c^wd^g
```

この文字列は、3つの「^」文字で区切られた4つの要素で構成されています。4つの要素は、それぞれ次の内容を示します。

1. ACE の最初の要素は、ACE が適用されるユーザを示します。

この要素には、ユーザ ID で指定された個々のユーザ、ドメイン、またはユーザのクラスタイプを指定できます。ユーザのクラスには、次の4つのタイプがあります。

- すべてのユーザ。文字列「@」で表します。
- カレンダーの主な所有者。文字列「@@p」で表します。
- カレンダーの所有者。文字列「@@o」で表します。
- カレンダーの非所有者。文字列「@@n」で表します。

2. ACE の2番目の要素は、ACE が適用される対象を示します。

ACE はカレンダーの次の範囲に適用できます。

- カレンダー全体
コンポーネントとカレンダープロパティの両方に適用します。カレンダー全体を示すには、値 **a** を渡します。
- カレンダーのコンポーネントだけ
カレンダーコンポーネント (イベントまたは仕事) に適用します。コンポーネントだけを示すには、値 **c** を渡します。
- カレンダーのカレンダープロパティだけ
カレンダープロパティ (表示名、所有者リスト) に適用します。カレンダープロパティを示すには、値 **p** を渡します。

3. ACE の 3 番目の要素は、ACE が適用されるアクセス値を示します。

同時に複数の値を指定できます。その際には、チェックしたいビットを示す文字列を呼び出し元から渡す必要があります。

ACE 文字列で使用されるアクセスコントロール文字を表 10-2 に示します。3 番目の要素は、1 つまたは複数のアクセスコントロール文字を含む文字列です。

表 10-2 アクセスコントロール文字

アクセスコントロール文字	説明
c	ユーザに代理キャンセルアクセスを許可する。キャンセルアクセスでは、すでに出席依頼が出されているコンポーネントを、カレンダーの主な所有者に代わってキャンセルする権限を持つ
d	ユーザに削除アクセスを許可する
e	ユーザに代理返信アクセスを許可する。カレンダーの主な所有者に代わって出席依頼を承諾または辞退する権限をユーザに与える
f	ユーザに予定なし / 予定ありに限定したアクセスを許可する
i	ユーザに代理出席依頼アクセスを許可する。カレンダーの主な所有者の代わりに、コンポーネントの作成およびすでに出席依頼が出されているコンポーネントの変更を行う権限を他のユーザに与える
r	ユーザに読み取りアクセスを許可する
s	ユーザにスケジュールアクセスを許可する。要求を送信し、返信が受け付けられ、その他の ITIP スケジューリング対話が有効になる
w	ユーザに書き込みアクセスを許可する。新しいアイテムの追加、アイテムの削除、および既存のアイテムの変更が含まれる

たとえば、読み取りアクセスを許可するには、値 `r` を渡します。書き込みおよび削除アクセスを許可するには、値 `wd` を渡します。

4. ACE の 4 番目の要素は、アクセスの許可または拒否を示します。

ACE では、アクセスを許可または拒否できます。

- アクセスを許可するには、値を `g` に設定します。
- アクセスを拒否するには、値を `d` に設定します。

ACE の要約

ACE の順序についてわかりやすくまとめると、次のようになります。

`who ^ flag ^ how ^ grant`

解説：

- `who` = 文字列、タイプ (`str`)
- `flag` = 文字 `c`、`p`、または `a`
- `how` = 表 10-2 で説明した 1 つまたは複数のアクセスコントロール文字で構成されるアクセス文字列
- `grant` = 文字 `g` または `d`

応用例

`jdoue` のカレンダーに関して、以下に、状況の例と、どのように ACE を `acl` パラメータに設定するかを示します。

- `john` にコンポーネントおよびカレンダープロパティの読み取りアクセスを許可し (`acl=john a r g`)、`susan` にコンポーネントだけの書き込みおよび削除アクセスを許可する場合 (`acl=susan c wd g`)、コマンド全体は次のようになります。

```
set_calprops.wcap?id=${SESSIONID}&calid=jdoue&acl=john^a^r^g;
susan^c^wd^g
```

- ドメインのすべてのユーザに対して、コンポーネントだけのスケジュール、予定なし / 予定あり、および読み取りアクセスを許可し (`@domainname c sfr g`)、所有者に対して、コンポーネントだけの書き込みおよび削除アクセスを許可し (`@@o c wd g`)、所有者に対して、コンポーネントおよびカレンダープロパティの自己管理、スケジュール、予定なし / 予定あり、および読み取りアクセスを許可し (`@@o a zsfr g`)、`susan` に対して、コンポーネントとカレンダープロパティのすべてのアクセスを拒否し (`susan a zsf dwr d`)、すべてのユーザに対して読み取りアクセスを許可 (`@ c r g`) する場合、コマンド全体は次のようになります。

```
set_calprops.wcap?id=${SESSIONID}&calid=jdoue&acl=@domainname^c^sfr^g;@@o^c^wd^g;@@o^a^zsfr^g;susan^a^zsf dwr^d;@^c^r^g
```

注 管理者は、管理者としてログインしていて、かつ、ics.conf ファイルのサーバ構成の設定 `service.admin.calmaster.overrides.accesscontrol` が「yes」に設定されている場合、すべての WCAP コマンドのアクセスコントロールを無視できます。

ほかの言語または文字セットの選択

システムデフォルト以外の言語でデータを取得する要求を出すには、lang または charset パラメータを設定します。言語のシステムデフォルトは、ics.conf ファイルで設定されたサーバ設定です。詳細は、『管理者ガイド』を参照してください。login コマンドでは、lang パラメータだけを使用します。

set_calprops コマンドでは、ほとんどの場合、lang パラメータだけの指定で十分です。ただし、場合によっては、lang パラメータの代わりに charset パラメータを使用する必要があります。たとえば、要求したデータを特定の文字セットで取得するには、charset を使用してその文字セットを指定する必要があります。charset 値には、iso-8859-1などを指定できます。形式仕様に関する詳細は、この章の 165 ページの「時刻、文字列、パラメータなどの形式」に記載されている RFC を参照してください。

iCalendar または XML 形式のデータを要求した場合は、RFC 仕様に従って常に UTF-8 形式でデータが返されます。charset を設定しても UTF-8 形式で返されます。

有効な lang 値を以下に示します。

de	ドイツ語
en	英語 (デフォルト)
es	スペイン語
fr	フランス語
it	イタリア語
ja	日本語
ko	韓国語
ru	ロシア語
sv	スウェーデン語
zh_CN	中国語 / 簡体字中国語
zh_TW	中国語 (台湾)

注 これは、これらの言語のすべてがサーバで現在サポートされていることを意味するものではありません。サーバで現在サポートされている言語については、iPlanet の購入先に問い合わせてください。

繰り返しコンポーネントの削除

コンポーネントを削除するときに、コンポーネントが繰り返しているかどうかを指定できます。場合によっては、必ず指定しなければならないこともあります。また、繰り返している場合は、その繰り返しているコンポーネントだけでなく元のイベントまたは仕事も削除するかどうかを指定できます。

mod パラメータを使用して、以下の削除オプションを選択します。

表 10-3 mod パラメータの削除オプション

値	オプション
1	このインスタンスだけを削除する
2	このインスタンスおよび以後の繰り返しをすべて削除する
3	このインスタンスおよび以前の繰り返しをすべて削除する
4	すべてのインスタンスを削除する

各オプションについて、rid パラメータを渡してイベントの繰り返し ID を指定し、さらに削除のタイプを渡す必要があります。

イベントの単一インスタンスだけを削除するには、mod パラメータを 1 に設定します。たとえば、次の URL は、2000 年 3 月 1 日の午前 11 時 22 分 33 秒 (世界標準時) に発生するイベントだけを削除します。

```
http://webcalendarserver/deleteevents_by_id.wcap?id=23423423434abc&calid=jdoe&uid=001&rid=20000301T112233Z&mod=1
```

このイベントとそれ以降のインスタンスをすべて削除するには、mod パラメータを 2 に設定します。たとえば、次の URL は、2000 年 3 月 1 日の午前 11 時 22 分 33 秒 (世界標準時) に発生するイベントとそれ以降のインスタンスをすべて削除します (uid 001)。

```
http://jdoe/deleteevents_by_id.wcap?id=23423423434abc&calid=jdoe&uid=001&rid=20000301T112233Z&mod=2
```


このイベントとそれ以前のインスタンスをすべて削除するには、`mod` パラメータを 3 に設定します。

たとえば、次の URL は、2000 年 3 月 1 日の午前 11 時 22 分 33 秒 (世界標準時) に発生するイベントとそれ以前のインスタンスをすべて削除します (`uid 001`)。

```
http://jdoe/deleteevents_by_id.wcap?id=23423423434abc&calid=jdoe&uid=001&rid=20000301T112233Z&mod=3
```

このイベントのインスタンスをすべて削除するには、`mod` パラメータを 4 に設定します。たとえば、次の URL は、このイベントのインスタンスをすべて削除します (`uid 001`)。

```
http://jdoe/deleteevents_by_id.wcap?id=23423423434abc&calid=jdoe&uid=001&rid=20000301T112233Z&mod=4
```

符号化文字

前述の例では、`cal` に関するパラメータの符号化したリストに、いくつかの符号化された文字が含まれています。符号化文字の例をいくつか挙げます。

`%3D` = `'=`

`%26` = `'&'`

`%22` = `''''`

`%XX` は、その文字の ASCII 値を 16 進で表しています。たとえば、「&」文字は 16 進で 26 です (10 進では 38)。

エラー処理

コンポーネントデータを返す WCAP コマンド (`fetch`、`delete`、および `store` コマンド) を呼び出すと、配列およびエラー文字列も返されます。

エラー文字列

エラー文字列 `errno` は、トランザクションに対してゼロ以外のエラー番号を返します。コマンドが正常に実行された場合、その値は 0 になります。

レイヤーエラー番号配列

トランザクションのエラーの他に、トランザクション中に1つまたは複数のレイヤーでエラーが発生する可能性があります。レイヤーとは、要求されている複数のアクションまたはアイテムの1つです。たとえば、`fetch_components_by_range` に取り込み元のカレンダー ID のリストを渡すことがあります。この場合、各カレンダーがレイヤーです。

エラー番号は、レイヤーに固有です。レイヤー配列のインデックスは、渡されたレイヤーのインデックス番号に対応します。つまり、`layer_errno[3]` は、取り込みコマンドに渡した3番目のアイテムを指します。

1つのレイヤーのエラーが、ほかのレイヤーの処理に影響することはありません。

次の3つのレイヤーエラー配列があります。

- `layer_errno`。 `fetch` コマンド用
- `delete_layer_errno`。 `delete` コマンド用
- `store_layer_errno`。 `store` コマンド用

レイヤーカウント配列

さらに、`deleteevents_by_id` および `deletetodos_by_id` の場合は、2つ目の配列 `delete_event_count` または `delete_todo_count` がそれぞれ返されます。この配列には、エラーが発生する前に正常に削除された回数が格納されます。この配列は、渡されたコンポーネントに繰り返しがある場合に有用です。レイヤーエラー番号配列と同様に、このレイヤーのインデックスは、コマンドに渡されたレイヤーのインデックス番号に対応します。

たとえば、3つのコンポーネントを `deleteevents_by_id` に渡したときに、2つ目のイベントでエラーが発生した場合は、`delete_layer_errno[2]` にエラーコードが格納されます。次に、`delete_event_count[2]` を参照すると、エラーが発生する前に正常に削除されたイベントの数を確認できます。

エラーコード

表 10-4 は、エラー番号配列に返されるエラーコードの一部です。

表 10-4 エラーの名前、値、および意味

エラー名	値	意味
LOGOUT	-1	ログアウトは正常に終了しました。
OK	0	コマンドは正常に終了しました
LOGIN_FAILED	1	ログインに失敗しました。セッション ID がタイムアウトしています。無効なセッション ID です

表 10-4 エラーの名前、値、および意味 (続き)

エラー名	値	意味
LOGIN_OK_DEFAULT_CALENDAR_NOT_FOUND	2	login.wcap は正常に終了しましたが、このユーザのデフォルトカレンダーが見つかりませんでした。このユーザ ID に設定されている新しいデフォルトカレンダーが作成されました
DELETE_EVENTS_BY_ID_FAILED	6	コマンドの実行に失敗しました
SETCALPROPS_FAILED	8	コマンドの実行に失敗しました
FETCH_EVENTS_BY_ID_FAILED	9	コマンドの実行に失敗しました
CREATECALENDAR_FAILED	10	コマンドの実行に失敗しました
DELETECALENDAR_FAILED	11	コマンドの実行に失敗しました
ADDLINK_FAILED	12	コマンドの実行に失敗しました
FETCHBYDATERANGE_FAILED	13	コマンドの実行に失敗しました
STOREEVENTS_FAILED	14	コマンドの実行に失敗しました
STORETODOS_FAILED	15	コマンドの実行に失敗しました
DELETE_TODOS_BY_ID_FAILED	16	コマンドの実行に失敗しました
FETCH_TODOS_BY_ID_FAILED	17	コマンドの実行に失敗しました
FETCHCOMPONENTS_FAILED_BAD_TZID	18	正しい TZID が見つかりませんでした。 fetchcomponents_by_range、 fetchevents_by_id、および fetchtodos_by_id に適用されます
SEARCH_CALPROPS_FAILED	19	コマンドの実行に失敗しました
GET_CALPROPS_FAILED	20	コマンドの実行に失敗しました
DELETECOMPONENTS_BY_RANGE_FAILED	21	コマンドの実行に失敗しました
DELETEEVENTS_BY_RANGE_FAILED	22	コマンドの実行に失敗しました
DELETETODOS_BY_RANGE_FAILED	23	コマンドの実行に失敗しました
GET_ALL_TIMEZONES_FAILED	24	コマンドの実行に失敗しました
CREATECALENDAR_ALREADY_EXISTS_FAILED	25	createcalendar.wcap の実行に失敗しました。指定した名前のカレンダーがデータベースにすでに存在します
SET_USERPREFS_FAILED	26	コマンドの実行に失敗しました
CHANGE_PASSWORD_FAILED	27	コマンドの実行に失敗しました
ACCESS_DENIED_TO_CALENDAR	28	コマンドの実行に失敗しました。このユーザはカレンダーへのアクセスを拒否されています

表 10-4 エラーの名前、値、および意味 (続き)

エラー名	値	意味
CALENDAR_DOES_NOT_EXIST	29	コマンドの実行に失敗しました。カレンダーがデータベースに存在しません
ILLEGAL_CALID_NAME	30	createcalendar.wcap の実行に失敗しました。渡された calid は無効です
CANNOT_MODIFY_LINKED_EVENTS	31	storeevents.wcap の実行に失敗しました。変更しようとしているイベントはリンクされているイベントです
CANNOT_MODIFY_LINKED_TODOS	32	storetodos.wcap の実行に失敗しました。変更しようとしている仕事はリンクされている仕事です
CANNOT_SENT_EMAIL	33	コマンドの実行に失敗しました。電子メールの通知に失敗しました。多くの場合、サーバが電子メールを送信できるように正しく設定されていないことが原因です。このエラーは、storeevents、storetodos、deleteevents_by_id、または deletetodos_by_id で発生する可能性があります
CALENDAR_DISABLED	34	コマンドの実行に失敗しました。カレンダーがデータベースで無効状態になっています
WRITE_IMPORT_FAILED	35	ファイルをサーバに書き込むときに、インポートに失敗しました
FETCH_BY_LAST_MODIFIED_FAILED	36	コマンドの実行に失敗しました
CAPI_NOT_SUPPORTED	37	CS&T カレンダーデータからの読み取りに失敗しました
CALID_NOT_SPECIFIED	38	カレンダー ID が指定されていません
GET_FREEBUSY_FAILED	39	コマンドの実行に失敗しました
STORE_FAILED_DOUBLE_BOOKED	40	storeevents または storetodos の実行に失敗しました。イベントまたは仕事を登録しようとしたが、時間枠に空きがありません。二重予約は許可されていません
FETCH_BY_ALARM_RANGE_FAILED	41	コマンドの実行に失敗しました
FETCH_BY_ATTENDEE_ERROR_FAILED	42	コマンドの実行に失敗しました

表 10-4 エラーの名前、値、および意味 (続き)

エラー名	値	意味
ATTENDEE_GROUP_EXPANSION_CLIPPED	43	拡張しようとしている LDAP グループが大きすぎます。拡張可能な最大数を超過しています。指定された最大数で拡張は停止されます。デフォルトの最大数は 200 です。最大数を変更するには、サーバの <code>calstore.group.attendee.maxsize</code> を設定します
USERPREFS_ACCESS_DENIED	44	この管理者レベルのユーザ設定の取得または変更がサーバで許可されていないか、アクセスの要求元が管理者ではありません
NOT_ALLOWED_TO_REQUEST_PUBLISH	45	要求元がイベントの開催者でないため、PUBLISH または REQUEST メソッドを使ってコンポーネントを編集することはできません

時刻、文字列、パラメータなどの形式

すべての時刻、文字列、パラメータなどの正確な形式および定義については、RFC2445、RFC2446、および RFC2447 を参照してください。特に指定がない限り、すべての WCAP コマンドはこれらの仕様に従います。

RFC は次の IETF Web サイトで参照できます。

- <http://www.ietf.org/rfc/rfc2445.txt>
- <http://www.ietf.org/rfc/rfc2446.txt>
- <http://www.ietf.org/rfc/rfc2447.txt>

注 WCAP の文字列パラメータのサイズに制限がない場合は、パラメータの長さを 1024 文字以下にすることをお勧めします。

予定なし / 予定ありに限定したアクセス

予定なし / 予定ありに限定したアクセスでは、カレンダーのスケジュール時刻の表示はユーザに許可されますが、イベントの詳細の表示は許可されません。代わりに、時間ブロックごとに「予定あり」という文字が表示されます。予定されたイベントのない時間ブロックの横には、「予定なし」という文字が表示されます。

たとえば、jdoe というカレンダーに次のイベントがあります。

10:00 ~ 11:00 最初の会議
12:00 ~ 1:00 昼食
3:00 ~ 4:00 次の会議

ユーザ john には、カレンダー jdoe への予定なし / 予定ありアクセス権が与えられていると想定します。jdoe の予定なし / 予定あり時間 (9:00 ~ 6:00) は、次のようになります。

9 ~ 10 : 空き時間
10 ~ 11 : 予定あり
11 ~ 12 : 空き時間
12 ~ 1 : 予定あり
1 ~ 3 : 空き時間
3 ~ 4 : 予定あり
4 ~ 6 : 空き時間

john は、予定の内容はわかりませんが、予定があることだけはわかります。

iPlanet Calendar Server 5.0 の新しいコマンドおよびパラメータ

新しい WCAP コマンドでは、2.x との下位互換性を保持しながら、多くの変更が加えられています。重要な変更点の 1 つは、set_calprops コマンドが置換モードではなく更新モードで機能することです。つまり、2.x のように、コマンドですべてのパラメータを指定する必要がなくなりました。set_calprops コマンドで指定されていないパラメータは、変更されません。

このバージョンには、いくつかの新しいコマンドと、既存のコマンド用の新しいパラメータが追加されています。

新しいコマンド

新しいコマンドは、次のとおりです。

- `check_id`
管理者はこのコマンドを使用して、セッションがまだ有効かどうかを確認できます。
- `fetchcomponents_by_alarmrange`
指定した期間内にアラームが配信されるコンポーネントを照会します。
- `fetchcomponents_by_attendee_error`
グループスケジューリングメッセージの送信中にエラーが発生したコンポーネントを照会します。
- `get_freebusy`
カレンダーの予定なし / 予定あり時間帯の情報を返します。

新しいパラメータ

新しいパラメータは、次のとおりです。

- `acl`
`set_calprops` に追加されたパラメータ。アクセスコントロールの ACE 文字列を登録します。156 ページの「アクセスコントロールエントリ (ACE)」を参照してください。
- `alarmAudio`
`storeevents` および `storetodos` に追加されたパラメータ。オーディオアラームを使用可能にします。
- `alarmFlashing`
`storeevents` および `storetodos` に追加されたパラメータ。フラッシュアラームを有効にします。
- `alarmPopup`
`storeevents` および `storetodos` に追加されたパラメータ。ポップアップダイアログアラームを有効にします。
- `compstate`
`fetchcomponents_by_attendee_error` 以外のすべての `fetch` コマンドでは、パラメータ `compstate` を使用して、状態別にコンポーネントを取り込むことができます。これらのコマンドでは通常、すべてのコンポーネントが取り込まれるため、このパラメータを使って取り込むコンポーネントのタイプを制限します。
パラメータが指定されていない場合、デフォルト値は ALL です。

表 10-5 は、コンポーネントの状態値の一覧です。コンポーネントの状態は、参加者と開催者のどちらかに属しています。

表 10-5 compstate パラメータに関するコンポーネントの状態値

値	開催者 / 参加者	コメント
REPLY-DECLINED	参加者	参加者が会議を辞退しました
REPLY-ACCEPTED	参加者	参加者が会議を承諾しました
REQUEST-COMPLETED	開催者	開催者から会議要求が送信されました。すべての参加者の返信が受信されました
REQUEST_NEEDS-ACTION	参加者	参加者は会議に関してまだ返信していません
REQUEST-NEEDSNOACTION	参加者	開催者は参加者からの返信を必要としていません
REQUEST-PENDING	開催者	開催者の会議要求が送信されました。グループスケジューリングエンジンは、すべての参加者の処理をまだ終了していません
REQUEST-WAITFORREPLY	開催者	開催者から会議要求が送信されました。参加者からの返信を待っています
ALL	該当なし	(デフォルト) すべてのイベントおよび仕事コンポーネントの状態を表しています

- `convertCalid`

`set_userprefs` で、カレンダーの予約購読リスト (`icsSubscribed`) およびグループの予約購読リスト (`icsSet`) を設定できるようにします。

コロン「:」の使用を禁止するサーバ制限の回避策として、`set_userprefs` に追加されました。

`icsSet` または `icsSubscribed` を設定する場合、WCAP ではこのパラメータの値だけが検索されます。

- `language`

`storeevents` および `storetodos` に追加されたパラメータ。コンポーネントの言語を指示します。

- `method`

グループスケジューリングにおいて、`storeevents` コマンドおよび `storetodos` コマンドで使用する ITIP メソッドを指定します。

- `orgUID`
`storeevents` および `storetodos` に追加されたパラメータ。
 イベントの開催者または仕事の依頼者のユーザ ID を保持します。
- `userid`
 管理者がユーザ設定を取得または設定できるように `get_userprefs` および `set_userprefs` に追加されたパラメータ。

このバージョンでは、以前は別のコマンドだけで使われていたパラメータが、次のコマンドに追加されています。

- `fmt-out` が `set_calprops` に追加されています。
- `orgEmail` が `storetodos` に追加されています。
- `tzid` が次のコマンドに追加されています。
 - `deleteevents_by_id`
 - `deletetodos_by_id`
 - `fetchcomponents_by_alarmrange`
 - `fetchcomponents_by_lastmod`
 - `fetchcomponents_by_range`
 - `fetchevents_by_id`
 - `fetchtodos_by_id`

さらに、`storeevents` および `storetodos` の `tzid` が変更されています。このパラメータでは、前述のコマンドと同じタイプ、つまり、「America/Los_Angeles」などのタイムゾーン ID 文字列を使用できます。

出力形式

管理者は、次の3つのコンテンツタイプの出力形式を要求できます。

1. `text/calendar -iCalendar`
2. `text/xml -iCalendar XML`
3. `text/js -JavaScript 出力`

出力形式を変更するには、`fmt-out` に目的の値を設定します。`fmt-out` が指定されていない場合は、デフォルト形式の `text/js` で返されます。

簡略出力

brief パラメータを使用すると、簡略化されたイベントおよび仕事データを JavaScript で出力することができます。返される出力は、通常の出力の約半分のサイズになり、次のパラメータが含まれます。

表 10-6 brief 出力に含まれる出力パラメータ

イベントに含まれるパラメータ	仕事に含まれるパラメータ
calid	calid
created	completed
desc	created
dtstart	desc
dtend	dtstart
isAllDay	due
lastMod	isAllDay
linkCalid	lastMod
location	linkCalid
rid	location
summary	percent
tzid	rid
uid	summary
	tzid
	uid

繰り返し処理

繰り返しの指定には、次の 6 つのパラメータを使用できます。

1. rrules。イベントの繰り返しに関する繰り返し規則文字列を二重引用符で囲み、各文字列をセミコロンで区切ったリスト
2. rdates。繰り返しの日付の ISO8601 日付文字列をセミコロンで区切ったリスト
3. exrules。除外する日付に関する繰り返し規則文字列を二重引用符で囲み、各文字列をセミコロンで区切ったリスト
4. exdates。除外する日付の ISO8601 日付文字列をセミコロンで区切ったリスト
5. rid。イベントの繰り返し ID を示す ISO8601 日付文字列

6. `mod`。1～4の数値。イベントのどのインスタンスを登録するかを指定する修飾子
7. `rchange`。storecomponents の繰り返しを継続するかどうかを指定するブール値

rrules

`rrules` パラメータは、繰り返し規則文字列を二重引用符で囲み、各文字列をセミコロンの区切ったリストの形式をとります。各文字列は、イベントの繰り返し規則を表します。各文字列は、二重引用符で囲む必要があります。繰り返し規則には、さまざまなパラメータを使用できます。構文の詳細は、RFC2445 を参照してください。

繰り返しを指定するときは、`freq` パラメータと `count` パラメータを使うと便利です。

- 規則内の `freq` パラメータには、イベントの周期性を定義します。次の値を指定できません。

DAILY	イベントを毎日繰り返します。
WEEKLY	イベントを毎週繰り返します。
MONTHLY	イベントを毎月繰り返します。
YEARLY	イベントを毎年繰り返します。

- 規則内の `count` パラメータには、会議を繰り返す回数を定義します。`count` を指定しなかった場合は、デフォルトの最大許容繰り返し数で使用されます。デフォルトの最大数は 60 です。最大数を変更するには、サーバ設定の `calstore.recurrence.bound` を設定します。

次の例では、`rrules` パラメータを使って 2 つの繰り返し規則を示しています。

