

# アプリケーション・コンティニュイティ 準備のためのチェックリスト

Oracle ホワイト・ペーパー | 2018 年 9 月



## 目次

目次 .....	0
アプリケーション・コンティニューイティ・チェックリスト .....	1
アプリケーションのタイムアウトの整合 .....	1
TNS または URL における高可用性の構成 .....	3
高速アプリケーション通知 (FAN) の有効化 .....	5
FAN の監視 .....	5
アプリケーション・コンティニューイティ・カバレッジ .....	6
アプリケーション・コンティニューイティを使用する場合の手順 .....	7
Oracle JDBC アプリケーションの場合 .....	9
その他の資料 .....	12

## アプリケーション・コンティニューイティ・チェックリスト

以下のチェックリストは、Oracle Database 12c のアプリケーション・コンティニューイティ機能を使用するように使用環境を準備する場合に役立ちます。アプリケーション・コンティニューイティがデータベース・サービスで有効になっていない場合、あるいはアプリケーションで使用されない場合でも、このホワイト・ペーパーに記載されている内容には、可用性を維持できるようにシステムを準備する上で大きな価値があります。

### アプリケーションのタイムアウトの整合

アプリケーション・レベルのタイムアウトが、基盤システムの検出およびリカバリのための時間として設定されているタイムアウトより短いと、基盤となるリカバリとリプレイを完了するための時間が不足します。時間の整合性が保たれていないと、システムのリカバリが完了する前にアプリケーション・コンティニューイティによるリプレイが開始され、潜在的に、複数のリプレイ試行が成功するまで繰り返される可能性があり、もっと悪い場合にはリクエストが完全にタイムアウトし、アプリケーションまたはユーザーにエラーが返されます。

アプリケーションで `READ_TIMEOUT` や `HTTP_REQUEST_TIMEOUT`、または何らかのカスタム・タイムアウトを使用することを考慮してください。続いて、次のシンプルな指針を適用します。

```
READ_TIMEOUT > EXADATA の特別なノード・エビクション (FDDN) (12.1.0.2 で 2 秒)
READ_TIMEOUT > MISSCOUNT (デフォルトは 30 秒、12.1.0.2 では変更可能)
READ_TIMEOUT > Data Guard Observer: FastStartFailoverThreshold (デフォルトで 30 秒、変更可能)
FastStartFailoverThreshold > MISSCOUNT (2 回以上)
READ_TIMEOUT > FAST_START_MTRR_TARGET
READ_TIMEOUT > NET level: (RETRY_COUNT+1) * RETRY_DELAY
および
READ_TIMEOUT < Replay_Initiation_Timeout (サービス上で変更可能、デフォルトで 300 秒)
```

リクエストのキャンセルが早すぎないようにするため、アプリケーションのタイムアウト値を次の最大値より大きい値にする必要があります。

```
(MISSCOUNT (または FDDN) + FAST_START_MTRR_TARGET),
(FastStartFailoverThreshold +
FAST_START_MTRR_TARGET + 開く時間)
```

## 現在推奨される 12.2.0.1 のパッチ

JDBC	26079621	BLR for 25977056	AC: UNITS FOR TRANSPORT_CONNECT_TIMEOUT CHANGED WITH NO QUALIFIER AC : 修飾子なしで変更された TRANSPORT_CONNECT_TIMEOUTの単位		
	26548364	BLR for 25608170	AFTER TIMEOUT OF AN ADDRESS IN URL, NOT TRY OTHER ADDRESS AND RETRY URLに含まれるアドレスのタイムアウト後は、他のアドレスで再試行しない		
	27132026	MLR	26150506	BLR for 25255689	AC: SPEC-J 4% INCREASE BETWEEN 12.1 JARS AND 12.2 JARS AC : 12.1 JARSと12.2 JARSの間で SPEC-J 4 %増加
			25824159	BLR for 25423296	AC: NEC OBSERVED THAT ALL IPS IN SCAN ADDRESS ARE NOT USED FOR JAVA NET AC : NECの観察では、スキャン・アドレス内のすべてのIPSはJava NETでは使用されない
			26526163	BLR for 26525873	KERBEROS: "protocol violation" exception while getting a connection (24336861) KERBEROS : 接続を確認中の"プロトコル違反"例外 (24336861)
			26548364	BLR for 25608170	AFTER TIMEOUT OF AN ADDRESS IN URL, NOT TRY OTHER ADDRESS AND RETRY URLに含まれるアドレスのタイムアウト後は、他のアドレスで再試行しない
UCP	26950182		ConnectionWaitTimeout should exceed CONNECT_TIMEOUT ConnectionWaitTimeoutはCONNECT_TIMEOUTを超過		
	27479395	MLR	26843664		UCP: java.lang.NullPointerException at ServiceMember.averageAdvisedLoad UCP : ServiceMember.averageAdvisedLoadでのjava.lang.NullPointerException
			27030974		UCP: ONE THREAD STUCK IN LOADBALANCER.UNDERLOADEDINSTANCE() GROWING CPU USAGE UCP : LOADBALANCER.UNDERLOADEDINSTANCE()でスレッドが1つスタックし、CPUの使用率が上昇
			27036408		UCP: connection attempt failed with ORA-12521 UCP : ORA-12521で接続の試行に失敗
	28538380	BLR for 27748210	BLR BACKPORT OF BUG 27748210 ON TOP OF 12.2.0.1.0 12.2.0.1.0へのBUG 27748210のBLRバックポート		NEED TO GUARD AGAINST OUTFOFMEMORYERROR FOR LARGE/LONG REQUESTS LARGE/LONGリクエストのための OUTFOFMEMORYERRORに対するガードの必要

