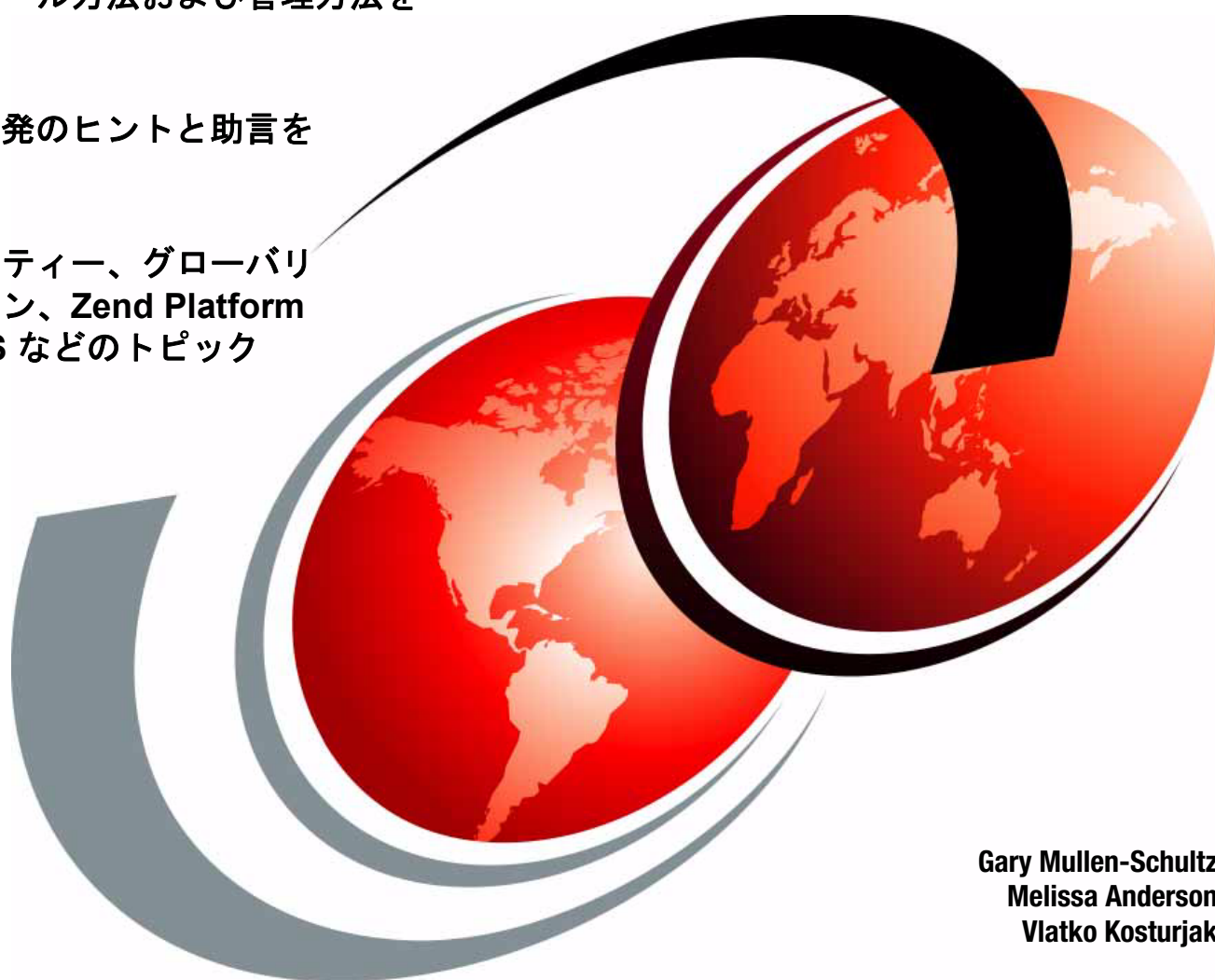


# PHP: Zend for i5/OS

インストール方法および管理方法を  
学ぶ

貴重な開発のヒントと助言を  
得る

セキュリティー、グローバリ  
ゼーション、Zend Platform  
for i5/OS などのトピック



Gary Mullen-Schultz  
Melissa Anderson  
Vlatko Kosturjak

# Redbooks





International Technical Support Organization

**PHP: Zend for i5/OS**

2007 年 5 月

注：本書および本書で紹介する製品をご使用になる前に、ix 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、Zend Core for i5/OS バージョン 1.0 リリース 5.0 モディフィケーション 0.0、Zend Platform for i5/OS バージョン 2.0 リリース 1.0 モディフィケーション 2.0、および Zend Platform for i5/OS バージョン 5.0 リリース 2.0 モディフィケーション 0.0 に適用されます。

本マニュアルに関するご意見やご感想は、次の URL からお送りください。今後の参考にさせていただきます。

<http://www.ibm.com/jp/manuals/main/mail.html>

なお、日本 IBM 発行のマニュアルはインターネット経由でもご購入いただけます。詳しくは <http://www.ibm.com/jp/manuals/> の「ご注文について」をご覧ください。

(URL は、変更になる場合があります)

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：	SG24-7327-00 PHP:Zend for i5/OS International Technical Support Organization
発行：	日本アイ・ビー・エム株式会社
担当：	ナショナル・ランゲージ・サポート

## 第 1 刷 2007.5

この文書では、平成明朝体™ W3、平成明朝体™ W9、平成角ゴシック体™ W3、平成角ゴシック体™ W5、および平成角ゴシック体™ W7 を使用しています。この(書体\*)は、(財)日本規格協会と使用契約を締結し使用しているものです。フォントとして無断複製することは禁止されています。

注\* 平成明朝体™ W3、平成明朝体™ W7、平成明朝体™ W9、平成角ゴシック体™ W3、平成角ゴシック体™ W5、平成角ゴシック体™ W7





# 目次

特記事項	ix
商標	x
前書き	xi
この Redbook を執筆したチーム	xi
<b>第 1 章 PHP on i5/OS へようこそ</b>	1
1.1 ようこそ	2
1.1.1 IBM および Zend Core	2
1.1.2 Zend Core for IBM	2
1.2 これまでの i5/OS での PHP サポート	3
1.3 現在の i5/OS での PHP サポート	3
1.3.1 Zend Core for i5/OS	3
1.3.2 Zend Studio for i5/OS	4
1.3.3 Zend Platform for i5/OS	5
1.4 Zend Core と i5 実装の違い	5
1.5 開発者およびカスタマーにとっての利点	5
<b>第 2 章 PHP の概要</b>	7
2.1 PHP の歴史	8
2.2 PHP を選択する理由	8
2.3 PHP のユーザー	9
2.4 Java との簡易比較	9
<b>第 3 章 管理</b>	11
3.1 パッケージ化	12
3.2 Zend Core for i5/OS のインストール	12
3.2.1 開始する前に	12
3.2.2 Zend Core for i5/OS 製品のインストール	16
3.2.3 Zend Core for i5/OS 環境の開始	18
3.2.4 インストールの確認	20
3.2.5 一般的なインストールの問題	21
3.2.6 Zend Core for i5/OS の再インストール	21
3.2.7 Zend Core for i5/OS のアンインストール	22
3.2.8 製品構成	22
3.2.9 ランタイム環境	24
3.3 Zend Core for i5/OS の構成	27
3.3.1 管理ツール	27
3.4 複数インスタンスの構成	31
3.4.1 複数の i5/OS Apache インスタンス、1 台の PHP サーバー	31
3.4.2 複数の i5/OS Apache インスタンス、複数の PHP サーバー	33
3.4.3 1 つの i5/OS Apache インスタンス、複数の PHP サーバー	35
3.5 Zend Core for i5/OS のバックアップ	37
<b>第 4 章 アプリケーション開発</b>	39
4.1 PHP および Web 開発	40
4.1.1 簡単な Web アプリケーションの例	40
4.2 コマンド行スクリプト言語としての PHP	41
4.2.1 PHP の対話モード	42

4.2.2	PHP 1 行スクリプト (通称 oneliner)	43
4.2.3	PHP とスクリプト記述	43
4.2.4	CL からの PHP 呼び出し	45
4.3	開発ツール	45
4.3.1	Zend Studio for i5/OS	46
4.3.2	Eclipse	47
4.3.3	NuSphere PhpED	48
4.3.4	Maguma	49
4.4	i5 PHP API ツールキットの関数	49
4.4.1	主な関数	50
4.5	XML サポート	53
4.6	XML-RPC サポート	55
4.7	SOAP サポート	56
<b>第 5 章</b>	<b>データベース・アクセス</b>	<b>59</b>
5.1	DB2	60
5.1.1	DB2 アクセス・オプション	60
5.1.2	パフォーマンスに関する考慮事項	70
5.1.3	トラブルシューティング	71
5.2	MySQL	71
5.2.1	MySQL のアーキテクチャー	72
5.2.2	MySQL のディレクトリー構造	72
5.2.3	インストールおよび構成	72
5.2.4	アクセス・オプション	76
5.3	MySQL から DB2 へのマイグレーション	78
5.4	MySQL または DB2 をいつ選択すべきか	79
<b>第 6 章</b>	<b>Java Bridge サポート</b>	<b>81</b>
6.1	概説	82
6.2	Java Bridge の構成	82
6.2.1	主構成ファイル: javamw.rc	82
6.2.2	Java Bridge の状況	82
6.3	サンプル・プログラム	83
6.3.1	シンプル・テスト・プログラム	83
6.3.2	IBM Toolbox for Java および PHP	85
6.4	トラブルシューティングのヒント	87
6.4.1	php_error_log ファイル	87
6.4.2	var_dump() 関数	88
<b>第 7 章</b>	<b>セキュリティ</b>	<b>89</b>
7.1	セキュリティに関する考慮事項	90
7.1.1	SSL 構成	90
7.1.2	ディレクトリー構造へのアクセス	90
7.1.3	リバース・プロキシ	91
7.1.4	ユーザー・プロファイル	92
7.1.5	その他の情報	92
<b>第 8 章</b>	<b>パフォーマンス</b>	<b>93</b>
8.1	パフォーマンス・チューニング	94
8.1.1	ハードウェア	94
8.1.2	i5/OS	95
8.1.3	i5/OS PASE	95
8.1.4	DB2 for i5/OS	95
8.1.5	Apache	95



8.1.6 Zend Core for i5/OS .....	95
<b>第9章 トラブルシューティング</b> .....	97
9.1 始める前に .....	99
9.2 ブラウザーのトラブルシューティング .....	99
9.3 Web サーバーのトラブルシューティング .....	100
9.4 Zend Core のトラブルシューティング .....	101
9.5 アプリケーション・リソースのトラブルシューティング .....	103
9.5.1 オブジェクト権限の処理 .....	103
9.5.2 i5/OS ジョブ・ログの処理 .....	104
9.5.3 スプール・ファイルの処理 .....	104
9.5.4 ヒストリー・ログの処理 .....	105
9.5.5 メッセージ・キューの処理 .....	105
9.5.6 デバッガーの使用 .....	106
9.5.7 Toolkit アクティビティのロギング .....	106
<b>第10章 グローバリゼーション</b> .....	107
10.1 概説 .....	108
10.1.1 PHP 国際化対応 .....	108
10.1.2 i5/OS のレイヤー .....	108
10.2 グローバリゼーションの構成 ( シングルバイト ) .....	112
10.2.1 i5/OS .....	112
10.2.2 i5/OS PASE .....	112
10.2.3 PHP .....	112
10.2.4 Apache .....	113
10.2.5 DB2 for i5/OS .....	113
<b>第11章 高度な開発トピック</b> .....	115
11.1 テンプレートのインプリメンテーション .....	116
11.2 Model View Controller (MVC) 設計およびフレームワーク .....	118
11.3 ヒントおよび秘訣 .....	120
11.3.1 事前定義配列を使用したシステム環境変数へのアクセス .....	120
11.3.2 リバース・プロキシの問題 .....	120
<b>関連資料</b> .....	121
IBM Redbooks .....	121
その他の資料 .....	121
オンライン・リソース .....	121
IBM Redbooks の入手方法 .....	122
IBM からのヘルプ .....	122
<b>略語および頭字語</b> .....	123
<b>索引</b> .....	125



# 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。  
〒106-8711 東京都港区六本木 3-2-12 IBM World Trade Asia Corporation  
Intellectual Property Law & Licensing

**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。

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。

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。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。

## 商標

以下は、IBM Corporation の商標です。

Redbooks (logo) ™  
developerWorks®  
eServer™  
ibm.com®  
iSeries™  
i5/OS®  
z/OS®  
AIX®

Cloudscape™  
DB2 Connect™  
DB2 Universal Database™  
DB2®  
DRDA®  
Informix®  
IBM®  
POWER™

Rational®  
Redbooks™  
System i™  
System i5™  
System z™  
WebSphere®

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。

Java、JavaScript、JDBC、J2EE、Sun およびすべての Java 関連の商標およびロゴは、Sun Microsystems, Inc. の米国およびその他の国における商標または登録商標です。

Internet Explorer、Microsoft、Windows、および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

UNIX は、The Open Group の米国およびその他の国における登録商標です。

Linux は、Linus Torvalds の米国およびその他の国における商標です。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。

# 前書き

この IBM® Redbook は、Zend Core for i5/OS® および Zend Platform for i5/OS を使用し、System i™ で PHP をインストール、構成するため、および生産性を向上するために有用です。本書では、背景情報や他のツールとの比較も示してあるため、PHP を評価する場合にも有用です。

この他、以下のようなトピックがあります。

- ▶ インストール、構成、および管理
- ▶ アプリケーション開発
- ▶ PHP アプリケーションから DB2® for i5/OS および MySQL のデータにアクセスするためのストラテジー
- ▶ セキュリティー、パフォーマンス、トラブルシューティングなど

本書では、すべてのプラットフォームでの汎用的な PHP 機能よりも、i5/OS 固有の機能に重点を置いてあります。

## この Redbook を執筆したチーム

本書は、International Technical Support Organization、Rochester Center で働く各国の専門家のチームによって製作されました。

**Gary L. Mullen-Schultz** は、ITSO、Rochester Center のコンサルティング IT スペシャリストです。彼は、Blue Gene/L 資料の製作チームのリーダーを務め、PHP、Java™、WebSphere® など i5/OS アプリケーションの開発に注力しています。彼は、Sun™ Certified Java Programmer、Developer、および Architect であり、3 件の交付済み特許を所有しています。

**Melissa Anderson** は、IBM Rochester System i Technology Center (iTC) e- ビジネス・チームの元メンバーであり、System i での Web テクノロジーについての教育やサポートを行っています。iTC で働く以前は、System i のさまざまな Java 製品および Web 製品を対象とする Web サイト開発者でありテクニカル・ライターでした。彼女は、複数の COMMON 教育コンファレンスでスピーカーとしてメダルを受賞しています。彼女は、Winona State University においてコンピューター・サイエンスの学位をまもなく取得するところです。

**Vlatko Kosturjak** は、IBM Croatia の IT セキュリティー・スペシャリストです。彼は、道徳的ハッキング、IT 監査、および国際セキュリティ標準に基づく ISMS の実装を経験しています。彼は、PHP、mod\_perl、およびクリティカルな Web 開発に関してセキュリティ重視の観点からさまざまな著述を行っています。彼は、多種多様なセキュア Linux® ソリューションの開発を経験した、LPI 認定の Linux スペシャリストです。Vlatko には、PDA から System z™ まで、ほとんどすべてのプラットフォームにおける開発、移植、オープン・ソース・ソフトウェアへの貢献など、Linux に関する豊富な経験があります。

プロジェクトに協力してくださった次の方々に感謝します。

Jim Bainbridge  
Tony Cairns  
Craig Johnson  
Kent Milligan  
David Peraza

Michael Sandberg  
Steve Will  
IBM Rochester

Whei-Jen Chen  
ITSO San Jose

本書を精読し、熟慮したコメントを寄せ、貢献してくれた Alan に特に感謝します。

Alan Seiden  
Strategic Business Systems, Inc.



# PHP on i5/OS へようこそ

本章では、PHP on i5/OS で新しくサポートされた機能について紹介します。

## 1.1 ようこそ

Zend for i5/OS は、Zend 社と IBM がパートナーとなってお届けする、完結した、オールインワン PHP 開発環境兼実稼働環境であり、革新的な i5/OS プラットフォームと完全に統合されます。Zend for i5/OS は、Zend Core、Zend Studio、および Zend Platform により構成され、最新型 Web アプリケーションの開発と管理に関する完全なソリューションを System i ビジネスに提供します。Zend for i5/OS は、System i ビジネスでの迅速な Web 活用および Web サービス対応 SOA 統合に向けての、最も創造性に富み、最もコスト効率のよい手段の 1 つです。

### 1.1.1 IBM および Zend Core

IBM と Zend 社は、Zend Core for IBM を発表しました。Zend Core for IBM は、広く普及した Web 言語である PHP に基づくソリューションであり、データベース・アプリケーションおよびサービスを展開する開発者の支援に特化して設計された業界初の統合ソリューションです。IBM および Zend 社は、高水準データベース統合フレームワークの向上および PHP Web サービス標準の拡張を目指す PHP テクノロジーの追求にも、共同して取り組むことを発表しました。同時に、Zend 社は、企業品質のアップデートおよびサポートを PHP 開発者に提供する、包括的なサービス・システム兼サポート・システムである Zend Network を発表しました。Zend Core for IBM は、The Apache Software Foundation の Derby プロジェクトをベースとした IBM DB2 Universal Database™ や IBM Cloudscape™ を Zend 社のオープン・ソース PHP 環境と統合します。Zend Core for IBM と Zend Network を組み合わせることにより、シームレスで、すぐに使用できる、企業グレードの PHP 環境が手に入り、PHP アプリケーションの開発と展開が容易になります。

Zend Core for IBM は PHP 5 テクノロジーに基づいており、Cloudscape データベース・サーバーおよび DB2 との緊密な統合や、XML および Web サービスのネイティブ・サポートを含みます。これにより、サービス指向アーキテクチャ (SOA) への適合性が向上します。Zend Core を使用すると、PHP アプリケーションで、(DB2 Connect™ を使用する System i および System z の環境など) すべての DB2 サーバー環境を使用できます。Zend Core は、ハイパフォーマンスであり、確かな統合機能を提供する一方で、インストール処理および構成処理をシームレスにすることおよび自動化することにより、自動化と展開コストの削減に役立ちます。さらに、Zend Core は一貫した API を開発者に提供しているため、使用が容易で軽量な Cloudscape データベースから、使用が容易で堅固な DB2 Universal Database プラットフォームへのアップグレード・パスも示します。

Zend Core for IBM の開発計画は 2006 年 2 月に発表されました。契約条項に基づき、IBM は、Zend 社と共同で、シームレスですぐに使用できる、業界初の PHP データベース・アプリケーション開発兼実稼働環境となる製品を開発しました。この過程において、Zend Core for IBM に対して実行されたバグ修正およびセキュリティ修正は、すべて PHP コミュニティーとも共有され、PHP コード・ベース全体の向上に貢献しました。

### 1.1.2 Zend Core for IBM

Zend Core はオープン・ソースの PHP エンジンの強化版です。Zend Core は、DB2、MySQL などのデータベースを扱うために必要なすべてのドライバーおよびサード・パーティー・ライブラリーを備えています。Zend Core のインストール・ユーティリティを使用すると、ソフトウェアのインストールが容易になり、高速化されます。Zend Core は XML および Web サービスのネイティブ・サポートを備えており、データベース主導型アプリケーションの高速な開発と展開の基礎となります。

Zend Core には、以下のような特長があります。

- ▶ PHP 5
- ▶ 簡単インストール



- ▶ Cloudscape サーバーおよび PHP 環境に対応した Web ベースのグラフィカル管理コンソール
- ▶ エンタープライズ・データベース Web アプリケーションの展開に使用する、IBM Cloudscape サーバーおよび DB2 ドライバーのバンドル
- ▶ アクセスの容易なドキュメンテーション
- ▶ Zend 社によるサポート・オプションの提供

IBM と Zend 社は、PHP 環境の開発とサポートで協力しています。このコラボレーション (共同作業) により、IBM のオープン・ソース・コミュニティに対するコミットメントが強化されます。IBM は、DB2 および Cloudscape に新しく導入された PHP 用拡張モジュールや、最適化された拡張モジュールを PHP コミュニティに提供し、これらを Zend Core for IBM に組み込んでいます。

このことは、開発者によるアプリケーションの開発と展開の効率を向上するために役立ち、IBM のオープン・ソース投資が生きることになります。PHP は、世界でもっとも普及した Web プログラミング言語の 1 つであるため、Zend Core for IBM を使用できることは、IBM Cloudscape オープン・ソース・データベースを使用する開発者にとって非常に有益です。さらに、徹底的なテストが実行され、そのバグ修正が市場のあらゆる分野のユーザーで構成される PHP コミュニティに通知されるため、Zend Core PHP メンテナンス・プロセスはより高い PHP の安定度と信頼性を確保します。

## 1.2 これまでの i5/OS での PHP サポート

Zend for i5/OS がリリースされる前は、一部の顧客が、オープン・ソースの PHP エンジン移植することにより i5/OS で PHP アプリケーションを実行することに成功している状態でした。実際、2003 年に発行された次の Redpaper には、PHP の移植方法の説明があります。「*Bringing PHP to Your IBM eSeries iSeries Server*」(REDP-3639)。

また、Web サイト <http://www.i5php.net/> には、数年前から、特定のリリースの i5/OS で使用できる、オープン・ソース PHP コードのコンパイル済みバイナリーが公開されています。したがって、PHP サポートは、以前から事実上実現されており、実動環境で使用されている場合もありました。

## 1.3 現在の i5/OS での PHP サポート

Zend 社によって完全スイートが提供されることの主な利点の 1 つに、実動レベルのサポートを受けることができるという点があります。これまで顧客は、真正のインストール・スクリプトや更新スクリプトを利用できない状況のなか、自分自身でバグを修正する必要がありました。非公式のフォーラムのみが、他者の支援を受けられる場所でした。

現在は、Zend 製品で発生した問題について Zend 社から直接サポートを受けられます。これは、PHP アプリケーションを主幹業務的に使用することを希望する大部分の顧客にとって、ずっと望ましい状態です。

### 1.3.1 Zend Core for i5/OS

Zend Core for i5/OS は、i5/OS に固有のリソースおよび機能を持つ、オープン・ソース PHP の機能強化バージョンです。Zend 社から System i 顧客向けに、無料で提供されています。

Zend for i5/OS は、オープン・ソース PHP エンジンを実徹的にテストし、機能強化したバージョンです。Zend for i5/OS には i5/OS 用の PHP ランタイムが含まれており、同梱のセットアップ・アプリケーションにより簡単、高速にソフトウェアをインストールできるようパッケージ化されています。Zend for i5/OS は、Zend Core for i5/OS、Zend Studio for i5/OS、および Zend Platform for i5/OS を含みます。

Zend Core for i5/OS は、i5/OS 固有のリソースおよび機能を利用するよう機能強化されており、Zend 社がフル・サポートする、シームレスな PHP 開発環境兼実稼働環境という成果を提供します。さらに Zend Core for i5/OS は、DB2 for i5/OS と統合します。Zend Core for i5/OS は、SOA アプリケーションの適合性を向上する XML および Web サービスのネイティブ・サポートを備えています。Zend Core for i5/OS は、データベース主導型アプリケーションの高速な開発と展開の土台となります。また、一貫した API を開発者に提供することにより Cloudscape データベースから DB2 へのアップグレード・パスも示します。

Zend for i5/OS の詳細については、次の URL を参照してください。

[http://www.zend.com/products/zend\\_core/zend\\_for\\_i5\\_os](http://www.zend.com/products/zend_core/zend_for_i5_os)

