

Oracle Calendar

アプリケーション開発者ガイド

リリース 2 (9.0.4)

2003 年 10 月

部品番号 : J07729-01

Oracle Calendar アプリケーション開発者ガイド, リリース 2 (9.0.4)

部品番号 : J07729-01

原本名 : Oracle Calendar Application Developer's Guide, Release 2 (9.0.4)

原本部品番号 : B10893-01

原本著者 : David Wood

原本協力者 : Graham Gilmore, Keith McDonald, Eric Plamondon, Jean-Marc Robillard

Copyright © 1998, 2003, Oracle Corporation. All rights reserved.

Printed in Japan.

制限付権利の説明

プログラム（ソフトウェアおよびドキュメントを含む）の使用、複製または開示は、オラクル社との契約に記された制約条件に従うものとします。著作権、特許権およびその他の知的財産権に関する法律により保護されています。

当プログラムのリバース・エンジニアリング等は禁止されております。

このドキュメントの情報は、予告なしに変更されることがあります。オラクル社は本ドキュメントの無謬性を保証しません。

* オラクル社とは、**Oracle Corporation**（米国オラクル）または日本オラクル株式会社（日本オラクル）を指します。

危険な用途への使用について

オラクル社製品は、原子力、航空産業、大量輸送、医療あるいはその他の危険が伴うアプリケーションを用途として開発されておりません。オラクル社製品を上述のようなアプリケーションに使用することについての安全確保は、顧客各位の責任と費用により行ってください。万一かかる用途での使用によりクレームや損害が発生いたしましても、日本オラクル株式会社と開発元である **Oracle Corporation**（米国オラクル）およびその関連会社は一切責任を負いかねます。当プログラムを米国国防総省の米国政府機関に提供する際には、『**Restricted Rights**』と共に提供してください。この場合次の **Notice** が適用されます。

Restricted Rights Notice

Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

このドキュメントに記載されているその他の会社名および製品名は、あくまでその製品および会社を識別する目的にのみ使用されており、それぞれの所有者の商標または登録商標です。

目次

はじめに	vii
対象読者	viii
このマニュアルの構成	viii
関連ドキュメント	ix
表記規則	x

第 I 部 Oracle Calendar SDK

1 Calendar SDK の概要

SDK の内容	1-2
このリリースの新機能	1-2
最適な使用例	1-2

2 Calendar SDK 実装時の考慮事項

キャラクタ・セット	2-2
iCalendar のサポート	2-2
iCalendar 入力	2-2
ATTACH	2-2
ATTENDEE	2-2
CATEGORIES	2-2
CLASS	2-3
DESCRIPTION	2-3
DTSTART、DTEND および DURATION	2-3
LOCATION	2-3
PRIORITY	2-3

RRULE	2-3
STATUS	2-4
SUMMARY	2-4
UID	2-4
VALARM	2-4
iCalendar 出力	2-4
ATTENDEE	2-4
CATEGORIES	2-4
CLASS	2-5
DESCRIPTION	2-5
DURATION または DTEND	2-5
DTSTART	2-5
LOCATION	2-5
ORGANIZER	2-5
PRIORITY	2-5
RESOURCES	2-5
RRULE	2-5
STATUS	2-6
SUMMARY	2-6
UID	2-6
VALARM	2-6
vCard のサポート	2-6
セキュリティ・モデル	2-6
アラーム	2-7
ユーザー識別情報	2-7
データ・ストリーム	2-9
アクセス制御	2-10

3 Oracle Calendar SDK 関数リファレンス

SDK 関数	3-2
CSDK_SetConfigFile	3-4
CSDK_CreateSession	3-5
CSDK_DestroySession	3-5
CSDK_GetCapabilities	3-6
CSDK_Authenticate	3-6
CSDK_ConfigureACE	3-9
CSDK_Connect	3-10

CSDK_ConnectAsSysop	3-11
CSDK_Deauthenticate	3-12
CSDK_Disconnect	3-12
CSDK_SetIdentity	3-13
CSDK_DestroyHandle	3-14
CSDK_DestroyMultipleHandles	3-15
CSDK_GetHandle	3-16
CSDK_GetHandleInfo	3-18
CSDK_CreateCallbackStream	3-19
CSDK_CreateFileStreamFromFilenames	3-21
CSDK_CreateMemoryStream	3-23
CSDK_DestroyMultipleStreams	3-26
CSDK_DestroyStream	3-26
CSDK_DeleteContacts	3-27
CSDK_FetchContactsByQuery	3-28
CSDK_FetchContactsByUID	3-29
CSDK_StoreContacts	3-30
CSDK_DeleteEvents	3-33
CSDK_FetchEventsByAlarmRange	3-35
CSDK_FetchEventsByRange	3-37
CSDK_FetchEventsByUID	3-39
CSDK_StoreEvents	3-40
CSDK_DeleteTasks	3-46
CSDK_FetchTasksByAlarmRange	3-47
CSDK_FetchTasksByRange	3-48
CSDK_FetchTasksByUID	3-50
CSDK_StoreTasks	3-51
CSDK_AddConditionToQuery	3-52
CSDK_CreateQuery	3-53
CSDK_DestroyQuery	3-54
CSDK_DestroyResult	3-54
CSDK_GetFirstFailure	3-55
CSDK_GetFirstParseError	3-56
CSDK_GetFirstResult	3-58
CSDK_GetNextFailure	3-59
CSDK_GetNextParseError	3-61
CSDK_GetNextResult	3-63
CSDK_GetStatusLevels	3-64

CSDK_GetStatusString	3-65
改訂された関数	3-65
レガシー関数	3-66
CAPI_CreateFileStream	3-67
CAPI_AuthenticateAsSysop	3-69

4 Oracle Calendar SDK の構成設定

blocking	4-2
client_name	4-2
client_version	4-3
cncachesize	4-3
direntycachesize	4-3
emailcachesize	4-4
itemcachesize	4-4
log_activity	4-4
log_modulesinclude	4-5
max_caldomain	4-5
max_masternode	4-6
max_sysop	4-6
max_user	4-6
securitycachesize	4-7
tzcachesize	4-7

5 Oracle Calendar SDK のフラグおよび機能

CSDK のフラグ	5-2
機能	5-5

6 Calendar SDK のステータス・コード

7 Oracle Calendar SDK の FAQ およびトラブルシューティング

FAQ	7-2
トラブルシューティング	7-3

第 II 部 Oracle Calendar Web Services ツールキット

8 Oracle Calendar Web Services ツールキットの概要

Oracle Calendar Web Services について	8-2
Oracle Calendar Web Services ツールキット	8-2
ツールキットの内容	8-3
セキュリティ	8-3

9 Oracle Calendar Web Services SOAP

HTTP ヘッダー	9-2
セキュリティおよび認証	9-3
設計	9-3
Basic 認証	9-4
問合せの作成	9-6
SOAP エンベロープ	9-6
SOAP ボディ	9-8
SOAP 障害	9-8
Calendar Web Service Language (CWSL)	9-9
NoOp	9-10
Search	9-11
Search の使用	9-11
vQuery	9-12
Create	9-15
Delete	9-17
Modify	9-18
プロパティの変更	9-19
プロパティの追加	9-19
プロパティの削除	9-20
変更可能なプロパティ	9-20
Ping	9-21
xCal	9-22
単純なイベント	9-22
繰返しイベント	9-23
定期的なイベント	9-25
展開されたフィルタ済の定期的 / 繰返しイベント	9-27
メモ	9-28

終日イベント	9-28
休日	9-29
タスク	9-30
X-ORACLE-EVENTTYPE	9-30
優先度のマッピング	9-30

10 Oracle Calendar Web Services のクライアント側 Java 実装

Java クラス	10-2
データのフェッチ	10-3
SOAP 障害および例外	10-5
ローカル時間	10-6

11 Oracle Calendar Web Services のステータス・コード

モジュール・コード	11-2
エラー・コード	11-3

索引

はじめに

このマニュアルでは、Oracle Calendar SDK および Oracle Calendar Web Services ツールキットを使用する際の考慮事項について説明し、参考資料を示します。

対象読者

このマニュアルは、Oracle Calendar SDK または Oracle Calendar Web Services ツールキットを使用して、カレンダーにアクセスするためのカスタム・アプリケーションを作成するプログラマおよび開発者を対象としています。

このマニュアルの構成

このマニュアルは、2 部構成の 11 章から構成されています。

第 I 部「Oracle Calendar SDK」

第 1 章「Calendar SDK の概要」

この章では、Oracle Calendar SDK の概要を説明します。

第 2 章「Calendar SDK 実装時の考慮事項」

この章では、Oracle Calendar SDK の様々な要素の概要と、実装前に考慮する必要がある項目について説明します。

第 3 章「Oracle Calendar SDK 関数リファレンス」

この章では、この開発キットで提供されている関数について詳しく説明します。

第 4 章「Oracle Calendar SDK の構成設定」

この章では、Oracle Calendar SDK で使用可能な構成パラメータについて説明します。

第 5 章「Oracle Calendar SDK のフラグおよび機能」

この章では、各種のフラグおよび機能について説明します。

第 6 章「Calendar SDK のステータス・コード」

この章では、SDK からフィードバックとして提供される各ステータス・コードについて、アルファベット順に簡単に説明します。

第 7 章「Oracle Calendar SDK の FAQ およびトラブルシューティング」

この章では、Oracle Calendar SDK に関する FAQ とトラブルシューティングについて説明します。

第 II 部「Oracle Calendar Web Services ツールキット」

第 8 章「Oracle Calendar Web Services ツールキットの概要」

この章では、Oracle Calendar Web Services ツールキットの概要を説明します。

第 9 章「Oracle Calendar Web Services SOAP」

この章では、Oracle Calendar Web Services ツールキットで eXtensible Markup Language (XML) と Simple Object Access Protocol (SOAP) を使用して、iCalendar オブジェクトを取得し、格納する方法について説明します。

第 10 章「Oracle Calendar Web Services のクライアント側 Java 実装」

この章では、Oracle Calendar Web Services を使用してカレンダー・データにアクセスし、コンテキストに基づく協同作業を実現するための Java クラス・セットの設計について説明します。

第 11 章「Oracle Calendar Web Services のステータス・コード」

この章では、Web サービスのステータス・コードとそれぞれの簡単な説明のリストを示します。

関連ドキュメント

詳細は、Oracle Collaboration Suite ドキュメント・セットの次のドキュメントを参照してください。

- 「Oracle Calendar SDK Javadoc」(製品 CD に収録)
- 「Oracle Calendar Web Services Javadoc」(製品 CD に収録)
- 『Oracle Calendar 管理者ガイド』
- 『Oracle Calendar リファレンス・マニュアル』

表記規則

例では、特に指定しないかぎり、各行の末尾で暗黙的に改行されます。入力行の末尾で [Enter] キーを押す必要があります。

このマニュアルでは、次の表記規則が使用されています。

規則	意味
.	垂直の省略記号は、例に直接関連しない複数の行が省略されていることを示します。
...	水平の省略記号は、例に直接関連しないコードの一部が省略されていることを示します。
太字	太字は、本文中で定義されている用語および用語集に記載されている用語を示します。
< >	山カッコで囲まれた名前は、ユーザーが指定する名前です。
[]	大カッコは、カッコ内の項目を任意に選択することを表します。

第I部

Oracle Calendar SDK

第I部では、Oracle Calendar SDK について説明します。

第I部は、次の章で構成されています。

- [第1章「Calendar SDK の概要」](#)
- [第2章「Calendar SDK 実装時の考慮事項」](#)
- [第3章「Oracle Calendar SDK 関数リファレンス」](#)
- [第4章「Oracle Calendar SDK の構成設定」](#)
- [第5章「Oracle Calendar SDK のフラグおよび機能」](#)
- [第6章「Calendar SDK のステータス・コード」](#)
- [第7章「Oracle Calendar SDK の FAQ およびトラブルシューティング」](#)

Calendar SDK の概要

Oracle Calendar SDK は、C/C++ で作成された関数と、対応する Java 関数から構成され、開発者はこれらを使用して Oracle Calendar とのインタフェースとなるアプリケーションを作成できます。SDK では、ネイティブの C インタフェースを使用することで、C 関数をネイティブでコールできる任意の言語を使用した実装が可能です。

開発者は、標準の iCalendar オブジェクトを使用して会議やイベントを表し、CSDK 関数を使用してカレンダー・データを読み書きするプログラムを作成し、Oracle Calendar Server に情報を格納できます。

作成できるプログラム例としては、カレンダー・サブスクリプションの自動化、Calendar Server とのカスタム・インタフェース、あるいは iCalendar 出力の生成が可能な他のシステムからデータを抽出する移行ユーティリティが含まれます。

SDK の内容

Oracle Calendar SDK には次のものが含まれています。

- API を実装する共有ライブラリ
- C ヘッダー・ファイル
- SDK の Javadoc HTML ドキュメント
- Oracle Calendar クライアント・ライブラリ
- Oracle Calendar ACE（認証、圧縮、暗号化）モジュール
- サンプル / デモ・プログラム

このリリースの新機能

このリリースの Oracle Calendar SDK には、次の新機能が含まれています。

- タスク操作（VTODO オブジェクト）
- 連絡先操作（VCARD オブジェクト）
- Java Native Interface（JNI）を備えた対応する Java 関数
- リモートの代理操作
- 接続プーリング

Java 関数が含まれたことにより、以前はサード・パーティによって行われていた Java 実装が容易になりました。

接続プーリングにより、SDK で使用する接続モデルの構成オプションが追加されました。このため、様々なアプリケーション（特に Web ベースおよびマルチスレッド環境）を実装する際のリソースの使用量と使用効率が大きく改善され、既存の接続の再利用が促進されます。

最適な使用例

Oracle Calendar SDK は、Oracle Calendar Desktop Client および Oracle Calendar Server アプリケーションの設計時に使用されるコア関数のサブセットです。既存の Oracle Calendar アプリケーションを拡張するユーティリティおよびアプリケーションを迅速に開発するために使用するもので、既存のインタフェースやロジックのかわりに使用するツールではありません。

SDK はポータル、FCGI ベースのアプリケーション、さらには1日のイベントを表示するような単純なコマンドライン・スクリプトとの統合に使用されています。SDK は、特定の明確な目的を実現する場合に使用してください。

Calendar SDK 実装時の考慮事項

この章では、Oracle Calendar SDK の実装時に考慮が必要な様々な項目について説明します。

- キャラクタ・セット
- iCalendar のサポート
- セキュリティ・モデル
- アラーム
- ユーザー識別情報
- データ・ストリーム
- アクセス制御

キャラクタ・セット

すべての SDK 関数は、UTF-8 のエンコード・テキストでのみ機能します。SDK 関数で使用する文字列は、UTF-8 である必要があります。また、SDK からの文字列もすべて UTF-8 で返されます。UTF-8 の詳細は、RFC 2279 を参照してください。

iCalendar のサポート

Oracle Calendar SDK では、(RFC 2445 の規定に従って) iCalendar 形式を使用して、カレンダー・データを処理します。SDK を使用して保存された iCalendar 情報は、後で取り出すことができます。ただし、SDK のこのリビジョンでは現在サポートされていない iCalendar データもあります。特に、VFREEBUSY および VJOURNAL コンポーネントはサポートされていません。

iCalendar 入力

iCalendar データは、ネイティブのデータ構造にマップされます。これらのプロパティのデータは、常に完全に保存されるわけではありません。一部のプロパティは、インスタンス単位ではなくイベント単位で格納されるため、1 つの値のみが保持されます。この項では、最も一般的なプロパティで使用するマッピングについて説明します。

ATTACH

現在このプロパティは無視されます。

ATTENDEE

ATTENDEE プロパティを格納する際、出席者のプロパティとカレンダー・ユーザーとの関連付けが試みられます。レガシー関数 `CAPL_StoreEvents` は、出席者のアドレスと提供された各ハンドルのアドレスとを比較して、カレンダー・ユーザーを識別します。

`CSDK_StoreEvents` は、Calendar Server で参照を実行し、対応するカレンダー・ユーザーを検索します。`CSDK_StoreEvents` を使用すると、カレンダー・ユーザー以外にも「外部参加者」として追加されます。

CATEGORIES

`CAPL_StoreEvents` を使用する場合、CATEGORIES プロパティを使用して、イベント・タイプを指定します。認識される値は、APPOINTMENT、DAILY NOTE、DAY EVENT および HOLIDAY です。`CSDK_StoreEvents` を使用する場合、CATEGORIES 値はサーバーに格納され、様々な `CSK_FetchEvents` 関数によって返されます（プロパティ `X-ORACLE-EVENTTYPE` は、`CSDK_StoreEvents` でイベント・タイプの指定に使用され、同じ値が認識されます）。

CLASS

このプロパティは、アクセス・レベルにマップされます。iCalendar と Calendar Server のアクセス・レベルは、次のようにマッピングされます。

iCalendar から Calendar Server へのマッピング：

- PUBLIC → PUBLIC
- PRIVATE → PERSONAL
- CONFIDENTIAL → CONFIDENTIAL
- X-ORACLE-NORMAL → NORMAL (CSDK_StoreEvents を使用)
- X-CST-NORMAL → NORMAL (CAPI_StoreEvents を使用)

このプロパティはイベント単位で格納されます。

DESCRIPTION

イベントの詳細に設定されます。32KB を超えると切り捨てられます。このプロパティは、CAPI_StoreEvents のコール時はイベント単位で、CSDK_StoreEvents のコール時はインスタンス単位で格納されます。

DTSTART、DTEND および DURATION

DTEND がある場合、イベントの所要時間の計算に使用されます。実際の終了時間は格納されません。イベントの時間は分で測定されるため、開始時間と所要時間の seconds コンポーネントは 0 に設定されます。

LOCATION

location フィールドに格納されます。CorporateTime Server 5.0 以下を使用する場合、この値は 32 バイトに切り捨てられます。

PRIORITY

このプロパティは、Calendar Server の 5 つの優先度値の 1 つにマップされます。このプロパティはイベント単位で格納されます。

RRULE

CSDK_StoreEvents を使用する場合、繰返しルール、およびルールに対する例外が Calendar Server に格納されます。レガシー関数 CAPI_StoreEvents を使用する場合、繰返しルールは、イベント作成時に展開され、繰返しルール自体は Calendar Server に格納されません。

STATUS

値が TENTATIVE の場合、仮のイベントを示します。他の値（CANCELLED も含む）は、仮のイベント以外を示します。

SUMMARY

このプロパティは、イベント・タイトルにマップされます。CorporateTime Server 5.0 以下を使用する場合、この値は、64 バイトに切り捨てられます。

UID

格納されているデータで UID が指定されていない場合、Calendar Server によって UID が割り当てられます。CAPI_StoreEvents を使用する場合、サーバーによって割り当てられた UID は、CAPI_GetLastStoredUIDs をコールして読み取ることができます。CSDK_StoreEvents を使用する場合、UID は CSDKRequestResult を介して返されます。

VALARM

カレント・ユーザーは、自分のリマインダをフェッチし、格納できます。他のユーザーのアラームにアクセスすることはできません。

iCalendar 出力

SDK または他の Oracle Calendar クライアントで作成されたデータをフェッチする場合、次のプロパティを使用できます。Oracle 固有の拡張（X-ORACLE-...）を含む他のプロパティが返される場合もあります。データをフェッチする際、返されるプロパティのリストを制限できます。制限することによって、パフォーマンスが向上する場合があります。

ATTENDEE

プロパティは、各 ATTENDEE に対して生成されます。出席者およびユーザー情報からパラメータ PARTSTAT、ROLE、CUTYPE および CN を取得できます。

CATEGORIES

CAPI_FetchEventsBy... をコールすると、このプロパティにイベント・タイプが返されます。CSDK_FetchEventsBy... は（SDK または他のクライアントを使用して格納された）ユーザー指定の値を返します。

CLASS

イベント・アクセス・レベル。アクセス・レベルは次のようにマップされます。

- PUBLIC → PUBLIC
- PRIVATE → PERSONAL
- CONFIDENTIAL → CONFIDENTIAL
- NORMAL → PUBLIC

DESCRIPTION

CAPI_FetchEventsBy... をコールすると、(ネイティブ・クライアント側から見た) イベント詳細が返されます。CSDK_FetchEventsBy... は、インスタンス固有の詳細がある場合はその詳細を、ない場合はイベント詳細を返します。

DURATION または DTEND

イベントの所要時間またはイベントの終了時間。

DTSTART

イベントの開始時間。

LOCATION

イベントの場所。

ORGANIZER

イベントの所有者。

PRIORITY

イベントの優先度。

RESOURCES

CAPI_FetchEventsBy... をコールすると、このプロパティにはリソースである出席者全員の名前が含まれます。CSDK_FetchEventsBy... は、CSDK_StoreEvents を使用して格納された文字列または他の Oracle Calendar クライアントによって格納された文字列を返します。

RRULE

サーバーに格納されている繰返しルール。CAPI_FetchEventsBy... のコール時には使用できません。

STATUS

仮のイベントのステータスは **TENTATIVE** です。仮のイベント以外の場合、**CONFIRMED** としてマークされます。他の **Status** 値は生成されません。

SUMMARY

イベントまたはインスタンスのタイトル。

UID

ユーザー指定の **UID**（イベント作成時に設定されている場合）。ない場合は、サーバーによって生成された **UID**。

VALARM

ネイティブのリマインダから変換されます（カレント・ユーザーの場合のみ）。

vCard のサポート

Oracle Calendar SDK では、vCard 形式を使用して連絡先データを処理します。SDK を使用して保存された vCard 情報は、後で取り出すことができます。バージョン 2.1 と 3.0 の両方がサポートされています。

セキュリティ・モデル

セキュリティ・モデルは、イベントの格納とフェッチという 2 つの部分に分けられます。これらは、異なるセキュリティ・パラダイムで処理されます。

イベントの所有者は、そのイベントのプロパティを追加または削除できます。カレンダー・ユーザーに **ATTENDEE** プロパティを作成するときに、そのユーザーにハンドルが提供されている場合、プロパティは、そのパラメータのデフォルト値を使用して作成されます。イベントの所有者は、そのプロパティのパラメータを変更できません。変更できるのはプロパティに対応付けられたユーザーだけです。

ユーザーが **ATTENDEE** プロパティを更新するときに、他のイベント・データを変更しようとしてもエラーは返されません。ただし、変更は行われません。

ユーザーは、他のユーザーからの出席依頼を拒否することができます。その場合、そのユーザーには **ATTENDEE** プロパティは作成されず、そのユーザーのハンドルのステータスは、出席依頼が拒否されたことを示します。これは、リソースをダブルブッキングしようとした場合にも発生することがあります。

イベントをフェッチする場合、セキュリティ・モデルは、イベントの iCalendar 分類に基づきます。ユーザーは、他のユーザーに、様々なイベント・クラスに対して異なるアクセス・レベルを付与します。アクセス・レベルには、読取りアクセスなし、イベントの開始時間および終了時間の読取りのみ、イベントの全詳細の読取り、という 3 つのレベルがあります。

SDK を使用してイベントをフェッチする場合、一部のイベントについては UID、DTSTART、DURATION および DTEND プロパティのみが返されます。他のすべてのイベントは表示されないか、それらのプロパティすべてが返されます。

Oracle Calendar SDK では、ユーザーが、この動作を制御するセキュリティ・レコードを変更することはできません。

アラーム

アラームは、各ユーザーの個人設定とみなされているため、ユーザー同士が互いに他のユーザーのアラームを読み取ったり、書込みを行うことはできません。ユーザーは、互いにアラームを読み取ることができないため、互いのカレンダーのアラーム範囲に基づいてフェッチを行うことはできません。ユーザーは自分が出席するイベントにアラームを設定できます。したがって、フェッチするユーザーごとに、同じイベントに異なるアラームが設定される場合があります。

ユーザー識別情報

Oracle Calendar SDK では、ユーザーは `userID` 文字列、またはユーザー名などを指定する検索文字列で識別されます。文字列の書式は柔軟で、コール元はオプションのパラメータを多数指定できます。サーバー構成によっては、これらのオプションの一部（ノード ID など）が識別情報文字列に必要な場合があります。ログイン時およびハンドルの取得時の両方で、同じユーザー識別情報文字列の書式が使用されますが、その両方ですべてのオプションが適用されるわけではありません。

サーバーにリソースとしてログインする方法はサポートされていませんが、リソースの代理として作業することは可能です。

すべてのオプションは、キーと値のペアを使用して指定します。文字列全体は、これらのペアの集合です。`userID` は、ASCII の「?」文字で拡張データと分離されます。そのすぐ後ろの文字は、後に続く各フィールド値のペアのデリミタです。デリミタには、数字、文字、ヌル文字、「*」および「=」を除く任意の ASCII 文字を使用できます。文字列の残りは、選択したデリミタで区切られたフィールドと値のペアで構成されます。文字列は、デリミタの後ろのヌル文字で終了します。フィールド値のペアは、フィールド名、等号(=)、値で構成されます。値は、デリミタ文字、ヌル文字を含まない文字列で、ユーザー識別情報文字列の場合にはその他にスラッシュ (/) 文字を含まない文字列です。

たとえば、フィールド名 `G` は名前、`S` は姓を表すとしてします。ユーザーを識別するための正しい文字列の例を次に示します。`userID` は指定されていないため、オプションのパラメータを使用してユーザーを検索します。（検索によって複数の一致項目が見つかった場合、コール元にはエラーが返されます。`userID` がある場合は、`userID` を使用してユーザーを指定するのが最良の方法です）。`userID` がない場合も、拡張文字列と `userID` を分離する疑問符「?」を使用します。そのすぐ後ろの文字（この場合はスラッシュ「/」）は、デリミタとして使用されています。文字列の最後の文字はデリミタ文字で、ヌル文字で終了する点に注意してください。

```
?/S=Bunny/G=Bugs/
```

ユーザーの識別に使用するフィールドは、ワイルドカードとして使用される「*」で終了することができます。これは、ノードの指定には使用できません。次の例でも、前のユーザーと一致します。

```
?/S=Bu*/G=Bugs/
```

特定の検索文字列に複数のユーザーが一致した場合、SDK コールからエラーが返されます。

リソースには異なる名前構造があります。これらは、リソース名を示す 1 つのフィールド RS で識別されます。

次の文法 (RFC 2234 で説明されている ABNF 形式) は、正しいログオン文字列です。二重引用符内の値では大 / 小文字が区別される点が、ABNF と異なります。つまりフィールド名は大文字である必要があります。また、どのような場合も、1 つの文字列で同じデリミタを使用する必要があります。

```
logon-string = userid "?" [ DELIMITER ext-string ] %x00
```

```
userid = *( ALPHA / DIGIT / "-")
```

```
ext-string = 1*( field )
```

```
field = ( node / company-domain / surname / given-name / initials / generation /
org-unit / organization / country / admin / private / resource-name ) DELIMITER
```

特定のフィールドを 2 度以上指定した場合、冗長ですが誤りではありません。ただし、最後のフィールドのみが使用されます。

```
node = "ND=" node-number
```

```
node-number = 1*DIGIT
```

```
company-domain = "CD=" 1*VALUE-CHAR
```

```
surname = "S=" 1*VALUE-CHAR ["*"]
```

```
given-name = "G=" 1*VALUE-CHAR ["*"]
```

```
initials = "I=" 1*VALUE-CHAR ["*"]
```

```
generation = "X=" 1*VALUE-CHAR ["*"]
```

```
org-unit = ( "OU1" / "OU2" / "OU3" / "OU4" ) "=" 1*VALUE-CHAR ["*"]
```

```
organization = "O=" 1*VALUE-CHAR ["*"]
```

```
country = "C=" 1*VALUE-CHAR ["*"] DELIMITER
```

```
admin = "A=" 1*VALUE-CHAR ["*"] DELIMITER
```

```
private = "P=" 1*VALUE-CHAR ["*"] DELIMITER
```

```
resource-name = "RS=" 1*VALUE-CHAR ["*"] DELIMITER
```

```
DELIMITER = %x01-%x29 / %x2B-%x2F / %x3A-%x3C / %x3E-%x40 / %x5B-%x60 / %x7B-%x7F
```

DELIMITER は VALUE-CHAR として使用できないことに注意してください。

```
VALUE-CHAR = %x01-29 / %x2B-2E / %x30-7F
```


データ・ストリーム

デフォルトでは、SDK は入出力の両方で、MIME (RFC 2045 を参照) にカプセル化された iCalendar および vCard オブジェクトを扱います。1 つの要求で、カレンダーのリストからデータをフェッチできます。そのような要求への返信は、内部が個別の MIME パートに分かれた、リスト内のカレンダーごとに 1 つの iCalendar オブジェクトで構成されています。つまり、calendarA と calendarB からのイベントの要求は、次の形式の MIME ストリームとして返されます。

```
... MIME envelope
--MIMEBOUNDARYasdfasdf
Content-type: text/calendar
Content-Transfer-Encoding: quoted-printable
BEGIN:VCALENDAR
... events from calendarA
END:VCALENDAR

--MIMEBOUNDARYasdfasdf
Content-type: text/calendar
Content-Transfer-Encoding: quoted-printable
BEGIN:VCALENDAR
... events from calendarB
END:VCALENDAR

--MIMEBOUNDARYasdfasdf--
```

iCalendar オブジェクトの順序は、要求リスト内のカレンダーの順序と一致します。要求の結果が空の解決セットだった場合、戻りストリームは空の iCalendar オブジェクトになります。カレンダーにエラーがある場合、そのカレンダーに対応する iCalendar 返信オブジェクトは空になります。

フェッチが成功した場合、VCALENDAR には、それぞれが要求されたプロパティを含む、多数の VEVENT コンポーネントが含まれている場合があります (該当する場合)。iCalendar では、これらの異なるコンポーネントに、同じイベントの異なるインスタンスに関する情報を入れることができます。返されるデータに次のいずれかの方法を適用して、インスタンス特有の情報を提供することができます。

- 各インスタンスのデータを、異なる DTSTART を持つ異なる VEVENT コンポーネントに入れることができます。
- 複数のインスタンスのデータを 1 つの VEVENT に入れ、プロパティ RRULE、RDATE、EXRULE および EXDATE を使用して各インスタンスを識別することができます。
- 上の 2 つの方法を組み合わせ、開始時間以外のすべてのプロパティを共有する複数のインスタンスを 1 つの VEVENT コンポーネントにまとめ、そのように処理された多数のコンポーネントを返すことができます。

返される **DISTART** プロパティは、カレンダー・ストアに含まれるイベントの最初のインスタンスの開始時間ではなく、そのプロパティが含まれている **VEVENT** コンポーネントで最初に識別されるインスタンスの開始時間を示すことに注意してください。さらに、カレンダーに入れて返される **VEVENT** コンポーネントの数は、イベントのインスタンス数とは関係ありません。したがって、イベントをフェッチする際、繰り返し識別プロパティを要求しない場合、インスタンスがいくつ存在するか、および返される各プロパティがどのインスタンスに適用されるのかを判断する方法はありません。

格納時、SDK に提供されるデータは、1 つの MIME パートに 1 つの **VCALENDAR** コンポーネントで構成されている必要があります。カレンダーには、多くの **VEVENT** オブジェクトを含めることができますが、これらはすべて 1 つのイベントに関する情報である必要があります。たとえば、次の入力は無効です。

```
Content-type: text/calendar
Content-Transfer-Encoding: 7bit
BEGIN:VCALENDAR
VERSION:2.0
BEGIN:VEVENT
event properties
END:VEVENT
BEGIN:VEVENT
event properties
END:VEVENT
END:VCALENDAR
```

アクセス制御

SDK からのデータへのアクセスは、Calendar Server によって制御されます。これは、要求側の識別情報と、要求されるデータや操作に基づきます。SDK は、任意のプロパティを組み合わせて読取りを要求できるインタフェースを提供します。読取り権限のないユーザーが要求したプロパティは返されません。

ユーザーには、自分に出席依頼されたイベントまたは自分が所有するイベントを変更する権限しかありません。ユーザーがイベントの所有者である場合、イベントを変更する完全な権限があります（他のユーザーの出席情報の変更を除く）。イベントに出席依頼されている場合、承諾やアラームなど、自分自身の出席に関連する情報を変更する限定的な権限が与えられます。

