

Oracle On Demand のベスト・プラクティス、Critical Patch Update

Oracle ホワイト・ペーパー
2008 年7月

| | |
|------------------------|----|
| はじめに | 3 |
| 概要 | 3 |
| 概要..... | 3 |
| 目標..... | 4 |
| 参照資料..... | 5 |
| CPU について..... | 5 |
| CPU のコンテンツ | 6 |
| CPU の実装フェーズ..... | 6 |
| 計画..... | 8 |
| テスト配置..... | 9 |
| 本番配置..... | 10 |
| CPU 実装の最適化..... | 12 |
| 環境チェッカー・ツール..... | 12 |
| テスト戦略..... | 12 |
| Cookbook | 13 |
| アプリケーション・パッチのマージ | 13 |
| 共有の製品ホーム..... | 13 |
| 適切なツールの選択..... | 13 |
| 予防型の保守機能..... | 14 |
| 結論 | 14 |

このホワイト・ペーパーは、Oracle システムや環境の変更を計画し、実行する IT 管理者や上級スタッフを対象としています。

はじめに

このホワイト・ペーパーは、Oracle システムや環境の変更を計画および実行する IT 管理者と上級スタッフを対象としています。

このホワイト・ペーパーでは、Oracle 製品を使用している企業ユーザーが Oracle Critical Patch Update を実装する際のベスト・プラクティスについて説明します。概要に続く最初の項では、Critical Patch Update (CPU) に関する一般的な情報を示します。次の項では、CPU 実装のフロー・ダイアグラムを示し、続いて CPU の計画、テスト、および配置の詳細について説明します。そのあとの項では、プロセスの可能な最適化、およびその利点について説明します。

概要

概要

Oracle Critical Patch Update は、オラクル製品のセキュリティ修正プログラムを提供するために四半期ごとにリリースされるパッチのバンドルです。

Oracle Critical Patch Update は、オラクル製品のセキュリティ修正プログラムを提供するために四半期ごとにリリースされるパッチのバンドルです。CPU の実装は、それぞれ 3 つの重要なフェーズで構成されます。

- 評価と計画
- テスト配置
- 本番配置

次のダイアグラムは、これらのフェーズの相互関係を示しています。

CPU の詳細については、OracleMetaink にある以下のドキュメントを参照してください。

- Oracle CPU FAQ
- Oracle CPU Program General FAQ
- FAQ for Oracle Security Alerts and CPU



このホワイト・ペーパーでは、これら 3 つのフェーズ、詳細なフロー・ダイアグラム、および各フェーズのヒントについて説明します。また、IT 組織が CPU を効率的に実装するために利用できる最適化のガイドラインも示します。CPU の実装を最適化するためのガイドラインは、次のとおりです。

- 複数の中間層インストール間で製品ホームを共有して、それぞれの CPU を 1 度だけ適用する。
- 複数の Oracle E-Business Suite パッチを 1 つのバンドルにマージする。
- 正しいテスト戦略を作成して、CPU 実装時のシステムの整合性を確保する。
- ユーザーの IT 環境用にカスタマイズされた、CPU 実装の Cookbook を作成する。
- CPU のサポートを最適化するために環境を予防的に保守する。
- 実装前後のヘルス・チェックのための適切なツールとプロセスを開発する。

これらのポイントについては、CPU 実装の最適化の項で詳しく説明します。

Oracle On Demand は、CPU のリリース後数日のうちに、数百の顧客に CPU を実装します。このホワイト・ペーパーに記載されている推奨事項の多くは、On Demand による CPU の実装経験からオラクルが得たものです。

目標

定期的な CPU の実装は、顧客に以下のような多数の利点をもたらします。

- セキュリティ・アラートなどのように、予告なしに計画外のパッチが通知されるわけではないため、十分な時間を確保できる。
- それぞれの CPU では複数のセキュリティ修正プログラムが 1 つのパッチに集約されているため、通常、コストを削減できる。このため、顧客はワンオフ・パッチを何度も適用する必要も、システムを停止してテストを繰り返す必要もありません。