このサイトには、ソフトウェア・ダウンロードの他、Zend Core for IBM に関するデータ・シートやホワイト・ペーパーなどの追加情報もあります。開発者の方は、ぜひ Developer Zone を参照してください。Developer Zone には、チュートリアル、資料、討議フォーラムなどがあります。Developer Zone は、次の場所にあります。

<http://www.zend.com/developers.php>

( 本書の執筆時点ではベータ版である ) Zend Core for i5/OS は、i5/OS V5R3 および V5R4 で の PHP 5.1 をサポートしており、2006 年 7 月 31 日にリリースされました。

Zend Core for i5/OS には、以下のような利点があります。

- ▶ ibm\_db2 拡張モジュールおよび ODBC PHP 拡張モジュールによる IBM DB2 のネイティブ・サポート
- ▶ Zend Network の自動オンライン更新サービス
- ▶ Web 管理 GUI
- ▶ Zend Core for i5/OS セットアップ・ツール
- ▶ PHP ツールキットによる、( 制御言語プログラム、RPG プログラム、データ域など ) 既存のネイティブ i5/OS リソースへのアクセス
- ▶ i5/OS での PHP 5.1.6 および Apache のサポート
- ▶ 簡単インストール
- ▶ Zend 社のフル・サポートを受けた認定 PHP
- ▶ 膨大な数の PHP アプリケーションおよびスクリプトへの対応を可能にする、多数の PHP 拡張モジュール

### 1.3.2 Zend Studio for i5/OS

Zend Studio for i5/OS は、PHP 開発の速度を大幅に向上できる統合開発環境 (IDE) です。

以下のような利点があります。

- ▶ Zend Core for i5/OS の即時オンライン・デバッグおよびエラー修正
- ▶ PHP 5 対応
- ▶ PHP Intelligence によるトラブルシューティング
- ▶ 構文補完
- ▶ DB2 for i5/OS および MySQL データベースを探索する SQL ツール
- ▶ WSDL ファイルの自動作成を備えた、優れた SOAP 対応

- ▶ Zend Core for i5/OS との排他的な連携

### 1.3.3 Zend Platform for i5/OS

Zend Platform for i5/OS は、i5/OS プラットフォームでの PHP アプリケーションを、常に使用でき、高速であり、信頼性が高く、容易に拡張できるようにする、オールインワン実稼働環境です。Zend Technologies 社は全製品をフル・サポートします。

**重要 :** Zend Platform for i5/OS は、Zend Core for i5/OS および Zend Studio for i5/OS のような無償製品ではありません。

Zend Platform for i5/OS のさらなる機能は以下のとおりです。

- ▶ 総合的な PHP アプリケーション内部分析
- ▶ Zend Studio for i5/OS を使用したオンライン・デバッグおよび即時エラー修正
- ▶ ランタイム・コードの最適化
- ▶ PHP/Java Integration Bridge
- ▶ 動的なコンテンツ・キャッシング

## 1.4 Zend Core と i5 実装の違い

Zend Core for i5/OS の主な利点は、PHP アプリケーションの展開とホスティングに関する、極めて安定して堅固なプラットフォームを得たことにあります。実績のある System i ハードウェアと Zend 社の PHP サポートを活用することで、ISV およびカスタマーは、ハードウェアやオペレーティング・システムの挙動の異常やバグの修正に時間を取られることなく、必要なアプリケーションの開発に集中できます。

Zend Core for i5/OS 用の PHP アプリケーションの開発と他のプラットフォームでの PHP アプリケーションの開発に極端に大きな違いはありません。主な違いは、i5/OS 固有のオペレーティング・システム・コマンドの呼び出しおよび高度なデータベース機能の使用に関するものです。

本書の執筆時点では、Zend Core for i5/OS にクラスタリング対応が実装されていなかったことに注意してください。Zend Core for i5/OS の仕様の詳細については、第 4 章『アプリケーション開発』(39 ページ) および第 3 章『管理』(11 ページ) を参照してください。

## 1.5 開発者およびカスタマーにとっての利点

中核業務処理にインターネットを活用する企業が増加するに連れ、カスタマー・サポート機能の向上や、既存のデータ、サービス、および IT インフラストラクチャーのさらなる有効活用に対する要求により、Web サイトに対する要求が高まっています。

PHP と IBM データ・サーバーというプラットフォームは、低コスト、高信頼性、ハイパフォーマンスであり、開発速度が速いため、このプラットフォームを選択する企業が次第に増加しています。

PHP は、CL スクリプト記述や RPG プログラミングの代わりとして使用できます。このことは、CL や RPG の知識が少なく、PHP の知識が豊富な開発者が、i5/OS 用のスクリプトを記述できることを意味します。

Zend Core for i5/OS ユーザーの場合は Zend Studio for i5/OS を無償で入手できることが、Zend for i5/OS ユーザーにとって非常に大きな利点の 1 つになります。





## PHP の概要

本章では、PHP およびその歴史と普及の様子について説明します。Java と比較した場合の長所および短所についても説明し、PHP の短所のいくつかを解決するための一般的な推奨事項を示します。

## 2.1 PHP の歴史

2005 年、PHP コミュニティーは PHP 10 周年を祝いました。PHP はデンマークの Rasmus Lerdorf により作成されました。フォームなど Web サイトの特定部分の自動処理を目的としたため、Personal Home Page/Form Interpreter (PHP/FI) と命名されました。

PHP が生まれる前の開発者は、別のオープン・ソース・スクリプト言語である Perl を使用することが一般的でした。Perl (Practical Extraction and Reporting Language) は、非常に便利ですが、一部の開発者はコードの記述や保守が厄介であると感じました。何年にもわたり、PHP 支持者と Perl 支持者の間で、どちらの言語が優れているのかを突き止めようとする多数の議論が繰り返されています。本書では、それ自体が優れた言語であると言える言語は存在しないと考えるため、この議論には加わりません。ただし、具体的なタスクを考えた場合は、それに適したツールや言語が存在すると考えます。よく引用される比較ですが、ハンマーを使ってねじを壁に打ち込むことはできても、この場合は一般にドライバーのほうが適しているのです。

1997 年に Lerdorf がオープン・ソース PHP/FI ツールの書き直しを行った後で、イスラエルのインターネット・サービス・プロバイダーに勤めるプログラマー Zeev Suraski と彼に学ぶ Andi Gutmans は、PHP/FI スクリプト言語とパーサーを完全に書き直す決心をしました。Rasmus Lerdorf と共同して、Zeev Suraski らは、1998 年に PHP 3 と呼ばれるバージョンをリリースしました。

1998 年に、PHP を使用していたインターネット・ドメインは約 50,000 ですが、これが 100 万ドメインの大台に達するのはまもなくのことでした。しかし Zeev Suraski および Andi Gutmans は、決して満足することなく、3 年半後の 2002 年 5 月、PHP 4 をリリースしました。彼らは、同時に、Zend と呼ばれる新しいエンジンも作成しました。Zend は、PHP を後援する会社の名前になっています。

ただし、Rasmus Lerdorf、Zeev Suraski、および Andi Gutmans のみを PHP の作成者であるとするのは公平ではありません。他の多くのオープン・ソース・プロジェクト同様、PHP には多くの人々の尽力が統合されています。PHP の開発に貢献した人々の名を全て上げることはスペースの関係上無理だと思われるかもしれませんが、PHP のユーザーとして、その尽力に感謝します。

2004 年 7 月に発表された PHP 5 は、Zend エンジンの別の書き直しを意味しました。PHP を使用している Web サイトの数は、2200 万件に達し、現在も増えています。Zend 社では、PHP を使用する開発者の数をおよそ 250 万人であるの見積もっています。

## 2.2 PHP を選択する理由

簡単に説明すれば、次のようになります。PHP は、数十万人または数百万人もユーザーが、何百万という Web サイトで使用しており、アマチュアから Web プロフェッショナルまで、また一家の Web サイトから本格的な商用アプリケーションの作成まで、さまざまな場所で使用されています。このような成功には何かしらの理由があるに違いありません。

PHP は Web 開発用に設計されました。PHP を使用すると、動的に生成される Web ページを簡単に素早く記述すること、ファイルの読み取りや書き込みを行うこと、フォーム・データにアクセスしたり処理したりすること、E メールを送信すること、クッキーの読み取りや書き込みを行うこと、およびセッション変数を維持することなどができます。忘れてならないのが次の長所です。すべての主要なデータベースからのレコードの読み取りおよび書き込みを含む、データベース・レコードへのアクセスの統合です。PHP を使用すると以上のすべての処理を比較的容易に実装でき、特に一部の開発言語の場合と比べて容易であるため、PHP は Web 開発者にとって、効率のよいプログラム言語であるということが出来ます。

さらに、PHP コミュニティーは、フリーで使用可能なコード・サンプル、PHP 拡張モジュール、および多種多様な資料によって、非常に膨大な追加の機能性を生み出しています。(コード・サンプルについては、12,000 個を超える PHP スクリプトが提供されている <http://www.hotscripts.com> または多数の PHP クラスが用意されている <http://www.phpclasses.org> を参照してください。PHP 拡張モジュールについては、<http://pear.php.net> を参照してください。)

また、PHP は、実行速度が速く、安定性があり、拡張性がありながら、システム・リソースの消費は多くありません。このことは、PHP が成功し、現在もさらに普及しつつある理由の 1 つです。この成功した PHP を i5/OS に導入することは、従来の PHP 開発者と、古くからの i5/OS プログラマーの両方にとって意味を持ちます。

## 2.3 PHP のユーザー

既に述べたように、多数の Web サイトで PHP が使用されています。PHP を採用した主要企業としては、Yahoo! (<http://public.yahoo.com/~radwin/talks/yahoo-phpcon2002.htm> の「Making the Case for PHP at Yahoo!」を参照してください)、E チケット・システムに採用した Lufthansa、Sim City Online に採用した Electronic Arts、積載量測定システムに採用した Boeing、および WAP ポータルに採用した Orange があります。

「German PHP Magazin」(<http://www.php-magazin.de>) の 2004 年 5 月号では、Java ベースのコンテンツ・マネージメント・システム (CMS) を PHP ベースの CMS に移行したときの方法が説明されています。もっと多くの証拠が必要な方は、この Redbook を読むことにより、Java ではなく PHP をベースにすることで、企業がはるかに首尾よく大規模なトランザクション・システムを実装できることを理解できます。

## 2.4 Java との簡易比較

PHP は、Web アプリケーションの作成を目的として考案、設計されました。Java は汎用プログラミング言語として設計されました。PHP の構文は、C、Java、および Perl に類似しています。

PHP は、特に小規模から中規模のインストール済み環境では、簡単に使用を開始できます。Java は、大規模な開発と展開を調整する標準装備の管理ツールが必要な大規模チームの場合や、アプリケーションが概して Web ベースでない場合に適しています。両方を使用する場合もあります。この場合は、バックエンドには Java を使用し、フロントエンドには PHP を使用します。これについては Dan Krook による、IBM developerWorks® の優れた記事「Pair J2EE with PHP to implement a common Web application infrastructure」を参照してください。

[http://www.ibm.com/developerworks/websphere/techjournal/0505\\_krook/0505\\_krook.html](http://www.ibm.com/developerworks/websphere/techjournal/0505_krook/0505_krook.html)

PHP と Java は構文や設計の一部分を共有していますが、それぞれに長所と短所があります。PHP には Java と比較して、以下のような長所があります。

### ▶ 使用の容易さ

PHP は HTML に直接埋め込むことのできるスクリプト言語です。これにより、プレゼンテーションとビジネス・オペレーションおよびデータ・アクセスを簡単に組み合わせることができます。ただし、クライアント側のブラウザでスクリプトが実行される JavaScript™ と違い、PHP はサーバーで実行されます。従って、エンド・ユーザーに示される最終的な HTML ソースには PHP のコードが含まれません。このことは、PHP をさらに安全にします。展開時に PHP プログラムをコンパイルする必要も、PHP を作成する展開ツールの学習に時間を費やす必要もないため、展開もずっと容易です。変更を Web ページに挿入するのみでよく、迅速に入れ替えることができます。

▶ 短い学習曲線

入門は簡単です。PHP は、堅固でセキュアな、最新式 Web ベース・アプリケーションを設計できる、オブジェクト指向 (OO) 機能も備えています。PHP を使用すると、お気に入りのデータベースから最新の情報を迅速に反映する Web ページを作成できます。Web ページを参照しているユーザーから情報を取得し、このユーザー専用ページをカスタマイズすることができます。他に、E メール、フラット・ファイル、または Lightweight Directory Access Protocol (LDAP) データ・ストアなど他のデータへのアクセスを提供する機能もあります。PHP は、スペル・チェッカー、XML 機能、イメージ生成などの機能なども持ちます。

▶ 活発な開発者コミュニティ

Web サイト、ブログ、討議フォーラムなど、PHP 開発者が使用できるリソースは多数存在します。すぐに利用でき、アプリケーション開発の高速化に使用できる、PHP ベースのアプリケーション、関数ライブラリー、コンポーネント、およびフレームワークが多数存在します。これらのソフトウェアの大半は、フリーであり、あらゆるプロジェクトで使用できます。これらは、PHP コミュニティーにとって、PHP に関するアイデア、例、テクニック、およびその他のヒントにより開発者を支援する、非常に貴重な資産になります。

Java には PHP と比較して、以下のような長所があります。

▶ オブジェクト指向言語機能の完全な実装

Java には、オブジェクト指向 (OO) 言語機能が完全に実装されています。PHP は、元々、オブジェクト指向言語として設計されませんでした。後の PHP バージョン 3 になって初めて、C++ および Java を主にベースとする OO プログラミングが一部の機能に限り導入されました。PHP 5 で、インターフェース、アクセス制御、抽象クラスなど、多くの OOP コンポーネントが追加されました。この結果、PHP はついに「OO レディー」になりました。ただし、PHP には、Java に備わっている多くの OO 機能がまだ存在しません。

▶ 優れたコードの保守性

PHP は、非常に簡単に使用を開始できるため、保守性の低いコードを記述することも容易です。Java では、保守性の高い明確なコードを記述することを、より確固として促進する方式および概念が発展しています。PHP 開発者が、(ビジネス・ロジックと GUI の分離、データベースの抽象化など) 一般的な「ベスト・プラクティス」プログラミング規則に気を配らなかった PHP アプリケーションは、他のプラットフォームやデータベースへの移植が困難な状態になりがちです。この場合は、後で、新機能を追加することも困難になります。

▶ J2EE™ によるエンタープライズ管理の容易性

IBM WebSphere Application Server 製品など、大部分のフル J2EE 実装では、エンタープライズ Web システムのクラスターリング、フェイルオーバー、および全体管理を幅広くサポートしています。PHP 実装は、一般に、この面では拡張されておらず、緊密に統合されていません。

以上の PHP および Java の長所、短所のリストは、執筆時点のものです。PHP も Java も急速に変化しており、問題の多くは改善されつつあります。さらに一部の短所は、いずれの言語でも、拡張モジュールなどのアドオン製品を使用することで軽減できます。





## 管理

この章ではインストール、構成、各種管理ツール、ランタイム環境の管理方法、製品のバックアップ方法など、Zend Core for i5/OS の管理方法の概要について説明します。

## 3.1 パッケージ化

Zend 社から Zend Core for i5/OS を介して System i 上で稼働する PHP 機能が提供されています。

## 3.2 Zend Core for i5/OS のインストール

i5/OS に Zend Core をインストールするには、Zend 社の Web サイトよりダウンロードした i5/OS 保管ファイルから、ライセンス・プログラムを復元します。インストール実行時に最も考慮すべき点は、ご使用のシステムが製品の要件をすべて満たしているかどうかです。

**ヒント：**時間を取って前提条件を確認してください。前提条件を慎重に確認することで、インストール時の問題の多くを回避できます。

Zend Core for i5/OS は以下の URL からダウンロードできます (本書執筆現在)。

[http://www.zend.com/products/zend\\_core/zend\\_for\\_i5\\_os](http://www.zend.com/products/zend_core/zend_for_i5_os)

### 3.2.1 開始する前に

時間を取ってシステムのチェックをいくつか実施することにより、インストール時の一般的な問題を回避できます。ご使用のシステムに Zend Core for i5/OS を確実にインストールできるように、以下に示した作業の実行をお勧めします。

#### ハードウェア前提条件の確認

現時点では、Zend Core for i5/OS を実行するための公式なハードウェア要件はありません。ソフトウェア前提条件を含めない Zend Core 製品用として、200 MB 以上のハード・ディスク・スペースを確保することをお勧めします。

Zend Core for i5/OS の提供する PHP 環境自体は、特に強力な処理能力を必要としたり、システム・リソースによって大きく制約を受けたりするものではありません。ハードウェア・リソースの要件は以下の点に左右されます。

- ▶ 実行する予定の PHP アプリケーションの数。実行するアプリケーションの規模や複雑さ。
- ▶ サポートする予定のユーザー数。ユーザーの使用の予測集中状況 (多い、または少ない)。
- ▶ PHP アプリケーションが必要とする処理能力。動的コンテンツの使用度 (高い、または低い)。
- ▶ PHP アプリケーションによるデータベースまたはシステム・オブジェクトへのアクセス頻度。

アプリケーションまたはファイル、ユーザー、処理、リソース・アクセスの数が多くなれば、ハードウェアのリソースもそれだけ必要になります。

#### ソフトウェア前提条件の確認

Zend Core for i5/OS をインストールする前に、前提ソフトウェアと前提修正のすべてがインストールされていることを確認してください。

#### ライセンス・プログラムの確認

以下のステップを実行して、ユーザーのシステムに前提となるライセンス・プログラムがすべてインストールされているかどうかを確認します。

1. i5/OS システムにサインオンし、「GO LICPGM」コマンドを実行します。

2. 「ライセンス・プログラムの処理」の表示画面で、オプション 10 を入力します ( 導入済みライセンス・プログラムの表示 )。
3. F11 キーを 2 回押して製品オプションを表示します。
4. 表 3-1 にリストされているソフトウェアがインストールされていることを確認します。

表 3-1 M ソフトウェア前提条件

ライセンス・プログラム	オプション	記述テキスト
5722SS1	*BASE	i5/OS バージョン 5 リリース 4 (V5R4)
5722SS1	13	システム・オープンネス・インクルード
5722SS1	30	Qshell
5722SS1	33	Portable Application Solutions Environment (PASE)
5722SS1	34	デジタル証明書マネージャー <sup>1</sup>
5722DG1	*BASE	IBM HTTP Server for i5/OS
5722JV1	*BASE	IBM Developer Kit for Java <sup>2</sup>
5722JV1	オプション 5 または 6	Java Development Kit 1.3 または 1.4 <sup>2</sup>
5722SC1	*BASE	IBM Portable Utilities for i5/OS
5722SC1	オプション 1	OpenSSH、OpenSSL、zlib

<sup>1</sup>Secure Sockets Layer (SSL) の構成には、デジタル証明書マネージャーが使用されます。

<sup>2</sup>IBM Developer Kit for Java (5722JV1) は、Zend Core for i5/OS の実行には必要ではありません。これは、IBM HTTP Server for i5/OS (5722DG1) に同梱の IBM Web Administration for i5/OS のツールをサポートするために必要です。Web Administration ツールはブラウザ・ベースのグラフィック・ツールで、これを使用して HTTP サーバーを管理できます。Zend Platform for i5/OS に付属する Java Bridge 機能を使用する場合は、Java も必要です ( 詳しくは 第 6 章『Java Bridge サポート』(81 ページ) を参照してください)。

5. F3 キーを 2 回押してメインメニューに戻ります。

### i5/OS 修正の確認

最新の個別修正、およびグループ修正がシステム上にインストールされていることを確認してください。( **グループ修正** とは特定の製品に関連する修正の集合です。)

最新の修正のレベルを特定するには以下のリソースを参照してください。

- ▶ i5/OS の修正 ( データベースを含む )

[http://www-912.ibm.com/s\\_dir/slkbasesf/recommendedfixes](http://www-912.ibm.com/s_dir/slkbasesf/recommendedfixes)

- ▶ i5/OS PASE の修正

<http://www.ibm.com/servers/enable/site/porting/series/pase/misc.html>

- ▶ IBM HTTP Server for i5/OS の修正

<http://www.ibm.com/servers/eserver/series/software/http/services/service.html#PTF>

- ▶ IBM Developer Kit for Java

[http://www-912.ibm.com/s\\_dir/sline003.NSF/GroupPTFs?OpenView&view=GroupPTFs](http://www-912.ibm.com/s_dir/sline003.NSF/GroupPTFs?OpenView&view=GroupPTFs)  
( 「R540」を展開し、「SF99291: 540 Java」をクリックします。)

PTF の表示 (DSPPTF) コマンド ( 個別修正用 ) および PTF グループの処理 (WRKPTFGRP) コマンド ( グループ修正用 ) を使用して、システムに適用されている修正を確認します。

修正が不足している場合は、Zend Core for i5/OS をインストールする前に、その修正をオーダーし、インストールする必要があります。

### ユーザー・プロファイル権限の確認

インストール・プロセス ( および他の管理作業 ) の実行時は、「\*SECOFR」ユーザー・クラスのユーザー・プロファイル ( すべての特殊権限を含む ) を使用する必要があります。ユーザー・プロファイルの処理 (WRKUSRPRF) コマンドを使用して、使用中のユーザー・プロファイルを確認します。

### Zend Core for i5/OS のアルファ版またはベータ版の除去

**ヒント** :Zend Core for i5/OS の初回インストール時は、『既存の製品ディレクトリーとライブラリーの確認』(14 ページ) にスキップしてください。

実動レベル版の Zend Core for i5/OS をインストールする前に、この製品のアルファ版、またはベータ版を除去する必要があります。システムで旧版の残りが検出されると、インストールは失敗します。

製品のプレリリースを除去するために実行するステップはお手持ちのバージョンによって異なります。3.2.7 『Zend Core for i5/OS のアンインストール』(22 ページ) のステップに従ってください。Zend Core 製品がライセンス・プログラム・リストに表示される場合は、アンインストール・ステップに従って製品を除去します。

Zend Core がライセンス・プログラム・リストに表示されない ( アルファ版、または初期のベータ版がインストールされていることを意味します ) 場合は、以下のステップを実行します。

1. 5250 セッションから i5/OS システムにサインオンします。
2. 以下のコマンドを実行し、アンインストール・プログラムを呼び出します。  
DLTLICPGM 1ZCORE5
3. アンインストーラーでは ZENDCORE ライブラリーが除去されないため、以下のコマンドを実行して、ライブラリーを除去します。  
DLTLIB LIB(ZENDCORE)

### 既存の製品ディレクトリーとライブラリーの確認

製品の一部のインストールが残らないようにしてください。製品が完全に除去されていないと、インストールの失敗につながる可能性があります。

以下のステップにより検査を実施して、インストールが残っている場合はクリーンアップします。

1. ZENDCORE ライブラリーが存在しないことを確認します。
  - a. i5/OS コマンド行で以下のコマンドを入力します。  
WRKLIB LIB(ZENDCORE)
  - b. ZENDCORE ライブラリーがリストされた場合は、ライブラリーの隣りのオプション行に 4 と入力し、Enter キーを押します。
  - c. 除去の確認で Enter キーを押します。
  - d. F3 キーを押して、メイン・コマンド・メニューに戻ります。