```
rrules="count%3D10%3Bfreq%3Ddaily";"freq%3Dweekly%3Bcount%3D4"
(COUNT=10;FREQ=DAILY and FREQ=WEEKLY;COUNT=4 encoded)
```

最初の規則では、イベントが毎日発生し、10 回繰り返されることを指定しています。2 番目の規則では、イベントが毎週発生し、4 回繰り返されることを指定しています。

次の URL は、`rrules` パラメータを渡す例です。

```
http://webcalendarserver/storecomponents.wcap?id=b5q2o8ve2rk02nv9t6
&calid=jdoe&uid=333&dtstart=20000301T112233Z
&rrules="count%3D10%3Bfreq%3Ddaily";"freq%3Dweekly%3Bcount%3D4"
&dtend=20000301T112233&summary=uuuu
```

rdates

`rdates` パラメータは、日付と時刻の指定をセミコロンの区切ったリストの形式をとります。それぞれの日付と時刻はイベントの繰り返し日付を指定しています。

たとえば、次の `rdates` パラメータでは、2つの繰り返し日付 (2000年3月31日11時22分33秒と2000年5月31日11時22分33秒) にイベントを繰り返すことを示しています。

```
rdates=20000331T112233;20000531T112233
```

次の URL は `rdates` パラメータを渡す例です。

```
http://webcalendarserver/storecomponents.wcap?id=b5q2o8ve2rk02nv9t6
&calid=jdoe&uid=333&dtstart=20000301T112233Z
&rdates=20000331T112233;20000531T112233
&dtend=20000301T112233&summary=uuuu
```

特定の日付のあとの繰り返し規則を変更する場合は、`rchange` を 1 に設定する必要があります。

exrules

`exrules` パラメータは、繰り返し規則文字列を二重引用符で囲み、各文字列をセミコロンの区切ったリストをとります。各規則は除外するイベントの繰り返し規則です。

たとえば、次の `exrules` パラメータでは、2つの規則で指定した時期に繰り返しイベントを繰り返さないことを指定します。

```
exrules="count%3D10%3Bfreq%3Ddaily";"freq%3Dweekly%3Bcount%3D4"
(COUNT=10;FREQ=DAILY and FREQ=WEEKLY;COUNT=4 encoded)
```

最初の規則では 10 回に渡って、イベントを毎日繰り返さないことを指定します。2 番目の規則では 4 回に渡って、イベントを毎週繰り返さないことを指定します。

次の URL は `exrules` パラメータを渡す例です。

```
http://webcalendarserver/storecomponents.wcap?id=b5q2o8ve2rk02nv9t6
&calid=jdoe&uid=333&dtstart=20000301T112233Z
&exrules="count%3D10%3Bfreq%3Ddaily";"freq%3Dweekly%3Bcount%3D4"
&rrules="count%3D10%3Bfreq%3Ddaily"&dtend=20000301T112233&summary=
uuuu
```

exdates

`exdates` パラメータは、日付と時刻の指定をセミコロンの区切ったリストの形式をとります。それぞれの日付と時刻は、除外するイベントの日時を表します。

たとえば、次の `exdates` パラメータでは、繰り返しイベントが 2つの指定された日付 (2000年3月31日11時22分33秒と2000年5月31日11時22分33秒) に発生しないことを示しています。

```
exdates=20000331T112233;20000531T112233
```

次の URL は `exdates` パラメータを渡す例です。

```
http://webcalendarserver/storecomponents.wcap?id=b5q2o8ve2rk02nv9t6
&calid=jdoe&uid=333&dtstart=20000301T112233Z
&exdates=20000331T112233;20000531T112233
&rrules="COUNT%3D200%3BFREQ=DAILY";dtend=20000301T112233&summary=uu
uu
```

rid

このパラメータは、イベントまたは仕事の繰り返し日付を一意に指定します。 `rid` と `mod` パラメータを組み合わせて、イベントおよび仕事の変更範囲を指定します。

例:

```
http://webcalendarserver/storecomponents.wcap?id=b5q2o8ve2rk02nv9t6
&calid=jdoe&uid=333&dtstart=20000301T112233Z
&rid=20000331T112233;dtend=20000301T112233&summary=uuuu&mod=1
```

mod

このパラメータでは、変更を適用するイベントまたは仕事のインスタンスを指定します。 `mod` 値によって、次のようになります。

値	オプション
1	このインスタンスのみ
2	このインスタンスおよびそれ以降のすべてのインスタンス
3	このインスタンスおよびそれ以前のすべてのインスタンス
4	このインスタンスを含むすべてのインスタンス

繰り返しのないイベントまたは仕事の場合、 `rid` は 0 です。

rchange

`rchange` パラメータは、 `storecomponents` で繰り返しを継続するかどうかを指定します。通常、イベントおよび仕事カレンダーコンポーネントは継続されるため、デフォルトは 1 になっています。

しかし、複数のイベントを修正する際には、繰り返しを継続したくない場合があります。たとえば、会議が 2000 年 1 月 1 日から毎週金曜日に繰り返されていたとします。2000 年 2 月 1 日以後の各イベントの `summary` パラメータを `changed-event` に変更するには、次の URL を使用します。この例では、`rchange` パラメータを 0 に設定しています。こうすることで、イベントを追加せずに変更を行うことができます。

```
http://webcalendarserver/storeevents.wcap?id=b5q2o8ve2rk02nv9t6
&calid=jdoe&uid=abcxyz&dtstart=20000201T112233Z
&rrules="byday%3Dfr%3Bfreq%3Dweekly"&summary=changed-event
&rid=20000201T112233Z&mod=2&rchange=0
```

コマンド

`id` パラメータ (セッション ID) を持つコマンドでは、このパラメータを必ず指定する必要があります。ただし、この規則には 2 つの例外があります。カレンダーへの匿名 (`anonymous`) アクセスを許可している場合、または公開カレンダーへの読み取りアクセスを許可している場合、このパラメータは指定する必要がありません。その他の場合は、セッション ID を `id` パラメータに指定する必要があります。

注 サーバは、特別で重要な名前として「`anonymous`」をサポートしています。`anonymous` ユーザは、任意のパスワードでログインできます。また、特定のドメインに関連付けられていません。

WCAP コマンドには、次の 33 のコマンドがあります。

- `addlink`
- `change_password`
- `check_id`
- `createcalendar`
- `deletecalendar`
- `deletecomponents_by_range`
- `deleteevents_by_id`
- `deleteevents_by_range`
- `deletetodos_by_id`
- `deletetodos_by_range`
- `export`
- `fetchcomponents_by_alarmrange`
- `fetchcomponents_by_attendee_error`

- `fetchcomponents_by_lastmod`
- `fetchcomponents_by_range`
- `fetchevents_by_id`
- `fetchtodos_by_id`
- `get_all_timezones`
- `get_calprops`
- `get_freebusy`
- `get_guids`
- `get_userprefs`
- `import`
- `login`
- `logout`
- `ping`
- `search_calprops`
- `set_calprops`
- `set_userprefs`
- `storeevents`
- `storetodos`
- `upload_file`
- `version`
- `write_file`

addlink

目的

特定のカレンダーのイベントまたは仕事から別のカレンダーへのリンクを追加します。

パラメータ

表 10-7 は、このコマンドの 5 つのパラメータの一覧です。

表 10-7 `addlink` のパラメータ

パラメータ	タイプ	目的	必須/任意	デフォルト
<code>id</code>	一意の識別子文字列	セッション ID	必須	該当なし

表 10-7 addlink のパラメータ (続き)

パラメータ	タイプ	目的	必須/任意	デフォルト
srcCal	文字列	リンク対象のイベントまたは仕事を含むカレンダー。このカレンダーへの読み取りアクセス権が必要	必須	該当なし
rid	文字列をセミコロンの区切ったリスト	リンクを張る先のイベント / 仕事の繰り返し ID のリスト	必須	該当なし
destCal	文字列	リンクを作成するカレンダー。このカレンダーへの書き込みアクセス権が必要	必須	該当なし
uid	文字列をセミコロンの区切ったリスト	リンクを張る先のイベント / 仕事の識別子のリスト	必須	該当なし

説明

このコマンドを使用して、srcCal カレンダー中の指定したイベント / 仕事へのリンクを、destCal カレンダーの中に追加します。

指定するカレンダーが公開カレンダーである場合を除き、コマンドと一緒に id パラメータを指定する必要があります。

uid リストと rid リストのアイテム数は、正確に一致していなければなりません。さらに、各 rid とそれに対応する uid は、リスト内で同じ位置になければなりません。イベントまたは仕事に繰り返し ID がない場合は、rid リストで 0 を使用します。

次の 3 つの要件を満たしていない場合は、トランザクションが失敗します。

- srcCal カレンダーと destCal カレンダーのパラメータが有効なカレンダー ID であること
- destCal カレンダーへの書き込みアクセス権と srcCal カレンダーの読み取りアクセス権を持っていること
- uid リストと rid リストの要素数が同じであること

戻り値

トランザクションが失敗した場合は、ADDLINK_FAILED (12) のエラーが返されます。成功した場合のリターンコードは 0 です。

例

この例では、2 つのイベントリンクをカレンダー jdoe に追加する方法を示しています。このリンクは、pub カレンダー内のイベントを指すこととなります。イベントは、1111 と 2222 です (イベントの uid)。どちらのイベントも繰り返さないため、繰り返し ID は 0 です。

次の URL でこのトランザクションが実行されます。

```
http://webcalendarserver/addlink.wcap?id=b5q2o8ve2rk02nv9t6&destCal=jdoe&srcCal=pub&uid=1111;2222&rid=0;0
```

change_password

目的

現在のユーザのパスワードを変更します。このコマンドは、5.0 では使わないでください。2.x との下位互換性の保持するために残されています。パスワードの変更に関する詳細は、『管理者ガイド』を参照してください。

パラメータ

表 10-8 は、このコマンドの 4 つのパラメータの一覧です。

表 10-8 change_password のパラメータ

パラメータ	タイプ	目的	必須 / 任意	デフォルト
id	文字列	セッション ID	必須	該当なし
newPassword	文字列	新しいユーザパスワード	必須	該当なし
oldPassword	文字列	以前のユーザパスワード	必須	該当なし
fmt-out	文字列	返されるデータの形式 次の形式を指定できます text/calendar text/xml text/js	任意	text/js

目的

このコマンドは、ユーザのパスワードを変更します。パスワードはプレーンテキストで渡されます。service.wcap.allowchangepassword が設定されている場合を除き、このコマンドを使用できるのは管理権限を持つユーザだけです。

指定するカレンダーが公開カレンダーである場合を除き、コマンドと一緒に id パラメータを指定する必要があります。

戻り値

コマンドの実行に失敗すると、エラー CHANGE_PASSWORD_FAILED (27) が返されます。

例

次の URL は、パスワードの変更を要求する例です。

```
http://webcalendarserver/change_password.wcap?id=b5q2o8ve2rk02nv9t6
&oldPassword=abc&newPassword=def
```

check_id

目的

管理者がセッションの有効性を確認するときに使う管理者用コマンドです。

パラメータ

表 10-8 は、このコマンドの 4 つのパラメータの一覧です。

表 10-9 change_password のパラメータ

パラメータ	タイプ	目的	必須 / 任意	デフォルト
id	文字列	セッション ID	必須	該当なし
fmt-out	文字列	返されるデータの形式 次の形式を指定できます text/calendar text/xml text/js	任意	text/js

目的

管理者はこのコマンドを使って、セッションの有効性を確認できます。

戻り値

サーバは、プロパティ X-NSCP-WCAP-CHECK-ID を返します。このプロパティの値が 1 の場合、セッションは有効です。ゼロ (0) が返された場合、セッションは無効です。原因は、セッションがタイムアウトしていたか、そのセッション ID を認識できなかったかのどちらかです。

例

次のコマンドは、指定されたセッションが有効かどうかを返します。

```
http://webcalendarserver/check_id.wcap?id=n3l0eeu6s3n3o3b8v&fmt-out
=text/calendar
```

次の出力が返されます。

```
HTTP/1.1 200
Date:Thu, 14 Dec 2000 19:48:17 GMT
Content-type:text/calendar; charset=UTF-8
Content-length: 131
```

```

Last-modified:Thu, 14 Dec 2000 19:48:17 GMT
Pragma:no-cache
Expires: 0
Cache-Control:no-cache
Connection:Keep-Alive

BEGIN:VCALENDAR
PRODID:-//iPlanet/Calendar Hosting Server//EN
METHOD:PUBLISH
VERSION:2.0
X-NSCP-WCAP-CHECK-ID:1
END:VCALENDAR lendar
    
```

createcalendar

目的

新しいカレンダーを作成します。

パラメータ

表 10-10 は、このコマンドの 4 つのパラメータの一覧です。

表 10-10 createcalendars のパラメータ

パラメータ	タイプ	目的	必須/任意	デフォルト
calid	文字列	ユーザの calid。この文字列を使って calid の文字列を作成する	必須	該当なし
id	一意の識別子文字列	セッション ID	必須	該当なし
fmt-out	文字列	返されるデータの形式 次の形式を指定できる text/calendar text/xml text/js	必須	text/js
set_calprops	整数 (0、1)	新しいカレンダープロパティを設定するかどうかを示すブール値 1 = プロパティを設定する 0 = プロパティを設定しない	任意	0

説明

このコマンドを使用して、現在のユーザに新しいカレンダーを作成します。管理権限のないユーザがこのコマンドを使用できるようにするには、ics.conf ファイルの service.wcap.allowcreatecalendars を設定します。

有効な calid の作成

作成したカレンダーの新しい calid は、ユーザの userid と渡された calid パラメータの組み合わせです。userid を取得するために、id パラメータに指定されたセッションが検索されます。新しいカレンダーの calid の形式は、userid:calid です。たとえば、ユーザが jdoe、calid パラメータが tv の場合、新しいカレンダーの calid は jdoe:tv です。

calid パラメータが長すぎる場合、または不正な文字を含んでいる場合は、サーバによって切り捨てられます。calid パラメータの切り捨てに失敗した場合は、エラー ILLEGAL_CALID_NAME (30) が返されます。

calid パラメータに有効な文字は次のとおりです。

- アルファベット文字 (A ~ Z, a ~ z)
- 数字 (0 ~ 9)
- 特殊文字
 - ダッシュ (-)
 - アンダースコア (_)
 - ピリオド (.)

たとえば、calendar1、calendar-1、calendar_1、calendar.1 は、calid パラメータの正しい値です。

カレンダープロパティの設定

バージョン 5.0 では、カレンダーの作成中にカレンダープロパティを設定できます。設定するには、set_calprops パラメータを 1 の値で渡します。そして、set_calprops コマンドで定義されている追加パラメータを渡すことによってカレンダープロパティを設定できます。

設定可能なカレンダープロパティの詳細は、set_calprops を参照してください。

戻り値

fetchcomponents_by_range コマンドを呼び出して取得されたカレンダープロパティが、fmt-out 値に基づいた形式で返されます。

エラーコード

処理に成功した場合は、エラー番号 0 がエラー文字列に追加されます。新しく作成された `calid` がデータベースにすでに存在する場合は、エラーコード `CREATECALENDAR_ALREADY_EXISTS_FAILED (25)` が返されます。

例

次の URL では、ユーザ `jdoue` に対して、ID `jdoue:newcal` のカレンダーが作成され、カレンダー名が `New-Calendar`、カテゴリが `business` および `work` に設定されます。

```
http://webcalendarserver/createcalendar.wcap?id=b5q2o8ve2rk02nv9t6&calid=newcal&set_calprops=1&name=New-Calendar&categories=business;work
```

deletecalendar

目的

このコマンドは、ユーザのカレンダーを削除します。

パラメータ

表 10-11 は、このコマンドの 3 つのパラメータの一覧です。

表 10-11 deletecalendars のパラメータ

パラメータ	タイプ	目的	必須/任意	デフォルト
<code>calid</code>	文字列	削除するカレンダーの名前	必須	該当なし
<code>id</code>	一意の識別子文字列	セッション ID	必須	該当なし
<code>fmt-out</code>	文字列	返されるデータの形式 次の形式を指定できる <code>text/calendar</code> <code>text/xml</code> <code>text/js</code>	必須	<code>text/js</code>

説明

このコマンドを使用して、ユーザのカレンダーを削除します。`calid` を渡して、削除するカレンダーの名前を指定する必要があります。

`service.wcap.allowdeletecalendars` が設定されている場合を除き、このコマンドを使用できるのは管理権限を持つユーザだけです。

戻り値

fetchcomponents_by_range を呼び出して書式設定された出力が返されます。

エラーコード

処理に成功した場合は、エラー番号 0 がエラー文字列に追加されます。calid がデータベースに存在しない場合、delete_layer_errno[x] 値は 1 に設定されます。x は、渡された calid リストのカレンダーのインデックスです。また、errno 変数には、エラー CALENDAR_DOES_NOT_EXIST (29) が格納されます。

例

たとえば、次の URL を送信すると、newcal という名前のカレンダーが削除されます。

```
http://webcalendarserver/deletecalendar.wcap?id=b5q2o8ve2rk02nv9t6&calid=newcal
```

deletecomponents_by_range

目的

指定された範囲のイベントおよび仕事をカレンダーから削除します。

パラメータ

表 10-12 は、このコマンドの 6 つのパラメータの一覧です。

表 10-12 deletecomponents_by_range のパラメータ

パラメータ	タイプ	目的	必須 / 任意	デフォルト
brief	整数 (0、1)	JavaScript の簡略出力を出力するかどうかを示すブール値 出力形式 (fmt-out) が JavaScript (text/js) の場合にだけ適用される 1 = 簡略出力 0 = 完全出力	任意	0
calid	文字列	削除するイベントおよび仕事が含まれているカレンダー ID を、セミコロンで区切ったリスト	任意	現在のユーザの calid
dtend	ISO8601 DateTime Z 文字列 (UTC)	削除するイベントまたは仕事の終了日時 値が 0 の場合は、時間の最後までまでのイベントがすべて削除される。この値は、世界協定時刻でなければならない	任意	0

表 10-12 deletecomponents_by_range のパラメータ (続き)

パラメータ	タイプ	目的	必須 / 任意	デフォルト
dtstart	ISO8601 DateTime Z 文字列 (UTC)	削除するイベントまたは仕事の開始日時 値が 0 の場合は、時間の最初からのイベント および仕事がすべて削除される。この値は、 世界協定時刻でなければならない	任意	0
fmt-out	文字列	返されるデータの形式 次の形式を指定できる text/calendar text/xml text/js	任意	text/js
id	一意の識別子文字 列	セッション ID	必須	該当なし

説明

このコマンドを使用すると、指定した範囲のイベントおよび仕事が指定したカレンダーから削除されます。範囲を指定しない場合は、すべてのイベントおよび仕事が削除されます。範囲パラメータ dtstart および dtend は、最後に「Z」を付けた UTC 時間で指定する必要があります。UTC 時間を使用しない場合は、予測できない結果が発生します。

エラーコード

処理に成功した場合は、エラー番号 0 がエラー文字列に追加されます。カレンダーからイベントまたは仕事を削除するときにエラーが発生した場合、delete_layer_errno[x] 値は 1 に設定されます。x は、渡された calid リストのカレンダーのインデックスです。また、errno には、エラー DELETecomponents_BY_RANGE_FAILED(21) が格納されます。

例

たとえば、ユーザがカレンダー jdoe および john の削除する権限を持っていると仮定します。次の URL では、その 2 つのカレンダーからイベントおよび仕事をすべて削除します。

```
http://deletecomponents_by_range.wcap?id=2342347923479asdf
&calid=jdoe;john&dtstart=0&dtend=0
```

deleteevents_by_id

目的

イベント識別子を使用して、1つまたは複数のイベントをカレンダーから削除します。

パラメータ

表 10-13 は、このコマンドの 8 つのパラメータの一覧です。

表 10-13 deleteevents_by_id のパラメータ

パラメータ	タイプ	目的	必須/任意	デフォルト
brief	整数 (0、1)	JavaScript の簡略出力を出力するかどうかを示すブール値 出力形式 (fmt-out) が JavaScript (text/js) の場合にだけ適用される 1 = 簡略出力 0 = 完全出力	任意	0
calid	文字列	削除するイベントのカレンダー ID	必須	該当なし
fmt-out	文字列	返されるデータの形式 次の形式を指定できる text/calendar text/xml text/js	任意	text/js
id	一意の識別子文字列	セッション ID。カレンダーが公開されている場合を除き、このパラメータは必須です	必須	NULL
mod	整数 1、2、3、4	どの繰り返し部分を削除するかを示す修飾子。セミコロンで区切った複数の修飾子のリスト。リストの場合は、uid リストと要素の数が同じでなければならない 次の値を指定できる 1 = THISINSTANCE 2 = THISANDFUTURE 3 = THISANDPRIOR 4 = THISANDALL	必須	該当なし
notify	整数 0、1	参加者にこの変更を通知するかどうかを示すブール値 1 = 参加者に通知する 0 = 参加者に通知しない	任意	0

表 10-13 deleteevents_by_id のパラメータ (続き)

パラメータ	タイプ	目的	必須 / 任意	デフォルト
rid	文字列	イベントの繰り返し ID、または複数の繰り返し ID をセミコロンで区切ったリスト リストの場合、uid リストと要素の数は同じでなければならない 繰り返しが無い場合、値は 0	必須	該当なし
tzid	タイムゾーン ID 文字列	rid パラメータでタイムゾーンが指定されていない場合は、このタイムゾーンがデフォルトとして使われる 例 : America/Los_Angeles	任意	サーバのデフォルトタイムゾーン
uid	文字列	削除するイベントの一意的識別子、または複数の一意的識別子をセミコロンで区切ったリスト	必須	該当なし

説明

このコマンドを使用して、指定した 1 つまたは複数のイベントを指定したカレンダーから削除します。

エラーコード

処理に成功した場合は、エラー番号 0 がエラー文字列に追加されます。エラーが発生した場合は、delete_layer_errno と delete_event_count の 2 つのエラー配列が返されます。配列の各要素は、uid リスト内の対応するイベントについて表しています。delete_layer_errno の各要素は、対応するイベントのエラー番号を示します。delete_event_count の各要素は、対応するイベントのうち削除が完了したイベント数を示します。

161 ページの「エラー処理」も参照してください。

通知

notify パラメータには、IMIP CANCEL メッセージをイベントの電子メール参加予定者に送信するかどうかを指定します。キャンセルメッセージを送信するには、notify 値を 1 に設定します。

たとえば、次の URL は、IMIP CANCEL メッセージを uid=001 のイベントのすべての参加者に送信します。

```
http://webcalendarserver/deleteevents_by_id.wcap?id=3423423asdfasf&calid=jdoe&uid=001&notify=1
```

繰り返し

rid パラメータを渡した場合は、mod パラメータの指定に従って、繰り返しも削除されます (160 ページの「繰り返しコンポーネントの削除」を参照)。複数のイベントを削除するには、uid、rid、および mod パラメータをセミコロンで区切ったリストを指定します。この 3 つのリストの要素数は同じでなければなりません。各リストの要素は、ほかの 2 つのリストの同じ要素と対応しています。

例

たとえば、uid が uid-EVENT1 と uid-EVENT2 である繰り返しのないイベントが、データベースに 2 つあると想定します。これらのイベントは繰り返さないため、各イベントの rid 値を 0、各イベントの mod 値を 1 に設定します。

次の URL では 2 つのイベントが削除されます。

```
http://webcalendarserver/deleteevents_by_id.wcap?id=br6p3t6bh5po35r
&uid=uid-EVENT1;uid-EVENT2&rid=0;0&mod=1;1
```

次のような結果データが返されます。

```
HTTP/1.0 200
Date:Thu, 27 May 2000 18:40:24 GMT
Content-type:text/html; charset=iso-8859-1
Content-length: 4822
Last-modified:Thu, 27 May 2000 18:40:24 GMT
Pragma:no-cache
Expires: 0
Cache-Control:no-cache

