

IBM Unica Interact
バージョン 8 リリース 6
2012 年 5 月 25 日

チューニング・ガイド

IBM

注

本書および本書で紹介する製品をご使用になる前に、13ページの『特記事項』に記載されている情報をお読みください。

本書は、IBM Unica Interact バージョン 8、リリース 6、モディフィケーション 0 および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原典： IBM Unica Interact
Version 8 Release 6
May 25, 2012
Tuning Guide

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

第1刷 2012.5

© Copyright IBM Corporation 2001, 2012.

目次

最高のパフォーマンスを得るための

Interact のチューニングについて 1

Interact API 1

インストールおよびネットワーク構成 1

セッション管理 2

 多数の並行セッションに対応する 2

対話式フローチャートの管理 3

サービスのチューニング 4

Web アプリケーション・サーバーのチューニング . . . 4

 JVM 引数 5

 接続プール 5

 チューニング用語 6

 データベースのチューニング 6

 ETL チューニング 7

 ロギング 9

IBM Unica 技術サポートへの連絡 11

特記事項 13

 商標 15

最高のパフォーマンスを得るための Interact のチューニングについて

Interact のインストールは、サード・パーティー・ツール (Web アプリケーション・サーバー、データベース、および Load Balancer など) および Marketing Platform と Campaign などの IBM® Unica コンポーネントを含むいくつかのコンポーネントで構成されています。これらのコンポーネントすべてには、パフォーマンスを向上させるためのいくつかのプロパティ、機能、および設定があります。Interact 自体にいくつかの構成プロパティがあり、これを使用してインストールのチューニングを行い、最高のパフォーマンスを得ることができます。

「最高のパフォーマンス」を定義することは困難です。それぞれの環境で、各実装の要求は異なります。例えば、対話式フローチャートのすべてのデータがリアルタイムのデータから収集される Interact の実装は、いくつかのデータベース・テーブルからの情報の読み取りを必要とする実装とはチューニング方法が異なります。

Interact ランタイム・パフォーマンスは、ハードウェア構成、ネットワーク構成、および Interact 構成など多くの要因の影響を受けます。以下のガイドラインおよび推奨事項は、ご使用の環境で得られる結果と異なる可能性があります。

以下のガイドラインは、関連のあるコンポーネント別に整理されています。設定を変更する順序は関係ありません。

Interact API

SOAP API の代わりに、Java Serialization API を使用します。Serialization API により、より良いスループットが実現し (アプリケーション構成によっては 5 から 10 倍)、レスポンス時間がより短くなります。

インストールおよびネットワーク構成

Interact サーバーが、複数の Interact API 呼び出しにまたがってセッション・データを維持する必要がある場合、分散セッション管理の代わりに、スティッキー・ロード・バランシングおよびローカル・セッション管理を使用します。分散モードでは、セッションの一貫性を保つために Interact ランタイム・サーバー間の通信コストがかかります。

Interact ランタイム・サーバーの IBM Unica Marketing Platform 構成設定で、Interact > sessionManagement > cacheType プロパティを local に設定します。

セッション管理

Interact ランタイム・サーバーの IBM Unica Marketing Platform 構成設定で、セッション・タイムアウト (Interact > sessionManagement > sessionTimeoutInSecs) を最小許容値に設定します。

それぞれの Interact セッションには、メモリー内のセッション・データの一部が入っています。セッションを長く維持するほど、並行メモリー所要量が増えます。例えば、毎秒 50 セッションを予想し、各セッションで 20 分間アクティブの状態が維持される場合、各セッションがまるまる 20 分間存続するとしたら、メモリーは 60,000 セッションに対応する必要があるかもしれません。

想定するシナリオに適した論理的な値にする必要があります。例えば、呼び出しのシステム・セッションは 1 分間アクティブ状態を維持する必要があるかもしれませんが、Web サイト・セッションは 10 分間アクティブ状態を維持する必要があります。