WLS	24919627	BLR for 23047782	AC: AQ JMS USES ORACLE CONCRETE CLASSES - DEPRECATED JAVA CODE AC : AQ JMSがOracle具象クラス (非推奨Javaコード) を使用		
	26336757	BLR for 26478166	AC: - WLS DO DB POOL REBUILD AFTER RECEIVE SERVICE DOWNEVENT FOR STANDBY AC : スタンバイのサービス停止イベントの受信後にWLS DO DBプールを再ビルド		
ONS	22743719	MLR	21867447		ONS JAVA CLIENT CLOSES SUBSCRIBERS ON GC OF ONS OBJECT ONS JavaクライアントがONSオブジェクトのGCでサブスクリバをクローズ
			21258754		BLOCKING WAIT WHEN SHUTTING DOWN ONS CLIENT ONSクライアントのシャットダウン時のブロッキング待ち
RDBMS	26877522	BLR for 16727454	AC:SUPPORT FOR REPLAYING ENQUEUE,AQ ID SEQUENCE NEED TO KEEP AC : エンキューのリプレイのサポート、AQ IDシーケンスの維持が必要		
	26546919	BLR for 23047782	AC: AQ JMS USES ORACLE CONCRETE CLASSES - DEPRECATED JAVA CODE AC : AQ JMSがOracle具象クラス (非推奨Javaコード) を使用		

## TNSまたはURLにおける高可用性の構成

フェイルオーバー、スイッチオーバー、フォールバック、基本的な起動時に接続を成功させるには、次の TNS/URL 構成を推奨します。

TNSnames または URL で、RETRY\_COUNT、RETRY\_DELAY、CONNECT\_TIMEOUT、TRANSPORT\_CONNECT\_TIMEOUT パラメータを設定し、サービスと接続が正常に行われるまで接続リクエストが待機できるようにします。

CONNECT\_TIMEOUT を高い値に設定して、ログイン・ストームを防止します。値を低くすると、アプリケーションまたはプールのキャンセルや接続の再試行のために、ログイン合戦が発生することがあります。

(RETRY\_COUNT+1) × RETRY\_DELAY または CONNECT\_TIMEOUT を応答時間の SLA より大きい値に設定しないでください。アプリケーションは応答時間の SLA 内で接続するか、エラーを受信することになります。

これらは、接続に高可用性を構成する場合の一般的な推奨事項です。EZCONNECT には高可用性の能力がないため、クライアントで簡易接続ネーミングを使用しないでください。

これは、18c と 12.2 用のすべての Oracle ドライバについて推奨される TNS です。

```
Alias (or URL) = (DESCRIPTION =  
  (CONNECT_TIMEOUT= 120) (RETRY_COUNT=20) RETRY_DELAY=3)  
  (TRANSPORT_CONNECT_TIMEOUT=3)  
  (ADDRESS_LIST =  
    (LOAD_BALANCE=on)  
    (ADDRESS = (PROTOCOL = TCP) (HOST=primary-scan) (PORT=1521)))  
  (ADDRESS_LIST =  
    (LOAD_BALANCE=on)  
    (ADDRESS = (PROTOCOL = TCP) (HOST=secondary-scan) (PORT=1521)))  
  (CONNECT_DATA=(SERVICE_NAME = gold-cloud)))
```