<html><head><script>
function color(s) { if (s) document.bgColor=s }
var id='br6p3t6bh5po35r'
var userid='jdoe'
var calid='jdoe'
var errno=new Array()
var delete_event_errno = new Array()
var delete_event_count = new Array()
delete_event_errno[0]=0
delete_event_count[0]=1
delete_event_errno[1]=0
delete_event_count[1]=1
errno[0]=0
...
...
parent.ceCB(window.name)
</script></head></html>
```

deleteevents_by_range

目的

指定した範囲のイベントをカレンダーから削除します。

パラメータ

表 10-14 は、このコマンドの 6 つのパラメータの一覧です。

表 10-14 deleteevents_by_range のパラメータ

パラメータ	タイプ	目的	必須/任意	デフォルト
brief	整数 (0, 1)	JavaScript の簡略出力を出力するかどうかを示すブール値 出力形式 (fmt-out) が JavaScript (text/js) の場合にだけ適用される 1 = 簡略出力 0 = 完全出力	任意	0
calid	文字列	削除するイベントが含まれているカレンダー ID を、セミコロンで区切ったリスト	任意	現在のユーザの calid
dtend	ISO8601 DateTime 文字列	削除するイベントの終了日時 値が 0 の場合は、時間の最後までイベントがすべて削除される	任意	0
dtstart	ISO8601 DateTime 文字列	削除するイベントの開始日時 値が 0 の場合は、時間の最初からのイベントがすべて削除される	任意	0
fmt-out	文字列	返されるデータの形式 次の形式を指定できる text/calendar text/xml text/js	任意	text/js
id	一意の識別子 文字列	セッション ID	必須	該当なし

説明

このコマンドを使用して、指定した範囲に含まれるイベントを指定したカレンダーから削除します。範囲 (dtstart および dtend) を指定しない場合、指定したカレンダーからイベントがすべて削除されます。

指定するカレンダーが公開カレンダーである場合を除き、コマンドと一緒に `id` パラメータを指定する必要があります。データは、`fmt-out` パラメータに指定した形式で返されます。このパラメータを渡さない場合、データはデフォルトの JavaScript 形式で返されます。

エラーコード

処理に成功した場合は、エラー番号 0 がエラー文字列 `errno` に追加されます。カレンダーからイベントを削除するときにエラーが発生した場合、`delete_layer_errno[x]` 値は 1 に設定されます。x は、渡された `calid` リストのカレンダーのインデックスです。また、`errno` 変数には、エラー `DELETEEVENTS_BY_RANGE_FAILED(22)` が格納されます。

161 ページの「エラー処理」も参照してください。

例

たとえば、ユーザがカレンダー `jdoue` および `john` の削除する権限を持っていると仮定します。次の URL では、カレンダー `jdoue` および `john` から、イベントがすべて削除されます。

```
http://webcalendarserver/deleteevents_by_range.wcap?id=2342347923479asdf&calid=jdoue;john&dtstart=0&dtend=0
```

deletetodos_by_id

目的

1 つまたは複数の仕事をカレンダーから削除します。

パラメータ

表 10-15 は、このコマンドの 8 つのパラメータの一覧です。

表 10-15 `deletetodos_by_id` のパラメータ

パラメータ	タイプ	目的	必須 / 任意	デフォルト
<code>brief</code>	整数 (0, 1)	JavaScript の簡略出力を出力するかどうかを示すブール値 出力形式 (<code>fmt-out</code>) が JavaScript (<code>text/js</code>) の場合にだけ適用される 1 = 簡略出力 0 = 完全出力	任意	0
<code>calid</code>	文字列	削除する 1 つまたは複数の仕事のカレンダー ID	必須	該当なし

表 10-15 deletetodos_by_id のパラメータ (続き)

パラメータ	タイプ	目的	必須 / 任意	デフォルト
fmt-out	文字列	返されるデータの形式 次の形式を指定できる text/calendar text/xml text/js	任意	text/js
id	一意の識別子 文字列	セッション ID	必須	該当なし
mod	整数	どの繰り返し部分を削除するかを示す修飾子。セミコロンで区切った複数の修飾子のリスト。リストの場合は、uid リストと要素の数が同じでなければならない 次の値を指定できる 1 = THISINSTANCE 2 = THISANDFUTURE 3 = THISANDPRIOR 4 = THISANDALL	必須	該当なし
notify	整数 (0、1)	参加者にこの変更を通知するかどうかを示すブール値 1 = 参加者に通知する 0 = 参加者に通知しない	任意	0
rid	文字列	仕事の繰り返し ID、または複数の繰り返し ID をセミコロンで区切ったリスト リストの場合は、uid リストと要素の数が同じでなければならない 繰り返しが無い場合、値は 0	必須	該当なし
tzid	タイムゾーン ID 文字列	rid パラメータでタイムゾーンが指定されていない場合は、このタイムゾーンがデフォルトとして使われる 例 : America/Los_Angeles	任意	サーバのデフォルトのタイムゾーン
uid	文字列	削除する仕事の一意の識別子、または複数の一意の識別子をセミコロンで区切ったリスト	必須	該当なし

説明

このコマンドを使用して、指定した仕事を指定したカレンダーから削除します。

指定するカレンダーが公開カレンダーである場合を除き、コマンドと一緒に `id` パラメータを指定する必要があります。データは、`fmt-out` パラメータで指定した形式で返されます。このパラメータを渡さない場合、データはデフォルトの JavaScript 形式で返されます。

エラーコード

エラーが発生した場合は、`delete_layer_errno` と `delete_todo_count` の 2 つのエラー配列が返されます。配列の各要素は、`uid` リスト内の対応する仕事を表しています。`delete_layer_errno` の各要素は、対応する仕事のエラー番号を示します。`delete_event_count` の各要素は、対応する仕事のうち削除が完了した仕事数を示します。

161 ページの「エラー処理」も参照してください。

通知

`notify` パラメータには、IMIP CANCEL メッセージを仕事の電子メール参加予定者に送信するかどうかを指定します。`notify` が 1 の場合、キャンセルメッセージが電子メールで送信されます。

たとえば、次の URL は、IMIP CANCEL メッセージを `uid=001` の仕事のすべての参加者に送信します。

```
http://webcalendarserver/deletetodos_by_id.wcap?id=3423423asdfasf&c  
alid=jdoe&uid=001&notify=1
```

繰り返し

`rid` パラメータを渡した場合は、`mod` パラメータの指定に従って、繰り返しも削除されます。160 ページの「繰り返しコンポーネントの削除」を参照してください。

複数の仕事を削除するには、`uid`、`rid`、および `mod` パラメータをセミコロンで区切ったリストを指定します。この 3 つのリストの要素数は同じでなければなりません。各リストの要素は、ほかの 2 つのリストの同じ要素と対応しています。`rid` パラメータを渡した場合は、`mod` パラメータの指定に従って、繰り返しも削除されます。

例

たとえば、`uid` が `uid-TODO1` と `uid-TODO2` である繰り返しのないイベントが、データベースに 2 つあると想定します。これらのイベントは繰り返さないため、各イベントの `rid` 値を 0、各イベントの `mod` 値を 1 に設定します。

次の URL では 2 つのイベントが削除されます。

```
http://webcalendarserver/deletetodos_by_id.wcap?id=br6p3t6bh5po35r  
&uid=uid-TODO1;uid-TODO2&rid=0;0&mod=1;1
```

次のようなデータが出力されます。

```

HTTP/1.0 200
Date:Thu, 27 May 2000 18:40:24 GMT
Content-type:text/html; charset=iso-8859-1
Content-length: 4822
Last-modified:Thu, 27 May 2000 18:40:24 GMT
Pragma:no-cache
Expires: 0
Cache-Control:no-cache

<html><head><script>
function color(s) { if (s) document.bgColor=s }
var id='br6p3t6bh5po35r'
var userid='jdoe'
var calid='jdoe'
var errno=new Array()
var delete_todo_errno = new Array()
var delete_todo_count = new Array()
delete_todo_errno[0]=0
delete_todo_count[0]=1
delete_todo_errno[1]=0
delete_todo_count[1]=1
errno[0]=0
...
...
parent.ceCB(window.name)
</script></head></html>

```

deletetodos_by_range

目的

指定した範囲の仕事をカレンダーから削除します。

パラメータ

表 10-16 は、このコマンドの 6 つのパラメータの一覧です。

表 10-16 deletetodos_by_range のパラメータ

パラメータ	タイプ	目的	必須 / 任意	デフォルト
brief	整数 (0、1)	JavaScript の簡略出力を出力するかどうかを示すブール値 出力形式 (fmt-out) が JavaScript (text/js) の場合にだけ適用される 1 = 簡略出力 0 = 完全出力	任意	0

表 10-16 deletetodos_by_range のパラメータ (続き)

パラメータ	タイプ	目的	必須 / 任意	デフォルト
calid	文字列	削除する仕事が含まれているカレンダー ID を、セミコロンで区切ったリスト	任意	現在のユーザの calid
dtstart	ISO8601 DateTime 文字列	削除する仕事の開始日時。値が 0 の場合は、もっとも古い日時から指定した終了日時までの仕事がすべて削除される	任意	0
dtend	ISO8601 DateTime 文字列	削除する仕事の終了日時。値が 0 の場合は、指定した開始日時から最後まで仕事がすべて削除される	任意	0
id	一意の識別子 文字列	セッション ID	必須	該当なし
fmt-out	文字列	返されるデータの形式 次の形式を指定できます text/calendar text/xml text/js	任意	text/js

説明

このコマンドを使用して、指定した範囲に含まれる仕事を指定したカレンダーから削除します。範囲を指定しない場合は、すべての仕事が削除されます。たとえば、次の URL は、2000 年 3 月 1 日の午前 11 時 22 分 33 秒 (世界標準時) に発生する仕事だけを削除します。

`http://webcalendarserver/deletetodos_by_id.wcap?id=23423423434abc&calid=jdoe&uid=001&rid=20000301T112233Z&mod=1`

指定するカレンダーが公開カレンダーである場合を除き、コマンドと一緒に id パラメータを指定する必要があります。データは、fmt-out パラメータで指定した形式で返されます。このパラメータを渡さない場合、データはデフォルトの JavaScript 形式で返されます。

エラーコード

処理に成功した場合は、エラー番号 0 がエラー文字列に追加されます。カレンダーから仕事を削除するときにエラーが発生した場合、delete_layer_errno[x] 値は 1 に設定されます。x は、渡された calid リストのカレンダーのインデックスです。また、errno 変数には、エラー DELETETODOS_BY_RANGE_FAILED (23) が格納されます。

161 ページの「エラー処理」も参照してください。

export

目的

イベントおよび仕事をカレンダーからファイルにエクスポートします。

パラメータ

表 10-17 は、このコマンドの 5 つのパラメータの一覧です。

表 10-17 export のパラメータ

パラメータ	タイプ	目的	必須 / 任意	デフォルト
calid	文字列	エクスポートするイベントおよび仕事が含まれているカレンダー ID を、セミコロンで区切ったリスト。リストのすべてのカレンダーの読み取りアクセス権が必要	任意	現在のユーザの calid
content-out	文字列	出力ファイルのコンテンツタイプ。次の値を指定できる text/calendar text/xml	必須	該当なし
dtend	ISO8601 DateTime Z 文字列 (UTC)	エクスポートするイベントおよび仕事の終了日時。値が 0 の場合は、開始日から最後まですべてのコンポーネントがエクスポートされる	任意	0
dtstart	ISO8601 DateTime Z 文字列 (UTC)	エクスポートするイベントおよび仕事の開始日時。値が 0 の場合は、最も古い日付から終了日までのすべてのコンポーネントがエクスポートされる	任意	0
id	一意の識別子文字列	セッション ID	必須	該当なし

説明

このコマンドを使用して、1 つまたは複数のカレンダーからイベントおよび仕事をファイルにエクスポートします。エクスポートしたファイルの内容は、あとで import コマンドを使用してカレンダーにインポートできます。content-out パラメータの値に応じて、export.ics または export.xml という名前のファイルが作成されます。

範囲

開始日または終了日を指定しない場合は、カレンダー内のすべてのイベントおよび仕事
がファイルに追加されます。開始日および終了日を指定すると、カレンダーのその期間
内のイベントおよび仕事だけがエクスポートされます。開始日および終了日は、日付と
時刻の最後に Z を付けた UTC 時間で指定します。

HTTP POST の例

このコマンドは、HTTP POST と共に使用する必要があります (参考:ほかのコマンドで
は HTTP GET も使用可)。

例 1

次の HTTP POST メッセージでは、カレンダー jdoe と john のコンポーネントをすべ
て export.ica という名前の iCalendar ファイルにエクスポートします。

```
POST
/export.wcap?id=t95qm0n0es3bo35r&calid=jdoe;john&dtstart=0&dtend=0
  &content-out=text/calendar
Content-type:multipart/form-data;
boundary=-----41091400621290
Content-Length: 47
-----41091400621290--
WinNT; U)
Host:jdoe:12345
Accept:image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
image/png
*/*
Accept-Encoding:gzip
Accept-Language:en
Accept-Charset:iso-8859-1,*,utf-8
```

例 2

次の HTML では、export コマンドを使用して、iCalendar および XML の両方の形式
でファイルを作成する POST メッセージを生成します。

```
<form METHOD=POST ENCTYPE="multipart/form-data" NAME="john.ics"
ACTION="http://webcalendarserver:12345/export.wcap?id=t9u9m0eh8x5pu
9b
  &calid=jdoe;john&dtstart=0&dtend=0&content-out=text/calendar">
<ul>
<li>Press Export ICAL Now:<input type="submit" value="Export ICAL
now">
</li> </ul> </form>
<form METHOD=POST ENCTYPE="multipart/form-data" NAME="john.xml"
ACTION="http://webcalendarserver:12345/export.wcap?id=t9u9m0eh8x5pu
9b
  &calid=jdoe;john&dtstart=0&dtend=0&content-out=text/xml">
```

```
<ul>
<li>Press Export XML Now:<input type="submit" value="Export XML
now">
</ul> </form>
```

次の出力が生成されます。

```
HTTP/1.0 200
Date:Thu, 03 Jun 2000 22:15:52 GMT
Content-type:text/calendar
Content-disposition:attachment; filename="export.ics"
Content-length: 7004

BEGIN:VCALENDAR
METHOD:PUBLISH
VERSION:2.0
BEGIN:VEVENT
UID:tm-001
RECURRENCE-ID:20000519T010000Z
DTSTAMP:20000603T221548Z
SUMMARY:Calendar Staff
DTSTART:20000518T170000Z
DTEND:20000518T190000Z
CREATED:20000603T024254Z
LAST-MODIFIED:20000603T024254Z
PRIORITY:1
SEQ:1
GEO:37.463581;-121.897606
DESC:This is the description for event with UID = tm-001
URL:http://webcalendarserver/susan?uid=tm-001
LOCATION:Green Conference Room
STATUS:CONFIRMED
TRANSP:OPAQUE
END:VEVENT
BEGIN:VEVENT
UID:tm-001
RECURRENCE-ID:20000526T010000Z
DTSTAMP:20000603T221548Z
SUMMARY:Calendar Staff
DTSTART:20000525T170000Z
DTEND:20000525T190000Z
CREATED:20000603T024254Z
LAST-MODIFIED:20000603T024254Z
PRIORITY:1
SEQ:1
GEO:37.463581;-121.897606
DESC:This is the description for event with UID = tm-001
URL:http://webcalendarserver/susan?uid=tm-001
```

```
LOCATION:Green Conference Room
STATUS:CONFIRMED
TRANSP:OPAQUE
END:VEVENT
END:VCALENDAR
```

fetchcomponents_by_alarmrange

目的

カレンダーからアラームトリガーを持つイベントおよび仕事を取得します。

パラメータ

表 10-18 は、このコマンドの 7 つのパラメータの一覧です。

表 10-18 fetchcomponents_by_alarmrange のパラメータ

パラメータ	タイプ	目的	必須 / 任意	デフォルト
brief	整数 (0, 1)	JavaScript の簡略出力を出力するかどうかを示すブール値 出力形式 (fmt-out) が JavaScript (text/js) の場合にだけ適用される 1 = 簡略出力 0 = 完全出力	任意	0
calid	文字列	取得するコンポーネントが含まれているカレンダー ID を、セミコロンで区切ったリスト	任意	現在のユーザの calid
compstate	コンポーネントの状態のキーワードを、セミコロンで区切ったリスト	取り込むコンポーネントの状態のリスト compstate 値については、168 ページの表 10-5 を参照	任意	ALL
dtend	ISO8601 DateTime Z 文字列	取得するイベントおよび仕事の終了日時 値が 0 の場合は、すべてのイベントが取り込まれる	任意	0
dtstart	ISO8601 DateTime Z 文字列	指定した期間に配信されるアラームを持つイベントまたは仕事の開始日時 値が 0 の場合は、最初の日時のイベントからすべて取り込まれる	任意	0

表 10-18 fetchcomponents_by_alarmrange のパラメータ (続き)

パラメータ	タイプ	目的	必須 / 任意	デフォルト
fmt-out	文字列	返されるデータの形式 次の形式を指定できる text/calendar text/xml text/js	任意	text/js
id	一意の識別子文字列	セッション ID	必須	該当なし
maxResults	整数	返されるイベントおよび仕事の最大数。0 の場合、最大数は適用されず、検出されたイベントおよび仕事がすべて返される	任意	0
tzid	タイムゾーン ID 文字列	rid パラメータにタイムゾーンが含まれていない場合は、このタイムゾーンがデフォルトとして使われる 例 : America/Los_Angeles	任意	サーバのデフォルトのタイムゾーン

説明

このコマンドは、指定した期間に配信されるアラームを持つイベントおよび仕事のリストを返します。

出力形式

データは、fmt-out パラメータで指定した形式で返されます。このパラメータを渡さない場合、データはデフォルトの JavaScript 形式で返されます。

JavaScript 出力の場合、イベントは 2 つの配列に分けられます。継続時間の短いイベント (ほとんどの通常のイベント) は、event 配列に出力されます。24 時間以上継続するイベント (終日イベント) は、すべて eventD 配列に出力されます。終日イベントがある場合は、isAllDay フラグがオンになります。

maxResults 値

最大数 n を指定すると、指定した範囲から最初の n 個のイベントと最初の n 個の仕事が返されます。たとえば、maxResults 値に 75 を指定すると、次の変数を含む JavaScript が返されます。

```
var maxResults=75 /* 渡された最大数 */
var size=75      /* イベントのサイズが最大 75 に制限されています */
var todosize=28 /* 仕事のサイズが 75 に満たないため、影響を受けません */
```

maxResults パラメータを 0 に設定するか、渡さなかった場合は、var maxResults 文を含まない JavaScript が返されます。

戻り値

calid で指定した各カレンダーから、dtstart および dtend で指定した時間範囲内に配信されるアラームを持つ、イベントおよび仕事 that 返されます。範囲は、ISO8601 DateTime Z 文字列形式で指定します。

開始日時も終了日時も指定しなかった場合は、アラームを持つすべてのイベントおよび仕事 that、指定した最大数まで返されます。

エラーコード

処理に成功した場合は、エラー番号 0 がエラー文字列に追加されます。カレンダーにアクセスできない場合、またはカレンダーが見つからない場合、errno は FETCH_BY_ALARM_RANGE_FAILED (41) です。

例

たとえば、次の 3 つのイベントがあると想定します。

- eventA: 2000 年 2 月 10 日の午前 10 時 00 分 (GMT) にアラーム
- eventB: 2000 年 12 月 25 日の午後 12 時 30 分 (GMT) にアラーム
- todoA: 2000 年 1 月 20 日の午後 1 時 15 分 (GMT) にアラーム

このカレンダーに対する 3 つのクエリーとその戻り値について説明します。

1. 次のクエリーは、アラームを持つイベントおよび仕事をすべて取り込みます。

```
http://webcalendarserver/fetchcomponents_by_alarmrange.wcap?id=abcdefg&dtstart=0&dtend=0
```

eventA、eventB、および todoA が返されます。

2. 次のクエリーは、2000 年 12 月 1 日～2001 年 1 月 31 日に配信されるアラームを持つイベントおよび仕事をすべて、取り込みます。

```
http://webcalendarserver/fetchcomponents_by_alarmrange.wcap?id=abcdefg&dtstart=20001201T112233Z&dtend=20010131T112233Z
```

eventB および todoA が返されます。

3. 次のクエリーは、2000 年 1 月 1 日～2000 年 6 月 1 日に配信されるアラームを持つイベントおよび仕事をすべて、取り込みます。

```
http://webcalendarserver/fetchcomponents_by_alarmrange.wcap?id=abcdefg&dtstart=20000101T112233Z&dtend=20000601T112233Z
```

eventA および todoA が返されます。

fetchcomponents_by_attendee_error

目的

グループスケジューリングメッセージの送信中にエラーが発生したコンポーネントのリストを取り込みます。

パラメータ

表 10-19 は、このコマンドの 5 つのパラメータの一覧です。

表 10-19 fetchcomponents_by_attendee_error のパラメータ

パラメータ	タイプ	目的	必須 / 任意	デフォルト
attendee	文字列	検索する参加者の calid。calid パラメータで指定されているカレンダーから、この参加者に関するイベントで発生したすべてのエラーを検索する このパラメータが指定されない場合、主な所有者のカレンダーから、すべての参加者のすべてのイベントエラーを検索する	任意	該当なし
brief	整数 (0、1)	JavaScript の簡略出力を出力するかどうかを示すブール値 出力形式 (fmt-out) が JavaScript (text/js) の場合にだけ適用される 1 = 簡略 0 = 完全	任意	0
calid	文字列	取得するコンポーネントが含まれているカレンダー ID を、セミコロンで区切ったリスト	任意	現在のユーザの calid
fmt-out	文字列	返されるデータの形式 次の形式を指定できる text/calendar text/xml text/js	任意	text/js
id	一意の識別子文字列	セッション ID	必須	該当なし
maxResults	整数	返されるイベントおよび仕事の最大数。0 の場合、最大数は適用されず、検出されたすべてのイベントおよび仕事は返される	任意	0

説明

このコマンドを使用して、グループスケジューリングメッセージの送信中にエラーが発生したイベントおよび仕事のリストを取得します。このコマンドの機能は、`fetchcomponents_by_range` と似ています。

出力形式

データは、`fmt-out` パラメータで指定した形式で返されます。このパラメータを渡さない場合、データはデフォルトの JavaScript 形式で返されます。

JavaScript 出力の場合、イベントは 2 つの配列に分けられます。継続時間の短いイベント (ほとんどの通常のイベント) は、`event` 配列に出力されます。24 時間以上継続するイベント (終日イベント) は、すべて `eventD` 配列に出力されます。終日イベントがある場合は、`isAllDay` フラグがオンになります。

maxResults 値

最大数 n を指定すると、指定した範囲から最初の n 個のイベントと最初の n 個の仕事が返されます。たとえば、`maxResults` 値に 75 を指定すると、次の変数を含む JavaScript が返されます。

```
var maxResults=75 /* 渡された最大数 */
var size=75      /* イベントのサイズが最大 75 に制限されています */
var todosize=28 /* 仕事のサイズが 75 に満たないため、影響を受けません */
```

`maxResults` パラメータを 0 に設定するか、渡さなかった場合は、`var maxResults` 文を含まない JavaScript が返されます。

戻り値

`calid` で指定した各カレンダーから、指定した参加者のイベントおよび仕事のうち、グループスケジューリングメッセージの送信中にエラーが発生したものが返されます。

たとえば、`calid` パラメータに `cal1` および `cal2` を指定し、`attendee` パラメータに `jdoue` を指定した場合、`cal1` および `cal2` から `jdoue` が参加者として含まれているイベントのエラーが検索されます。次の表の `cal1` と `cal2` には、4 つのイベントとそれぞれに関連付けられている参加者が格納されています。

cal1 のイベント	cal2 のイベント
イベント - 1c1	イベント - 1c2
参加者: jdoe	参加者: john
状態: エラー	状態: OK
イベント - 2c1	イベント - 2c2
参加者: susan	参加者: jdoe
状態: エラー	状態: エラー

cal1 のイベント	cal2 のイベント
イベント - 3c1	イベント - 3c2
参加者 : jdoe	参加者 : susan
状態 : OK	状態 : OK
イベント - 4c1	イベント - 4c2
参加者 : john	参加者 : susan
状態 : OK	状態 : エラー

参加者 jdoe に関してこのコマンドを実行すると、イベント 1c1 と 2c2 が返されます。

エラーコード

処理に成功した場合は、エラー番号 0 がエラー文字列に追加されます。何らかの理由でコマンドの実行に失敗した場合、errno は FETCH_BY_ATTENDEE_ERROR_FAILED(42) です。

fetchcomponents_by_lastmod

目的

指定した期間中に変更されたコンポーネントのリストを取り込みます。

パラメータ

表 10-20 は、このコマンドの 7 つのパラメータの一覧です。

表 10-20 fetchcomponents_by_lastmode のパラメータ

パラメータ	タイプ	目的	必須 / 任意	デフォルト
brief	整数 (0, 1)	JavaScript の簡略出力を出力するかどうかを示すブール値 出力形式 (fmt-out) が JavaScript (text/js) の場合にだけ適用される 1 = 簡略 0 = 完全	任意	0
calid	文字列	取得するコンポーネントが含まれているカレンダー ID を、セミコロンで区切ったリスト	任意	現在のユーザの calid

表 10-20 fetchcomponents_by_lastmode のパラメータ (続き)

パラメータ	タイプ	目的	必須 / 任意	デフォルト
compstate	コンポーネントの状態のキーワードをセミコロンで、区切ったリスト	取り込むコンポーネントの状態のリスト compstate 値については、168 ページの表 10-5 を参照	任意	ALL
dtend	ISO8601 DateTime Z 文字列	取得するイベントの終了日時 値が 0 の場合は、すべてのイベントが取り込まれる	任意	0
dtstart	ISO8601 DateTime Z 文字列	取得するイベントの開始日時 値が 0 の場合は、最初の日時のイベントからすべて取り込まれる	任意	0
fmt-out	文字列	返されるデータの形式 次の形式を指定できる text/calendar text/xml text/js	任意	text/js
id	一意の識別子文字列	セッション ID	必須	該当なし
maxResults	整数	返されるイベントおよび仕事の最大数。0 の場合、最大数は適用されず、検出されたイベントおよび仕事すべて返される	任意	0
tzid	タイムゾーン ID 文字列	rid パラメータにタイムゾーンが含まれていない場合は、このタイムゾーンがデフォルトとして使われる 例: America/Los_Angeles	任意	サーバのデフォルトのタイムゾーン

説明

このコマンドを使用して、指定した期間中に変更されたイベントおよび仕事のリストを取得します。このコマンドの機能は、fetchcomponents_by_range と似ています。

出力形式

データは、fmt-out パラメータで指定した形式で返されます。このパラメータを渡さない場合、データはデフォルトの JavaScript 形式で返されます。

JavaScript 出力の場合、イベントは 2 つの配列に分けられます。継続時間の短いイベント (ほとんどの通常のイベント) は、`event` 配列に出力されます。24 時間以上継続するイベント (終日イベント) は、すべて `eventD` 配列に出力されます。終日イベントがある場合は、`isAllDay` フラグがオンになります。

maxResults 値

最大数 `n` を指定すると、指定した範囲から最初の `n` 個のイベントと最初の `n` 個の仕事が返されます。たとえば、`maxResults` 値に 75 を指定すると、次の変数を含む JavaScript が返されます。