多数の並行セッションに対応する

いくつかの Interact 環境では、並行セッションの数が多いと、Interact ランタイムが使用可能メモリーを超え、システム減速またはメモリー不足エラーを引き起こします。メモリー不足の状況が起こりやすいのは、`maxNumberOfSessions` 構成パラメーター (**Interact | sessionManagement | maxNumberOfSessions**) をデフォルト設定よりも大きくした場合で、デフォルト設定を最大 100,000 セッションにしたとしても起こる場合があります。この問題を避けるために、`maxNumberOfSessions` の値を小さくするか、このセクションの指示に従ってシステム・メモリー・キャッシュを変更し、キャッシュ・メモリーをディスク・ストレージに切り替えます。この変更により、通常より多くの並行セッションが可能になります。

Interact ランタイムが、Java 仮想マシン (JVM) メモリー・ヒープにある使用可能なメモリーを超過するのを避けるために、メモリーのキャッシング・メカニズムを変更し、使用可能なメモリーを超えるデータのキャッシングにディスク・ストレージを使用するようにします。

Interact は、データのキャッシングに Ehcache と呼ばれるオープン・ソースの分散キャッシング・システムを使用します。デフォルトでは、Interact は、Ehcache キャッシングを管理するため IBM Unica Marketing Platform で指定された設定を使用します。ただし、Interact が開始するときにはいつでも、自動でロードされる Ehcache 構成ファイルを作成することによって、Interact の設定をオーバーライドすることができます。

開始時にカスタムの Ehcache 構成ファイルをロードするには、以下の条件を満たしている必要があります。

- 以下の例のように、ご使用の JVM には、パラメーター `interact.ehcache.config` プロパティーが入っている必要があります。

```
-Dinteract.ehcache.config=/temp/abc.xml
```

開始コマンド・スクリプト (Oracle WebLogic) または Admin Console (IBM WebSphere®) の Web アプリケーション・サーバー用に JVM プロパティーを設

定することができます。 `/temp/abc.xml` にある情報は、開始時にロードしたい Ehcache 構成を含む XML ファイルへの実際のパスです。

- 有効な Ehcache 構成設定を含む XML フォーマットの構成ファイルは、JVM プロパティーで指定された場所にある必要があります。

このプロパティーを設定しない場合、またはこのプロパティーを設定したのに、指定された場所に構成ファイルが存在しない場合、Interact はデフォルトのキャッシュ構成を使用します。

両方の条件が真である場合、Ehcache 構成ファイルは開始時にロードされ、その設定は、キャッシング・セッション・データ用のデフォルトInteract の構成パラメーターをオーバーライドします。

以下の例では、Ehcache のカスタマイズで使用することができる構成ファイルの例 (XML 形式) を表しています。

```
<ehcache xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="../../main/config/ehcache.xsd">

<defaultCache
maxElementsInMemory="10"
eternal="false"
timeToIdleSeconds="5"
timeToLiveSeconds="10"
overflowToDisk="true"
/>

<cache name="InteractCache"
maxElementsInMemory="5"
eternal="false"
timeToIdleSeconds="600"
timeToLiveSeconds="600"
overflowToDisk="true">

<cacheEventListenerFactory
class="com.unicacorp.interact.cache.EHCacheEventListenerFactory"
properties=""/>
</cache>

</ehcache>
```

このファイルを `/IBM/Interact/conf/Ehcache.xml` として保存する場合、Web アプリケーション用の JVM プロパティーを以下の例のように設定します。

```
-Dinteract.ehcache.config=/IBM/Interact/conf/Ehcache.xml
```

ソフトウェアの変更に関するオプションすべての一覧は <http://ehcache.org/files/documentation/EhcacheUserGuide-2.0-2.3.pdf> をご覧ください。

対話式フローチャートの管理

すべての対話式フローチャートを実行するには、少なくとも 1 つのスレッドが必要です。稼働中のシステムをモニターし、すべての対話式フローチャートに対して十分なスレッドがあるかを確認することができます。JMX コンソールを使用して、`com.unicacorp.interact.flowchart` にある `CurrentJobsInProcessBoxQueue` および `CurrentJobsInSchedulerQueue` の JMX 統計情報をモニターします。ロードのピー