変更の許可が与えられていないプロパティを変更しようとしても無視されます。

特定の予定表については、イベントを変更または作成しようとするエラーが発生することがあります。これらのエラーは、提供されている一連のステータス値を使用して返され、残りの操作は続行できます。

Oracle Calendar SDK 関数リファレンス

この章では、Oracle Calendar SDK で使用できる便利な関数について詳しく説明します。

SDK 関数

次の関数の詳細を説明します。

- [CSDK_SetConfigFile](#)
- [CSDK_CreateSession](#)
- [CSDK_DestroySession](#)
- [CSDK_GetCapabilities](#)
- [CSDK_Authenticate](#)
- [CSDK_ConfigureACE](#)
- [CSDK_Connect](#)
- [CSDK_ConnectAsSysop](#)
- [CSDK_Deauthenticate](#)
- [CSDK_Disconnect](#)
- [CSDK_SetIdentity](#)
- [CSDK_DestroyHandle](#)
- [CSDK_DestroyMultipleHandles](#)
- [CSDK_GetHandle](#)
- [CSDK_GetHandleInfo](#)
- [CSDK_CreateCallbackStream](#)
- [CSDK_CreateFileStreamFromFilenames](#)
- [CSDK_CreateMemoryStream](#)
- [CSDK_DestroyMultipleStreams](#)
- [CSDK_DestroyStream](#)
- [CSDK_DeleteContacts](#)
- [CSDK_FetchContactsByQuery](#)
- [CSDK_FetchContactsByUID](#)
- [CSDK_StoreContacts](#)
- [CSDK_DeleteEvents](#)
- [CSDK_FetchEventsByAlarmRange](#)
- [CSDK_FetchEventsByRange](#)

- CSDK_FetchEventsByUID
- CSDK_StoreEvents
- CSDK_DeleteTasks
- CSDK_FetchTasksByAlarmRange
- CSDK_FetchTasksByRange
- CSDK_FetchTasksByUID
- CSDK_StoreTasks
- CSDK_AddConditionToQuery
- CSDK_CreateQuery
- CSDK_DestroyQuery
- CSDK_DestroyResult
- CSDK_GetFirstFailure
- CSDK_GetFirstParseError
- CSDK_GetFirstResult
- CSDK_GetNextFailure
- CSDK_GetNextParseError
- CSDK_GetNextResult
- CSDK_GetStatusLevels
- CSDK_GetStatusString

CSDK_SetConfigFile

```
CAPIStatus CSDK_SetConfigFile ( const char *   in_configFileName,
                                const char *   in_logFileName
                                )
```

この関数をコールすると、SDK でエラー・ロギングを制御する構成設定や、このマニュアルの第 4 章「Oracle Calendar SDK の構成設定」にリストした他の構成パラメータを読み取ることができます。

この関数をコールする場合は、プロセスの最初の SDK 関数としてコールし、各スレッドでコールしないようにする必要があります。

パラメータ：

- `in_configFileName`: 構成ファイルのファイル名を含むヌル文字で終了する文字列。
- `in_logFileName`: ログ・メッセージを書き込むファイルの名前。このファイルの作成や書込みを行えない場合、出力はカレント・ディレクトリの `Console.log` というファイルに送られます。

返される結果：

CAPIStatus

戻り値：

CAPI_STAT_API_NULL: 入力パラメータの 1 つが NULL でした。

CAPI_STAT_CONFIG_CANNOT_OPEN: `in_configFileName` を開けませんでした。

対応する Java メソッド

`oracle.calendar.sdk.Api.init()`

例 3-1 構成ファイルの設定

次の内容でファイル `capi.ini` を作成します。

```
[LOG]
log_activity = true
log_modulesinclude = { CAPI }
```

次に、`CSDK_SetConfigFile` をコールします。

```
CAPIStatus stat = CSDK_SetConfigFile("capi.ini", "capi.log");
```

SDK のアクティビティ・レベルのログがオンになり、出力は `capi.log` に記録されます。

CSDK_CreateSession

```
CAPIStatus CSDK_CreateSession (  CAPIFlag    in_flags,  
                                CAPISession *  out_session  
                                )
```

新しいセッションを作成します。

パラメータ :

- in_flags: CSDK_FLAG_NONE
- out_session: 新規セッション

必要なクリーンアップ :

セッションは、CSDK_DestroySession() を介して破棄する必要があります。

返される結果 :

CAPIStatus

対応する Java メソッド :

oracle.calendar.sdk.Session コンストラクタ

CSDK_DestroySession

```
CAPIStatus CSDK_DestroySession (  CAPISession *  io_session )
```

セッションを破棄します。

パラメータ :

- io_session: 破棄するセッションを示すポインタ。出力で CAPI_SESSION_INITIALIZER を示します。

返される結果 :

CAPIStatus

対応する Java メソッド :

oracle.calendar.sdk.Session ファイナライザ

CSDK_GetCapabilities

```
CAPIStatus CSDK_GetCapabilities ( CAPISession    in_session,
                                  CAPICapabilityID in_capabilityID,
                                  CAPIFlag        in_flags,
                                  const char **    out_value
                                )
```

このリリースの SDK と Calendar Server、またはそのいずれかの情報を返します。

パラメータ :

- `in_session`: セッション。NULL の場合、サーバー機能を要求できません。
- `in_capabilityID`: 機能の ID (ctapi.h の `CAPI_CAPAB_*` を参照)。
- `in_flags`: 現時点では `CSDK_FLAG_NONE`。
- `out_value`: このパラメータに情報が返されます。値は、読取り専用文字列として返され、同じセッションを使用する次の CAPI コールまで有効です。

変更 :

CAPI 2.5: `in_capabilityID` の型が、long から `CAPICapabilityID` に変更されました。

対応する Java メソッド :

`oracle.calendar.sdk.Session.getCapabilities`

CSDK_Authenticate

```
CAPIStatus CSDK_Authenticate ( CAPISession    in_session,
                                CAPIFlag        in_flags,
                                const char *    in_user,
                                const char *    in_password
                              )
```

カレンダー・ユーザーを認証します。

データの格納またはフェッチを行うコールの前に実行する必要があります。

パラメータ :

- `in_session`: セッション。
- `in_flags`: 動作を変更するビット・フラグ。現時点では `CSDK_FLAG_NONE` に設定する必要があります。
- `in_user`: スル文字で終了する文字列を使用します。
- `in_password`: ユーザーのパスワード。NULL でもかまいません。

返される結果：

CAPIStatus

例 3-2 カレンダ・ユーザーの認証 - ノード指定なし

calserver.acme.com のデフォルト・ポートで実行されているサーバーに接続し、デフォルトの ACE 設定を使用して、userID を keithm として認証します。

(ノードを指定しない場合、指定したホストでマスター・ノードまたはデフォルト・ノードが構成されている必要があります。)

```
{
    CAPISession mySession = CSDK_SESSION_INITIALIZER;
    CAPIStatus myStatus = CSDK_CreateSession(&mySession);
    if (myStatus == CAPI_STAT_OK)
    {
        myStatus = CSDK_Connect(mySession, CAPI_FLAG_NONE, "calserver.acme.com");
    }
    if (myStatus == CAPI_STAT_OK)
    {
        myStatus = CSDK_Authenticate(mySession,
                                     CAPI_FLAG_NONE,
                                     "keithm",
                                     "abcdefg");
    }
}
```

例 3-3 カレンダ・ユーザーの認証

calserver.acme.com のデフォルト・ポートで実行されているサーバーに接続し、デフォルトの ACE 設定を使用して、ユーザー Keith MacDonald として認証します。

```
{
    CAPISession mySession = CSDK_SESSION_INITIALIZER;
    CAPIStatus myStatus = CSDK_CreateSession(&mySession);
    if (myStatus == CAPI_STAT_OK)
    {
        myStatus = CSDK_Connect(mySession, CAPI_FLAG_NONE, "calserver.acme.com");
    }
    if (myStatus == CAPI_STAT_OK)
    {
        myStatus = CSDK_Authenticate(mySession,
                                     CAPI_FLAG_NONE,
                                     "?/S=MacDonald/G=Keith/ND=200/",
                                     "abcdefg");
    }
}
```

例 3-4 カレンダ・ユーザーの認証 - デフォルト・ポート

calserver.acme.com のデフォルト・ポートで実行されているサーバーに接続し、デフォルトの ACE 設定を使用して、ノード 200 で userID を **ernesth** として認証します。

```
{
    CAPISession mySession = CSDK_SESSION_INITIALIZER;
    CAPIStatus myStatus = CSDK_CreateSession(&mySession);
    if (myStatus == CAPI_STAT_OK)
    {
        myStatus = CSDK_Connect(mySession, CAPI_FLAG_NONE, "calserver.acme.com");
    }
    if (myStatus == CAPI_STAT_OK)
    {
        myStatus = CSDK_Authenticate(mySession,
                                     CAPI_FLAG_NONE,
                                     "ernesth?/ND=200/",
                                     "abcdefg");
    }
}
```

例 3-5 カレンダ・ユーザーの認証 - Kerberos

calserver.acme.com のポート 12345 で実行されているサーバーに接続し、gssapi:kerberos5 認証を使用します。

```
{
    CAPISession mySession = CSDK_SESSION_INITIALIZER;
    CAPIStatus myStatus = CSDK_CreateSession(&mySession);
    if (myStatus == CAPI_STAT_OK)
    {
        myStatus = CSDK_Connect(mySession, CAPI_FLAG_NONE, "calserver.acme.com:12345");
    }
    if (myStatus == CAPI_STAT_OK)
    {
        myStatus = CSDK_ConfigureACE(mySession,
                                     CAPI_FLAG_NONE,
                                     "gssapi:kerberos5",
                                     NULL,
                                     NULL);
    }
    if (myStatus == CAPI_STAT_OK)
    {
        myStatus = CSDK_Authenticate(mySession,
                                     CAPI_FLAG_NONE,
                                     "", // don't pass in user string
                                     ""); // don't pass in password
    }
}
```

必要なクリーンアップ:

CSDK_Authenticate コールの間に CSDK_Deauthenticate をコールする必要があります。

対応する Java メソッド:

oracle.calendar.sdk.Session.authenticate()

CSDK_ConfigureACE

```
CAPIStatus CSDK_ConfigureACE (  CAPISession    in_session,
                                CAPIFlag      in_flags,
                                const char *   in_authMechanism,
                                const char *   in_compMechanism,
                                const char *   in_encrMechanism
                                )
```

この関数は、指定されたセッションを、SDK クライアントと Calendar Server の間で特定の ACE（認証、圧縮、暗号化）メカニズムを使用するように構成します。

この関数がコールされない場合、Calendar Server で設定されているデフォルトのメカニズムが使用されます。

NULL 値を指定して、3 種類のメカニズムのいずれかにサーバーのデフォルト・メカニズムを選択することもできます。

パラメータ:

- in_session: セッション
- in_flags: CSDK_FLAG_NONE
- in_authMechanism: 認証メカニズムの名前 ("cs-standard"、"gssapi:kerberos5"、NULL など)
- in_compMechanism: 圧縮メカニズムの名前 ("cs-simple"、"NONE"、NULL など)
- in_encrMechanism: 暗号化メカニズムの名前 ("cs-acipher1"、"NONE"、NULL など)

返される結果:

CAPIStatus

対応する Java メソッド:

oracle.calendar.sdk.Session.configureACE()

CSDK_Connect

```
CAPIStatus CSDK_Connect (  CAPISession  in_session,
                           CAPIFlag     in_flags,
                           const char *  in_host
                           )
```

カレンダー・サービスとの接続を確立します。

パラメータ：

- in_session: セッション。
- in_flags: ビット・フラグ。
- in_host: Calendar Server ホスト（オプションのポート番号付き。
"calserver.acme.com"、"calserver.acme.com:12345" など）。host[:port] の後ろに、オプションで ?/CD=<calendar domain>/ を付けることができます。

返される結果：

CAPIStatus

例 3-6 Calendar Server への接続

通常的使用方法： Calendar Server の calserver.acme.com に接続します。この接続を使用して、マスター・ノードで認識される任意のユーザーとして認証できます。

```
{
    CAPIStatus myStatus = CAPI_STAT_OK;
    CAPISession mySession = CAPI_SESSION_INITIALIZER;
    myStatus = CSDK_CreateSession(CAPI_FLAG_NONE, &mySession);
    if (myStatus == CAPI_STAT_OK)
    {
        myStatus = CSDK_connect(mySession, CAPI_FLAG_NONE, "calserver.acme.com");
    }
}
```

必要なクリーンアップ：

CSDK_Disconnect をコールして、サーバー接続を解放する必要があります。

対応する Java メソッド：

oracle.calendar.sdk.Session.connect()

CSDK_ConnectAsSysop

```
CAPIStatus CSDK_ConnectAsSysop ( CAPISession    in_session,
                                CAPIFlag        in_flags,
                                const char *    in_host,
                                const char *    in_nodeId,
                                const char *    in_password
                                )
```

Calendar SYSOP としてログオンします。

ログオン後、Calendar SYSOP は、CSDK_SetIdentity() をコールして同じノード上の任意のユーザーの識別情報を使用できます。

Calendar SYSOP としてログオンしている間は、マスター・ノードおよびカレンダー・ドメイン機能を使用できないため、必ずノードを指定する必要があります。

セッションで ACE メカニズムが構成されている場合、これらは無視されます。Calendar Server からの管理デフォルト ACE 設定が、すべての Calendar SYSOP 接続に使用されます。

Calendar SYSOP 認証は、リリース 5.3 以上のサーバーでのみ使用できます。提供されているホストがこの機能をサポートしていない場合、エラーが返されます。SDK を介した Calendar SYSOP ログオンを拒否するように Calendar Server を構成できます。この方法でログオンした場合、セキュリティ・エラーが返されます。

Calendar SYSOP が使用できる操作は、次の操作に限定されます。

- CSDK_Disconnect() のコールによる切断
- CSDK_SetIdentity() のコールによる別のユーザーへの識別情報の切替え

識別情報を別のユーザーに設定すると、すべての操作が、そのユーザーがログインしているかのように実行されます。

in_host パラメータの書式については、「[CSDK_Connect](#)」の項を参照してください。

パラメータ :

- in_session: セッション。
- in_flags: 動作を変更するビット・フラグ。現時点では CSDK_FLAG_NONE に設定する必要があります。
- in_host: Calendar Server のホスト名 (オプションのポート番号)。
- in_nodeId: Calendar SYSOP として接続するためのノード ID。ノードの別名は現在サポートされていません。
- in_password: Calendar SYSOP のパスワード。

返される結果 :

CAPIStatus

対応する Java メソッド:`oracle.calendar.sdk.Session.connectAsSysop()`**関連項目:**[「CSDK_SetIdentity」](#)**CSDK_Deauthenticate**

```
CAPIStatus CSDK_Deauthenticate ( CAPISession  in_session,
                                  CAPIFlag     in_flags
                                )
```

カレント・ユーザーの認証を解除します。

認証解除後もサーバー接続は開かれたままなので、再認証に使用できます。

CSDK_Disconnect() がコールされるか、セッションが破棄されるまで、サーバー接続は保持されます。

パラメータ:

- `in_session`: セッション。
- `in_flags`: 動作を変更するビット・フラグ。現時点では CSDK_FLAG_NONE に設定する必要があります。

返される結果:

CAPIStatus

対応する Java メソッド:`oracle.calendar.sdk.Session.deauthenticate()`**CSDK_Disconnect**

```
CAPIStatus CSDK_Disconnect ( CAPISession  in_session,
                              CAPIFlag     in_flags
                            )
```

Calendar Server から切断します。

パラメータ:

- `in_session`: セッション。
- `in_flags`: 動作を変更するビット・フラグ。現時点では CSDK_FLAG_NONE に設定する必要があります。

返される結果：

CAPIStatus

対応する Java メソッド：

oracle.calendar.sdk.Session.disconnect()

CSDK_SetIdentity

```
CAPIStatus CSDK_SetIdentity (  CAPISession    in_session,
                               const char *    in_user,
                               CAPIFlag       in_flags
                               )
```

認証済のユーザーが、別のカレンダー・ユーザーまたはリソースにかわって作業できます。

後続のコールでは、認証済のユーザーに代理権限が付与されている必要があります。

in_user パラメータの書式は、CSDK_Authenticate 関数と同じです。認証済のユーザーは、ユーザー名に NULL を使用して、元の識別情報に戻ることができます。

Calendar SYSOP としてログインした場合 (CSDK_ConnectAsSysop)、代理権限は無視され、任意のカレンダー・ユーザーまたはリソースとして作業できます。すべてのカレンダー操作は、ユーザーにかわって代理が行ったのではなく、ユーザー自身が行ったように見えます。

パラメータ：

- in_session: セッション
- in_user: X400 または uid として作業するユーザー (またはリソース)
- in_flags: 現時点では CSDK_FLAG_NONE

返される結果：

CAPIStatus

例 3-7 別のユーザーにかわっての作業

```
myStatus = CSDK_SetIdentity(mySession, "dantea", CAPI_FLAG_NONE);
myStatus = CSDK_SetIdentity(mySession, "?/S=Alighieri/G=Dante/", CAPI_FLAG_NONE);
myStatus = CSDK_SetIdentity(mySession, "?/RS=Conference Room/ND=1234/", CAPI_FLAG_NONE);
```

変更:

CAPI 2.5: リソース名が正確に一致する必要があります。(これまでは、文字列の最後に暗黙のワイルドカードが使用されていました。)

SDK 9.0.4: 代理権限を使用して別のノードでユーザーのかわりに作業する際、`SetIdentity` を使用できます。これは、`CSDK_ConnectAsSysop()` を使用して開かれた接続には適用されません。

対応する Java メソッド:

`oracle.calendar.sdk.Session.setIdentity()`

CSDK_DestroyHandle

```
CAPIStatus CSDK_DestroyHandle ( CAPISession    in_session,
                                CAPIHandle *    io_handle
                                )
```

`CSDK_GetHandle()` から返された 1 つのハンドルを破棄します。

パラメータ:

- `in_session`: セッション・ハンドル
- `io_handle`: (`CSDK_GetHandle` から返された) 破棄するハンドル

返される結果:

`CAPIStatus`

例 3-8 ハンドルの破棄

```
{
    CAPIHandle h1 = CSDK_HANDLE_INITIALIZER;
    CSDK_GetHandle(mySession, "arthur", CSDK_FLAG_NONE, &h1);
    ...
    CSDK_DestroyHandle(mySession, &h1);
}
```

対応する Java メソッド:

なし。`oracle.calendar.sdk.Handle` ファイナライザがハンドルを破棄します。

CSDK_DestroyMultipleHandles

```

CAPISStatus CSDK_DestroyMultipleHandles (  CAPISession    in_session,
                                           CAPIHandle *   io_handles,
                                           int           in_numHandles,
                                           CAPIFlag      in_flags
                                           )

```

CSDK_GetHandle() のコールによって返された複数のハンドルを破棄します。

パラメータ :

- in_session: ログイン・セッション・ハンドル
- io_handles: (CSDK_GetHandle から返された) 破棄するハンドルの配列
- in_numHandles: ハンドル配列のサイズ
- in_flags: ビット・フラグ (現時点ではなし。CSDK_FLAG_NONE に設定。)

返される結果 :

CAPISStatus

例 3-9 複数のハンドルの破棄

```

{
    CAPIHandle h1 = CSDK_HANDLE_INITIALIZER;
    CAPIHandle h2 = CSDK_HANDLE_INITIALIZER;
    CSDK_GetHandle(mySession, "arthur", CSDK_FLAG_NONE, &h1);
    CSDK_GetHandle(mySession, "tim...", CSDK_FLAG_NONE, &h2);
    ...
    CAPIHandle handles[] = {h1, h2};
    CSDK_DestroyMultipleHandles(mySession, handles, 2, CSDK_FLAG_NONE);
}

```

対応する Java メソッド :

なし。oracle.calendar.sdk.Handle ファイナライザがハンドルを破棄します。

CSDK_GetHandle

```
CAPIStatus CSDK_GetHandle (  CAPISession  in_session,
                             const char *   in_user,
                             CAPIFlag      in_flags,
                             CAPIHandle *   out_handle
                             )
```

この関数は、特定ユーザーのカレンダ・ストアへのハンドルを返します。

このハンドルを使用して、後続のコールでこの予定表の項目にアクセスできます。エラーが返された場合、CAPIHandle は割り当てられず、クリーンアップは必要ありません。

in_user 文字列は、CSDK_Authenticate で使用する文字列と同じ書式です。

in_user が NULL の場合、カレント・ユーザーへのハンドルが返されます。

ユーザーの識別情報を使用していない Calendar SYSOP は、この関数を使用できません。

パラメータ：

- in_session: ログイン・セッション・ハンドル。
- in_user: CSDK_Authenticate に対して定義されたユーザー。NULL を使用できます。その場合、カレント・ユーザーへのハンドルが返されます。
- in_flags: ビット・フラグ（現時点ではなし。CSDK_FLAG_NONE に設定。）。
- out_handle: in_user のハンドル。入力時は NULL を示している必要があります。

返される結果：

CAPIStatus

戻り値：

- CAPI_STAT_OK
- CAPI_STAT_DATA_USERID
- CAPI_STAT_SERVICE_MEM
- CAPI_STAT_SERVICE_FILE
- CAPI_STAT_SERVICE_NET
- CAPI_STAT_API_FLAGS
- CAPI_STAT_API_NULL
- CAPI_STAT_API_HANDLE
- CAPI_STAT_API_SESSION
- CAPI_STAT_LIBRARY

必要なクリーンアップ:

この関数では、`CSDK_DestroyHandle` のコールによってクリーンアップする必要があるハンドルが割り当てられます。エラーが返された場合、ハンドルは割り当てられず、クリーンアップは必要ありません。

例 3-10 UserID を使用したハンドルの取得

userID が "ivoa" のユーザーのハンドルを取得します。

```
{
    CAPIHandle shrubber = CSDK_HANDLE_INITIALIZER;
    stat = CSDK_GetHandle(mySession, "ivoa", CAPI_FLAG_NONE, &shrubber);
}
```

例 3-11 姓名を使用したハンドルの取得

Arnold Layne という名前（姓が Layne、名前が Arnold）のユーザーのハンドルを取得します。

```
{
    CAPIHandle arnold = CSDK_HANDLE_INITIALIZER;
    stat = CSDK_GetHandle(mySession, "?/S=Layne/G=Arnold/", CAPI_FLAG_NONE, &arnold);
}
```

例 3-12 リソースのハンドルの取得

ノード 1234 でリソース名が keg のハンドルを取得します。

```
{
    CAPIHandle keg = CSDK_HANDLE_INITIALIZER;
    stat = CSDK_GetHandle(mySession, "?/RS=keg/ND=1234/", CAPI_FLAG_NONE, &keg);
}
```

例 3-13 カレント・ユーザーのハンドルの取得

カレント・ユーザーのハンドルを取得します。

```
{
    CAPIHandle currUser = CSDK_HANDLE_INITIALIZER;
    stat = CSDK_GetHandle(mySession, NULL, CAPI_FLAG_NONE, &currUser);
}
```

変更:

CAPI 2.5: リソース名が正確に一致する必要があります。（これまでは、文字列の最後に暗黙のワイルドカードが使用されていました。）

対応する Java メソッド:

oracle.calendar.sdk.Session.getHandle()

CSDK_GetHandleInfo

```
CAPIStatus CSDK_GetHandleInfo (  CAPISession    in_session,
                                CAPIHandle     in_handle,
                                CAPIFlag       in_flags,
                                const char **   out_info
                                )
```

この関数は、指定されたハンドルの予定表に関する情報を返します。

3 種類の情報を、**in_flags** の値に基づいて返します。情報は、読取り専用文字列を示すポインタとして返されます。

CAPI_HANDLE_TYPE は、ハンドルのタイプを示します。**user** または **resource** で、予定表のタイプを示します。**CAPI_HANDLE_NAME** は、予定表の所有者、つまりリソースの名前を返します。一連のフィールドと値のペアが「/」で区切られています。この文字列の前に「?」が付いている場合、**CSDK_GetHandle** に渡すことができる適切な書式です。この書式の説明は、このマニュアルの「ユーザー識別情報」の項を参照してください。

CAPI_HANDLE_MAILTO は、予定表が属するユーザーの電子メール・アドレスを返します。すべてのユーザーの電子メール・アドレスが **Calendar Server** に設定されているわけではありません。電子メール・アドレスが設定されていないユーザーの場合、エラー (**CAPI_STAT_DATA_EMAIL_NOTSET**) が返されます。

パラメータ:

- **in_session:** ログイン・セッション・ハンドル
- **in_handle:** 情報を取得するハンドル
- **in_flags:** **CAPI_HANDLE_TYPE**、**CAPI_HANDLE_NAME** または **CAPI_HANDLE_MAILTO**
- **out_info:** 読取り専用ハンドル情報

返される結果:

CAPIStatus

変更:

CAPI 2.5: サーバーで電子メール・アドレスが設定されていない場合、**CAPI_STAT_DATA_EMAIL_NOTSET** を返します。

例 3-14 ログイン・ユーザー名の印刷

```

{
    CAPIHandle  loginUser = CSDK_HANDLE_INITIALIZER;
    const char * fullName = NULL;
    stat = CSDK_GetHandle(mySession, NULL, CSDK_FLAG_NONE, &loginUser);
    stat = CSDK_HandleInfo(mySession, loginUser, CAPI_HANDLE_NAME, &fullName);
    cout << "Currently logged in as " << fullName << endl;
    CSDK_DestroyHandle(mySession,
                      &loginUser);
}

```

例 3-15 電子メール・アドレスの印刷

Doctor Winston の電子メール・アドレスの印刷は、次のようになります。

```

{
    CAPIHandle  doctor = CSDK_HANDLE_INITIALIZER;
    const char * email = NULL;
    stat = CAPI_GetHandle(mySession, "drwinston", CSDK_FLAG_NONE, &doctor);
    stat = CAPI_HandleInfo(mySession, doctor, CAPI_HANDLE_MAILTO, &email);
    cout << "drwinston's email address is " << email << endl;
    CSDK_DestroyHandle(mySession,
                      &doctor);
}

```

対応する Java メソッド:

- oracle.calendar.sdk.Session.Handle.getEmail()
- oracle.calendar.sdk.Session.Handle.getName()
- oracle.calendar.sdk.Session.Handle.getType()

CSDK_CreateCallbackStream

```

CAPIStatus CSDK_CreateCallbackStream ( CAPISession    in_session,
                                       CAPIStream *    out_stream,
                                       CAPICallback    in_sendCallback,
                                       void *          in_sendUserData,
                                       CAPICallback    in_recvCallback,
                                       void *          in_recvUserData,
                                       CAPIFlag        in_flags
                                       )

```

コールバック・ストリームを使用して、SDK との間でデータを送受信できます。

アクション（送信、受信）ごとに SDK に C 関数ポインタが提供され、SDK はこれをコールしてデータを読み書きします。

CSDK_Store...() のコール中、SDK は関数 `in_sendCallback` をコールして、値 `in_sendUserData` を渡します（通常コールバック関数で使用するコンテキストの格納に使用）。

CSDK_Fetch...() のコール中、SDK は関数 `in_rcvCallback` をコールして、値 `in_rcvUserData` を渡します（通常コールバック関数で使用するコンテキストの格納に使用）。

どちらのタイプのコールバック関数でも、同じ関数シグネチャを使用します。

```
typedef int (*CAPICallback) (
    void *    in_userData,    // user-defined data (the value supplied in CAPI_CreateCallbackStream)
    char *    io_data,        // buffer to read or write
    size_t    in_dataSize,    // the number of characters to read or write
    size_t *  out_datSize);   // the number of characters read or written
```

コールバックからの戻り値は次のとおりです。

- 送信コールバック：
 - `CAPI_CALLBACK_CONTINUE`: ストリームから読み取る追加データがあります。
 - `CAPI_CALLBACK_DONE`: これ以上ストリームから読み取るデータはありません。
 - 正の整数: エラーが発生しました。この正の整数は、ビット 5 で返される `CAPIStatus` の一部として、値 `CAPI_STAT_API_CALLBACK_ERROR` とともに返されます。
- 受信コールバック：
 - `CAPI_CALLBACK_CONTINUE`: エラーはありません。
 - 正の整数: エラーが発生しました（ストリームがこれ以上のデータを受信できないなど）。この正の整数は、ビット 5 で返される `CAPIStatus` の一部として、値 `CAPI_STAT_API_CALLBACK_ERROR` とともに返されます。

SDK が受信コールバックへのデータの書き込みを終了すると、`in_dataSize == 0` を使用してコールバックがコールされます。

多くのアプリケーションでは、コールバック・ストリームよりメモリー・ストリームまたはファイル・ストリームを使用した方が簡単です。

パラメータ：

- `in_session`: ログイン・セッション・ハンドル。
- `out_stream`: 出力で新しいストリームを示します。
- `in_sendCallback`: データ送信コールバック。
- `in_sendUserData`: `in_sendCallback` に渡される値。
- `in_rcvCallback`: データ受信コールバック。

- `in_recvUserData`: `in_recvCallback` に渡される値。
- `in_flags`: ビット・フラグ（現時点では `CSDK_FLAG_NONE` に設定する必要がある）。

必要なクリーンアップ:

この関数によって返されるストリームは、`CSDK_DestroyStreams()` をコールして破棄する必要があります。

返される結果:

`CAPIStatus`

戻り値:

`CAPI_STAT_API_NULL`: 指定されたコールバックはどちらも `NULL` でした。

関連項目:

「[CSDK_CreateMemoryStream](#)」、「[CSDK_CreateFileStreamFromFilenames](#)」

対応する Java メソッド:

なし。Java API では、データの送受信には、`String` および `StringBuffer` オブジェクトのみを使用します。

CSDK_CreateFileStreamFromFilenames

```
CAPIStatus CSDK_CreateFileStreamFromFilenames ( CAPISession    in_session,
                                                CAPIStream *   out_stream,
                                                const char *   in_readFileName,
                                                const char *   in_readMode,
                                                const char *   in_writeFileName,
                                                const char *   in_writeMode,
                                                CAPIFlag       in_flags
                                                )
```

SDK がファイルとの間で読み書きを行うためのファイル・ストリームを作成します。

パラメータ:

- `in_session`: ログイン・セッション・ハンドル。
- `out_stream`: 出力で新しいストリームを示します。
- `in_readFileName`: 読取り先のファイル名。
- `in_readMode`: `in_readFileName` を開く間に `fopen()` に渡すモード。
- `in_writeFileName`: 書込み先のファイル名。

- `in_writeMode`: `in_writeFileName` を開く間に `fopen()` に渡すモード。
- `in_flags`: ビット・フラグ（現時点では `CSDK_FLAG_NONE` に設定する必要がある）。

必要なクリーンアップ:

この関数によって返されるストリームは、`CSDK_DestroyStreams` をコールして破棄する必要があります。

返される結果:

`CAPIStatus`

戻り値:

- `CAPI_STAT_SERVICE_FILE_MODE`: 無効なモードが渡されました。
- `CAPI_STAT_SERVICE_FILE_OPEN`: ファイルを開けませんでした。

例 3-16 ファイル `events.ics` からのイベントの格納

```
CAPIStream myInputStream = NULL;
CAPIStatus status = CSDK_CreateFileStreamFromFilenames(mySession,
                                                         &myInputStream,
                                                         "events.ics",
                                                         "rb",
                                                         NULL,      // no output file
                                                         NULL,      // no output file mode
                                                         CSDK_FLAG_NONE);

if (status == CAPI_STAT_OK)
{
    status = CAPI_StoreEvent(mySession,
                             myHandles,
                             numHandles,
                             handleStatus,
                             CAPI_STORE_REPLACE,
                             myInputStream);
    CSDK_DestroyStreams(mySession,
                        &myInputStream,
                        1,
                        CSDK_FLAG_NONE);
}
```