CPU は、（停止時間と作業にかかる両方の）コストおよび CPU 実装のリスクを最小限に抑えるために、今後も改善され続けます。

Oracle On Demand は、CPU のリリース後数日のうちに、数百の顧客に CPU を実装します。このホワイト・ペーパーに記載されている推奨事項の多くは、On Demand による CPU の実装経験からオラクルが得たものです。このホワイト・ペーパーでは、CPU の実装を最適化するためのいくつかのベスト・プラクティスについて説明します。結果として得られる利点は、次のとおりです。

- 顧客が低コスト低リスクで迅速に CPU を実装できるようになる。
- CPU の計画および実装にかかわる人々のエクスペリエンスが向上する。

- CPU の実装に必要な停止時間の合計時間を短縮することで、ビジネスおよびエンドユーザーに対する CPU 実装の影響を軽減する。

参照資料

Oracle On Demand を利用すると、CPU を迅速に取り込んで、実装を成功させることができます。Oracle On Demand では、このドキュメントに記載されている多くのベスト・プラクティスや推奨事項を適用することでコストを大幅に削減し、さまざまな利点を提供しています。同時に、Oracle On Demand を利用している顧客は、停止時間の短縮、優れた一貫性、および CPU 実装の信頼性も実感しています。

CPU について

2005 年以降、CPU はオラクル製品のセキュリティ修正プログラムをリリースするための主要な手段となっています。

CPU は、オラクル製品のセキュリティ修正プログラムを提供するために四半期ごとにリリースされるパッチのバンドルです。2005 年以降、オラクル製品のセキュリティ修正プログラムをリリースするための主要な手段となっており、四半期ごとに事前に通知された日にリリースされます。CPU は、サポート製品の次のような脆弱性に対応するためにリリースされます。

- データの秘匿性。攻撃者が本来参照できない情報を参照することに関連する。
- データの整合性。攻撃者が本来変更できない情報を変更することに関連する。
- システムの可用性。攻撃者がシステムの正当な使用またはシステムへのアクセスを妨害することに関連する。

CPU は、システムの秘匿性、整合性、または可用性からなる脆弱性を修正します。

オラクルでは、Oracle Database、Oracle Application Server、および Oracle Enterprise Manager のバージョンごとに統合された 1 つのパッチをリリースしています。Oracle Collaboration Suite も同様ですが、修正プログラムはコンポーネント単位で提供されます。可能な場合は、Oracle E-Business Suite の修正プログラムも 1 つのパッチに統合しています。しかし、通常は、それぞれの CPU には、それぞれ Oracle E-Business Suite リリースに対する複数のパッチが含まれます。たとえば、Oracle Database Release 9.2.0.6 および Oracle Application Server Release 9.0.4.1（中間層）に 1 つの CPU パッチをリリースし、Oracle E-Business Suite Release 11.5.10 に複数のワンオフ・パッチをリリースします。PeopleSoft、JD Edwards、および BEA の各パッチも、現在はワンオフ・パッチとして提供されています。

CPU パッチは、データベース製品およびミドルウェア製品に対して累積的なものです。つまり、最新の CPU のデータベース・パッチまたはミドルウェア・パッチには、別途記載がある場合を除き、それまでのすべての CPU の修正プログラムが含まれています。ただし、Oracle E-Business Suite の CPU パッチは累積的ではありません。PeopleSoft および JD Edwards のパッチは、現在は累積的ではありませんが、将来的には累積的なものにする予定です。BEA 製品のセキュリティ・アドバイザ・パッチも累積的ではありません（別途記載がある場合を除く）。各オラクル製品スイートの CPU のドキュメントには、関連のパッチが累積的であるか、追加的であるかが記載されています。

CPU のコンテンツ

それぞれの CPU には、セキュリティ・パッチ、およびドキュメント・ロードマップをはじめとしたサポート・ドキュメントが含まれています。ドキュメント・ロードマップは、システムに対し、正しいパッチと正しいサポート・ドキュメントを選択するために役立ちます。CPU に含まれているドキュメントには、次の 3 つのレベルがあります。