2. Qshell を開始します。i5/OS のコマンド行で以下のコマンドを入力します。

STRQSH

**ヒント:** Qshell ではディレクトリーとその内容の再帰的削除が可能なことから、Qshell を使用して既存のリソースを確認することをお勧めします。

3. /www/zendcore ディレクトリーが存在しないことを確認してください。

**重要:** このディレクトリーを削除する前に、重要なファイル (php.ini および httpd.conf など) をバックアップしてください。

- a. /www ディレクトリーに移動します。

cd /www

- b. ディレクトリーの内容を表示します。

ls

- c. zendcore という名前の付いたディレクトリーが表示された場合は、以下のコマンドを使用して削除します。

rm -r zendcore

4. /usr/local/Zend ディレクトリーが存在しないことを確認します。

- a. /usr/local ディレクトリーに移動します。

cd /usr/local

- b. ディレクトリーの内容を表示します。

ls

- c. Zend という名前の付いたディレクトリーが表示された場合は、以下のコマンドを使用して削除します。

rm -r Zend

5. F3 キーを押して、Qshell を終了します。

## TCP/IP の構成の確認

Web テクノロジーは TCP/IP に大きく依存しています。Zend Core 製品をインストールする前に、TCP/IP が適切に構成されていることを確認してください。

特に、以下の構成の設定を確認する必要があります。

- ▶ システムにホスト名が定義されていることを確認します。TCP/IP の構成 (CFGTCPIP) コマンドを実行し、オプション 12 (TCP/IP ドメイン情報の変更) を選択してこの設定を表示します。値がホスト名フィールドにリストされていることを確認します。
- ▶ ループバック・エントリー (「localhost」または「127.0.0.1」を示す) が TCP/IP のホスト・テーブルで構成されていることを確認してください。TCP/IP の構成 (CFGTCPIP) コマンドを実行し、オプション 10 (TCP/IP の構成) を選択してホスト・テーブルを表示します。IP アドレス 127.0.0.1 のエントリーが存在し、LOOPBACK と LOCALHOST のホスト名にマップされていることを確認してください。

この表示画面ではさらに、i5 システムの IP アドレスがそのホスト名 (SYSTEMA など) と完全修飾ホスト名 (SYSTEMA.MYCOMPANY.COM) にマップされていることも確認します。i5/OS からのご使用のシステムのループバック・アドレスおよび完全修飾ホスト名と、PHP スクリプトへのアクセスに使用されるブラウザからの完全修飾ホスト名のいずれにも ping できる必要があります。

### 3.2.2 Zend Core for i5/OS 製品のインストール

前提条件のセットアップとチェックが完了したら、Zend Core 製品をインストールするための準備が整います。

1. Zend.com から Zend Core for i5/OS 製品をダウンロードします。

[https://www.zend.com/core/oem\\_registration.php?access\\_code=IBMi5OSZend](https://www.zend.com/core/oem_registration.php?access_code=IBMi5OSZend)

最初に、Zend.com に登録する必要があります。このプロセスの間に、Zend Network のユーザー ID とパスワードを作成します。この ID とパスワードで Zend ダウンロード・ページへのログインと、インストール後の製品の更新が可能になります。このユーザー ID とパスワードは製品に登録可能で、インストール中か、インストール後にセットアップ・メニューから指定できます。

ワークステーション・システムに Zip ファイルをダウンロードします。

2. Zip ファイルをワークステーションの一時ディレクトリーに解凍します。

以下のファイルが解凍されます。

**zcoresavf.savf** Zend Core for i5/OS 製品を含む保管ファイル。

**Release\_notes** フィードバックの送信に関する情報、製品のフィーチャーと拡張機能、システム前提条件、インストール指示、Zend Studio および Zend Core 間のデバッグ接続のセットアップに関する情報、および各種コンポーネントのライセンスへのリンクなどが含まれます。

**Readme** 基本的なインストールの説明とシステム前提条件が記載されています。

**Zend\_Core\_User\_Guide\_i5OS.pdf**

インストール手順、「始めに」情報、および Zend Core コンソールに関する説明が記載されています。

3. すべての特殊権限を持つ \*SECOFR ユーザー・クラスのユーザー・プロファイルで i5/OS システムにログオンします。

4. 以下のコマンドで保管ファイルを作成します。

```
CRTSAVF FILE(QGPL/ZCORESAVF) TEXT('Zend Core product save file')
```

5. 以下のコマンドを入力して、i5/OS システムで FTP が実行していることを確認します。

```
NETSTAT *CNN
```

「Local Port」列で、FTP または 21 を見つけます。

6. ワークステーション側で、コマンド・プロンプトを開き、Zend Core の保管ファイルを i5/OS システムに転送します。

- a. Zip ファイルから解凍したファイルがあるディレクトリーに移動します。以下に例を示します。

```
cd temp
```

- b. i5/OS システムの名前を指定して FTP コマンドを実行します。以下に例を示します。

```
ftp systema
```

- c. ユーザー・プロファイルとパスワードの入力を促すメッセージが表示されたら有効なユーザー・プロファイルとパスワードを入力します。

- d. バイナリー形式による転送を指定するために **bin** コマンドを入力します。

- e. 保管ファイルを i5/OS システムに転送します。

```
put zcoresavf.savf qgpl/zcoresavf
```

- f. 転送が終了した時点で、**quit** コマンドを入力します。

7. 5250 セッションに戻り、ライセンス・プログラムの復元 (RSTLICPGM) コマンドを実行して、製品をインストールします。

RSTLICPGM LICPGM(1ZCORE5) DEV(\*SAVF) SAVF(QGPL/ZCORESAVF)

**注:** ライセンス・プログラム (LICPGM) のパラメーターの値は、数字の 1 から始まります。英大文字の i ではありません。

代わりに、RSTLICPGM と入力し、F4 キーを押して、コマンド・パラメーターを表示することもできます。F10 キーを押すと、他のパラメーターも参照できます。製品、デバイス、保管ファイル、ライブラリーの値を指定して、Enter キーを押します。

ライセンス・プログラム復元 (RSTLICPGM)		
選択項目を入力して、実行キーを押してください。		
プロダクト .....	<u>1ZCORE5</u>	文字値
装置 .....	<u>*SAVF</u>	名前, *SAVF
値の続きは +		
復元される任意選択部分 .....	<u>*BASE</u>	*BASE, 1, 2, 3, 4, 5, 6, 7...
復元するオブジェクトのタイプ ..	<u>*ALL</u>	*ALL, *PGM, *LNG
ライセンス・プログラムの言語 ..	<u>*PRIMARY</u>	文字値, *PRIMARY, *SAVVOL
出力 .....	<u>*NONE</u>	*NONE, *PRINT
リリース .....	<u>*FIRST</u>	文字値, *FIRST
リリースの置き換え .....	<u>*ONLY</u>	文字値, *ONLY, *NO
Volume identifier .....	<u>*MOUNTED</u>	
値の続きは +		
Sequence number .....	<u>*SEARCH</u>	1-16777215, *SEARCH
End of media option .....	<u>*REWIND</u>	*REWIND, *LEAVE, *UNLOAD
保管ファイル .....	<u>ZCORESAVFZ</u>	名前
ライブラリー .....	<u>QGPL</u>	名前, *LIBL, *CURLIB
		続く ...
F3= 終了    F4= プロンプト    F5= 最新表示    F10= 追加のパラメーター F12= 取り消し    F13= この画面の使用法    F24= キーの続き		

図 3-1 ライセンス・プログラムの復元 (RSTLICPGM) コマンドのプロンプト

8. 続いて、インストール・ウィザードが開始します。ウィザードの実行中、以下のアクションを実行します。
- ご使用条件を読んで、受諾します。
  - Zend Core Web 管理コンソールへのアクセスに使用するパスワードを指定します。このパスワードは任意で、いずれのユーザー・プロファイルや ID にも関連付けられていません。
  - Zend Network ユーザー ID とパスワード (製品 Zip ファイルのダウンロードに使用したもの) を入力します。この情報を指定することで、Zend Core for iOS インストール済み環境に自動更新をセットアップできます。ここでユーザー ID とパスワードを入力する必要はありません。この情報は後で指定できます。

**注:** F3 キーを押すか、インストール・プロセスが完了する前にインストール・ウィザードが終了するような選択をした場合は、『既存の製品ディレクトリーとライブラリーの確認』(14 ページ)に記載されているステップに従い、システムをクリーンアップしてください。「Zend Network ID」プロンプトで F3 キーを押してもインストール・プロセスは終了されない点に注意してください。つまり、その特定のフィーチャーが構成されないだけで、インストールは続行します。

### 3.2.3 Zend Core for i5/OS 環境の開始

製品がインストールされたら、環境を開始します。Zend Core for IBM i5/OS Setup Tool を使用します。

1. i5/OS コマンド行で以下のコマンドを入力します。

GO ZENDCORE/ZCMENU

図 3-2 で示すように、Zend Core Setup Tool が表示されます。

```
ZCMENU                                Zend Core for IBM i5/OS Setup Tool

Select one of the following:

    1. Set Zend Core Web Administration Console password
    2. Update via Zend Network menu
    3. Run Support Tool

    5. Service Management menu

   90. Signoff

Selection or command
===> _____

F3=Exit  F4=Prompt  F9=Retrieve  F12=Cancel  F23=WRKUSRJOB
Copyright Zend Technologies LTD (2006)
```

図 3-2 Zend Core for IBM i5/OS Setup Tool

2. コマンド行で 5 と入力して「Service Management」メニューを選択し、Enter キーを押します。

図 3-3 (19 ページ) で示すように、「Service Management」メニューが表示されます。



ZCAMENU	Zend Core for IBM i5/OS Setup Tool
Select one of the following:	
1. Start Zend Core Subsystem 2. Stop Zend Core Subsystem  4. Start Apache server instances 5. Stop Apache server instances 6. ReStart Apache server instances  8. Start i5_COMD service 9. End i5_COMD service	
Selection or command ==> _____	
F3=Exit    F4=Prompt    F9=Retrieve    F12=Cancel    F23=WRKUSRJOB	

図 3-3 「Service Management」メニュー

3. コマンド行で 1 と入力して Zend Core サブシステムを開始するためのオプションを選択し、Enter キーを押します。

このアクションにより ZEND サブシステムで実行するジョブが開始されます。

4. ZEND サブシステムを開始すると、必要なすべての Core および Platform のジョブが開始されます。
5. 適切なジョブが実行しているかどうかを確認してください。

- a. 最初に IBM HTTP Server for i5/OS のインスタンス (ZENDCORE) が実行していることを確認します。

コマンド行で、以下のコマンドを入力します。

WRKACTJOB SBS(QHTTPSVR)

QHTTPSVR サブシステムで実行しているアクティブなジョブのリストが表示されます。ZENDCORE という名前のジョブが 5 つ、リストに表示されていることを確認します。(Page Down キーを押してリストをスクロールする必要がある場合があります。)

- b. F3 キーを押して「Service Management」メニューに戻ります。
- c. Zend Core のジョブが実行しているかどうかを確認します。以下のコマンドを入力します。

WRKACTJOB SBS(ZEND)

ZEND サブシステムで実行しているアクティブなジョブのリストが表示されます。以下のジョブのうち、1 つ以上のインスタンスがリストに表示されていることを確認します。

- I5\_COMD
- ZC\_STR\_PRN
- ZENDCOREAP

- d. F3 キーを押して「Service Management」メニューに戻ります。

6. F3 キーを 2 回押してメイン・コマンド・メニューに戻ります。

### 3.2.4 インストールの確認

Zend Core 環境が開始したら、インストールが正常に終了したことを確認します。

インストールを確認するには、PHP ファイルが処理できることを確認します。おそらく、最も簡単な確認方法は、Zend Core for i5/OS の管理コンソールにアクセスすることです。コンソールはブラウザ・ベースのツールで、Zend Core 製品の管理に使用されます。ツールは PHP で記述されているため、コンソールが機能するということは PHP 環境も機能することを意味します。

Zend Core for i5/OS の管理コンソールにアクセスするには、以下のステップを実行します。

1. Web ブラウザーを開き、以下の URL を入力します。

`http://<server_name>:89/ZendCore`

<server\_name> には i5 システムのホスト名か IP アドレスを入力します。

2. リダイレクト・ページが表示されます。続いて、図 3-4 で示すように、Zend Core コンソールのログイン・ページが表示されます。

インストール・ウィザード実行中に入力した管理パスワードでコンソールにログインします。



図 3-4 Zend Core for i5/OS コンソールのログイン・ページ

3. 図 3-5 (21 ページ) で示すように、コンソールが表示されます。

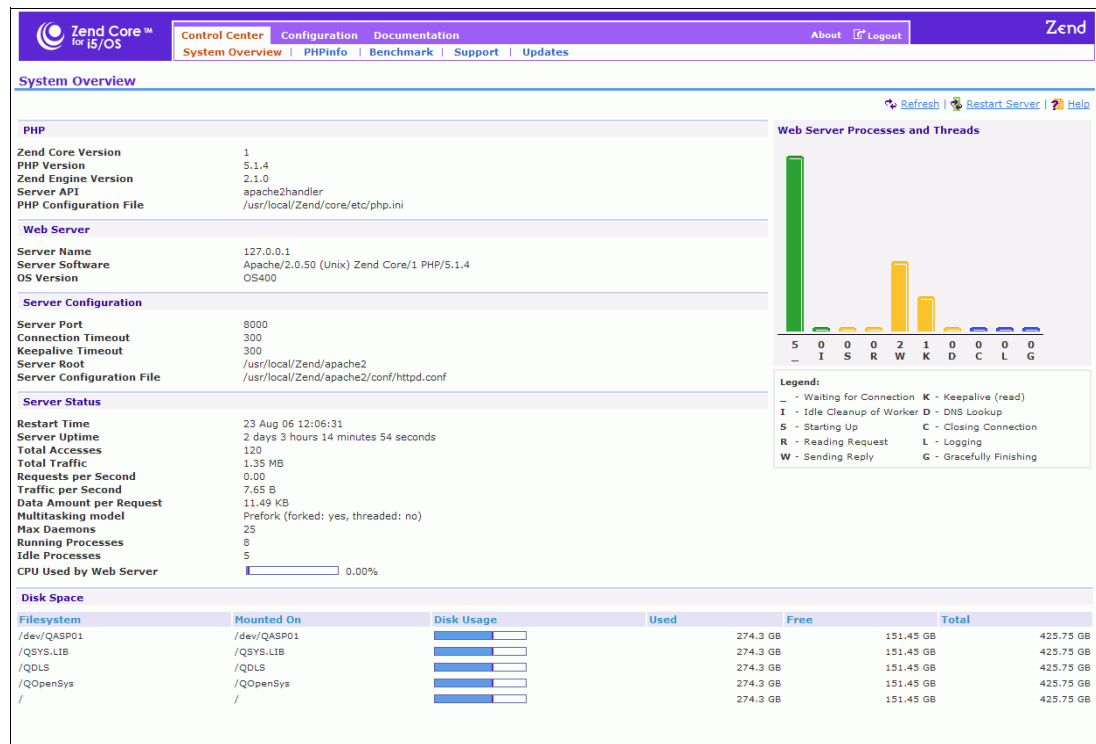


図 3-5 Zend Core for i5/OS コンソール

コンソールが機能していることを確認したら、コンソールの右上隅にある「Logout」をクリックし、ブラウザ・ウィンドウを閉じます。

### 3.2.5 一般的なインストールの問題

インストールが失敗すると、通常、以下のエラー・メッセージが表示されます。

プロダクト 1ZCORE5 オプション \*BASE リリース \*FIRST のオブジェクトは復元されなかった。

ジョブ・ログでメッセージを確認します。インストールの失敗は、通常、以下の条件が 1 つ以上発生した場合に起こります。

- ▶ ご使用のユーザー・プロファイルに十分な権限がない。
- ▶ 旧インストールの構成が、インストーラーにより検出された。
- ▶ ライブラリー・リストに QGPL ライブラリー、または QTEMP ライブラリーが含まれていない。
- ▶ 前提条件のソフトウェア製品または修正が不足している。
- ▶ /usr/local がシンボリック・リンクとして存在する。このシンボリック・リンクを削除します。任意で、本物のディレクトリーとして再作成します。

問題を修正した後は、問題のあったインストール中に作成された製品ファイルをすべて削除してください。詳しくは、『既存の製品ディレクトリーとライブラリーの確認』（14 ページ）を参照してください。

### 3.2.6 Zend Core for i5/OS の再インストール

何らかの理由で Zend Core for i5/OS を再インストールする必要がある場合は、次のセクションで概説されているとおり、必ず、先に既存のインストール済み環境をアンインストールしてください。

### 3.2.7 Zend Core for i5/OS のアンインストール

Zend Core for i5/OS をアンインストールする前に、後の参照用として保管するすべてのファイルをバックアップしてください。アンインストーラーはすべての製品ファイルと構成ファイルを削除します。詳しくは、3.5 『Zend Core for i5/OS のバックアップ』 (37 ページ) を参照してください。

1. すべての特殊権限を持つ \*SECOFR ユーザー・クラスのユーザー・プロファイルで、5250 セッションから i5/OS システムにサインオンします。
2. ライセンス・プログラムの削除 (DLTLICPGM) コマンドを実行して、Zend Core for i5/OS をアンインストールします。  
DLTLICPGM LICPGM(1ZCORE5)
3. アンインストールが完了すると、以下のようなメッセージが表示されます。  
プロダクト 1ZCORE5 オプション \*BASE リリース V5R4M0 の \*PGM オブジェクトが削除された。  
プロダクト 1ZCORE5 オプション \*ALL リリース \*ONLY のオブジェクトが削除された。

アンインストール時の一般的な問題：

- ▶ ZENDCORE ライブラリーが現行ライブラリーに設定されている。現行ライブラリー変更 (CHGCURLIB) コマンドを使用して、別のライブラリーを現行ライブラリーに設定します。
- ▶ 製品ファイルが不足している。アンインストーラーは ZENDCORE ライブラリーの一部のファイルを使用しています。通常は起こり得ませんが、これらのファイルの一部が不足している場合、アンインストーラーが失敗する可能性があります。この問題を解決するには、製品を再インストールして完全なインストール状態に戻し、その後にアンインストールします。(『Zend Core for i5/OS の再インストール』 (21 ページ) を参照してください。)

### 3.2.8 製品構成

Zend Core for i5/OS インストール済み環境では以下のオブジェクトが作成されます。

#### ライブラリー

ZENDCORE ライブラリーにはインストール、構成、および製品環境の開始に関連した i5/OS 固有のコードが含まれています。

#### ユーザー・プロファイル

Zend Core のインストールにより、表 3-2 にリストされているユーザー・プロファイルが作成されます。

表 3-2 Zend Core ユーザー・プロファイル

ユーザー・プロファイル	ユーザー・クラス	特殊権限	グループ・プロファイル	説明
ZENDADMIN	*SECOFR	(すべて)	*NONE	Zend Core 管理ユーザー。特定の ZEND サブシステム・ジョブを開始するほか、暗号サービス用の Zend Core 疑似乱数発生ルーチン・ジョブを実行するために使用されます。
ZENDTECH	*USER	*NONE	*NONE	PHP の構成を更新するために使用されます。

ユーザー・ プロファイル	ユーザー・ クラス	特殊権限	グループ・ プロファイル	説明
NOBODY	*USER	*NONE	NOGROUP	i5/OS PASE で Apache および Zend Core ジョブ (ZENDCOREAP) を実行するために使用されます。
NOGROUP	*USER	*NONE	*NONE	NOBODY ユーザー・プロファイルのグループ・プロファイル。i5/OS PASE Apache HTTP Server の構成で指定されます。

**重要：**これらのユーザー・プロファイルにはパスワードが定義されていないため、システムへのサインオンには使用できません。

Zend Core for i5/OS は以下のシステム提供のユーザー・プロファイルを使用します。

▶ QSECOFR

ジョブの ZENDCOREAP グループのジョブ制御は、QSECOFR の下で実行されます。

▶ QTMHHTTP

IBM HTTP Server for i5/OS サーバー (ZENDCORE) をサポートするジョブは、デフォルトの HTTP サーバー・ユーザー・プロファイルの下で実行されます。これらのジョブは QHTTSPVR サブシステムで実行されます。

▶ QTCP

i5 Toolkit for Zend Core をサポートするジョブは QTCP ユーザー・プロファイルの下で実行されます。

## ディレクトリー

Zend Core for i5/OS は以下のルート・ディレクトリー構造を使用します。

▶ /www/zendcore

このディレクトリーは i5/OS HTTP サーバー・インスタンスの IBM HTTP Server (powered by Apache) が含まれます。このディレクトリーのサブディレクトリーは以下のとおりです。

– /www/zendcore/conf

このディレクトリーには ZENDCORE サーバー・インスタンスの構成を表す httpd.conf ファイルが含まれています。

– /www/zendcore/htdocs

このディレクトリーは IBM HTTP Server インスタンスと i5/OS PASE Apache サーバー・インスタンスのドキュメント・ルートです。PHP ファイルと他のアプリケーションのリソースはこのディレクトリーにあります。

– /www/zendcore/logs

このディレクトリーには ZENDCORE サーバー・インスタンスのログ・ファイル (デフォルトで、access\_log ファイルおよび error\_log ファイル) が含まれます。

▶ /usr/local/Zend

このディレクトリーにはライセンスおよび製品の情報ファイルが含まれます。また、2種類の重要なサブディレクトリーもここに配置されています。

– /usr/local/Zend/apache2

このディレクトリーには i5/OS PASE で実行する Apache HTTP Server が入っています。複数のサブディレクトリーに製品のバイナリーが含まれます。管理者としてよく理解しておく必要があるのは、以下のサブディレクトリーのみです。

- /usr/local/Zend/apache2/bin

このディレクトリーには Apache HTTP Server インスタンスのバイナリーが含まれます。

- /usr/local/Zend/apache2/conf

このディレクトリーには i5/OS PASE Apache サーバー・インスタンスの構成を表す httpd.conf ファイルが含まれています。

- /usr/local/Zend/apache2/logs

このディレクトリーには製品ログ・ファイル (デフォルトで、access\_log ファイルおよび error\_log ファイル) が含まれます。

– /usr/local/Zend/Core

このディレクトリーには Zend Core 製品が含まれます。さまざまなサブディレクトリーの中で、重要なディレクトリーを以下に示します。

- /usr/local/Zend/Core/bin

このディレクトリーには Zend Core 製品のバイナリーが含まれます。

- /usr/local/Zend/Core/etc

このディレクトリーには PHP 環境の構成ファイルが含まれます。その中でも重要なのは php.ini ファイルです。

- /usr/local/Zend/Core/logs

このディレクトリーには Zend Core 製品のログ・ファイル (php\_error\_log および ini\_modifier\_log ファイル) が含まれます。

### 3.2.9 ランタイム環境

このセクションでは管理アクティビティーに関連して、Zend Core for i5/OS のランタイム環境について取り扱います。

#### コンポーネント

Zend Core for i5/OS 管理の技術的な詳細を理解する上で、ランタイムの各コンポーネントの全体図 (図 3-6) を把握し、それぞれのコンポーネントがどのように相互作用しているかを知ることは有用です。