ク時でもゼロになっていることが理想的で、それはフローチャート実行の要求に対応するための十分なスレッドがあることを意味します。

注: JMX コンソールの実行は、パフォーマンスに影響します。問題を診断するとき以外は、実稼働環境で JMX コンソールを実行しないようにしてください。

対話式フローチャートで使用するスレッドの数によって、これらのキューを制御できます。IBM Unica Marketing にあるフローチャート・スレッド・プールのサイズを Interact > flowchart で、Interact ランタイム用に設定します。

- `maxNumberOfFlowchartThreads` を少なくとも Interact クライアントで予想される同時ユーザーの最大数に設定します。例えば、同時ユーザーの最大数が 50 で、セグメンテーションへのそれぞれの呼び出しが 1 つのフローチャートで実行される場合、`maxNumberOfFlowchartThreads` を 50 に設定します。
- `maxNumberOfProcessBoxThreads` をフローチャート内の同時パスの平均数およびフローチャートが CPU 制約か入出力制約かに基づいて設定します。少なくとも `maxNumberOfFlowchartThreads` に等しくなるようにします。例えば、フローチャート内の同時パスの平均数が 2 で、すべての処理ボックスが CPU 制約の場合、`maxNumberOfProcessBoxThreads` を $2 * \text{maxNumberOfFlowchartThreads}$ に設定します。処理ボックスが入出力制約されている場合 (処理ボックスが「選択」または「スナップショット」処理などのデータベースのルックアップまたは書き込みを行う場合など)、その数値はより大きい値に設定する必要があるかもしれません。
- `minNumberOfFlowchartThreads` を `maxNumberOfFlowchartThreads` に等しくなるように設定します。同様に、`minNumberOfProcessBoxThreads` を `maxNumberOfProcessBoxThreads` に等しくなるように設定します。

サービスのチューニング

Interact には、Interact の様々なコンポーネントによるデータベースの読み取りおよび書き込みを管理するいくつかのサービスがあります。例えば、組み込まれた学習モジュール、およびコンタクトとレスポンス履歴モジュールがあります。

(Interact > services > *service name* > cache > threshold にある) 各サービスのしきい値を、1秒あたりのオペレーションの数およびデータベースへの各挿入にかかる時間に基づいて、適切な値に設定します。例えば、システムのスループット要求が毎秒 500 トランザクションで、それぞれのトランザクションに 2 つのログのコンタクトの呼び出しがある場合、`contactHist` しきい値は、バッチの書き込みにかかる平均時間および毎秒 1000 個のログのコンタクトに基づく値に設定する必要があります。

Web アプリケーション・サーバーのチューニング

以下のセクションに加え、パフォーマンスのチューニングのベスト・プラクティスについては、ご使用の Web アプリケーション・サーバーおよびオペレーティング・システムに関する資料をご覧ください。

JVM 引数

Java 仮想マシン (JVM) 引数は、開始コマンド・スクリプト (Oracle WebLogic) またはご使用の Web アプリケーション・サーバーの Admin Console (IBM WebSphere) で定義されています。

- ご使用のオペレーティング・システム、Web アプリケーション・サーバー、および JVM に、最新のサービス・パックおよびパッチがインストールされていることを確認します。
- Sun HotSpot VM を使用して、最高のパフォーマンスを得るには、`-server` 引数を使用します。
- サーバーのメモリーの可用性に基づき、JVM の最大ヒープ・サイズを決定します。(Interact はメモリー集約的なアプリケーションではありません)。ヒープの最大サイズおよび最小サイズを同じに設定すると (`-Xmx` および `-Xms` 引数を使用)、起動時間が長くなりますが、より良いスループットが得られます。
- レスポンス時間が数秒間に及ぶなど、周期的にアプリケーションが無応答になる場合、ガーベッジ・コレクション・ポリシーのチューニングが必要かもしれません。JMX コンソールを使用して、以下の引数を使用可能にした後にガーベッジ・コレクション出力を調べることで、ガーベッジ・コレクションの実行をモニターします。

```
-verbosegc -XX:+PrintGCDetails
```