- **レベル 1** : *Oracle CPU Advisory* には、リスク評価のマトリクスをはじめとしたセキュリティ修正プログラム関連の情報が含まれています。これらのマトリクスには、新しいセキュリティ修正プログラムによって影響を受けるコンポーネントのリストがあります。2006 年 10 月、オラクルでは、リスク・マトリクスにおいてセキュリティの脆弱性の相対的な重大度を示す方法を独自の方式から **Common Vulnerability Scoring System (CVSS)** に切り替えました。*Oracle CPU Advisory* には、影響を受ける製品の一覧、サポートされなくなった製品または以前の製品リリースのパッチに関する情報、セキュリティの脆弱性に関する追加情報に対するリクエストのオラクルでの取扱い方法が記載されています。

- **レベル 2** : *Critical Patch Update Availability Information for Oracle Server and Middleware Products* には、Oracle Server Technology 製品の環境設定、最小要件、およびパッチの可用性に関する情報が含まれています。Oracle Server Technology 製品には、Oracle Database、Oracle Application Server、Oracle Collaboration Suite、および Oracle Enterprise Manager Grid Control が含まれます。

E-Business Suite Critical Patch Update Note には、Oracle E-Business Suite 環境に適用可能な Oracle Database、Oracle HTTP Server、Oracle Developer Suite、Oracle JInitiator、および Oracle E-Business Suite に関するパッチ情報が記載されています。また、既知の問題も含まれています。

- **レベル 3** : *Bundled Patch README* には、特定のパッチを適用するための指示書をはじめ、個別のパッチが付属しています。

Critical Patch Update - July 2006 以前は、レベル 2 のドキュメントはプレインストールと呼ばれていました。

一覧表示されているパッチは、Oracle 環境に適用可能です。

CPU の実装フェーズ

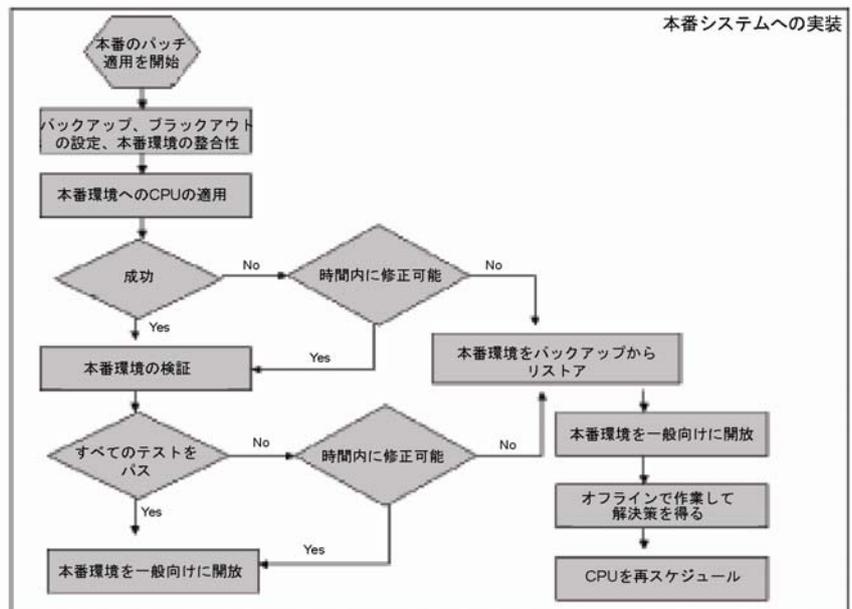
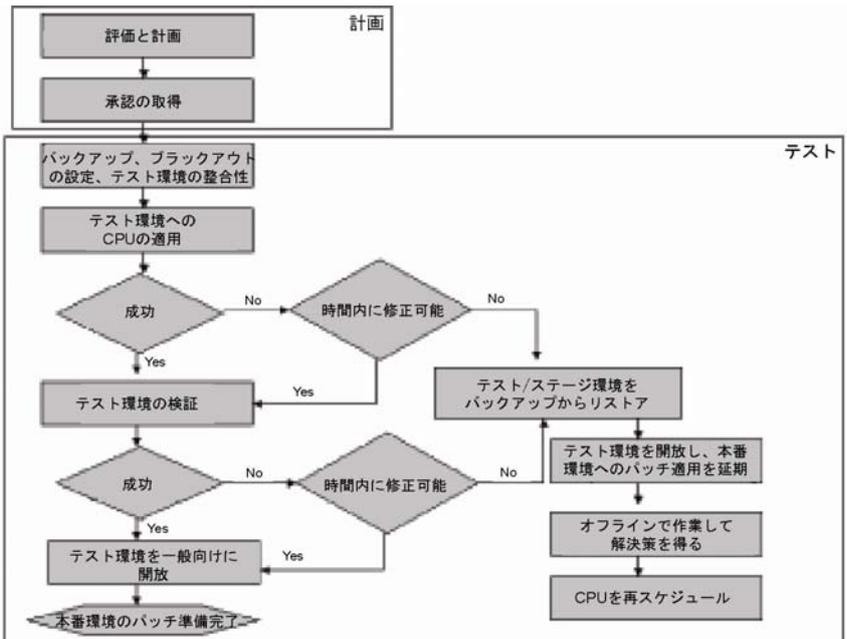
CPU の実装プロセスは、次の 3 つのフェーズから成ります。