12.1 の JDBC 接続では、次の式を使用します。

```
(DESCRIPTION =
(CONNECT_TIMEOUT= 15) (RETRY_COUNT=20) (RETRY_DELAY=3)
(ADDRESS_LIST =
  (LOAD_BALANCE=on)
  (ADDRESS = (PROTOCOL = TCP) (HOST=primary-scan) (PORT=1521)))
(ADDRESS_LIST =
  (LOAD_BALANCE=on)
  (ADDRESS = (PROTOCOL = TCP) (HOST=secondary-scan) (PORT=1521)))
(CONNECT_DATA=(SERVICE_NAME = gold-cloud)))
```

### 高速アプリケーション通知 (FAN) の有効化

FAN を有効にする必要があります。FAN は、アプリケーションのフェイルオーバーを中断する場合の必須コンポーネントです。ノードまたはネットワークが停止した場合は、アプリケーションをリアルタイムで中断する必要があります。FAN を有効化しておかないと、HARD フィジカル障害の発生時にアプリケーションがハングします。FAN なしでのハング時間は 8~15 分間です。Oracle Database 12c 以降、FAN では重要な 3 つの機能強化が図られています。

- FAN は、Oracle Real Application Clusters (Oracle RAC) に付属しているデフォルトの機能で、自動構成されて有効化されています。FAN は URL を読み取り、それ自体をクライアントで構成します。FAN は、いつでも、Grid Infrastructure クラスタで構成されます。上記の URL 形式を使用することは、FAN の自動構成を行う上で重要です (別の形式を使用すると、FAN を自動構成できなくなります)。
- すべての Oracle クライアントでは、Oracle Notification Service を FAN の転送サービスとして使用します。
- FAN は、FAN イベントでデータセンターを包含することができるよう、Oracle Global Data Services (Oracle GDS) によってポストされます。

### FANの監視

FANWatcher ユーティリティを使用してイベントのポスティングと受信を検証します。更新されたバージョン (2017 年 7 月) の FANWatcher は、OTN から、または WebLogic Server ブログ経由で入手できます。以下を参照してください。

高速アプリケーション通知 -

<http://www.oracle.com/technetwork/database/options/clustering/applicationcontinuity/learnmore/fastapplicationnotification12c-2538999.pdf>

FAN イベントの監視 -

<https://blogs.oracle.com/weblogicserver/monitoring-fan-events>

UCP トレースは、以下の手順で有効化できます。UCP/Tomcat で FAN が処理中であることを示すには：ロギングはプロパティ・ファイルを使用して構成可能です。プロパティ・ファイルの場所は、ロギング構成ファイルのプロパティの Java プロパティとして設定する必要があります。たとえば、次のように設定します。

```
java -Djava.util.logging.config.file=myLog.properties
```

ロギング・プロパティ・ファイルでは、ログの書込みに使用するハンドラ、ログの書式指定に使用するフォーマッタ、デフォルトのログ・レベル、および特定のパッケージまたはクラスのログ・レベルを定義します。たとえば、次のように設定します。

```
handlers = java.util.logging.ConsoleHandler
java.util.logging.ConsoleHandler.level = ALL
java.util.logging.ConsoleHandler.formatter =
java.util.logging.SimpleFormatter

oracle.ucp.level = FINEST
oracle.ucp.jdbc.PoolDataSource = WARNING
```

機能に固有のロギングは、ロギング構成ファイルでプロパティを設定することによって有効化できます。すべての機能の場合：

```
clio.feature.all = on
```

または、次のような固有の機能の場合：

```
clio.feature.conn_construction = on
clio.feature.high_availability = on
clio.feature.load_balancing = on
clio.feature.transaction_affinity = on
```

詳しくは、オラクルのドキュメント

『Universal Connection Pool 開発者ガイド』 - 第 12 章「UCP でのロギングの概要」、  
『Oracle Database JDBC 開発者ガイド』 - 第 34 章「JDBC の診断機能」を参照してください。