例 3-17 イベントをフェッチし、ファイル myAgenda.ics に書き込む

```

CAPIStream myOutputStream = NULL;
CAPIStatus status = CSDK_CreateFileStreamFromFileNames(mySession,
                                                         &myOutputStream,
                                                         NULL, // no input file
                                                         NULL, // no input file mode
                                                         "myAgenda.ics",
                                                         "wb",
                                                         CSDK_FLAG_NONE);

if (status == CAPI_STAT_OK)
{
    status = CAPI_FetchEventsByRange(mySession,
                                     myHandles,
                                     numHandles,
                                     handleStatus,
                                     CAPI_FLAG_NONE,
                                     "20020722T000000",
                                     "20020722T235900",
                                     NULL,
                                     0,
                                     myOutputStream);

    CSDK_DestroyStreams(mySession,
                       &myOutputStream,
                       1,
                       CSDK_FLAG_NONE);
}

```

対応する Java メソッド:

なし。Java API では、データの送受信には、String および StringBuffer オブジェクトのみを使用します。

CSDK_CreateMemoryStream

```

CAPIStatus CSDK_CreateMemoryStream ( CAPISession   in_session,
                                     CAPIStream *   out_stream,
                                     const char *   in_readBuffer,
                                     const char **   out_writeBufferPtr,
                                     CAPIFlag       in_flags
                                     )

```

メモリー・ストリームでは、データ・バッファを使用して、アプリケーションと SDK の間でデータをやり取りします。

多くの場合、このストリームを使用するのが最も簡単です。

読取りバッファは、`CSDK_Store...` のコール中に SDK によって読み取られ、書込みバッファは、`CSDK_Fetch...` のコール中に SDK によって書き込まれます。読取りバッファはアプリケーションによって制御されますが、書込みバッファの割当ておよび解放は SDK が行います。書込みバッファは、メモリー・ストリームの破棄時に SDK によって解放されます。

パラメータ：

- `in_session`: セッション。
- `out_stream`: 出力で新しいストリームを示します。
- `in_readBuffer`: SDK の読取り先バッファ。
- `out_writeBufferPtr`: このアドレスは、SDK の書込み先バッファを示します。
- `in_flags`: ビット・フラグ（現時点では `CSDK_FLAG_NONE` に設定する必要があります）。

必要なクリーンアップ：

この関数によって返されるストリームは、[CSDK_DestroyStream](#) をコールして破棄する必要があります。

返される結果：

`CAPIStatus`

戻り値：

`CAPI_STAT_API_NULL`: 指定されたバッファはどちらも NULL でした。

例 3-18 バッファ events からのイベントの格納

```
const char events[] = "MIME-Version: 1.0\r\n"\r\n"Content-Type: text/calendar\r\n"\r\n"Content-Transfer-Encoding: quoted-printable\r\n\r\n"\r\n"BEGIN:VCALENDAR\r\n"\r\n"VERSION:2.0\r\n"\r\n"...etc\r\n"END:VCALENDAR\r\n";\r\n\r\nCAPIStream myInputStream = NULL;\r\nCAPIStatus status = CSDK_CreateMemoryStream(mySession,\r\n                                             &myInputStream,\r\n                                             events,\r\n                                             NULL, // no write buffer\r\n                                             CSDK_FLAG_NONE);
```

```

if (status == CAPI_STAT_OK)
{
    status = CAPI_StoreEvent(mySession,
                             myHandles,
                             numHandles,
                             handleStatus,
                             CAPI_STORE_REPLACE,
                             myInputStream);

    CSDK_DestroyStreams(mySession,
                        &myInputStream,
                        1,
                        CSDK_FLAG_NONE);
}

```

例 3-19 イベントをフェッチし、バッファに書き込む

```

const char * todaysEvents = NULL;
CAPIStream myOutputStream = NULL;
CAPIStatus status = CSDK_CreateMemoryStream(mySession,
                                             &myOutputStream,
                                             NULL, // no read buffer
                                             &todaysEvents,
                                             CSDK_FLAG_NONE);

if (status == CAPI_STAT_OK)
{
    status = CAPI_FetchEventsByRange(mySession,
                                     myHandles,
                                     numHandles,
                                     handleStatus,
                                     CAPI_FLAG_NONE,
                                     "20020722T000000",
                                     "20020722T235900",
                                     NULL,
                                     0,
                                     myOutputStream);

    if (status == CAPI_STAT_OK)
    {
        printf("Today's events:%s", todaysEvents);
    }
    CSDK_DestroyStreams(mySession,
                        &myOutputStream,
                        1,
                        CAPI_FLAG_NONE);
}

```

CSDK_DestroyMultipleStreams

```
CAPIStatus CSDK_DestroyMultipleStreams (  CAPISession    in_session,
                                           CAPIStream *   in_streams,
                                           int            in_numStreams,
                                           CAPIFlag       in_flags
                                           )
```

この関数は、様々な CSDK_Create*Stream 関数によって作成されたストリームを破棄します。

パラメータ：

- in_session: ストリームが関連付けられているセッション。
- in_streams: 破棄するストリームの配列。
- in_numStreams: in_streams 内の破棄するストリーム数。
- in_flags: 動作を変更するビット・フラグ。現時点では CSDK_FLAG_NONE に設定する必要があります。

返される結果：

CAPIStatus

対応する Java メソッド：

なし。Java API では、データの送受信には、String および StringBuffer オブジェクトのみを使用します。

CSDK_DestroyStream

```
CAPIStatus CSDK_DestroyStream (  CAPISession    in_session,
                                  CAPIStream *   io_stream
                                  )
```

この関数は、様々な CSDK_Create*Stream 関数のいずれかによって作成された 1 つのストリームを破棄します。

パラメータ：

- in_session: ストリームが関連付けられているセッション
- io_stream: 破棄するストリーム

返される結果：

CAPIStatus

対応する Java メソッド:

なし。Java API では、データの送受信には、String および StringBuffer オブジェクトのみを使用します。

CSDK_DeleteContacts

```
CAPIStatus CSDK_DeleteContacts ( CAPISession      in_session,
                                  CAPIFlag          in_flags,
                                  CAPIUIDSet         in_UIDs,
                                  CSDKRequestResult * out_requestResult
                                )
```

一連の UID で指定された vCard を削除します。

ユーザーの識別情報を使用していない Calendar SYSOP は、この関数を使用できません。

パラメータ:

- in_session: ログイン・セッション・ハンドル。
- in_flags: CSDK_FLAG_NONE。
- in_UIDs: 削除する vCard の UID を含む文字列のヌル文字で終了する配列。
- out_requestResult: NULL 以外の場合、トランザクションの詳細な結果が含まれます。

返される結果:

CAPIStatus

戻り値:

- CAPI_STAT_API_HANDLE_NULL: セッションが NULL でした。
- CAPI_STAT_API_NULL: in_UIDs が NULL でした。

対応する Java メソッド:

oracle.calendar.sdk.Session.deleteContacts()

CSDK_FetchContactsByQuery

```

CAPISStatus CSDK_FetchContactsByQuery (  CAPISession      in_session,
                                          CAPIFlag         in_flags,
                                          CSDKQuery         in_query,
                                          const char **      in_requestProperties,
                                          CAPIStream        in_stream,
                                          CSDKRequestResult * out_requestResult
                                          )

```

この関数は、問合せで指定された条件を満たす連絡先をフェッチします。

vCard は、in_sendStream を介して返されます。デフォルトでは MIME 書式です。フェッチされた各 vCard は、個別の MIME パートに含まれます。キャラクタ・セットは UTF-8 です。

ストリームを MIME でカプセル化しない場合は、フラグ CSDK_FLAG_STREAM_NOT_MIME を渡します。

パラメータ：

- in_session: セッション・ハンドル。
- in_flags: ビット・フラグ：
 - CSDK_FLAG_NONE
 - CSDK_FLAG_STREAM_NOT_MIME: 出力を MIME コンテナに入れません。
- in_query: 検索条件を含む問合せオブジェクト。
- in_requestProperties: 連絡先の特定の vCard プロパティのみをフェッチするには、プロパティ名の配列を渡します。配列は、NULL（ゼロ）ポインタまたは空（長さがゼロ）の文字列で終了する必要があります。空の配列（NULL または空の文字列）を使用すると、使用可能なすべてのプロパティが返されます。
- in_stream: SDK の書込み先のストリーム。
- out_requestResult: NULL 以外の場合、トランザクションの詳細な結果が含まれます。

必要なクリーンアップ：

要求は、CSDK_DestroyRequestResult を使用して破棄する必要があります。

返される結果：

CAPISStatus

対応する Java メソッド：

oracle.calendar.sdk.Session.fetchContactsByQuery()

CSDK_FetchContactsByUID

```

CAPISStatus CSDK_FetchContactsByUID (  CAPISession      in_session,
                                       CAPIFlag         in_flags,
                                       CAPIUIDSet        in_UIDs,
                                       const char **      in_requestProperties,
                                       CAPIStream         in_stream,
                                       CSDKRequestResult * out_requestResult
                                       )

```

`in_session` を介して認証された接続から vCard をフェッチします。

フェッチされた vCard は、`in_sendStream` を介して MIME 書式で返されます。フェッチされた各 vCard は、個別の MIME パートに含まれます。キャラクタ・セットは UTF-8 です。

ストリームを MIME でカプセル化しない場合は、フラグ `CSDK_FLAG_STREAM_NOT_MIME` を渡します。

ユーザーの識別情報を使用していない Calendar SYSOP は、この関数を使用できません。

パラメータ：

- `in_session`: ログイン・セッション・ハンドル。
- `in_flags`: ビット・フラグ：
 - `CSDK_FLAG_NONE`
 - `CSDK_FLAG_STREAM_NOT_MIME`: 出力を MIME コンテナに入れません。
- `in_UIDs`: フェッチする vCard の UID を含む文字列のヌル文字で終了する配列。
- `in_requestProperties`: 連絡先の特定の vCard プロパティのみをフェッチするには、プロパティ名の配列を渡します。配列は、NULL（ゼロ）ポインタまたは空（長さがゼロ）の文字列で終了する必要があります。空の配列（NULL または空の文字列）を使用すると、使用可能なすべてのプロパティが返されます。
- `in_stream`: SDK の書込み先のストリーム。
- `out_requestResult`: NULL 以外の場合、トランザクションの詳細な結果が含まれます。

必要なクリーンアップ：

結果は、`CSDK_DestroyRequestResult` を使用して破棄する必要があります。

返される結果：

CAPIStatus

戻り値：

- CAPI_STAT_API_HANDLE_NULL: セッションが NULL でした。
- CAPI_STAT_API_STREAM_NULL: ストリームが NULL でした。
- CAPI_STAT_API_NULL: in_UIDSet が NULL でした。
- CAPI_STAT_API_BADPARAM: in_UIDCount が 0 でした。

対応する Java メソッド：

oracle.calendar.sdk.Session.fetchContactsByUID()

CSDK_StoreContacts

```
CAPIStatus CSDK_StoreContacts ( CAPISession      in_session,
                                CAPIFlag         in_flags,
                                CAPIStream        in_stream,
                                CSDKRequestResult * out_requestResult
                                )
```

in_session によって認証された接続を介してサーバーに vCard を格納します。

vCard は、CAPIStream を使用して渡す必要があります。

デフォルトでは、着信ストリームは、MIME でカプセル化された vCard とみなされます。MIME でカプセル化されていないストリームを格納する場合、フラグ CSDK_FLAG_STREAM_NOT_MIME を指定します。

vCard のバージョン 2.1 と 3.0 がサポートされています。

複数の vCard を格納する場合、各 vCard を個別の MIME パートに含める必要があります。vCard を含む MIME パートには、Content-Type: text/x-vcard ヘッダーが必要です。MIME パートでサポートされているキャラクタ・セットは、UTF-8 と US-ASCII のみです。

使用できるモードは次のとおりです。

- CSDK_FLAG_STORE_REPLACE: サーバーにすでに存在する vCard を完全に置き換えます。指定された vCard の UID プロパティ内に含まれている UID を読み取り、サーバーからその連絡先を削除して、新しい連絡先を格納します。サーバー上のその UID に連絡先が存在しない場合、エラーが返されます。
- CSDK_FLAG_STORE_MODIFY: サーバー上に存在する連絡先を新しい vCard に更新します。次にその UID の vCard が更新されます。サーバー上の vCard に含まれるプロパティの中で、渡された vCard に存在するプロパティはすべて、渡された vCard のプロパティ値を含むように変更されます。また、サーバーの vCard には存在せず、渡され

た vCard には存在するすべてのプロパティが、サーバーの vCard に追加されます。サーバーに存在し、渡された vCard には存在しない他のプロパティは無視されます。

- CSDK_FLAG_STORE_REMOVE: サーバー上の連絡先がフェッチされ、渡された vCard 内に含まれるプロパティがフェッチした vCard から削除され、フェッチした vCard がサーバーに格納されます。
- CSDK_FLAG_STORE_IMPORT: このモードは、UID を使用して、連絡先がすでにサーバーに存在するかどうかをチェックします。存在する場合、CSDK_FLAG_STORE_REPLACE が渡された場合と同じように動作します。存在しない場合は、CSDK_FLAG_STORE_CREATE が渡された場合と同じように動作します。

注意:

指定されたフラグは、指定された各 vCard に対して使用されます。結果は、CSDKRequestResult に書き込まれます。

このバージョンの SDK では、サーバーに連絡を追加する際、提供された UID を保持できません。CSDKRequestResult には、サーバーによって割り当てられた UID が含まれ、この UID を使用して格納されている vCard を参照できます。

CSDKRequestResult には、vCard ストリームの解析エラーに関する情報が含まれている場合があります。

ユーザーの識別情報を使用していない Calendar SYSOP は、この関数を使用できません。

パラメータ:

- in_session: セッション。
- in_flags: 動作を変更するビット・フラグ:
 - CSDK_FLAG_STORE_CREATE: 指定された UID を持つ連絡先が存在しない場合、連絡先を作成します。存在する場合、エラーを返します。
 - CSDK_FLAG_STORE_REPLACE: サーバー上の連絡先をこのコピーに完全に置き換えます。オブジェクトが存在しない場合、エラーを返します。
 - CSDK_FLAG_STORE_IMPORT: 連絡先が存在する場合は、CSDK_FLAG_STORE_REPLACE と、存在しない場合は、CSDK_FLAG_STORE_CREATE と等しくなります。
 - CSDK_FLAG_STORE_MODIFY: 既存連絡先の指定されたプロパティのみを変更します。
 - CSDK_FLAG_STORE_REMOVE: 指定されたプロパティを削除します。
 - CSDK_FLAG_STREAM_NOT_MIME: 着信ストリームは MIME ラッパー内に入れられません。
- in_stream: vCard を含むストリーム。

- `out_requestResult`: 値が収められる `RequestResult` を示すポインタ（この情報を受け取らない場合は `NULL` を渡す）。

返される結果：

`CAPIStatus`

戻り値：

- `CAPI_STAT_API_HANDLE_NULL`: セッションが `NULL` でした。
- `CAPI_STAT_API_STREAM_NULL`: ストリームが `NULL` でした。
- `CAPI_STAT_DATA_VCARD_DUPERROR`: サーバーにすでに存在する `vCard` を `CSDK_FLAG_STORE_CREATE` モードで格納しようとした。
- `CAPI_STAT_DATA_UID_NOTFOUND`: プロパティを更新、置換または削除しようとしたが、渡された `vCard` の `UID` が見つかりませんでした。
- `CAPI_STAT_API_BADPARAM`: 無効なフラグ。または複数のフラグが設定されています。

例 3-20 連絡先の格納

```
stat = CSDK_StoreContacts(mySession,
                          myStream,
                          CSDK_FLAG_STORE_IMPORT,
                          &myRequestResult);
```

例 3-21 vCard を使用した連絡先の格納

```
strcpy(outVCard,
"MIME-Version: 1.0¥015¥012¥
Content-Type: multipart/mixed;¥015¥012¥
boundary=¥"-----CA94974D4D8713DE5B12E6CD¥"¥015¥012¥
¥015¥012¥
This is a multi-part message in MIME format.¥015¥012¥
-----CA94974D4D8713DE5B12E6CD¥015¥012¥
Content-Type: text/x-vcard; charset=UTF-8;¥015¥012¥
name=¥"example.vcf¥"¥015¥012¥
Content-Disposition: attachment;¥015¥012¥
filename=¥"example.vcf¥"¥015¥012¥
Content-Transfer-Encoding: quoted-printable¥015¥012¥
¥015¥012¥
BEGIN:VCARD¥015¥012¥
URL:http://www.somewebsite.com¥015¥012¥
ORG:oracle;windows;¥015¥012¥
TITLE:worker¥015¥012¥
EMAIL;TYPE=INTERNET:someone@somewhere.com¥015¥012¥
ADR;TYPE=WORK:;;who knows;snodown;qc;h1l 2H1;Canada¥015¥012¥
NOTE;ENCODING=QUOTED-PRINTABLE;:This is a note¥015¥012¥
```

```

N;ENCODING=QUOTED-PRINTABLE:Last;First;Middle¥015¥012¥
FN;ENCODING=QUOTED-PRINTABLE:First Middle Last¥015¥012¥
REV:20011105T145136Z¥015¥012¥
VERSION:2.1¥015¥012¥
END:VCARD¥015¥012¥
¥015¥012¥
-----CA94974D4D8713DE5B12E6CD--¥015¥012¥015¥012");
//
CAPIStatus stat;
//
stat = CSDK_CreateMemoryStream(mySession,
                                &myStream,
                                outVCard,
                                NULL,
                                CAPI_FLAG_NONE);

//
stat = CSDK_StoreContacts(mySession,
                           myStream,
                           CSDK_FLAG_STORE_IMPORT,
                           &myRequestResult);

```

必要なクリーンアップ:

要求は、CSDK_DestroyRequestResult を使用して破棄する必要があります。

対応する Java メソッド:

oracle.calendar.sdk.Session.storeContacts()

CSDK_DeleteEvents

```

CAPIStatus CSDK_DeleteEvents (  CAPISession      in_session,
                                CAPIFlag          in_flags,
                                CAPIUIDSet         in_uids,
                                const char *       in_RECURRENCEID,
                                int                in_modifier,
                                CSDKRequestResult * out_requestResult
                                )

```

この関数は、指定されたイベントを削除します。

削除するには、イベント所有者として作業する必要があります。CAPI_DeleteEvent() と異なり、出席者への出席依頼を取り消すのではなく、イベントを削除します。個々のインスタンス（または範囲）は、{in_RECURRENCEID, in_modifier} を使用して削除できます。ただし、この場合、使用できる UID は 1 つだけです。

パラメータ :

- `in_session`: ログイン・セッション・ハンドル。
- `in_flags`: ビット・フラグ。
- `in_UIDs`: 削除するイベントの UID を含む文字列の配列。配列は、NULL (ゼロ) ポインタまたは空 (長さがゼロ) の文字列で終了する必要があります。
- `in_RECURRENCEID`: 1 つのイベントのすべてのインスタンスを削除するには、NULL または空の文字列を渡します。個々の (または一連の) インスタンスを削除するには、DATE または DATE-TIME 書式でイベントの 1 つのインスタンスを識別する iCalendar の `recurrence-id` を指定します。
- `in_modifier`: `in_RECURRENCEID` を使用して `recurrence-id` を指定した場合、この修飾子によって、指定したインスタンスを削除するか一連のインスタンスを削除するかが決まります。値は次のとおりです。
 - `CAPI_THISINSTANCE`
 - `CAPI_THISANDPRIOR`
 - `CAPI_THISANDFUTURE`
- `out_requestResult`: NULL 以外の場合、トランザクションの詳細な結果が含まれます。

必要なクリーンアップ :

要求は、`CSDK_DestroyRequestResult` を使用して破棄する必要があります。

返される結果 :

`CAPIStatus`

対応する Java メソッド :

`oracle.calendar.sdk.Session.deleteEvents()`

CSDK_FetchEventsByAlarmRange

```

CAPISStatus CSDK_FetchEventsByAlarmRange ( CAPISession    in_session,
                                           CAPIFlag      in_flags,
                                           CAPIHandle *   in_agendas,
                                           const char *   in_start,
                                           const char *   in_end,
                                           const char **  in_requestProperties,
                                           CAPIStream     in_stream,
                                           CSDKRequestResult * out_requestResult
                                           )

```

この関数は、指定された時間範囲内にトリガーされるリマインダを持つイベントをフェッチします。

時間範囲の終わりは除外されます。

パラメータ:

- **in_session:** ログイン・セッション・ハンドル。
- **in_flags:** ビット・フラグ:
 - **CSDK_FLAG_NONE:** デフォルトの動作。
 - **CSDK_FLAG_FETCH_AGENDA_ATTENDEE_ONLY:** 参照している予定表の ATTENDEE のみを取得または返します。
 - **CSDK_FLAG_FETCH_COMBINED:** 予定表ごとに 1 つの VCALENDAR を返すのではなく、1 つの VCALENDAR にすべてのイベントを入れて返します。高速に処理されます。
 - **CSDK_FLAG_FETCH_EXCLUDE_HOLIDAYS:** 休日をフェッチしません。
 - **CSDK_FLAG_FETCH_EXCLUDE_DAILYNOTES:** メモをフェッチしません。
 - **CSDK_FLAG_FETCH_EXCLUDE_DAYEVENTS:** 終日イベントをフェッチしません。
 - **CSDK_FLAG_FETCH_EXCLUDE_APPOINTMENTS:** 予定をフェッチしません。
 - **CSDK_FLAG_FETCH_EXCLUDE_ACCEPTED:** 承諾イベントをフェッチしません。
 - **CSDK_FLAG_FETCH_EXCLUDE_DECLINED:** 辞退されたイベントをフェッチしません。
 - **CSDK_FLAG_FETCH_EXCLUDE_UNCONFIRMED:** 未確定のイベントをフェッチしません。
 - **CSDK_FLAG_FETCH_LOCALTIMES:** すべての日付と時間をユーザーのタイムゾーンで返します (ユーザーのタイムゾーン設定項目は Calendar Server に格納されています)。

- `CSDK_FLAG_FETCH_DO_NOT_EXPAND_RRULE`: 繰返しルールを展開しません。範囲内にトリガーが予定されているアラームを持つインスタンスだけでなく、イベント全体が返されます。
- `CSDK_FLAG_STREAM_NOT_MIME`: `iCalendar` を MIME コンテナに入れません。
- `in_agendas`: イベントを検索する予定表。このパラメータは `CAPIHandles` の配列で、ゼロ値で終了する必要があります (`CAPI_HANDLE_INITIALIZER`)。NULL 値、つまり空の配列の場合、カレント・ユーザーの予定表が検索されます。
- `in_start`: 日時範囲の始まり。次のいずれかの書式を使用できます。
 - `DATE`。(20020928 など。)
 - `DATE-TIME`。変動時間 (20020929T120000 など) または UTC 時間 (20020929T170000Z など) のいずれかです。変動時間では、ユーザーのタイムゾーンを使用します (ユーザーのタイムゾーン設定項目は `Calendar Server` に格納されています)。
 - `DURATION`。所要時間表記を使用した相対的な日数または日数と時間 (将来の 5 日と 3 時間の場合は `+PT5DT3H`、過去の 1 週間の場合は `-PT1W` など)。
- `in_end`: 日時範囲の終わり。`in_start` に示した書式のいずれかを使用できます。
- `in_requestProperties`: イベントの特定の `iCalendar` プロパティのみをフェッチするには、プロパティ名の配列を渡します。配列は、NULL (ゼロ) ポインタまたは空 (長さがゼロ) の文字列で終了する必要があります。空の配列 (NULL または空の文字列) を使用すると、使用可能なすべてのプロパティが返されます。
- `in_stream`: SDK の書込み先のストリーム。
- `out_requestResult`: NULL 以外の場合、トランザクションの詳細な結果が含まれます。

必要なクリーンアップ:

要求は、`CSDK_DestroyRequestResult` を使用して破棄する必要があります。

返される結果:

`CAPIStatus`

対応する Java メソッド:

`oracle.calendar.sdk.Session.fetchEventsByAlarmRange()`

CSDK_FetchEventsByRange

```

CAPIStatus CSDK_FetchEventsByRange ( CAPISession    in_session,
                                     CAPIFlag       in_flags,
                                     CAPIHandle *    in_agendas,
                                     const char *    in_start,
                                     const char *    in_end,
                                     const char **   in_requestProperties,
                                     CAPIStream     in_stream,
                                     CSDKRequestResult * out_requestResult
                                     )

```

この関数は、指定された時間範囲内に発生するイベントをフェッチします。

パラメータ:

- **in_session:** ログイン・セッション・ハンドル。
- **in_flags:** ビット・フラグ:
 - **CSDK_FLAG_NONE:** デフォルトの動作。
 - **CSDK_FLAG_FETCH_AGENDA_ATTENDEE_ONLY:** 参照している予定表の ATTENDEE のみを取得または返します。
 - **CSDK_FLAG_FETCH_COMBINED:** 予定表ごとに 1 つの VCALENDAR を返すのではなく、1 つの VCALENDAR にすべてのイベントを入れて返します。高速に処理されます。
 - **CSDK_FLAG_FETCH_EXCLUDE_HOLIDAYS:** 休日をフェッチしません。
 - **CSDK_FLAG_FETCH_EXCLUDE_DAILYNOTES:** メモをフェッチしません。
 - **CSDK_FLAG_FETCH_EXCLUDE_DAYEVENTS:** 終日イベントをフェッチしません。
 - **CSDK_FLAG_FETCH_EXCLUDE_APPOINTMENTS:** 予定をフェッチしません。
 - **CSDK_FLAG_FETCH_EXCLUDE_ACCEPTED:** 承諾イベントをフェッチしません。
 - **CSDK_FLAG_FETCH_EXCLUDE_DECLINED:** 辞退されたイベントをフェッチしません。
 - **CSDK_FLAG_FETCH_EXCLUDE_UNCONFIRMED:** 未確定のイベントをフェッチしません。
 - **CSDK_FLAG_FETCH_LOCALTIMES:** すべての日付と時間をユーザーのタイムゾーンで返します (ユーザーのタイムゾーン設定項目は Calendar Server に格納されています)。

- `CSDK_FLAG_FETCH_DO_NOT_EXPAND_RRULE`: 繰返しルールを展開しません。範囲内に該当するインスタンスだけでなく、イベント全体が返されます。
- `CSDK_FLAG_STREAM_NOT_MIME`: `iCalendar` を MIME コンテナに入れません。
- `in_agendas`: イベントを検索する予定表。このパラメータは `CAPIHandles` の配列で、ゼロ値で終了する必要があります (`CAPI_HANDLE_INITIALIZER`)。NULL 値、つまり空の配列の場合、カレント・ユーザーの予定表が検索されます。
- `in_start`: 日時範囲の始まり。次のいずれかの書式を使用できます。
 - `DATE`。(20020928 など。)
 - `DATE-TIME`。変動時間 (20020929T120000 など) または UTC 時間 (20020929T170000Z など) のいずれかです。変動時間では、ユーザーのタイムゾーンを使用します (ユーザーのタイムゾーン設定項目は `Calendar Server` に格納されています)。
 - `DURATION`。所要時間表記を使用した相対的な日数または日数と時間 (将来の 5 日と 3 時間の場合は `+PT5DT3H`、過去の 1 週間の場合は `-PT1W` など)。
- `in_end`: 日時範囲の終わり。`in_start` に示した書式のいずれかを使用できます。
- `in_requestProperties`: イベントの特定の `iCalendar` プロパティのみをフェッチするには、プロパティ名の配列を渡します。配列は、NULL (ゼロ) ポインタまたは空 (長さがゼロ) の文字列で終了する必要があります。空の配列 (NULL または空の文字列) を使用すると、使用可能なすべてのプロパティが返されます。
- `in_stream`: SDK の書込み先のストリーム。
- `out_requestResult`: NULL 以外の場合、トランザクションの詳細な結果が含まれます。

必要なクリーンアップ:

結果は、`CSDK_DestroyRequestResult` を使用して破棄する必要があります。

返される結果:

`CAPIStatus`

対応する Java メソッド:

`oracle.calendar.sdk.Session.fetchEventsByRange()`

CSDK_FetchEventsByUID

```

CAPISession  in_session,
CAPIFlag     in_flags,
CAPIHandle   in_agenda,
CAPIUIDSet   in_UIDs,
const char * in_RECURRENCEID,
int          in_modifier,
const char ** in_requestProperties,
CAPIStream   in_stream,
CSDKRequestResult * out_requestResult
)

```

この関数は、イベントを UID 別にフェッチします。

1 つのイベントの特定のインスタンスは、`in_RECURRENCEID` および `in_modifier` パラメータを使用してフェッチできます。

`in_requestProperties` パラメータを使用して、特定のプロパティを要求できます。このパラメータは、返されるプロパティ名を含む、ヌル文字（ゼロ）または空の文字列で終了する C 文字列の配列です。最適なパフォーマンスを得るには、要求するプロパティ（特に ATTENDEE プロパティ）を必要なものだけに限定してください。

パラメータ：

- `in_session`: ログイン・セッション・ハンドル。
- `in_flags`: ビット・フラグ：
 - `CSDK_FLAG_NONE`: デフォルトの動作。
 - `CSDK_FLAG_FETCH_AGENDA_ATTENDEE_ONLY`: 参照している予定表の ATTENDEE のみを取得または返します。
 - `CSDK_FLAG_FETCH_LOCALTIMES`: すべての日付と時間をユーザーのタイムゾーンで返します（ユーザーのタイムゾーン設定項目は Calendar Server に格納されています）。
 - `CSDK_FLAG_FETCH_EXPAND_RRULE`: 繰返しルールを展開し、VEVENT のセット（繰返しルールによって生成されるインスタンスごとに 1 つ）を返します。
 - `CSDK_FLAG_STREAM_NOT_MIME`: iCalendar を MIME コンテナに入れません。
- `in_agenda`: 指定された UID でイベントを検索する予定表。NULL 値の場合、カレント・ユーザーの予定表が検索されます。
- `in_UIDs`: フェッチするイベントの UID を含む文字列の配列。配列は、NULL（ゼロ）ポインタまたは空（長さがゼロ）の文字列で終了する必要があります。

- **in_RECURRENCEID:** 1つのイベントのすべてのインスタンスをフェッチするには、NULL または空の文字列を渡します。個々の（または一連の）インスタンスをフェッチするには、DATE または DATE-TIME 書式でイベントの1つのインスタンスを識別する iCalendar の recurrence-id を指定します。
- **in_modifier:** in_RECURRENCEID を使用して recurrence-id を指定した場合、この修飾子によって、指定したインスタンスをフェッチするか一連のインスタンスをフェッチするかが決まります。値は次のとおりです。
 - CAPI_THISINSTANCE
 - CAPI_THISANDPRIOR
 - CAPI_THISANDFUTURE
- **in_requestProperties:** イベントの特定の iCalendar プロパティのみをフェッチするには、プロパティ名の配列を渡します。配列は、NULL（ゼロ）ポインタまたは空（長さがゼロ）の文字列で終了する必要があります。空の配列（NULL または空の文字列）を使用すると、使用可能なすべてのプロパティが返されます。
- **in_stream:** SDK の書込み先のストリーム。
- **out_requestResult:** NULL 以外の場合、トランザクションの詳細な結果が含まれます。

必要なクリーンアップ:

結果は、CSDK_DestroyRequestResult を使用して破棄する必要があります。

返される結果:

CAPIStatus

対応する Java メソッド:

oracle.calendar.sdk.Session.fetchEventsByUID()

CSDK_StoreEvents

```
CAPIStatus CSDK_StoreEvents (  CAPISession    in_session,
                               CAPIFlag      in_flags,
                               CAPIStream    in_stream,
                               CSDKRequestResult *    out_requestResult
                               )
```

この関数は、in_stream から1つの VCALENDAR オブジェクトを読み取り、含まれている各 VEVENT をサーバー上で格納します。

出席者

CAPL_StoreEvent と異なり、イベントの出席者は、iCalendar で ATTENDEE を使用して指定された人だけです (CSDK_FLAG_STORE_INVITE_SELF を使用した場合を除きます。この場合、プロパティの設定に関係なく、ログイン・ユーザーは常に出席依頼されます)。

ATTENDEE プロパティで指定されたアドレスを使用して、カレンダー・ユーザーを識別します。ATTENDEE プロパティ値で指定されたアドレスを持つカレンダー・ユーザーが存在しない場合、出席者は「外部」とみなされ、Calendar Server が対応している場合は iMIP などのインターネットの標準プロトコルを使用して出席依頼されます。