- このテストでは、Low Pause Collector が見つかり、ガーベッジ・コレクション関連の処理が遅いという問題を、スループットを保ったまま除去しました。以下は、2 GB JVM ヒープに役立つことが判明したオプションの 1 つのセットです。

```
-XX:+UseConcMarkSweepGC -Xmn512m -XX:SurvivorRatio=6
```

一般的に、若いコレクションは、全体のヒープの 1/4 から 1/2 までであるべきです。Survivor Space は、若いコレクションのサイズの 1/8 に設定することができます。

参照

- 5.0 Java 仮想マシンで、ガーベッジ・コレクションをチューニング (http://java.sun.com/docs/hotspot/gc5.0/gc_tuning_5.html)
- Java ホワイト・ペーパーのチューニング (<http://java.sun.com/performance/reference/whitepapers/tuning.html>)

接続プール

アプリケーション・サーバー・コンソールを使用して、Interact ランタイム・データ・ソースの接続プールのサイズを設定します。プロファイルのロード、オファーマイグレーションのロード、フローチャートからの読み取りおよび書き込み、および学習からの読み取りなど、セッションの存続期間中の同時ユーザー数および接続数を考慮に入れます。

機能/オプション	有効な場合、接続が必要
以下の機能の少なくとも 1 つが有効 <ul style="list-style-type: none"> プロファイル・テーブルのロード オファー非表示テーブルのロード スコア・オーバーライド・テーブルのロード 	startSession または setAudience へのクライアント同時呼び出しにつき 1 つの接続 1 つのテーブルのみのロードが有効になっているか、3 つのテーブルすべてのロードが有効になっているかは関係ありません。
学習	2 つの接続
少なくとも 1 つのロギング・サービスまたはトラッキング・サービスが有効です	Interact > services > threadManagement > flushCacheToDB > maxPoolSize の値。デフォルトは 5 です。
少なくとも 1 つのデータベース呼び出しを行うフローチャート	Interact > flowchart > maxNumberOfFlowchartThreads の値。デフォルトは 25 です。

例えば、以下の要件があるとします。

- データベース接続を取得する場合、startSession への 30 個の同時呼び出しが待ち状態ではないことが必要 (30)
- 学習をオンにする (2)
- すべてのサービスをオンにする (5)
- データベース接続を行う少なくとも 1 つのフローチャートをデプロイする (25)
- 現在のデフォルトのままにする (0)

その後、データベース接続プールのサイズを最小 62 (30+2+5+25) に設定し、利用者が一人も接続待ちにならない最適なパフォーマンスにします。

チューニング用語

以下のチューニングのガイドラインは、下記の用語に基づいて決定されました。

レスポンス時間: Interact ランタイム・サーバーが API 要求にレスポンスするのにかかる時間を、クライアント・サイドから測定した時間です。

スループット: 1 秒当たりのトランザクションの数です。

トランザクション: startSession および setAudience などの InteractAPI クラスによって定義された呼び出しを含む Interact による APIInteract ランタイム・サーバーへの呼び出しです。複数のコマンドを含むとしても、executeBatch 呼び出しは 1 つのトランザクションです。これらには、Offer クラスなどのレスポンス・オブジェクトと同時に処理されるメソッドは含まれません。

データベースのチューニング

プロファイル・テーブル、オファー非表示テーブル、およびスコア・オーバーライド・テーブルに適切なインデックスを追加します。

- プロファイル・テーブル。オーディエンス項目に一意のインデックスを作成します。

- オファー非表示テーブル。オーディエンス項目にインデックスを作成します。
- スコア・オーバーライド・テーブル。オーディエンス項目にインデックスを作成します。

また、これらのインデックスに関する統計情報が最新であることを確認します。例えば、オーディエンス ID が 2 つの列 CustomerId および HouseholdId の組み合わせである場合、すべてのテーブルのこれらの列にインデックスを作成し、統計情報を更新します。

ETL チューニング

コンタクトおよびレスポンス履歴モジュールを構成するとき、モジュールは、バックグラウンドで抽出、変換、ロード (ETL) プロセスを行い、データをランタイム・ステーキング・テーブルから Campaign コンタクトおよびレスポンス履歴テーブルに移動させます。