## アプリケーション・コンティニューイティ・カバレッジ

Oracle Database 12c のアプリケーション・コンティニューイティは以下のクライアントをサポートします。

- Oracle JDBC Replay ドライバ 12c 以降。"リプレイ・ドライバ"として参照されるアプリケーション・コンティニューイティのために Oracle Database 12c とともに提供される JDBC ドライバ機能。
- Oracle Universal Connection Pool (UCP)
- Oracle WebLogic Server 12c および UCP を使用するサード・パーティの JDBC アプリケーション・サーバー
- Java 接続プール、あるいは Oracle JDBC を使用するスタンドアロン Java アプリケーション - リクエスト・バウンダリとの Replay ドライバ 12c 以降。
- Oracle Tuxedo

- OCI セッション・プール 12c リリース 2 以降。
- SQL\*Plus 12c リリース 2 以降。 (“ac”コマンド・ライン・スイッチを使用)
- ODP.NET 非管理プロバイダ 12c リリース 2 以降。

サードパーティの Java ベース・アプリケーション・サーバーを使用する場合、高可用性を実現する最も効果的な方法は、データ・ソースを UCP に置き換えます。このアプローチは、IBM WebSphere、Apache Tomcat、Red Hat JBoss、Spring、Hibernate などを含む多くのアプリケーション・サーバーでサポートされています。UCP をデータ・ソースとして使用する高速接続フェイルオーバー、ランタイム・ロード・バランシング およびアプリケーション・コンティニューイティなどの UCP 機能を完全なサーティフィケーションで使用されま

## アプリケーション・コンティニューイティを使用する場合の手順

### 接続プールへの接続の返却

アプリケーションは、リクエストごとに接続を接続プールに返却します。ベスト・プラクティスは、必要とする場合にのみアプリケーションが接続をチェックアウトすることです。使用していないときに接続を保持することは、勧められる方法ではありません。したがってアプリケーションでは、接続をチェックアウトし、その後処理が完了すると直ちにその接続をチェックインします。続いて接続は、後続の他のスレッドまたは再び必要になったときに自分のスレッドで使用できるようになります。この手法に従うことにより、キャプチャを再開および終了するのに安全な場所を識別するためのアプリケーション・コンティニューイティで使用するリクエストの境界も組み込まれます。

## FAILOVER\_RESTORE の使用

アプリケーションが接続の値を事前設定しているかどうかをチェックします。

一部のアプリケーションと中間層アプリケーションでは、たとえばすべての接続の言語またはタイムゾーンが事前設定されるように、接続プールを構成します。

セッションの状態がリクエスト外部の接続で意図的に設定され、リクエストにおいてこの状態が予期されている場合、リプレイでは、リプレイする前にこの状態を再作成する必要があります。

もっとも一般的な状態は、FAILOVER\_RESTORE を LEVEL1 に設定することによって自動的にリストアされます。

以下のいずれかのオプションを選択します。

- FAILOVER\_RESTORE=LEVEL1 (サービスで設定)
- 接続初期化コールバック (Java の場合) または (古い) TAF コールバック (OCI の場合)
- Oracle Universal Connection Pool または Oracle WebLogic Server の接続ラベル付け

Oracle RDBMS 12.2 以降では、FAILOVER\_RESTORE=LEVEL1 が推奨される方法です。

## アプリケーションでの mutable 使用の有効化

mutable 関数とは、実行されるたびに新しい値を返す可能性のある関数です。mutable 関数の元の結果を保持する機能は、SYSDATE、SYSTIMESTAMP、SYS\_GUID、sequence.NEXTVAL 向けに提供されています。元の値が保持されず、再実行時に異なる値がアプリケーションに返される場合は、再実行が拒否されます。

アプリケーション・ユーザーには GRANT KEEP を使用し、シーケンス所有者には KEEP 句を使用して、可変オブジェクトを構成します。KEEP 権限が付与されると、再実行時に元の関数の結果が適用されます。

たとえば、次のように指定します。

```
SQL> GRANT [KEEP DATE TIME | KEEP SYSGUID] ... to USER
```

```
SQL> GRANT KEEP SEQUENCE mySequence to myUser on sequence.object
```

## サイド・エフェクト無効化の必要性

アプリケーションが別途指定しない限り、サイド・エフェクトが再生されます。外部アクションを使用するアプリケーションは、サイド・エフェクトのあるリクエストがリプレイに意味があるかどうかを判断するためにレビューされます。例えば、アプリケーションは再度電子メールを送信するか、再度監査するか、ファイルを再度転送したいか？頻繁にサイド・エフェクトを再生することが望ましいです。しかし、時にはそれがない方が良いかもしれません。リクエストに再生すべきではない外部アクションがある場合、そのリクエストは、アプリケーション・コンティニューイティを有効にしていない接続を使用することができます。または、再生を無効にすることができます。そのリクエストは、Java の場合は `disableReplay()` API、OCI の場合は `OCIRequestDisableReplay` を使用します。他のすべてのリクエストは引き続き再生されます。