ATTENDEE.PARTSTAT パラメータは、次の場合を除いて無視されます。

- ログイン・ユーザー
- 外部 (カレンダー以外の) 出席者

リソース

Oracle Calendar Server は、各リソースの PARTSTAT 値を格納しますが、リソースには電子メール・アドレスはありません。リソースの予約依頼に ATTENDEE プロパティを使用できるようにするために、次の構文がサポートされています。

```
ATTENDEE;CUTYPE=RESOURCE;CN=projecter:MAILTO:ignored@foobar.com
```

グループ

次の形式の非標準 (ただし正当な) ATTENDEE プロパティを使用して、Oracle Calendar グループに出席依頼できます。

```
ATTENDEE;CUTYPE=GROUP;CN=developers:MAILTO:ignored@foobar.com
```

例で示されているように、プロパティ値 (MAILTO:ignored@foobar.com) は使用されません。この例では、グループ **developers** が展開され、メンバーがカレンダー・ユーザーとして出席を依頼されます。このイベントをフェッチすると、グループのメンバー (CSDK_StoreEvents のコール時) が、個別の ATTENDEE プロパティとして返されます。

サーバーでは、グループ名の一意性は強制されません。ただし、複数のグループが一致した場合、エラーが返されます。

エラー :

out_requestResult パラメータを使用して、詳細なエラー情報が返されます。解析エラー結果が返されないかぎり、格納されている VEVENT ごとに、その VEVENT を格納する CAPISStatus 値を含む少なくとも 1 つの結果が返されます。out_requestResult に NULL (ゼロ) 値を渡すと、要求の結果は返されませんが、エラーとはみなされません。

繰返しルール

この関数では繰返しルール（RRULE）がサポートされ、イベントの DTSTART をローカル時間（TZID=... と VTIMEZONE コンポーネントを使用）または変動時間（RFC 2445 に基づく）で指定する必要があります。Calendar Server の制約により、特定の VEVENT に指定できる RRULE は 1 つだけです。また、イベントの変更時に RRULE を変更することはできません（RDATE と EXDATE を使用する以外にインスタンスを変更する方法はありません）。

UID

Calendar Server では、ユーザー / リソースが特定の UID で複数のイベントを所有することはできません。ただし、Calendar Server では、UID は一意である必要がないため、ユーザー / リソースを特定の UID を持つ複数のイベントに召集することができます。SDK のユーザーは、Calendar Server にイベントを追加する際、グローバルに一意の UID を提供するようにしてください。

UID を指定せずにイベントを格納すると、Calendar Server によって新しい UID が生成され、パフォーマンスがわずかに低下します。生成された UID は、out_requestResult に結果の一部として返されます。

次の処理を行う方法

- **イベントへのインスタンスの追加：** 定期的なイベント（RRULE を使用）の場合、フラグ CSDK_FLAG_STORE_MODIFY を使用して、イベントの UID および 1 つ以上の RDATE プロパティとともに VEVENT を格納します。繰返しイベント（RRULE を使用しない）の場合、イベントの UID とともに VEVENT を格納します。
- **イベントからのインスタンスの削除：** イベントの UID と削除するインスタンスの EXDATE を指定して VEVENT を格納します。EXDATE 値は、インスタンスの RECURRENCE-ID と正確に一致する必要があります。
- **イベントへの出席者の追加：** フラグ CSDK_FLAG_STORE_MODIFY を使用して、新規出席者の ATTENDEE プロパティとともに VEVENT を格納します。VEVENT で RECURRENCE-ID プロパティを指定して、指定されたインスタンスのみに出席を依頼することができます。
- **イベントからの出席者の削除：** イベントをフェッチし、出席依頼を取り消すユーザーの ATTENDEE プロパティを削除します。その後 CSDK_FLAG_STORE_REPLACE を使用して格納します。

パラメータ :

- `in_session`: ログイン・セッション・ハンドル。
- `in_flags`: 動作を変更するビット・フラグ。次のいずれかです。
 - `CSDK_FLAG_STORE_CREATE`: 指定された UID を持つイベントが存在しない場合にイベントを作成します。存在する場合、エラーを返します。
 - `CSDK_FLAG_STORE_REPLACE`: サーバー上のイベントをこのコピーに完全に置き換えます。オブジェクトが存在しない場合、エラーを返します。
 - `CSDK_FLAG_STORE_IMPORT`: イベントが存在する場合、`CSDK_FLAG_STORE_REPLACE`。存在しない場合は、`CSDK_FLAG_STORE_CREATE`。
 - `CSDK_FLAG_STORE_MODIFY`: 既存イベントの指定されたプロパティのみを変更します。
 - `CSDK_FLAG_STORE_REPLY`: イベントの任意の出席者がこのモードを使用して、自分の出席ステータスおよびアラームを更新できます。

追加で、次のパラメータを組み合わせることができます。

- `CSDK_FLAG_STORE_INVITE_SELF`: ATTENDEE が iCal にもない場合も、カレンダー・ユーザーを出席者として追加します。
- `CSDK_FLAG_STREAM_NOT_MIME`: 着信ストリームは MIME ラッパー内に入れられません。
- `CAPI_NOTIFY_EMAIL`: 電子メールによる通知を送信します（デフォルトでは送信されません）。
- `CAPI_NOTIFY_SMS`: SMS による通知を送信します（デフォルトでは送信されません）。
- `in_stream`: SDK がデータを読み取るストリーム。
- `out_requestResult`: NULL 以外の場合、トランザクションの詳細な結果が含まれます。データの読取りに関するエラー・メッセージまたは要求の処理中に発生した他のエラーが含まれる場合があります。

返される結果 :

`CAPIStatus`

必要なクリーンアップ :

要求は、`CSDK_DestroyRequestResult` を使用して破棄する必要があります。

例 3-22 カレント・ユーザーのカレンダーへのイベントの追加

```

{
    static const char * ical = {"BEGIN:VCALENDAR\r\n"
                                "VERSION:2.0\r\n"
                                "BEGIN:VEVENT\r\n"
                                "DTSTART:20021225T100000Z\r\n"
                                "DTEND:20021225T233000Z\r\n"
                                "SUMMARY:work\r\n"
                                "LOCATION:office\r\n"
                                "END:VEVENT\r\n"
                                "END:VCALENDAR\r\n"};

    //
    CAPIStream memoryStream = CSDK_STREAM_INITIALIZER;
    status = CSDK_CreateMemoryStream(mySession,
                                    &memoryStream,
                                    ical,
                                    NULL,
                                    CSDK_FLAG_NONE);

    if (!status)
    {
        status = CSDK_StoreEvents(mySession,
                                CSDK_FLAG_STORE_CREATE | CSDK_FLAG_STORE_INVITE_SELF |
                                CSDK_FLAG_STREAM_NOT_MIME,
                                memoryStream,
                                NULL);

        //
        CSDK_DestroyStream(mySession,
                           &memoryStream);
    }
}

```

例 3-23 複数人に対する会議への出席依頼

```

{
    // invite these people to a meeting:
    const char * attendees[] = {"?/S=Who/G=Cindy Lou/",
                                "?/S=Who/G=Lou Lou/",
                                "?/S=Who/G=Betty Lou/"
                                "grinch");

    //
    const int    numAttendees = (sizeof(attendees) / sizeof(attendees[0]));
    const char ** emailAddresses = (const char **)malloc((numAttendees + 1) * sizeof(const char *));
    CAPHandle *  handles      = (CAPHandle *)malloc(numAttendees * sizeof(CAPHandle *));

    //
    // get handles:
    for (int i = 0; !status && (i < numAttendees); i++)
    {
        status = CSDK_GetHandle(mySession, attendees[i], CSDK_FLAG_NONE, &handles[i]);
    }
}

```

```

//
// terminate the array:
handles[numAttendees] = CSDK_HANDLE_INITIALIZER;
//
if (!status)
{
    // get e-mail addresses for each handle using CSDK_GetHandleInfo()
    ...
}
//
if (!status)
{
    static const char * iCalEvent1 = {"BEGIN:VCALENDAR\r\n"
                                       "VERSION:2.0\r\n"
                                       "BEGIN:VEVENT\r\n"
                                       "DTSTART:20021225T100000Z\r\n"
                                       "DTEND:20021225T103000Z\r\n"
                                       "SUMMARY:SING\r\n"
                                       "LOCATION:Whoville town square\r\n"};

    //
    static const char * iCalEvent2 = {"END:VEVENT\r\n"
                                       "END:VCALENDAR\r\n"};

    //
    string iCalEvent = iCalEvent1;
    for (int attendee = 0; attendee < numAttendees; attendee++)
    {
        if (emailAddresses[attendee])
        {
            iCalEvent += "ATTENDEE:mailto:";
            iCalEvent += emailAddresses[attendee];
            iCalEvent += "\r\n";
        }
    }
    iCalEvent += iCalEvent2;
}
//
if (emailAddresses)
{
    free(emailAddresses);
}
//
CSDK_DestroyMultipleHandles(mySession,
                           handles,
                           numAttendees,
                           CSDK_FLAG_NONE);

if (handles)
{
    free(handles);
}

```

```
//
if (!status)
{
    CAPIStream memoryStream = CSDK_STREAM_INITIALIZER;
    status = CSDK_CreateMemoryStream(mySession,
                                     &memoryStream,
                                     iCalEvent.c_str(),
                                     NULL,
                                     CSDK_FLAG_NONE);

    if (!status)
    {
        status = CSDK_StoreEvents(mySession,
                                   CSDK_FLAG_STORE_CREATE | CSDK_FLAG_STREAM_NOT_MIME,
                                   memoryStream,
                                   NULL);
    }
    //
    CSDK_DestroyStream(mySession,
                       &memoryStream);
}
}
```

対応する Java メソッド:

oracle.calendar.sdk.Session.storeEvents

CSDK_DeleteTasks

```
CAPIStatus CSDK_DeleteTasks (  CAPISession      in_session,
                               CAPIFlag         in_flags,
                               CAPIUIDSet        in_UIDs,
                               CSDKRequestResult * out_requestResult
                               )
```

この関数は、カレント・ユーザーの予定表でタスクを削除します。

ユーザーの識別情報を使用していない Calendar SYSOP は、この関数を使用できません。

パラメータ:

- in_session: セッション
- in_flags: CSDK_FLAG_NONE
- in_UIDs: ヌル文字で終了するタスク UID の配列
- out_requestResult: 返される要求結果オブジェクト

返される結果：

- CAPIStatus

戻り値：

- CAPI_STAT_OK
- CAPI_STAT_API_SESSION_NULL: in_session が NULL です。
- CAPI_STAT_API_NULL: in_UIDSet が NULL です。

対応する Java メソッド：

oracle.calendar.sdk.Session.deleteTasks()

CSDK_FetchTasksByAlarmRange

```
CAPIStatus CSDK_FetchTasksByAlarmRange ( CAPISession    in_session,
                                          CAPIFlag      in_flags,
                                          CAPIHandle *   in_handles,
                                          const char *   in_start,
                                          const char *   in_end,
                                          const char **  in_requestProperties,
                                          CAPIStream     in_stream,
                                          CSDKRequestResult * out_requestResult
                                          )
```

この関数は、指定された時間範囲内にトリガーされるリマインダを持つタスクをフェッチします。

時間範囲の終わりは除外されます。

パラメータ：

- in_session: ログイン・セッション・ハンドル。
- in_flags: ビット・フラグ：
 - CSDK_FLAG_NONE
 - CSDK_FLAG_STREAM_NOT_MIME: iCalendar を MIME コンテナに入れません。
- in_handles: タスクを検索する予定表。このパラメータは CAPIHandles の配列で、ゼロ値で終了する必要があります (CSDK_HANDLE_INITIALIZER)。NULL 値、つまり空の配列の場合、カレント・ユーザーの予定表が検索されます。
- in_start: 日時範囲の始まり。次のいずれかの書式を使用できます。
 - DATE。(20020928 など。)
 - DATE-TIME。変動時間 (20020929T120000 など) または UTC 時間 (20020929T170000Z など) のいずれかです。

- DURATION。所要時間表記を使用した相対的な日数または日数と時間（将来の 5 日と 3 時間の場合は +PT5DT3H、過去の 1 週間の場合は -PT1W など）。
- in_end: 日時範囲の終わり。in_start に示した書式のいずれかを使用できます。
- in_requestProperties: タスクの特定の iCalendar プロパティのみをフェッチするには、プロパティ名の配列を渡します。配列は、NULL（ゼロ）ポインタまたは空（長さがゼロ）の文字列で終了する必要があります。空の配列（NULL または空の文字列）を使用すると、使用可能なすべてのプロパティが返されます。
- in_stream: SDK の書き込み先のストリーム。
- out_requestResult: NULL 以外の場合、トランザクションの詳細な結果が含まれます。

必要なクリーンアップ:

要求は、CSDK_DestroyRequestResult を使用して破棄する必要があります。

返される結果:

CAPIStatus

対応する Java メソッド:

oracle.calendar.sdk.Session.fetchTasksByAlarmRange()

CSDK_FetchTasksByRange

```
CAPIStatus CSDK_FetchTasksByRange ( CAPISession      in_session,
                                     CAPIFlag         in_flags,
                                     CAPIHandle *      in_handles,
                                     const char *      in_start,
                                     const char *      in_end,
                                     const char **     in_requestProperties,
                                     CAPIStream        in_stream,
                                     CSDKRequestResult * out_requestResult
                                     )
```

この関数は、指定された時間範囲内にアクティブになるタスクをフェッチします。

時間範囲の終わりは除外されます。

パラメータ:

- `in_session`: ログイン・セッション・ハンドル。
- `in_flags`: ビット・フラグ:
 - `CSDK_FLAG_NONE`
 - `CSDK_FLAG_FETCH_NO_MIME`: `iCalendar` を MIME コンテナに入れません。
- `in_handles`: タスクを検索する予定表。このパラメータは `CAPIHandles` の配列で、ゼロ値で終了する必要があります (`CSDK_HANDLE_INITIALIZER`)。NULL 値、つまり空の配列の場合、カレント・ユーザーの予定表が検索されます。
- `in_start`: 日時範囲の始まり。次のいずれかの書式を使用できます。
 - `DATE`。(20020928 など。)
 - `DATE-TIME`。変動時間 (20020929T120000 など) または UTC 時間 (20020929T170000Z など) のいずれかです。
 - `DURATION`。所要時間表記を使用した相対的な日数または日数と時間 (将来の 5 日と 3 時間の場合は `+PT5DT3H`、過去の 1 週間の場合は `-PT1W` など)。
- `in_end`: 日時範囲の終わり。 `in_start` に示した書式のいずれかを使用できます。
- `in_requestProperties`: タスクの特定の `iCalendar` プロパティのみをフェッチするには、プロパティ名の配列を渡します。配列は、NULL (ゼロ) ポインタまたは空 (長さがゼロ) の文字列で終了する必要があります。空の配列 (NULL または空の文字列) を使用すると、使用可能なすべてのプロパティが返されます。
- `in_stream`: SDK の書込み先のストリーム。
- `out_requestResult`: NULL 以外の場合、トランザクションの詳細な結果が含まれます。

必要なクリーンアップ:

要求は、`CSDK_DestroyRequestResult` を使用して破棄する必要があります。

返される結果:

`CAPIStatus`

対応する Java メソッド:

`oracle.calendar.sdk.Session.fetchTasksByRange()`

CSDK_FetchTasksByUID

```

CAPIStatus CSDK_FetchTasksByUID (  CAPISession    in_session,
                                   CAPIHandle     in_handle,
                                   CAPIFlag       in_flags,
                                   CAPIUIDSet     in_UIDs,
                                   const char **   in_requestProperties,
                                   CAPIStream     in_stream,
                                   CSDKRequestResult * out_requestResult
                                   )

```

この関数は、指定された予定表で指定された UID を持つタスクを取得します。

パラメータ：

- in_session: ログイン・セッション・ハンドル。
- in_flags: ビット・フラグ：
 - CSDK_FLAG_NONE
 - CSDK_FLAG_STREAM_NOT_MIME: iCalendar を MIME コンテナに入れません。
- in_handle: 指定された UID を持つタスクを検索する予定表。NULL 値の場合、カレンダー・ユーザーの予定表が検索されます。
- in_UIDs: フェッチするタスクの UID を含む文字列の配列。配列は、NULL (ゼロ) ポインタまたは空 (長さがゼロ) の文字列で終了する必要があります。
- in_requestProperties: タスクの特定の iCalendar プロパティのみをフェッチするには、プロパティ名の配列を渡します。配列は、NULL (ゼロ) ポインタまたは空 (長さがゼロ) の文字列で終了する必要があります。空の配列 (NULL または空の文字列) を使用すると、使用可能なすべてのプロパティが返されます。
- in_stream: SDK の書き込み先のストリーム。
- out_requestResult: NULL 以外の場合、トランザクションの詳細な結果が含まれます。

必要なクリーンアップ：

結果は、CSDK_DestroyRequestResult を使用して破棄する必要があります。

返される結果：

CAPIStatus

対応する Java メソッド：

oracle.calendar.sdk.Session.fetchTasksByUID()

CSDK_StoreTasks

```

CAPIStatus CSDK_StoreTasks ( CAPISession    in_session,
                             CAPIFlag      in_flags,
                             CAPIStream    in_stream,
                             CSDKRequestResult * out_requestResult
                             )

```

この関数は、渡された SDK 格納フラグに基づいて、カレント・ユーザーの予定表でタスクを作成または変更します。

1 つのフラグのみを使用する必要があります。複数のフラグを渡すと、エラー CAPI_STAT_API_BADPARAM が返されます。次の 5 つのフラグを使用できます。

- CSDK_FLAG_STORE_IMPORT: タスクが存在しない場合はタスクを格納し、存在する場合はタスクを変更します。
- CSDK_FLAG_STORE_CREATE: タスクを格納します。タスクが存在する場合、エラーが返されます。
- CSDK_FLAG_STORE_REPLACE: 既存のタスクを変更します。タスクが存在しない場合、エラーが返されます。
- CSDK_FLAG_STORE_MODIFY: 指定されたプロパティを変更します。
- CSDK_FLAG_STORE_REMOVE: 指定されたプロパティを削除します。

他のフラグを、上の格納フラグの 1 つとともに指定できます。

- CSDK_FLAG_STREAM_NOT_MIME: 着信ストリームは MIME ラッパー内に入れられません。

ユーザーの識別情報を使用していない Calendar SYSOP は、この関数を使用できません。

パラメータ :

- in_session: セッション
- in_flags: 格納フラグ
- in_stream: 入力ストリーム
- out_requestResult: 返される要求結果オブジェクト (NULL の場合もあり)

返される結果 :

CAPIStatus

戻り値 :

- CAPI_STAT_OK
- CAPI_STAT_API_SESSION_NULL: in_session が NULL です。

- CAPI_STAT_API_STREAM_NULL: in_inputStream が NULL です。
- CAPI_STAT_API_FLAGS: in_flags が無効です。

必要なクリーンアップ:

要求は、CSDK_DestroyRequestResult を使用して破棄する必要があります。

対応する Java メソッド:

oracle.calendar.sdk.Session.storeTasks

CSDK_AddConditionToQuery

```
CAPIStatus CSDK_AddConditionToQuery ( CSDKQuery    in_query,
                                       CSDKCondition * in_condition,
                                       CSDKOperator   in_operator
                                       )
```

問合せオブジェクトに条件を追加します。各問合せには複数の条件を含めることができ、それぞれが前の条件に対して AND 処理または OR 処理されます。条件をグループ化する方法はありません。また、OR 演算子 (CSDK_LOP_OR) の方が AND 演算子 (CSDK_LOP_AND) より優先度は上です。したがって、C1 OR C2 AND C3 は、(C1 OR C2) AND C3 として評価されます。

パラメータ:

- in_query: CSDK_CreateQuery によって作成される問合せオブジェクト。
- in_condition: 問合せに追加する条件。
- in_operator: 既存の条件とこの条件の間で使用する演算子を指定します (OR、AND など)。

返される結果:

CAPIStatus

対応する Java メソッド:

oracle.calendar.sdk.Query.addCondition()

CSDK_CreateQuery

```
CAPIStatus CSDK_CreateQuery ( CSDKCondition *   in_condition,
                              CSDKQuery *      out_query
                              )
```

CSDK_FetchContactsByQuery で使用される問合せオブジェクトを作成します。最初の条件を指定し（「A で始まる名前」など）、CSDK_AddConditionToQuery を使用してさらに条件を追加できます。

パラメータ：

- in_condition: 問合せの最初の条件。
- out_query: 出力時、新しい問合せオブジェクトが含まれます。

必要なクリーンアップ：

問合せオブジェクトは、CSDK_DestroyQuery をコールして破棄する必要があります。

返される結果：

CAPIStatus

例 3-24 問合せの作成

```
//
CSDKCondition cond;
//
cond.prop  = "N";
cond.op    = CSDK_OP_STARTSWITH;
cond.value = "A";
//
CSDKQuery myQuery = CSDK_QUERY_INITIALIZER;
stat = CSDK_CreateQuery(&cond,
                        &myQuery);

//
stat = CSDK_FetchContactsByQuery(mySession,
                                CSDK_FLAG_STREAM_NOT_MIME,
                                myQuery,
                                NULL, // get all properties
                                myStream,
                                &requestResult);

//
CSDK_DestroyQuery(&myQuery);
```

対応する Java メソッド：

oracle.calendar.sdk.Query コンストラクタ

CSDK_DestroyQuery

CAPIStatus CSDK_DestroyQuery (CSDKQuery * io_query)

CSDK_CreateQuery によって作成された問合せオブジェクトを破棄します。

パラメータ :

- io_query: 破棄する問合せオブジェクトを示すポインタ。終了時 CSDK_QUERY_INITIALIZER を示します。

返される結果 :

CAPIStatus

対応する Java メソッド :

なし。oracle.calendar.sdk.Query ファイナライザがオブジェクトを破棄します。

CSDK_DestroyResult

CAPIStatus CSDK_DestroyResult (CSDKRequestResult * io_requestResult)

この関数は、in_requestResult のすべての結果を廃棄します。

パラメータ :

- io_requestResult: 破棄する RequestResult

返される結果 :

CAPIStatus

戻り値 :

- CAPI_STAT_API_NULL: io_requestResult に NULL 値が含まれています。

対応する Java メソッド :

oracle.calendar.sdk.Result ファイナライザ

CSDK_GetFirstFailure

```
CAPIStatus CSDK_GetFirstFailure ( CSDKRequestResult  in_requestResult,
                                  CAPIHandle *        out_user,
                                  const char **        out_uid,
                                  CAPIStatus *         out_status
                                )
```

この関数は、in_requestResult を返した API 関数から取得した最初の失敗を返します。失敗は、CAPI_STAT_OK 以外のステータスを持つ結果です。

注意： 要求結果には、現在の失敗への参照が含まれます。したがって、特定の要求結果から失敗を抽出するのは一度に 1 つのスレッドだけです。

パラメータ：

- in_requestResult: 情報を抽出する RequestResult
- out_user: この結果の原因となったユーザー
- out_uid: この結果によって返される UID
- out_status: この結果によって返されるステータス

返される結果：

CAPIStatus

戻り値：

- CAPI_STAT_API_NULL: 必須パラメータの 1 つが NULL 値でした。
- CAPI_STAT_DATA_RRESULT_EOR: RequestResult に失敗はありませんでした。

例 3-25 API から返された最初の失敗を取得

```
const char * vcardUID = 0;
CAPIStatus  vcardStatus = CAPI_STAT_OK;
CSDKRequestResult * myRequestResult = 0;
//
stat = CSDK_StoreContacts(mySession,
                          myStream,
                          CSDK_FLAG_STORE_IMPORT,
                          &myRequestResult);
//
CAPIStatus failStat = CSDK_GetFirstFailure(myRequestResult,
                                           NULL,
                                           &vcardUID,
                                           &vcardStatus);
```

```
//
if (failStat == CAPI_STAT_DATA_RRESULT_EOR)
{
    cout << "Store of VCARD with UID " << vcardUID << " succeeded." << endl;
}
else
{
    const char * statusName = 0;
    CSDK_GetStatusString(vcardStatus, &statusName);
    cout << "Store of VCARD with UID " << vcardUID << " failed with CAPIStatus " <<
    statusName << "." << endl;
}
//
CSDK_DestroyResult(&myRequestResult);
```

対応する Java メソッド:

oracle.calendar.sdk.Result.getFirstFailure()

CSDK_GetFirstParseError

```
CAPIStatus CSDK_GetFirstParseError ( CSDKRequestResult    in_requestResult,
                                     CAPIStatus *         out_status,
                                     const char **         out_errorBuffer,
                                     const char **         out_errorLocation,
                                     const char **         out_message
                                     )
```

この関数は、要求結果から取得した最初の解析エラーを返します。

解析エラーは、CSDK_Store* 関数が着信 iCalendar または vCard を解釈しようとしたときに生成されることがあります。

注意: 要求結果には、現在の解析エラーへの参照が含まれます。したがって、特定の要求結果から解析エラーを抽出するのは一度に 1 つのスレッドだけです。

データ・ストリームのコピーを示すポインタは out_errorBuffer から返され、バッファ内の解析エラーの場所を示すポインタは、out_errorLocation から返されます。どちらのポインタも、要求結果が破棄されるまで有効です。

パラメータ :

- `in_requestResult`: 情報を抽出する `RequestResult`。
- `out_status`: 結果のステータス。
- `out_errorBuffer`: エラーが含まれるバッファの始まり。
- `out_errorLocation`: `*out_errorBuffer` 内のエラーが発生した場所。
- `out_message`: 追加情報が含まれている場合があります (メッセージがない場合、`NULL` が返されます)。

返される結果 :

`CAPIStatus`

戻り値 :

- `CAPI_STAT_API_NULL`: 必須パラメータの 1 つが `NULL` 値でした。
- `CAPI_STAT_DATA_RRESULT_EOR`: `RequestResult` に解析エラーはありませんでした。

例 3-26 要求結果からの最初の解析エラーの取得

```
CAPIStatus      stat = CAPI_STAT_OK;
CSDKRequestResult * myRequestResult = 0;
//
stat = CSDK_StoreContacts(mySession,
                          myStream,
                          CSDK_FLAG_STORE_IMPORT,
                          &myRequestResult);

//
const char * buffer = 0;
const char * errorLocation = 0;
const char * message = 0;
//
CAPIStatus parseStat = CSDK_GetFirstParseError(myRequestResult,
                                                NULL,
                                                &buffer,
                                                &errorLocation,
                                                &message);
```

```
//
if (parseStat != CAPI_STAT_DATA_RRESULT_EOR)
{
    cout << "Error (" << message << ") parsing vCard. Buffer:'" << vcardUID << "'
    Error starting at:'" << errorLocation << "'" << endl;
}
//
CSDK_DestroyResult(&myRequestResult);
```

対応する Java メソッド:

oracle.calendar.sdk.Result.getFirstParseError()

CSDK_GetFirstResult

```
CAPIStatus CSDK_GetFirstResult ( CSDKRequestResult    in_requestResult,
                                CAPIHandle *          out_user,
                                const char **          out_uid,
                                CAPIStatus *           out_status
                                )
```

この関数は、in_requestResult を返した API 関数から取得した最初の結果を返します。

結果は失敗または成功です。失敗は、CAPI_STAT_OK 以外のステータスを持つ結果です。

注意: 要求結果には、現在の結果への参照が含まれます。したがって、特定の要求結果から結果を抽出するのは一度に 1 つのスレッドだけです。

パラメータ:

- in_requestResult: 情報を抽出する RequestResult
- out_user: この結果の原因となったユーザー
- out_uid: この結果によって返される UID
- out_status: この結果によって返されるステータス

返される結果:

CAPIStatus

戻り値:

- CAPI_STAT_API_NULL: 必須パラメータの 1 つが NULL 値でした。
- CAPI_STAT_DATA_RRESULT_EOR: RequestResult に結果はありませんでした。

例 3-27 API からの最初の結果の取得

```

const char * vcardUID = 0;
CAPIStatus  vcardStatus = CAPI_STAT_OK;
CSDKRequestResult * myRequestResult = 0;
//
stat = CSDK_StoreContacts(mySession,
                          myStream,
                          CSDK_FLAG_STORE_IMPORT,
                          &myRequestResult);
//
CSDK_GetFirstResult(myRequestResult,
                    NULL,
                    &vcardUID,
                    &vcardStatus);
//
if (vcardStatus == CAPI_STAT_OK)
{
    cout << "Store of VCARD with UID " << vcardUID << " succeeded." << endl;
}
else
{
    const char * statusName = 0;
    CSDK_GetStatusString(vcardStatus, &statusName);
    cout << "Store of VCARD with UID " << vcardUID << " failed with CAPIStatus " <<
    statusName << "." << endl;
}
//
CSDK_DestroyResult(&myRequestResult);

```

対応する Java メソッド:

oracle.calendar.sdk.Result.getFirstResult()

CSDK_GetNextFailure

```

CAPIStatus CSDK_GetNextFailure ( CSDKRequestResult  in_requestResult,
                                CAPIHandle *       out_user,
                                const char **       out_uid,
                                CAPIStatus *        out_status
                                )

```

この関数は、CSDKRequestResult に含まれている次の失敗を返します。

このコールの前に、CSDK_GetFirstFailure をコールする必要があります。

注意: 要求結果には、現在の失敗への参照が含まれます。したがって、特定の要求結果から失敗を抽出するのは一度に 1 つのスレッドだけです。

パラメータ :

- in_requestResult: 情報を抽出する RequestResult
- out_user: この結果の原因となったユーザー
- out_uid: この結果によって返される UID
- out_status: この結果によって返されるステータス

返される結果 :

CAPIStatus:

戻り値 :

- CAPI_STAT_API_NULL: 必須パラメータの 1 つが NULL 値でした。
- CAPI_STAT_DATA_RRESULT_EOR: RequestResult に次の失敗はありませんでした。

例 3-28 CSDKRequestResult に含まれる次の失敗の取得

```
const char * vcardUID = 0;
CAPIStatus stat = CAPI_STAT_OK;
CSDKRequestResult * myRequestResult = 0;
//
stat = CSDK_StoreContacts(mySession,
                        myStream,
                        CSDK_FLAG_STORE_IMPORT,
                        &myRequestResult);
//
stat = CSDK_GetFirstFailure(myRequestResult,
                        NULL,
                        &vcardUID,
                        &vcardStatus);
//
while (stat != CAPI_STAT_DATA_RRESULT_EOR)
{
    const char * statusName = 0;
    CSDK_GetStatusString(vcardStatus, &statusName);
    cout << "Store of VCARD with UID " << vcardUID << " failed with status " << statusName << endl;
    //
    stat = CSDK_GetNextFailure(myRequestResult,
                        NULL,
                        &vcardUID,
                        &vcardStatus);
}
//
CSDK_DestroyResult(&myRequestResult);
```

対応する Java メソッド:

oracle.calendar.sdk.Result.getNextFailure()

CSDK_GetNextParseError

```
CAPIStatus CSDK_GetNextParseError ( CSDKRequestResult  in_requestResult,
                                     CAPIStatus *      out_status,
                                     const char **      out_errorBuffer,
                                     const char **      out_errorLocation,
                                     const char **      out_message
                                     )
```

この関数は、要求結果から取得した次の解析エラーを返します。

このコールの前に、CSDK_GetFirstParseError をコールする必要があります。

注意: 要求結果には、現在の解析エラーへの参照が含まれます。したがって、特定の要求結果から解析エラーを抽出するのは一度に 1 つのスレッドだけです。

データ・ストリームのコピーを示すポインタは out_errorBuffer から返され、バッファ内の解析エラーの場所を示すポインタは、out_errorLocation から返されます。どちらのポインタも、要求結果が破棄されるまで有効です。

パラメータ:

- in_requestResult: 情報を抽出する RequestResult。
- out_status: 結果のステータス。
- out_errorBuffer: エラーが含まれるバッファの始まり。
- out_errorLocation: *out_errorBuffer 内のエラーが発生した場所。
- out_message: 追加情報が含まれていることがあります。

返される結果:

CAPIStatus

戻り値:

- CAPI_STAT_API_NULL: 必須パラメータの 1 つが NULL 値でした。
- CAPI_STAT_DATA_RRESULT_EOR: RequestResult に解析エラーはありませんでした。