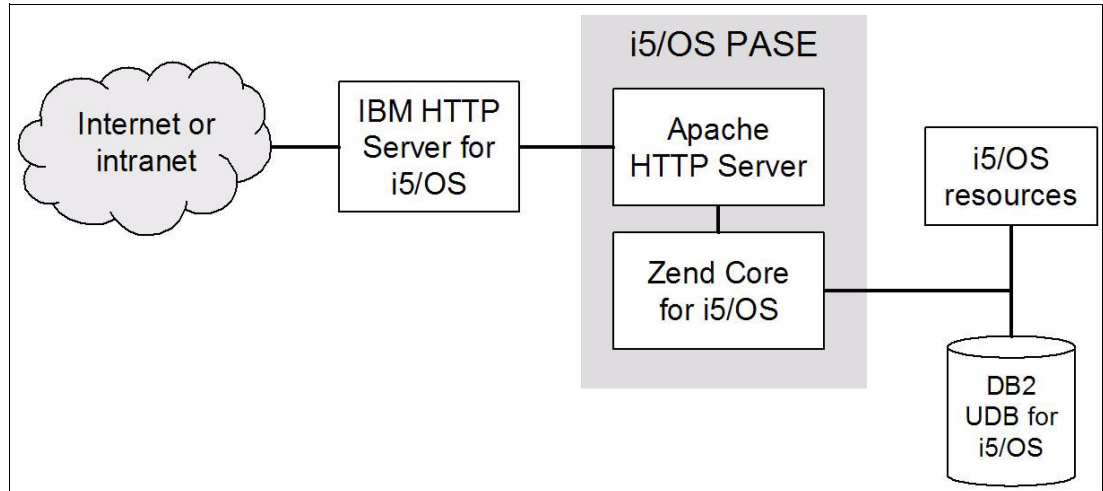


図 3-6 Zend Core for i5/OS ランタイム環境

Zend Core for i5/OS が提供する PHP 環境のランタイムは、以下のコンポーネントにより構成されています。

► IBM HTTP Server または i5/OS

クライアント要求は i5/OS インスタンスの IBM HTTP Server が受信します。サーバー・インスタンスは ZENDCORE という名前で、デフォルトでポート 89 を listen します。

このサーバーは i5/OS PASE で実行する Apache HTTP Server にすべての要求を渡すようセットアップされます。また、PASE ベースの Apache HTTP Server からの応答を、ユーザーのブラウザーに返す働きもあります。最初のケースでは、i5/OS サーバーはプロキシとして機能します。2 番目のケースでは、リバース・プロキシとして機能します。

フロントエンドでネイティブ i5/OS HTTP サーバーを使用することには、さまざまな利点があります。

- サーバー・インスタンスは IBM Web Administration for i5/OS ツールで管理されます。i5 固有のツールである、Web Administration ではポイント・アンド・クリック方式とウィザード・ベースで HTTP サーバーを構成できます。
- プロキシ/リバース・プロキシのスキームを使用することで、サーバーへのエントリー・ポイントと、ダイナミック・コードが実際に処理される場所との間に、守備用のレイヤーをもう一枚作ることができます。
- この構成では、構成へのセキュア・ソケット (SSL または HTTPS) の追加が簡単になります。詳しくは、第 7 章『セキュリティ』(89 ページ) を参照してください。

► i5/OS PASE

i5/OS PASE は i5/OS での AIX® ランタイムです。エミュレーターではありません。i5 システムと p5 システムを 1 つのハードウェアで共用できること、また異なるランタイム間で切り替えられることが利点です。さらに、AIX アプリケーションを System i5™ に接続することも容易になり、堅固な AIX アプリケーションの利点を生かしながら、管理しやすい i5 の利点も享受できます。

► Apache HTTP Server

Zend Core for i5/OS は i5/OS PASE 環境で稼働するため、処理のパフォーマンスを向上するため、2 つ目の HTTP サーバー・インスタンスが用意されています。こちらも i5/OS PASE 上で実行されます。

Apache サーバーは PHP アプリケーション用の Web サーバー環境を提供します。デフォルトでは、ポート 8000 を listen します。

前述のように、要求は IBM HTTP Server for i5/OS インスタンスによって Apache サーバーに転送されて処理されます。Apache サーバーはローカル・システム (127.0.0.1) からの要求のみ受け入れるよう、構成されています。

▶ Zend Core for i5/OS

Zend Core は PHP ランタイム環境を提供します。PHP 処理の要求は Apache HTTP Server を経由して Zend Core に渡されます。Zend Core の PHP エンジン は PHP コードを解釈し、データベース照会やシステム・リソースへのアクセスなど、必要な処理を実行します。

PHP エンジン は PHP コードを処理の結果で置き換え、Apache HTTP Server に応答を返し、その応答が IBM HTTP Server を介して、ユーザーに返されます。

▶ i5/OS のデータとリソース

PHP アプリケーションは i5/OS データのほか、プログラム (\*PGM) オブジェクト、ファイル、データ待ち行列、コマンド、データ域など、各種リソースにもアクセスできます。多様なアプリケーション・プログラミング・インターフェース (API) を使用して、プログラマチックにこれらのリソースにアクセスできます。詳しくは、第 4 章『アプリケーション開発』(39 ページ) で説明します。

## サブシステムとジョブ

図 3-7 (26 ページ) で示すサブシステムとジョブは、Zend Core for i5/OS のデフォルトです。

活動ジョブの処理						
現行 Opt サブシステム / ジョブ	ユーザー	タイプ	CPU %	機能	状況	
QHTTPSVR	QSYS	SBS	.0		DEQW	
ZENDCORE	QTMHHTTP	BCH	.0	PGM-QZHBMAIN	SIGW	
ZENDCORE	QTMHHTTP	BCI	.0	PGM-QZSRLOG	SIGW	
ZENDCORE	QTMHHTTP	BCI	.0	PGM-QZSRLOG	SIGW	
ZENDCORE	QTMHHTTP	BCI	.0	PGM-QZSRHTTP	SIGW	
ZENDCORE	QTMHHTTP	BCI	.0	PGM-QZSRHTTP	DEQW	
ZEND	QSYS	SBS	.0		DEQW	
I5_COMD	QTCP	ASJ	.0	PGM-EASYCOMD	TIMW	
ZC_STR_PRN	ZENDADMIN	BCI	.0	PGM-prngd	SELW	
ZENDCOREAP	QSECOFR	BCI	.0	PGM-httpd	SELW	
ZENDCOREAP	QSECOFR	BCI	.0	PGM-httpd	DEQW	
ZENDCOREAP	NOBODY	BCI	.0	PGM-httpd	SELW	
ZENDCOREAP	NOBODY	BCI	.0	PGM-httpd	TIM W	
ZENDCOREAP	NOBODY	BCI	.0	PGM-httpd	TIMW	
ZENDCOREAP	NOBODY	BCI	.0	PGM-httpd	TIMW	
ZENDCOREAP	NOBODY	BCI	.0	PGM-httpd	TIMW	
ZENDCOREAP	NOBODY	BCI	.0	PGM-httpd	TIMW	
ZENDCOREAP	NOBODY	BCI	.0	PGM-httpd	TIMW	
ZENDCOREAP	NOBODY	BCI	.0	PGM-httpd	TIMW	
ZENDCOREAP	NOBODY	BCI	.0	PGM-httpd	TIMW	
ZENDCOREAP	NOBODY	BCI	.0	PGM-httpd	TIMW	

図 3-7 Zend Core for i5/OS に関するサブシステムとジョブ

各サブシステムとジョブには以下のような特定のタスクがあります。

- ▶ QHTTPSVR サブシステム :ZENDCORE のジョブは IBM HTTP Server for i5/OS インスタンスをサポートします。最初のジョブ (関数 PGM\_QZHBMAIN を使用) はメイン・ジョブです。PGM-QZSRLOG 関数を使用するジョブは、サーバー・インスタンスのログ機能処理します。残りの 2 つのジョブはサーバー・プロセス用です。



- ▶ ZEND サブシステム :3 種類のジョブが ZEND 製品サブシステムで実行しています。
  - I5\_COMD:i5 PHP Toolkit のサービス
  - ZC\_STR\_PRN: 暗号キーの算出に使用される疑似乱数発生ルーチン・ジョブ
  - ZENDCOREAP:i5/OS PASE Apache HTTP Server ジョブ

## 3.3 Zend Core for i5/OS の構成

このセクションでは Zend Core for i5/OS 環境の構成について解説します。

### 3.3.1 管理ツール

Zend Core と i5/OS には多様なツールが用意されており、これらを使用して、PHP 環境を構成および管理できます。

#### Zend Core Setup Tool

ZENDCORE ライブラリーに含まれ、Setup Tool は環境の開始や更新のほか、トラブルシューティングに使用されます。

i5/OS のコマンド行で以下のコマンドを実行して、Setup Tool にアクセスします。

GO ZENDCORE/ZCMENU

図 3-8 に Setup Tool を示します。

```
ZCMENU                                Zend Core for IBM i5/OS Setup Tool

Select one of the following:

    1. Set Zend Core Web Administration Console password
    2. Update via Zend Network menu
    3. Run Support Tool

    5. Service Management menu

   90. Signoff

Selection or command
===> _

F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel   F23=WRKUSRJOB
Copyright Zend Technologies LTD (2006)
```

図 3-8 Zend Core for IBM i5/OS Setup Tool

Setup Tool のメインメニューには以下のオプションがあります。

- ▶ 1. Set Zend Core Web Administration Console password  
このオプションを使用して、ブラウザー・ベースの Zend Core コンソールにアクセスするためのパスワードを設定または変更します。
- ▶ 2. Update via Zend Network menu  
このオプションを使用して、製品の更新を確認します。このプロセスでは Zend Network にアクセスするための Zend Network ユーザー ID とパスワード (製品ダウンロード時に

Zend に登録した際、作成したもの) が必要です。このオプションの下にこのユーザー ID とパスワードを入力することもできます。

「Update via Zend Network menu」では、更新をロールバックすることもできます。

▶ 3. Run Support Tool

Support Tool は診断データを集め、GZ ファイルにバンドルします。( ツールを実行するとき、ファイルの場所はメッセージ内に表示されます。) そのファイルを Zend サポートに送信し、問題判別に役立てることができます。

▶ 5. Service Management menu

「Service Management menu」には Zend サブシステムを開始および停止するためのオプション、Apache HTTP Server インスタンスを開始、再始動、および停止するためのオプション、i5\_COMD サービス (i5 Toolkit for Zend Core で使用される) を開始および停止するためのオプションが含まれます。

▶ 90. Signoff

このオプションは 5250 セッションをサインオフします。

## Zend Core ブラウザー・ベース・コンソール

Zend Core には PHP 構成を編集できる PHP アプリケーションが用意されています。変更点は製品の php.ini ファイルに書き込まれます。Zend Core の構成にはコンソールを使用することをお勧めします。php.ini ファイルを手動で編集すると、エラーが生じやすく、構成で問題を引き起こす可能性があります。

Zend Core コンソールを開くには、Web ブラウザーに以下の URL を入力します。

<system\_name> は i5 マシンのホスト名か IP アドレスで置き換えます。

http://<system\_name>:89/ZendCore

**注:** システムの TCP/IP 構成によっては、URL に指定するシステムに完全修飾ホスト名 (ホストおよびドメイン) を使用すると、コンソールがロードされない場合があります。この場合、URL に短いホスト名を使用してください。

インストール・プロセス中に指定したパスワードを使用して、コンソールにログインします。Zend Core Setup Tool を使用して、パスワードを初期化、または変更します。

図 3-9 に示されるように、ページ上部にあるメニューを使用して、コンソールをナビゲートします。最上位の行はメイン・コンソール・エリアのタブです。最上位の行にあるアイテムをクリックすると、そのほとんどで 2 番目の行にサブメニューが表示され、そのサブメニューから特定のコンソール・エリアにあるページにアクセスできます。



図 3-9 Zend Core コンソールのナビゲーション・メニュー

すべてのコンソール機能をリストした詳細情報については製品資料を参照してください。ここでは簡単に各タブの概要と、そのページで実行できるアクティビティーについて解説します。

▶ Control Center

Control Center エリアにはシステム、ロードされた PHP モジュール、インストール済みで使用可能な製品の更新、サポートへの問い合わせ方法など、製品に関する情報が含まれます。その中でも特に興味深いのはベンチマーク・ツールで、特定の URL へのヒット数をシミュレートし、応答パフォーマンスを測定できます。

► Configuration

「Configuration」タブでは、PHP コア・モジュール、PHP 拡張機能、Zend 製品、サーバー・ディレクティブ、および Zend Studio Server (サーバー側の各種機能を Zend Studio 開発ツールに統合) などの製品構成を変更できます。

► Documentation

「Documentation」タブからは PHP、Zend Core、PEAR などの組み込まれた製品資料にアクセスできます。資料を検索するための機能も備えています。

多くのコンソール・ページには構成に変更点を保管して、サーバーを再始動するためのコントロールが含まれています。変更点を反映させるために、サーバーを再始動するリンクをクリックしても変更点が現れない場合は、Zend Core Setup Tool を使用してサーバーまたは Zend サブシステムを再始動します。

「Logout」リンクをクリックして、Zend Core コンソールを終了します。

## IBM Web Administration for i5/OS

IBM HTTP Server for i5/OS インスタンスを管理するには、IBM Web Administration for i5/OS ツールを使用します。これはブラウザ・ベースのツールで、IBM HTTP Server for i5/OS 製品に付属します。管理アプリケーションは IBM HTTP Server の \*ADMIN インスタンスからアクセスします。

ツールにアクセスするには、最初に \*ADMIN インスタンスを開始します。System i Navigator または以下の i5/OS コマンドを使用できます。

```
STRTCPSVR SERVER(*HTTP) HTTPSVR(*ADMIN)
```

\*ADMIN インスタンスが起動した後、Web ブラウザーを使用して、以下の URL を開きます。<server\_name> には i5 システムのホスト名か IP アドレスを入力します。

```
http://<server_name>:2001/HTTPAdmin
```

プロンプトが表示されたら、\*IOSYSCFG 権限を持つユーザー・プロファイル名と、関連付けられたパスワードを入力します。

**注：**ブラウザのポップアップ・ブロッカーが有効になっている場合は、IBM Web Administration for i5/OS ツールを使用する前に無効にするか、サイトのポップアップ・ウィンドウにアクセスできるようにブラウザを構成します。ポップアップがブロックされると、ツールの一部の機能が正常に動作しません。

図 3-10 に Web Administration ツールのメインページを示します。



図 3-10 IBM Web Administration for i5/OS

インターフェースの上部にあるタブはさまざまなアクティビティーに関連します。特定のタブを選択すると、メイン・タブの下にサブタブの行が表示されます。「**Manage**」→「**HTTP Servers**」をクリックし、「Server」リストから「**ZENDCORE**」を選択すると（図 3-11 を参照）、IBM HTTP Server for i5/OS サーバー・インスタンスを表示、および変更できます。

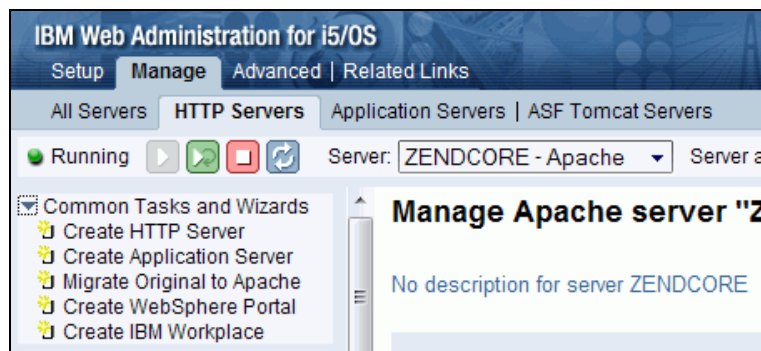


図 3-11 ZENDCORE Apache サーバー・インスタンスの管理

Web Administration ツールを使用して、他の HTTP サーバー・インスタンスを作成できます。

**ヒント** :ZENDCORE インスタンスはすべてのコンテンツを i5/OS PASE Apache HTTP Server に転送することから、静的コンテンツ (HTML ページ、画像、その他) 用として、サーバー・インスタンスをもう 1 つ作成することをお勧めします。ZENDCORE インスタンスは PHP ファイル専用として使用します。

さらに、ZENDCORE インスタンスを構成して、SSL (Secure Sockets Layer) または認証を使用することもできます。

IBM HTTP Server の構成について詳しくは、「*IBM HTTP Server (powered by Apache): An Integrated Solution for IBM eServer™ iSeries™ Servers (SG24-6716-02)*」を参照してください。

## 3.4 複数インスタンスの構成

System i5 ( およびその先行製品 ) の集中型マルチユーザー・マシンとしてのこれまでの流れから、複数インスタンスのサポートは i5/OS 製品の共通要件となっています。以降のページでは複数インスタンスに関わる、より一般的なシナリオを説明します。

複数の Web サーバーから PHP アプリケーションをホストできる機能は極めて重要で、その機能は完全にサポートされています。i5/OS 上で PHP の複数インスタンスを実装する方法は複数あります。各シナリオを実現するために必要な構成の大半が、i5/OS や PHP 環境ではなく、むしろ Apache に左右される点に注意してください。このことから、i5/OS 上の複数インスタンス構成は、Apache Web サーバーをサポートする他のオペレーティング・システムで同じタスクを実行することと、ほとんど変わりません。このシナリオ・リストは完全に網羅されたものではない点に注意してください。仮想ホストなどのテクノロジーを追加することにより、ここにリストされていないバリエーションも可能です。

現時点で、Zend Core for i5/OS では同一システム・パーティション上に製品を複数インストールすることはサポートされていません。「複数のインストール」とは、i5/OS の同一パーティション内に、複数のバージョンの製品がインストールされることを意味します。例えば、IBM WebSphere Application Server では、バージョン 5.0、6.0、および 6.1 のすべてを同時に、1 つの i5/OS パーティションにインストールすることが可能です。ほとんどのお客様にはこの機能は不要で、異なる方法で同じ結果が得られます ( 複数の i5/OS パーティション、複数のハードウェア・プラットフォームなど )。

**重要：**「apachectl」コマンドを使用して Web サーバー・インスタンスを開始、および停止するときは、QSECOFR としてサインオンする必要がある点に留意してください。他のユーザー・プロファイルを使用すると、PHP スクリプトによって実行されるジョブがある場合、Zend Core for i5/OS が NOBODY ユーザー・プロファイルに切り替えられません。

### 3.4.1 複数の i5/OS Apache インスタンス、1 台の PHP サーバー

このシナリオでは IBM HTTP Server for i5/OS (i5/OS Apache) の複数インスタンスがあり、これらのすべてのインスタンスが要求を PASE for i5/OS の 1 つの Apache/PHP インスタンスに転送することを想定します。

図 3-12 に、このシナリオを示します。

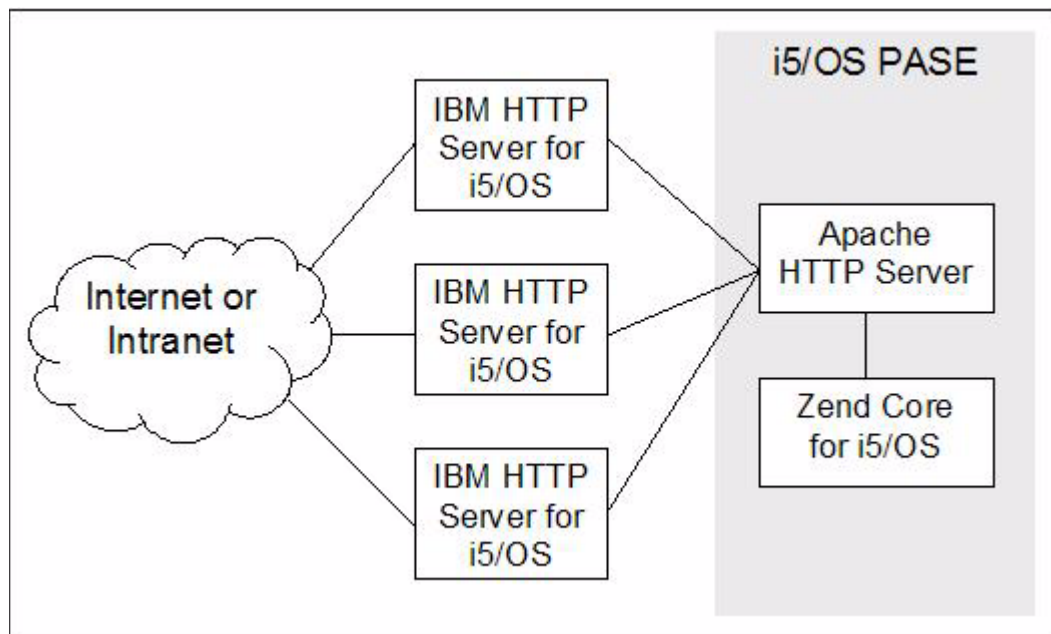


図 3-12 複数の i5/OS Apache インスタンス、1 台の PHP サーバー

このケースでは、3 つの IBM HTTP Server for i5/OS インスタンスがあることがわかります。このシナリオを実現するにはさまざまな方法があります。例えば、複数のポートを listen する 1 つの IBM HTTP Server for i5/OS インスタンスを構築します。このオプションが適した一例として、HTTP ( デフォルトでポート 80) および HTTPS ( デフォルトでポート 443) のいずれもインターネット / イントラネットに開き、これらの要求の処理は PHP サーバー 1 台だけで行う場合などが挙げられます。

図 3-13 ではポート 2001 で使用できる HTTP Server Administration ユーティリティを介して、この構成を行う方法を示します。この例では 2 番目のポート ( この場合はポート 666) が、ZENDCORE インスタンスに追加されています。デフォルトのリバース・プロキシ構成 ( Zend Core for i5/OS インストール・プログラムにより実現 ) は、既にこのインスタンスに構成されています。つまりポート 89 またはポート 666 で受信したすべての要求は、ポート 8000 を listen する内部 ( PASE for i5/OS) サーバーに送信されます。

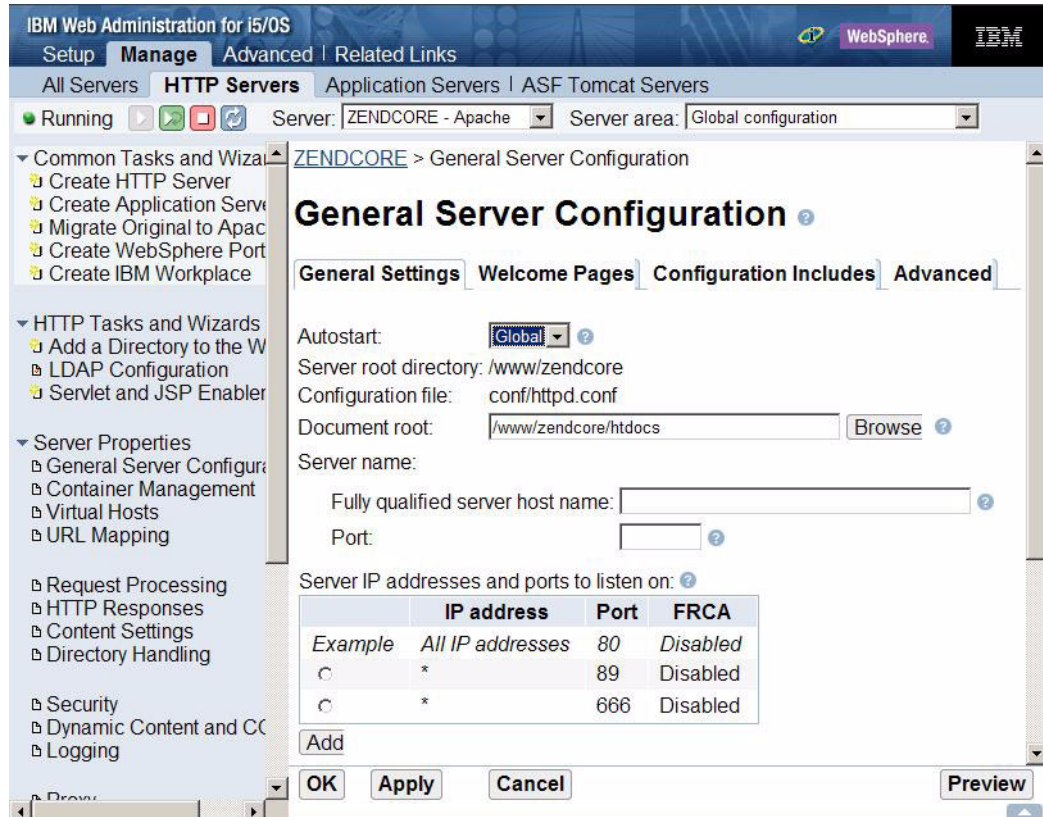


図 3-13 ZENDCORE インスタンスへの 2 つ目のポート (666) の追加

### 3.4.2 複数の i5/OS Apache インスタンス、複数の PHP サーバー

このシナリオでは、複数の IBM HTTP Server for i5/OS (i5/OS Apache) インスタンスが、PASE for i5/OS で実行する複数の Apache/PHP インスタンスに要求を転送します。このようなシナリオが役立つ理由として、いくつか挙げられます。例えば、他の企業向けのアプリケーションをホスティングしているような環境で、完全に他から隔離されたメモリー・スペースでアプリケーションを実行させる場合などです。他の理由としては、開発者やシステム管理者が PHP アプリケーションまたは PHP 環境を構築している物理的に同じパーティションに、実動環境の PHP サーバーを置く場合などです。つまり、開発者が PHP 環境を再始動させる必要がある変更を行っても、実動サーバーは再始動させることはできない場合などがこのケースに当たります。