## プロテクション・レベル

ORAchk アプリケーション・コンティニューイティのカバレッジ分析レポートを使用して、フェイルオーバーの前に、アプリケーション・コンティニューイティによって完全に保護されていないリクエストに対しては、どこに発生したのかが報告されています：

Measure Coverage

<http://www.oracle.com/technetwork/database/options/clustering/applicationcontinuity/overview/application-continuity-wp-12c-1966213.pdf>

## Oracle JDBCアプリケーションの場合

### Oracle JDBC ステートメント・キャッシュの有効化

Oracle JDBC ベースのアプリケーションの場合は、パフォーマンスを高めるため常に Oracle JDBC ステートメント・キャッシュを使用します。アプリケーション・コンティニューイティを使用している場合は、Oracle JDBC のリファレンス・ドキュメントの説明に従って、アプリケーション・サーバー・レベル (Tomcat、WebLogic、WebSphere など) でステートメント・キャッシュを無効にすることが必須となります。

アプリケーション・サーバー・レベルのステートメント・キャッシュ (WebLogic やサード・パーティ製アプリケーション・サーバーのステートメント・キャッシュなど) が有効化されている場合は、リプレイが使用されるときは無効化する必要があります。代わりに、Oracle JDBC ステートメント・キャッシュを構成します。このステートメント・キャッシュは、Oracle JDBC や Oracle 製品用に最適化されており、アプリケーション・コンティニューイティに対応しているため、パフォーマンスに優れています。

`oracle.jdbc.implicitstatementcachesize=nnn` を使用してください。

`nnn` を正の値に設定すると、暗黙的ステートメント・キャッシュが有効化されます。

## ガベージ・コレクタのチューニング

多くのアプリでは、デフォルトのガベージ・コレクタをチューニングすることが適切です。極めて高い性能のアプリの場合には、JVM チューニングを実行する必要があります。性能が極めて高い場合に推奨されるのは、以下の設定を Java コマンドラインに追加することです（両方の属性を同じ値に設定する必要があります）。

```
java -Xms 2000m -Xmx 2000m
```

## COMMIT

アプリケーション・コンティニューイティでは、すべてのスタイルの COMMIT（トップレベル、AUTOCOMMIT、PLSQL に埋め込まれている COMMIT）をサポートしています。アプリケーションでトップレベルの COMMIT を使用する場合は、それはスタンドアロンの OCOMMIT または COMMIT () であり、SESSION\_STATE\_CONSISTENCY=STATIC モード（12.2.0.1）および透過的アプリケーション・コンティニューイティ（TAC）を使用する場合を含め、リプレイを完全にサポートしています。アプリケーションで PLSQL に埋め込まれた COMMIT または AUTOCOMMIT を使用するとき、COMMIT を含む呼び出しの実行が完了しなかったことがアプリケーション・コンティニューイティで検出された場合は、リプレイできません。アプリケーション・コンティニューイティでは正しい処理が行われます。

Oracle JDBC アプリケーションで“トップレベル”の COMMIT を使用する場合は、アプリケーションで、または UCP のプロパティとして、AUTOCOMMIT を無効にする必要があります。これは、Apache Tomcat、IBM WebSphere、RedHat JBoss などのサード・パーティのアプリケーション・サーバーに UCP が埋め込まれている場合には特に重要です（アプリケーションで AUTOCOMMIT が不要な場合）。

### 非推奨 Java の削除

Java アプリケーションに対して、以下の場所にあるアプリケーション・コンティニューイティ・ホワイト・ペーパーの説明に従って、具象クラス分析を実行してください。

Oracle Database 12c Release 2 のアプリケーション・コンティニューイティ：

*具象クラスの AC チェックの使用*

<http://www.oracle.com/technetwork/jp/database/database-cloud/private/application-continuity-wp-12c-1966213-ja.pdf>

Oracle JDBC の非推奨クラス（具象クラス）は使用しないようにする必要があります。これは、12.1 でのハード制限ですが、18c のいくつかの具象クラスでは緩和されています。具象クラスの非推奨化に関する情報（アプリケーションで推奨されない具象クラスを使用している場合の対策を含む）については、My Oracle Support Note 1364193.1