例 3-29 要求結果からの次の解析エラーの取得

```
CAPIStatus          stat = CAPI_STAT_OK;
CSDKRequestResult * myRequestResult = 0;
//
stat = CSDK_StoreContacts(mySession,
                          myStream,
                          CSDK_FLAG_STORE_IMPORT,
                          &myRequestResult);

//
const char * buffer = 0;
const char * errorLocation = 0;
const char * message = 0;
//
CAPIStatus parseStat = CSDK_GetFirstParseError(myRequestResult,
                                                NULL,
                                                &buffer,
                                                &errorLocation,
                                                &message);

//
while (parseStat != CAPI_STAT_DATA_RRESULT_EOR)
{
    cout << "Error (" << message << ") parsing vCard. Buffer:" << vcardUID << " "
Error starting at:" << errorLocation << " " << endl;
    parseStat = CSDK_GetNextParseError(myRequestResult,
                                       NULL,
                                       &buffer,
                                       &errorLocation,
                                       &message);
}
//
CSDK_DestroyResult(&myRequestResult);
```

対応する Java メソッド:

oracle.calendar.sdk.Result.getNextParseError()

CSDK_GetNextResult

```

CAPIStatus CSDK_GetNextResult ( CSDKRequestResult    in_requestResult,
                                CAPIHandle *          out_user,
                                const char **          out_uid,
                                CAPIStatus *          out_status
                                )

```

この関数は、CSDKRequestResult に含まれている次の結果を返します。

このコールの前に、CSDK_GetFirstResult をコールする必要があります。

注意： 要求結果には、現在の結果への参照が含まれます。したがって、特定の要求結果から結果を抽出するのは一度に 1 つのスレッドだけです。

パラメータ：

- in_requestResult: 情報を抽出する RequestResult
- out_user: この結果の原因となったユーザー
- out_uid: この結果によって返される UID
- out_status: この結果によって返されるステータス

返される結果：

CAPIStatus

戻り値：

- CAPI_STAT_API_NULL: 必須パラメータの 1 つが NULL 値でした。
- CAPI_STAT_DATA_RRESULT_EOR: RequestResult に次の結果はありませんでした。

例 3-30 CSDKRequestResult に含まれる次の結果の取得

```

const char * vcardUID = 0;
CAPIStatus  stat = CAPI_STAT_OK;
CSDKRequestResult * myRequestResult = 0;
//
stat = CSDK_StoreContacts(mySession,
                          myStream,
                          CSDK_FLAG_STORE_IMPORT,
                          &myRequestResult);
//
stat = CSDK_GetFirstResult(myRequestResult,
                           NULL,
                           &vcardUID,
                           &vcardStatus);

```

```
//
while (stat != CAPI_STAT_DATA_RRESULT_EOR)
{
    const char * statusName = 0;
    CSDK_GetStatusString(vcardStatus, &statusName);
    cout << "Store of VCARD with UID " << vcardUID << " returned status " << statusName
    << endl;
    //
    stat = CSDK_GetNextResult(myRequestResult,
                             NULL,
                             &vcardUID,
                             &vcardStatus);
}
//
CSDK_DestroyResult(&myRequestResult);
```

対応する Java メソッド:

oracle.calendar.sdk.Result.getNextResult()

CSDK_GetStatusLevels

```
void CSDK_GetStatusLevels (  CAPIStatus    in_status,
                             unsigned long * out_level1,
                             unsigned long * out_level2,
                             unsigned long * out_level3,
                             unsigned long * out_level4,
                             unsigned long * out_level5
                             )
```

この関数は、CAPIStatus をサブパートに分解します。

ステータス・コードの各パートでは、実際のエラーをより詳しく指定します。

パラメータ:

- in_status: SDK ステータス。
- out_level1: レベル 1 の int 型の結果が含まれます。
- out_level2: レベル 2 の int 型の結果が含まれます。
- out_level3: レベル 3 の int 型の結果が含まれます。
- out_level4: レベル 4 の int 型の結果が含まれます。
- out_level5: レベル 5 の int 型の結果が含まれます。

変更:
CAPI 2.5: out_level[12345] の型が int * から unsigned long * に変更されました。

対応する Java メソッド:
oracle.calendar.sdk.Api.getStatusLevels()

CSDK_GetStatusString

```
void CSDK_GetStatusString ( CAPIStatus    in_status,
                           const char **  out_string
                           )
```

この関数は、CAPIStatus の読取り専用文字列表現を返します。
一般に、数値表現より有用です。

- パラメータ:**
- in_status: SDK ステータス。
 - out_string: 結果文字列を示す const ポインタが含まれます。

必要なクリーンアップ:
なし。返される文字列は const 文字列なので、解放できません。

対応する Java メソッド:
oracle.calendar.sdk.Api.getStatusString()

改訂された関数

次の関数は新しい関数に改訂され、名前も変更されています。新しい関数を使用することをお薦めしますが、CSDK_USE_OLD_NAMES を定義して古い関数を使用することもできます。

改訂された関数	新規バージョン
CAPI_CreateCallbackStream	CSDK_CreateCallbackStream
CAPI_CreateFileStreamFromFileNames	CSDK_CreateFileStreamFromFileNames
CAPI_CreateMemoryStream	CSDK_CreateMemoryStream
CAPI_DestroyHandles	CSDK_DestroyHandles
CAPI_DestroyStreams	CSDK_DestroyStreams

改訂された関数	新規バージョン
CAPI_GetCapabilities	CSDK_GetCapabilities
CAPI_GetHandle	CSDK_GetHandle
CAPI_GetStatusLevels	CSDK_GetStatusLevels
CAPI_GetStatusString	CSDK_GetStatusString
CAPI_HandleInfo	CSDK_GetHandleInfo
CAPI_SetConfigFile	CSDK_SetConfigFile
CAPI_SetIdentity	CSDK_SetIdentity

レガシー関数

次の関数は、SDK の旧バージョンに含まれていた関数で、このリリースでもサポートされています。

- CAPI_CreateFileStream
- CAPI_AuthenticateAsSysop
- CAPI_Connect
- CAPI_DeleteEvent
- CAPI_FetchEventByID
- CAPI_FetchEventsByAlarmRange
- CAPI_FetchEventsByRange
- CAPI_GetLastStoredUIDs
- CAPI_Logoff
- CAPI_Logon
- CAPI_StoreEvent

次に、関数 CAPI_CreateFileStream および CAPI_AuthenticateAsSysop について説明します。これらの関数および他のレガシー関数の詳細は、SDK ソース・ファイルまたは旧リリースのマニュアルを参照してください。

CAPI_CreateFileStream

```
CSDKStatus CAPI_CreateFileStream (  CAPISession  in_session,
                                     CAPIStream *  out_stream,
                                     FILE *        in_readFile,
                                     FILE *        in_writeFile,
                                     CAPIFlag      in_flags
                                     )
```

SDK が、開かれたファイルとの間で読み書きを行うためのファイル・ストリームを作成します。

互換性の理由から、アプリケーションと CAPI の間で FILE * 変数を渡す必要がない CSDK_CreateFileStreamFromFilenames を使用する方が安全です。

ファイルが、適切なモードで fopen() を使用して開かれている必要があります。

パラメータ：

- in_session: ログイン・セッション・ハンドル。
- out_stream: 出力で新しいストリームを示します。
- in_readFile: 読取り先の FILE *。
- in_writeFile: 書込み先の FILE *。
- in_flags: ビット・フラグ（現時点では CSDK_FLAG_NONE に設定する必要があります）。

返される結果：

CAPIStatus

必要なクリーンアップ：

この関数によって返されるストリームは、CSDK_DestroyStreams() をコールして破棄する必要があります。

例 3-31 ファイル events.ics からのイベントの格納

```
FILE * myFileFullOfMIMEEncapsulatediCal = fopen("events.ics", "rb");
if (myFileFullOfMIMEEncapsulatediCal != NULL)
{
    CAPIStream myInputStream = NULL;
    CAPIStatus status = CAPI_CreateFileStream(mySession,
                                              &myInputStream,
                                              myFileFullOfMIMEEncapsulatediCal,
                                              NULL, // no output file
                                              CSDK_FLAG_NONE);
```

```
if (status == CAPI_STAT_OK)
{
    status = CAPI_StoreEvent(mySession,
                             myHandles,
                             numHandles,
                             handleStatus,
                             CAPI_STORE_REPLACE,
                             myInputStream);
}
fclose(myFileFullofMIMEEncapsulatediCal);
CSDK_DestroyStreams(mySession,
                    &myInputStream,
                    1,
                    CSDK_FLAG_NONE);
}
```

例 3-32 イベントをフェッチし、ファイル myAgenda.ics に書き込む

```
FILE * outputFile = fopen("myAgenda.ics", "wb");
if (outputFile != NULL)
{
    CAPIStream myOutputStream = NULL;
    CAPIStatus status = CAPI_CreateFileStream(mySession,
                                              &myOutputStream,
                                              NULL, // no input file
                                              outputFile,
                                              CSDK_FLAG_NONE);

    if (status == CAPI_STAT_OK)
    {
        status = CAPI_FetchEventsByRange(mySession,
                                         myHandles,
                                         numHandles,
                                         handleStatus,
                                         CSDK_FLAG_NONE,
                                         "20020722T000000",
                                         "20020722T235900",
                                         NULL,
                                         0,
                                         myOutputStream);
    }
    fclose(outputFile);
    CSDK_DestroyStreams(mySession,
                        &myOutputStream,
                        1,
                        CSDK_FLAG_NONE);
}
```

対応する Java メソッド:

なし。Java API では、データの送受信には、String および StringBuffer オブジェクトのみを使用します。

CAPI_AuthenticateAsSysop

```
CAPIStatus CAPI_AuthenticateAsSysop (  const char *   in_password,
                                       const char *   in_host,
                                       const char *   in_node,
                                       CAPIFlag      in_flags,
                                       CAPISession *   io_session
                                       )
```

Calendar SYSOP としてログオンします。

ログオン後、Calendar SYSOP は、CSDK_SetIdentity() をコールして任意のユーザーの識別情報を使用できます。

Calendar SYSOP としてログオンしている間は、マスター・ノードおよびカレンダ・ドメイン機能を使用できないため、必ずノードを指定する必要があります。

ホスト・パラメータで ACE メカニズムが指定されている場合でも、その設定は無視されません。Calendar Server からの管理デフォルト ACE 設定が使用されます。

Calendar SYSOP 認証は、リリース 5.3 以上のサーバーでのみ使用できます。提供されているホストがこの機能をサポートしていない場合、エラーが返されます。CAPI を介した Calendar SYSOP ログオンを拒否するように Calendar Server を構成できます。この方法でログオンした場合、セキュリティ・エラーが返されます。

Calendar SYSOP が使用できる操作は、次の操作に限定されます。

- CAPI_Logoff() のコールによるログオフ
- CAPI_SetIdentity() のコールによる別のユーザーへの識別情報の切替え

識別情報を別のユーザーに設定すると、すべての操作が、そのユーザーがログインしているかのように実行されます。

パラメータ:

- in_password: Calendar SYSOP のパスワード。NULL でもかまいません。
- in_host: Calendar Server のホスト名 (オプションのポート番号)。
- in_node: Calendar SYSOP として接続するためのノード ID。
- in_flags: 動作を変更するビット・フラグ。現時点では CAPI_FLAG_NONE に設定する必要があります。
- io_session: CAPI_Connect によって開かれるセッション。

返される結果：

CAPIStatus

例 3-33 Calendar SYSOP としてのログオン

Example:

```
// Connect to myNode on the server running on the default port of
calserver.acme.com, to authenticate as Calendar SYSOP:
{
    CAPIStatus myStatus = CAPI_AuthenticateAsSysop("theSysopPassword",
                                                    "calserver.acme.com",
                                                    "1",
                                                    CAPI_FLAG_NONE,
                                                    &mySession);
}
```

必要なクリーンアップ：

このルーチンによって作成されるセッションは、CAPI_Logoff をコールして破棄する必要があります。

変更：

SDK 9.0.4: in_node を数値として指定する必要があります ("100" など)。ノードの別名はサポートされていません。また NULL を渡すこともできません。

Oracle Calendar SDK の構成設定

この章では、Oracle Calendar SDK の構成設定について詳しく説明します。これらの設定はテキスト・ファイルで指定でき、ファイルの名前を関数 `CSDK_SetConfigFile` に渡す必要があります。このファイルの構造は次のとおりです。

[<セクション>]

< キーワード > = < 値 >

...

例 4-1 構成設定テキスト・ファイル

```
[CAPI]
client_name = mycool.fcgi
client_version = 1.0
[LOG]
log_activity = true
log_modulesinclude = {CAPI}
```

サポートされている構成設定は次のとおりです。

- `blocking`
- `client_name`
- `client_version`
- `cncachesize`
- `direntycachesize`
- `emailcachesize`
- `itemcachesize`
- `log_activity`
- `log_modulesinclude`

- [max_caldomain](#)
- [max_masternode](#)
- [max_sysop](#)
- [max_user](#)
- [securitycachesize](#)
- [tzcachesize](#)

blocking

使用可能な接続がない場合に、接続要求をブロックするか、エラーを返すかを設定します。

セクション

CONNPOOL

値

true/false

デフォルト値

true

client_name

サーバー統計に表示するアプリケーション名を設定します。

セクション

CAPI

値

任意の文字列

デフォルト値

''''

client_version

サーバー統計に表示するアプリケーション・バージョンを設定します。

セクション

CAPI

値

任意の文字列

デフォルト値

""

cncachesize

共通名キャッシュに保持するエントリの最大数を設定します。

セクション

CACHE

値

[0..U32MAX]

デフォルト値

512

direntrycachesize

ディレクトリ・エントリ・キャッシュに保持するエントリの最大数を設定します。

セクション

CACHE

値

[0..U32MAX]

デフォルト値

512

emailcachesize

電子メール・アドレス・キャッシュに保持するエントリの最大数を設定します。

セクション

CACHE

値

[0..U32MAX]

デフォルト値

512

itemcachesize

項目キャッシュに保持するエントリの最大数を設定します。

セクション

CACHE

値

[0..U32MAX]

デフォルト値

512

log_activity

アクティビティ（高レベル）ロギングを有効にします。

セクション

LOG

値

true/false

デフォルト値

false

関連項目：

「log_modulesinclude」

log_modulesinclude

ロギングを有効にするモジュールを制御します。

"{CAPI}" に設定する必要があります。この値に設定しない場合、(log_activity = true などを使用して) ロギングを有効にしてもロギングは実行されません。

セクション

LOG

値

"" または "{ CAPI }"

デフォルト値

""

max_caldomain

接続プールの特定のサーバー名 / ノード ID に対するカレンダー・ドメイン・サーバー接続の最大数を設定します。

セクション

CONNPOOL

値

[0..S32MAX]

デフォルト値

0

max_masternode

接続プールの特定のサーバー名 / ノード ID に対するマスター・ノード接続の最大数を設定します。

セクション

CONNPOOL

値

[0..S32MAX]

デフォルト値

0

max_sysop

接続プールの特定のサーバー名 / ノード ID に対する Calendar SYSOP 接続の最大数を設定します。

セクション

CONNPOOL

値

[0..S32MAX]

デフォルト値

0

max_user

接続プールの特定のサーバー名 / ノード ID に対するユーザー接続の最大数を設定します。

セクション

CONNPOOL

値

[0..S32MAX]

デフォルト値

なし。接続プールを使用するために必要な値。

securitycachesize

セキュリティ・レコード・キャッシュに保持するエントリの最大数を設定します。

セクション

CACHE

値

[0..U32MAX]

デフォルト値

512

tzcachesize

タイムゾーン・レコード・キャッシュに保持するエントリの最大数を設定します。

セクション

CACHE

値

[0..U32MAX]

デフォルト値

256

Oracle Calendar SDK のフラグおよび機能

CSDK のフラグ

CSDK_FLAG_NONE

デフォルトの動作を選択します。

CAPI_FLAG_STORE_DELPROPS

CAPI_Store* 関数とともに使用して、指定されたプロパティをサーバー上でクリアまたは削除するように指定します。

CAPI_FLAG_STORE_MODPROPS

CAPI_Store* 関数とともに使用して、他のプロパティを変更せずに（可能な場合）、指定されたプロパティをサーバー上で変更するように指定します。

CSDK_FLAG_FETCH_AGENDA_ATTENDEE_ONLY

CSDK_FetchEvent* コールとともに使用して、参照している予定表以外の出席者全員の ATTENDEE プロパティをフィルタリングします。

CSDK_FLAG_FETCH_COMBINED

CSDK_FetchEvent* コールとともに使用して、予定表ごとに 1 つの VCALENDAR を返すのではなく、1 つの VCALENDAR にすべてのイベントを入れて返します。

この方が高速で、出力も小さくなります。

CSDK_FLAG_FETCH_DO_NOT_EXPAND_RRULE

CSDK_FetchEvent* コールに渡して、繰返しルールを展開しないように要求します。

このフラグは、デフォルトでは CSDK_FetchEventsByUID とともに設定され、CSDK_FLAG_FETCH_EXPAND_RRULE を使用してオーバーライドできます。

CSDK_FLAG_FETCH_EXCLUDE_ACCEPTED

CSDK_FetchEvent* コールとともに使用して、コール元が承諾したイベントを除外します。

CSDK_FLAG_FETCH_EXCLUDE_APPOINTMENTS

CAPI_FetchEvent* および CSDK_FetchEvent* コールとともに使用して、標準の会議（予定）を除外します。

CSDK_FLAG_FETCH_EXCLUDE_DAILYNOTES

CAPI_FetchEvent* および CSDK_FetchEvent* コールとともに使用して、メモを除外します。

CSDK_FLAG_FETCH_EXCLUDE_DAYEVENTS

CAPI_FetchEvent* および CSDK_FetchEvent* コールとともに使用して、終日イベントを除外します。

CSDK_FLAG_FETCH_EXCLUDE_DECLINED

CSDK_FetchEvent* コールとともに使用して、コール元が辞退したイベントを除外します。

CSDK_FLAG_FETCH_EXCLUDE_HOLIDAYS

CAPI_FetchEvent* および CSDK_FetchEvent* コールとともに使用して、休日を除外します。

CSDK_FLAG_FETCH_EXCLUDE_NOTOWNER

CSDK_FetchEvent* コールとともに使用して、コール元が所有していないイベントを除外します。

CSDK_FLAG_FETCH_EXCLUDE_UNCONFIRMED

CSDK_FetchEvent* コールとともに使用して、コール元が確認していないイベントを除外します。

CSDK_FLAG_FETCH_EXPAND_RRULE

CSDK_FetchEvent* コールに渡して、繰返しルールを展開するように要求します。

このフラグは、デフォルトでは CSDK_FetchEventsByRange および CSDK_FetchEventsByAlarmRange とともに設定され、CSDK_FLAG_FETCH_DO_NOT_EXPAND_RRULE を使用してオーバーライドできます。

CSDK_FLAG_FETCH_LOCALTIMES

CSDK_FetchEvent* および CSDK_FetchTask* コールとともに使用して、時間をローカル・タイムゾーンで返すように要求します。

Calendar Server で設定されたカレント・ユーザーの優先タイムゾーンがローカル・タイムゾーンとして使用されます。

CAPI_FLAG_FETCH_NO_FIELDHOLDERS

CAPI_FetchEventsBy* とともに使用して、サーバーから一部のイベント・プロパティが読み取られないようにします。

このフラグを使用するとパフォーマンスは大幅に向上しますが、X- プロパティとユーザーが割り当てた UID は返されません。

CSDK_FLAG_FETCH_VCARD_VERSION_2_1

CSDK_FetchContacts* コールとともに使用して、バージョン 2.1 の vCard を要求します。

CSDK_FLAG_FETCH_VCARD_VERSION_3_0

CSDK_FetchContacts* コールとともに使用して、バージョン 3.0 の vCard（デフォルト）を要求します。

CSDK_FLAG_STORE_CREATE

CSDK_Store* コールとともに使用して、サーバーに新しいオブジェクトを作成します。

同じ UID のオブジェクトが存在する場合、エラーが返されます。

CSDK_FLAG_STORE_IMPORT

CSDK_Store* コールとともに使用して、指定された UID のオブジェクト（仕事 / 連絡先）がサーバーに存在しない場合は作成（CSDK_FLAG_STORE_CREATE）し、存在する場合は完全に置き換えます（CSDK_FLAG_STORE_REPLACE）。

CSDK_FLAG_STORE_INVITE_SELF

CSDK_StoreEvents とともに使用して、ATTENDEE プロパティを使用せずにカレント・ユーザーに出席依頼します。

CSDK_FLAG_STORE_MODIFY

CSDK_Store* コールとともに使用して、指定されたプロパティをサーバー上の既存オブジェクトに追加するか、そのプロパティを変更します。

CSDK_FLAG_STORE_REMOVE

CSDK_StoreContacts および CSDK_StoreTasks* コールとともに使用して、指定されたプロパティをサーバー上の既存オブジェクトから削除します。

入力により UID プロパティを指定する必要がありますが、プロパティ自体はサーバーから削除されません。このモードでは、サーバー上のオブジェクトは完全に削除されません。完全に削除するには CSDK_Delete* を使用します。

CSDK_FLAG_STORE_REPLACE

CSDK_Store* 関数とともに使用して、サーバー上のオブジェクト（イベント / 仕事 / 連絡先）を、指定されたオブジェクトで完全に置き換えるように指定します。

CSDK_FLAG_STORE_REPLY

CSDK_StoreEvents とともに使用して、他の人が開催するイベントに返信します（ATTENDEE および VALARM を設定）。

CSDK_FLAG_STREAM_NOT_MIME

CSDK_Store* および CSDK_Fetch* コールとともに使用して、ストリームを MIME でカプセル化しないように指定します。

機能

Typedefs

long 型を CAPICapabilityID 型に typedef します。

CAPI_CAPAB_AUTH

サーバーでサポートされている認証メカニズムを返します。

例: cs-standard,gssapi:kerberos5,sasl:KERBEROS_V4。この機能を読み取るには、サーバー接続が存在している必要があります。

CAPI_CAPAB_CAPI_VERSION

SDK バージョンを文字列として返します。

例: 9.0.4

CAPI_CAPAB_COMP

サーバーでサポートされている圧縮メカニズムを返します。

例: cs-simple,none。この機能を読み取るには、サーバー接続が存在している必要があります。

CAPI_CAPAB_ENCR

サーバーでサポートされている暗号化メカニズムを返します。

例: cs-light,none。この機能を読み取るには、サーバー接続が存在している必要があります。

CAPI_CAPAB_MAXDATE

SDK が処理できる日付の最大値を返します (20371129)。

CAPI_CAPAB_SERVER_VERSION

サーバーのバージョンを文字列として返します。

例: 6.0。この機能を読み取るには、サーバー接続が存在している必要があります。

CAPI_CAPAB_UNSUPPORTED_ICAL_COMP

SDK が処理しない iCal コンポーネントのカンマで区切られたリストを返します。

例： VJOURNAL, VFREEBUSY

CAPI_CAPAB_UNSUPPORTED_ICAL_PROP

SDK が処理しない iCal プロパティのカンマで区切られたリストを返します。

例： GEO, COMMENT。この機能を読み取るには、サーバー接続が存在している必要があります。

CAPI_CAPAB_VERSION

CAPI_CAPAB_CAPI_VERSION と同じです。

Calendar SDK のステータス・コード

この章では、SDK 関数から返されるすべての `CAPIStatus` 値についてアルファベット順に説明します。関数 `CSDK_GetStatusString` および `CSDK_GetStatusLevels` は、`CAPIStatus` 値を解釈する際に便利です。