図 3-14 では、このシナリオをどのように実現するかの概略図を示しています。

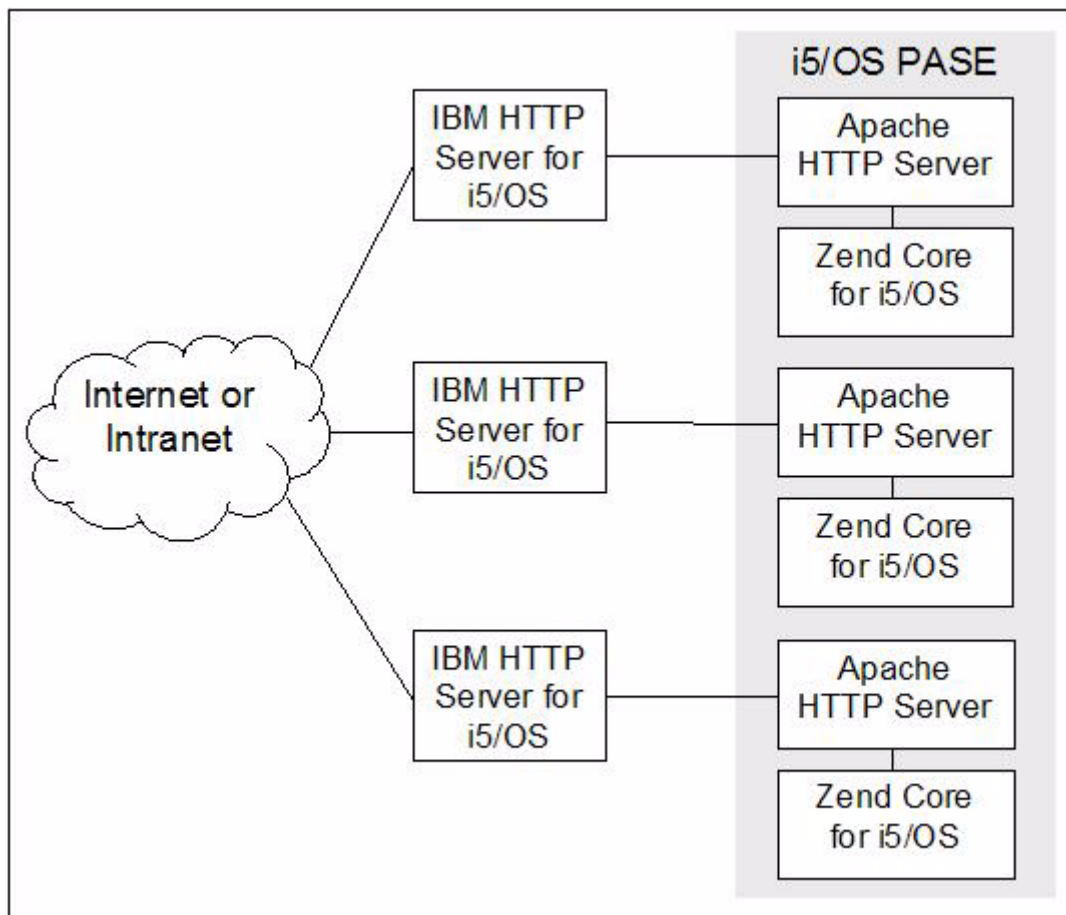


図 3-14 複数の i5/OS Apache インスタンス、複数の PHP サーバー

ここでも、図 3-14 に示されるように、同種の環境を構築するには、さまざまな手法が考えられます。『複数の i5/OS Apache インスタンス、1 台の PHP サーバー』(31 ページ) では 2 つ目の listen するポートを既存の ZENDCORE サーバー構成に追加する方法を示します。独立して開始、停止が可能で、異なるポートを listen する 2 番目の構成を作成する方法は当然他にもあります。

PASE for i5/OS に新しい Apache/PHP インスタンスを作成する、最も簡単な方法は以下のとおりです。

最初に、ディレクトリー「/usr/local/Zend/Core/etc」にある「php.ini」ファイルの新しいバージョンを作成します。ここでは「php.ini」をコピーして、「php2.ini」という新しいファイルを作成します。必要な変更はこの新しいファイルに行います。

続いて、ディレクトリー「/usr/local/Zend/apache2/conf」にある「httpd.conf」ファイルの新しいバージョンを作成します。ここでは「httpd.conf」をコピーして、「httpd2.conf」というファイルを同じディレクトリーに作成します。

次に「httpd2.conf」ファイルを編集し、固有になるよう構成します。以下に変更する行を示します。



例 3-1 新しい Apache/PHP インスタンスを作成するために、httpd2.conf で変更する行

```
PidFile logs/httpd2.pid
...
DocumentRoot "/www/ZendCore/htdocs2"
...
<Directory "/www/ZendCore/htdocs2">
...
```

当然、ご使用の構成や目的に応じて、他にも変更する必要がある場合は変更してかまいません。

新しい「php2.ini」ファイルにポイントさせるため、以下の行を「httpd2.conf」に追加する必要があります。

例 3-2 新しい Apache/PHP インスタンスを作成するために httpd2.conf に追加する行

```
### Start of Zend Core
PHPINIDir /usr/local/Zend/Core/etc/php2.ini
```

これで、以下のようなコマンドを実行することで 2 番目のインスタンスを開始できるようになりました。

```
/usr/local/Zend/apache2/bin/apachectl
-f /usr/local/Zend/apache2/conf/httpd2.conf
-k start
```

ご承知のとおり、インスタンスは以下のように「-k」フラグを追加することで、停止、開始、および再始動できます。

Stop	-k stop
Start	-k start
Restart	-k restart

**重要：**これらのコマンドを実行する場合は常に、「-f」フラグも指定して、使用する構成ファイルを指定するようにします。

これにより、IBM HTTP Server for i5/OS または PASE for i5/OS インスタンスの Apache/PHP を任意で、個別に開始、または停止できます。

### 3.4.3 1 つの i5/OS Apache インスタンス、複数の PHP サーバー

このシナリオでは、単体の IBM HTTP Server for i5/OS (インターネット/イントラネットに開いているポートは 1 つのみ) のインスタンスが、複数の Apache/PHP サーバー・インスタンスに要求を送信することを想定します。通常、どの Apache/PHP サーバーに要求をルートさせるかは、URL に基づいて決定されます。

図 3-15 に、このシナリオを実現する 1 つの方法を示します。

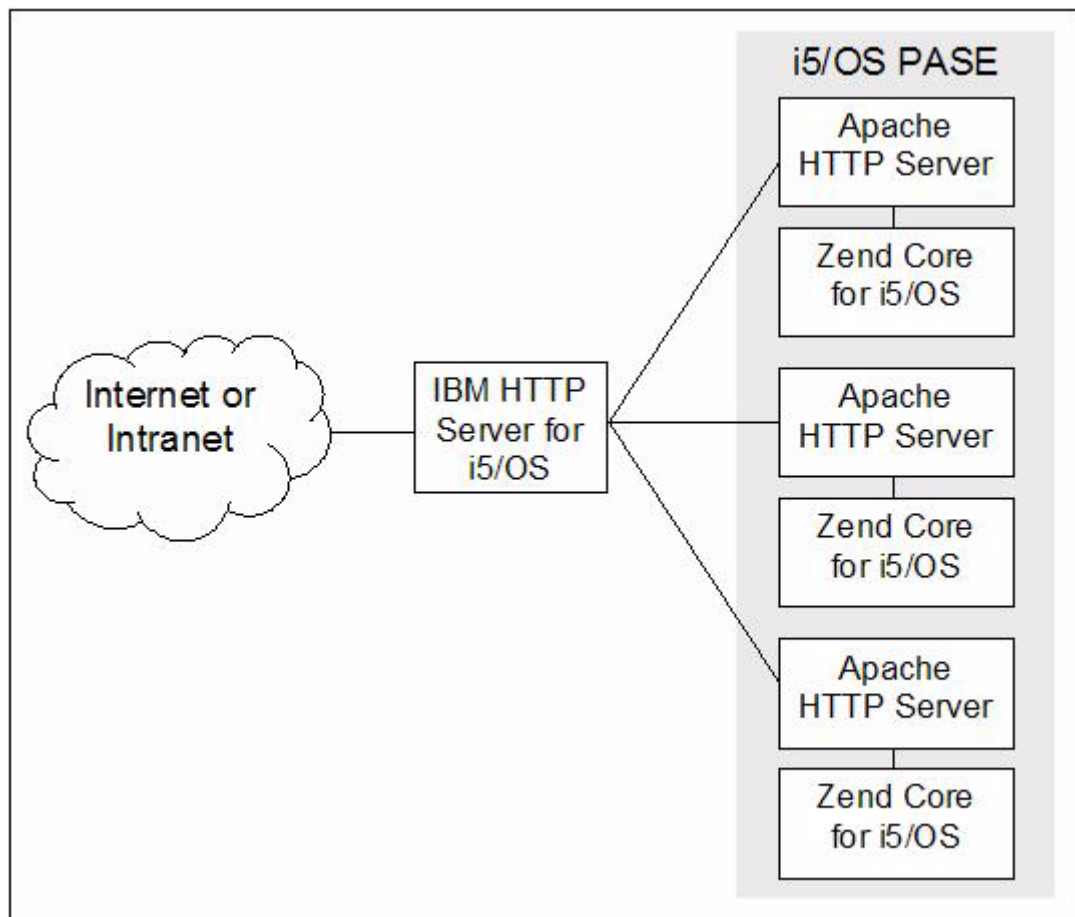


図 3-15 1 つの i5/OS Apache インスタンス、複数の PHP サーバー

両方の Apache/PHP サーバーの利点を活かすには、図 3-16 で示されるように、IBM Web Administration for i5/OS ツール (ポート 2001) を使用して、IBM HTTP Server for i5/OS インスタンスを構成します。

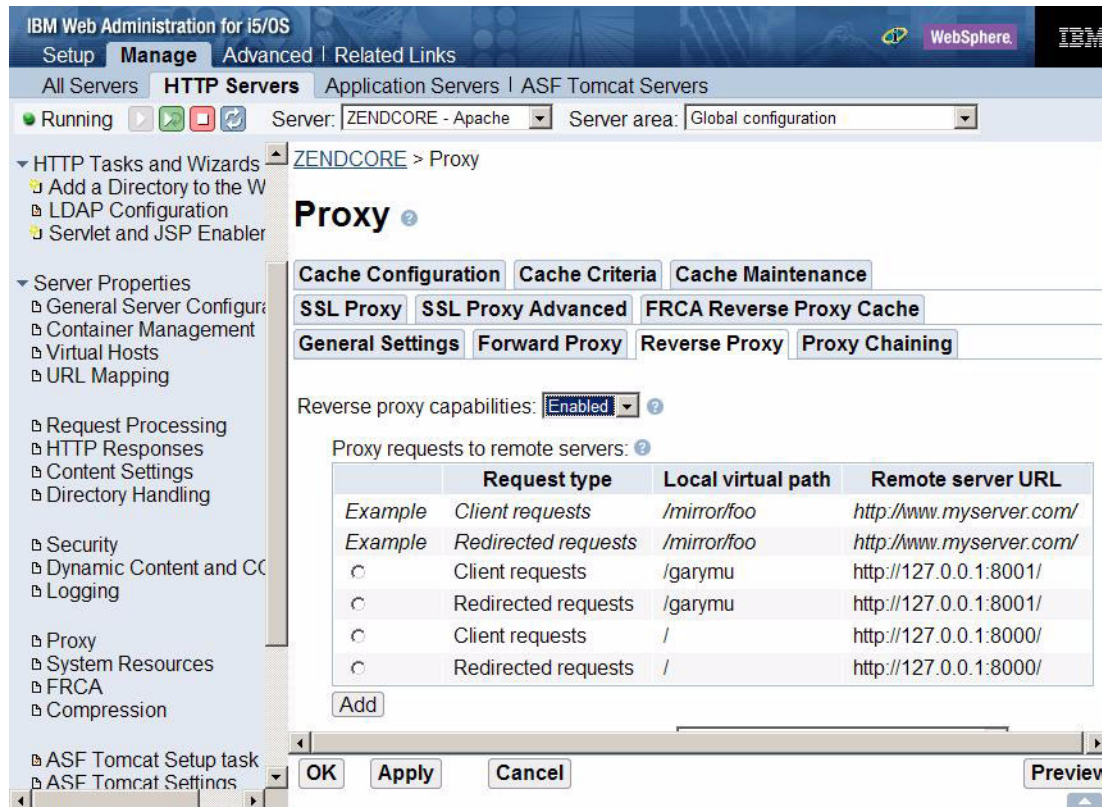


図 3-16 2 回目のリバース・プロキシ構成の追加

この場合、URL に「/garymu」を指定したすべての着信要求（例えば、`http://mysystem.ibm.com/garymu/myfile.php` など）はポート 8001 を listen する PASE for i5/OS の Apache/PHP インスタンスに送信されます。すべての他の要求（「/garymu」の指定のないもの）はポート 8000 のデフォルト・サーバーに送信されます。ここで、「/garymu」とポート 8001 の 2 つの構成項目は、「/」とポート 8000 より上位に指定する必要があります。これは「/」はすべての要求に当てはまるためです。もし「/」がリストの先頭にあると、すべての要求がそこで条件を満たしてしまい、他の一致条件まで到達せず、テストできません。

### 3.5 Zend Core for i5/OS のバックアップ

i5/OS の保管コマンドと復元コマンドを使用して、PHP 環境をバックアップおよび復元します。

Zend for i5/OS 製品ライブラリーとディレクトリーを保管および復元するには、SAVLICPGM と RSTLICPGM を使用します。

バックアップにはオブジェクトの保管 (SAV) コマンドおよびライブラリー保管 (SAVLIB) コマンドを使用し、復元操作にはオブジェクトの復元 (RST) コマンドおよびライブラリー復元 (RSTLIB) コマンドを使用します。各種コマンドについて詳しくは、V5R4 iSeries の Information Center の「システム管理：バックアップおよび回復」を参照してください。

<http://publib.boulder.ibm.com/infocenter/iseries/v5r4/topic/rzahg/rzahgbackup.htm>

特に、以下のオブジェクトは定期的にバックアップしてください。

- ▶ /www/zendcore/htdocs/\*

このディレクトリー ( およびすべてのサブディレクトリー ) には PHP ファイルと他の Web ファイルが含まれます。

- ▶ /www/zendcore/conf/httpd.conf


IBM HTTP Server for i5/OS サーバー・インスタンス (ZENDCORE) の構成を変更した場合は、その構成ファイルもバックアップすることをお勧めします。

- ▶ /usr/local/Zend/apache2/conf/\*

i5/OS PASE Apache HTTP Server をカスタマイズした場合は、このディレクトリーの構成ファイルをバックアップします。

- ▶ /usr/local/Zend/Core/etc/\*

Zend Core と PHP 構成への変更は、このディレクトリーのファイルに格納されます。



## アプリケーション開発

本章では、Web スクリプト言語としておよび汎用スクリプト言語としての PHP の使用方法を説明します。PHP を使用する動的アプリケーションを開発する場合に使用する一般的なソフトウェアおよび概念について、および i5/OS のデータ、オブジェクト、およびプログラムについて説明します。

本章では、以下のトピックについて説明します。

- ▶ PHP および Web 開発
- ▶ コマンド行スクリプト言語としての PHP
- ▶ 開発ツール

## 4.1 PHP および Web 開発

PHP は主に、Web アプリケーションの開発に使用されます。Web アプリケーションは、一般に 3 層構造をしています。この 3 層構造については、以降の本書の理解に欠かせないため、本章で説明します。

ソフトウェア・サービスのそれぞれのタイプについて、IBM 環境における一般的なアーキテクチャー上に構築した Web アプリケーションというコンテキストで詳細に説明します。

### ▶ プレゼンテーション層

この層は、ユーザー・インターフェースとして機能します。Web サーバーが、ユーザーからの要求を処理します。Web サーバーは、ファイル・システムから文書を取得することにより Web サーバー自体がユーザーの要求に応じられるのか、処理を完了するにはロジック層にある他のプログラムの手助けが必要であるのかを、Web サーバーの構成を基に決定します。

### ▶ ロジック層

ロジック・エンジンは、Web サーバーが要求を委任できるインターフェースを提供します。ロジック・エンジンは、アプリケーションのビジネス・ロジックを実行します。ロジック・エンジンは、入力を受け入れ、受け入れた入力に対する操作を実行し、動的に作成した出力を Web サーバーに返します。Web サーバーは、この出力をクライアントに返します。

### ▶ データ層

データ・ストアは、永続情報を管理し、規則と条件に基づいてこの情報を変更または取得するインターフェースを提供します。

IBM DB2、Informix®、Cloudscape などのデータベース・サーバーは、データ層のサービスを提供します。Apache HTTP Server およびその IBM 強化版である IBM HTTP Server は、Web サーバーの役割を務めます。PHP は、データベースに格納された情報を基に、Web サーバーに返す応答を生成することを主とする、ビジネス・ロジック・エンジンとして使用されます。

図 4-1 は、Web アプリケーションのインフラストラクチャー・コンポーネントおよび HTTP の要求と応答のサイクルを示します。

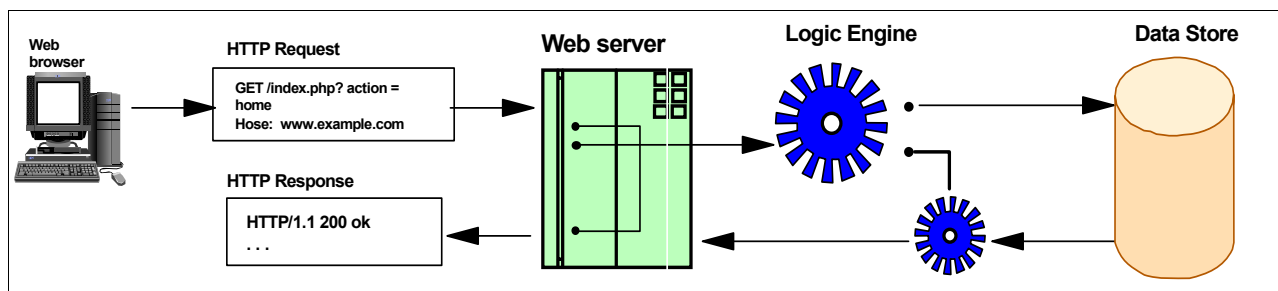


図 4-1 Web アプリケーション・インフラストラクチャー

### 4.1.1 簡単な Web アプリケーションの例

PHP コードを HTML に埋め込むことにより、Web アプリケーションの作成をすぐに開始できます。埋め込んだ PHP コードは、個別の PHP ファイルのように再使用可能ではありませんが、小規模な Web サイトを開発する場合は、埋め込み方式のほうが迅速です。

#### 例 4-1 PHP を埋め込んだ HTML

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<title>IBM Redbook: Hello World</title>
</head>
<h3>My first PHP web application</h3>
<p>
<?php echo "Hello, World" ?></p>
<body>
</body>
</html>
```

Zend Studio for i5/OS、FTP など i5/OS にファイルを転送する任意の手法を使用して、このファイルを /www/zendcore/htdocs/ ディレクトリーに配置します。これで、図 4-2 に示すように、ご使用の Zend Core for i5/OS インストール済み環境の IP アドレスを使用し、Web ブラウザー経由でこの PHP アプリケーションにアクセスできます。

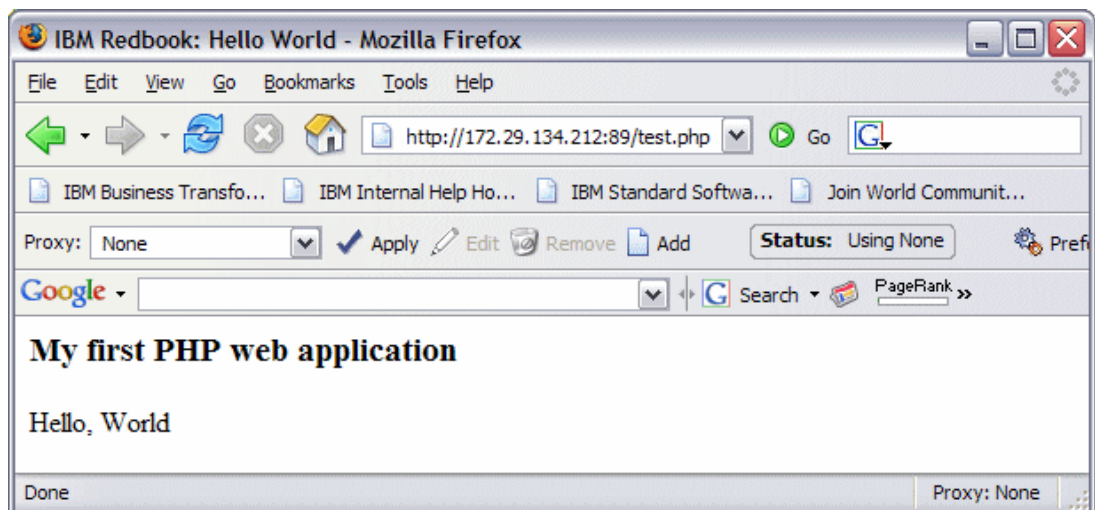


図 4-2 PHP 版 Hello World

以上で、最初の PHP アプリケーションの作成とテストはすべて完了します。

## 4.2 コマンド行スクリプト言語としての PHP

PHP は主に Web アプリケーションの開発に使用されますが、CL 代替のコマンド行スクリプト言語としても使用できます。PHP には、対話式、一度に 1 行、スクリプト内、という 3 つのコマンド行モードがあります。

コマンド行ツールとして PHP を使用する場合の詳細については、次の Web アドレスを参照してください (i5/OS 限定の内容ではありません)。

<http://www.php.net/manual/en/features.commandline.php>

## 4.2.1 PHP の対話モード

対話モードでは、PHP コードの一部をテストできます。PHP を対話モードで実行するには、i5/OS PASE で次のコマンドを入力します。

```
/usr/local/Zend/Core/bin/php -a
```

対話モードの PHP では、`<?php` および `?>` を区切り文字として囲んで、PHP 文を記述できます。例：

```
<?php for ($i=0; $i<10; $i++) { print "Counting $i\n"; } ?>
```

通常、PHP の出力は、バッファに配置されます。次のコマンドを入力すると、バッファリングを無効にできます。