```
var maxResults=75 /* 渡された最大数 */
var size=75      /* イベントのサイズが最大 75 に制限されています */
var todosize=28 /* 仕事のサイズが 75 に満たないため、影響を受けません */
```

`maxResults` パラメータを 0 に設定するか、渡さなかった場合は、`var maxResults` 文を含まない JavaScript が返されます。

戻り値

`calid` で指定した各カレンダーから、`dtstart` および `dtend` で指定した時間範囲内に変更されたカレンダーのイベントおよび仕事が返されます。範囲は、ISO8601 DateTime Z 文字列形式で指定します。

開始日時も終了日時も指定しなかった場合は、変更されたイベントおよび仕事がすべて、指定した最大数まで返されます。

エラーコード

処理に成功した場合は、エラー番号 0 がエラー文字列に追加されます。カレンダーにアクセスできない場合、またはカレンダーが見つからない場合は、エラー番号がエラー文字列に追加されます。

例

たとえば、カレンダー `jdoe` に次の 3 つのイベントがあります。

- `eventA`: 最終更新日時は 2000 年 2 月 10 日の午前 10 時 00 分 (GMT)
- `eventA`: 最終更新日時は 2000 年 12 月 25 日の午後 12 時 30 分 (GMT)
- `todoA`: 最終更新日時は 2000 年 1 月 20 日の午後 1 時 15 分 (GMT)

このカレンダーに対するいくつかのクエリーとその戻り値について説明します。

```
http://webcalendarserver/fetchcomponents_by_lastmod.wcap?id=jdoe
&dtstart=0&dtend=0
```

このクエリーは、これまでに修正されたイベントおよび仕事をすべて取り込みます。したがって、`eventA`、`eventB`、および `todoA` が返されます。

```
http://webcalendarserver/fetchcomponents_by_lastmod.wcap?id=jdoe
&dtstart=20001201T112233Z&dtend=20000131T112233Z
```

このクエリーは、2000年12月1日～2000年1月31日に変更されたイベントおよび仕事をすべて取り込みます。したがって、eventB および todoA が返されます。

```
http://webcalendarserver/fetchcomponents_by_lastmod.wcap?id=jdoe
&dtstart=20000101T112233Z&dtend=20000601T112233Z
```

このクエリーは、2000年1月1日～2000年6月1日に変更されたイベントおよび仕事をすべて取り込みます。したがって、eventA および todoA が返されます。

fetchcomponents_by_range

目的

カレンダーのイベントおよび仕事を取得します。

パラメータ

表 10-21 は、このコマンドの7つのパラメータの一覧です。

表 10-21 fetchcomponents_by_range のパラメータ

パラメータ	タイプ	目的	必須 / 任意	デフォルト
brief	整数 (0, 1)	JavaScript の簡略出力を出力するかどうかを示すブール値 出力形式 (fmt-out) が JavaScript (text/js) の場合にだけ適用される 1 = 簡略出力 0 = 完全出力	任意	0
calid	文字列	取得するコンポーネントが含まれているカレンダー ID を、セミコロンで区切ったリスト	任意	現在のユーザの calid
compstate	コンポーネントの状態のキーワードを、セミコロンで区切ったリスト	取り込むコンポーネントの状態のリスト compstate 値については、168 ページの表 10-5 を参照	任意	ALL
dtend	ISO8601 DateTime Z 文字列	取得するイベントの終了日時 値が 0 の場合は、すべてのイベントが取り込まれる	任意	0
dtstart	ISO8601 DateTime Z 文字列	取得するイベントの開始日時 値が 0 の場合は、最初の日時のイベントからすべて取り込まれる	任意	0

表 10-21 fetchcomponents_by_range のパラメータ (続き)

パラメータ	タイプ	目的	必須 / 任意	デフォルト
fmt-out	文字列	返されるデータの形式 次の形式を指定できる text/calendar text/xml text/js	任意	text/js
id	一意の識別子文字列	セッション ID	必須	該当なし
maxResults	整数	取得するイベントおよび仕事の最大数。0 の場合、最大数は適用されず、検出されたすべてのイベントおよび仕事は返される	任意	0
tzid	タイムゾーン ID 文字列	rid パラメータにタイムゾーンが指定されていない場合は、デフォルトのタイムゾーンが使われる 例: America/Los_Angeles	任意	サーバのデフォルトのタイムゾーン

説明

このコマンドを使用して、指定した 1 つまたは複数のカレンダーのプロパティ、イベント、および仕事を取得します。

出力形式

データは、fmt-out パラメータで指定した形式で返されます。このパラメータを渡さない場合、データはデフォルトの JavaScript 形式で返されます。

JavaScript 出力の場合、イベントは 2 つの配列に分けられます。継続時間の短いイベント (ほとんどの通常のイベント) は、event 配列に出力されます。24 時間以上継続するイベント (終日イベント) は、すべて eventD 配列に出力されます。終日イベントがある場合は、isAllDay フラグがオンになります。

戻り値

calid で指定した各カレンダーに対して、dtstart および dtend に指定した時間範囲にあるカレンダープロパティ、イベント、および仕事が返されます。範囲は、ISO8601 DateTime Z 文字列形式で指定します。

開始日時も終了日時も指定しなかった場合は、イベントおよび仕事がすべて、指定した最大数まで返されます。

エラーコード

処理に成功した場合は、エラー番号 0 がエラー文字列に追加されます。カレンダーにアクセスできない場合、またはカレンダーが見つからない場合は、エラー番号がエラー文字列に追加されます。

例

たとえば、カレンダー `jdoe` および `susan` に関する 2000 年 1 月 1 日～2000 年 2 月 1 日のイベントを取得するには、次の URL を使用します。

```
http://webcalendarserver:81/fetchcomponents_by_range.wcap?id=b5q2o8ve2rk02nv9t6&calid=jdoe
```

ヒント カレンダーに有効なタイムゾーン (`tzid`) が関連付けられていない場合は、GMT タイムゾーンが使用されます。

次のテキストが返されます。

```
timezoneList[0]=new TZ('GMT',
'GMT',
'GMT',
'+0000',
'+0000',
new Array())
```

出力形式

データは、`fmt-out` パラメータで指定した形式で返されます。このパラメータを渡さない場合、データはデフォルトの JavaScript 形式で返されます。

JavaScript 出力の場合、イベントは 2 つの配列に分けられます。継続時間の短いイベント (ほとんどの通常のイベント) は、`event` 配列に出力されます。24 時間以上継続するイベント (終日イベント) は、すべて `eventD` 配列に出力されます。終日イベントがある場合は、`isAllDay` フラグがオンになります。

maxResults 値

最大数 `n` を指定すると、指定した範囲から最初の `n` 個のイベントと最初の `n` 個の仕事が返されます。たとえば、`maxResults` 値に 75 を指定すると、次の変数を含む JavaScript が返されます。

```
var maxResults=75 /* 渡された最大数 */
var size=75 /* イベントのサイズが最大 75 に制限されています */
var todosize=28 /* 仕事のサイズが 75 に満たないため、影響を受けません */
```

`maxResults` パラメータを 0 に設定するか、渡さなかった場合は、`var maxResults` 文を含まない JavaScript が返されます。

エラーコード

処理に成功した場合は、エラー番号 0 がエラー文字列に追加されます。カレンダーにアクセスできない場合、またはカレンダーが見つからない場合は、エラー番号がエラー文字列に追加されます。

例

たとえば、カレンダー jdoe および john に関する 2000 年 1 月 1 日～2001 年 2 月 1 日のイベントを取得するには、次の URL を使用します。

```
http://webcalendarserver/fetchcomponents_by_range.wcap?id=asdfasdf32424&calid=jdoe;john&dtstart=20000101T000000&dtend=20010101T000000
```

この例のように、開始日および終了日を指定しない場合は、カレンダー内のすべてのイベントおよび仕事が返されます。

```
http://webcalendarserver/fetchcomponents_by_range.wcap?id=b5q2o8ve2rk02nv9t6&calid=jdoe
```

```
HTTP/1.0 200
```

```
Date:Tue, 28 Jul 2000 22:43:01 GMT
```

```
Content-type: text/html; charset=iso-8859-1
```

```
Content-length: 5504
```

```
Last-modified:Tue, 15 Jun 2000 22:43:01 GMT
```

```
Pragma:no-cache
```

```
Expires: 0
```

```
Cache-Control:no-cache
```

```
<html><head><script>
function color(s) { if (s) document.bgColor=s }
var id='bb2ot68wp0t95q'
var userid='jdoe'
var calid='jdoe'
var errno=new Array()
var timezoneList = new Array()
var calprops = new Array()
var layer_errno = new Array()

function iso(p) {
if (!p) return null;
var y = p.substring(0,4)
var m = p.substring(4,6)
var d = p.substring(6,8)
var h = p.substring(9,11)
var i = p.substring(11,13)
var s = p.substring(13,15)
return new Date(y, m - 1, d, h, i, s, 0)
```

コマンド

```
}  
function  
E(a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11,a12,a13,a14,a15,a16,a17,a18,a1  
9,a20,a21,a22,a23,a24,a25,a26,a27,a28,a29,a30,a31,a32,a33) {  
  
this.uid=a1  
this.rid=a2  
this.calid=a3  
this.dtstart=iso(a4)  
this.dtend=iso(a5)  
this.status=a6  
this.isAllDay=a7  
this.summary=a8  
this.created=iso(a9)  
this.lastMod=iso(a10)  
this.seq=a11  
this.geo=new Array(a12,a13)  
this.desc=a14  
this.icsUrl=a15  
this.icsClass=a16  
this.location=a17  
this.alarmStart=a18  
this.alarmEmails=a19  
this.rrules=a20  
this.rdates=a21  
this.exrules=a22  
this.exdates=a23  
this.tzid=a24  
this.priority=a25  
this.relatedTos=a26  
this.resources=a27  
this.categories=a28  
this.attachments=a29  
this.contacts=a30  
this.attendees=a31  
this.orgEmail=a32  
this.linkCalid=a33  
}function  
T(a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11,a12,a13,a14,a15,a16,a17,a18,a1  
9,a20,a21,a22,a23,a24,a25,a26,a27,a28,a29,a30,a31,a32,a33,a34,a35) {  
  
this.uid=a1  
this.rid=a2  
this.calid=a3  
this.dtstart=iso(a4)  
this.due=iso(a5)  
this.status=a6  
this.isAllDay=a7
```



```
this.summary=a8
this.created=iso(a9)
this.lastMod=iso(a10)
this.seq=a11
this.geo=new Array(a12,a13)
this.desc=a14
this.icsUrl=a15
this.icsClass=a16
this.location=a17
this.alarmStart=a18
this.alarmEmails=a19
this.rrules=a20
this.rdates=a21
this.exrules=a22
this.exdates=a23
this.completed=iso(a24)
this.percent=a25
this.tzid=a26
this.priority=a27
this.relatedTos=a28
this.resources=a29
this.categories=a30
this.attachments=a31
this.contacts=a32
this.attendees=a33
this.orgEmail=a34
this.linkCalid=a35
}
function TZ(a1,a2,a3,a4,a5,a6) {
this.tzid=a1
this.sName=a2
this.dName=a3
this.sOffset=a4
this.dOffset=a5
this.crossOver=a6
}
function
CP(a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11,a12,a13,a14,a15) {
this.calid=a1
this.name=a2
this.parent=a3
this.tzid=a4
this.read=a5
this.write=a6
this.charset=a7
this.lang=a8
this.master=a9
```

コマンド

```
this.desc=a10
this.lastMod=iso(a11)
this.created=iso(a12)
this.primaryOwner=a13
this.owners=a14
this.categories=a15
}
var event=new Array()
var event=new Array()
var todo=new Array()
var status_types=new Array('confirmed', 'tentative',
'cancelled')
calprops[0] = new CP('John Doe',

'jdoe',
'',
'',
'0',
'0',
'',
'',
'',
'',
'Work Calendar for John Doe',
'20000615T164146Z',
'20000615T164146Z',
'jdoe',
new Array('john'),
new Array(),
new Array('business'))
event[0]=new E('52452531427158444',
'0',
'jdoe',
'20001223T233000',
'20001224T023000',
0,0,'jdoe event 3',
'20000615T222725','20000615T222725',
'1',
'37.463581','-121.897606',
'This is the description for event with UID =
52452531427158444',
'http://webcalendarserver/'+ 'john?uid=52452531427158444',
'',
'Green Conference Room',
0,
new Array(),
new Array(),
new Array(),
```

```

new Array(),
new Array(), 'GMT', '1',
new Array(),
new Array(),
new Array(),
new Array(),
new Array(),
new Array(),
'',
'')
event[1]=new E('51452531524178674', '0',
'jdoe',
'20001223T130000',
'20001223T123000',

0,0,'jdoe event 2',
'20000615T222725','20000615T222725',
'1',
'37.463581','-121.897606',
'This is the description for event with UID =
51452531524178674',
'http://webcalendarserver/'+ 'john?uid=51452531524178674',
'',
'Green Conference Room',
0,
new Array(),
new Array(),
new Array(),
new Array(),
new Array(), 'GMT', '1',
new Array(),
new Array(),
new Array(),
new Array(),
new Array(),
new Array(),
new Array(),
new Array(),
new Array(),
'',
'')
event[2]=new E('tm-001',
'20000519T010000Z',
'jdoe',
'20000518T170000',
'20000518T190000',
0,0,'Calendar Staff',
'20000615T222725', '
20000615T222725',
'1',

```

コマンド

```
'37.463581','-121.897606',
'This is the description for event with UID = tm-001',
'http://webcalendarserver/'+ 'john?uid=tm-001',
'',
'Green Conference Room',
0,
new Array(),
new Array(),
new Array(),
new Array(),
new Array(), 'GMT', '1',
new Array(),
new Array(),
new Array(),
new Array(),

new Array(),
new Array(),
'',
'')
event[3]=new E('tm-001',
'20000526T010000Z',
'jdoe',
'20000525T170000',
'20000525T190000',
0,0,'Calendar Staff',
'20000615T222725','20000615T222725',
'1',
'37.463581','-121.897606',
'This is the description for event with UID = tm-001',
'http://webcalendarserver/'+ 'john?uid=tm-001',
'',
'Green Conference Room',
0,
new Array(),
new Array(),
new Array(),
new Array(),
new Array(), 'GMT', '1',
new Array(),
new Array(),
new Array(),
new Array(),
new Array(),
new Array(),
new Array(),
new Array(),
new Array(),
'',
'')
```

```

event[4]=new E('tm-001',
'20000602T010000Z',
'jdoe',
'20000601T170000',
'20000601T190000',
0,0,'Calendar Staff',
'20000615T222725','20000615T222725',
'1',
'37.463581','-121.897606',
'This is the description for event with UID = tm-001',
'http://webcalendarserver/'+ 'john?uid=tm-001',
'',
'Green Conference Room',
0,
new Array(),
new Array(),

new Array(),
new Array(),
new Array(),'GMT','1',
new Array(),
new Array(),
new Array(),
new Array(),
new Array(),
new Array(),
'',
'')
todo[0]=new T('abctodo',
'0',
'jdoe',
'20000304T143000',
'20000620T222725',
3,0,'Fix the bugs',
'20000615T222725','20000615T222725',
'1',
'37.463581','-121.897606',
'This is the description for event with UID = abctodo',
'http://webcalendarserver/'+ 'john?uid=TODO1',
'',
'Green Conference Room',
0,
new Array(),
new Array(),
new Array(),
new Array(),
new Array(),'19700101T000000',

```

```
'-1',
'GMT','1',
new Array(),
new Array(),
new Array(),
new Array(),
new Array(),
new Array(),
'',
'')
layer_errno[0]=0
var size=5
var todosize=1
var dtstartrange="0"

var dtendrange="0"
errno[0]=0
parent.ceCB(window.name)
</script></head></html>
```

fetchevents_by_id

目的

指定したカレンダーのイベントを取得します。

パラメータ

表 10-22 は、このコマンドの7つのパラメータの一覧です。

表 10-22 fetchevents_by_id のパラメータ

パラメータ	タイプ	目的	必須/任意	デフォルト
brief	整数 (0、1)	JavaScript の簡略出力を出力するかどうかを示すブール値 出力形式 (fmt-out) が JavaScript (text/js) の場合にだけ適用される 1 = 簡略出力 0 = 完全出力	任意	0
calid	文字列	取得するイベントが含まれているカレンダーの一意の識別子	任意	現在のユーザの calid

表 10-22 fetchevents_by_id のパラメータ (続き)

パラメータ	タイプ	目的	必須 / 任意	デフォルト
compstate	コンポーネントの状態のキーワードを、セミコロンで区切ったリスト	取り込むコンポーネントの状態のリスト compstate 値については、168 ページの表 10-5 を参照	任意	ALL
fmt-out	文字列	返されるデータの形式 次の形式を指定できる text/calendar text/xml text/js	任意	text/js
id	一意の識別子文字列	セッション ID	必須	該当なし
mod	整数	取得する繰り返し修飾子。次の値を指定できる 1 = THISINSTANCE 2 = THISANDFUTURE 3 = THISANDPRIOR 4 = THISANDALL	任意	1 (THISINSTANCE)
rid	ISO8601 DateTime Z 文字列	イベントの繰り返し ID。繰り返しのないイベントの場合は、0 に設定する	任意	0
tzid	タイムゾーン ID 文字列	rid パラメータにタイムゾーンが含まれていない場合は、このタイムゾーンがデフォルトとして使われる 例 : America/Los_Angeles	任意	サーバのデフォルトのタイムゾーン
uid	文字列	イベントの一意の識別子	必須	該当なし

説明

このコマンドを使用して、指定したカレンダーから指定したイベントおよびその繰り返しを取得します。指定するカレンダーが公開カレンダーである場合を除き、コマンドと一緒に id パラメータを指定する必要があります。このコマンドは、mod パラメータで指定した範囲の繰り返しを返します。170 ページの「繰り返し処理」を参照してください。

出力形式

データは、fmt-out パラメータで指定した形式で返されます。このパラメータを渡さない場合、データはデフォルトの JavaScript 形式で返されます。

JavaScript 出力の場合、イベントは 2 つの配列に分けられます。継続時間の短いイベント (ほとんどの通常のイベント) は、event 配列に出力されます。24 時間以上継続するイベント (終日イベント) は、すべて eventD 配列に出力されます。終日イベントがある場合は、isAllDay フラグがオンになります。

戻り値

データは、fmt-out パラメータで指定した形式で返されます。このパラメータを渡さない場合、データはデフォルトの JavaScript 形式で返されます。

エラーコード

処理に成功した場合は、エラー番号 0 がエラー文字列に追加されます。カレンダーにアクセスできない場合、またはカレンダーが見つからない場合は、エラー番号がエラー文字列に追加されます。

fetchtodos_by_id

目的

指定したカレンダーの仕事を取得します。

パラメータ

表 10-23 は、このコマンドの 7 つのパラメータの一覧です。

表 10-23 fetchtodos_by_id のパラメータ

パラメータ	タイプ	目的	必須 / 任意	デフォルト
brief	整数 (0、1)	JavaScript の簡略出力を出力するかどうかを示すブール値 出力形式 (fmt-out) が JavaScript (text/js) の場合にだけ適用される 1 = 簡略出力 0 = 完全出力	任意	0
calid	文字列	取得する仕事が含まれているカレンダーの一意的識別子	任意	現在のユーザの calid
compstate	コンポーネントの状態のキーワードを、セミコロンで区切ったリスト	取り込むコンポーネントの状態のリスト compstate 値については、168 ページの表 10-5 を参照	任意	ALL

表 10-23 fetchtodos_by_id のパラメータ (続き)

パラメータ	タイプ	目的	必須 / 任意	デフォルト
fmt-out	文字列	返されるデータの形式 次の形式を指定できる text/calendar text/xml text/js	任意	text/js
id	一意の識別子文字列	セッション ID	必須	該当なし
mod	整数	取得する繰り返し修飾子。次の値を指定できる 1 = THISINSTANCE 2 = THISANDFUTURE 3 = THISANDPRIOR 4 = THISANDALL	任意	1 (THISINSTANCE)
rid	ISO8601 DateTime Z 文字列	仕事の繰り返し ID。繰り返しのない 仕事の場合は、0 に設定する	任意	0
tzid	タイムゾーン ID 文字列	rid パラメータにタイムゾーンが指 定されていない場合は、デフォルト のタイムゾーンが使われる 例 : America/Los_Angeles	任意	サーバの デフォルトの タイムゾーン
uid	文字列	仕事の一意的識別子	必須	該当なし

説明

このコマンドを使用して、指定したカレンダーから指定した仕事およびその繰り返しを取得します。指定するカレンダーが公開カレンダーである場合を除き、コマンドと一緒に id パラメータを指定する必要があります。

出力形式

データは、fmt-out パラメータで指定した形式で返されます。このパラメータを渡さない場合、データはデフォルトの JavaScript 形式で返されます。

JavaScript 出力の場合、イベントは 2 つの配列に分けられます。継続時間の短いイベント (ほとんどの通常のイベント) は、event 配列に出力されます。24 時間以上継続するイベント (終日イベント) は、すべて eventD 配列に出力されます。終日イベントがある場合は、isAllDay フラグがオンになります。

戻り値

calid で指定した各カレンダーから、そのカレンダーの仕事が返されます。仕事に繰り返しがある場合は、rid パラメータおよび mod パラメータの指定に従って、繰り返しが返されます。170 ページの「繰り返し処理」を参照してください。

エラーコード

処理に成功した場合は、エラー番号 0 がエラー文字列に追加されます。カレンダーにアクセスできない場合、またはカレンダーが見つからない場合は、エラー番号がエラー文字列に追加されます。

get_all_timezones

目的

サーバがサポートしているすべてのタイムゾーンに関するデータを取得します。

パラメータ

表 10-24 は、このコマンドの 4 つのパラメータの一覧です。

表 10-24 get_all_timezones のパラメータ

パラメータ	タイプ	目的	必須 / 任意	デフォルト
dtend	ISO8601 DateTime Z 文字列	取得するクロスオーバー値の終了日 値が 0 の場合は、認識されている最後の年 (2087) 以前のクロスオーバー日付がすべて取得される	任意	0
dtstart	ISO8601 DateTime Z 文字列	取得するクロスオーバー値の開始日 値が 0 の場合は、認識されている最初の年 (1987) 以降のクロスオーバー日付がすべて取得される	任意	0
fmt-out	文字列	返されるデータの形式 次の形式を指定できる text/calendar text/xml text/js	任意	text/js
id	一意の識別子文字列	セッション ID	必須	該当なし

説明

このコマンドを使用して、サーバがサポートしているすべてのタイムゾーンに関するデータを取得します。クロスオーバー値は、そのタイムゾーンで夏時間が開始 / 終了する日付になるように定義されています。奇数番目の日付に、夏時間が始まります。偶数番目の日付に、夏時間が終わります。タイムゾーンに夏時間がない場合、この値は空の文字列に設定されます。

戻り値

`dtstart` パラメータおよび `dtend` パラメータで年の範囲を指定すると、範囲内の年のクロスオーバー日付だけが返されます。年の範囲を指定しない場合は、認識されている最初の年から最後の年 (1987 ~ 2087) までのクロスオーバー日付がすべて返されます。

データは、`fmt-out` パラメータで指定した形式で返されます。`fmt-out` パラメータを渡さない場合は、デフォルトの JavaScript 形式が使用されます。

エラーコード

タイムゾーンの取得中にエラーが発生した場合は、エラー `GET_ALL_TIMEZONES_FAILED (24)` が返されます。

例

クロスオーバー日付が 1998 年の特定の時点から 2006 年に制限されている場合の、タイムゾーン配列要素の例を挙げます。