CAPI_STAT_API

API クラス・ステータス。

CAPI_STAT_API_BADPARAM

不良パラメータが渡されました。

CAPI_STAT_API_CALLBACK

コールバックに問題がありました。

CAPI_STAT_API_CALLBACK_ERROR

コールバックがエラーを返しました。ビット・フィールド 5 に返されます。

CAPI_STAT_API_FLAGS

不良フラグが渡されました。

CAPI_STAT_API_HANDLE

ハンドルに問題がありました。

CAPI_STAT_API_HANDLE_BAD

渡されたハンドルが破損していました。

CAPI_STAT_API_HANDLE_NOTNULL

渡されたハンドルが NULL ではありませんでした。

CAPI_STAT_API_HANDLE_NULL

渡されたハンドルが NULL でした。

CAPI_STAT_API_NULL

NULL ポインタが渡されました。

CAPI_STAT_API_SESSION

セッションに問題がありました。

CAPI_STAT_API_SESSION_BAD

渡されたセッションが破損していました。

CAPI_STAT_API_SESSION_NOTNULL

渡されたセッションが NULL ではありませんでした。

CAPI_STAT_API_SESSION_NULL

渡されたセッションが NULL でした。

CAPI_STAT_API_STREAM

ストリームに問題がありました。

CAPI_STAT_API_STREAM_BAD

渡されたストリームが破損していました。

CAPI_STAT_API_STREAM_NOTNULL

渡されたストリームが NULL ではありませんでした。

CAPI_STAT_API_STREAM_NULL

渡されたストリームが NULL でした。

CAPI_STAT_DATA

データ・クラス・ステータス。

CAPI_STAT_DATA_COOKIE

提供された Cookie に関する情報。

CAPI_STAT_DATA_DATE

日付に関する情報。

CAPI_STAT_DATA_DATE_FORMAT

日付データの書式が正しくありません。

CAPI_STAT_DATA_DATE_INVALID

指定された日付が無効です (Feb 30th など)。

CAPI_STAT_DATA_DATE_OUTOFRANGE

指定された日付が、実装でサポートされている範囲外です。

CAPI_STAT_DATA_DATE_RANGE

日付範囲が正しくありません。

CAPI_STAT_DATA_EMAIL

電子メールに関する情報。

CAPI_STAT_DATA_EMAIL_NOTSET

サーバーで、1人以上のユーザーまたは1つ以上のリソースに電子メール・アドレスが設定されていません。

CAPI_STAT_DATA_ENCODING

提供されたデータのエンコードに関する情報。

CAPI_STAT_DATA_HOSTNAME

ホスト名に関する情報。

CAPI_STAT_DATA_HOSTNAME_FORMAT

ホスト名の書式が正しくありませんでした。

CAPI_STAT_DATA_HOSTNAME_HOST

ホスト名文字列をホストに解決できませんでした。

CAPI_STAT_DATA_HOSTNAME_SERVER

指定されたホストおよびポートでサーバーが見つかりませんでした。

CAPI_STAT_DATA_ICAL

iCalendar データに関する情報。

CAPI_STAT_DATA_ICAL_COMPEXTRA

余分なコンポーネントが検出されました。1度だけ必要なコンポーネントが複数回指定されているか、使用できないコンポーネントが使用されています。

CAPI_STAT_DATA_ICAL_COMPMISSING

想定されたコンポーネントまたは必須コンポーネントがありませんでした。

CAPI_STAT_DATA_ICAL_COMPNAME

コンポーネント名に問題がありました。

CAPI_STAT_DATA_ICAL_COMPVALUE

コンポーネントの内容に問題がありました。

CAPI_STAT_DATA_ICAL_FOLDING

行の折返しに問題がありました。

CAPI_STAT_DATA_ICAL_IMPLEMENT

この特定の iCalendar 実装に問題があります。

CAPI_STAT_DATA_ICAL_LINEOVERFLOW

iCal データ行の 1 つが長すぎました。iCalendar 仕様（RFC 2445）違反です。

CAPI_STAT_DATA_ICAL_NONE

提供されたデータは iCalendar データではありませんでした。

CAPI_STAT_DATA_ICAL_OVERFLOW

iCalendar データの解析時にオーバーフローがありました。これは、仕様違反ではなく、iCalendar ライブラリの内部制約が原因です。

CAPI_STAT_DATA_ICAL_PARAMEXTRA

余分なパラメータが検出されました。1 度だけ必要なパラメータが複数回指定されているか、使用できないパラメータが使用されています。

CAPI_STAT_DATA_ICAL_PARAMMISSING

想定されたパラメータまたは必須パラメータがありませんでした。

CAPI_STAT_DATA_ICAL_PARAMNAME

パラメータ名に問題がありました。

CAPI_STAT_DATA_ICAL_PARAMVALUE

パラメータ値に問題がありました。

CAPI_STAT_DATA_ICAL_PROPEXTRA

余分なプロパティが検出されました。1 度だけ必要なプロパティが複数回指定されているか、使用できないプロパティです。

CAPI_STAT_DATA_ICAL_PROPMISSING

想定されたプロパティまたは必須プロパティがありませんでした。

CAPI_STAT_DATA_ICAL_PROPNAME

プロパティ名に問題がありました。

CAPI_STAT_DATA_ICAL_PROPVALUE

プロパティ値に問題がありました。

CAPI_STAT_DATA_ICAL_RECURMODE

繰返しの指定に問題がありました。CAPI_StoreEvent の説明で規定されたルールに違反しています。

CAPI_STAT_DATA_MIME

MIME データに関する情報。

CAPI_STAT_DATA_MIME_CHARSET

MIME ヘッダーで、サポートされていないキャラクタ・セットが指定されました。

CAPI_STAT_DATA_MIME_COMMENT

コメントを解析できませんでした。

CAPI_STAT_DATA_MIME_ENCODING

MIME オブジェクトで指定されたエンコードはサポートされていません。

CAPI_STAT_DATA_MIME_HEADER

ヘッダーを解析できませんでした。

CAPI_STAT_DATA_MIME_IMPLEMENT

この MIME 実装に固有の制約に違反しています。

CAPI_STAT_DATA_MIME_IMPLEMENT_NESTING

MIME オブジェクトのネストが深すぎました。

CAPI_STAT_DATA_MIME_LENGTH

ヘッダー行の 1 つが長すぎました。

CAPI_STAT_DATA_MIME_NOICAL

ヘッダーは iCalendar データが含まれていることを示していますが、MIME パートが見つかりませんでした。

CAPI_STAT_DATA_MIME_NONE

MIME データが見つかりませんでした。

CAPI_STAT_DATA_MIME_OVERFLOW

MIME データの読み取り中にオーバーフローが発生しました。

CAPI_STAT_DATA_UID

UID に関する情報。

CAPI_STAT_DATA_UID_FORMAT

UID 文字列の書式が正しくありませんでした。

CAPI_STAT_DATA_UID_NOTFOUND

指定された UID のオブジェクトが見つかりませんでした。

CAPI_STAT_DATA_UID_RECURRENCE

指定された繰返しが見つかりませんでした。

CAPI_STAT_DATA_USERID

userID に関する情報。

CAPI_STAT_DATA_USERID_EXT

UserId 文字列の拡張部分に問題がありました。

CAPI_STAT_DATA_USERID_EXT_CONFLICT

userID と X.400 が指定されたか、ノードとカレンダー・ドメインの両方が指定されました。

CAPI_STAT_DATA_USERID_EXT_FORMAT

拡張文字列の書式が正しくありませんでした。

CAPI_STAT_DATA_USERID_EXT_INIFILE

構成ファイルに問題がありました。

CAPI_STAT_DATA_USERID_EXT_MANY

文字列で複数のユーザーが識別されました。

CAPI_STAT_DATA_USERID_EXT_NODE

指定されたノードが見つかりませんでした。

CAPI_STAT_DATA_USERID_EXT_NODE

文字列でユーザーが識別されませんでした。

CAPI_STAT_DATA_USERID_FORMAT

UserId 文字列の書式が正しくありませんでした。

CAPI_STAT_DATA_USERID_ID

UserId 文字列の Id 部分に問題がありました。

CAPI_STAT_LIBRARY

ライブラリ・クラス・ステータス。

CAPI_STAT_LIBRARY_IMPLEMENTATION

機能が完全に実装されていません。

CAPI_STAT_LIBRARY_INTERNAL

ライブラリで内部エラーが発生しました。

CAPI_STAT_LIBRARY_INTERNAL_COSMICRAY

内部でまったく予期しないことが発生しました。

CAPI_STAT_LIBRARY_INTERNAL_DATA

ライブラリ内のデータが破損していました。

CAPI_STAT_LIBRARY_INTERNAL_EXPIRY

ライブラリが異常終了しました。

CAPI_STAT_LIBRARY_INTERNAL_FUNCTION

ライブラリで、関数が誤ってコールされました。

CAPI_STAT_LIBRARY_INTERNAL_OVERFLOW

内部で最大値のいずれかを超過しました。

CAPI_STAT_LIBRARY_INTERNAL_PROTOCOL

ライブラリで、プロトコルが誤って使用されました。

CAPI_STAT_LIBRARY_INTERNAL_UNKNOWN_EXCEPTION

SDK が未知の C++ 例外を受け取りました。

CAPI_STAT_LIBRARY_INTERNAL_UNKNOWN_LIBRARY_ERRCODE

依存ライブラリからのエラー・コードのマッピングに失敗しました。

CAPI_STAT_LIBRARY_SERVER

サーバーの制約またはサーバーでの状態変化です。

CAPI_STAT_LIBRARY_SERVER_BUSY

サーバーがビジー状態のため、すぐに要求を処理することができません。

CAPI_STAT_LIBRARY_SERVER_SUPPORT

サーバーは、必要なサポートの一部を提供していません。

CAPI_STAT_LIBRARY_SERVER_SUPPORT_CHARSET

必要なキャラクタ・セットに対するサポートがありません。

CAPI_STAT_LIBRARY_SERVER_SUPPORT_STANDARDS

このサーバーでは SDK はサポートされていません。

CAPI_STAT_LIBRARY_SERVER_SUPPORT_UID

UID の格納はサポートされていません。

CAPI_STAT_LIBRARY_SERVER_USERDATA

サーバーのユーザー・データに問題があります。

CAPI_STAT_OK

操作は正常に完了しました。値は 0 です。

CAPI_STAT_SECUR

セキュリティ・クラス・ステータス。

CAPI_STAT_SECUR_LOGON

ログオン時にセキュリティ・エラーが発生しました。

CAPI_STAT_SECUR_LOGON_AUTH

ログオン認証に失敗しました。

CAPI_STAT_SECUR_LOGON_LOCKED

指定されたアカウントはロックされています。

CAPI_STAT_SECUR_LOGON_LOCKED_RESOURCE

リソースに対するログオンがロックされています。

CAPI_STAT_SECUR_LOGON_LOCKED_SYSOP

Calendar SYSOP に対するログオンがロックされています。

CAPI_STAT_SECUR_READ

読取り時にセキュリティ・エラーが発生しました。

CAPI_STAT_SECUR_READ_ALARM

アラーム・データの読取り時にセキュリティ・エラーが発生しました。

CAPI_STAT_SECUR_READ_PROPS

プロパティの読取り時にセキュリティ・エラーが発生しました。

CAPI_STAT_SECUR_SERVER

サーバーでセキュリティ・エラーが発生しました。

CAPI_STAT_SECUR_SERVER_LICENSE

サーバーでライセンス・エラーが発生しました。

CAPI_STAT_SECUR_SERVER_SET_IDENTITY_SYSOP

サーバーで操作を実行するには、Calendar SYSOP のログオン後に SetIdentity をコールする必要があります。

CAPI_STAT_SECUR_WRITE

書き込み時にセキュリティ・エラーが発生しました。

CAPI_STAT_SECUR_WRITE_AGENDA

予定表への書き込み時にセキュリティ・エラーが発生しました。

CAPI_STAT_SECUR_WRITE_EVENT

イベントへの書き込み時にセキュリティ・エラーが発生しました。

CAPI_STAT_SERVICE

サービス・クラス・ステータス。

CAPI_STAT_SERVICE_ACE

ACE プラグインの 1 つを原因とする問題が発生しました。

CAPI_STAT_SERVICE_ACE_LOAD

必要な ACE プラグインをロードできませんでした。

CAPI_STAT_SERVICE_ACE_SUPPORT

要求された ACE オプションはサポートされていません。

CAPI_STAT_SERVICE_FILE

システム・ファイル・サービスに問題がありました。

CAPI_STAT_SERVICE_FILE_CLOSE

ファイルを閉じる際に問題が発生しました。

CAPI_STAT_SERVICE_FILE_DELETE

ファイルを削除する際に問題が発生しました。

CAPI_STAT_SERVICE_FILE_MODE

ファイルの読取りまたは書き込みモードに問題がありました。

CAPI_STAT_SERVICE_FILE_OPEN

ファイルを開く際に問題が発生しました。

CAPI_STAT_SERVICE_FILE_READ

ファイルからの読取り時に問題が発生しました。

CAPI_STAT_SERVICE_FILE_TEMP

一時ファイルの割当て時に問題が発生しました。

CAPI_STAT_SERVICE_FILE_WRITE

ファイルへの書込み時に問題が発生しました。

CAPI_STAT_SERVICE_LIBRARY

標準ライブラリ・サービスに問題がありました。

CAPI_STAT_SERVICE_MEM

システム・メモリー・サービスに問題がありました。

CAPI_STAT_SERVICE_MEM_ALLOC

メモリーを割り当てられませんでした。

CAPI_STAT_SERVICE_NET

ネットワーク・サービスに問題がありました。

CAPI_STAT_SERVICE_NET_TIMEOUT

ネットワーク・サービスの待機中にタイムアウトが発生しました。

CAPI_STAT_SERVICE_THREAD

システム・スレッド・サービスに問題がありました。

CAPI_STAT_SERVICE_TIME

標準時間サービスに問題がありました。

CAPI_STAT_SERVICE_TIME_GMTIME

GMTime を取得できませんでした。

Oracle Calendar SDK の FAQ および トラブルシューティング

この章では、Oracle Calendar SDK に関する FAQ とトラブルシューティングについて説明します。

FAQ

新しい Oracle Calendar Server で SDK の旧バージョン（CAPI）を使用できますか。既存のアプリケーションは併用できますか。

最新バージョンの Oracle Calendar SDK には、9.0.4 より前のリリース用の移行ユーティリティが含まれます。このユーティリティを使用して、古いアプリケーションを新しい Calendar Server に移植できます。

Steltor パッケージと最新の Oracle Calendar SDK はどこが違うのですか。

Oracle Calendar SDK リリース 9.0.4 には、多くの新機能が追加されています。新しいサーバーではパフォーマンスが向上し、処理効率が改善されているほか、SDK パッケージに次の機能が追加されています。

- 接続プーリング設計
- JNI 関数
- リモート代理のサポート
- タスクおよび連絡先のサポート
- Calendar SYSOP ログイン

Visual Basic や他のプログラミング言語を使用して、Oracle Calendar SDK プログラムを作成できますか。

Oracle Calendar SDK は C/C++ 関数コールのパッケージであるため、C をネイティブでサポートできる任意の言語を使用してラッパーを作成し、これらの関数にアクセスできます。

Visual Basic、Perl、Java および Python を使用できます。これらの他にも、様々な取組みが行われています。

イベントからユーザーへの出席依頼を取り消すにはどうすればいいですか。

CSDK_StoreEvents をコールし、ユーザーの ATTENDEE プロパティを指定せずに iCalendar を使用してイベントを置き換えるか、出席者のリストから削除するユーザーの CAPI_GetHandle を使用し、CAPI_DeleteEvent をコールします。

ユーザーにイベントへの出席を依頼するにはどうすればいいですか。

CSDK_StoreEvents のコール時に、ユーザーの ATTENDEE プロパティを指定します。

Oracle Calendar SDK を使用して、Calendar Server 管理者（Calendar SYSOP）としてログインできますか。

できます。Calendar SYSOP ログインは、リリース 9.0.4 の Oracle Calendar SDK からサポートされています。

イベントを完全に削除するにはどうすればいいですか。

イベントの UID を指定して、関数 `CSDK_DeleteEvents` を使用します。

トラブルシューティング

Oracle Calendar SDK のデモを実行すると、「libcapi.so not found, no such file or directory」または「Cannot load library libcapi」というエラーが表示されます。どのように解決すればよいでしょうか。

この問題については、最新の Oracle Calendar SDK README ファイルを参照してください。

Oracle Calendar SDK 関数をコールするとエラー 2148073984 が返されます。このエラーの意味を教えてください。Oracle Calendar SDK エラーの詳細は、何を参照すればいいですか。

第 6 章「[Calendar SDK のステータス・コード](#)」に、Oracle Calendar SDK のステータス・コードのリストがあります。各コードは 5 つのフィールドに分けられ、それぞれが異なるレベルの問題について説明しています。

2 つのヘルパー関数 `CSDK_GetStatusLevels()` と `CSDK_GetStatusString()` を使用すると、エラーを簡単に解釈できます。Oracle Calendar SDK に付属のデモ・アプリケーションに、エラー・コードの解説方法の例が含まれています。

Oracle Calendar SDK を使用してイベントをフェッチする際、ATTENDEE プロパティを取得できません。

iCalendar の ATTENDEE プロパティを生成するには電子メール・アドレスが必要なため、Oracle Calendar SDK では、iCalendar イベントのエクスポート時に電子メール・アドレスのない出席者はスキップされます。この問題を解決するには、Calendar Server にすべての出席者の電子メール・アドレスが格納されていることを確認してください。

Oracle Calendar SDK を使用して取り出したイベントでは、アクセント付きの e 文字が =C3=A9 として表示されます。

Oracle Calendar SDK では、出力は UTF-8 でエンコードされます。UTF-8 では、この文字は 2 バイトで表され、0xC3 と 0xA9 になります。MIME でカプセル化されたデータを返す際、Oracle Calendar SDK ではさらにこれらのバイトが引用符付きの印刷可能文字列でエンコードされ、これが =C3=A9 となります。

**ATTENDEE プロパティに不要な =3D 文字がたくさんあるように見えます。
バグなのでしょう。**

いいえ。これはバグではありません。=3D は、実際は引用符付き印刷可能文字列でエンコードされた等号 (=) です。したがって、partstat=confirmed と表示されるはずの部分が、実際には partstat=3Dconfirmed と表示されます。

**5 つの ATTENDEE プロパティを使用して iCalendar イベントを格納しました。
ネイティブのクライアントには 1 つも表示されません。**

CAPL_StoreEvents を使用して複数の人にイベントへの出席を依頼する場合、各出席者に CAPIHandle を提供する必要があります。

Microsoft Visual C++ を使用して Oracle Calendar SDK をコンパイルすると、LNK1106 エラーが返されます。

Oracle Calendar SDK は Visual C++ 6 を使用してコンパイルされています。旧バージョンを動作させるにはパッチが必要です。入手可能なパッチについては、Microsoft 社の Web サイトを検索してください。

Visual C++ を使用することで、他の C コンパイラで発生するリンク・エラーを減らすことができます。推奨コンパイラおよびリンカーの設定についてはデモ・アプリケーション・プロジェクトを参照してください。

**CSDK_CreateFileStream() を使用するとプログラムが常に異常終了します。
他のタイプのストリームを試すと、正常に機能します。**

静的 C ライブラリではなく、Microsoft 社の C ランタイム・ライブラリ（コマンドラインを使用している場合は /MD スイッチ）を使用して、アプリケーションをリンクする必要があります。これは、ファイル・ストリームで使用される FILE ポインタの定義が、使用する C ライブラリのバージョンによって異なるためです。

CSDK_CreateFileStreamFromFilenames を使用できます。この場合、SDK とアプリケーションの間で FILE ポインタを受け渡す必要はありません。

CSDK_GetHandle() からリソースを検出できないと通知されます。

いくつかの原因が考えられます。

1. リソース名に「à」など、アクセント付きの文字や特殊文字が含まれている場合。
Oracle Calendar SDK ではユーザーおよびリソースの識別情報文字列には UTF-8 を使用する必要があります。適切な UTF-8 値を使用して、アクセント付き文字または特殊文字を記述してください。
2. リソース名に「Training / Meeting Room」など、スラッシュ (/) が使用されている場合。
リソース名内部でこの文字を使用すると、デフォルトのフィールド・デリミタとして使用されている / と競合します。この問題を解決するには、RS=Work/ というように、デリミタとして別の文字を使用します。

CAPI_StoreEvent を使用すると、Oracle Calendar SDK で自分の vAlarm 情報が無視されます。

イベントに vAlarm（リマインダ）を設定できるのは、自分の予定表だけです。

CAPI_FetchEventsByRange() を使用しても、自分のイベント・リマインダがまったく表示されません。

これは、この関数の正常な動作です。リマインダ（vAlarms）を取得するには、関数 CSDK_FetchEventsBy...、CAPI_FetchEventByID() または CAPI_FetchEventsByAlarmRange() を使用することをお勧めします。

第Ⅱ部

Oracle Calendar Web Services ツールキット

第Ⅱ部では、Oracle Calendar Web Services ツールキットについて説明します。

第Ⅱ部は、次の章で構成されています。

- 第8章「Oracle Calendar Web Services ツールキットの概要」
- 第9章「Oracle Calendar Web Services SOAP」
- 第10章「Oracle Calendar Web Services のクライアント側 Java 実装」
- 第11章「Oracle Calendar Web Services のステータス・コード」

Oracle Calendar Web Services ツールキット の概要

この章では、Oracle Calendar Web Services および Oracle Calendar Web Services ツールキットの概要を説明します。

関連資料：

- IETF iCal RFC: <http://www.ietf.org/rfc/rfc2445.txt>
- Web サービス・アクティビティ: <http://www.w3.org/2002/ws/>
- Web Services Description Language (WSDL) : <http://www.w3.org/TR/wsdl>
- Simple Object Access Protocol (SOAP) 1.1: <http://www.w3.org/TR/SOAP/>

Oracle Calendar Web Services について

Oracle Calendar Web Services は、Oracle Calendar アプリケーション・システムのコンポーネントで、アプリケーション・レベルのサービスを処理します。アプリケーションは、Web サービスを使用して一般的な XML 問合せを実行し、カレンダー・データを取得し、任意のポータル、クライアント・アプリケーションまたはバックエンド・サーバーに表示できます。iCal データは XML でコード化され、iCal は xCal になります。SOAP は、配信するメッセージのカプセル化に使用されます。カレンダー・データの Web サービス SOAP は、Calendar Server ストアに直接格納されます。これは、実際にはカレンダー Web サービス言語 (CWSL) です。

Oracle Calendar Web Services は、3 つの主要な要素で構成されています。1 番目の要素は SOAP で、クライアントとサーバーが HTTP を介してオブジェクトを交換するための通信プロトコルを提供します。2 番目の要素は Web Services Description Language (WSDL) で、サードパーティから Oracle Calendar Web Services に機能の説明を問合せできます (IDL と同様)。最後の要素は、Web サービスを公開し、クライアントの検索先となるレジストリです。この要素は、Universal Description, Discovery and Integration (UDDI) と呼ばれます。

現在のこの実装では、WSDL または UDDI サポートは提供されていません。ただし、今後のバージョンで、WSDL を UDDI レジストリに公開する機能が提供される可能性があります。

Oracle Calendar Web Services ツールキット

開発者は、Oracle Calendar Web Services ツールキットを使用して、Web サービス・アプリケーションを構築し、SOAP 1.1 問合せを作成できます。ツールキットには、カレンダー・イベントの検索、作成、変更および削除機能、およびタスクの検索機能が含まれています。ツールキットは、カレンダーレットと呼ばれる一連の Java クラスを使用して、Calendar Server データベースへの SOAP アクセスを提供します。開発者は、Java IDE を使用して、アプリケーションの構築に必要な XML 構造を抽象化できます。

カレンダーレットを使用して、独自のクライアントを作成し、カレンダー・データを自分のアプリケーションに統合します。ポータル、クライアント・アプリケーション、またはバックエンド・サーバーにカレンダー・データを統合するには、Web サービスのホストとなる Web サーバーへの HTTP 接続、SOAP メッセージの生成、SOAP 応答の解析 (HTTP 文字列の送受信が可能なテクノロジーを使用)、および既存の XML ツールキットの使用により、SOAP クライアント・ツールキットで送信 HTTP 文字列の生成と着信 HTTP 文字列の解析を行えるようにする必要があります。ツールキットは、HTTP プロキシの使用をサポートしています。

ツールキットの内容

Oracle Calendar Web Services ツールキットには次のものが含まれています。

- `Calendarlet.tar`: カレンダレット Jar ファイル
- `Javadoc.tar`: カレンダレットの Javadoc HTML ドキュメント
- `Ws_testtool.tar`: サンプル・ソース・コードを含む、Oracle Calendar Web Services ツールキット・テスト・ツールの Java ソース

セキュリティ

- Basic 認証
 - プレーン・テキストの `userID` およびパスワードを使用します。
 - カレンダをトラステッド環境外で利用できるようにする場合は、HTTP Server を SSL 用に構成します。
 - マスター・ノード Calendar Server 構成が必要です。

すべての認証は、Calendar Server によって実行されます。Java 側のクラスでは、Web サービスを使用して、処理を加えることなく情報を渡します。ただし、セキュリティ情報を認証するには、それらの情報が Java Web サービス・クラス内で設定されている必要があります。

Oracle Calendar Web Services SOAP

この章では、Oracle Calendar Web Services ツールキットで XML と Simple Object Access Protocol (SOAP) を使用して、iCalendar オブジェクトを取得し、格納する方法について説明します。

HTTP ヘッダー

正しい SOAP v1.1 トランザクションの HTTP ヘッダーには次の要素が含まれている必要があります。

```
POST <uri> HTTP/1.1
Content-Type: text/xml; charset="UTF-8"
Content-Length: <char length>
SOAPAction: <urn>
```

...soap envelope...

<uri> は通常、Oracle Calendar アプリケーション・システムの URI (通常は /ocas-bin/ocas.fcgi) です。これは、主に Web サーバー (Oracle HTTP Server または Apache) が アプリケーション・システムの識別、内部 fcgi プロトコル・モジュールの起動、Web サービスへの要求の受け渡しを行うために使用します。

Oracle Collaboration Suite 内では、シングル・サインオン・メカニズム (SSO) を回避するために、別の URL が必要になる場合があります (通常は /ocws-bin/ocas.fcgi)。

Content-Type charset の識別子はオプションです。指定がない場合、UTF-8 とみなされます。ただし、サポートされている charset エンコーディングは UTF-8 のみです。他のすべての charset はエラーになります。

エラーの HTTP 応答は、ステータス・コード 500 です (内部サーバー・エラー)。実際の SOAP エンベロープが破損している場合 (受信したデータの内容を特定できない場合)、または SOAP レベル・エラーが発生した場合に、このエラーが返されます。すべてのアプリケーション・レベル・エラーは、SOAP 障害内で HTTP ステータス・コード 500 とともに返されます。

```
HTTP/1.1 500 Internal Server Error
Content-Type: text/xml
Content-Length: <char length>
```

...Optional soap envelope...

SOAP エンベロープが正しく実行された場合、SOAP 情報は正しく、アプリケーション・レベルの関数は成功し、ステータス・コード 200 (成功) が返されます。

```
HTTP/1.1 200 OK
Content-Type: text/xml
Content-Length: <char length>
```

... Soap envelope...

セキュリティおよび認証

この項では、次の項目について説明します。

- セキュリティおよび認証設計
- Basic 認証

データの暗号化は非常に重要なセキュリティ要素ですが、現時点では、SOAP 要求内でデータを暗号化する計画はありません。

設計

SOAP の分野では、SOAP メッセージの認証、セキュリティおよび暗号化の定義と標準化について様々な取組みが行われています。W3C、IETF、OASIS、WS-I などのグループが、この目的のために尽力しています。Oracle Calendar Web Services の開発時点では、決定的な仕様は確立されていません。ただし、Web サービスがサポートする機能の定義については、次の傾向が一般的と考えられています。

- HTTP SSL および Web ベースの証明書
- 簡易認証
- アプリケーション固有の認証メカニズム（Oracle Collaboration Suite 用）

注意： 策定中の仕様 WS-Security では、いくつかの認証メカニズムのアウトラインが示されていますが、これはまだ業界の承認を得ていないドラフト段階です。

これらのすべてのメカニズムでは、HTTP レベルの機能（つまり SSL と証明書）を除く必要な情報を一連の SOAP ヘッダーに含める方法が採用されています。

```
<SOAP-ENV:Envelope>
  <SOAP-ENV:Header>
    ... some encryption, signature, and
      authentication info goes here ...
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    ... a soap method goes here ...
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope >
```

アプリケーション・レイヤーでは、プレーン・テキスト認証のみがサポートされています。ユーザーのパスワードは、各トランザクションの最初に（Base64 でエンコードされた文字列ではなく）プレーン・テキストのみで指定する必要があります。

セキュリティは、トランスポート、プロトコルおよびアプリケーション・レベルで提供されます。HTTP レベルでは、2つのオプション（標準およびSSL）があります。このレイヤーは、Web サーバー・レベル（つまり Apache と Oracle HTTP Server）で完全に処理され、HTTP クライアントと HTTP サーバー間で暗号化されたデータが提供されます。Calendar Application Server には、このレベルでの依存性はありません。

SOAP クライアントは SSL をサポートする必要があります。このサポートを提供していないツールキットもあります。

Basic 認証

Web サービスの Basic 認証は、SOAP ヘッダーを使用して実装されます。

初期バージョンでは、各要求のヘッダーに BasicAuth 要素が必要です。この要素がない場合、SOAP 障害が生成されます。

```
HTTP/1.1 200 OK
Content-Type: text/xml;charset="UTF-8"
Content-Length: <char length>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <soap:Header>
    <auth:BasicChallenge
      xmlns:auth=
        "http://www.soap-authentication.org/2002/01/">
      <Realm>Oracle Web Services</Realm>
    </auth:BasicChallenge>
  </soap:Header>
  <soap:Body>
    <Reply xmlns:cws1=
      "http://www.oracle.com/Webservices/Calendaring/1.0/">
    </Reply>
  </soap:Body>
</soap:Envelope>
```

<Realm> は、クライアントにヒントを提供するために使用されます。これは、ocws.conf ファイルの構成可能パラメータです。

```
[basicauth]
Realm=Oracle Web Services          # default
```

Basic 認証を使用する一般的な SOAP セッションでは、最初のメッセージの SOAP ヘッダーにユーザーの資格証明が含まれます。

```
POST <uri> HTTP/1.1
Content-Type: text/xml; charset="UTF-8"
Content-Length: <char length>
SOAPAction: <urn>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <soap:Header>
    <auth:BasicAuth
      xmlns:auth="http://www.soap-authentication.org/2002/01/">
      <Name>myname</Name>
      <Password>mypassword</Password>
    </auth:BasicAuth>
  </soap:Header>
  <soap:Body>
    ...
  </soap:Body>
</soap:Envelope>
```

ユーザー名は、Calendar Server のユーザー ID です。X.400 ログインは使用できません。また、ユーザー ID とパスワードは、XML で正しくエンコードされている必要があります。

Basic 認証に失敗すると、問題の原因を示す SOAP 障害が返されます。

```
HTTP/1.1 500 Internal Server Error
Content-Type: text/xml
Content-Length: <char length>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:Server::Data::CalConnection</faultcode>
      <faultstring>A security error occurred</faultstring>
      <detail>
        <cws1:Error xmlns:cws1=
          "http://www.oracle.com/Webservices/Calendaring/1.0/">
          <Code>0020-00-00-00000017</Code>
          ...
        </cws1:Error>
      </detail>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

BasicAuth メカニズムは、主に開発およびテスト目的に使用します。プレーン・テキスト・パスワードを使用しているため、このメカニズムだけでは、セキュリティとしてはほとんど機能しません。本番環境でこのメカニズムを使用する場合は、SLL Web 構成を使用することを強くお勧めします。

問合せの作成

SOAP 問合せでは、Universal Identifier (UID) と Global Unique Identifier (GUID) を使用します。Web サービス API は、カレンダー・ストア・オブジェクトを一意に識別する機能、取得する機能、および最後の使用まで参照を格納する機能に基づいています。Web サービスでは、データに依存しない次のプロパティを使用します。

x-oracle-data-guid

このデータ GUID は、Calendar Server に格納されている様々なデータ型固有プロパティにマップされます。イベントの場合、次のプロパティを使用できます。

```
uid                # a UID settable upon creation
x-oracle-event-guid # identifier of the main event
x-oracle-eventinstance-guid # identifier of the instance within
                        # the event
x-oracle-data-guid  # mapped to x-oracle-eventinstance-guid
```

タスクの場合、次のプロパティを使用できます。

```
uid                # a UID settable upon creation
x-oracle-data-guid # generated internally by the Application
                        # Server (OCAS) and cannot be used against
                        # any other product. This will be changed
                        # once the Calendar Store supports the
                        # x-oracle-task-guid attribute.
```

SOAP エンベロープ

SOAP エンベロープは、SOAP メッセージの識別に使用される定義済 XML パッケージです。

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  ...soap header...
  ...soap body...
</soap:Envelope>
```

Xsi と xsd は、メッセージ内の名前空間を定義するオプションです。これらは、必要に応じて使用されます (SOAP ヘッダーまたは SOAP ボディにこれらを必要とする要素がある場合)。

Xsd は、文字列、整数など基本的な定義済の型定義を指定するために使用されます。Xsi は、エントリの `type` 属性を定義するために使用されます。

```
<location xsi:type="xsd:string">Soleil</location>
```

SOAP 内で型情報を定義するには主に 3 つの方法があります。

- データ・コンテンツ・タイプの使用。両方の当事者間で前もって合意します。この方法は 1 対 1 のサイト統合であるため、一般的な SOAP 対話には適していません。
- XML Schema の使用。スキーマと名前空間を使用して、すべての型情報を関連付けます。
- XML Schema と明示的な型属性の使用。SOAP XML ツリーの各要素に `xsi:type` 属性が必要です。

xCal と CWSL には独自の XML Schema があるため、明示的な型属性は使用しません。

この時点で注意が必要な重要な問題があります。現在ほとんどの SOAP 実装では、SOAP エンベロープの前に XML ドキュメント・ヘッダー行が追加されます。これは、現在の SOAP v1.1 仕様には含まれていませんが、SOAP v1.2 で改善されます。

```
<?xml version="1.0" encoding="UTF-8" ?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  ...soap header...
  ...soap body...
</soap:Envelope>
```

SOAP 実装の一貫性を維持するために、デフォルトの動作では、元の要求に存在する場合は XML ドキュメント・ヘッダーが提供されます。

```
POST <uri> HTTP/1.1
Content-Type: text/xml; charset="UTF-8"
Content-Length: <char length>
User-Agent: <user agent>
SOAPAction: <urn>

...soap envelope...
```

SOAP ボディ

SOAP ボディには、Calendar Server および Oracle Calendar Web Services エラーに対してアクションを実行するための実際のメソッドが含まれます。

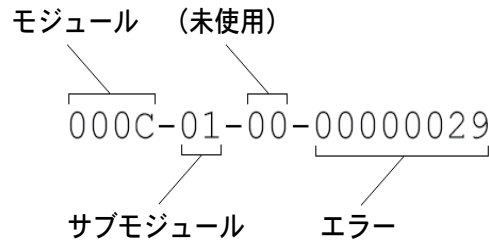
SOAP 障害

なんらかのエラーが返された場合、SOAP 応答のボディには SOAP Fault 要素が含まれます。SOAP 障害内には、特定の要素が提供されます。

- **faultcode:** 次の値のいずれかです。
 - **VersionMismatch:** SOAP 名前空間が誤っていることを示します。
 - **Client:** 着信メッセージの問題であることを示します。
 - **Server:** 要求の処理中に発生した問題であることを示します。
- **faultstring:** 発生したエラーのテキスト・メッセージです。アプリケーション・システムのエラー文字列です。デフォルトの文字列言語は英語です。
- **detail 要素:** 詳細な情報を提供するコンテナとして使用されます。この例では、SOAP クライアントに、完全なアプリケーション・システム・エラー・ログ・エントリが返されます。ocas.conf でサーバー側のイベント・ロギングがデバッグに設定されている場合、Line、FileName、Version、LastMod および Author が返されます。

```
<soap:Body>
  <soap:Fault>
    <faultcode>soap:Server</faultcode>
    <faultstring>Unable to locate the entry in
      the preferences
    </faultstring>
    <detail>
      <cws1:Error xmlns:cws1=
        "http://www.oracle.com/Webservices/Calendaring/1.0/">
        <Class>Error::Data::CalConnection</Class>
        <Code>000C-01-00-00000029</Code>
        <Line>1450</Line>
        <FileName>UniapiConnection.cpp,v</FileName>
        <Version>1.43</Version>
        <LastMod>2002/05/23 20:54:48</LastMod>
        <Author>frederic</Author>
        <Date>Web May 29 14:05:42 2002</Date>
        <PID>19458</PID>
        <TID>5</TID>
      </cws1:Error>
    </detail>
  </soap:Fault>
</soap:Body>
```

たとえば、前述の Code タグでは、エラーのタイプは次のように示されています。



一般に、ユーザーに関連があるのは最初と最後のセグメントだけです。この場合は次が該当します。

- モジュール 000C = SYS_MODULE_DATAACCESS
- エラー 00000029 = e_soapSOAPRequestCode_MissingModifyCmd

モジュールとエラーの値の一覧は、[第 11 章「Oracle Calendar Web Services のステータス・コード」](#)を参照してください。

アプリケーション・システムおよび Calendar Server 内で様々なコンポーネントとの対話が発生する任意の時点で障害が発生する可能性があります。

Calendar Web Service Language (CWSL)

CWSL では、カレンダー SOAP クライアントとカレンダー SOAP サーバー間のデータ交換に使用する文法を定義します。2002 年 3 月付けの CAP ドラフトから直接採用した次のメソッドは、Calendar Web Service Language の主要な機能を提供します。一部の CAP メソッド名は、この CWSL で再使用されていますが、セマンティックと意味は Web ベースのプロトコル環境を反映するように変更されています。

カレンダー言語では、名前空間として "http://www.oracle.com/Webservices/Calendaring/1.0/" を使用します。

次のセッション・コマンドがサポートされています。

- **NoOp:** データ・ストアで操作を実行しません。事前認証に使用されます。

次のカレンダー・コマンドがサポートされています。

- **Create:** 新しい会議を作成します。
- **Delete:** イベントまたはインスタンスを削除します。
- **Modify:** イベントまたはインスタンスの特定のプロパティを更新します。
- **Search:** サービスからデータを取得する要求を実行します。

次のカレンダー・コマンドも定義されています。

- **Ping:** Web サービスがアクティブであることを確認する簡単なチェックを実行します。これらのメソッドの一部は、このリリースでは大幅に制限されているので注意してください。

NoOp

このコマンドは、認証 SOAP ヘッダーの妥当性を確認するために使用されます。サーバーでは何も実行されません。

```
POST <uri> HTTP/1.1
Content-Type: text/xml; charset="UTF-8"
Content-Length: <char length>
SOAPAction: http://www.oracle.com/Webservices/Calendaring/ 1.0/NoOp

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <soap:Header>
    ...
  </soap:Header>
  <soap:Body>
    <cws1:NoOp xmlns:cws1=
      "http://www.oracle.com/Webservices/Calendaring/1.0/">
      <CmdId>a command id</CmdId>
    </cws1:NoOp>
  </soap:Body>
</soap:Envelope>
```

成功したコマンドに対する Web サービスの応答は次のとおりです。

```
HTTP/1.1 200 OK
Content-Type: text/xml
Content-Length: <char length>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  ...
  <soap:Body>
    <cws1:NoOpReply xmlns:cws1=
      "http://www.oracle.com/Webservices/Calendaring/1.0/">
      <CmdId>a command id</CmdId>
    </cws1:NoOpReply>
  </soap:Body>
</soap:Envelope>
```

NoOp コマンドは、SOAP ヘッダーに問題がある場合にのみ失敗します。

Search

この項では、Search および vQuery コマンドを使用してデータを取得する方法を説明します。

Search の使用

このコマンドは、Calendar Server からイベント、タスク、連絡先およびユーザー情報を取得するために使用します。

```
POST <uri> HTTP/1.1
Content-Type: text/xml; charset="UTF-8"
Content-Length: <char length>
SOAPAction: http://www.oracle.com/Webservices/Calendaring/1.0/Search

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <soap:Header>
    <h:BasicAuth xmlns:h="http://soap-authentication.org/2002/10/">
      <Name>myname</Name>
      <Password>mypassword</Password>
    </h:BasicAuth>
  </soap:Header>
  <soap:Body>
    <cws1:Search xmlns:cws1=
      "http://www.oracle.com/Webservices/Calendaring/1.0/">
      <CmdId>a client id</CmdId>
      <vQuery>
        ...
      </vQuery>
    </cws1:Search>
  </soap:Body>
</soap:Envelope>
```