```
<?php while (@ob_end_clean()); ?>
```

PHP の対話モードを終了するには、次のコマンドを実行します。

```
<?php exit ?>
```

図 4-3 は、対話モードでの PHP の使用例を示します。

```
/QOpenSys/usr/bin/-sh

Interactive mode enabled

> <?php while (@ob_end_clean()); ?>
> <?php for ($i=0;$i<10;$i++) { print "Counting $i\n"; } ?>
Counting 0
Counting 1
Counting 2
Counting 3
Counting 4
Counting 5
Counting 6
Counting 7
Counting 8
Counting 9

==> <?php exit ?>

F3= 終了   F6= 印刷   F9= コマンドの複写  F11= 切り捨て / 折り返し
F13= 消去  F17= 先頭   F18= 最後   F21=CL コマンド入力
```

図 4-3 PHP 対話モードの例



## 4.2.2 PHP 1 行スクリプト (通称 oneliner)

PHP を 1 行スクリプト・インタプリタとして使用するには、-r コマンド行オプションを使用する必要があります。例：

```
/usr/local/Zend/Core/bin/php -r 'for ($i=0;$i<10;$i++) { print"Counting $i\n"; }'
```

この例を実行すると、図 4-4 のように出力されます。

```
/QOpenSys/usr/bin/-sh

$
> /usr/local/Zend/Core/bin/php -r 'for ($i=0;$i<10;$i++) { print "Counting $i\n"; }'
Counting 0
Counting 1
Counting 2
Counting 3
Counting 4
Counting 5
Counting 6
Counting 7
Counting 8
Counting 9
$

==>

F3= 終了   F6= 印刷   F9= コマンドの複写  F11= 切り捨て / 折り返し
F13= 消去  F17= 先頭   F18= 最後   F21=CL コマンド入力
```

図 4-4 PHP 1 行スクリプトの出力

## 4.2.3 PHP とスクリプト記述

PHP でコマンド行スクリプトを実行するには、例 4-2 に示すように PHP コードを記述したファイルを作成し、次のコマンドにより実行します (この例では myfile.php を使用します)。

```
/usr/local/Zend/Core/bin/php myfile.php
```

例 4-2 PHP スクリプトの例 (myfile.php)

```
<?php
for ($i=0;$i<10;$i++)
{
    print "Counting $i\n";
}
?>
```

出力は、図 4-5 のようになります。

```
/QOpenSys/usr/bin/-sh

kost      myfile.php  pero.php   test.php   testzend.php
$
> /usr/local/Zend/Core/bin/php myfile.php
Counting 0
Counting 1
Counting 2
Counting 3
Counting 4
Counting 5
Counting 6
Counting 7
Counting 8
Counting 9
$

===>

F3= 終了   F6= 印刷   F9= コマンドの複写  F11= 切り捨て / 折り返し
F13= 消去  F17= 先頭  F18= 最後   F21=CL コマンド入力
```

図 4-5 PHP スクリプトの出力

PHP ファイルの構文を検査するのみの場合は、次の -l コマンド行オプションを使用します。

```
/usr/local/Zend/Core/bin/php -l myfile.php
```

構文エラーが検出されなかった場合は、図 4-6 のような出力になります。

```
/QOpenSys/usr/bin/-sh

$
> cd /www/ZendCore/htdocs/
$
> /usr/local/Zend/Core/bin/php -l myfile.php
No syntax errors detected in myfile.php
$

===>

F3= 終了   F6= 印刷   F9= コマンドの複写  F11= 切り捨て / 折り返し
F13= 消去  F17= 先頭  F18= 最後   F21=CL コマンド入力
```

図 4-6 PHP の構文検査

## 4.2.4 CL からの PHP 呼び出し

CL アプリケーションから PHP スクリプトを呼び出すことができます。次の例は、組み込みの "mail" 関数を使用して E メールを送信する PHP スクリプトの起動方法を示します。

例 4-3 は、実際に E メール送信操作を実行する PHP スクリプトを示します。もちろん、差出人および宛先を実際の値に置換する必要がありますし、PHP 構成に SMTP が正しく構成されている必要があります。

*例 4-3 PHP スクリプト : emailme.php*

---

```
<?php
// "from" ユーザーを設定します (オプション)
ini_set("sendmail_from","sender@example.org");

// パラメーター : recipient, subject, body
mail("me@example.org", "Sent from CL", "CL can call PHP scripts.");

?>
```

---

例 4-4 は、i5/OS の PASE を通じて先の PHP スクリプトを起動する、CALLPHP としてコンパイルされる制御言語プログラムを示します。

*例 4-4 制御言語プログラム : CALLPHP*

---

```
CALL      PGM(QP2SHELL) +
          PARM('/usr/local/zend/Core/bin/php' +
              '/www/zendcore/htdocs/emailme.php')
```

---

CALLPHP 制御言語プログラムは、他のプログラムを i5/OS で起動する場合と同じ方法で起動できます。

CALL CALLPHP

例 4-5 は、この制御言語プログラムを呼び出した結果を示します。

*例 4-5 CALLPHP 制御言語プログラムの起動結果*

---

```
From: sender@example.org
To: <me@example.org>
Subject: Sent from CL
```

```
CL can call PHP scripts.
```

---

## 4.3 開発ツール

PHP は、メモ帳などのテキスト・エディターで編集できますが、強力な統合開発環境 (IDE) を使用すれば開発速度を大幅に向上できます。PHP が普及しているおかげで、さまざまな IDE が使用できます。表 4-1 (46 ページ) を参照して、必要に応じた適切なツールを選択してください。

表 4-1 一般的な PHP 開発環境のリスト

名称	ライセンス	ソース管理	ファイル転送	Web アドレス
Zend Studio for i5/OS	商用 (Zend Core for i5/OS をご利用の i5/OS ユーザーの場合はフリー)	CVS/SVN	FTP	<a href="http://www.zend.com">http://www.zend.com</a>
Eclipse	フリー	CVS および SVN (プラグインを使用)	FTP、Web DAV、SFTP	<a href="http://www.eclipse.org">http://www.eclipse.org</a>
Maguma	フリー / 商用	CVS	FTP	<a href="http://www.maguma.com">http://www.maguma.com</a>
NuSphere PhpED	商用	CVS	FTP、SFTP、Web DAV	<a href="http://www.nusphere.com">http://www.nusphere.com</a>

他のさまざまな開発環境については、次のサイトを参照してください。

<http://www.php-editors.com/review/>

このサイトでは、商用およびフリーの IDE およびエディターがレビューされており、製品の実ユーザーからのフィードバックが紹介されています。

### 4.3.1 Zend Studio for i5/OS

Zend Studio for i5/OS (図 4-7) は、Zend Core のデフォルト開発環境です。Zend Studio for i5/OS には、内蔵デバッガー、CVS/SVN、および FTP サポートが付属しています。

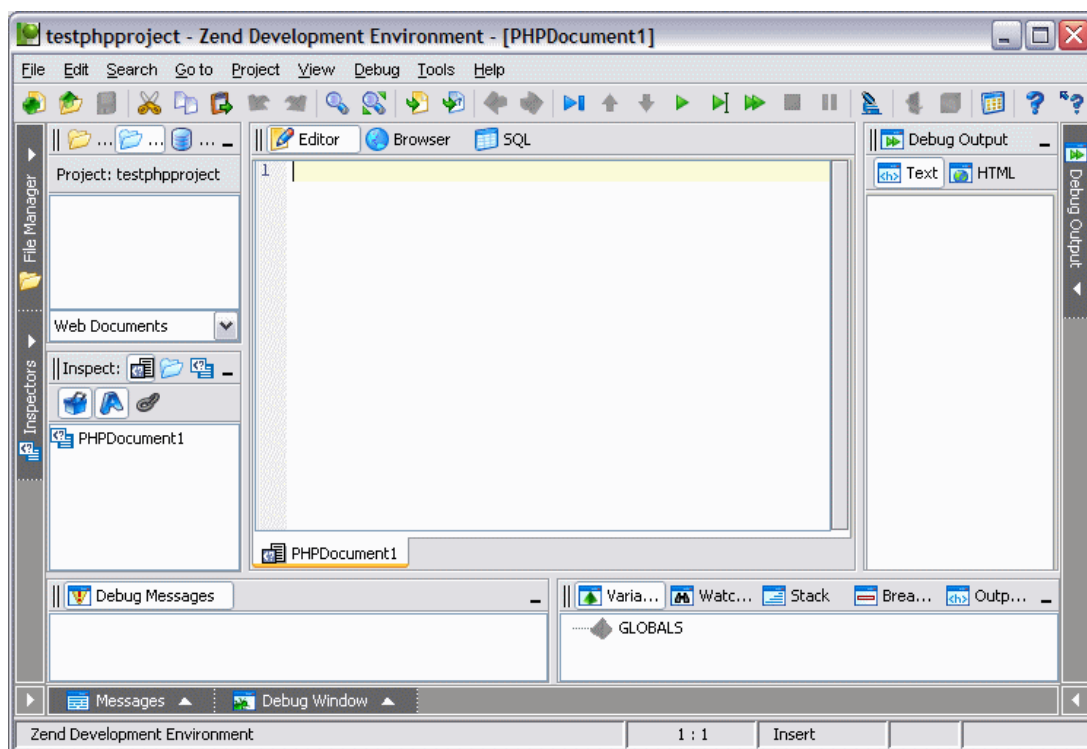


図 4-7 Zend 開発環境: Zend Studio for i5/OS

Zend Studio for i5/OS は、Linux、Mac OS X、および Windows® プラットフォームで使用できます。Zend Studio for i5/OS は、Zend Core for i5/OS をダウンロードしてご利用のお客様の場合はフリーであり、PHP を開発する場合の確かな選択肢です。Zend Studio for i5/OS は Zend Web サイトからダウンロードできます。

[https://www.zend.com/core/oem\\_registration.php?access\\_code=IBMi5OSZend](https://www.zend.com/core/oem_registration.php?access_code=IBMi5OSZend)

詳細については、次の Web アドレスを参照してください。

[http://www.zend.com/products/zend\\_core/zend\\_for\\_i5\\_os](http://www.zend.com/products/zend_core/zend_for_i5_os)

## 4.3.2 Eclipse

Eclipse は Java で記述されたオープンかつ拡張可能なプラットフォームを基にしています。元々 Eclipse は、Java による開発での使用がほとんどでしたが、PHP をサポートするよう Eclipse を拡張する、多数の優れた PHP プラグインが存在します。Eclipse は、Windows、Linux などさまざまなプラットフォームで使用できます。図 4-8 は、PHPeclipse プラグインを導入した Eclipse を示します。

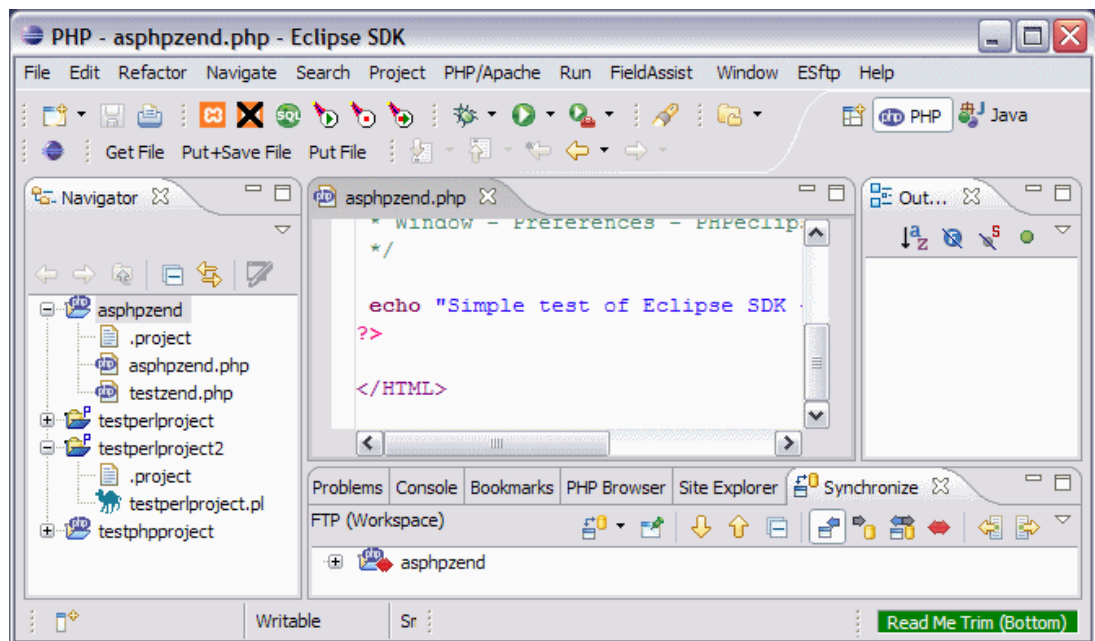


図 4-8 PHPeclipse プラグインを導入した Eclipse

PHP をサポートする Eclipse プラグインは、商用のものと、フリーのものがあります。広く普及したフリーの PHP プラグインの 1 つが PHPeclipse です。PHPeclipse は次のサイトからダウンロードできます。

<http://phpeclipse.de/>

広く普及した商用 PHP プラグインの 1 つが TruStudio です。TruStudio にはフリー・バージョンもあります。TruStudio の詳細については、次の Web アドレスを参照してください。

<http://www.xored.com/trustudio>

### 4.3.3 NuSphere PhpED

NuSphere PhpED (図 4-9) は、堅固な PHP 統合開発環境です。

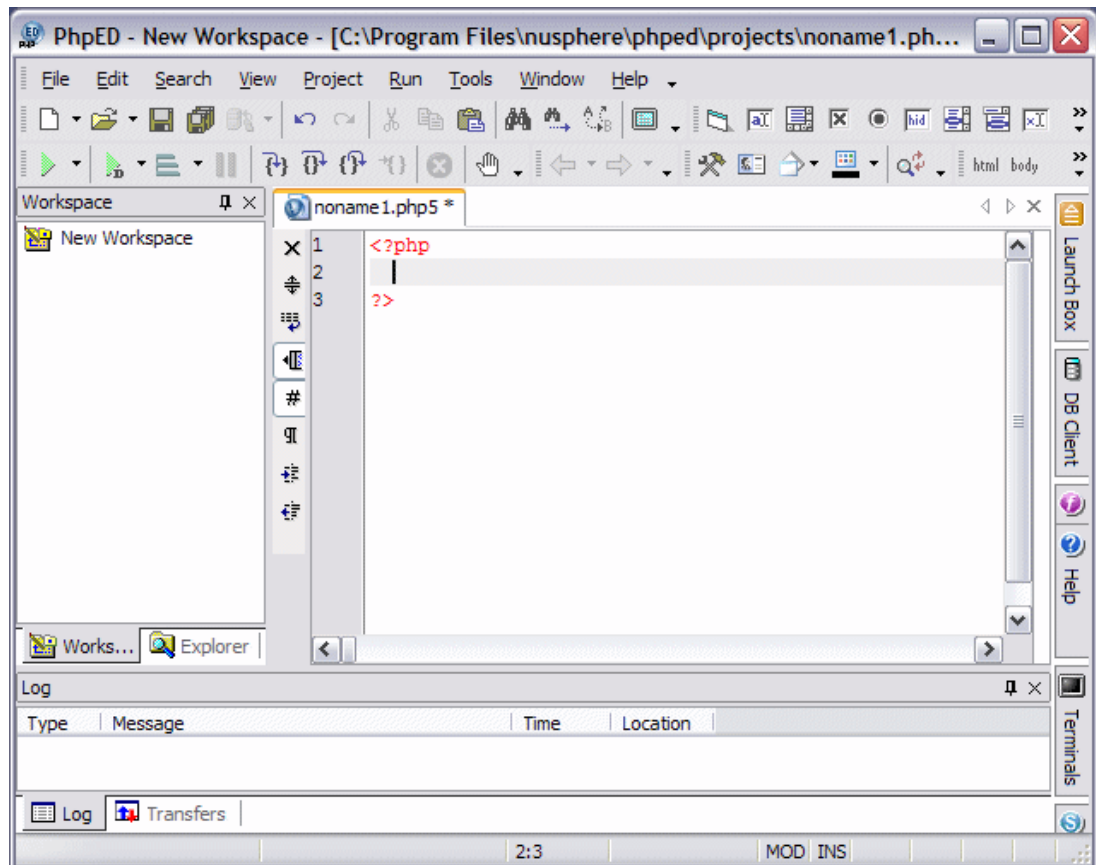


図 4-9 NuSphere PhpED

NuSphere PhpED は、Windows プラットフォームでのみ使用できます。次の Web サイトから評価版をダウンロードできます。

<http://www.nusphere.com/download.php.ide.htm>

### 4.3.4 Maguma

Maguma には PHP 用の IDE が複数バージョン存在します。図 4-10 に示すフリー・バージョンの Maguma Open Studio と商用バージョンの Maguma Workbench があります。

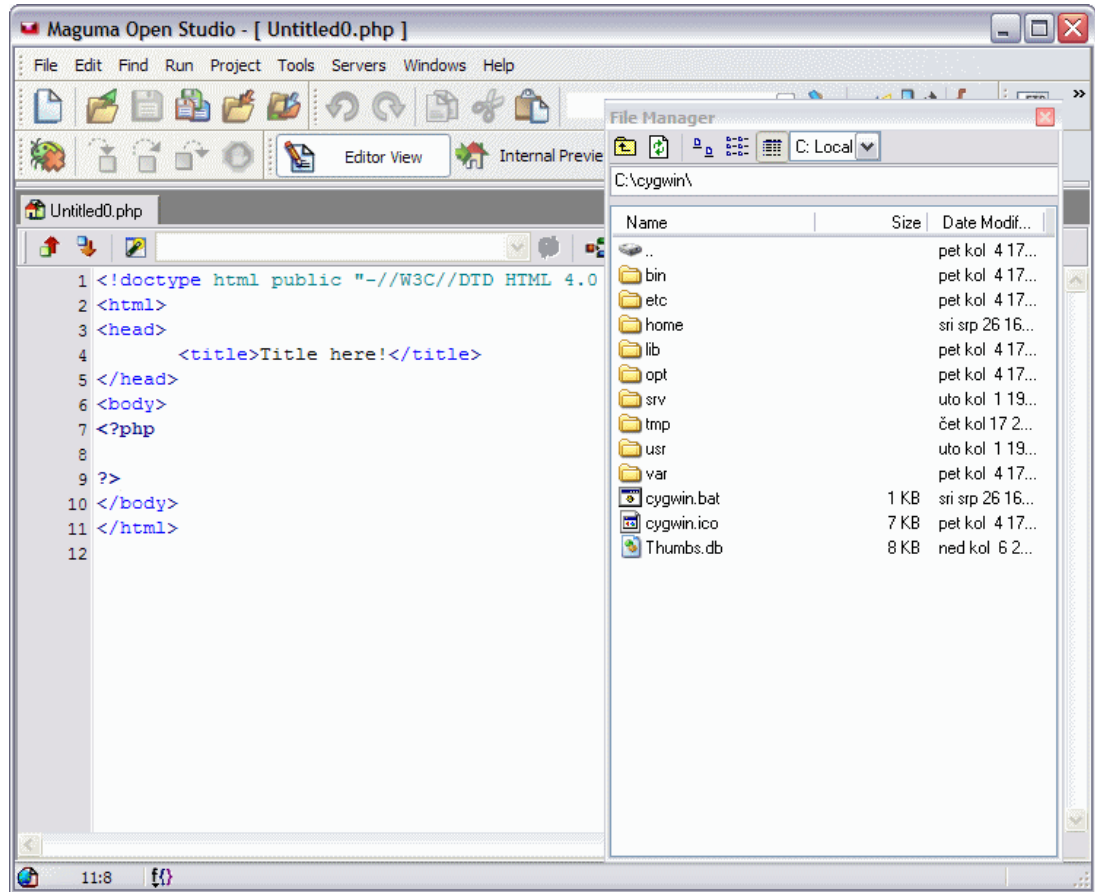


図 4-10 Maguma Open Studio

Maguma 製品は、Windows および Linux で使用できます。次の Web ページからダウンロードできます。

<http://www.maguma.com/en/download.html>

## 4.4 i5 PHP API ツールキットの関数

i5 PHP API ツールキットを使用すると、PHP からネイティブ i5/OS リソースにアクセスできます。i5 PHP API ツールキットにより、特定の i5/OS 関数を呼び出したり、i5/OS データ・ファイルへのネイティブ・アクセスを行ったりできます。

**重要：**この機能は、さまざまな名称で呼ばれます。例えば、Zend 構成メニューでは「i5\_COMD service」です。

「Zend Core for i5/OS User Guide」には、i5 PHP API ツールキットに含まれるすべての i5/OS API の説明があります。ここでは詳細な説明は省いて、ツールキットにより提供される機能の概要を示します。

### 4.4.1 主な関数

本書で取り上げる各関数によって実行できる機能の概要を以下に示します。

- ▶ 接続の管理
  - i5\_connect
  - i5\_close
  - i5\_adapt\_authority
  - i5\_error
  - i5\_errormsg
- ▶ CL 呼び出し
  - i5\_command
- ▶ プログラム呼び出し
  - i5\_program\_prepare
  - i5\_program\_call
  - i5\_program\_close
- ▶ データ・リトリーブ
  - i5\_fetch\_array
  - i5\_fetch\_assoc
  - i5\_fetch\_object
  - i5\_fetch\_row
  - i5\_info
  - i5\_field\_len
  - i5\_field\_name
  - i5\_field\_scale
  - i5\_field\_type
  - i5\_list\_fields
  - i5\_num\_fields
  - i5\_result
- ▶ ネイティブ・ファイル・アクセス
  - i5\_open
  - i5\_addnew
  - i5\_edit
  - i5\_delete
  - i5\_cancel\_edit
  - i5\_setvalue
  - i5\_update
  - i5\_range\_from
  - i5\_range\_to
  - i5\_range\_clear
  - i5\_data\_seek
  - i5\_seek
  - i5\_bookmark
  - i5\_free\_file
  - i5\_new\_record
  - i5\_update\_record
  - i5\_delete\_record
  - i5\_get\_keys
- ▶ システム値
  - i5\_get\_system\_value



- ▶ データ域
  - i5\_data\_area\_create
  - i5\_data\_area\_read
  - i5\_data\_area\_write
  - i5\_data\_area\_delete
- ▶ データ・キュー
  - i5\_dtaq\_prepare
  - i5\_dtaq\_receive
  - i5\_dtaq\_send
  - i5\_dtaq\_close
- ▶ ジョブ・ログ・リスト
  - i5\_jobLog\_list
  - i5\_jobLog\_list\_read
  - i5\_jobLog\_list\_close
- ▶ オブジェクト・リスト
  - i5\_objects\_list
  - i5\_objects\_list\_read
  - i5\_objects\_list\_close
- ▶ スプール・ファイル
  - i5\_spool\_list
  - i5\_spool\_list\_read
  - i5\_spool\_get\_data
  - i5\_spool\_list\_close
- ▶ ユーザー・スペース
  - i5\_userspace\_create
  - i5\_userspace\_prepare ( 新規 )
  - i5\_userspace\_get
  - i5\_userspace\_put ( 新規 )

i5/OS リソースにアクセスするには、まず i5/OS に接続する必要があります。例 4-6 は、接続方法を示します。接続は、i5 PHP API ツールキットの大部分の呼び出しに必要です。

例 4-6 i5 PHP API ツールキットからの i5/OS への接続

---