```
timezoneList[20] = new TZ('America/Los_Angeles',
  'PST',
  'PDT',
  '-0800',
  '-0700',
  new Array
  ('19981025T090000Z', '20000404T100000Z', '20001031T090000Z',
  '20000402T100000Z', '20001029T090000Z', '20010401T100000Z',
  '20011028T090000Z', '20020407T100000Z', '20021027T090000Z',
  '20030406T100000Z', '20031026T090000Z', '20040404T100000Z',
  '20041031T090000Z', '20050403T100000Z', '20051030T090000Z',
  '20060402T100000Z', '20061029T090000Z'))
```

「America/Phoenix」タイムゾーンに夏時間はありません。したがって、夏時間を表している項目は、標準時間の項目とまったく同じになります。また、クロスオーバー文字列は空の文字列に設定されます。

```
timezoneList[23] = new TZ('America/Phoenix',
  'MST',
  'MST',
  '-0700',
  '-0700',
  new Array())
```

get_calprops

目的

カレンダープロパティを取得します

パラメータ

表 10-25 は、このコマンドの 3 つのパラメータの一覧です。

表 10-25 get_calprops のパラメータ

パラメータ	タイプ	目的	必須 / 任意	デフォルト
calid	文字列をセミコロンで区切ったリスト	プロパティを取得するカレンダーのカレンダー ID のリスト	任意	現在のユーザの calid
fmt-out	文字列	返されるデータの形式 次の形式を指定できる text/calendar text/xml text/js	任意	text/js
id	一意の識別子文字列	セッション ID	必須	該当なし

説明

このコマンドを使用して、指定したカレンダーのカレンダープロパティを取得します。

戻り値

指定したカレンダーの次のプロパティに関する情報を示すページが返されます。

- 相対 ID
- 表示名
- 親カレンダー ID
- タイムゾーン ID
- 読み取りアクセス値 (0 = 読み取り不可、1 = 読み取り可能)
- 書き込みアクセス値 (0 = 書き込み不可、1 = 書き込み可能)
- 文字セット (空の場合、デフォルトは us-ascii)
- 言語 (空の場合、デフォルトは en = 英語)
- cal-master (連絡先情報。通常は、主な所有者の電子メールアドレス)
- 説明
- 最終更新日時

- 作成日
- 主な所有者
- ほかの所有者のリスト (セミコロンで区切られたリスト)
- カテゴリのリスト (セミコロンで区切られたリスト)

エラーコード

カレンダーは存在しているが、ユーザにそのカレンダーの READ アクセス権がない場合、そのカレンダーのインデックス番号に対応する `layer_errno[x]` の値は 1 に設定されます。

カレンダーが見つからない場合、そのカレンダーのインデックス番号に対応する `layer_errno[x]` の値は 2 に設定されます。

カレンダーの取り込みに失敗した場合、そのエラー番号 `errno` は `GET_CALPROPS_FAILED(20)` に設定されます。

例

次の例では、カレンダー `jdoe`、`susan`、`pub`、`john`、および `hasdf` の順で、カレンダープロパティを取得します。

想定する条件は次のとおりです。

- ユーザは、カレンダー `jdoe`、`pub`、および `john` の読み取りアクセス権を持っているが、カレンダー `susan` の読み取りアクセス権は持っていない。
- カレンダー `hasdf` は、データベースに存在しない。

上記の結果は次のようになります。

- `get_calprops.wcap` が呼び出されて、カレンダー `jdoe`、`pub`、および `john` に関するデータが返される。それらのインデックスの `layer_errno[]` の値は、0 である。
- カレンダー `susan` に関するデータは返されない。また、このユーザにカレンダーへのアクセス権がないため、`layer_errno[1]` の値が 1 に設定される。
- カレンダー `hasdf` に関するデータは返されない。カレンダーが存在しないため、`layer_errno[4]` の値が 2 に設定される。
- カレンダーの取り込みに 1 つでも失敗した場合、そのエラー番号は `GET_CALPROPS_FAILED(20)` に設定される。

URL と返されるデータは、次のようになります。

```
http://webcalendarserver/get_calprops.wcap?id=2mu95r5so0hq68ts6q3
&calid=jdoe;susan;pub;john;hasdf
```

```
HTTP/1.0 200
Date:Wed, 16 Jun 2000 22:21:27 GMT
Content-type:text/html; charset=iso-8859-1
Content-length: 1624
```

コマンド

```
Last-modified:Wed, 16 Jun 2000 22:21:27 GMT
Pragma:no-cache
Expires: 0
Cache-Control:no-cache
```

```
<html><head><script>
function color(s) { if (s) document.bgColor=s }
var id='bu9p3eb8x5p2nm0q3'
var userid='jsun'
var calid='jsun'
var errno=new Array()
var timezoneList = new Array()
var calprops = new Array()

function iso(p)
{
if (!p) return null;
var y = p.substring(0,4)
var m = p.substring(4,6)
var d = p.substring(6,8)
var h = p.substring(9,11)
var i = p.substring(11,13)
var s = p.substring(13,15)
return new Date(y, m - 1, d, h, i, s, 0)
}

function
CP(a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11,a12,a13,a14,a15) {
this.calid=a1
this.name=a2
this.parent=a3
this.tzid=a4
this.read=a5
this.write=a6
this.charset=a7
this.lang=a8
this.master=a9
this.desc=a10
this.lastMod=iso(a11)
this.created=iso(a12)
this.primaryOwner=a13
this.owners=a14
this.categories=a15
}

calprops[0] = new CP('jdoe',
'John Doe',
'',
'',
```

```

'0',
'0',
'',
'',
'',
'',
    'Work Calendar for John Doe',
'20000615T222725Z',
'20000615T222725Z',
    'jdoe',
new Array('susan'),
new Array(),
new Array('business'))

layer_errno[0]=0
calprops[1] = new CP('susan')
layer_errno[1]=1
calprops[2] = new CP('pub',
'Public Calendar',
'',
'',
'1',
'1',
'',
'',
'',
'',
    'Public Calendar, Anyone can read and write',
'20000615T222725Z',
    '20000615T222725Z',
'susan',
new Array(),
new Array(),
new Array('group','business'))
layer_errno[2]=0
calprops[3] = new CP('john',
'John Calendar Hosting Server',
'',
'',
'0',
'0',
'',
'',
'',
    'John Project Calendar',
'20000615T222725Z',
'20000615T222725Z',
'susan',
new Array('smith','jones','fred','jdoe'),
new Array(),

```

```

new Array('group','business'))
layer_errno[3]=0
calprops[4] = new CP('hasdf')
layer_errno[4]=2
errno[0]=0
parent.ceCB(window.name)

new Array()

```

get_freebusy

目的

ユーザの予定なし / 予定あり情報を取得します。

パラメータ

表 10-26 は、このコマンドの 5 つのパラメータの一覧です。

表 10-26 get_freebusy のパラメータ

パラメータ	タイプ	目的	必須 / 任意	デフォルト
calid	文字列をセミコロンで区切ったリスト	プロパティを取得するカレンダーのカレンダー ID のリスト	任意	現在のユーザのデフォルトカレンダー
dtstart	ISO8601 DateTime Z 文字列	予定なし / 予定あり検索の開始時刻	必須	該当なし
dtend	ISO8601 DateTime Z 文字列	予定なし / 予定あり検索の終了時刻	必須	該当なし
fmt-out	文字列	返されるデータの形式 次の形式を指定できる text/calendar text/xml text/js	任意	text/js
id	一意の識別子文字列	セッション ID	必須	該当なし

説明

このコマンドは、指定したユーザの予定なし / 予定あり情報を取得します。予定なし / 予定あり情報は、特定の時間帯にユーザの予定があるかどうかを示します。ただし、その予定の内容はわかりません。

エラーコード

何らかの理由でコマンドの実行に失敗した場合、`errno` は `GET_FREEBUSY_FAILED(39)` に設定されます。

例

たとえば、`jdoue` というカレンダーに次のイベントがあります。

```
10:00 ~ 11:00   最初の会議
12:00 ~ 1:00   昼食
3:00 ~ 4:00    次の会議
```

`jdoue` の予定なし / 予定あり時間 (9:00 ~ 6:00) は、次のようになります。

```
9 ~ 10   :   空き時間
10 ~ 11  :   予定あり
11 ~ 12  :   空き時間
12 ~ 1   :   予定あり
1 ~ 3    :   空き時間
3 ~ 4    :   予定あり
4 ~ 6    :   空き時間
```

次の URL では、カレンダー `jdoue` の 2000 年 5 月 1 日 ~ 2000 年 7 月 1 日に検出された予定なし / 予定あり情報が生成されます。

出力は `text/calendar` 形式で返されます。

```
http://webcalendarserver/get_freebusy.wcap?id=2mu95r5so0hq68ts6q3&c
alid=jsun&dtstart=20000501T112233Z&dtend=20000701T112233Z&fmt-out=t
ext/calendar
```

出力は次のようになります。

```
HTTP/1.1 200
Date:Wed, 17 May 2000 01:23:23 GMT
Content-type:text/calendar; charset=UTF-8
Content-length: 1430
Last-modified:Wed, 17 May 2000 01:23:23 GMT
```

コマンド

```
Pragma:no-cache
Expires: 0
Cache-Control:no-cache
Connection:Keep-Alive

BEGIN:VCALENDAR
PRODID:-//iPlanet/Calendar Hosting Server//EN
METHOD:PUBLISH
VERSION:2.0
X-NSCP-CALPROPS-LAST-MODIFIED:20000517T012259Z
X-NSCP-CALPROPS-CREATED:20000517T012259Z
X-NSCP-CALPROPS-READ:999
X-NSCP-CALPROPS-WRITE:999
X-NSCP-CALPROPS-DESCRIPTION:Work Calendar for John Doe
X-NSCP-CALPROPS-RELATIVE-CALID:jdoe
X-NSCP-CALPROPS-NAME:John Doe
X-NSCP-CALPROPS-PRIMARY-OWNER:jdoe
X-NSCP-CALPROPS-OWNERS:susan
X-NSCP-CALPROPS-CATEGORIES:business
X-NSCP-CALPROPS-ACCESS-CONTROL-ENTRY:@^a^S^g
BEGIN:VFREEBUSY
DTSTART:20000501T112233Z
DTEND:20000701T112233Z
FREEBUSY;FBTYPE=FREE:20000501T112233Z/20000518T170000Z
FREEBUSY;FBTYPE=BUSY:20000518T170000Z/20000518T190000Z
FREEBUSY;FBTYPE=FREE:20000518T190000Z/20000525T170000Z
FREEBUSY;FBTYPE=BUSY:20000525T170000Z/20000525T190000Z
FREEBUSY;FBTYPE=FREE:20000525T190000Z/20000601T170000Z
FREEBUSY;FBTYPE=BUSY:20000601T170000Z/20000601T190000Z
FREEBUSY;FBTYPE=FREE:20000601T190000Z/20000608T170000Z
FREEBUSY;FBTYPE=BUSY:20000608T170000Z/20000608T190000Z
FREEBUSY;FBTYPE=FREE:20000608T190000Z/20000615T170000Z
FREEBUSY;FBTYPE=BUSY:20000615T170000Z/20000615T190000Z
FREEBUSY;FBTYPE=FREE:20000615T190000Z/20000622T170000Z
FREEBUSY;FBTYPE=BUSY:20000622T170000Z/20000622T190000Z
FREEBUSY;FBTYPE=FREE:20000622T190000Z/20000629T170000Z
FREEBUSY;FBTYPE=BUSY:20000629T170000Z/20000629T190000Z
FREEBUSY;FBTYPE=FREE:20000629T190000Z/20000701T112233Z
END:VFREEBUSY
X-NSCP-WCAP-ERRNO:0
END:VCALENDAR
```

次の例は、fmt-out に text/xml を指定した場合の出力です。

```
http://webcalendarserver/get_freebusy.wcap?id=2mu95r5so0hq68ts6q3&c
alid=jdoe&dtstart=20000501T112233Z&dtend=20000701T112233Z&fmt-out=t
ext/xml
```

text/xml 形式の出力は次のようになります。

```

HTTP/1.1 200
Date:Wed, 17 May 2000 01:24:58 GMT
Content-type:text/xml; charset=UTF-8
Content-length: 1868
Last-modified:Wed, 17 May 2000 01:24:58 GMT
Pragma:no-cache
Expires: 0
Cache-Control:no-cache
Connection:Keep-Alive

<?xml version="1.0" encoding="UTF-8"?>
<iCalendar>
<iCal version="2.0" prodid="-//iPlanet/Calendar Hosting
Server//EN" METHOD="PUBLISH">
<X-NSCP-CALPROPS-LAST-MODIFIED>20000517T012259Z</X-NSCP-CALPR
PS-LAST-MODIFIED>
<X-NSCP-CALPROPS-CREATED>20000517T012259Z</X-NSCP-CALPROPS-CREATED>
<X-NSCP-CALPROPS-READ>999</X-NSCP-CALPROPS-READ>
<X-NSCP-CALPROPS-WRITE>999</X-NSCP-CALPROPS-WRITE>
<X-NSCP-CALPROPS-DESCRIPTION>Work Calendar for John Doe
</X-NSCP-CALPROPS-DESCRIPTION>
<X-NSCP-CALPROPS-RELATIVE-CALID>jdoe</X-NSCP-CALPROPS-RELATIVE-CALI
D>
<X-NSCP-CALPROPS-NAME>John Doe</X-NSCP-CALPROPS-NAME>
<X-NSCP-CALPROPS-PRIMARY-OWNER>jdoe</X-NSCP-CALPROPS-PRIMARY-OWNER>
<X-NSCP-CALPROPS-OWNERS>susan</X-NSCP-CALPROPS-OWNERS>
<X-NSCP-CALPROPS-CATEGORIES>business</X-NSCP-CALPROPS-CATEGORIES>
<X-NSCP-CALPROPS-ACCESS-CONTROL-ENTRY>@^a^S^g</X-NSCP-CALPROPS-ACCE
SS-CONTROL-ENTRY>

<FREEBUSY>
<START>20000501T112233Z</START>
<END>20000701T112233Z</END>
<FB FBTYPE="FREE">20000501T112233Z/20000518T170000Z</FB>
<FB FBTYPE="BUSY">20000518T170000Z/20000518T190000Z</FB>
<FB FBTYPE="FREE">20000518T190000Z/20000525T170000Z</FB>
<FB FBTYPE="BUSY">20000525T170000Z/20000525T190000Z</FB>
<FB FBTYPE="FREE">20000525T190000Z/20000601T170000Z</FB>
<FB FBTYPE="BUSY">20000601T170000Z/20000601T190000Z</FB>
<FB FBTYPE="FREE">20000601T190000Z/20000608T170000Z</FB>
<FB FBTYPE="BUSY">20000608T170000Z/20000608T190000Z</FB>
<FB FBTYPE="FREE">20000608T190000Z/20000615T170000Z</FB>
<FB FBTYPE="BUSY">20000615T170000Z/20000615T190000Z</FB>
<FB FBTYPE="FREE">20000615T190000Z/20000622T170000Z</FB>
<FB FBTYPE="BUSY">20000622T170000Z/20000622T190000Z</FB>
<FB FBTYPE="FREE">20000622T190000Z/20000629T170000Z</FB>

```

```
<FB FBTYPE="BUSY">20000629T170000Z/20000629T190000Z</FB>
<FB FBTYPE="FREE">20000629T190000Z/20000701T112233Z</FB>
</FREEBUSY>
<X-NSCP-WCAP-ERRNO>0</X-NSCP-WCAP-ERRNO>
</iCal>
</iCalendar>
```

get_guids

目的

GUID (全体に対して一意の識別子) のセットを生成します。

パラメータ

表 10-27 は、このコマンドの 2 つのパラメータの一覧です。

表 10-27 get_guids のパラメータ

パラメータ	タイプ	目的	必須 / 任意	デフォルト
guidCount	整数	取得する GUID の数	任意	1
fmt-out	文字列	返されるデータの形式 次の形式を指定できる text/calendar text/xml text/js	任意	text/js

説明

このコマンドは、指定された数の GUID を返します。このコマンドの呼び出しに、クライアントの認証は必要ありません。

例

URL とその返された出力の例を、3 つの形式で示します。

iCalendar 形式

```
http://webcalendarserver/get_guids.wcap?guidCount=10
&fmt-out=text/calendar
```

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:iPlanet Calendar Server 5.0
X-NSCP-GUID0:e5e4b537465600000b000000c3000000
X-NSCP-GUID1:e5e4b537d47900000c000000c3000000
```

```
X-NSCP-GUID2:e5e4b53796140000d000000c3000000
X-NSCP-GUID3:e5e4b5373d3a00000e000000c3000000
X-NSCP-GUID4:e5e4b537f31400000f000000c3000000
X-NSCP-GUID5:e5e4b5378259000010000000c3000000
X-NSCP-GUID6:e5e4b537b026000011000000c3000000
X-NSCP-GUID7:e5e4b537c263000012000000c3000000
X-NSCP-GUID8:e5e4b537241f000013000000c3000000
X-NSCP-GUID9:e5e4b537e733000014000000c3000000
END:VCALENDAR
```

XML 形式

```
http://webcalendarserver/get_guids.wcap?guidCount=10&fmt-out=text/xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<iCalendar>
<iCal>
<X-NSCP-GUID0>fde4b537d604000015000000c3000000<X-NSCP-GUID0>
<X-NSCP-GUID1>fde4b5379478000016000000c3000000<X-NSCP-GUID1>
<X-NSCP-GUID2>fde4b5372a6d000017000000c3000000<X-NSCP-GUID2>
<X-NSCP-GUID3>fde4b537a355000018000000c3000000<X-NSCP-GUID3>
<X-NSCP-GUID4>fde4b5377768000019000000c3000000<X-NSCP-GUID4>
<X-NSCP-GUID5>fde4b5376e2e00001a000000c3000000<X-NSCP-GUID5>
<X-NSCP-GUID6>fde4b537a07700001b000000c3000000<X-NSCP-GUID6>
<X-NSCP-GUID7>fde4b537744700001c000000c3000000<X-NSCP-GUID7>
<X-NSCP-GUID8>fde4b537ab1f00001d000000c3000000<X-NSCP-GUID8>
<X-NSCP-GUID9>fde4b5371d2200001e000000c3000000<X-NSCP-GUID9>
</iCal>
</iCalendar>
```

JavaScript 形式

この例では、`fmt-out` パラメータが指定されていないため、デフォルトの `text/js` 形式で出力されます。

```
http://webcalendarserver/get_guids.wcap?guidCount=10

<html><head><script>
function color(s) { if (s) document.bgColor=s }
var id='0'
var userid=''
var calid=''
var errno=new Array()
var guid=new Array()
guid[0]=9ee4b5375778000001000000c3000000
guid[1]=9ee4b5376a52000002000000c3000000
guid[2]=9ee4b5375a47000003000000c3000000
guid[3]=9ee4b537bf01000004000000c3000000
guid[4]=9ee4b537e05e000005000000c3000000
```

```

guid[5]=9ee4b537be5a000006000000c3000000
guid[6]=9ee4b537d544000007000000c3000000
guid[7]=9ee4b5372867000008000000c3000000
guid[8]=9ee4b5375731000009000000c3000000
guid[9]=9ee4b537830600000a000000c3000000
parent.ceCB(window.name)
</script></head></html>

```

get_userprefs

目的

現在のユーザのカレンダー設定を取得します。

パラメータ

表 10-28 は、このコマンドの 2 つのパラメータの一覧です。

表 10-28 get_userprefs のパラメータ

パラメータ	タイプ	目的	必須 / 任意	デフォルト
fmt-out	文字列	取得するデータの形式 次の形式を指定できる text/calendar text/xml text/js	任意	text/js
id	一意の識別子 文字列	セッション ID	必須	該当なし
userid	文字列	管理者専用。取得するユーザの設定を指定する	任意	該当なし

説明

このコマンドは、現在のユーザのすべてのカレンダー設定およびこのユーザに関連している次のサーバ設定を取得します。

- allowchangeapassword。ユーザはパスワードを変更できます。
- allowcreatecalendars。ユーザはカレンダーを作成できます。
- allowdeletecalendars。ユーザはカレンダーを削除できます。
- allowpublicwritablecalendars。ユーザは、他のユーザが書き込みできる公開カレンダーを所有できます。
- validateowners。この値を 1 に設定すると、ディレクトリのメカニズムが LDAP か CSAPI 互換のユーザメカニズムかに関わらず、そのディレクトリ内にカレンダーの所有者が存在することを確約するため検証がサーバによって行われます。

- allowsetprefs. この値が 1 に設定されている場合は、set_userprefs.wcap を使ってユーザ設定を変更できます。

userid パラメータを使用するには、ics.conf ファイルで、サーバの service.admin.calmaster.wcap.allowgetmodifyuserprefs が「yes」に設定されている必要があります。また、要求を送信するユーザは、login.wcap コマンドを使用して管理者としてログインしておく必要があります。

サーバ設定の詳細は、『管理者ガイド』を参照してください。

例

次の URL では、現在のユーザの設定を取得します。

```
http://webcalendarserver/get_userprefs.wcap?id=b5q2o8ve2rk02nv9t6
```

次のデータが返されます。

```
<html><head><script>
function A (name, readonly)
{this.name=name
this.readonly=readonly
this.vals=new Array (A.arguments.length - 2)
for (var i = 2; i < A.arguments.length; i++)
this.vals[i - 2]=A.arguments[i]
}
var usrattrval=new Array()
usrattrval[0]=new A('cn',true,'John Doe')
usrattrval[1]=new A('givenName',true,'John')
usrattrval[2]=new A('mail',true,'jdoe@mailserver')
usrattrval[3]=new A('preferredlanguage',false)
usrattrval[4]=new A('sn',true,'Doe')
usrattrval[5]=new A('cebgcolor',false,'black')
usrattrval[6]=new A('ceFgcolor',false,'green')
usrattrval[7]=new A('ceBFgcolor',false,'black')
function M(name, flags, msgs, size)
{this.name=name
this.msgs=msgs
this.flags=flags
this.size=size
}
var flags=new Array ('//noinferiors', '//hasnochildren',
 '//haschildren', '//noselect')
var flag=new Array(flags.length)
flag['//noinferiors']=1
flag['//hasnochildren']=2
flag['//haschildren']=4
flag['//noselect']=8
var allowchangepassword="no"
var allowcreatecalendars="yes"
```

```

var allowdeletecalendars="yes"
var allowexpungecalendar="yes"
var errno=new Array()
errno[0]=0
var errstr=''
</script></head> </script></head>
<body bgcolor=#FFFFFF>
<br><form action=set_userprefs.wcap method=post name=form>
cmd<input name=cmd>
old<input name=oldPassword>
new<input name=newPassword>
<input name=add_attrs>
<input name=del_attrs>
<input name=set_attrs>
<input name=id value='e2np3w9o0v6k0u9v9b' >
<input name=bgcolor>
<input name=security>
<input type=submit>
</form>
</body>
<script>var form=document.form
parent.cfgCB()</script>

```

import

目的

ファイルからイベントおよび仕事をカレンダーにインポートします。

パラメータ

表 10-29 は、このコマンドの 5 つのパラメータの一覧です。

表 10-29 import のパラメータ

パラメータ	タイプ	目的	必須 / 任意	デフォルト
calid	文字列	イベントのインポート先のカレンダー ID	必須	該当なし
content-in	文字列	入力データのコンテンツタイプ。次の値を指定できる text/calendar text/xml	必須	該当なし
dtend	ISO8601 DateTime Z 文字列	インポートするイベントおよび仕事の終了日時 値が 0 の場合は、ファイル内の最初の日付から最後の日付までのコンポーネントがすべてインポートされる	任意	0

表 10-29 import のパラメータ (続き)

パラメータ	タイプ	目的	必須 / 任意	デフォルト
dtstart	ISO8601 DateTime Z 文字列	インポートするイベントおよび仕事の開始 日時 値が 0 の場合は、ファイル内の最も古い日時 から終了日時までのコンポーネントがすべて インポートされる	任意	0
id	一意の識別子 文字列	セッション ID。カレンダーが公開されている 場合を除き、このパラメータは必須	必須	該当なし

説明

このコマンドを使用して、以前に export コマンドを使ってファイルにエクスポートしたカレンダーから、指定したイベントおよび仕事をインポートします。インポートするファイルの MIME コンテンツタイプを、content-in パラメータに指定する必要があります。

開始日または終了日を指定しない場合、または dtstart および dtend の値として 0 を渡した場合は、指定したカレンダーにファイル内のすべてのイベントおよび仕事が追加されます。開始日および終了日を指定すると、その期間に該当するファイル内のイベントおよび仕事だけがインポートされます。開始日および終了日は、日付と時刻の最後に Z を付けた UTC 時間として指定します。

このコマンドは、HTTP GET メッセージを使用できるほかのコマンドと異なり、HTTP POST メッセージを使用する必要があります。エクスポートしたイベントおよび仕事が含まれるファイルの内容を、POST メッセージに添付します。このファイルは、iCalendar (.ics) または XML (.xml) 形式でなければなりません。