- 計画
- テスト配置
- 本番配置

次のフロー・ダイアグラムは、各フェーズのおもな手順を示しています。後続の項では、これらの手順とフェーズについて説明します。

高レベルの CPU の実装プロセスは、次の 3 つのフェーズに分けられます。

- 計画
- テスト配置
- 本番配置



計画

CPU の IT 環境への実装は、そのほかのパッチ・セットの実装と似ています。実装を成功させるには、適切な変更管理システムとプロセスを準備する必要があります。

評価

CPU の実装を正常におこなうためには、以下の作業をおこなう必要があります。

- 適切な変更管理システムとプロセスを準備する。
- 主要な参加者と CPU の実装について話し合い、計画を立てる。

初期の計画フェーズでは、CPU で対処するセキュリティ・フローの重要性を評価し、必要な承認を得るために、その重要性を経営陣やそのほかの関連部門に報告する必要があります。さらに、システムに優先順位をつけて、パッチを適用する順番を決めることも重要です。次のガイドラインに従う必要があります。

- 企業内にあるすべてのオラクル製品（フル・バージョン番号を含む）の完全なインベントリを手元に用意します。
- Oracle CPU Advisory を確認して、パッチの重要性を評価し、必要な承認を得るための概要を得ます。オラクルでは、CPU で新たに修正されているそれぞれの脆弱性に対して CVSS メトリックの値を提供し、脆弱性を不当に活用するために必要な前提条件、ターゲット・システムへの秘匿性、整合性、および可用性に関する攻撃の影響を示しています。MetaLink Note 394487.1 には、CVSS の評価が CPU のドキュメントにどのように適用されているかに関する詳細な説明があります。
- CPU Availability または Update Notes を確認して、パッチを適用する必要がある環境またはシステム、および事前に適用する必要があるパッチを特定します。さらに、特定の製品の Known Issues のドキュメントを参照して、アップデート、問題、およびソリューション/回避方法を確認します。

計画

変更をおこなう際には、主要な参加者と CPU の実装について話し合い、計画を立てる必要があります。そうすることで実装を円滑におこなえます。また、リスクを評価し、コンティンジェンシー・プランを作ることも重要です。次のガイドラインを確認してください。

- Oracle MetaLink から必要なパッチを中央のストレージ・システムにダウンロードします。このストレージ・システムからは、CPU のパッチ適用のためにターゲット・システムに読み取り専用アクセスがおこなえる必要があります。
- すべての環境の完全なバックアップを取得し、予期せぬ障害の発生時に、割り当てられた停止時間内にシステムを効率的にリストアするためのコンティンジェンシー・プランを作成します。
- テスト環境とステージ環境は可能な限り本番環境に近いものにします。このことは、パッチ・セットとワンオフ・パッチを含む製品レベルではとくに重要です。そうすることで、まず非本番環境で CPU を効果的にテストできます。