```
<HTML>
<?php
/* サーバーに接続する */
$conn = i5_connect("localhost", "PHPUSER", "MYPASSWORD");
if (!$conn)
    die("<br>Connection using \"localhost\" with USERID and PASSWORD failed. Error number
    =".i5_errno()." msg=".i5_errormsg())."<br>");
else
    echo "<br>Connection using \"localhost\" with USERID and PASSWORD OK!<br>\n";
/* 接続を閉じる */
i5_close($conn);
?>
</HTML>
```

---

**重要：**バージョン 1.6 より前のバージョンでは、ユーザー ID とパスワードの両方を大文字で指定する必要がありました。バージョン 1.6 以降では、この制限がなくなりました。

接続が確立されたら、i5/OS リソースにアクセスできます (この例では、ユーザー PHPUSER およびパスワード MYPASSWORD により接続します)。例 4-7 (52 ページ) は、i5\_command を使用して CL 呼び出しを実行する方法を示します。この例では、ネットワーク属性検索コマンド (RTVNETA) を呼び出して、コマンドの出力を表示しています。

#### 例 4-7 i5\_command の例

---

```
<HTML>
<?php
/* サーバーに接続する */
$conn = i5_connect("localhost", "PHPUSER", "MYPASSWORD");
if (!$conn)
    die("<br>Connection using \"localhost\" with USERID and PASSWORD failed. Error number
    =".i5_errno().". msg="i5_errormsg())."<br>");
else
    echo "<br>Connection using \"localhost\" with USERID and PASSWORD OK!<br>\n";

/* ネットワーク属性検索コマンドを呼び出す */
$ret = i5_command("rtvneta", array(), array("sysn" => "sysn", "lclnetid" => "lclnet"));
if (!$ret) die("<br>rtvneta command failed. errno=".i5_errno().". msg="i5_errormsg());
print "<h1><b>Results of \"rtvneta\" command </b></h1><br>";
print "System Name : $sysn<br>";
print "Local Net ID : $lclnet<br>";

/* 接続を閉じる */
i5_close($conn);
?>
</HTML>
```

---

例 4-8 は、PHP での i5/OS プログラム呼び出しの実行方法を示します。まず i5\_program\_prepare によりプログラム呼び出しの準備を行い、次に i5\_program\_call を呼び出してプログラム呼び出しを実行します。i5\_program\_close を呼び出して終了します。

#### 例 4-8 プログラム呼び出しの例

---

```
<HTML>
<?php
/* サーバーに接続する */
$conn = i5_connect("localhost", "PHPUSER", "MYPASSWORD");
if (!$conn)
    die("<br>Connection using \"localhost\" with USERID and PASSWORD failed. Error number
    =".i5_errno().". msg="i5_errormsg())."<br>");
else
    echo "<br>Connection using \"localhost\" with USERID and PASSWORD OK!<br>\n";

$description = array(
    array("Name" => "FIRST", "IO" => I5_IN, "Type" => I5_TYPE_CHAR, "Length" => "15"),
    array("Name" => "LAST", "IO" => I5_IN, "Type" => I5_TYPE_CHAR, "Length" => "15"),
    array("Name" => "ACCOUNT", "IO" => I5_OUT, "Type" => I5_TYPE_CHAR, "Length" => "15"),
    array("Name" => "AMOUNT", "IO" => I5_INOUT, "Type" => I5_TYPE_PACKED, "Length" => "5.2")
);

$pgm = i5_program_prepare("PHPLIB/INCRAMT", $description);
if (!$pgm) die("<br>Program prepare error. Error number =".i5_errno().". msg="i5_errormsg());
$parmIn = array(
    "FIRST" => $_POST["first"],
    "LAST" => $_POST["last"],
    "AMOUNT" => $_POST["amount"]
);
$parmOut = array(
    "FIRST" => "FIRST",
    "LAST" => "LAST",
    "ACCOUNT" => "ACCOUNT",
    "AMOUNT" => "AMOUNT"
);
```

```

$ret = i5_program_call($pgm, $parmln, $parmOut);
if (!$ret) die("<br>Program call error. Error number=".i5_errno().". msg=".i5_errormsg());
echo "<BR>FIRST :". $FIRST;
echo "<BR>LAST : $LAST";
echo "<BR>AMOUNT : $AMOUNT";
echo "<BR>ACCOUNT : $ACCOUNT";

/* プログラム呼び出しを閉じる */
i5_program_close($pgm);

/* 接続を閉じる */
i5_close($conn);
?>
</HTML>

```

---

例 4-9 は、i5 呼び出しを実行する別の例です。この例は、特定のシステム値の取得方法を示します。この例では、QDATE の値の取得を試みます。

#### 例 4-9 システム値の取得例

```

<HTML>
<?php
/* サーバーに接続する */
$conn = i5_connect("localhost", "PHPUSER", "MYPASSWORD");
if (!$conn)
    die("<br>Connection using \"localhost\" with USERID and PASSWORD failed. Error number
    =".i5_errno().". msg=".i5_errormsg())."<br>";
else
    echo "<br>Connection using \"localhost\" with USERID and PASSWORD OK!<br>\n";

/* システム日付の値を取得する */
print "Date is: ".i5_get_system_value("QDATE");

/* 接続を閉じる */
i5_close($conn);

```

---

他の例および使用可能な関数の完全なリファレンスについては、「Zend Core for i5/OS User Guide」を参照してください。

## 4.5 XML サポート

PHP が非常に普及している理由の 1 つは、強力でありながら、簡単に使用できることです。このことを裏付ける、短い例を次に示します。最初に、ラストネームおよびファーストネームをいくつか含む簡単な XML ファイルを示します (例 4-10)。このファイルを myXML.xml とします。

#### 例 4-10 ラストネームおよびファーストネームを含むサンプル XML ファイル

```

<?xml version="1.0" encoding="iso-8859-1"?>
<myXML>
<person>
<name>Anderson</name>
<firstname>Melissa</firstname>
</person>
<person>
<name>Mullen-Schultz</name>
<firstname>Gary</firstname>
</person>

```

```
<person>
<name>Kosturjak</name>
<firstname>Vlatko</firstname>
</person>
</myXML>
```

---

次に、若干の PHP スクリプトを含んだサンプル HTML 文書を示します (例 4-11)。この例では、ファイルが存在することを確認してから、1 文でファイルをロードします。次に、単純に終わりまでループしながら、XML ファイルから読み取った要素、この場合は本書の執筆者の名前を出力します。

#### 例 4-11 XML ファイルの内容を表示する HTML/PHP 文書

---

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<title>Read an XML file</title>
</head>
<body>
<h3>IBM Redbook authors</h3>
<?php
if (file_exists('myXML.xml')) {
    $x = simplexml_load_file('myXML.xml');
    foreach ($x->person AS $person) {
        printf($person->name . ' ' . $person->firstname . '<br>');
    }
}
else {
    exit('Failed to open myXML.xml.');
```

---

PHP 5 で導入された SimpleXML 拡張モジュールを使用すると、一般的な XML 操作を簡略化できます。SimpleXML に **simplexml\_load\_file()** というコマンドがあります。先の例では、XML データを配列にロードするときに使用しています。例 4-12 (54 ページ) は、**simplexml\_load\_file()** を使用して、RSS フィード・パーサーを実装する方法を示します。このパーサーは、IBM というタグの付いた technorati からのブログ・ポストを取り込み、RSS を使用して表示します。

#### 例 4-12 PHP の simplexml\_load\_file() による RSS 実装

---

```
<HTML>
<?php
$rssurl="http://feeds.technorati.com/feed/posts/tag/ibm";
foreach(simplexml_load_file($rssurl)->channel->item as $i) {
    echo '<p><a href="'. $i->link. '">'. $i->title. '</a><br />';
    echo $i->description. '</p>';
}
?>
</HTML>
```

---

**注:** 例 4-12 は、一例であり、実動向けではありません。実動するには、例えば、キャッシングを実装する必要性が生じる可能性があります。

## 4.6 XML-RPC サポート

XML-RPC は、非常にシンプルなりモート・プロシージャー・コール・プロトコルです。XML-RPC では、呼び出しのエンコードに XML を使用し、転送メカニズムとして HTTP を使用します。XML-RPC の詳細については、次の Web アドレスを参照することをお勧めします ( ケース・センシティブです )。

<http://en.wikipedia.org/wiki/XML-RPC>

簡単に言うと、XML-RPC は、Simple Object Access Protocol (SOAP) の簡易な代替方式として使用できます。XML-RPC は、SOAP では複雑すぎるかもしれませんが、リモート・プロシージャー・コールが必要な、小規模のプロジェクトに最適です。SOAP 技術の詳細については、4.7 『SOAP サポート』 (56 ページ) を参照してください。

Zend Core for i5/OS のデフォルトでは、XML-RPC サポートが使用可能に設定されていないため、図 4-11 に示すように、Zend 管理パネルを使用して XML-RPC 拡張モジュールを使用可能に設定する必要があります。

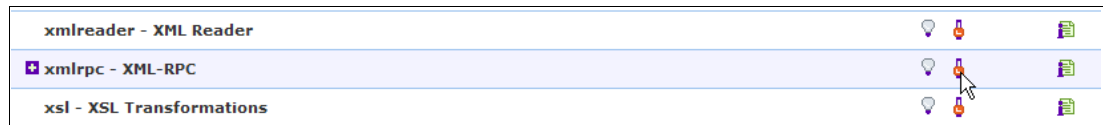


図 4-11 XML-RPC 拡張モジュールの使用可能化

例 4-13 のコードは、hellofunc という関数を使用する、簡単な XML-RPC サービス・プロバイダーを実装します。hellofunc はストリング hello world を返します。

例 4-13 XML-RPC サービス・プロバイダーの例

```
<HTML>
<?php
function hellofunc() {
    return "hello world";
}

$xmlrpc = xmlrpc_server_create();
xmlrpc_server_register_method($xmlrpc, 'hellofunc', 'hellofunc');
$r = xmlrpc_server_call_method($xmlrpc, $GLOBALS['HTTP_RAW_POST_DATA'], "");
echo $r;
xmlrpc_server_destroy($xmlrpc);
?>
</HTML>
```

PHP での XML-RPC 実装の詳細については、次の Web アドレスを参照してください。

<http://www.php.net/manual/en/ref.xmlrpc.php>

Zend Core に標準装備の XML-RPC サポートの代替として、純粋に PHP で実装された XML-RPC も存在します。XML-RPC for PHP は、次の Web アドレスからダウンロードできます。

<http://phpxmlrpc.sourceforge.net/>

XML-RPC for PHP は全面的に PHP で記述されており、このライブラリーを使用すると非常に簡単に XML-RPC サービスを実装できます。ただし、IBM および Zend は XML-RPC for PHP を公式にはサポートしていません。

## 4.7 SOAP サポート

XML-RPC の後継である SOAP を使用して PHP で Web サービスを実装できます。SOAP は、XML-RPC に比べ機能が豊富ですが、その代償として複雑になっています。SOAP の詳細については、次の Web ページ ( ケース・センシティブ ) を参照してください。

<http://en.wikipedia.org/wiki/SOAP>

PHP の SOAP 拡張モジュールは、SOAP 使用時の複雑さを軽減するために有用です。Zend Core for i5/OS では、デフォルトで SOAP 拡張モジュールが使用可能です。例 4-14 は、PHP で SOAP サービスを実装するのがどれ程簡単であるのかを示します。

例 4-14 SOAP サービス・プロバイダーの例

---

```
<?php
function Echolt($val) {
    return $val;
}

$soapsrv = new SoapServer(null, array('location'=>"http://xx.xx.xx.xx/soap.php",
'uri'=>"http://xx.xx.xx.xx/"));
$soapsrv->addFunction("Echolt");
$soapsrv->handle();
?>
```

---

例 4-14 は、サービスに渡された値を返す SOAP サービスを示します。PHP での SOAP 実装の詳細については、次の Web ページを参照することをお勧めします。

<http://www.php.net/manual/en/ref.soap.php>

例 4-15 は、既存の RPG プログラムの Web サービスを作成する方法を示します。

例 4-15 既存の RPG プログラムに対する Web サービスの作成

---

```
<?php
/*
このサービスは、2 個のパラメーターを指定して RPG プログラムを起動します。
*/
class i5_program_service {

    private $conn = false;

    function __construct() {

        $this->conn = i5_connect('127.0.0.1', 'user', 'password'/*, $connection_parameters*/);
        if (!is_resource($this->conn)) {
            throw new SoapFault('i5_program_service', 'Connection to i5 server failed, use i5_errormsg() to get the failure reason');
        }

    }

    public function service_for_i5_program($var_0, $var_1) {

        $description = Array (
            array ('Name' => 'code', 'IO' => I5_INOUT, 'Type' => I5_TYPE_CHAR, 'Length' => '10'),
            array ('Name' => 'name', 'IO' => I5_INOUT, 'Type' => I5_TYPE_CHAR, 'Length' => '10'));

        $prog = i5_program_prepare('eacdemo/teststp2', $description, $this->conn);
```

```

if (is_resource($prog)) {

    /* プログラムを実行します。 */
    $params = array (
        'code' => $var_0, 'name' => $var_1);

    $retvals = array(
        'code' => 'ret_val_1', 'name' => 'ret_val_2');

    $ret = i5_program_call($prog, $params, $retvals) ;

    if ($ret === true) {
        $ret = array($ret_val_1, $ret_val_2);
        return $ret;
    }

    else {
        throw new SoapFault('i5_program_service', 'Failed to call the program, use i5_errormsg() to
get the failure reason');
    }

    if (!i5_program_close ($prog) ) {
        throw new SoapFault('i5_program_service', 'Failed to free program resource handle, use i5_
errmsgsg() to get the failure reason');
    }
}

else {
    throw new SoapFault('i5_program_service', 'Program prepare failed, use i5_errormsg() to get th
e failure reason');
}
}

function __destruct() {

    if (!i5_close($this->conn)) {
        // i5 サーバーからの切断に失敗しました。i5_errormsg() を使用して、失敗した理由を取得して
ください。
    }
}

}

ini_set('soap.wsdl_cache_enabled', '0');
$server = new SoapServer('wsdl_service.wsdl');
$server->setClass('i5_program_service');
$server->handle();

?>

```

---







## データベース・アクセス

この章では、i5/OS に保管されているデータに PHP アプリケーションからアクセスする方法の詳細について説明します。DB2 for i5/OS に保管されているデータに焦点を当て、MySQL について簡単に説明します。

## 5.1 DB2

IBM DB2 は、オープン・スタンダードに従った高機能リレーショナル・データベース管理システムであり、大規模データと高速トランザクションの両方を管理できます。DB2 データ・サーバーは、そこで実行されるすべてのタイプとサイズのデータベースおよびアプリケーションに、高性能かつ堅固な環境を提供します。

DB2 データ・サーバーは、32 ビットおよび 64 ビット環境両方の、AIX、Sun、HP-UX、Linux、Windows、i5/OS、および z/OS® などの、すべてのプラットフォームで実行されるデータベース・ソリューションを提供します。DB2 ファミリーは、共用テクノロジーと共通アプリケーション・プログラミング・インターフェースを使用するリレーショナル・データベース管理システム (RDBMS) の一貫したセットです。

### 5.1.1 DB2 アクセス・オプション

DB2 for PHP 拡張機能は、コール・レベル・インターフェース (CLI) を使用して DB2 と通信します。このインターフェースは、C 言語で記述され DB2 ライブラリーを使用してコンパイルされた PECL (PHP Extension Community Library) の一部である、PHP 拡張機能です。

DB2 拡張機能を使用するためには、DB2 for i5/OS への接続時に、IP システム名や IP アドレスではなく、リレーショナル・データベース名を指定する必要があります。

RDB ディレクトリー項目を表示または変更するには、以下のコマンドを使用します。

DSRPRDBDIRE - RDB ディレクトリー項目を表示します

適切な権限があれば、次のコマンドを使用して名前を変更できます。

WRKRDBDIRE - リレーショナル・データベース・ディレクトリー項目を処理します

**ヒント :** DB2 for i5/OS へのアクセスに関して役立つ情報 :

a. db2\_connect では、実際にリモート項目である RDB ディレクトリー項目 (1 つの i5 が、別の i5 の IP アドレスまたはドメイン名を指す RDB ディレクトリー項目を持っている) を指定できるため、PHP は別の System i (PHP または Zend がインストールされていないシステムも含む) 上のデータにアクセスできます。このリモート・メソッドは DRDA® を使用します。

b. Zend がインストールされている System i に接続する場合は、システム名をブランクにしておきます。

DB2 を使用したアプリケーションを記述する場合に使用できる主な PHP の拡張機能は、3 つあります。

- ▶ IBM\_DB2
- ▶ PDO\_ODBC
- ▶ Unified ODBC

**重要 :** IBM\_DB2 および Unified ODBC 拡張機能は、Zend Core for i5/OS に既に備わっておりインストールされています。したがって、このサポートの使用を開始するために、コードのダウンロードや構成変更の実行は必要ありません。

IBM は、IBM DB2 データベースから最良の結果を取得するために IBM\_DB2 または PDO\_ODBC を使用することをお勧めします。

**制約事項** : PDO\_ODBC は、現在は DB2 for i5/OS ではサポートされていません。

## IBM\_DB2

ibm\_db2 拡張機能によって提供されるこれらの API に関する資料を以下の場所で入手できます。

<http://www.zend.com/manual/ref.ibm-db2.php>

Linux、UNIX®、および Windows の各プラットフォーム用の ibm\_db2 拡張機能に関する資料は、次のアドレスにあります。その情報は i5/OS を対象としていませんが、大部分は正確で適用可能です。

<http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp?topic=/com.ibm.db2.udb.apdv.php.doc/doc/t0023484.htm>

また、IBM Redbook「*Developing PHP Applications for IBM Data Servers* (SG24-7218)」で、ここでは繰り返し説明しませんが非常に有益な技術詳細など、ibm\_db2 拡張機能に関する優れた情報を参照してください。このブックは以下の場所にあります。

<http://www.redbooks.ibm.com/abstracts/sg247218.html?Open>

PHP の ibm\_db2 拡張機能は、PHP から IBM DB2、Cloudscape、および Apache Derby の各データベースに接続するためのインターフェースを提供します。この拡張機能は、アプリケーション開発者が SQL QUERY を発行し、ラージ・オブジェクトを処理し、ストアード・プロシージャを呼び出し、パーシステント・コネクションを使用し、準備済み SQL ステートメントを使用するメカニズムを提供します。また、これはバージョン 5 より前の PHP リリースでも機能します。PDO\_ODBC とは異なり、ibm\_db2 は従来のプロシージャ型プログラミングに基づいており、Unified ODBC の関数と比較した場合のパフォーマンスは優れています。ibm\_db2 は、システム・テーブルを照会することにより DB2 データベース・サーバーおよびクライアントに関する詳細を取得するための組み込み関数を提供しています。これにより、DB2 データベース管理システムに関する多くの情報が提供されます。

**ヒント** : C 言語を理解していて、ドライバが実際にどのように動作するかを参照する場合は、次の URL のソース・コードを参照してください。

[http://viewcvs.php.net/viewvc.cgi/pecl/ibm\\_db2/](http://viewcvs.php.net/viewvc.cgi/pecl/ibm_db2/)

## i5/OS の考慮事項

ユーザー ID またはパスワードを指定せずに DB2 for i5/OS に (db2\_connect を使用して) 接続すると、データベースは PHP スクリプトが実行しているプロセスと同じプロセス内でアクセスされます。通常は (ただし、必ずではない)、これは、例 5-1 に示したジョブのような、「ZEND」サブシステム内のいくつかのジョブの 1 つです。

例 5-1 PHP スクリプトが実行されるジョブ

活動ジョブの処理				RCHXXXXX	12/15/06 15:20:03
CPU %:	3.4	経過時間:	00:00:00	活動ジョブ数:	311
オプションを入力して、実行キーを押してください。 2= 変更 3= 保留 4= 終了 5= 処理 6= 解放 7= メッセージの表示 8= スプール・ファイルの処理 13= 切断 ...					
現行					
Opt	サブシステム / ジョブ	ユーザー	Type	CPU %	機能 状況
	ZENDCOREAP	NOBODY	BCI	.0	PGM-httpd TIMW
	ZENDCOREAP	NOBODY	BCI	.0	PGM-httpd TIMW
	ZENDCOREAP	NOBODY	BCI	.0	PGM-httpd TIMW

パラメーターまたはコマンド  
==>

F3= 終了 F5= 最新表示 F7= 検索 F10= 統計の再始動  
F11= 経過データの表示 F12= 取り消し F23= オプション続き F24= キーの続き

ユーザー ID とパスワードを渡して接続すると、「サーバー・モード」と呼ばれるモードが使用されます。この場合、データベースは QSQSRVR ジョブを介してアクセスされます。

i5/OS 上のユーザー・プロファイルおよび PHP の詳細については、7.1.4 『ユーザー・プロファイル』(92 ページ) を参照してください。

**重要:** db2\_connect() 関数でオプション DB2\_ATTR\_CASE を DB2\_CASE\_LOWER に指定して使用していない場合、SQL ステートメント内の列名は大文字でなければなりません。

## API のリスト

以下に、ibm\_db2 拡張機能によってサポートされる API のリストを示します。便宜上、それらは共通のグループに配置されています。

表 5-1 ibm\_db2 API

API	説明
<b>サーバー / 接続 API</b>	
db2_bind_param	PHP 変数を SQL ステートメントのパラメーターにバインドします。
db2_client_info	DB2 データベース・クライアントを記述するプロパティを持つオブジェクトを戻します。
db2_close	データベース接続を閉じます。
db2_connect	データベースへの接続を戻します。
db2_cursor_type	ステートメント・リソースによって使用されるカーソル・タイプを戻します。
db2_exec	SQL ステートメントを直接実行します。
db2_execute	準備済み SQL ステートメントを実行します。
db2_prepare	実行する SQL ステートメントを準備します。
db2_pconnect	データベースへのパーシステント・コネクションを戻します。
db2_server_info	DB2 データベース・サーバーを記述するプロパティを持つオブジェクトを戻します。
db2_statistics	テーブルの索引および統計をリストする結果セットを戻します。
<b>結果</b>	
db2_free_result	結果セットと関連付けられたリソースを解放します。
db2_next_result	ストアド・プロシージャから次の結果セットを要求します。
db2_result	結果セット内の行から単一の列を戻します。
<b>コミット / ロールバック</b>	

API	説明
db2_autocommit	データベース接続の AUTOCOMMIT 状態を戻すか、または設定します。
db2_commit	トランザクションをコミットします。
db2_rollback	トランザクションをロールバックします。
<b>フェッチ</b>	
db2_fetch_array	結果セット内の行を表す配列を戻します ( 列位置で索引付け )。
db2_fetch_assoc	結果セット内の行を表す配列を戻します ( 列名で索引付け )。
db2_fetch_both	結果セット内の行を表す配列を戻します ( 列名と列位置の両方で索引付け )。
db2_fetch_object	取り出された行内の列を表すプロパティを持つオブジェクトを戻します。
db2_fetch_row	結果ポインターを次の行または要求された行に設定します。
<b>フィールド情報</b>	
db2_field_display_size	列を表示するために必要な最大バイト数を戻します。
db2_field_name	結果セット内の列の名前を戻します。
db2_field_num	結果セット内の名前付き列の位置を戻します。
db2_field_precision	結果セット内の示された列の精度を戻します。
db2_field_scale	結果セット内の示された列の位取りを戻します。
db2_field_type	結果セット内の示された列のデータ・タイプを戻します。
db2_field_width	結果セット内の示された列の現行値の幅を戻します。
<b>キー情報</b>	
db2_foreign_keys	テーブルの外部キーをリストする結果セットを戻します。
db2_primary_keys	テーブルの主キーをリストする結果セットを戻します。
<b>ステートメント</b>	
db2_free_stmt	示されたステートメント・リソースと関連付けられたリソースを解放します。
<b>エラー</b>	
db2_conn_error	前回の接続試行によって戻された SQLSTATE を含むストリングを戻します。
db2_conn_errormsg	前回の接続エラー・メッセージおよび SQLCODE 値を戻します。
db2_stmt_error	SQL ステートメントによって戻された SQLSTATE を含むストリングを戻します。
db2_stmt_errormsg	前回の SQL ステートメント・エラー・メッセージを含むストリングを戻します。