このセクションでは、ETL ツールのパフォーマンスを向上させるために IBM Unica Interact で変更したいオプションの構成設定について説明します。これらデフォルト設定の構成パラメーターを変更する必要はないかもしれませんが、変更する場合、このセクションおよび「IBM Unica Interact 管理者ガイド」にあるガイドラインに従い、ツールのパフォーマンスを変更します。

このセクションで説明されているすべてのプロパティは、Campaign キャンペーン | パーティション | partition[n] | 対話 | contactAndResponseHistTracking の構成プロパティにあります。

構成プロパティ	値および説明
processSleepIntervalInMinutes	Interact コンタクトおよびレスポンス履歴モジュールの、Interact ランタイム・ステーキング表から Campaign コンタクトおよびレスポンス履歴テーブルにデータをコピーする待ち時間を分単位で表したものです。デフォルト値は 60 です。
purgeOrphanResponseThresholdInMinutes	このプロパティでは、対応するコンタクトがない場合に、Interact がレスポンスをパージする待ち時間を決定します (別名「孤立したレスポンス」)。デフォルトは 180 です。ただし、レコード処理のためにコンタクトおよびレスポンスの処理時間に大きな遅延が生じるため、この値を上げて、あまりにも急にレスポンスがパージされてしまうのを避けることができます。

構成プロパティ	値および説明
maxJDBCInsertBatchSize	<p>1 回の反復でコンタクトおよびレスポンス履歴モジュールが処理するレコードの合計数のうち、これが Campaign システム・テーブルにクエリーをコミットする前に JDBC バッチが処理する (およびバッチで集める) レコードの最大数です。デフォルト値は 1000 です。</p> <p>この値は maxJDBCFetchBatchSize プロパティと共に処理されるため、このプロパティの値が大幅に増えた場合、この値も増やす必要があるかもしれません。例えば、maxJDBCFetchBatchSize を 2,500,000 に設定した場合、この値を 10,000 に増やし、レコードの増加に対応できるようにします。</p> <p>この値を増やすと、メモリー所要量も増えることに注目してください。このプロパティの 10,000 という設定は、メモリー要求にかなう適切な上限です。</p>
maxJDBCFetchBatchSize	<p>ステージング・データベースからフェッチするレコードの最大数を決定し、ETL バッチ処理操作を行います。デフォルト値は 1000 ですが、コンタクトおよびレスポンス履歴モジュールのパフォーマンスをチューニングするには、この値を通常毎日処理されるコンタクト履歴レコードの数よりも大きい値に設定するようにします。</p> <p>このプロパティは、maxJDBCFetchChunkSize および maxJDBCInsertBatchSize と共に使用され、レコードがどのように処理されるかを決定します。例えば、ここに示されているような値が設定されているとします。</p> <ul style="list-style-type: none"> • maxJDBCFetchBatchSize: 30000 • maxJDBCFetchChunkSize: 1000 • maxJDBCInsertBatchSize: 1000 <p>この例では、30,000 個のレコードをフェッチします (または、レコード数が 30,000 以下の場合はそのレコード総数)。その後、コンタクトおよびレスポンス履歴モジュールは、1 回に 1,000 個のレコードを処理しながら、30,000 個のレコード内をループし、ステージング・テーブルの 1,000 個のレコードにマークを付け、1,000 個のレコードをコンタクト履歴テーブルの詳細に挿入します。</p>

構成プロパティ	値および説明
maxJDBCFetchChunkSize	JDBC チャンクのレコードの最大数 (最大 maxJDBCFetchBatchSize レコードまで) を決定し、それぞれのパスを通して処理します。デフォルト値は 1000 です。場合によっては、この値を MaxJDBCInsertBatchSize プロパティ値よりも大きくすることで、パフォーマンスを向上できます。
deleteProcessedRecords	コンタクトおよびレスポンス履歴レコードを処理した後、レコードを保持するかどうかを指定するこのプロパティは、デフォルトで「はい」に設定されています。この値を変更することによって、ETL プロセスのデータ・フローをさらにコントロールし、(後の判断時までこれらのレコードのページを遅らせることで) パフォーマンスに影響を与えることができるようになります。ただし、これらのレコードを適切な時に削除できるように、手動でレコードのメンテナンスを行うことに精通している必要があります。この設定を変更する場合は注意が必要です。
fetchSize	JDBC fetchSize に値を入れることにより、レコードの大規模なバッチ処理のパフォーマンスを向上させることができますが、ネットワークのパフォーマンスが向上する代わりに、メモリー使用におけるフェッチのサイズが大きくなってしまいます。この設定の調整の詳細については「 <i>IBM Unica Interact 管理者ガイド</i> 」にあるこの構成プロパティの説明を参照してください。