- CPU の実装後にテストする必要がある、各システムの主要な機能または領域を特定します。総合的なテスト計画またはテストする項目のチェックリストを作成すると役立ちます。
- テスト配置と本番配置は、少なくとも 1 日空けるようにします。1 日空けることで、予測していなかった問題に対応してから本番環境に実装できます。
- CPU の実装に必要な停止時間に対する承認を事前に取得します。

承認

停止時間は、ビジネスおよびユーザーに対する影響がもっとも少ない時間帯を選択することが重要です。

CPU の実装に必要な停止時間に対する承認を事前に取得することは、計画における重要な手順の 1 つです。Oracle では、CPU で対応している各製品のセキュリティ上の欠陥に関する高レベルの概要を提供しています。この概要には、CPU で対応している脆弱性に関する説明が「わかりやすく」記述されています。これは、初期の承認を得るための経営陣への説明に使用できます。影響を受けるすべてのユーザー・コミュニティのメンバーに、かなり余裕をもって計画停止時間を伝えるようにします。停止時間は、ビジネスおよびユーザーに対する影響がもっとも少ない時間帯を選択することが重要です。たとえば、テスト環境やステージ環境には平日にパッチを適用し、本番インスタンスの計画停止時間は週末に設定します。

CPU は四半期ごとにリリースされます。適切な評価、レビュー、および承認プロセスがまだ存在しない場合は、それらを変更管理システムの一部として確立することを推奨します。

テスト配置

CPU をテストする場合。

- パッチを適用する必要がある領域を特定する。
- テスト・システムを可能な限り本番システムに近づける。

Oracle では、それぞれの CPU をテスト配置で徹底的にテストしてから、本番システムに適用することを推奨しています。多数の本番アプリケーションとシステムを備えている大きな組織では、本番ロールアウトの初期フェーズは、テスト・プロセスの追加コンポーネントだと考えることができます。ただし、これはテスト環境におけるテストに置き換えるべきではありません。

本番の準備テストを効果的におこなうには、テスト環境を可能な限り本番環境に近づけます。CPU は、環境に適用されるパッチ・セット・レベル、ファミリー・バック、およびワンオフ・パッチに依存します。テスト環境に適用されるそれらのパッチ・セット・レベル、ファミリー・バック、およびワンオフ・パッチが本番環境に適用されるものと一致することを確認してください。必要に応じて、テスト環境を本番環境の内容で更新してから CPU をテストします。

Oracle のパッチは、Oracle Universal Installer インベントリを使用して、競合を検出して、そのあとのレコードを移入します。そのため、Oracle パッチは、その大部分を Oracle Universal Installer インベントリの正確さと整合性に依存しています。

CPU をテスト環境に適用する前に、必要な機能テストを実行して、環境が予想どおりに動作するのを確認することを推奨します。そうすることで、既存の問題と CPU によって生じた問題を区別できます。

CPU をテスト環境に適用する前には、いくつかの簡単なテストを実行して、環境が予想どおりに動作するのを確認することを推奨します。

CPU を適用中になんらかの競合が発生し、その競合が無視できるものとして記載されていない場合は、パッチの適用を中止して、その競合を Oracle Support に報告する必要があります。

パッチの適用でエラーが発生しなかった場合は、オラクル・テクノロジー製品の `opatch` コマンドを使用して `OraInventory` をチェックし、CPU パッチが正常に適用されたことを確認します。さらに、Oracle E-Business Suite パッチの `ad_bugs` 表または `ad_applied_patches` 表も確認します。その後、必要なテストを実行して、環境が以前と同じように機能することを確認する必要があります。この配置後の検証タスクは、後に本番環境で予期しない事態や損害が生じることを避けるために同じく重要なタスクです。