まだ認証されていない場合、<soap:Header> に、すべての機能が 1 回のラウンドトリップで実行されることを確認する <BasicAuth> ブロックを含めることができます。

<CmdId> は、SOAP クライアントが提供する文字列です。応答に含まれ、送信側の Search エントリを識別します。

<vQuery> は検索問合せ条件で、一度だけ使用できます（次の「[vQuery](#)」の項を参照してください）。

返されるデータは、<cws1:Reply> に含まれます。各 <cws1:Search> 要素に 1 つの <cws1:Reply> 要素があります。

```
HTTP/1.1 200 OK
Content-Type: text/xml
Content-Length: <char length>
SOAPAction: http://www.oracle.com/Webservices/Calendaring/1.0/Search

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <soap:Body>
    <cws1:Reply xmlns:cws1=
      "http://www.oracle.com/Webservices/Calendaring/1.0/">
      <CmdId>a client id</CmdId>
      ... data ...
    </cws1:Reply>
  </soap:Body>
</soap:Envelope>
```

<CmdId> は応答に含まれ、送信側の Search コマンドを識別します。

要求の処理中にエラーが発生した場合、SOAP 障害が返されます。複数の <cws1:Search> 要素がある場合も、障害が返されます。

vQuery

vQuery は、Web サービスからデータを要求するベースとなります。初期リリースの vQuery には特に固有の制限があります。

Calendar Server に与える影響を小さくするために、ocws.conf ファイルには、受け取った問合せを無効にするいくつかの設定があります。

```
[webservices]
maxattendee=200          # limit the total number of attendees that can
                          # be returned per instance. The default is 200.
maxresults=200           # limit the total number of meetings or tasks
                          # that can be returned in one query
```

イベントを、1 つの一意識別子または一連の識別子に基づいて問合せできます。

```
<!-- Ability to fetch a single event -->
<!-- Remove the Where clause to return all events -->
<vQuery>
  <Select>*</Select>
  <From>VEVENT</From>
  <Where>x-oracle-data-guid='event guid'</Where>
</vQuery>
```

```

<!-- Ability to fetch a multiple events -->
<vQuery>
  <Select>*</Select>
  <From>VEVENT</From>
  <Where>
    x-oracle-data-guid ='event id 1' OR
    x-oracle-data-guid ='event id 2' OR
    x-oracle-data-guid ='event id 3'
  </Where>
</vQuery>

```

<Select> も使用できますが、現在のバージョンの Oracle Calendar Web Services ではサポートされていません。すべての属性が必要な場合、<Select> は含まれないことがあります。値を「*」にすると、すべてのイベント属性を要求できます。プロパティは、必要な各属性をリストするために複数回表れることがあります。個々の属性のみが必要な場合、明示的にリストする必要があります。

イベントを、日付範囲に基づいて問合せできます。

```

<!-- Ability to fetch events within a time range -->
<vQuery>
  <Select>*</Select>
  <From>VEVENT</From>
  <Where>dtend &gt;= 'starttime' AND dtstart &lt;= 'endtime'</Where>
</vQuery>

```

この例では、starttime と endtime は、UTC で返される時間範囲を指定します。<Where> 句内の文字列の適切な XML エンコーディングに注意してください。

他のイベント問合せ形式を使用すると、エラーが生成されます。

イベント問合せ結果セットは、xCal ドラフト仕様を使用して、<cws1:Reply> タグ内に埋め込まれて返されます。

```

<soap:Body>
  <cws1:Reply xmlns:cws1=
    "http://www.oracle.com/WebServices/Calendaring/1.0/">
    <CmdId>a client id</CmdId>
  <xCal:iCalendar xmlns:xCal=
    "http://www.oracle.com/WebServices/Calendaring/1.0/">
    <vcalendar version="2.0" prodid=...>
      <vevent>
        <x-oracle-data-guid>fjldjfdslkjfdksj
        </x-oracle-data-guid>
        <dtstamp>19980309T231000Z</dtstamp>
        <uid>ffdtasfdtasfdta</uid>
        <summary>My event</summary>
        <location>Soleil</location>
        <x-oracle-eventtype>PUBLIC</x-oracle-eventtype>
      </vevent>
    </vcalendar>
  </xCal:iCalendar>
</cws1:Reply>

```

```
        </vevent>
    </vcalendar>
</xCal:iCalendar>
</cws1:Reply>
</soap:Body>
```

返されたデータのソートは行われません。

タスクを、1 つの一意識別子または一連の識別子に基づいて問合せできます。

```
<!-- Ability to fetch a single task -->
<!-- Remove the Where clause to return all tasks -->
<vQuery>
    <Select>*</Select>
    <From>VTOD</From>
    <Where>x-oracle-data-guid = 'task guid'</Where>
</vQuery>
```

```
<!-- Ability to fetch multiple tasks -->
<vQuery>
    <Select>*</Select>
    <From>VTOD</From>
    <Where>
        x-oracle-data-guid = 'task id 1' OR
        x-oracle-data-guid = 'task id 2' OR
        x-oracle-data-guid = 'task id 3'
    </Where>
</vQuery>
```

<Where> 句には、文字列 x-oracle-data-guid = が含まれます。右側は iCal のタスク GUID です。

アクティブなタスクを日付範囲に基づいて問合せできます。

```
<!-- Ability to fetch active tasks by time range -->
<vQuery>
    <Select>*</Select>
    <From>VTOD</From>
    <vCall>
        <ActiveTasks>
            <StartTime>20020701T000000Z</StartTime>
            <EndTime>20020801T000000Z</EndTime>
        </ActiveTasks>
    </vCall>
</vQuery>
```

<Select> も使用できますが、現在のバージョンの Oracle Calendar Web Services ではサポートされていません。すべての属性が必要な場合、<Select> は含まれないことがあります。値を「*」にすると、すべてのタスク属性を要求できます。個々の属性のみが必要な場合、明示的にリストする必要があります。

注意： 返されるデータのソート順はありません。

他のタスク問合せ形式を使用すると、エラーが生成されます。

<vCall> 要素は、内部プロシージャ（データベースのストアド・プロシージャなど）が使用されていることを示します。子要素は、起動するストアド・プロシージャの名前と、コールに必要な引数を提供します。

タスク問合せ結果セットは、xCal ドラフト仕様を使用して、<cws1:Reply> タグ内に埋め込まれて返されます。

```
<soap:Body>
  <cws1:Reply xmlns:cws1=
    "http://www.oracle.com/WebServices/Calendaring/1.0/">
    <CmdId>a client id</CmdId>
  <xCal:iCalendar xmlns:xCal=
    "http://www.oracle.com/WebServices/Calendaring/1.0/">
    <vcalendar version="2.0" prodid=...>
      <vtodo>
        <x-oracle-data-guid>JKLFJLJK</x-oracle-data-guid>
        <uid>ffdtasfdtasfdta</uid>
        <dtstamp>19980309T231000Z</dtstamp>
        <summary>My task</summary>
        <priority>2</priority>
      </vevent>
    </vcalendar>
  </xCal:iCalendar>
</cws1:Reply>
</soap:Body>
```

Create

Calendar Server で会議を作成する機能は、SOAP の Create メソッドを通して提供されます。他のメソッドと同様、実装は単純な会議またはメモの作成に限定されます。属性は、class、description、dtstart、duration、location、priority、summary、start time、UID および event-type にかざられています。

```
<!-- Sent to Web Services -->
<Body>
  <Create>
    <CmdId>a command id</CmdId>
    <!-- here is the event select query -->
    <!-- The original event property/values -->
    <iCalendar>
      <vcalendar version="2.0" prodid="any string">
        <vevent>
```

```

        <class>PUBLIC|PRIVATE|CONFIDENTIAL</class>
        <description>a long description</description>
        <dtstart>mmmm</dtstart>
        <duration>nnnnM/ duration >
        <location>my location</location>
        <priority>0|2|5|7|9</priority>
        <summary>a title</summary>
        <uid>a client uid</uid>
        <x-oracle-eventtype>APPOINTMENT|DAY EVENT|
                        DAILY NOTE|HOLIDAY
        </x-oracle-eventtype>
    </vevent>
</vcalendar>
</iCalendar>
</Create>
</Body>
```

指定する属性は次のとおりです。

属性	必須 / オプション	サポート
class	必須	○
description	オプション	○
dtstart	必須	○
dtend	オプション	×
duration	必須	○
location	オプション	○
organizer	オプション	×
priority	オプション	○
summary	オプション	○
uid	オプション	○
attendee	オプション	×
x-oracle-eventtype	必須	○
x-oracle-event-guid	オプション	×
x-oracle-eventinstance-guid	オプション	×
x-oracle-data-guid	オプション	×

```

<!-- Received from Web Services -->
<!-- failure will result in a SOAP fault -->
<Body>
  <CreateReply>
    <CmdId>a command id</CmdId>
    <x-oracle-data-guid>cccc</x-oracle-data-guid>
  </CreateReply>
</Body>

```

応答は、SOAP 障害または作成されたイベントの GUID を含む **CreateReply** です。

このリリースでは、**Create** には次のような問題があります。

- ユーザーの **attendee** ステータスは **ACCEPTED** になります。
- ユーザーのデフォルトのリマインダは適用されません (**vAlarms**)。
- **CLASS** プロパティは必須です。ユーザーのデフォルト・クラス (**normal**、**public**、**confidential**、**private**) のアクセス・レベルを適用して会議を作成する方法はありません。
- 現在、定期的な会議を作成するメカニズムはありません。
- 現在、出席者を追加して会議を作成するメカニズムはありません。

Delete

SOAP の **Delete** メソッドでは、会議、メモまたは終日イベントを削除できます。これには、繰り返し会議および定期的会議のインスタンスを削除する機能も含まれます。

```

<!-- Sent to Web Services -->
<Body>
  <Delete>
    <CmdId>a command id</CmdId>
    <!-- here is the event select query -->
    <vQuery>
      <select>*</select>
      <from>VEVENT</from>
      <where>x-oracle-data-guid = 'some guid'</where>
    </vQuery>
  </Delete>
</Body>

```

vQuery は、前に返されたデータの GUID を一意に識別します。GUID には、繰り返し会議のインスタンスまたは繰り返しルールの反復を一意に識別するためのすべての情報が含まれています。

vQuery は、削除する項目の識別に使用されます。削除できる項目は一度に 1 つだけです。

```

<!--Received from Web Services -->
<!-- failure will result in a SOAP fault -->

```

```

<Body>
  <DeleteReply>
    <CmdId>a command id</CmdId>
    <x-oracle-data-guid>cccc</x-oracle-data-guid>
  </DeleteReply>
</Body>

```

DeleteReply には、削除された項目の GUID が含まれます。

Modify

SOAP の Modify メソッドでは、xCal プロパティおよび属性を変更できます。少数の属性のみを変更できます。Modify メソッドは、問合せ、元の iCalendar オブジェクト・プロパティ、新しい iCalendar オブジェクト・プロパティの 3 つのセクションで構成されます。

```

<!-- Sent to Web Services -->
<Body>
  <Modify>
    <CmdId>a command id</CmdId>
    <!-- here is the event select query -->
    <vQuery>
      ...
    </vQuery>
    <!-- The original event property/values -->
    <iCalendar>
      ...
    </iCalendar>
    <!-- The modified event property/values -->
    <iCalendar>
      ...
    </iCalendar>
  </Modify>
</Body>

```

vQuery は、前に返されたデータの GUID を一意に識別します。GUID には、繰返し会議のインスタンスまたは繰返しルールの反復を一意に識別するためのすべての情報が含まれています。vQuery は、変更する項目の識別に使用されます。変更できる項目は一度に 1 つだけで、これには 1 つの単純なイベント（会議、メモまたは終日イベント）または繰返し会議の 1 つのインスタンスが含まれます。変更操作では、プロパティを変更、追加または削除できます（後述の表を参照）。

Modify の応答は、イベントの GUID を返します。Calendar Server に適用される変更のタイプによって、GUID が変化するため注意が必要です。一部の更新では、削除 / 再作成タイプの対話が必要です。

```

<!--Received from Web Services -->
<!-- failure will result in a SOAP fault -->
<Body>

```



```

    <ModifyReply>
      <CmdId>a command id</CmdId>
      <x-oracle-data-guid>cccc</x-oracle-data-guid>
    </ModifyReply>
  </Body>

```

プロパティの変更

最初の iCalendar オブジェクトには、変更するプロパティと元の値が含まれます。属性が関連付けられている場合は、これらも含まれます。2 番目の iCalendar オブジェクトには、適用する新しいプロパティ値が含まれます。

```

<!-- The original event property/values -->
<iCalendar>
  <vcalendar>
    <vevent>
      <summary>my old title</summary>
      <location>my old location</location>
    </vevent>
  </vcalendar>
</iCalendar>
<!-- The modified event property/values -->
<iCalendar>
  <vcalendar>
    <vevent>
      <summary>my new title</summary>
      <location>my new location</location>
    </vevent>
  </vcalendar>
</iCalendar>

```

プロパティの追加

最初の iCalendar オブジェクトには、追加するプロパティへの参照は含まれません。2 番目の iCalendar オブジェクトには、新しいプロパティと値が含まれます。

```

<!-- The original event property/values -->
<iCalendar>
  <vcalendar>
    <vevent>
      </vevent>
  </vcalendar>
</iCalendar>
<!-- The modified event property/values -->
<iCalendar>
  <vcalendar>
    <vevent>
      <summary>my new title</summary>
    </vevent>
  </vcalendar>
</iCalendar>

```

```
        </vevent>
      </vCalendar>
    </iCalendar>
```

プロパティの削除

最初の iCalendar オブジェクトには、元のプロパティと値が含まれます。2 番目の iCalendar オブジェクトには、プロパティは含まれません。

```
<!-- The original event property/values -->
<iCalendar>
  <vcalendar>
    <vevent>
      <summary>my old title</summary>
    </vevent>
  </vcalendar>
</iCalendar>
<!-- The modified event property/values -->
<iCalendar>
  <vcalendar>
    <vevent>
    </vevent>
  </vcalendar>
</iCalendar>
```

変更可能なプロパティ

プロパティ	変更	追加	削除
class	○	×	×
description	○	○	○
dtstart	○	×	×
dtend	○	×	×
duration	×	×	×
location	○	○	○
organizer	×	×	×
priority	○	×	×
summary	○	○	○
uid	×	×	×
attendee	×	×	×

プロパティ	変更	追加	削除
x-oracle-eventtype	×	×	×
x-oracle-event-guid	×	×	×
x-oracle-eventinstance-guid	×	×	×
x-oracle-data-guid	×	×	×

SOAP の Modify メソッドに他のプロパティがある場合、SOAP 障害が生成されます (class、uid、x-oracle-eventtype、Web Conferencing 属性など)。

Ping

このコマンドは、Oracle Calendar Web Services アプリケーション・サーバーがアクティブかどうかをテストするために使用します。このコマンドでは、それ以外の操作は実行されません。

```
POST <uri> HTTP/1.1
Content-Type: text/xml; charset="UTF-8"
Content-Length: <char length>
SOAPAction: http://www.oracle.com/Webservices/Calendaring/1.0/Ping
```

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <soap:Body>
    <cws1:Ping xmlns:cws1=
      "http://www.oracle.com/Webservices/Calendaring/1.0/">
    </cws1:Ping>
  </soap:Body>
</soap:Envelope>
```

成功したコマンドに対する Web サービスの応答は次のとおりです。

```
HTTP/1.1 200 OK
Content-Type: text/xml
Content-Length: <char length>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <soap:PingReply xmlns:cws1=
    "http://www.oracle.com/Webservices/Calendaring/1.0/">
  </soap:PingReply>
</soap:Envelope>
```

xCal

プロトコルに関する処理を終えた後、データに関する処理を行います。イベントとタスクの場合、xCal ドラフトが SOAP 応答のデータ形式出力の公的な参照として使用されています。

Oracle Calendar Web Services の現在の実装では、繰返し会議と定期的会議全体の取得はサポートされていません。Search を実行すると、サーバーに格納されているインスタンスまたは繰返しルール付きの会議が展開され、各インスタンスは個々の会議に分けられます。これは、処理および UI の生成に役立ちます。繰返しルールの解釈は非常に複雑で、エラーが発生しやすくなります。

Oracle Calendar Web Services で、Calendar Server に格納されている繰返し会議や定期的会議がどのようにサポートされているかを理解するために、いくつかの例を示します。

注意： xCal 仕様では vEvents および vTodos のみがサポートされています。vJournal および vFreebusy はサポートされていません。

このリリースでは、すべての xCal 要素とプロパティおよびすべての Calendar Server 属性がサポートされているわけではありません。

単純なイベント

```
<vcalendar>
  <vevent>
    <class>CONFIDENTIAL</class>
    <description>a description</description>
    <dtend>20021101T120000Z</dtend>
    <dtstart>20021101T110000Z</dtstart>
    <location></location>
    <organizer cn="James Baldwin">
      mailto:james.baldwin@oracle.com
    </organizer>
    <priority>1</priority>
    <status>CONFIRMED</status>
    <summary>a meeting</summary>
    <uid>ORACLE:CALSERV:EVENT:48390483290843290</uid>
    <attendee cn="James Baldwin" partstat="ACCEPTED">
      mailto:james.baldwin@oracle.com
    </attendee>
    <x-oracle-eventtype>APPOINTMENT</x-oracle-eventtype>
    <x-oracle-event-guid>fdjskljfdlkj</x-oracle-event-guid>
    <x-oracle-eventinstance-guid>fdjskljfdlkj
    </x-oracle-eventinstance-guid>
    <x-oracle-data-guid>fdjskljfdlkj</x-oracle-data-guid>
```

```

    </vevent>
  </vcalendar>

```

繰返しイベント

Calendar Server から繰返し会議を直接取り出した場合、1 つの vCalendar 要素に関連する複数の vEvents が含まれています。メインの vEvent には、会議のルールを説明する rrule 要素の後ろにインスタンス、例外、タイムゾーン情報が含まれています。すべてのイベント GUID は vCalendar を通して同じですが、インスタンス GUID は異なります。

次に繰返し会議の例を示します。

```

<vcalendar>
  <vevent>
    <uid>ORACLE:CALSERV:EVENT:6205/391</uid>
    <summary>SDK Development Review</summary>
    <status>CONFIRMED</status>
    <priority>5</priority>
    <organizer cn="Ray Bradbury">mailto:</organizer>
    <location>Salle Saturne (Client/Server)</location>
    <description>
      Review of SDK 9.0.4 and Calendar Web Services
      9.0.4 Development Plans with PM
    </description>
    <dtstart tzid="EST5EDT">20021002T103000</dtstart>
    <dtend tzid="EST5EDT">20021002T120000</dtend>
    <class>PUBLIC</class>
    <rrule>FREQ=DAILY;INTERVAL=1;COUNT=1</rrule>
    <rdate>20021030T153000Z</rdate>
    <attendee cn="James Baldwin"
      partstat="NEEDS-ACTION">mailto:</attendee>
    <attendee partstat="NEEDS-ACTION">
      mailto:Salle Saturne (Client/Server)</attendee>
    <attendee cn="Albert Camus"
      partstat="NEEDS-ACTION">mailto:</attendee>
    <attendee cn="Lewis Carroll"
      partstat="NEEDS-ACTION">mailto:</attendee>
    <attendee cn="Frederic Chopin"
      partstat="ACCEPTED">mailto:</attendee>
    <attendee cn="Emily Dickinson"
      partstat="NEEDS-ACTION">mailto:</attendee>
    <attendee cn="Ray Bradbury"
      partstat="ACCEPTED">mailto:</attendee>
    <attendee cn="Alfred Hitchcock"
      partstat="NEEDS-ACTION">mailto:</attendee>
    <attendee cn="Yasunari Kawabata"
      partstat="ACCEPTED">mailto:</attendee>
  </vevent>
</vcalendar>

```

```
<x-oracle-eventtype>APPOINTMENT</x-oracle-eventtype>
<x-oracle-event-guid>fdjskljfdlkj</x-oracle-event-guid>
</vevent>
<vevent>
  <uid>ORACLE:CALSERV:EVENT:6205/391</uid>
  <summary>SDK Development Review</summary>
  <status>CONFIRMED</status>
  <recurrence-id>20021030T153000Z</recurrence-id>
  <priority>5</priority>
  <organizer cn="Ray Bradbury">mailto:</organizer>
  <location>Salle Saturne (Client/Server)</location>
  <description></description>
  <dtstart>20021030T153000Z</dtstart>
  <dtend>20021030T170000Z</dtend>
  <class>PUBLIC</class>
  <attendee cn="James Baldwin"
    partstat="ACCEPTED">mailto:</attendee>
  <attendee partstat="NEEDS-ACTION">
    mailto:Salle Saturne (Client/Server)</attendee>
  <attendee cn="Albert Camus"
    partstat="ACCEPTED">mailto:</attendee>
  <attendee cn="Lewis Carroll"
    partstat="ACCEPTED">mailto:</attendee>
  <attendee cn="Frederic Chopin"
    partstat="ACCEPTED">mailto:</attendee>
  <attendee cn="Emily Dickinson"
    partstat="ACCEPTED">mailto:</attendee>
  <attendee cn="Ray Bradbury"
    partstat="ACCEPTED">mailto:</attendee>
  <attendee cn="Alfred Hitchcock"
    partstat="ACCEPTED">mailto:</attendee>
  <attendee cn="Yasunari Kawabata"
    partstat="DECLINED">mailto:</attendee>
  <x-oracle-eventtype>APPOINTMENT</x-oracle-eventtype>
  <x-oracle-event-guid>fdjskljfdlkj</x-oracle-event-guid>
  <x-oracle-eventinstance-guid>fdjskljfdlkj
  </x-oracle-eventinstance-guid>
  <x-oracle-data-guid>fdjskljfdlkj</x-oracle-data-guid>
</vevent>
<timezone>
  <tzid>EST5EDT</tzid>
  <daylight>
    <tzoffsetto>-400</tzoffsetto>
    <tzoffsetfrom>-500</tzoffsetfrom>
    <dtstart>19910401T020000</dtstart>
    <rrule>
      FREQ=YEARLY;UNTIL=UNTIL=20750407T020000Z;BYMONTH=4;BYDAY=1SU
```

```

        </rrule>
        <tzname>
            Eastern Standard Time, Eastern Daylight Time
        </tzname>
    </daylight>
    <standard>
        <tzoffsetto>-0500</tzoffsetto>
        <tzoffsettfrom>-0400</tzoffsettfrom>
        <dtstart>19911025T020000</dtstart>
        <rrule>
            FREQ=YEARLY;UNTIL=20751031T020000Z;BYMONTH=10;BYDAY=-1SU
        </rrule>
        <tzname>
            Eastern Standard Time, Eastern Daylight Time
        </tzname>
    </standard>
</timezone>
</vcalendar>

```

定期的なイベント

繰返し会議と同様、繰返しルール付きのイベントでは、vCalendar を使用するデータを解釈する必要があります。ここでも、繰返しルールは、すべての会議固有の情報を解釈または生成するために必要な情報を提供します。これには、クライアント側で複雑な処理が必要です。

Oracle Connector for Outlook は、純粋な繰返しルールを持つイベントを作成できます。

```

<vcalendar>
  <vevent>
    <uid>ORACLE:CALSERV:EVENT:10750/391</uid>
    <summary>test recur weekly 3 times</summary>
    <status>TENTATIVE</status>
    <priority>5</priority>
    <organizer cn="Stephen King">mailto:</organizer>
    <location></location>
    <description></description>
    <dtstart tzid="EST5EDT">20021025T123000</dtstart>
    <dtend tzid="EST5EDT">20021025T133000</dtend>
    <created>20021101T191154Z</created>
    <class>PUBLIC</class>
    <rrule>
        FREQ=WEEKLY;BYDAY=FR;COUNT=3;INTERVAL=1;WKST=SU
    </rrule>
    <rdate>20021101T173000Z</rdate>
    <attendee cn="Stephen King"
        partstat="ACCEPTED">mailto:</attendee>

```

```
<x-oracle-event-guid>fdjskljfdlkj</x-oracle-event-guid>
<x-oracle-eventtype>APPOINTMENT</x-oracle-eventtype>
</vevent>
<vevent>
  <uid>ORACLE:CALSERV:EVENT:10750/391</uid>
  <summary>test recur weekly 3 times</summary>
  <status>CONFIRMED</status>
  <recurrence-id>20021101T173000Z</recurrence-id>
  <priority>5</priority>
  <organizer cn="Stephen King">mailto:</organizer>
  <location></location>
  <description></description>
  <dtstart>20021101T173000Z</dtstart>
  <dtend>20021101T183000Z</dtend>
  <created>20021101T191154Z</created>
  <class>PUBLIC</class>
  <attendee cn="Stephen King"
partstat="ACCEPTED">mailto:</attendee>
  <x-oracle-event-guid>fdjskljfdlkj</x-oracle-event-guid>
  <x-oracle-eventtype>APPOINTMENT</x-oracle-eventtype>
  <x-oracle-eventinstance-guid>fdjskljfdlkj
</x-oracle-eventinstance-guid>
  <x-oracle-data-guid>fdjskljfdlkj</x-oracle-data-guid>
</vevent>
<vtimezone>
  <tzid>EST5EDT</tzid>
  <daylight>
    <tzoffsetto>-0400</tzoffsetto>
    <tzoffsettfrom>-0500</tzoffsettfrom>
    <dtstart>19910401T020000</dtstart>
    <rrule>
      FREQ=YEARLY;UNTIL=20750407T020000Z;BYMONTH=4;BYDAY=1SU
    </rrule>
    <tzname>
      Eastern Standard Time, Eastern Daylight Time
    </tzname>
  </daylight>
  <standard>
    <tzoffsetto>-0500</tzoffsetto>
    <tzoffsettfrom>-0400</tzoffsettfrom>
    <dtstart>19911025T020000</dtstart>
    <rrule>
      FREQ=YEARLY;UNTIL=20751031T020000Z;BYMONTH=10;BYDAY=-1SU
    </rrule>
    <tzname>
      Eastern Standard Time, Eastern Daylight Time
    </tzname>
  </standard>
</vtimezone>
```



```

    </standard>
  </vtimezone>
</vcalendar>

```

展開されたフィルタ済の定期的 / 繰返しイベント

Oracle Calendar Web Services は繰返しルールを、分解（展開）された個々の要素として提供します。

展開された繰返し会議および定期的会議の形式は次のとおりです。

```

<vcalendar>
  <vevent>
    <uid>ORACLE:CALSERV:EVENT:6205/391@20021030T153000Z</uid>
    <summary>SDK Development Review</summary>
    <status>CONFIRMED</status>
    <priority>5</priority>
    <organizer cn="Ray Bradbury">mailto:</organizer>
    <location>Salle Saturne (Client/Server)</location>
    <description></description>
    <dtstart>20021030T153000Z</dtstart>
    <dtend>20021030T170000Z</dtend>
    <class>PUBLIC</class>
    <attendee cn="James Baldwin"
      partstat="ACCEPTED">mailto:</attendee>
    <attendee partstat="NEEDS-ACTION">
      mailto:Salle Saturne (Client/Server)</attendee>
    <attendee cn="Albert Camus"
      partstat="ACCEPTED">mailto:</attendee>
    <attendee cn="Lewis Carroll"
      partstat="ACCEPTED">mailto:</attendee>
    <attendee cn="Frederic Chopin"
      partstat="ACCEPTED">mailto:</attendee>
    <attendee cn="Emily Dickinson"
      partstat="ACCEPTED">mailto:</attendee>
    <attendee cn="Ray Bradbury"
      partstat="ACCEPTED">mailto:</attendee>
    <attendee cn="Alfred Hitchcock"
      partstat="ACCEPTED">mailto:</attendee>
    <attendee cn="Yasunari Kawabata"
      partstat="DECLINED">mailto:</attendee>
    <x-oracle-event-guid>fdjfkfsjfljkds</x-oracle-event-guid>
    <x-oracle-eventtype>APPOINTMENT</x-oracle-eventtype>
    <x-oracle-eventinstance-guid>fdjskljfdlkj
    </x-oracle-eventinstance-guid>
    <x-oracle-data-guid>fdjskljfdlkj</x-oracle-data-guid>
  </vevent>
</vcalendar>

```

```
</vevent>
</vcalendar>
```

メモ

```
<vcalendar>
  <vevent>
    <class>CONFIDENTIAL</class>
    <description>a description</description>
    <dtend value="DATE">20021101</dtend>
    <dtstart value="DATE">20021101</dtstart>
    <organizer cn="Par Lagerkvist">
      mailto:par.lagerkvist@oracle.com
    </organizer>
    <priority>3</priority>
    <status>CONFIRMED</status>
    <summary>a daily note</summary>
    <uid>ORACLE:CALSERV:EVENT:49304932-04932-09</uid>
    <attendee cn="Par Lagerkvist" partstat="ACCEPTED">
      mailto:par.lagerkvist@oracle.com
    </attendee>
    <x-oracle-event-guid>fdjskljfdlkj</x-oracle-event-guid>
    <x-oracle-eventtype>DAILY NOTE</x-oracle-eventtype>
    <x-oracle-eventinstance-guid>fdjskljfdlkj
    </x-oracle-eventinstance-guid>
    <x-oracle-data-guid>fdjskljfdlkj</x-oracle-data-guid>
  </vevent>
</vcalendar>
```

終日イベント

```
<vcalendar>
  <vevent>
    <class>CONFIDENTIAL</class>
    <description>a description</description>
    <dtend value="DATE">20021101</dtend>
    <dtstart value="DATE">20021101</dtstart>
    <organizer cn="Par Lagerkvist">
      mailto:par.lagerkvist@oracle.com
    </organizer>
    <priority>3</priority>
    <status>CONFIRMED</status>
    <summary>a day event</summary>
    <uid>ORACLE:CALSERV:EVENT:49304932-04932-09</uid>
    <attendee cn="Par Lagerkvist" partstat="ACCEPTED">
      mailto:par.lagerkvist@oracle.com
    </attendee>
  </vevent>
</vcalendar>
```

```

        </attendee>
        <x-oracle-event-guid>fdjskljfdlkj</x-oracle-event-guid>
        <x-oracle-eventtype>DAY EVENT</x-oracle-eventtype>
        <x-oracle-eventinstance-guid>fdjskljfdlkj
        </x-oracle-eventinstance-guid>
        <x-oracle-data-guid>fdjskljfdlkj</x-oracle-data-guid>
    </vevent>
</vcalendar>

```

休日

休日は次のように表されます。

```

<vcalendar>
  <vevent>
    <categories>
      <item>Holiday</item>
    </categories>
    <class>PUBLIC</class>
    <description>a description</description>
    <dtend value="DATE">20021031</dtend>
    <dtstart value="DATE">20021031</dtstart>
    <organizer cn="Par Lagerkvist">
      mailto:par.lagerkvist@oracle.com
    </organizer>
    <priority>3</priority>
    <status>CONFIRMED</status>
    <summary>a holiday</summary>
    <uid>ORACLE:CALSERV:EVENT:49304932-04932-09</uid>
    <attendee cn="Par Lagerkvist" partstat="ACCEPTED">
      mailto:par.lagerkvist@oracle.com
    </attendee>
    <x-oracle-event-guid>fdjskljfdlkj</x-oracle-event-guid>
    <x-oracle-eventtype>HOLIDAY</x-oracle-eventtype>
    <x-oracle-eventinstance-guid>fdjskljfdlkj
    </x-oracle-eventinstance-guid>
    <x-oracle-data-guid>fdjskljfdlkj</x-oracle-data-guid>
  </vevent>
</vcalendar>

```

タスク