<https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1364193.1> を参照してください。

## 診断

“OracleLog.properties”ファイルで以下を使用します（ログ・ファイルをホスティングしているディレクトリへの絶対パスを使用する必要があります）。

```
handlers = java.util.logging.FileHandler
java.util.logging.FileHandler.pattern = <REPLACE THIS PATH
FIRST>/jdbc%u.log
java.util.logging.FileHandler.limit = 10000000
java.util.logging.FileHandler.count = 90000
java.util.logging.FileHandler.formatter =
oracle.ucp.util.logging.UCPFormatter
.level = WARNING
oracle.jdbc.internal.replay.level = FINEST
oracle.ucp.jdbc.PoolDataSourceImpl.level = FINE
oracle.ucp.jdbc.oracle.level = FINEST
```

Oracle RDBMS サーバーでのトレースは、init.ora ファイルを介し、次のように指定して有効化できます。

```
alter system set event='10602 trace name context forever, level
28:trace[progin_t_appcont_rdbms]:10702 trace name context forever, level 16'
scope = spfile ;
```

統計監視 Bean は、次の場所の ucpdemos.jar サンプル・コードに含まれています。

<http://www.oracle.com/technetwork/database/features/jdbc/jdbc-ucp-122-3110062.html>:

MBean について詳しくは、  
コード・サンプル `UniversalConnectionPoolManagerMBeanSample.java` および  
`UniversalConnectionPoolManagerSample.java` を参照してください。

## その他の資料

アプリケーション・コンティニューイティの OTN ホームページ

<http://www.oracle.com/technetwork/jp/database/options/clustering/applicationcontinuity/overview/index.html>

FAN ホワイト・ペーパー

<http://www.oracle.com/technetwork/database/options/clustering/applicationcontinuity/learnmore/fastapplicationnotification12c-2538999.pdf>

アプリケーション・コンティニューイティ・ホワイト・ペーパー

<http://www.oracle.com/technetwork/jp/database/options/clustering/application-continuity-wp-12c-1966213-ja.pdf>

アプリケーション・コンティニューイティ OOW 2017

<http://www.oracle.com/technetwork/database/options/clustering/applicationcontinuity/learnmore/hidingunplannedoutages-2872659.pdf>

クライアント・フェイルオーバー



<http://www.oracle.com/technetwork/database/options/clustering/applicationcontinuity/learnmore/client-failover-brief-2430007.pdf>



**Oracle Corporation, World Headquarters**  
500 Oracle Parkway  
Redwood Shores, CA 94065, USA

海外からのお問い合わせ窓口  
電話：+1.650.506.7000  
ファクシミリ：+1.650.506.7200

#### CONNECT WITH US

-  [blogs.oracle.com/oracle](https://blogs.oracle.com/oracle)
-  [facebook.com/oracle](https://facebook.com/oracle)
-  [twitter.com/oracle](https://twitter.com/oracle)
-  [oracle.com](https://oracle.com)

## Integrated Cloud Applications & Platform Services

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. 本文書は情報提供のみを目的として提供されており、ここに記載される内容は予告なく変更されることがあります。本文書は、その内容に誤りがないことを保証するものではなく、また、口頭による明示的保証や法律による黙示的保証を含め、商品性ないし特定目的適合性に関する黙示的保証および条件などのいかなる保証および条件も提供するものではありません。オラクルは本文書に関するいかなる法的責任も明確に否認し、本文書によって直接的または間接的に確立される契約義務はないものとします。本文書はオラクルの書面による許可を前もって得ることなく、いかなる目的のためにも、電子または印刷を含むいかなる形式や手段によっても再作成または送信することはできません。

Oracle および Java は Oracle およびその子会社、関連会社の登録商標です。その他の名称はそれぞれの会社の商標です。

Intel および Intel Xeon は Intel Corporation の商標または登録商標です。すべての SPARC 商標はライセンスに基づいて使用される SPARC International, Inc. の商標または登録商標です。AMD、Opteron、AMD ロゴおよび AMD Opteron ロゴは、Advanced Micro Devices の商標または登録商標です。UNIX は、The Open Group の登録商標です。0618

アプリケーション・コンティニューイティ 準備のためのチェックリスト 2018 年 6 月

著者：Troy Anthony

共著者：Carol Colrain、Carmen Frank



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