例

次の POST メッセージは、import コマンドを使用して、添付した iCalendar ファイルをカレンダー jdoe にインポートします (セッション ID が必要です)。

```
POST /import.wcap?id=t95qm0n0es3bo35r&calid=jdoe&dtstart=0&dtend=0
Content-type:multipart/form-data;
boundary=-----33111928916708
Content-Length: 679
-----33111928916708
Content-Disposition:form-data; name="Upload";
filename="C:\TEMP\icall.ics"
BEGIN:VCALENDAR
BEGIN:VEVENT
DTSTART:20000105T100000
DTEND:20000105T110000
```

```
DTSTAMP:20000104T120000
CREATED:20000105T110000Z
LAST-MODIFIED:20000104T120000Z
SUMMARY:Weekly QA Meeting
UID:random-uid001
END:VEVENT
BEGIN:VEVENT
DTSTART:20000106T100000
DTEND:20000106T110000
DTSTAMP:20000104T120000
CREATED:20000105T110000Z
LAST-MODIFIED:20000104T120000Z
SUMMARY:Weekly QA Meeting 2
UID:random-uid002
END:VEVENT
END:VCALENDAR
-----33111928916708--
```

次の HTML フォームでは、指定されたファイルを添付した POST メッセージが作成されます。

```
<FORM METHOD=POST ENCTYPE="multipart/form-data"
ACTION="http://webcalendarserver:12345/import.wcap?id=t95qm0n0es3bo
35r&calid=jdoe&dtstart=0&dtend=0&content-in=text/calendar">
<ol>
<li>file to import:<input type="file" accept="text" name="Upload">
</li>
<li>Press Import Now:<input type="submit" value="Import Now"></li>
</ol>
</FORM>
```

login

目的

指定したユーザを認証します。

パラメータ

表 10-30 は、このコマンドの 5 つのパラメータの一覧です。

表 10-30 login のパラメータ

パラメータ	タイプ	目的	必須 / 任意	デフォルト
fmt-out	文字列	取得するデータの形式 次の形式を指定できる text/calendar text/xml text/js text/calendar または text/xml の場合、refresh パラメータは自動的に 1 に設定される	任意	text/js
lang	enum	ユーザの好みの言語	任意	NULL
password	文字列	ユーザのパスワード	任意	該当なし
refresh	整数 (0、1)	新しいセッション ID だけを取得するか、またはセッション ID およびその他のパラメータを取得するかどうかを示すブール値 1 = 新しいセッション ID だけを返す 0 = セッション ID およびその他のパラメータを返す	任意	0
user	文字列	ユーザの名前	任意	NULL

説明

このコマンドは、指定したユーザを、ユーザ名とパスワードの規則に基づいて認証し、iPlanet Calendar Server にログインさせます。

ユーザ名は、サーバがユーザを一意に識別するためのプレーンテキスト文字列です。このユーザ名は、たとえば、ユーザの電子メールアドレスでもかまいません。パスワードもプレーンテキストです。

認証

内部認証は、デフォルトの LDAP 認証を使用するか、独自の CSAPI プラグイン経由で既存のユーザ認証方式に接続することによって行われます。CSAPI 認証の詳細は、52 ページの「csIAuthentication」を参照してください。プロキシ認証 SDK の詳細は、第 6 章の概要および第 7 章の API リファレンスを参照してください。

ユーザの認証に失敗した場合は、ログインに失敗したことを示すエラーメッセージがログインウィンドウに表示されます。

例

たとえば、次の URL は、ユーザ `jdoe` としてログインします。

```
http://webcalendarserver/login.wcap?user=jdoe&password=mypword
```

戻り値

`refresh` パラメータを渡さないで、`fmt-out` に `text/js` を指定した場合、`refresh` パラメータのデフォルト値は `0` になります。`fmt-out` に `text/calendar` または `text/xml` を指定した場合、そのデフォルト値は `1` になります。

`refresh` パラメータに `1` を指定した場合、返されるページには、次のようなセッション ID が 1 行含まれます。

```
var id='bu9p3eb8x5p2nm0q3'
```

これは、非公開カレンダーにアクセスする場合に、さまざまな `WCAP` コマンドに渡す値です。また、この値は `logout` などの特定のコマンドに必要なパラメータ値です。

`refresh` パラメータに `0` を指定すると、iPlanet Calendar Server のユーザインタフェースへのエントリページの場所を含んだ JavaScript 出力が返されます。

パラメータを `0` に設定すると、`login` コマンドはデフォルトの `html` ファイルを含むすべての値を返します。

```
HTTP/1.0 302 OK
Date:Tue, 11 May 2000 22:38:33 GMT
Pragma:no-cache
Expires: 0
Cache-Control:no-cache
Content-Length: 0
Last-modified:Tue, 11 May 2000 22:38:33 GMT
Location:
http://webcalendarserver/en/main.html?id=er6en05tv6n3bv9&lang=en
  &host=http://webcalendarserver/

<html><head><script>
function color(s) { if (s) document.bgColor=s }
var id='er6en05tv6n3bv9'
var userid='jdoe'
var calid='jdoe'
var errno=new Array()
var errstr=''
</script></head>
<body bgcolor='9999CC' onLoad=parent.ceCB(window.name)>
```

logout

目的

現在のユーザのセッションを終了します。

パラメータ

表 10-31 は、このコマンドの 2 つのパラメータの一覧です。

表 10-31 logout のパラメータ

パラメータ	タイプ	目的	必須 / 任意	デフォルト
fmt-out	文字列	取得するデータの形式 次の形式を指定できる text/calendar text/xml text/js	任意	text/js
id	一意の識別子文字列	セッション ID	必須	該当なし

説明

このコマンドは、指定したユーザのセッションを終了し、セッションテーブル上のユーザのセッションインスタンスを削除します。ユーザはログイン画面に戻ります。

このコマンドを使用した URL の例を次に示します。

```
http://webcalendarserver/logout.wcap?id=bu9p3eb8x5p2nm0q3
```

ping

目的

カレンダーサーバがアクティブかどうかを調べます。

パラメータ

このコマンドにパラメータは必要ありません。

説明

このコマンドは、サーバの応答だけを示す最小限の HTML ページを返します。

このコマンドを使用できるのは、管理権限を持つユーザだけです。

戻り値

次の例では、管理者の `userid` と `calid` は、両方とも `adminX` です。

```
HTTP/1.0 200
Date:Thu, 03 Jun 2000 21:31:42 GMT
Content-type:text/html; charset=iso-8859-1
Content-length: 190
Last-modified:Thu, 03 Jun 2000 21:31:42 GMT
Pragma:no-cache
Expires: 0
Cache-Control:no-cache

<html><head><script>
function color(s) { if (s) document.bgColor=s }

var id='bb5rt6eb5pu9v9w9'
var userid='adminX'
var calid='adminX'
var errno=new Array()
parent.ceCB(window.name)
</script></head></html>
```

search_calprops

目的

カレンダープロパティを検索します

パラメータ

表 10-32 は、このコマンドの 7 つのパラメータの一覧です。

表 10-32 search_calprops のパラメータ

パラメータ	タイプ	目的	必須 / 任意	デフォルト
calid	整数 (0, 1)	calid プロパティを検索するかどうかを示すブール値 1 = calid プロパティを検索する 0 = 検索しない	任意	0 (primaryOwner と name が両方とも 0 の場合を除く)
id	一意の識別子文字列	セッション ID	必須	該当なし
maxResults	整数	返される結果の最大数	任意	200

表 10-32 search_calprops のパラメータ (続き)

パラメータ	タイプ	目的	必須 / 任意	デフォルト
name	整数 (0, 1)	name プロパティを検索するかどうかを示すブール値 1 = name プロパティを検索する 0 = 検索しない	任意	0
primaryOwner	整数 (0, 1)	primaryOwner プロパティを検索するかどうかを示すブール値 1 = primaryOwner プロパティを検索する 0 = 検索しない	任意	0
searchOpts	整数 0、1、2、3	検索の実行方法。次の値を指定できる 0 = CONTAINS 1 = BEGINS_WITH 2 = ENDS_WITH 3 = EXACT	任意	0
search-string	文字列	カレンダー内で検索する文字列	必須	該当なし

説明

このコマンドは、searchOpts で指定したクエリタイプを使用して、カレンダーを検索します。指定したプロパティ (primaryOwner、calid、name) の文字列が、指定した searchOpts の検索方法に基づいて、search-string と一致するすべてのカレンダーのカレンダープロパティが返されます。ただし、返されるプロパティ数は、最大で maxResults に指定された数までです。

検索するプロパティ

このコマンドは、次の 3 つのプロパティの 1 つと一致する文字列を検索します。

- calid。カレンダーの一意的識別子
- name。カレンダーの共通名 (テキスト)
- primaryOwner。カレンダーの主な所有者。

特定のプロパティの値を検索するには、そのパラメータを 1 に設定します。

primaryOwner および name を 0 に設定すると、calid はデフォルトで 1 になります。また、calid パラメータの文字列にかかわらず、search-string が calid と見なされます。

検索オプション

次の 4 つの検索オプションがあります。

- search-string を含むカレンダープロパティを返す (CONTAINS)
- search-string で始まるカレンダープロパティを返す (BEGINS_WITH)
- search-string で終わるカレンダープロパティを返す (ENDS_WITH)
- search-string と完全に一致するカレンダープロパティを返す (EXACT)

例

次の URL の例では、すべてのカレンダーから、文字列「susan」を含む (searchOpts=0)、主な所有者のプロパティ (primaryOwner=1) が検索されます。

```
http://webcalendarserver/search_calprops.wcap?id=b2nehr3eq6bh5s
&search-string=susan&primaryOwner=1&searchOpts=0&maxResults=50
```

返されたデータの calsize 変数には、出力として返されたカレンダーの数が指定されています。

次のデータはこの URL の結果です。

```
HTTP/1.0 200
Date:Fri, 30 Apr 2000 23:49:19 GMT
Content-type:text/html; charset=iso-8859-1
Content-length: 10113
Last-modified:Fri, 30 Apr 2000 23:49:19 GMT
Pragma:no-cache
Expires: 0
Cache-Control:no-cache

<html><head><script>
function color(s) { if (s) document.bgColor=s }

var id='bu9p3eb8x5p2nm0q3'
var userid='jdoe'
var calid='jdoe'
var errno=new Array()
var timezoneList = new Array()
var calprops = new Array()
var layer_errno = new Array()

function iso(p) {
if (!p) return null;
var y = p.substring(0,4)
var m = p.substring(4,6)
var d = p.substring(6,8)
var h = p.substring(9,11)
var i = p.substring(11,13)
var s = p.substring(13,15)
return new Date(y, m - 1, d, h, i, s, 0)
}
function
```



```

CP(a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11,a12,a13,a14,a15)
{
this.calid=a1
this.name=a2
this.parent=a3
this.tzid=a4
this.read=a5
this.write=a6
this.charset=a7
this.lang=a8
this.master=a9
this.desc=a10
this.lastMod=iso(a11)
this.created=iso(a12)
this.primaryOwner=a13
this.owners=a14
this.categories=a15
}
calprops[0] = new CP('hockey',
'',
'',
'',
'1',
'1',
'',
'',
'',
'',
'',
'',
'20000615T222725Z',
'20000615T222725Z',
'susan',
new Array(),
new Array(),
new Array('Sports'))
var calsize=1
var maxReturnSize=50
errno[0]=0
parent.ceCB(window.name)
</script></head></html>

```

set_calprops

目的

カレンダーのカレンダープロパティを設定します。

パラメータ

表 10-33 は、このコマンドの 15 個のパラメータの一覧です。

表 10-33 set_calprops のパラメータ

パラメータ	タイプ	目的	必須 / 任意	デフォルト
acl	文字列	アクセスコントロールエントリの新しい値を指定した文字列を、セミコロンで区切ったリスト	任意	""
cal	符号化文字列	復号化するパラメータのリスト このパラメータの複数のインスタンスを指定できる	任意	該当なし
calid	文字列	変更するカレンダーの識別子	必須	該当なし
categories	文字列	カレンダーが属する新しいカテゴリを指定した文字列を、セミコロンで区切ったリスト	任意	該当なし
charset	文字列	カレンダーの文字セット	任意	該当なし
description	文字列	カレンダーの説明文	任意	該当なし
fmt-out	文字列	取得するデータの形式 次の形式を指定できる text/calendar text/xml text/js	任意	text/js
id	一意の識別子文字列	セッション ID	必須	該当なし
lang	文字列	カレンダーの言語	任意	該当なし
master	文字列	カレンダーの連絡先電子メールアドレス	任意	該当なし
multiple	整数	これらの設定を設定するカレンダーの数	任意	0
name	文字列	カレンダーの新しいテキスト名	任意	該当なし
owners	文字列	主な所有者以外の新しい所有者を指定した文字列を、セミコロンで区切ったリスト	任意	該当なし

表 10-33 set_calprops のパラメータ (続き)

パラメータ	タイプ	目的	必須 / 任意	デフォルト
read	整数	このパラメータは、バージョン 5.0 では機能しない。2.x との下位互換性を保持するために残されている。このパラメータの代わりに、acl パラメータを使用してください カレンダーの新しい読み取りアクセス値。 次の値を指定できる 0 PRIVATE 1 PUBLIC 4 PRIMARY_OWNER_ONLY	任意	該当なし
tzid	文字列	このカレンダーの新しいタイムゾーン ID	任意	""
write	整数	このパラメータは、バージョン 5.0 では機能しない。2.x との下位互換性を保持するために残されている。このパラメータの代わりに、acl パラメータを使用してください カレンダーの新しい書き込みアクセス値。 次の値を指定できる 0 PRIVATE 1 PUBLIC 4 PRIMARY_OWNER_ONLY	任意	該当なし

説明

このコマンドは、指定されたパラメータ値だけを変更します。つまり、2.x のような置換コマンドではなく、更新コマンドとして機能します。すべてのパラメータをコマンドに指定する必要はありません。変更するパラメータだけを指定します。カレンダープロパティとは、カレンダーの名前、読み取り / 書き込みアクセス権の値 (acl パラメータ)、所有者のリスト、およびカテゴリのリストなど、カレンダー固有の状態のことです。

バージョン 5.0 では、read パラメータおよび write パラメータは使用できません。この機能は、acl パラメータに置き換えられています。これらのパラメータは 2.x との下位互換性を保持するために残されています。

set_calprops を使用して次の操作を行います。

- カレンダーの名前を変更する
- カレンダーの所有者を変更する
- カレンダーのカテゴリを変更する

- カレンダーのイベントの読み取りアクセス権を変更する
- カレンダーのイベントの書き込みアクセス権を変更する
- カレンダーの説明文を変更する
- カレンダーの文字セットを変更する
- カレンダーの言語を変更する
- カレンダーの連絡先電子メールアドレスを変更する
- カレンダーのタイムゾーン ID を変更する

単一カレンダーの例

次の例は、カレンダープロパティを設定するサンプル URL です。calid パラメータが必要です。

```
http://webcalendarserver?set_calprops.wcap?id=dfasdfzd3ds&calid=jdoe&categories=business;meeting&name=John%39s%32Calendar
```

複数のカレンダーの例

複数のカレンダーのプロパティを一度に設定するには、multiple パラメータにカレンダー数を設定して、設定するカレンダーごとに cal パラメータを渡します。cal パラメータには、個々のカレンダーのプロパティパラメータをすべて符号化して指定します。この文字列内では、すべての特殊文字は、パーセント記号 (%) と特殊文字用の 16 進 ASCII コードの組み合わせに置き換えてください。一般的な特殊文字の ASCII 16 進コードは、次のとおりです。

文字	コード
=	%3D
&	%26
"	%22

たとえば、次の URL では、ID が xxxx、yyyy、および zzzz である 3 つのカレンダーの説明文が、それぞれ X-Calendar、Y-Calendar、および Z-Calendar に変更されます。

```
http://webcalendarserver?id=fasdfzd3ds
&multiple=3
&cal=calid%3Dxxxx%26description%3DX-Calendar
&cal=calid%3Dyyyy%26description%3DY-Calendar
&cal=calid%3Dzzzz%26description%3DZ-Calendar
```

この URL は、次の 3 つの URL と等価です。

```
http://webcalendarserver?id=fasdfzd3ds&calid=xxxx&desc=X-Calendar
```

`http://webcalendarserver?id=fasdfzd3ds&calid=yyyy&desc=Y-Calendar`

`http://webcalendarserver?id=fasdfzd3ds&calid=zzzz&desc=Z-Calendar`

この例では、multiple パラメータが 3 に設定されているため、cal パラメータのインスタンスが 3 個存在します。各 cal パラメータの値は、符号化されたパラメータとその値のリストです。各 cal パラメータは、サーバによって復号化され、該当するプロパティが設定されます。

アクセスコントロールエントリ

この章の初めにある「共通のトピック」の 156 ページの「アクセスコントロールエントリ (ACE)」を参照してください。

予定なし / 予定ありに限定したアクセス

この章の初めにある「共通のトピック」の 165 ページの「予定なし / 予定ありに限定したアクセス」を参照してください。

ほかの言語または文字セットの選択

この章の最初にある「共通のトピック」の 159 ページの「ほかの言語または文字セットの選択」を参照してください。

set_userprefs

目的

セッションの設定またはパスワードを変更します。

パラメータ

表 10-34 は、このコマンドの 5 つのパラメータの一覧です。

表 10-34 set_userprefs のパラメータ

パラメータ	タイプ	目的	必須 / 任意	デフォルト
add_attrs	文字列	新しい設定を追加する	任意	該当なし
convertCalid	整数 (0、1)	このパラメータを 1 に設定し、icsSet または icsSubscribed を設定すると、calid を登録するときに、文字「^」が「:」に変換される 0 に設定すると、そのパラメータは無視される	任意	0
del_attrs	文字列	既存の設定を削除する	任意	該当なし

表 10-34 set_userprefs のパラメータ (続き)

パラメータ	タイプ	目的	必須 / 任意	デフォルト
fmt-out	文字列	取得するデータの形式 次の形式を指定できる text/calendar text/xml text/js	必須	text/js
id	一意の識別子文字列	セッション ID	必須	該当なし
set_attrs	文字列	設定値を変更する	任意	該当なし
userid	文字列	管理者専用。ユーザ設定を行う対象ユーザのユーザ ID を示す	任意	該当なし

説明

このコマンドは、現在のユーザ設定を変更します。ユーザのパスワードを、LDAP を介して変更することもできます。

パラメータ `convertCalid` は、カレンダーの予約購読リスト (`icsSubscribed`) またはグループの予約購読リスト (`icsSet`) を設定する場合にだけ使用します。着信したコマンドの `calid` では、コロン「:」がキャレット「^」に置き換えられていなければなりません。たとえば、`calid` が `jdoue:personal` の場合、コマンドが正しく機能するには、WCAP は `jdoue^personal` として受け取る必要があります。

`convertCalid` の値が 1 になっている場合は、WCAP によって「^」が「:」に戻されます。`convertCalid` の値が 0 の場合、この変換は行われません。

管理者がログインしているときに、`ics.conf` ファイルの `service.admin.calmaster.wcap.allowgetmodifyuserprefs` が「yes」に設定されている場合は、`userid` パラメータには設定対象のユーザ ID を指定します。

戻り値

`get_userprefs` のテキストを返します。

例

たとえば、次の URL では、新しい設定 `ceBgcolor` がカレンダーに追加され、その値が `black` に設定されます。

```
http://webcalendarserver/set_userprefs.wcap?id=b5q2o8ve2rk02nv9t6
&add_attrs=ceBgcolor=black
```

次の URL では、カレンダー設定 `ceBgcolor` がユーザ設定から削除されます。

```
http://webcalendarserver/set_userprefs.wcap?id=b5q2o8ve2rk02nv9t6
&del_attrs=ceBgcolor
```

次の URL では、カレンダー設定 `ceBgcolor` の値が `white` に変更されます。

```
http://webcalendarserver/set_useprefs.wcap?id=b5q2o8ve2rk02nv9t6
&set_attrs=ceBgcolor=white
```

次の URL を実行すると、ログインしている管理者が、ユーザ `jdoe` のカレンダー設定 `ceBgcolor` の値を `black` に変更できます。

```
http://webcalendarserver/set_userprefs.wcap?id=b5q2o8ve2rk02nv9t6&u
serid=jdoe&set_attrs=ceBgcolor=black
```

storeevents

目的

イベントをカレンダーに追加します。

パラメータ

表 10-35 は、このコマンドの 43 個のパラメータの一覧です。

表 10-35 storeevents のパラメータ

パラメータ	タイプ	目的	必須 / 任意	デフォルト
alarmAudio	ISO8601 Date Time Z 文字列	オーディオアラームを再生する時刻	任意	該当なし
alarmEmails	電子メールアドレスをセミコロ ンで区切ったリ スト	イベントのアラーム通知の受信者	任意	該当なし
alarmFlashing	ISO8601 Date Time Z 文字列	フラッシュアラームを実行する時刻	任意	該当なし
alarmPopup	ISO8601 Date Time Z 文字列	ダイアログアラームを表示する時刻	任意	該当なし
alarmStart	ISO8601 DateTime Z 文字列	イベントのアラーム通知を送信する 時刻	任意	該当なし
attachments	文字列をセミコ ロンで区切った リスト	iCalendar と相互運用するときのパラ メータ。 文字列は URL	任意	該当なし
attendees	文字列をセミコ ロンで区切った リスト	イベントの参加者	任意	該当なし

表 10-35 storeevents のパラメータ (続き)

パラメータ	タイプ	目的	必須 / 任意	デフォルト
brief	整数 (0、1)	JavaScript の簡略出力を出力するかどうかを示すブール値 出力形式 (fmt-out) が JavaScript (text/js) の場合にだけ適用される 1 = 簡略 0 = 完全	任意	0
calid	文字列	イベントの登録先のカレンダー ID	必須	該当なし
categories	文字列をセミコロンで区切ったリスト	イベントのカテゴリ	任意	該当なし
contacts	文字列をセミコロンで区切ったリスト	イベントの連絡先	任意	該当なし
desc	文字列	イベントの目的の説明文。任意の長さの文字列。このパラメータを渡さない場合、desc は summary 値に設定される 文字列にスペースを含める場合は、コード %20 を使用する	任意	summary パラメータ の値
dtend	ISO8601 DateTime Z 文字列	イベントの終了日時	任意	該当なし
dtstart	ISO8601 DateTime Z 文字列	イベントの開始日時 イベントを変更するときには必要なし イベントを作成するときには必要	任意 必須	該当なし
duration	ISO8601 継続時間文字列	イベントの継続時間。イベントに duration と dtend が指定されている場合、duration は無視される	任意	該当なし
exdates	ISO8601 DateTime Z 文字列をセミコロンで区切ったリスト	除外するイベント繰り返し日付	任意	該当なし

表 10-35 storeevents のパラメータ (続き)

パラメータ	タイプ	目的	必須 / 任意	デフォルト
exrules	文字列をセミコロンで区切ったリスト	除外するイベント繰り返し規則。繰り返し規則の文字列をセミコロンで区切ったリスト 各規則の値は、二重引用符で囲むことが必要。170 ページの「繰り返し処理」を参照	任意	該当なし
fetch	整数 (0、1)	新しく登録された仕事を取り込んで返すかどうかを示すブール値 1 = 新しく登録された仕事を取得して返す 0 = 返さない	任意	0
fmt-out	文字列	取得するデータの形式 次の形式を指定できる text/calendar text/xml text/js	任意	text/js
geo	セミコロンで区切られた 2 つの実数	イベントの地理的位置を表す 2 つの実数 (緯度と経度) を、セミコロンで区切った文字列 たとえば、37.31;-123.2 のように指定する	任意	0;0
icsClass	文字列	イベントのクラス 次の値を指定できる PUBLIC PRIVATE CONFIDENTIAL	任意	PRIVATE
icsUrl	文字列	イベントの URL	任意	" "
id	一意の識別子文字列	セッション ID	必須	該当なし
isAllDay	整数 (0、1)	イベントが終日イベントかどうかを示すブール値 1 = 終日イベント 0 = 終日イベントではない	任意	0
language	文字列	イベントの言語 (例 : 「en」、「fr」、「de」)	任意	該当なし

表 10-35 storeevents のパラメータ (続き)

パラメータ	タイプ	目的	必須 / 任意	デフォルト
location	文字列	イベントの場所	任意	" "
method	整数 (1、2、4、8、16、32)	グループスケジューリングの ITIP メソッド 1 = PUBLISH (開催者専用) 2 = REQUEST (開催者専用) 4 = REPLY (参加者専用) 8 = CANCEL (開催者専用) 16 = MOVE 32 = COUNTER (参加者専用)	任意	1 (PUBLISH)
mod	整数	登録または変更する繰り返しを指定する イベントを作成するときは必要なし イベントを変更するときに必要 次の値を指定できる 1 = THISINSTANCE 2 = THISANDFUTURE 3 = THISANDPRIOR 4 = THISANDALL	任意 必須	該当なし
notify	整数 (0、1)	このパラメータは 5.0 では使用できない。2.x との下位互換性を保持するために残されている。GSE (Group Scheduling Engine) モジュールでは、すべての電子メール通知が処理される 参加者にイベントの変更を通知するかどうかを示すブール値 1 = 参加者に通知する 0 = 参加者に通知しない	任意	0
orgEmail	電子メールアドレス	イベントの連絡先 (通常、開催者) の電子メールアドレス	任意	該当なし
orgUID	userid	開催者の userid	任意	該当なし
priority	整数 (0 ~ 9)	イベントの優先順位 0 = 最も低い優先順位 9 = 最も高い優先順位	任意	0

表 10-35 storeevents のパラメータ (続き)