本番配置

CPU を実装するための一般的なヒントをいくつか紹介します。

- 計画停止時間の予告ページにすべてのターゲット・システムが含まれていることを確認してから、パッチ適用のためにシステムを停止します。必要な停止時間は、CPU や環境ごとに異なります。アップデートする本番システムが多数ある場合は、特定の CPU に関する情報を得るにしがたい、概算時間を調整できるようにします。
- 本番環境の完全なバックアップがあり、予期せぬ障害が発生した場合のコンティンジェンシー・プランとしてリストア戦略が作成されていることを確認します。
- `Opatch` または `Oracle Universal Installer` コマンドを使用して、`Oracle Universal Installer` インベントリの正確さと整合性を検証します。CPU パッチの適用中に `Oracle Universal Installer` インベントリを無視する場合、パッチの競合は検出されず、気づかぬうちに以前の修正プログラムを上書きしてしまうことがあります。`Oracle Universal Installer` インベントリの詳細については、ホワイト・ペーパーの『[Inventory, see the Understanding and Automating Operations around Oracle Installer Inventory](#)』を参照してください。
- 潜在的なパッチの競合は、次のコマンドを実行することで事前に検出できます。

```
opatch apply -silent -no_bug_superset -report
```

`-report` オプションを指定すると、パッチを適用せずに競合を検出できます。

- `Opatch` を使用してパッチを適用する場合、環境内にあるすべてのサブセット・パッチがロールバックされます。デフォルトでは、各サブセット・パッチはロールバック後に依存バイナリまたはライブラリを再リンクします。たとえば、特定のパッチが 5 つのパッチに代わり適用される場合、ロールバック・アクション中に再リンクが 5 回おこなわれます。この再リンク・アクションには、追加の時間が必要です。ただし、`opatch` には `-norelink` オプションがあり、ロールバック中の再リンクを回避で

きます。このオプションを使用する場合は、`opatch` 完了後にパッチで提供されている `makefile` コマンドを使用して、手動で再リンクをおこなう必要があります。たとえば、パッチ 4560421 の場合、コマンドは次のようになります。

```
$> cd $ORACLE_HOME/.patch_storage/4560421  
  
$> sh 4560421_make.txt | tee 4560421_relink.log
```

- Oracle E-Business Suite の環境は、`adpatch` ユーティリティを使用してパッチを適用します。以下は、パッチを適用する際の時間短縮に役立つ `adpatch` のオプションです。

`noautoconfig` オプションは、パッチ後の `Autoconfig` をスキップします。`Critical Patch Update` の複数のパッチを適用する場合、パッチのあとで毎回 `Autoconfig` を実行する代わりに、最後のパッチ適用のあとにだけ実行することができます。

`nocompilejsp` オプションは、パッチ中の `JSP` のコンパイルをスキップします。`JSP` のコンパイルをページがアクセスされる時まで延期でき、停止時間を短縮できます。

`nomaintainmrc` オプションは、パッチ中の `Multiple Reporting Currency (MRC)` アップデートをスキップします。環境内で `MRC` を設定している場合は、`adadmin` ユーティリティを使用することで、すべてのパッチ適用後、1 度アップデートをスキップできます。

- 本番環境を停止する前に、必要なテストまたはヘルス・チェックを実行して、本番環境の整合性および状態を確認します。
- 中央のストレージからターゲット・サーバーのローカル・ストレージ領域に必要なパッチをコピーします。パッチの適用中にパッチがパッチ・ステージ領域に書き込まれる際、**Oracle** ユーザーまたはグループは、ローカル・パッチ・ステージ領域への書き込み権限を保持している必要があります。
- 該当する `CPU Availability` または `Update Notes` を慎重に確認して、パッチ間の依存関係と順番を理解する必要があります。
- 実装前後に必要な手順、およびパッチの `README` に記載されているパッチ適用の指示に従います。
- パッチの出力ログをスキャンして、エラーがないことを確認します。
- エラーか障害がある場合は、`OracleMetaLink` にある製品の `Known Issues` ドキュメントを参照して、それが既存の問題であるかどうかを確認して、ドキュメントに記載されているソリューション/回避方法を実施します。
- パッチの検証手順（提供されている場合）に従い、パッチが正常に適用されていることを確認します。