```
<vcalendar>
  <vtodo>
    <class>PRIVATE</class>
    <completed>20021002T210000Z</completed>
    <created>20021002T210000Z</created>
    <description>the task description</description>
    <percent>0</percent>
    <priority>9</priority>
    <summary>The task title</summary>
    <uid>fdjskljfdlkj</uid>
    <due>20021102T210000Z</due>
    <x-oracle-data-guid>
      ORACLE:CALSERV:TASK:328321890328/489043209
    </x-oracle-data-guid>
  </vtodo>
</vcalendar>
```

X-ORACLE-EVENTTYPE

この新しいプロパティは、Calendar Server に格納されているイベントのタイプを識別するために使用されます。この属性は、<select> 句で選択可能で、vEvent に要素として返されます。使用可能な値は次のとおりです。

- APPOINTMENT: ブロックされている標準の会議を識別します。
- DAILY NOTE: カレンダの日付に関連付けられているブロックされていないメモを識別します。
- DAY EVENT: ブロックされていないすべての終日カレンダー・イベントを識別します。
- HOLIDAY: 終日イベントのブロックされていない特殊な休日を識別します。

優先度のマッピング

xCal では、カレンダー・クライアント製品との互換性を最大限に得るために、優先度が特定の Calendar Server レベルにマップされます。

- 0 は HIGHEST にマップされます。
- 1 は HIGH にマップされます。
- 5 は NORMAL にマップされます。
- 7 は LOW にマップされます。
- 9 は LOWEST にマップされます。

Oracle Calendar Web Services の クライアント側 Java 実装

この章では、Oracle Calendar Web Services を使用してカレンダー・データにアクセスし、コンテキストに基づく協同作業を実現するための Java クラス・セットの設計について説明します。これらのカレンダーレット・クラスは、Java 環境で Oracle Calendar Web Services テクノロジを使用する際の多くの詳細を隠します。

クラスの実装は、すべての iCalendar プロパティや属性を提供するものではなく、Oracle Calendar Web Services ツールキットの V2 実装でサポートされている要素が中心になります。

注意： Oracle Calendar Web Services ツールキットには、JavaDoc 情報とテスト・ツールのサンプルが含まれています。

Java クラス

カレンダーレット・クラスを使用するには、いくつかの一般的な手順に従う必要があります。

- 認証メカニズムを初期化します。
- データ型を含む問合せを初期化します。
- 認証および問合せオブジェクトをカレンダーレット・インスタンスにバインドします。
- カレンダーレット・インスタンスでターゲット URL を設定します。
- SOAP コールを実行します。
- 結果を解析します。

カレンダーレット・クラスの実装では、プロトコル・レベルの処理の多くを Apache SOAP クラスに大きく依存しています。着信および送信メッセージについては、これらの同じ Apache SOAP クラスが W3C DOM クラスとともに使用されます。送信メッセージを生成する場合、カレンダーレット・クラスと iCalendar クラスがインスタンス化され、親クラスで設定されます。最終的な XML ストリームの生成では、すべてのクラスが `getElement()` メソッドを実装します。これは、送信する SOAP メッセージの XML DOM 表現を構築するためです。下位レベルの Apache SOAP コールでは、最終的なストリームを取得するにはこの DOM 構造が必要です。

着信メッセージの場合、各クラスで `static` メソッド `unmarshall()` を使用してカレンダーレット・クラスと iCalendar クラスが再構築され、ここでも、下位レベルの Apache SOAP クラスから受け取る XML DOM を使用します。この DOM のアンマーシャリングは、子タグを認識し、その子タグのクラスの `unmarshall()` メソッドを起動する親クラスで構成されます。

なんらかの理由で、XML 解析エラーが発生した場合、下位レベルの Apache SOAP 例外が発生します。カレンダーレット・クラスではデータを解析することはできません。XML は有効でも、要素が間違った場所で使用されたり、認識されないなどのコンテキスト・エラーがある場合、カレンダーレット例外が発生します。

すべての着信 xCal (iCalendar の XML バインディング) では、データ内の要素が拡張されていることが理想です。ただし、この実装では、拡張された要素は vEvent レベルでのみ処理されます。

カレンダーレット・クラスはある程度のデバッグ・サポートを提供します。これらには主に 2 つの特徴があります。

- 入力および出力バッファのキャプチャ機能。SOAP メソッドを起動する前に `setWantIOBuffers()` メソッドをコールする必要があります (最終デプロイにはお薦めしません)。入力バッファと出力バッファの両方が取得され、CalendarlingResponse クラスに格納されます。
- SOAP 要求の総処理時間 (ミリ秒) を取得する機能。CalendarletResponse に格納されます。

データのフェッチ

Java のカレンダーレット・クラスと次のコードを使用して、SOAP 要求を生成できます。

```
// initialize the authentication information
// and set the user id
BasicAuth auth = new BasicAuth();

auth.setName("*** userid ***");
auth.setPassword("*** password ***");

// initialize the event search command and query
SearchCommand search = new SearchCommand();

search.setCmdId("my cmd id 1");

// create a query to retrieve unconfirmed events
vQuery          query          = new vQuery();
query.setFrom(vQuery.k_queryFromEvent);

// determine the datestamps for a weeks worth of events

// use the CalendarUtils to get a proper timestamp with
// time zone information set properly
Calendar today = CalendarUtils.getToday();

int dayOfWeek = today.get(Calendar.DAY_OF_WEEK);

Calendar beginWeek = (Calendar)today.clone();
Calendar endWeek   = (Calendar)today.clone();

beginWeek.add(Calendar.DATE, 1 - dayOfWeek);
endWeek.add(Calendar.DATE, 8 - dayOfWeek);
endWeek.add(Calendar.MINUTE, -1);

// use the CalendarUtils to help generate a date range query
query.setWhere(CalendarUtils.getDateRangeQuery(beginWeek, endWeek));

search.setQuery(query);

// create the calendar client SOAP stub
// and set the basic authentication header
Calendarlet cws = new Calendarlet();
```

```
// set the Web Services host URL
cws.setEndPointURL("http://www.thehost.com");

cws.setAuthenticationHeader(auth.getElement());

// make the SOAP call
CalendaringResponse response = cws.Search(search.getElement());
```

結果の応答には `iCalendar` オブジェクトが含まれ、要求されたプロパティが返されます。
Java のカレンダーレット・クラスを使用して `iCalendar` を横断し、`vEvent` を検索できます。

```
// get the SOAP reply
Reply reply = (Reply)response.getCalendarReply();

// traverse all the iCalendar objects
Vector someiCalendars = iCalendar.unmarshallVector(
    reply.getEntries());
int    numiCalendars = someiCalendars.size();

for (int i = 0; i < numiCalendars; i++)
{
    iCalendar iCalObj = (iCalendar)someiCalendars.get(i);

    // traverse all the vCalendar objects
    Vector somevCalendars = iCalObj.getvCalendars();
    int    numvCalendars = somevCalendars.size();

    for (int j = 0; j < numvCalendars; j++)
    {
        vCalendar vCalObj = (vCalendar)somevCalendars.get(j);

        // traverse all the vEvent objects
        Vector somevEvents = vCalObj.getComponents();
        int    numvEvents = somevEvents.size();

        for (int k = 0; k < numvEvents; k++)
        {
            vEvent vEventObj = (vEvent)somevEvents.get(i);

            // get the specific properties
            String title      = vEventObj.getSummary();
            String dtstart    = vEventObj.getDtStart();
            String dtend      = vEventObj.getDtEnd();
            String eventType  = vEventObj.getXEventType();
```



```

        // do something interesting with the meeting info
    }
}

```

SOAP 障害および例外

Oracle Calendar Web Services ツールキットでは、Java 例外または SOAP 障害という 2 種類のエラーが発生する可能性があります。

Java 例外は、カレンダーレット・クラス、基となる Apache SOAP または W3C DOM クラス、あるいは Java Runtime から発生します。カレンダーレット・クラスで起動される各 SOAP メソッドに対して、内部処理エラーまたは XML 解析の問題の結果として例外が発生することがあります。これらは通常クライアント側の予期しない例外で、適切に処理する必要があります。

SOAP 障害は、Oracle Calendar Web Services のエラーの結果です（リモート・サーバー側のエラー）。サーバー側のエラーが発生した場合は常に、Web サービスから HTTP トランザクションへの応答として SOAP 障害が返されます。Java ベースの例外は発生しません。SOAP 障害内の詳細フィールドに、Oracle Calendar Web Services のエラー・オブジェクトと重要なエラー・コードが含まれている場合があります。エラー・コードの一覧は、[第 11 章「Oracle Calendar Web Services のステータス・コード」](#)を参照してください。

CalendarUtils メソッドは、SOAP 障害が発生したかどうかを判別する際、および Oracle Calendar Web Services エラーを取得する際に役立ちます。

```

// Calendar response
CalendarResponse response = cws.Search(...);

// get the vector of entries embedded
// in the SOAP body
Vector bodyEntries = response.getBodyEntries();

// determine if there was a SOAP Fault
if (!CalendarUtils.isSOAPFault(bodyEntries))
{
    // do regular processing
}
else
{
    // get the SOAP fault object
    org.apache.soap.Fault soapFault =
        CalendarUtils.getSOAPFault(bodyEntries);

    // get the calendar Web Services error
    Error calendaringError = Error.unmarshall(soapFault);
}

```

```
// get the Web Services error code
String errorCode = calendaringError.getCode();
}
```

ローカル時間

Date および DateTime という 2 つの重要な日付の書式に注意する必要があります。DateTime 書式には、文字列内に日付と時間の両方が含まれますが、Date には時間情報のみが含まれます。通常、DateTime は標準の会議に、Date は終日イベント、メモおよび休日 に使用されます。

vQuery の UTC 日時文字列を生成しやすくするために、CalendarUtils には標準の Java Calendar クラス・オブジェクトを取り、yyyyMMddThhmmssZ 形式の適切な文字列を生成するクラスが含まれます。Java の Calendar クラスには、Java タイムゾーンを関連付けることができます。適切なタイムゾーンを決定し、Java の Calendar クラスで設定するのは、カレンダーレット・クラスのユーザーが行います。

```
// set the date through some mechanism
// ensure the proper time zone is set
TimeZone localTimezone = TimeZone.getDefault();
Calendar theDate        = Calendar.getInstance(localTimezone);

String utcString = CalendarUtils.getUTCDateTime(theDate);
```

vQuery の UTC 日付文字列を生成しやすくするために、CalendarUtils には標準の Java Calendar クラス・オブジェクトを取り、yyyyMMdd 形式の適切な文字列を生成するクラスが含まれます。Java の Calendar クラスには、Java タイムゾーンを関連付けることができます。適切なタイムゾーンを決定し、Java の Calendar クラスで設定するのは、カレンダーレット・クラスのユーザーが行います。

```
// set the date through some mechanism
// ensure the proper time zone is set
TimeZone localTimezone = TimeZone.getDefault();
Calendar theDate        = Calendar.getInstance(localTimezone);

String utcString = CalendarUtils.getUTCDate(theDate);
```

多くのカレンダー問合せ操作は、現在の日付を基準としているため、vQuery 日時スタンプをベースにするための CalendarUtils メソッドも提供されています。このメソッドは、ローカル時間で今日の真夜中の日時スタンプを返します。形式は yyyyMMddTxxxx00Z で、xxxx は UTC からの時間と分のオフセットです（一部のタイムゾーンのオフセットは 30 分）。

```
String utcToday = CalendarUtils.getToday();
```

一般的な Calendar Server の問合せ時間範囲は、特定の日付の真夜中（ローカル時間）から最終日付の前日の真夜中 1 分前までです。たとえば、今日が EST 時間で 2003 年 6 月 1 日の

場合、`getToday()` メソッドは `20030602T040000Z` を返します。日付範囲が 1 日の場合、最終日付は `20030603T035900Z` となります。

Oracle Calendar Web Services の ステータス・コード

SOAP 障害の各ステータス・コードは、4つのセグメントで構成されています。最初のセグメントはソース・モジュールを説明し、最後のセグメントはエラー・タイプを説明します。(現時点では、2番目と3番目のセグメントは通常使用しません)。この章では、SOAP 障害に表示されるモジュール・コードとエラー・コードを一覧します。

次に示すのは、Code タグのサンプルです。

モジュール (未使用)

000C-01-00-00000029

サブモジュール エラー

一般に、ユーザーに関連があるのは最初と最後のセグメントだけです。上の例の場合、この章に掲載した表から次のように解釈できます。

- モジュール 000C = SYS_MODULE_DATAACCESS
- エラー 00000029 = e_soapSOAPRequestCode_MissingModifyCmd

SOAP 障害の操作の詳細は、[第9章「Oracle Calendar Web Services SOAP」](#)を参照してください。

モジュール・コード

これら 64 ビットの各コードは、Oracle Calendar のエラーのソース・モジュールに対応しています。各モジュール名の前には、"SYS_MODULE_" が付きます。

コード	モジュール	説明
0x0000	NONE	N/A
0x0001	UNIAPI	Calendar Server
0x0002	APPLICATION	カレンダ・アプリケーション
0x0003	MEMORYMGR	メモリー・マネージャ
0x0004	CONNECTION	接続サービス
0x0005	DISPATCH	ディスパッチ・サービス
0x0006	LINKDB	リンク・データベース・サービス
0x0007	MESSAGECAT	メッセージ・カタログ・サービス
0x0008	PREFERENCE	作業環境サービス
0x0009	REGISTRY	レジストリ・サービス
0x000A	SESSIONDB	セッション・データベース・サービス
0x000B	SYSTEM	システム・サービス
0x000C	DATAACCESS	データ・アクセス・サービス
0x000D	DATAMANAGER	データ・マネージャ・サービス
0x000E	SYNC	同期サービス
0x000F	SYNCML	SyncML モジュール
0x0010	MALCLIENT	MAL モジュール
0x0011	NLSSERVICE	NLS サービス
0x0012	MALSYSTEM	MAL システム・サービス
0x0013	PLUGINCONFIG	プラグイン（コンポーネント）サービス
0x0014	MOBILE	モバイル・モジュール
0x0015	XMLSERVICE	SOAP モジュール
0x0016	WINDOWS	Windows エラー
0x0017	FCGI	FCGI ツールキット・エラー

エラー・コード

次の表に、0015 の SOAP モジュールで生成されるエラー・コードをリストします。

コード	エラー
00000001	Ok = 1
00000002	NoInputData
00000003	NoSoapAction
00000004	NoPost
00000005	NoUTF8
00000006	MethodNotSupported
00000007	NoSoapContent
00000008	UnexpectedBasicAuthFailure
00000009	WrongSoapMethod
0000000A	SOAPActionMismatch
0000000B	SoapActionNamespace
0000000C	VersionMismatch
0000000D	BasicAuthNamespace
0000000E	Unused001
0000000F	SearchGenerationError
00000010	UnexpectedParserError
00000011	SAX2FatalError
00000012	SAX2Error
00000013	SAX2Warning
00000014	SAXException
00000015	XMLException
00000016	PingGenerationError
00000017	SecurityError
00000018	Unused002
00000019	Unused003
0000001A	Unused004

コード	エラー
0000001B	Unused005
0000001C	Unused006
0000001D	SearchNamespace
0000001E	MissingSearchCmd
0000001F	BadSearchFrom
00000020	BasicAuthMissingName
00000021	BasicAuthMissingPassword
00000022	MissingQuery
00000023	Unexpected
00000024	CreateGenerationError
00000025	CreateNamespace
00000026	MissingCreateCmd
00000027	ModifyGenerationError
00000028	InvalidModifyNamespace
00000029	MissingModifyCmd
0000002A	MissingModifyQuery
0000002B	InvalidModifyQueryFrom
0000002C	MissingModifyOriginalElement
0000002D	MissingModifyModifiedElement
0000002E	DeleteGenerationError
0000002F	DeleteNamespace
00000030	MissingDeleteCmd
00000031	MissingDeleteQuery
00000032	InvalidDeleteQueryFrom
00000033	MissingCreateElement
00000034	InvalidCreateElement
00000035	NoopGenerationError
00000036	InvalidAuthentication
00000037	InvalidTrustedAuthNamespace

コード	エラー
00000038	TrustedAuthMissingName
00000039	TrustedAuthMissingToken
0000003A	UnexpectedTrustedAuthFailure
0000003B	InvalidProxyAuthNamespace
0000003C	ProxyAuthMissingName
0000003D	ProxyAuthMissingAppName
0000003E	ProxyAuthMissingAppPassword
0000003F	UnexpectedProxyAuthFailure
00000040	UnableToLocateEvent
00000041	MoreThanOneEvent
00000042	MissingEventType
00000043	MissingClass
00000044	MissingDtStart
00000045	MissingDuration
00000046	UnableToConvertxCal
00000047	InvalidClass
00000048	MissingUID
00000049	UnsupportedCreateEventType
0000004A	InvalidModifyComponent
0000004B	ModifyFoundDataGuidProperty
0000004C	BadPriority
0000004D	QuietLoginGenerationError
0000004E	QuietLoginNamespace
0000004F	ModifyClassNotSupported
00000050	ModifyDtStartNotSupported
00000051	ModifyDtEndNotSupported
00000052	ModifyDurationNotSupported
00000053	ModifyEventTypeNotSupported
00000054	ModifyPriorityNotSupported

コード	エラー
00000055	ModifyDataGuidNotSupported
00000056	ModifyUIdNotSupported
00000057	UnsupportedDtEnd
00000058	UnsupportedDataGuid
00000059	UnsupportedOrganizer
0000005A	UnsupportedAttendee
0000005B	ModifyOrganizerNotSupported
0000005C	ModifyAttendeeNotSupported
0000005D	SearchScore
0000005E	ModifySearchScoreNotSupported

B

blocking, 4-2

C

CAPI_AuthenticateAsSysop, 3-69
CAPI_FLAG_STORE_DELPROPS, 5-2
CAPI_FLAG_STORE_MODPROPS, 5-2
CAPI_STAT_API, 6-2
CAPI_STAT_API_BADPARAM, 6-2
CAPI_STAT_API_CALLBACK, 6-2
CAPI_STAT_API_CALLBACK_ERROR, 6-2
CAPI_STAT_API_FLAGS, 6-2
CAPI_STAT_API_HANDLE, 6-2
CAPI_STAT_API_HANDLE_BAD, 6-2
CAPI_STAT_API_HANDLE_NOTNULL, 6-2
CAPI_STAT_API_HANDLE_NULL, 6-2
CAPI_STAT_API_NULL, 6-2
CAPI_STAT_API_SESSION, 6-2
CAPI_STAT_API_SESSION_BAD, 6-2
CAPI_STAT_API_SESSION_NOTNULL, 6-3
CAPI_STAT_API_SESSION_NULL, 6-3
CAPI_STAT_API_STREAM, 6-3
CAPI_STAT_API_STREAM_BAD, 6-3
CAPI_STAT_API_STREAM_NOTNULL, 6-3
CAPI_STAT_API_STREAM_NULL, 6-3
CAPI_STAT_DATA, 6-3
CAPI_STAT_DATA_COOKIE, 6-3
CAPI_STAT_DATA_DATE, 6-3
CAPI_STAT_DATA_DATE_FORMAT, 6-3
CAPI_STAT_DATA_DATE_INVALID, 6-3
CAPI_STAT_DATA_DATE_OUTOFRANGE, 6-3
CAPI_STAT_DATA_DATE_RANGE, 6-4
CAPI_STAT_DATA_EMAIL, 6-4

CAPI_STAT_DATA_EMAIL_NOTSET, 6-4
CAPI_STAT_DATA_ENCODING, 6-4
CAPI_STAT_DATA_HOSTNAME, 6-4
CAPI_STAT_DATA_HOSTNAME_FORMAT, 6-4
CAPI_STAT_DATA_HOSTNAME_HOST, 6-4
CAPI_STAT_DATA_HOSTNAME_SERVER, 6-4
CAPI_STAT_DATA_ICAL, 6-4
CAPI_STAT_DATA_ICAL_COMPEXTRA, 6-4
CAPI_STAT_DATA_ICAL_COMPMISSING, 6-4
CAPI_STAT_DATA_ICAL_COMPNAME, 6-4
CAPI_STAT_DATA_ICAL_COMPVALUE, 6-5
CAPI_STAT_DATA_ICAL_FOLDING, 6-5
CAPI_STAT_DATA_ICAL_IMPLEMENT, 6-5
CAPI_STAT_DATA_ICAL_LINEOVERFLOW, 6-5
CAPI_STAT_DATA_ICAL_NONE, 6-5
CAPI_STAT_DATA_ICAL_OVERFLOW, 6-5
CAPI_STAT_DATA_ICAL_PARAMEXTRA, 6-5
CAPI_STAT_DATA_ICAL_PARAMMISSING, 6-5
CAPI_STAT_DATA_ICAL_PARAMNAME, 6-5
CAPI_STAT_DATA_ICAL_PARAMVALUE, 6-5
CAPI_STAT_DATA_ICAL_PROPEXTRA, 6-5
CAPI_STAT_DATA_ICAL_PROPMISSING, 6-6
CAPI_STAT_DATA_ICAL_PROPNAME, 6-6
CAPI_STAT_DATA_ICAL_PROPVALUE, 6-6
CAPI_STAT_DATA_ICAL_RECURMODE, 6-6
CAPI_STAT_DATA_MIME, 6-6
CAPI_STAT_DATA_MIME_CHARSET, 6-6
CAPI_STAT_DATA_MIME_COMMENT, 6-6
CAPI_STAT_DATA_MIME_ENCODING, 6-6
CAPI_STAT_DATA_MIME_HEADER, 6-6
CAPI_STAT_DATA_MIME_IMPLEMENT, 6-6
CAPI_STAT_DATA_MIME_IMPLEMENT_NESTING, 6-6
CAPI_STAT_DATA_MIME_LENGTH, 6-6
CAPI_STAT_DATA_MIME_NOICAL, 6-7

CAPI_STAT_DATA_MIME_NONE, 6-7
 CAPI_STAT_DATA_MIME_OVERFLOW, 6-7
 CAPI_STAT_DATA_UID, 6-7
 CAPI_STAT_DATA_UID_FORMAT, 6-7
 CAPI_STAT_DATA_UID_NOTFOUND, 6-7
 CAPI_STAT_DATA_UID_RECURRENCE, 6-7
 CAPI_STAT_DATA_USERID, 6-7
 CAPI_STAT_DATA_USERID_EXT, 6-7
 CAPI_STAT_DATA_USERID_EXT_CONFLICT, 6-7
 CAPI_STAT_DATA_USERID_EXT_FORMAT, 6-7
 CAPI_STAT_DATA_USERID_EXT_INIFILE, 6-7
 CAPI_STAT_DATA_USERID_EXT_MANY, 6-8
 CAPI_STAT_DATA_USERID_EXT_NODE, 6-8
 CAPI_STAT_DATA_USERID_FORMAT, 6-8
 CAPI_STAT_DATA_USERID_ID, 6-8
 CAPI_STAT_LIBRARY, 6-8
 CAPI_STAT_LIBRARY_IMPLEMENTATION, 6-8
 CAPI_STAT_LIBRARY_INTERNAL, 6-8
 CAPI_STAT_LIBRARY_INTERNAL_COSMICRAY, 6-8
 CAPI_STAT_LIBRARY_INTERNAL_DATA, 6-8
 CAPI_STAT_LIBRARY_INTERNAL_EXPIRY, 6-8
 CAPI_STAT_LIBRARY_INTERNAL_FUNCTION, 6-8
 CAPI_STAT_LIBRARY_INTERNAL_OVERFLOW, 6-9
 CAPI_STAT_LIBRARY_INTERNAL_PROTOCOL, 6-9
 CAPI_STAT_LIBRARY_INTERNAL_UNKNOWN_EXCEPTION, 6-9
 CAPI_STAT_LIBRARY_INTERNAL_UNKNOWN_LIBRARY_ERRCODE, 6-9
 CAPI_STAT_LIBRARY_SERVER, 6-9
 CAPI_STAT_LIBRARY_SERVER_BUSY, 6-9
 CAPI_STAT_LIBRARY_SERVER_SUPPORT, 6-9
 CAPI_STAT_LIBRARY_SERVER_SUPPORT_CHARSET, 6-9
 CAPI_STAT_LIBRARY_SERVER_SUPPORT_STANDARDS, 6-9
 CAPI_STAT_LIBRARY_SERVER_USERDATA, 6-9
 CAPI_STAT_OK, 6-9
 CAPI_STAT_SECUR, 6-10
 CAPI_STAT_SECUR_LOGON, 6-10
 CAPI_STAT_SECUR_LOGON_AUTH, 6-10
 CAPI_STAT_SECUR_LOGON_LOCKED, 6-10
 CAPI_STAT_SECUR_LOGON_LOCKED_RESOURCE, 6-10
 CAPI_STAT_SECUR_LOGON_LOCKED_SYSOP, 6-10
 CAPI_STAT_SECUR_READ, 6-10
 CAPI_STAT_SECUR_READ_ALARM, 6-10
 CAPI_STAT_SECUR_READ_PROPS, 6-10
 CAPI_STAT_SECUR_SERVER, 6-10
 CAPI_STAT_SECUR_SERVER_LICENSE, 6-10
 CAPI_STAT_SECUR_SERVER_SET_IDENTITY_SYSOP, 6-10
 CAPI_STAT_SECUR_WRITE, 6-11
 CAPI_STAT_SECUR_WRITE_AGENDA, 6-11
 CAPI_STAT_SECUR_WRITE_EVENT, 6-11
 CAPI_STAT_SERVICE, 6-11
 CAPI_STAT_SERVICE_ACE, 6-11
 CAPI_STAT_SERVICE_ACE_LOAD, 6-11
 CAPI_STAT_SERVICE_ACE_SUPPORT, 6-11
 CAPI_STAT_SERVICE_FILE, 6-11
 CAPI_STAT_SERVICE_FILE_CLOSE, 6-11
 CAPI_STAT_SERVICE_FILE_DELETE, 6-11
 CAPI_STAT_SERVICE_FILE_MODE, 6-11
 CAPI_STAT_SERVICE_FILE_OPEN, 6-11
 CAPI_STAT_SERVICE_FILE_READ, 6-12
 CAPI_STAT_SERVICE_FILE_TEMP, 6-12
 CAPI_STAT_SERVICE_FILE_WRITE, 6-12
 CAPI_STAT_SERVICE_LIBRARY, 6-12
 CAPI_STAT_SERVICE_MEM, 6-12
 CAPI_STAT_SERVICE_MEM_ALLOC, 6-12
 CAPI_STAT_SERVICE_NET, 6-12
 CAPI_STAT_SERVICE_NET_TIMEOUT, 6-12
 CAPI_STAT_SERVICE_THREAD, 6-12
 CAPI_STAT_SERVICE_TIME, 6-12
 CAPI_STAT_SERVICE_TIME_GMTIME, 6-12
 client_name, 4-2
 client_version, 4-3
 cncachesize, 4-3
 Create メソッド, 9-15
 CSDK_AddConditionToQuery, 3-52
 CSDK_Authenticate, 3-6
 CSDK_ConfigureACE, 3-9
 CSDK_Connect, 3-10
 CSDK_ConnectAsSysop, 3-11
 CSDK_CreateCallbackStream, 3-19
 CSDK_CreateFileStream, 3-67
 CSDK_CreateFileStreamFromFilenames, 3-21
 CSDK_CreateMemoryStream, 3-23
 CSDK_CreateQuery, 3-53
 CSDK_CreateSession, 3-5
 CSDK_Deauthenticate, 3-12
 CSDK_DeleteContacts, 3-27
 CSDK_DeleteEvents, 3-33
 CSDK_DeleteTasks, 3-46

CSDK_DestroyHandle, 3-14
CSDK_DestroyMultipleHandles, 3-15
CSDK_DestroyMultipleStreams, 3-26
CSDK_DestroyQuery, 3-54
CSDK_DestroyResult, 3-54
CSDK_DestroySession, 3-5
CSDK_DestroyStream, 3-26
CSDK_Disconnect, 3-12
CSDK_FetchContactsByQuery, 3-28
CSDK_FetchContactsByUID, 3-29
CSDK_FetchEventsByAlarmRange, 3-35
CSDK_FetchEventsByRange, 3-37
CSDK_FetchEventsByUID, 3-39
CSDK_FetchTasksByAlarmRange, 3-47
CSDK_FetchTasksByRange, 3-48
CSDK_FetchTasksByUID, 3-50
CSDK_FLAG_FETCH_AGENDA_ATTENDEE_ONLY, 5-2
CSDK_FLAG_FETCH_COMBINED, 5-2
CSDK_FLAG_FETCH_EXCLUDE_ACCEPTED, 5-2
CSDK_FLAG_FETCH_EXCLUDE_APPOINTMENTS, 5-2
CSDK_FLAG_FETCH_EXCLUDE_DAILYNOTES, 5-2
CSDK_FLAG_FETCH_EXCLUDE_DAYEVENTS, 5-3
CSDK_FLAG_FETCH_EXCLUDE_DECLINED, 5-3
CSDK_FLAG_FETCH_EXCLUDE_HOLIDAYS, 5-3
CSDK_FLAG_NONE, 5-2
CSDK_GetCapabilities, 3-6
CSDK_GetFirstFailure, 3-55
CSDK_GetFirstParseError, 3-56
CSDK_GetFirstResult, 3-58
CSDK_GetHandle, 3-16
CSDK_GetHandleInfo, 3-18
CSDK_GetNextFailure, 3-59
CSDK_GetNextParseError, 3-61
CSDK_GetNextResult, 3-63
CSDK_GetStatusLevels, 3-64
CSDK_GetStatusString, 3-65
CSDK_SetConfigFile, 3-4
CSDK_SetIdentity, 3-13
CSDK_StoreContacts, 3-30
CSDK_StoreEvents, 3-40
CSDK_StoreTasks, 3-51

D

Delete メソッド, 9-17
direntycachesize, 4-3

E

emailcachesize, 4-4

H

HTTP ヘッダー, 9-2

I

iCalendar, 2-2
itemcachesize, 4-4

J

Java クラス, 10-2

L

log_activity, 4-4
log_modulesinclude, 4-5

M

max_caldomain, 4-5
max_masternode, 4-6
max_user, 4-6
Modify メソッド, 9-18

N

NoOp, 9-10

O

Oracle Calendar Web Services
SOAP, 9-1
概要 (ツールキット), 8-2
説明, 8-2

P

Ping, 9-21

S

SDK

FAQ, 7-1

iCalendar, 2-2

アクセント付き文字, 7-3

アラーム, 2-7

概要, 1-1

関数, 3-4 ~ 3-65

キャラクタ・セット, 2-2

旧バージョン, 7-2

言語, 使用, 7-2

構成設定, 4-1

最適な使用例, 1-2

実装時の考慮事項, 2-1

セキュリティ・モデル, 2-6

データ・ストリーム, 2-9

トラブルシューティング, 7-1

内容, 1-2

ユーザー識別情報, 2-7

securitycachesize, 4-7

SOAP, 9-1

SOAP 障害, 10-5, 11-1

T

tzcachesize, 4-7

V

Visual C (SDK), 7-4

vQuery, 9-12

W

Web サービス

Create メソッド, 9-15

CWSL, 9-9

Delete メソッド, 9-17

HTTP ヘッダー, 9-2

Java クラス, 10-2

Modify メソッド, 9-18

NoOp, 9-10

Ping, 9-21

vQuery, 9-12

xCal, 9-22

X-ORACLE-EVENTTYPE, 9-30

休日, 9-29

繰返しイベント, 9-23

検索, 9-11

時間, ローカル, 10-6

終日イベント, 9-28

障害, 10-5, 11-1

セキュリティ, 9-3

タスク, 9-30

単純なイベント, 9-22

定期的なイベント, 9-25

データのフェッチ, 10-3

メモ, 9-28

優先度のマッピング, 9-30

X

xCal, 9-22

X-ORACLE-EVENTTYPE, 9-30

あ

アクセント付き文字 (SDK), 7-3

アプリケーション, タイプ

SDK, 1-1, 1-2

Web サービス, 8-2

アラーム, 2-7

い

イベント

繰返し, 9-23

終日, 9-28

単純, 9-22

定期的, 9-25

か

関数

SDK, 3-4, 3-65

改訂された (SDK), 3-65

レガシー (SDK), 3-66

き

キャラクタ・セット, 2-2
休日, 9-29

け

検索 (Web サービス), 9-11

し

時間, ローカル (WS), 10-6

せ

セキュリティ
 SDK, 2-6
 Web サービス, 8-3, 9-3

た

対象読者, viii
タスク, 9-30

て

データ・ストリーム (SDK), 2-9

ゆ

ユーザー識別情報 (SDK), 2-7