パラメータ	タイプ	目的	必須 / 任意	デフォルト
rchange	整数 (0、1)	繰り返しイベントを継続するかどうかを示すブール値 1 = 継続する 0 = 継続しない	任意	1
rdates	ISO8601 DateTime Z 文字列をセミコロンの区切ったリスト	イベントの繰り返し日付	任意	該当なし
relatedTos	二重引用符で囲んだ文字列をセミコロンの区切ったリスト	このイベントに関連するほかのイベント	任意	該当なし
resources	文字列をセミコロンの区切ったリスト	イベントに関連付けられたリソース	任意	該当なし
rid	ISO8601 DateTime Z 文字列	イベントの繰り返し ID イベントを作成するときには必要なし イベントを変更するときには必要	任意 必須	該当なし
rrules	文字列をセミコロンの区切ったリスト	イベントの繰り返し規則。繰り返し規則の文字列をセミコロンの区切ったリスト 各規則の値は、二重引用符で囲む必要がある。170 ページの「繰り返し処理」を参照	任意	該当なし
seq	整数	イベントのシーケンス番号	任意	0
status	整数	イベントの状態コード。次の値を指定できる 0 CONFIRMED 1 CANCELLED 2 TENTATIVE 3 NEEDS_ACTION 4 COMPLETED 5 IN_PROCESS 6 DRAFT 7 FINAL	任意	該当なし

表 10-35 storeevents のパラメータ (続き)

パラメータ	タイプ	目的	必須 / 任意	デフォルト
summary	文字列	イベントの要約。任意の長さの文字列		
		新しいイベントを作成するときに必要	必須	該当なし
		イベントを変更するときには必要なし	任意	default summary
		文字列にスペースを含める場合は、コード %20 を使用する		
tzid	タイムゾーン ID 文字列 「America/Los_ Angeles」など	タイムゾーン。サーバに渡された、またはサーバから返されたあらゆる日付の解釈に使用される	任意	"GMT"
		デフォルトは、世界協定時刻 (UTC または Z 形式)		
uid	文字列	登録されるイベントの一意の識別子		
		新しいイベントの場合は、システムによって生成される	任意	該当なし
		イベントを変更するときに必要な	必須	default uid

説明

このコマンドは、指定された属性を持つイベントの作成、変更、およびデータベース内の指定されたカレンダーへの登録を行います。

このコマンドは、rrules、exrules、rid、mod、および rchange パラメータの指定に従って、繰り返しを作成および登録します。170 ページの「繰り返し処理」を参照してください。

notify 値が 1 の場合、IMIP PUBLISH メッセージがイベントのすべての参加者に送信されます。

イベントの言語を指定するには、language パラメータを使用します。使用可能な言語コードのリストについては、159 ページの「ほかの言語または文字セットの選択」を参照してください。

カレンダーサーバでは、ファイルの添付はサポートされていません。attachments パラメータは、iCalendar と相互運用するときに使います。カレンダーサーバでは機能しません。

戻り値

fmt-out パラメータを text/calendar または text/xml に設定すると、次のように HTML コメントの中にエラー値だけが返されます。

```
<!-- store_errno="0" -->
```

fmt-out パラメータを text/js に設定するか、設定しなかった (このパラメータのデフォルトは text/js) 場合、このコマンドは、fetchcomponents_by_range.wcap コマンドが返した JavaScript を返します。

エラーコード

このコマンドを使用してリンクされているイベントを変更することはできません。変更しようとする、コマンドの実行が失敗し、errno 配列に CANNOT_MODIFY_LINKED_EVENTS (31) が返されます。

すでに予定が設定されている時間枠にイベントを登録しようとする (二重予約)、コマンドの実行が失敗し、エラー STORE_FAILED_DOUBLE_BOOKED (40) が返されます。

必要なパラメータ

このコマンドを使って、新しいイベントの作成や、既存のイベントの変更を行います。必要なパラメータは、実行する内容によって次のように異なります。

- 新しいイベントを作成する場合は、次の 2 つのパラメータだけが必要になります。
 - dtstart
 - summary

その他のパラメータはすべてオプションです。uid は、サーバによって生成されます。

- 既存のイベントを変更する場合は、次の 3 つのパラメータが必要です。
 - uid
 - rid
 - mod

その他のパラメータはすべてオプションです。パラメータを指定しなかった場合は、プロパティの以前の値が保持されます。

継続時間

duration は、次のように ISO8601 形式で指定します。

- P1Y2M3DT1H30M10S は、1 年、2 か月、3 日、1 時間、30 分、10 秒の継続時間を表します。
- PT1H30M は、1 時間 30 分の継続時間を表します。
- P1D は、1 日の継続時間を表します。
- PT15M は、15 分の継続時間を表します。

文字列に含まれる文字 T で、日付情報 (年、月、日) と時刻情報 (時、分、秒) を区切ります。

ヒント 終了日時 (dtend) は、duration よりも優先されます。duration と dtend の両方を指定すると、duration は無視されます。

例

たとえば、次の URL では、storeevents.wcap を呼び出して、イベントをカレンダー john に登録します。

```
http://webcalendarserver/storeevents.wcap?id=3423423asdfasf
&calid=john&dtstart=20000101T103000&dtend=20000101T113000&uid=001
&summary=new%20year%20event
```

この例を実行すると、iCalendar データベースに次のエントリが作成されます。

```
BEGIN:VEVENT
DTSTART:20000101T183000Z
DTEND:20000101T193000Z
UID:001
SUMMARY:new year event
END:VEVENT
```

グループスケジューリング - attendee パラメータ

グループスケジュールイベントを作成するとき最も重要なパラメータは、attendee と method です。attendee パラメータは、参加者エントリをセミコロンで区切ったリストです。次に、参加者エントリについて説明します。

各参加者エントリには、イベントの参加状況や返答が必要かどうかなど、いくつかのパラメータが含まれます。これらのすべてのパラメータには、iCalendar の仕様 (RFC 2445) に定義されている ATTENDEE プロパティと似た構文が使用されています。WCAP の attendee の構文を理解するには、このドキュメントにすべて目を通し、必要な基本情報を理解されることをお勧めします。WCAP では、パラメータの区切り文字として「^」を使用するなど、いくつかの違いがあります。ただし、複数の参加者を区切るときは、iCalendar で標準に使われるセミコロンを使用します。

たとえば、iCalendar では、次のような構文が使用されます。

```
PARSTAT=ACCEPTED;RSVP=TRUE:mailto:abc@xyz.com
```

WCAP では、次のように記述します。

```
PARSTAT=ACCEPTED^RSVP^=TRUE^mailto:abc@xyz.com
```

WCAP 参加者エントリの例

参加者 A (attA) が出席依頼を承諾する場合、WCAP コマンドは次のようなパラメータを含みます。

```
PARTSTAT=ACCEPTED^RSVP=TRUE^attA
```

参加者 B (attB) が出席依頼を辞退する場合、WCAP コマンドは次のようなパラメータを含みます。

```
PARTSTAT=DECLINED^RSVP=TRUE^attB
```

電子メールによる参加者 `jdoe@xyz.com` は、出席するかどうかをまだ決めていないが、返答する必要がない場合は、WCAP コマンドは次のようなパラメータを含みます。

```
PARTSTAT=NEEDS-ACTION^RSVP=FALSE:mailto:jdoe@xyz.com
```

表 10-36 は、WCAP が理解する iCalendar ATTENDEE プロパティのパラメータの一覧です。ほとんどのパラメータはオプションです。これらのすべてが Calendar Server で完全にサポートされているわけではありませんが、情報は登録されます。グループスケジューリングに関連するパラメータは、PARTSTAT と RSVP だけです。

表 10-36 WCAP で認識される iCalendar ATTENDEE パラメータ

パラメータ	目的
PARTSTAT	必須のパラメータ。参加者の参加状態を示す
CUTYPE	カレンダーユーザのタイプ
MEMBER	参加者が所属するグループのリスト。WCAP ではこれらのグループは使用されない
ROLE	この会議での参加者の役割
RSVP	参加者の返答が必要かどうか
DELEGATED-TO	出席を委任されたユーザ
DELEGATED-FROM	出席を委任したユーザ
SENT-BY	指定したユーザの代理を務めるカレンダーユーザ
CN	参加者の表示名
DIR	ディレクトリエントリ参照
LANG	エントリの言語

グループスケジューリング - method パラメータ

method パラメータを使用して、出席依頼、返答、キャンセルなど、使用するメッセージのタイプを指定します。

出席依頼では、次の3つのメッセージタイプを使用できます。

- 開催者が出席を依頼する

開催者が会議を作成する場合、次の2つの方法で出席を依頼します。

 - PUBLISH メッセージの送信。会議を作成または変更して、参加者に通知します。method パラメータを「1」に設定します。
 - REQUEST メッセージの送信。会議を作成または変更して、参加者からの返答を要求します。method パラメータを「2」に設定します。

PUBLISH または REQUEST メッセージを送信できるのは、会議の開催者だけです。
- 参加者が出席依頼に対して返答する

参加者が REPLY メッセージを送信する。出席依頼を承諾するか辞退するかを通知します。method パラメータを「4」に設定します。
- 開催者が会議をキャンセルする

開催者が会議をキャンセルする場合は、deleteevents コマンドを使って CANCEL を送信することにより、参加者に通知されます。method パラメータを「8」に設定します。

注 キャンセルするときは、storeevents ではなく、deleteevents コマンドを使用することをお勧めします。

次の例は、開催者「org」が参加者「attA」および「attB」に会議への出席を依頼する場合の WCAP コマンドを示しています。参加者「attA」は出席依頼を承諾し、参加者「attB」は辞退します。会議の uid は、「event_u1」です。イベントは、両方の参加者のカレンダーに作成されます。各参加者は、各自のカレンダーでイベントに返答します。この返答は、iPlanet Calendar Server 5.0 の Group Scheduling Engine から開催者のカレンダーに返信されます。

出席依頼

```
storeevents.wcap?id=${SESSIONID of org}&calid=org&dtstart=
20010201T200000Z&dtend=20010201T210000Z&summary=invite_attA_attB
&method=2&attendees=PARTSTAT=ACCEPTED^RSVP=TRUE^org;PARTSTAT=
NEEDS-ACTION^RSVP=TRUE^attA;PARTSTAT=NEEDS-ACTION^RSVP=
TRUE^attB&fmt-out=text/xml
```

承諾

```
storeevents.wcap?id=${SESSIONID ofattA}&calid=attA&uid=event_u1
&method=4&attendees=PARTSTAT=ACCEPTED^RSVP=TRUE^attA
&fmt-out=text/xml
```

出席の辞退


```
storeevents.wcap?id=${SESSIONID ofattB}&calid=attB&uid=event_u1
&method=4&attendees=PARTSTAT=DECLINED^RSVP=TRUE^attA
&comments=I_cannot_make_it_Sorry&fmt-out=text/xml
```

storetodos

目的

1つまたは複数の仕事をカレンダーに追加します。

パラメータ

表 10-37 は、このコマンドの 44 個のパラメータの一覧です。

表 10-37 storetodos のパラメータ

パラメータ	タイプ	目的	必須 / 任意	デフォルト
alarmAudio	ISO8601Date Time Z 文字列	オーディオアラームを再生する時刻	任意	該当なし
alarmEmails	電子メールアドレスをセミコロンの区切ったリスト	仕事のアラーム通知の受信者	任意	該当なし
alarmFlashing	ISO8601 Date Time Z 文字列	フラッシュアラームを実行する時刻	任意	該当なし
alarmPopup	ISO8601 Date Time Z 文字列	ダイアログアラームを表示する時刻	任意	該当なし
alarmStart	ISO8601 DateTime Z 文字列	仕事のアラーム通知を送信する時刻	任意	該当なし
attachments	文字列をセミコロンの区切ったリスト	このパラメータは、iCalendar と相互運用するときのパラメータ。文字列は URL	任意	該当なし
attendees	文字列をセミコロンの区切ったリスト	仕事の参加者	任意	該当なし
brief	整数 (0、1)	JavaScript の簡略出力を出力するかどうかを示すブール値 出力形式 (fmt-out) が JavaScript (text/js) の場合にだけ適用される 1 = 簡略出力 0 = 完全出力	任意	0
calid	文字列	仕事の登録先のカレンダー ID	必須	該当なし

表 10-37 storetodos のパラメータ (続き)

パラメータ	タイプ	目的	必須 / 任意	デフォルト
categories	文字列をセミコロ ンで区切ったリス ト	仕事のカテゴリ	任意	該当なし
completed	ISO8601 DateTime Z 文字列	仕事の完了日。値が 0 の場合は、仕 事が完了していないことを意味す る	任意	0
contacts	文字列をセミコロ ンで区切ったリス ト	仕事の連絡先	任意	該当なし
desc	文字列	仕事の目的。任意の長さの文字列 このパラメータを渡さない場合、 desc は summary 値に設定される 文字列にスペースを含める場合は、 コード %20 を使用する	任意	summary パラメータ の値
dtstart	ISO8601 DateTime 文字列	仕事の開始日時 仕事を変更するときは必要なし 仕事を作成するときに必要な	任意 必須	該当なし
due	ISO8601 DateTime Z 文字列	仕事の終了日時	任意	該当なし
duration	ISO8601 継続時間 文字列	仕事の継続時間	任意	該当なし
exdates	ISO8601 DateTime Z 文字列をセミコ ロンで区切ったリ スト	除外する仕事繰り返し日付	任意	該当なし
exrules	文字列をセミコロ ンで区切ったリス ト	除外する仕事繰り返し規則。繰り返 し規則の文字列をセミコロンの区 切ったリスト 各規則の値は、二重引用符で囲むこ とが必要。170 ページの「繰り返し 処理」を参照	任意	該当なし

表 10-37 storetodos のパラメータ (続き)

パラメータ	タイプ	目的	必須 / 任意	デフォルト
fetch	整数 (0、1)	新しく登録された仕事を取り込んで返すかどうかを示すブール値 1 = 新しく登録された仕事を取得して返す 0 = 返さない	任意	0
fmt-out	文字列	取得するデータの形式 次の形式を指定できる text/calendar text/xml text/js	任意	text/js
geo	セミコロンで区切った2つの実数	仕事の地理的位置を表す2つの実数 (緯度と経度) を、セミコロンで区切った文字列 たとえば、37.31;-123.2 のように指定する	任意	0;0
icsClass	文字列	仕事のクラス。次の値を指定できる PUBLIC PRIVATE CONFIDENTIAL	任意	PRIVATE
icsUrl	文字列	仕事の URL	任意	""
id	一意の識別子文字列	セッション ID	必須	該当なし
isAllDay	整数 (0、1)	終日の仕事かどうかを示すブール値 1 = 終日の仕事 0 = 終日の仕事ではない	任意	0
language	文字列	仕事の言語 (例: 「en」、「fr」、「de」)	任意	該当なし
location	文字列	仕事の場所	任意	""

表 10-37 storetodos のパラメータ (続き)

パラメータ	タイプ	目的	必須 / 任意	デフォルト
method	整数 (1、2、4、8、16、32)	グループスケジューリングの ITIP メソッド 次の値を指定できる 1 = PUBLISH 2 = REQUEST 4 = REPLY 8 = CANCEL 16 = MOVE 32 = COUNTER	任意	1 (PUBLISH)
mod	整数	登録または変更する繰り返しを指定する 新しい仕事を作成するときには必要なし 仕事を変更するときには必要 次の値を指定できます 1 = THISINSTANCE 2 = THISANDFUTURE 3 = THISANDPRIOR 4 = THISANDALL	任意 必須	該当なし
notify	整数 0,1	このパラメータは 5.0 では使用できない。2.x との下位互換性を保持するために残されている。GSE (Group Scheduling Engine) モジュールでは、電子メール通知が処理される 参加者に仕事の変更を通知するかどうかを示すブール値 1 = 参加者に変更を通知する 0 = 参加者に通知しない	任意	0
orgEmail	電子メールアドレス	仕事の連絡先電子メールアドレス (通常、開催者の電子メールアドレス)	任意	該当なし
orgUID	userid	開催者の userid	任意	該当なし
percent	整数 (0 ~ 100)	仕事の完了率	任意	0

表 10-37 storetodos のパラメータ (続き)

パラメータ	タイプ	目的	必須 / 任意	デフォルト
priority	整数 (0 ~ 9)	仕事の優先順位 0 = 最も低い優先順位 9 = 最も高い優先順位	任意	0
rchange	整数 (0, 1)	繰り返し仕事を継続するかどうかを示すブール値 1 = 繰り返し仕事を継続する 0 = 継続しない	任意	1
rdates	ISO8601 DateTime Z 文字列をセミコロンで区切ったリスト	仕事の繰り返し日付	任意	該当なし
relatedTos	二重引用符で囲んだ文字列をセミコロンで区切ったリスト	この仕事に関連するほかの仕事	任意	該当なし
resources	文字列をセミコロンで区切ったリスト	仕事に関連付けられたリソース	任意	該当なし
rid	ISO8601 DateTime Z 文字列	仕事の繰り返し ID 新しい仕事を作成するときには必要なし 仕事を変更するときには必要	任意 必須	該当なし
rrules	文字列をセミコロンで区切ったリスト	仕事の繰り返し規則。繰り返し規則の文字列をセミコロンで区切ったリスト 各規則の値は、二重引用符で囲む必要がある。170 ページの「繰り返し処理」を参照	任意	該当なし
seq	整数	仕事のシーケンス番号	任意	0

表 10-37 storetodos のパラメータ (続き)

パラメータ	タイプ	目的	必須 / 任意	デフォルト
status	整数	仕事の状態コード。次の値を指定できる 0 CONFIRMED 1 CANCELLED 2 TENTATIVE 3 NEEDS_ACTION 4 COMPLETED 5 IN_PROCESS 6 DRAFT 7 FINAL	任意	該当なし
summary	文字列	仕事の要約。任意の長さの文字列 新しい仕事を作成するときに必要な 仕事を変更するときには必要なし 文字列にスペースを含める場合は、コード %20 を使用する	必須 任意	default summary 該当なし
tzid	タイムゾーン ID 文字列。「America/Los_Angeles」など	タイムゾーン。すべての日付は、このタイムゾーンを参照して解釈される タイムゾーンを渡さない場合、すべての日付は世界協定時刻 (UTC または Z 形式) として解釈される	任意	"GMT"
uid	文字列	登録する仕事の一意の識別子 新しい仕事の場合は、システムによって生成される 仕事を変更するときには必要	任意 必須	該当なし default uid

説明

このコマンドは、指定した属性を持つ仕事の作成、変更、およびデータベース内の指定したカレンダーへの登録を行います。

このコマンドは、rrules、exrules、rid、mod、および rchange パラメータの指定に従って、繰り返しを作成および登録します。170 ページの「繰り返し処理」を参照してください。

notify 値が 1 の場合、IMIP PUBLISH メッセージが仕事の参加者に送信されます。

カレンダーサーバでは、ファイルの添付はサポートされていません。attachments パラメータは、iCalendar と相互運用するときに使います。カレンダーサーバでは機能しません。

method パラメータ

グループスケジューリングには、次の ITIP メソッドのどれかを指定します。

1	PUBLISH	開催者専用
2	REQUEST	開催者専用
4	REPLY	参加者専用
8	CANCEL	開催者専用
16	MOVE	該当なし
32	COUNTER	参加者専用

戻り値

fmt-out パラメータを text/calendar または text/xml に設定すると、次のように HTML コメントの中にエラー値だけが返されます。

```
<!-- store_errno="0" -->
```

fmt-out パラメータを text/js に設定するか、設定しなかった (このパラメータのデフォルトは text/js) 場合、このコマンドは、fetchcomponents_by_range.wcap コマンドが返した JavaScript を返します。

エラーコード

このコマンドを使用してリンクされている仕事を変更することはできません。変更しようとするコマンドの実行が失敗し、errno 配列に CANNOT_MODIFY_LINKED_TODOS (32) が返されます。

すでに予定が設定されている時間枠に仕事を登録しようとする (二重予約)、コマンドの実行が失敗し、エラー STORE_FAILED_DOUBLE_BOOKED (40) が返されます。

必要なパラメータ

このコマンドを使って、新しい仕事の作成や、既存の仕事の変更を行います。必要なパラメータは、実行する内容によって次のように異なります。

- 新しい仕事を作成する場合は、次の 2 つのパラメータだけが必要です。
 - dtstart
 - summary

その他のパラメータはすべてオプションです。uid は、サーバによって生成されま
す。

- 既存の仕事を変更する場合は、次の 3 つのパラメータが必要です。
 - uid
 - rid
 - mod

その他のパラメータはすべてオプションです。パラメータを指定しなかった場合
は、プロパティの以前の値が保持されます。

継続時間

期限 (due) は、duration よりも優先されます。duration と due の両方を指定すると、
duration は無視されます。

継続時間は、次のように ISO8601 形式で指定します。

- P1Y2M3DT1H30M10S は、1 年、2 か月、3 日、1 時間、30 分、10 秒の継続時間を表します。
- PT1H30M は、1 時間 30 分の継続時間を表します。
- P1D は、1 日の継続時間を表します。
- PT15M は、15 分の継続時間を表します。

文字列に含まれる文字 T で、日付情報 (年、月、日) と時刻情報 (時、分、秒) を区切ります。

upload_file

目的

ファイルをサーバにアップロードします。ファイルのインポート用の write_file と
組み合わせて使用します。

パラメータ

表 10-38 は、このコマンドのパラメータです。

表 10-38 upload_file パラメータ

パラメータ	タイプ	目的	必須 / 任意	デフォルト
idt	一意の識別子文字列	セッション ID	必須	該当なし

説明

このコマンドは、write_file と組み合わせて、iPlanet Calendar Server 2.0 互換のファ
イルをインポートします。ファイルを添付するため、POST を使用する必要があります。

ファイルのインポートは、UI を使用して次のように処理されます。

- `upload_file` コマンドを使って、ユーザのコンピュータのファイルをサーバにアップロードします。`upload_file` コマンドは、POST でなければなりません。
- アップロードした 1 つまたは複数のファイル名を取得し、`write_file` コマンドを使用してデータベースに書き込みます。

このように処理が分かれるのは、パラメータ値と添付ファイルの両方を JavaScript UI で連結して処理できないためです。また、この分離によって、実際にファイルをデータベースに書き込む前に、複数のファイルをサーバにアップロードすることができます。

`upload_file` の例

`upload_file` で送信するデータは、`import.wcap` のデータとまったく同じです。使用できるファイルは、`iCalendar` ファイルと `XML` ファイルだけです。任意のファイルをアップロードすることができますが、データベースに正常に追加できるのは、`iCalendar` ファイルと `XML` ファイルだけです。

`upload.wcap` から返される JavaScript は、次のようになります。

```
Completed sending of import HTTP/1.1 200
Date:Tue, 05 Oct 2000 23:37:51 GMT
Content-type:text/html; charset=iso-8859-1
Content-length: 301
Last-modified:Tue, 05 Oct 2000 23:37:51 GMT
Pragma:no-cache
Expires: 0
Cache-Control:no-cache
Connection:Keep-Alive

<html><head><script>
function color(s) { if (s) document.bgColor=s
}
var
filename='s://ns//server//msg//calendar//core//parser//test//ical1.
ics'

var uniquefilename='cf8bfa37b347000001000000ef000000'

var size=640
var errno=0
var errstr=''
</script></head>
<body onLoad=parent.uploadCB()>
</body></html>
```

write_file の例

前述の例では、`uniquefilename` という JavaScript 変数を含む行が太字になっています。この変数を、`write_file` コマンドの `uniquefilename` パラメータへの入力として使用します。

UI を使うと、`upload_file` で該当するファイルをアップロードし、`write_file` が呼び出され、ファイルが実際にデータベースに書き込まれます。

たとえば、次の URL を使用して、前述の例のアップロード呼び出しをカレンダー `jdoe` に書き込みます。`uniquefilename` パラメータが、`upload_file.wcap` に返された `uniquefilename` という JavaScript 変数と一致していることに注意してください。

```
http://webcalendarserver/write_file.wcap?id=abc&calid=jdoe&dtstart=0&dtend=0&content-in=text/calendar&uniquefilename=cf8bfa37b347000001000000ef000000
```

2 つのファイルをサーバに書き込むには、一意なファイル名をセミコロンで区切ります。2 つのファイルの `content-type` は同じでなければなりません。

```
http://webcalendarserver/write_file.wcap?id=abc&calid=jdoe&dtstart=0&dtend=0&content-in=text/calendar&uniquefilename=cf8bfa37b347000001000000ef000000;dg9cgb48c458000001000000ab000000
```

version

目的

現在の WCAP バージョンを取得します。

パラメータ

表 10-39 は、コマンドのパラメータです。

表 10-39 version のパラメータ

パラメータ	タイプ	目的	必須 / 任意	デフォルト
<code>fmt-out</code>	文字列	取得するデータの形式 次の形式を指定できる <code>text/calendar</code> <code>text/xml</code> <code>text/js</code>	任意	<code>text/js</code>

説明

このコマンドは、WCAP の現在のバージョンを取得します (注: このバージョンは、サーバおよび HTTP のバージョンとは異なります。)

戻り値

このコマンドでは、iCalendar、XML、およびJavaScriptの出力形式がサポートされています。iCalendar およびXML では、変数 `X-NSCP-WCAPVERSION` に WCAP バージョン番号が返されます。JavaScript では、変数 `wcapversion` にバージョン番号が返されます。

例

それぞれの出力データ形式の例を以下に示します。

- JavaScript: (URL: /version.wcap)

```
<html><head><script>
function color(s) { if (s) document.bgColor=s
}
var id='0'
var userid=''
var calid=''
var errno=new Array()
var wcapversion=1.0.0
parent.ceCB(window.name)
</script></head></html>
```