- CPU を正常に適用したら、環境の機能を検証します。チェックリストに従い、事前に指定した機能テストをパッチ環境で徹底的に実行します。

CPU 実装の最適化

CPU の実装を最適化するには、以下のツールと戦略が役立ちます。

- 環境チェッカー・ツール
- テスト戦略
- Cookbook
- マージ済みのアプリケーション・パッチ
- 共有の製品ホーム
- Oracle Enterprise Manager
- 予防型の保守機能

オラクルでは、CPU の実装に伴うコストとリスクを最小限に抑えています。しかし、自社の IT ニーズに合わせてカスタマイズしたプロセスとツールを使用することで、CPU の実装をさらに最適化できます。

以下の項には、労力と停止時間の両方の観点から CPU の実装コストを削減するために役立つ推奨事項（順不同）が含まれています。アップデートする Oracle システムが多数ある場合、これらの推奨事項により、さらなる利点が得られます。

環境チェッカー・ツール

CPU の実装を成功させるには、ターゲット環境の整合性を取ることが重要になります。整合性が取れていない場合、CPU の実装中に CPU とは無関係なデバッグの問題が生じることがあります。以下のガイドラインは、環境の整合性を検証するのに役立ちます。

- `opatch lsinventory` コマンドを使用して、パッチの Oracle Universal Installer インベントリをチェックします。
- Oracle E-Business Suite インスタンスの場合、カスタマイズされたすべての構成で自動構成が有効になっていることを確認します。`adchkcfg.sh` コマンドを使用して環境内の構成のカスタマイズ内容を特定し、カスタム・ディレクトリに適切なテンプレート・ファイルを作成して、それらをベースラインとすることができます。そうすることで、デフォルトで自動構成を起動する特定のパッチにより、パッチの適用中にカスタマイズ情報が上書きされるのを防ぐことができます。
- CPU を適用する前に、データベース内のすべての無効なオブジェクトのリストを取得します。CPU を実装後、CPU によって無効なオブジェクトが生成されているかどうかを確認できます。
- 必要な機能テストを実行して、環境の機能がすべて正常なことを確認してから、CPU を実装するためにシステムを停止します。

これらのチェックをすべて自動化すると、コストと人的エラーを削減でき、CPU 実装の成功率も高くなります。

テスト戦略

それぞれの CPU は、製品の動作、ユーザー・エクスペリエンス、ユーザーのカスタム・アプリケーションには（影響がある場合も）最小限の影響しか及ぼしません。そのため、ユーザーは投資効果を最大限に引き出す生産的なテストに集中できます。

これは、効率的な変更管理システムにおけるもっとも重要で難しい手順の 1 つです。明示的に記載されている場合を除き、CPU で機能や製品 API が変更されることはありません。CPU は、製品の動作、ユーザー・エクスペリエンス、カスタム・アプリケーションには（影響がある場合も）最小限の影響しか及ぼしません。そのため、ユーザーは投資効果を最大限に引き出す生産的なテストに集中できます。システムの健全性を保証する正しいテストのセットを特定し、可能ならばそれらを自動化すると、CPU の実装効率を高めることができます。

Cookbook

CPUの実装には、通常、個々のパッチの付属ドキュメント（CPU Availability または Update Notes、および README）を読む作業が含まれます。これらのドキュメントは、一般的な読者を対象に書かれたもので、ユーザーの IT 環境には適用できない項が含まれている場合があります。

自分の環境に適用可能なドキュメントの部分を特定し、それらを使用して、初期の CPU の分析およびテスト中に Cookbook を作成できます。それぞれの IT 環境には、一定の順序でシステムを停止するなどの特定のパッチ前の手順、機能の検証などのパッチ後の手順など、独自の手順が存在します。Cookbook には、これらの手順やそのほかのカスタム手順を含めることができます。これは、CPU でアップデートする同一のシステムが多数ある場合、とくに有用です。Cookbook を作成することで、CPU の実装にかかる時間や労力を削減できるだけでなく、CPU の実装の信頼性と予測可能性も増します。

Ad Merge Patch ユーティリティの詳細については、『Oracle Applications Maintenance Utilities』を参照してください。