このセクションで説明されている各構成プロパティの詳細については、構成ページのオンライン・ヘルプを参照するか、「*IBM Unica Interact 管理者ガイド*」を参照してください。

ロギング

ログ・レベルが、INFO または ERROR に設定されていることを確認します。実稼働環境で、DEBUG または TRACE などの冗長ログ設定を使用することは決してしないでください。

3 つの場所でロギングを構成できます。

- interact_log4j.properties ファイルで、ロギング・レベルを設定します。デフォルトでは、このファイルは <install_dir>/Interact/conf ディレクトリーにインストールされています。<install_dir> は、ご使用の IBM Unica 製品がインストールされている親ディレクトリーです。
- Interact API がロギングしていないことを確認します。ロギングは、startSession および setDebug 方式で使用可能なブール設定によって決定されます。

- JMX モニターが activateInfo JMX オペレーションで、Info に設定されていることを確認します。

IBM Unica 技術サポートへの連絡

ドキュメンテーションを参照しても解決できない問題があるなら、指定されているサポート窓口を通じて IBM Unica 技術サポートに電話することができます。このセッションの情報を使用するなら、首尾よく効率的に問題を解決することができます。

サポート窓口が指定されていない場合は、IBM Unica 管理者にお問い合わせください。

収集する情報

IBM Unica 技術サポートに連絡する前に、以下の情報を収集しておいてください。

- 問題の性質の要旨。
- 問題発生時に表示されるエラー・メッセージの詳細な記録。
- 問題を再現するための詳しい手順。
- 関連するログ・ファイル、セッション・ファイル、構成ファイル、およびデータ・ファイル。
- 「システム情報」の説明に従って入手した製品およびシステム環境に関する情報。

システム情報

IBM Unica 技術サポートに電話すると、実際の環境に関する情報について尋ねられることがあります。

問題が発生してもログインは可能である場合、情報の大部分は「バージョン情報」ページで入手できます。そのページには、インストールされている IBM Unica のアプリケーションに関する情報が表示されます。

「バージョン情報」ページは、「ヘルプ」>「バージョン情報」を選択することにより表示できます。「バージョン情報」ページを表示できない場合、どの IBM Unica アプリケーションについても、そのインストール・ディレクトリの下にある `version.txt` ファイルを表示することにより、各アプリケーションのバージョン番号を入手できます。

IBM Unica 技術サポートの連絡先情報

IBM Unica 技術サポートとの連絡を取る方法については、IBM Unica 製品技術サポートの Web サイト (<http://www.unica.com/about/product-technical-support.htm>) を参照してください。

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒103-8510
東京都中央区日本橋箱崎町19番21号
日本アイ・ビー・エム株式会社
法務・知的財産
知的財産権ライセンス渉外

以下の保証は、国または地域の法律に沿わない場合は、適用されません。IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Corporation
170 Tracer Lane
Waltham, MA 02451
U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

表示されている IBM の価格は IBM が小売り価格として提示しているもので、現行価格であり、通知なしに変更されるものです。卸価格は、異なる場合があります。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。

できます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。これらのサンプル・プログラムは特定物として現存するままの状態を提供されるものであり、いかなる保証も提供されません。IBM は、お客様の当該サンプル・プログラムの使用から生ずるいかなる損害に対しても一切の責任を負いません。

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

商標

IBM、IBM ロゴ、および ibm.com は、世界の多くの国で登録された International Business Machines Corp. の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、『www.ibm.com/legal/copytrade.shtml』をご覧ください。



Printed in Japan