- iCalendar: (URL: /version.wcap?fmt-out=text/calendar)

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//iPlanet/Calendar Hosting Server//EN
X-NSCP-WCAPVERSION:WCAP
END:VCALENDAR
```

- XML: (URL: /version.wcap?fmt-out=text/xml)

```
<?xml version="1.0" encoding="UTF-8">
<iCalendar>
<iCal>
<X-NSCP-WCAPVERSION>1.0.0</X-NSCP-WCAPVERSION>
</iCal>
</iCalendar>
```

write_file

目的

ファイルをデータベースに書き込みます。

パラメータ

表 10-40 は、このコマンドの 6 つのパラメータの一覧です。

表 10-40 write_file のパラメータ

パラメータ	タイプ	目的	必須 / 任意	デフォルト
calid	文字列	ファイルのインポート先カレンダー	必須	該当なし
content-in	文字列	入力データのコンテンツタイプ	必須	該当なし
dtend	ISO8601 DateTime "Z" 文字列	インポートするイベントの終了範囲 値が 0 の場合は、時間の最後まで のイベントがすべてインポートされる	任意	0
dtstart	ISO8601 DateTime "Z" 文字列	インポートするイベントの開始範囲 値が 0 の場合は、時間の最初から のイベントがすべてインポートされる	任意	0
id	一意の識別子 文字列	セッション ID	必須	該当なし
uniquefilename	文字列をセミ コロンで区 切ったリスト	サーバからインポートするファイルの名 前	必須	該当なし

説明

write_file コマンドは、upload_file コマンドと組み合わせて、ファイルをインポートします。このコマンドのパラメータは、import コマンドとまったく同じです。ファイルのインポート処理の例および詳細は、upload_file を参照してください。

記号

% 符号化文字, 151

数字

5.0 の新機能

Palm Pilot との同期, 21

SHTML, 21

グループスケジューリング, 21

水平方向のスケラビリティ, 21

A

acl パラメータ (WCAP), 156, 245

addlink コマンド (WCAP), 175

API

authSDK

CEXP_GenerateLoginURL, 130

CEXP_GetVersion, 131

CEXP_Init, 131

CEXP_SetHttpPort, 132

CEXP_Shutdown, 132

アーキテクチャ, 126

紹介, 125

初期化, 126

CSAPI

csIAccessControl, 50

csIAuthentication, 52

csICalendarLookup, 57

csICalendarServer, 72

csIDataTranslator, 60

csIMalloc, 73

csIPlugin, 64

csIQualifiedCalidLookup, 66

csIUserAttributes, 68

インタフェース, 49

紹介, 36, 41

ENS

公開および購読ディスプレイパッチャー, 110

サブスクライバ, 102

パブリッシャ, 95

SSO

紹介, 135

WCAP

addlink, 175

change_password, 177

check_id, 178

createcalendar, 179

deletecalendar, 181

deletecomponents_by_range, 182

deleteevents_by_id, 184

deleteevents_by_range, 187

deletetodos_by_id, 188

deletetodos_by_range, 191

export, 193

fetchcomponents_by_alarmrange, 196

fetchcomponents_by_attendee_error, 199

fetchcomponents_by_lastmod, 201

fetchcomponents_by_range, 204

fetchevents_by_id, 214

fetchtodos_by_id, 216

get_all_timezones, 218

- get_calprops, 220
- get_freebusy, 224
- get_guids, 228
- get_userprefs, 230
- import, 232
- login, 234
- logout, 237
- ping, 237
- search_calprops, 238
- set_calprops, 241
- set_userprefs, 245
- storeevents, 247
- storetodos, 257
- upload_file, 264
- version, 266
- write_file, 267
- はじめに, 145

authSDK

- アーキテクチャ, 126

関数

- CEXP_GenerateLoginURL, 126, 130
- CEXP_GetVersion, 126, 131
- CEXP_Init, 126, 131
- CEXP_SetHttpPort, 126, 132
- CEXP_Shutdown, 126, 132

- クリーンアップ, 126

- 検索, 126

- 紹介, 125

- 初期化, 126

- 定義, 125

- 統合および使用方法, 133

C

- Calloc メソッド (CSAPI), 74

- change_password コマンド (WCAP), 177

- ChangePassword メソッド (CSAPI), 53

- check_id コマンド (WCAP), 178

- CheckAccess メソッド (CSAPI), 50

- cookie (SSO), 135, 136, 138

- createcalendar コマンド (WCAP), 179

- csadmind, 23

- csadmind デーモン, 25, 26, 27

- csadmind デーモン (ENS), 86

CSAPI

- アーキテクチャ, 42

依存性

- NSPR, 44

- XPCOM, 44

- インタフェースのリスト, 49

クライアント API

- csIAccessControl, 50

- csIAuthentication, 52

- csICalendarLookup, 57

- csIDataTranslator, 60

- csIPlugin, 64

- csIQualifiedCalidLookup, 66

- csIUserAttributes, 68

- リスト, 46

サーバ API

- csICalendarServer, 72

- csIMalloc, 73

- リスト, 46

- 紹介, 41

- 定義, 36

- メソッドのリターンコード, 53

- モジュール構造, 46

要件

- スレッドに対して安全なプラグイン, 44

- csdwpd, 23

- csdwpd デーモン, 25

- cshtpd, 23

- cshtpd デーモン, 25, 27

- csIAccessControl (CSAPI)

- CheckAccess メソッド, 50

- Init メソッド, 52

- csIAuthentication (CSAPI)

- ChangePassword メソッド, 53

- Init メソッド, 54

- Logon メソッド, 55

- Logout メソッド, 55

- VerifyUserExists メソッド, 56

- csICalendarLookup (CSAPI)

- FindCalid メソッド, 67

- FreeCalid メソッド, 58

- FreeType メソッド, 60
- Init メソッド, 57, 68
- QualifyCalid メソッド, 58
- QueryType メソッド, 59
- csICalendarServer (CSAPI)
 - GetVersion メソッド, 72
 - Init メソッド, 73
- csIDataTranslator (CSAPI)
 - GetSupportedContentType メソッド, 61
 - Init メソッド, 62
 - Translate メソッド, 63
- csIMalloc (CSAPI)
 - Calloc メソッド, 74
 - FreeIf メソッド, 75
 - Free メソッド, 74
 - Init メソッド, 75
 - Malloc メソッド, 76
- csIPlugin (CSAPI)
 - GetDescription メソッド, 64
 - GetVendorName メソッド, 65
 - GetVersion メソッド, 65
 - Init メソッド, 66
- csIRealloc (CSAPI)
 - Calloc メソッド, 76
- csIUserAttributes (CSAPI)
 - FreeAttribute メソッド, 69
 - GetAttribute メソッド, 69
 - Init メソッド, 70
 - SetAttribute メソッド, 71
- csnotifyd, 23
- csnotifyd デーモン, 25
- csnotifyd デーモン (ENS), 86

D

- deletecalendar コマンド (WCAP), 181
- deletecomponents_by_range コマンド (WCAP), 182
- deleteevents_by_id コマンド (WCAP), 184
- deleteevents_by_range コマンド (WCAP), 187
- deletetodos_by_id コマンド (WCAP), 188

- deletetodos_by_range コマンド (WCAP), 191
- DWP (データベースワイヤプロトコル), 23

E

- enpd, 24
- enpd デーモン, 25
- ENS

API

関数リスト

- 公開および購読ディスパッチャー, 90, 110
 - サブスクリイバ, 89, 102
 - パブリッシャ, 89, 95
- 公開および購読ディスパッチャー関数
 - pas_dispatch, 112
 - pas_dispatcher_delete, 111
 - pas_dispatcher_new, 111
 - pas_dispatcher_t 定義, 111
 - pas_shutdown, 112
- サブスクリイバ関数
 - renl_cancel_subscriber, 110
 - renl_create_subscriber, 109
 - subscribe_a, 106
 - subscriber_cb_t, 103
 - subscriber_delete, 108
 - subscriber_new_a, 104
 - subscriber_new_s, 105
 - subscriber_notify_cb_t, 104
 - subscriber_t, 103
 - subscription_t, 103
 - unsubscribe_a, 107
- パブリッシャ関数
 - publish_a, 98
 - publish_s, 99
 - publisher_cb_t, 96
 - publisher_delete, 100
 - publisher_new_a, 96
 - publisher_new_s, 97
 - publisher_t, 95
 - renl_cancel_publisher, 102
 - renl_create_publisher, 101

- Calendar Server との対話, 84

- RENL 定義, 95

- subscriber_new_a 関数, 104
- アラーム
 - ディスパッチャー, 84
- アラームキュー, 85
- イベント参照, 81
- 公開および購読ディスパッチャー API, 110
- コードサンプル
 - パブリッシャ, 113
- サブスクライバ API, 102
- 購読, 83
- 紹介, 79
- 通知
 - 一般的な流れ, 86
 - サーバ, 82
 - 非送信請求, 82
 - 「キャンセル」の定義, 83
 - 「購読する」の定義, 83
- 定義されたサービス, 82
- 定義されているサービス, 24
- デーモン
 - csadmind, 85, 86, 93
 - csnotifyd, 85, 86, 93
- デーモン、enpd, 82
- パブリッシャ API, 95
- 用語集, 80
- 「通知する」の定義, 82

ENS (イベント通知サービス), 37

export コマンド (WCAP), 193

F

- fetchcomponents_by_alarmrange コマンド (WCAP), 196
- fetchcomponents_by_attendee_error コマンド (WCAP), 199
- fetchcomponents_by_lastmod コマンド (WCAP), 201
- fetchcomponents_by_range コマンド (WCAP), 204
- fetchevents_by_id コマンド (WCAP), 214
- fetchtodos_by_id コマンド (WCAP), 216
- FindCalid メソッド (CSAPI), 67

- FreeAttribute メソッド (CSAPI), 69
- FreeCalid メソッド (CSAPI), 58
- FreeIf メソッド (CSAPI), 75
- FreeType メソッド (CSAPI), 60
- Free メソッド (CSAPI), 74

G

- get_all_timezones コマンド (WCAP), 218
- get_calprops コマンド (WCAP), 220
- get_freebusy コマンド (WCAP), 224
- get_guids コマンド (WCAP), 228
- get_userprefs コマンド (WCAP), 230
- GetAttribute メソッド (CSAPI), 69
- GetDescription メソッド (CSAPI), 64
- GetSupportedContentType メソッド (CSAPI), 61
- GetVendorName メソッド (CSAPI), 65
- GetVersion メソッド (CSAPI), 65, 72
- GSE、「グループスケジューリング」を参照, 24
- GUID (WCAP), 228

H

- HTTP サービスデーモン, 23

I

- import コマンド (WCAP), 232
- Init メソッド (CSAPI), 52, 54, 57, 62, 66, 68, 70, 73, 75
- ITIP メソッド (WCAP), 263

L

- LDAP ディレクトリサーバ, 25

login コマンド (WCAP), 234
Logon メソッド (CSAPI), 55
logout コマンド (WCAP), 237
Logout メソッド (CSAPI), 55

M

Malloc メソッド (CSAPI), 76
method パラメータ (WCAP), 263
MIME タイプ (CSAPI), 61

P

Palm Pilot との同期, 21
pas_dispatcher_delete 関数 (ENS), 111
pas_dispatcher_new 関数 (ENS), 111
pas_dispatcher_t 定義 (ENS), 111
pas_dispatch 関数 (ENS), 112
pas_shutdown 関数 (ENS), 112
ping コマンド (WCAP), 237
publish_s 関数 (ENS), 99
publish_a 関数 (ENS), 98
publisher_cb_t 関数 (ENS), 96
publisher_delete 関数 (ENS), 100
publisher_new_a 関数 (ENS), 96
publisher_new_s 関数 (ENS), 97
publisher_t 関数 (ENS), 95

Q

QualifyCalid メソッド (CSAPI), 58
QueryType メソッド (CSAPI), 59

R

Realloc メソッド (CSAPI), 76
RENL (Reliable Event Notification Link) (ENS), 95,
86, 88
renl_cancel_publisher 関数 (ENS), 102
renl_cancel_subscriber 関数 (ENS), 110
renl_create_publisher 関数 (ENS), 101
renl_create_subscriber 関数 (ENS), 109

S

search_calprops コマンド (WCAP), 238
set_calprops コマンド (WCAP), 241
set_userprefs コマンド (WCAP), 245
SetAttribute メソッド (CSAPI), 71
.shtml 拡張子、「SHTML」を参照, 32
storeevents コマンド (WCAP), 247
storetodos コマンド (WCAP), 257
subscribe_a 関数 (ENS), 106
subscriber_cb_t 関数 (ENS), 103
subscriber_delete 関数 (ENS), 108
subscriber_new_a 関数 (ENS), 104
subscriber_new_s 関数 (ENS), 105
subscriber_t 関数 (ENS), 103
subscription_t 関数 (ENS), 103

T

Translate メソッド (CSAPI), 63

U

UI ジェネレータ
SHTML, 32, 145
WCAP, 32, 145

unsubscribe_a 関数 (ENS), 107
upload_file コマンド (WCAP), 264
URI/URL
 イベント参照 (ENS), 81
 形式 (WCAP), 151

V

VerifyUserExists メソッド (CSAPI), 56
version コマンド (WCAP), 266

W

WCAP

HTML フォームの送信, 151
UI ジェネレータ, 32, 145
URI 形式, 151
アクセスコントロールエントリ (ACE), 156, 245
新しいコマンド
 fetchcomponents_by_alarmrange, 146, 166
 fetchcomponents_by_attendee_error, 146, 166
 get_freebusy, 146, 166
新しいパラメータ, 168
 compstate, 147, 167
 compstate 値, 168
 method, 147, 167
エラー処理, 161
管理者コマンド
 change_password, 177
 createcalendar, 179
 deletecalendar, 181
 ping, 237
クライアントサイドのイベント通知, 151
クライアントの要求形式, 150
繰り返し処理, 170
グループスケジューリング, 263
グループスケジューリングの変更, 147, 167, 168
コマンド
 addlink, 175
 change_password, 177
 check_id, 178

 createcalendar, 179
 deletecalendar, 181
 deletecomponents_by_range, 182
 deleteevents_by_id, 184
 deleteevents_by_range, 187
 deletetodos_by_id, 188
 deletetodos_by_range, 191
 export, 193
 fetchcomponents_by_alarmrange, 196
 fetchcomponents_by_attendee_error, 199
 fetchcomponents_by_lastmod, 201
 fetchcomponents_by_range, 204
 fetchevents_by_id, 214
 fetchtodos_by_id, 216
 get_all_timezones, 218
 get_calprops, 220
 get_freebusy, 224
 get_guids, 228
 get_userprefs, 230
 import, 232
 login, 234
 logout, 237
 ping, 237
 search_calprops, 238
 set_calprops, 241
 set_userprefs, 245
 storeevents, 247
 storetodos, 257
 upload_file, 264
 version, 266
 write_file, 267
 概要, 148
 形式, 150
 リスト, 153
出力形式, 151, 169
セッション ID, 149
はじめに, 145
符号化文字, 151
予定なし / 予定ありに限定したアクセス, 165, 245
リターンコード, 161
.wcap 拡張子、「WCAP」を参照, 32
write_file コマンド (WCAP), 267

あ

- アーキテクチャ
 - authSDK, 126
 - CSAPI, 36
 - iPlanet Calendar Server の概要, 17
 - アクセスコントロールサブシステム, 32
 - イベントフィード, 34
 - カレンダーの所有権, 35
 - 基本, 30
 - データ形式, 35
 - 電子メールアラーム, 34
 - ユーザ設定, 35
- アクセスコントロール, 35
- アクセスコントロールエントリ (ACE)
(WCAP), 156, 245
- アクセスコントロールサブシステム, 32
- アクティブサーバのテスト (WCAP), 237
- 新しいパラメータ
 - compstate 値, 168
- 新しいパラメータ (WCAP), 168
 - compstate, 147, 167
 - method, 147, 167
- アラーム
 - アラーム通知 (ENS), 86
 - イベント参照 (ENS), 85
- アラームキュー (ENS), 85

い

- 移行
 - iPlanet Calendar Server 2.x から, 21
- イベント
 - アラーム (ENS)
 - アラーム
キュー, 85
 - アラームトリガ (WCAP), 196
 - インポート (WCAP), 232
 - エクスポート (WCAP), 193
 - エラーの取得 (WCAP), 199
 - 繰り返し処理 (WCAP), 170

- 削除 (WCAP), 184, 187
- 参照 (ENS), 81
 - URI 構文スキーマ, 81
 - アラーム URI プロトタイプ, 85
- 取得 (WCAP), 204, 214
- 追加 (WCAP), 247
- 定義 (ENS), 81
- フィード, 34
- 変更の取得 (WCAP), 201
- イベントサブスクリバ (ENS)
 - csnotifyd, 85, 86
- イベント通知
 - クライアントサイド (WCAP), 151
 - サーバ (ENS), 82
 - 非請求通知 (ENS), 82
 - 「キャンセル」の定義 (ENS), 83
 - 「購読」の定義 (ENS), 83
 - 「通知」の定義 (ENS), 82
- イベント通知サービス, 37
- イベント通知サービス (ENS), 79, 82
- イベントパブリッシャ (ENS)
 - csadmind, 85
- イベントパブリッシャ (ENS)
 - csadmind, 86
- イベントフィード, 34
- インタフェース
 - csIAAuthentication, 53
 - csIDataTranslator, 60
 - csiMalloc, 73
 - csiPlugin, 64
 - csiUserAttributes, 68

え

- エラー
 - リターンコード (WCAP), 161

お

主な所有者, 35

か

カレンダー

MIME タイプ (CSAPI), 61

URI/URL, 34

アクセスコントロール, 35

イベントの削除 (WCAP), 184, 187

イベントフィールド, 34

主な所有者, 35

グループ, 33

公開カレンダーの定義, 35

コンポーネントの削除 (WCAP), 182

削除 (WCAP), 181

仕事の削除 (WCAP), 188, 191

詳細の表示を制限する (WCAP), 165, 245

新規作成 (WCAP), 179

スケジュールの作成 (WCAP), 165, 245

設定 (WCAP), 230

データ形式, 33

非公開カレンダーの定義, 35

プロパティ (WCAP), 241

ユーザ設定, 35

予定なし / 予定あり (WCAP), 165, 224, 245

カレンダーサーバ API (CSAPI) の定義, 36

カレンダーの検索 (WCAP), 238

カレンダーの取得 (WCAP), 238

カレンダープロパティ

取得 (WCAP), 220

管理者コマンド (WCAP)

change_password, 177

createcalendar, 179

deletecalendar, 181

ping, 237

管理者の問題 (SSO), 142

管理ツール

サーバ構成ファイル, 22

管理デーモン, 23

き

キャンセル (ENS), 83

く

クライアント API

リスト, 46

クライアント API (CSAPI)

csIAccessControl, 50

csIAuthentication, 52

csICalendarLookup, 57

csIDataTranslator, 60

csIPlugin, 64

csIQualifiedCalidLookup, 66

csIUserAttributes, 68

クライアントの要求形式 (WCAP), 150

クライアントユーザインタフェース

「ユーザインタフェース」を参照, 21

繰り返し処理 (WCAP), 170

exdates パラメータ, 172

exrules パラメータ, 172

mod パラメータ, 173

rchange パラメータ, 173

rdates パラメータ, 171

rid パラメータ, 173

rrules パラメータ, 171

繰り返しコンポーネントの削除, 160

削除オプション, 160

グループスケジュールリング

5.0 の新機能, 21

新しい fetch コマンドパラメータ (WCAP), 167

新しい WCAP パラメータ, 147, 168, 263

概要, 24

け

形式

クライアントの要求 (WCAP), 150

サーバの応答形式 (WCAP), 151

出力形式 (WCAP), 169

二

公開および購読ディスパッチャー関数 (ENS)

- pas_dispatch, 112
- pas_dispatcher_delete, 111
- pas_dispatcher_new, 111
- pas_dispatcher_t 定義, 111
- pas_shutdown, 112
- リスト, 110

公開カレンダー, 35

構成

- 単純な単一インスタンス, 27
- ネットワークフロントエンド、データベースバックエンド, 27
- 複数のフロントエンド、複数のバックエンド, 28

構成の例、水平方向のスケラビリティ, 26

構成パラメータ (SSO), 139

構成ファイル, 22

購読する (ENS), 83

コマンドの概要 (WCAP), 148

コマンドの形式 (WCAP), 150

コンポーネント (WCAP)

- インポート, 232
- エラーの取得, 199
- 繰り返し処理, 170
- 取得, 204
- 変更の取得, 201

コンポーネントの状態値の表 (WCAP), 168

な

サーバ API (CSAPI)

- csICalendarServer, 72
- csIMalloc, 73

サーバインタフェース (CSAPI), 47

サーバ構成ファイル, 22

サービス, 23, 24

- csadmin, 23
- csdwpd, 23
- cshttpd, 23
- csnotifyd, 23
- enpd, 24
- HTTP, 23
- イベント通知サービス, 24
- 管理, 23
- 通知, 23
- ディレトリサーバ, 25
- データベースワイヤプロトコル, 23

削除

- コンポーネント (WCAP), 182

サブスクリバデーモン、csnotifyd (ENS), 85, 86

購読 (ENS), 83

し

仕事 (WCAP)

- アラームトリガ, 196
- インポート, 232
- エクスポート, 193
- エラーの取得, 199
- 繰り返し処理, 170
- 削除, 188, 191
- 取得, 204, 216
- 追加, 257
- 変更の取得, 201

仕事の追加 (WCAP), 257

実装 (SSO), 138

自発通知 (ENS), 82

出力形式 (WCAP), 151, 169

処理の流れ (SSO), 136

シングルサインオフパラメータ (SSO), 139

シングルサインオン (SSO)

- cookie, 135, 136, 138
- 管理者の問題, 142
- 構成パラメータ, 139
- 実装の要件, 138

- 紹介, 135
- 処理の流れ, 136
- シングルサインオフパラメータ, 139
- 信頼されたアプリケーション, 135
- 信頼されたアプリケーションレコード, 139
- 信頼リストの管理, 142
- 推奨設定, 139
- スケーラビリティ, 143
- 制限, 136
- セキュリティ, 142
- 接頭辞の文字列, 139
- パフォーマンス, 143
- 複数の信頼されたアプリケーション, 139
- 問題、想定、および要件, 142
- 例, 139

信頼されたアプリケーション (SSO), 139

- 定義, 135
- 複数, 139

信頼リストの管理 (SSO), 142

す

推奨設定 (SSO), 139

水平方向にスケーラブル, 21, 25

水平方向のスケーラビリティ, 21, 25

- 5.0 の新機能, 21

スケーラビリティ (SSO), 143

せ

制限 (SSO), 136

セキュリティの問題 (SSO), 142

セッション

- 設定の変更 (WCAP), 245
- パスワードの変更 (WCAP), 245

セッション ID (WCAP), 149

セッション、有効性の確認 (WCAP), 178

設定

取得 (WCAP), 230

変更 (WCAP), 245

設定、推奨 (SSO), 139

接頭辞の文字列 (SSO), 139

た

タイムゾーン、取得 (WCAP), 218

つ

追加

- イベント (WCAP), 247

通知 (ENS)

- サーバ, 82
- 非送信請求, 82
- 「キャンセル」の定義, 83
- 「購読」の定義, 83
- 「通知する」の定義, 82

通知サービスデーモン, 23

通知する (ENS), 82

て

ディレクトリサーバサービス, 25

ディレクトリスキーマの変更, 25

データベースワイヤプロトコル (DWP), 23

デーモン

- csadmind, 23, 24, 25, 26, 85, 86
- csdwpd, 23, 24, 25, 26
- cshttpd, 23, 24, 25
- csnotifyd, 23, 24, 25, 85, 86
- enpd, 24, 25, 82
- 起動順, 22, 24
- 構成の例, 26

デフォルトの形式、WCAP コマンド, 151

電子メール

アラーム, 34
メッセージ形式, 34

に

認証

セッション ID (WCAP), 149
ユーザ (WCAP), 234

は

パスワード、変更 (WCAP), 177, 245
公開と購読の流れ (ENS), 86
パフォーマンス (SSO), 143
パブリッシャデーモン、csadmind (ENS), 85, 86

ひ

非請求通知 (ENS), 82

ふ

ファイルのインポート (WCAP), 264, 267
符号化文字として使用される % (パーセント記号), 161
符号化文字の例 (WCAP), 161
プラグインインタフェース (CSAPI), 46
プロキシ認証 SDK、authSDK を参照, 125
プロパティ
 カレンダーの取得 (WCAP), 220
 カレンダーの設定 (WCAP), 241

へ

変更

設定 (WCAP), 245
パスワード (WCAP), 245

ま

マルチプロセッサ, 21, 25

む

無効な cookie (SSO), 143

ゆ

ユーザアクセス (WCAP), 156, 245
ユーザインタフェース
 5.0 の SHTML 新機能, 21
 SHTML, 145
ユーザセッションの終了 (WCAP), 237
ユーザ設定
 定義, 35

よ

予定なし / 予定あり
 取得 (WCAP), 224
 定義 (WCAP), 165, 245

り

リターンコード
 エラーの配列および文字列 (WCAP), 161
 CSAPI メソッド, 53

れ

レイヤーエラー (WCAP), 161