アプリケーション・パッチのマージ

Ad Merge Patch ユーティリティ (admgrpch.sh) を使用して、複数の Oracle E-Business Suite のパッチを1つのバンドルにマージします。同一バージョンの Oracle E-Business Suite 環境が複数ある場合は、このマージをおこなうことを強く推奨します。マージすることで、パッチを適用するのに必要な停止時間と労力を削減できます。

共有の製品ホーム

可能な場合は、複数ノードの配置に共有の製品ホームを使用することを検討してください。たとえば、特定の環境で、Oracle Applications 製品ホームの APPL_TOP を複数の同一アプリケーションの中間層で共有すると、パッチの適用時間を大幅に短縮でき、アプリケーション・パッチの適用も1度だけで済みます。オラクルは最近、共有アプリケーション・ファイル・システムのサポートを導入しており、APPL_TOP、iAS ORACLE_HOME、および Developer6i ORACLE_HOME の共有が含まれています。これにより、複数の中間層環境に CPU を実装するのに必要な労力を大幅に削減できます。

共有製品ホームは、パッチの適用というよりアーキテクチャに関する決定であり、徹底的に評価してからこのアーキテクチャを導入する必要があります。

適切なツールの選択

IT 企業にはそれぞれ固有のニーズがあり、すべての企業のニーズを満たす単一のツールは存在しません。ただし、CPU を実装するために、既存の管理インフラストラクチャを活用し、管理ツールを変更するのは大事なことです。たとえば、Oracle Enterprise Manager には、企業内の Oracle 環境を管理するための一元管理コンソールが用意されています。この管理コンソールを使用して、以下のことができます。

Oracle Enterprise Manager の詳細については、Oracle Enterprise Manager の Web サイトを参照してください。アドレスは次のとおりです。
http://www.oracle.com/enterprise_manager/index.html

- 複数のホストに計画停止時間を伝える。
- Enterprise Manager の豊富なレポート機能を使用して、適用されているパッチと適用されていないパッチを検出し、環境に適用可能なパッチを特定する。
- ジョブ・システムを使用して、複数のホスト間での CPU パッチの適用をスケジュールする。複数の ORACLE_HOME に同時にパッチを適用します (スケーラブル・パッチング・モデル)。
- ジョブが失敗した場合に、修正アクションをおこない、リトライ機能を使用する。

既存の変更管理ツールまたはシステムに必要な拡張機能を開発して、CPU の実装タスクの一部を自動化することもできます。

予防型の保守機能

Oracle では、通常、過去 2 セット分のパッチ・リリースに対してのみ CPU パッチをリリースします。最新の Oracle パッチ・セットとファミリー・パックを適切なタイミングで適用して、システムを予防的に保守することで、CPU の実装中にそのほかのアップグレードが実行されることを回避できます。

結論

CPU は、定期的に適用するそのほかのパッチ・セットと似ています。ただし、多くの場合、CPU は、事後対応型ではなく、予防型の保守機能です。正しいテクノロジーとプロセスのセットを備えた適切な変更管理システムにより、CPU の実装を定期的なメンテナンス・ジョブにすることができます。

Oracle On Demand では、CPU のリリース後、すぐにこれらのベスト・プラクティスを使用して、数百の On Demand の顧客に CPU を実装します。Oracle On Demand を利用している企業では、このドキュメントに記載されている多くの推奨事項を適用することでコストや停止時間を削減しています。



Oracle On Demand のベスト・プラクティス : Critical Patch Update

2008 年 7 月

著者 : Rajesh Shah

共著者 : Eric Maurice、Gopal Parthasarathay Ajay Srivastava、Niloy Banerjee、Darius Wiles、Debashis Saha、Sudip Datta

Oracle Corporation

World Headquarters

500 Oracle Parkway

Redwood Shores, CA 94065

U.S.A.

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

www.oracle.com

Copyright © 2006, 2008 Oracle. 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、JD Edwards、および PeopleSoft は、米国 Oracle Corporation およびその子会社、関連会社の登録商標です。他の製品名は、それぞれの所有者の商標です。