API	説明
<b>列 / プロシージャ</b>	
db2_column_privileges	テーブルの列および関連特権をリストする結果セットを返します。
db2_columns	テーブルの列および関連メタデータをリストする結果セットを返します。
db2_procedure_columns	ストアド・プロシージャのパラメーターをリストする結果セットを返します。
db2_procedures	データベースに登録されているストアド・プロシージャをリストする結果セットを返します。
db2_special_columns	テーブルのユニーク行 ID 列をリストする結果セットを返します。
<b>テーブル情報</b>	
db2_num_fields	結果セット内に含まれるフィールド数を返します。
db2_num_rows	SQL ステートメントの影響を受けた行数を返します。
db2_table_privileges	データベース内のテーブルおよび関連特権をリストする結果セットを返します。
db2_tables	データベース内のテーブルおよび関連メタデータをリストする結果セットを返します。

### i5/OS 固有の機能

db2\_connect() 呼び出しで設定可能な多数の構成可能属性があります。完全なリストは、表 5-2 に含まれています。属性キーの「i5\_」接頭部から明らかなように、これらの属性は i5/OS に固有であることに注意してください。

これらの属性を渡す方法の例を、次のコード断片で示します。

```
$options = array("i5_lib"=>"mylibrary", "i5_commit"=>DB2_I5_TXN_READ_UNCOMMITTED);
$i5 = db2_connect($i5localhost, $i5user, $i5password, $options);
```

表 5-2 db2\_connect() 呼び出しの構成可能属性

キー	値
i5_lib	<p>修飾されていないファイル参照の解決に使用されるデフォルトのライブラリーを示す文字値。これは、システム命名モードを使用した接続の場合は無効です。</p> <p>注：この属性が渡されない場合、テーブル名を DB2 for i5/OS のスキーマ (ライブラリー) で完全修飾する必要があります。例えば、「myschema.mytable」となります。</p>

キー	値
i5_naming	<p>DB2_I5_NAMING_ON - DB2 UDB CLI System i 命名モードをオンにします。ファイルは、スラッシュ (/) 区切り文字を使用して修飾されます。修飾されていないファイルは、そのジョブのライブラリー・リストを使用して解決されます。</p> <p>DB2_I5_NAMING_OFF - DB2 UDB CLI デフォルト命名モード (SQL 命名) をオフにします。ファイルは、ピリオド (.) 区切り文字を使用して修飾されます。修飾されていないファイルは、デフォルト・ライブラリーまたは現行ユーザー ID のいずれかを使用して解決されます。</p>
i5_dbcs_alloc	<p>DB2_I5_DBCS_ALLOC_ON - DBCS 翻訳列サイズ増加に対する DB2 6X 割り振り方式をオンにします。</p> <p>DB2_I5_DBCS_ALLOC_OFF - DBCS 翻訳列サイズ増加に対する DB2 6X 割り振り方式をオフにします。</p>
i5_commit	<p>SQL_ATTR_COMMIT 属性は、SQLConnect() の前に設定する必要があります。接続が確立された後でこの値が変更され、その接続がリモート・データ・ソースに対するものであった場合、その接続ハンドルに対する SQLConnect() が次回正常終了するまでは、その変更内容は有効になりません。</p> <p>DB2_I5_TXN_NO_COMMIT - コミットメント制御は使用されません。</p> <p>DB2_I5_TXN_READ_UNCOMMITTED - ダーティー読み取り、反復不可読み取り、および幻影読み取りが可能です。</p> <p>DB2_I5_TXN_READ_COMMITTED - ダーティー読み取りは不可能です。反復不可読み取り、および幻影読み取りは可能です。</p> <p>DB2_I5_TXN_REPEATABLE_READ - ダーティー読み取りおよび反復不可読み取りは不可能です。幻影読み取りは可能です。</p> <p>DB2_I5_TXN_SERIALIZABLE - トランザクションは順序付け可能です。ダーティー読み取り、反復不可読み取り、および幻影読み取りは不可能です。</p>

キー	値
i5_date_fmt	<p>DB2_I5_FMT_ISO - 国際標準化機構 (ISO) 日付形式 yyyy-mm-dd が使用されます。これはデフォルトです。</p> <p>DB2_I5_FMT_USA - アメリカ合衆国日付形式 mm/dd/yyyy が使用されます。</p> <p>DB2_I5_FMT_EUR - 欧州日付形式 dd.mm.yyyy が使用されます。</p> <p>DB2_I5_FMT_JIS - 日本工業規格日付形式 yyyy-mm-dd が使用されます。</p> <p>DB2_I5_FMT_MDY - 日付形式 mm/dd/yyyy が使用されます。</p> <p>DB2_I5_FMT_DMY - 日付形式 dd/mm/yyyy が使用されます。</p> <p>DB2_I5_FMT_YMD - 日付形式 yy/mm/dd が使用されます。</p> <p>DB2_I5_FMT_JUL - ユリウス日付形式 yy/ddd が使用されます。</p> <p>DB2_I5_FMT_JOB - ジョブのデフォルトが使用されます。</p>
i5_date_sep	<p>DB2_I5_SEP_SLASH - 日付区切り記号としてスラッシュ (/) が使用されます。これはデフォルトです。</p> <p>DB2_I5_SEP_DASH - 日付区切り記号としてダッシュ (-) が使用されます。</p> <p>DB2_I5_SEP_PERIOD - 日付区切り記号としてピリオド (.) が使用されます。</p> <p>DB2_I5_SEP_COMMA - 日付区切り記号としてコンマ (,) が使用されます。</p> <p>DB2_I5_SEP_BLANK - 日付区切り記号として空白が使用されます。</p> <p>DB2_I5_SEP_JOB - ジョブのデフォルトが使用されます。</p>
i5_time_fmt	<p>DB2_I5_FMT_ISO - 国際標準化機構 (ISO) 時刻形式 hh.mm.ss が使用されます。これはデフォルトです。</p> <p>DB2_I5_FMT_USA - アメリカ合衆国時刻形式 hh:mmxx が使用されます。xx は AM または PM です。</p> <p>DB2_I5_FMT_EUR - 欧州時刻形式 hh.mm.ss が使用されます。</p> <p>DB2_I5_FMT_JIS - 日本工業規格時刻形式 hh:mm:ss が使用されます。</p> <p>DB2_I5_FMT_HMS - hh:mm:ss 形式が使用されます。</p>



キー	値
i5_time_sep	<p>DB2_I5_SEP_COLON - 時刻区切り記号としてコロン (:) が使用されます。これはデフォルトです。</p> <p>DB2_I5_SEP_PERIOD - 時刻区切り記号としてピリオド (.) が使用されます。</p> <p>DB2_I5_SEP_COMMA - 時刻区切り記号としてコンマ (,) が使用されます。</p> <p>DB2_I5_SEP_BLANK - 時刻区切り記号としてブランクが使用されます。</p> <p>DB2_I5_SEP_JOB - ジョブのデフォルトが使用されます。</p>
i5_decimal_sep	<p>DB2_I5_SEP_PERIOD - 小数部区切り記号としてピリオド (.) が使用されます。これはデフォルトです。</p> <p>DB2_I5_SEP_COMMA - 小数部区切り記号としてコンマ (,) が使用されます。</p> <p>DB2_I5_SEP_JOB - ジョブのデフォルトが使用されます。</p>
i5_query_optimize	<p>DB2_FIRST_IO - すべての QUERY は、出力の最初のページをできるだけ素早く戻すことを目標にして最適化されます。この目標は、出力データの最初のページを参照した後で QUERY を取り消すと予想されるユーザーによって出力が制御されている場合に有効です。OPTIMIZE FOR nnn ROWS 節を使用してコーディングされた QUERY は、その節によって指定された目標に従います。</p> <p>DB2_ALL_IO - すべての QUERY は、QUERY 全体の実行が最短の経過時間で完了するように最適化されます。これは、QUERY の出力がファイルまたはレポートに書き込まれている場合、またはインターフェースが出力データをキューイングしている場合に、良いオプションになります。OPTIMIZE FOR nnn ROWS 節を使用してコーディングされた QUERY は、その節によって指定された目標に従います。これはデフォルトです。</p>
i5_fetch_only	<p>DB2_I5_FETCH_ON - カーソルは読み取り専用であり、位置指定の更新または削除には使用できません。 SQL_ATTR_FOR_FETCH_ONLY 環境が SQL_FALSE に設定されていない限り、これはデフォルトです。</p> <p>DB2_I5_FETCH_OFF - カーソルは、位置指定の更新および削除に使用できます。</p>

### コード例

以下に、いくつかの簡単なコード例を示します。これらは断片であり、完全なプログラムではありません。

例 5-2 は、プリペアド・ステートメントを作成し、パラメーターをバインドし、それを実行した後で、その結果セット内を反復する方法を示しています。

#### 例 5-2 プリペアド・ステートメントの使用

---

```
...
$options = array("i5_lib"=>"QIWS");
$i5 = db2_connect($db, $user, $password, $options);

if (!$i5) :
die ("Problems connecting to the database.");
endif;

/* Construct the SQL statement, using CDTLMT as the search substring */
$sql = "select * from QCUSTCDT where CDTLMT > ?";

/* Prepare, bind and execute the DB2 SQL statement */
$stmt= db2_prepare($i5, $sql);
$lower_limit = 1000;
db2_bind_param($stmt, 1, "lower_limit", DB2_PARAM_IN);
db2_execute($stmt);
...
/* Execute the statement again with a new input parameter */
$upper_limit = 99999;
db2_bind_param($stmt, 1, "upper_limit", DB2_PARAM_IN);
db2_execute($stmt);
```

---

例 5-3 (68 ページ) は、db2\_procedures API を使用して、i5/OS で使用可能なストアド・プロシージャのリストを取得する方法を示しています。

#### 例 5-3 ストアド・プロシージャのリストの取得

---

```
...
$procName = '%';
$schemaName = '%';

$procs = db2_procedures($conn, "", $schemaName, $procName);
if (!$procs){
    print ('Error in db2_procedures() call');
} else {
    print '<h1>Procedures found in schema ' . $schemaName . ' on System i</h1>';
    print '<h2>Using db2_procedures and db2_fetch_array</h2>';
    print '<br><table border=1 cellpadding=5 cellspacing=5>';
    print '<tr><td
align=center>procedure_cat<td>procedure_schem<td>procedure_name<td>num_input_params<td>nu
m_output_params<td>num_result_sets<td>remarks</td></tr>';
    while ($row = db2_fetch_array($procs)) {
        if (!$row=="") {
            $procedure_cat = $row[0];
            $procedure_schem = $row[1];
            $procedure_name = $row[2];
            $num_input_params = $row[3];
            $num_output_params = $row[4];
            $num_result_sets = $row[5];
            $remarks = $row[6];
            print '<tr><td align=center>' . $procedure_cat . '<td>' . $procedure_schem .
            '<td>' . $procedure_name . '<td>' . $num_input_params . '<td>' . $num_output_params . '<td>' .
            $num_result_sets . '<td>' . $remarks . '</td></tr>';
        }
    }
    print '</table><br>';
}
...
```

---

## PDO\_ODBC

PDO (PHP Data Objects) は、オブジェクト指向の標準に基づいた PHP でのデータ・アクセス方式で、同じ方式を使用して、データベースを照会し、サポートされているデータベースからデータを取り出すことができます。PDO\_ODBC 拡張機能は、PDO 仕様のインプリメンテーションです。DB2 ライブラリーを使用してコンパイルすると、それを使用して、DB2、Cloudscape、および Apache Derby の各データベースにアクセスできます。この拡張機能は、ローカルにカタログされたデータベースとカタログされていないデータベースの両方に接続するメカニズムを提供します。ローカルにカタログされたデータベースの場合、PDO\_ODBC は、データベース・サーバーの詳細をクライアント・マシンから取得します。ローカルにカタログされていないデータベースの場合、リモート・データベースの完全な詳細は接続 URL に指定されます。PDO\_ODBC は、パーシステント・コネクション、準備済み SQL ステートメント、ラージ・オブジェクト、ストアード・プロシージャなどの、DB2 の拡張機能へのアクセスも提供します。これは、Unified ODBC の関数と比較した場合、優れたパフォーマンスを提供します。

前述したとおり、PDO\_ODBC は、現在は DB2 for i5/OS ではサポートされていません。

## Unified ODBC

Unified ODBC は、ibm\_db2 拡張機能がリリースされるまでは、PHP が DB2 と対話する唯一の方法でした。ibm\_db2 拡張機能と同様に、Unified ODBC はネイティブ CLI 呼び出しを使用して DB2 と対話します。これは、基礎メカニズムが異なる場合でも、同じ PHP メソッドを使用してさまざまなデータベースと対話します。したがって、ODBC が提供するデータベース独立性は、特定の DBMS 用に設計された拡張機能よりも優れています。ただし、この API は DB2 のストアード・プロシージャ呼び出しには使用できません。さらに、ibm\_db2 拡張機能は DB2 アクセス用に最適化されているため、ibm\_db2 拡張機能を使用することでより良いパフォーマンスが実現されます。

例 5-4 に、Unified ODBC の使用例を示します。

例 5-4 Unified ODBC と DB2 for i5/OS

---

```
<HTML>
<?php
# connect to rdb directory entry called 'mysystem' ...
# ...using 'kost' as username and 'newpass' as password
$dbh=odbc_connect('mysystem','kost','newpass');

# check if connected
if (!$dbh) {
    die ('Connection Failed:' . $dbh);
}

# execute the query
$sql="SELECT Host,User FROM schema.user";
$res=odbc_exec($dbh,$sql);
if (!$res) {
    die ("Error in SQL:$sql");
}

# output the results
while (odbc_fetch_row($res))
{
    print odbc_result($res,"Host").'.';
    print odbc_result($res,"User").'<br />';
}

# close the connection
odbc_close($dbh);
```

?>  
</HTML>

---

Unified ODBC 拡張機能およびその機能の詳細については、次のアドレスを参照してください。

<http://www.php.net/manual/en/ref.uodbc.php>

## 5.1.2 パフォーマンスに関する考慮事項

一般に、可能な場合は常に `ibm_db2` ドライバーを使用して DB2 のデータにアクセスすることをお勧めします。これは最良のパフォーマンスを提供しますが、PDO\_ODBC や Unified ODBC ほどデータベースに対して中立的ではありません。

パフォーマンスに関する推奨事項の多くは、DB2 for i5/OS にアクセスするその他のプログラミング言語の場合と類似しています。例えば、Java で JDBC™ サポートを使用する際に開発者が注意すると考えられる考慮事項の大部分は、PHP の場合にも該当します。

### 1 回準備し、多数回実行する

完全なデータベース・パスのオープン回避のための最も効率的な方法は、「1 回準備し、多数回実行する」と呼ばれるプログラミング・パラダイムを採用することです。理想的には、複数回実行される SQL ステートメントは 1 回だけ準備し、それに続く一連の実行で再利用します。

プリペアド・ステートメントの使用例については、例 5-2 (68 ページ) を参照してください。

### NOBODY ユーザー・プロファイルとして接続する

特定のユーザーではなく NOBODY として DB2 for i5/OS に接続することによって、より良いパフォーマンスが実現されます。これにより、データベース・アクセスは PHP スクリプトが実行しているジョブと同じ i5/OS ジョブ内で発生します。これは、`db2_connect` 呼び出しのユーザー ID とパスワードに空ストリング (&dlq;&dlq;) を渡すことによって行うことができます。

特定のユーザー・プロファイルで実行すると、データベース・アクセスは QSQSRVR ジョブを介して発生し、ある程度のオーバーヘッドが追加されます。

詳しくは、『i5/OS の考慮事項』(61 ページ) を参照してください。

### トランザクション分離レベル

分離レベルは、データのアクセス中にどのようにデータがロックされその他のプロセスから分離されるかを決定するため、並行性とデータ保全性を最適なバランスに保つ分離レベルを選択する必要があります。ユーザーが指定する分離レベルは、その作業単位の期間有効です。

DB2\_I5\_TXN\_SERIALIZABLE は制限と保護が最大のトランザクション分離レベルで、かなりのオーバーヘッドが生じます。多くのワークロードはこのレベルの分離保護を必要としません。所定のアプリケーションは、基礎データの更新、またはその他の並行する更新元との同時実行を行わない可能性があります。その場合は、アプリケーションは、ダーティー読み取り、反復不可読み取り、または幻影読み取りの問題を気にする必要はありません。(詳しくは、DB2 製品資料を参照してください。) そのような場合は、DB2\_I5\_TXN\_READ\_UNCOMMITTED で十分であると考えられます。分離レベル要件を確認し、パフォーマンスを向上させるように適切に調整してください。

## パーシステント・コネクション

パーシステント・コネクションにより全体のパフォーマンスが飛躍的に向上する可能性があります。注意して使用する必要があります。

パーシステント・コネクションを使用する主な利点は、通常はデータベースへの接続と関連付けられている初期化と分解の大部分が回避されるということです。パーシステント・コネクションに対して `db2_close()` が呼び出されると、その呼び出しは常に TRUE を返しますが、基礎 DB2 クライアント接続は開かれたまま次の一致する `db2_pconnect()` 要求にサービスを提供するのを待っています。

パーシステント・コネクションの 1 つの主な関心領域は、コミットメント制御の領域です。パーシステント・コネクションは、自動コミットメント制御（「自動コミット」）がオンになっている場合にのみ使用するべきです。管理されたコミットメント制御とパーシステント・コネクションを混用すると、エラーが発生した場合に不明なトランザクション状態になる可能性があります。次のシナリオを考慮してみます。

1. 要求 A がパーシステント・コネクションを取得し、自動コミットをオフにし、さまざまなデータベース更新（挿入 / 更新 / 削除）操作を実行します。
2. 要求 A でエラーが発生し、トランザクションのコミットもロールバックも行いません。`db2_close()` 操作が呼び出されます。希望する更新がすべて実行されたわけではないことに注意してください。
3. 要求 B がパーシステント・コネクションを要求し、要求 A によって使用された接続と同じ接続が与えられます。いくつかのデータベース更新を実行し、その変更内容をコミットします。
4. 要求 B が変更内容をコミットしたときに、要求 A によって実行された不完全なデータベース変更内容もコミットされました。したがって、データベースには不整合かつ不完全なトランザクションがあります。

### 5.1.3 トラブルシューティング

ibm\_db2 ドライバーに組み込まれているトレース機能はありません。問題発生時に参照すべき箇所は、以下のとおりです。

- ▶ i5/OS ジョブ・ログ、メッセージ・キューなど
- ▶ PHP ログ・ファイル (`php_error_log` が最良の参照箇所と考えられる)
- ▶ アプリケーションによって生成されたメッセージ（ブラウザに送信されたか、または別のログ・ファイルに書き込まれたもの）

特に PHP には関係しない DB2 for i5/OS のデバッグおよびトラブルシューティングの豊富な情報源が、IBM Redbook「*SQL Performance Diagnosis on IBM DB2 Universal Database for iSeries* (SG24-6654)」にあります。これは以下のサイトにあります。

<http://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/sg246654.html?Open>

## 5.2 MySQL

MySQL は、共通オープン・ソース SQL データベース管理システムです。これは、MySQL AB (<http://www.mysql.com>) によって開発され、配布されています。この会社は、MySQL 関連のサービスを提供することによってビジネスを行っています。

MySQL は、最初は UNIX および Linux のアプリケーション用に開発されました。MySQL は、無償でインターネット顧客に提供可能で動的 Web アプリケーションが必要とするすべてのストレージ機能および検索機能を提供できることがインターネット・サービス・プロバイダー (ISP) によって知られるようになると、一般的になりました。ISP の大部分がお気に

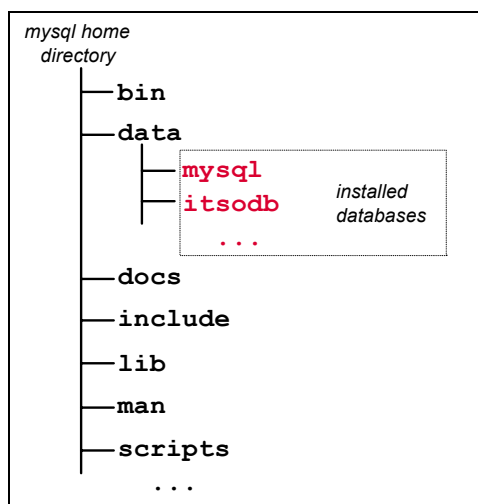
入りの Web サーバー環境として Linux または UNIX を Apache とともに使用していることも有利でした。ただし、MySQL は、ほとんどすべてのオペレーティング・システム上の多くのアプリケーションに統合されたデータベースとして使用されています。

## 5.2.1 MySQL のアーキテクチャー

MySQL は、TCP/IP に基づくクライアント / サーバー・アーキテクチャーです。MySQL サーバーは指定されたポートで接続を待ち、クライアントからの SQL ステートメントに応答します。

## 5.2.2 MySQL のディレクトリー構造

MySQL は、1 つのサーバー内の 1 つ以上のデータベースとしてインストールできます。それぞれのデータベースに対して、MySQL は、データ、索引、およびその他の関連情報を保持するディレクトリーを作成します。



左側の図は、ディレクトリー構造を示しています。この例では、インストールされたデータベースは *mysql* (デフォルトでセキュリティ情報を保持) とサンプル・データベース *itsodb* です。それぞれのデータベースに対して、表あたり 3 つのファイルを含むディレクトリーがあります。「MYD」拡張子を持つファイルは表データを含んでいます。「MYI」拡張子を持つファイルは表の索引を含んでいます。frm 拡張子を持つファイルは、スキーマと呼ばれる表の構造定義を含んでいます。

## 5.2.3 インストールおよび構成

以下の説明は、MySQL を i5/OS PASE 環境にダウンロード、インストール、および実行する方法を理解するために役立ちます。

**制約事項：**本書の作成時には、i5/OS では MySQL は正式にはサポートされていませんでした。

1. i5/OS で次のコマンドを発行して、ユーザー・プロファイル *mysql* を作成します。  
CRTUSRPRF USRPRF(MYSQL) STATUS(\*DISABLED) LCLPWD MGT(\*NO) TEXT('MySQL user id')  
MySQL プロファイルは使用不可の ID であるためシステムにサインオンするために使用することはできませんが、MySQL による使用は可能であることに注意してください。
2. <http://www.mysql.com> にアクセスし、MySQL 5.0 の AIX 5.2 (POWER™ 64 ビット) バージョンをワークステーションにダウンロードします。本書の作成時には、*mysql-max-5.0.24a-aix5.2-powerpc-64bit.tar.gz* は MySQL の Max バージョンのファイル名、*mysql-standard-5.0.24a-aix5.2-powerpc-64bit.tar.gz* は MySQL の Standard バージョンのファイル名でした。ここでは、Max バージョンを使用しました。

3. 図 5-1 に示したように、TAR アーカイブを取得するためにファイルをワークステーションに解凍します。ここでは、「7-zip」と呼ばれる GPL ライセンスのツール（無償でダウンロードおよび使用可能）を使用しました。これは次の Web アドレスでダウンロードできます。

<http://www.7-zip.org>

注：WinRAR は .tar アーカイブも自動的に開くため、.gz アーカイブの解凍には使用できません。

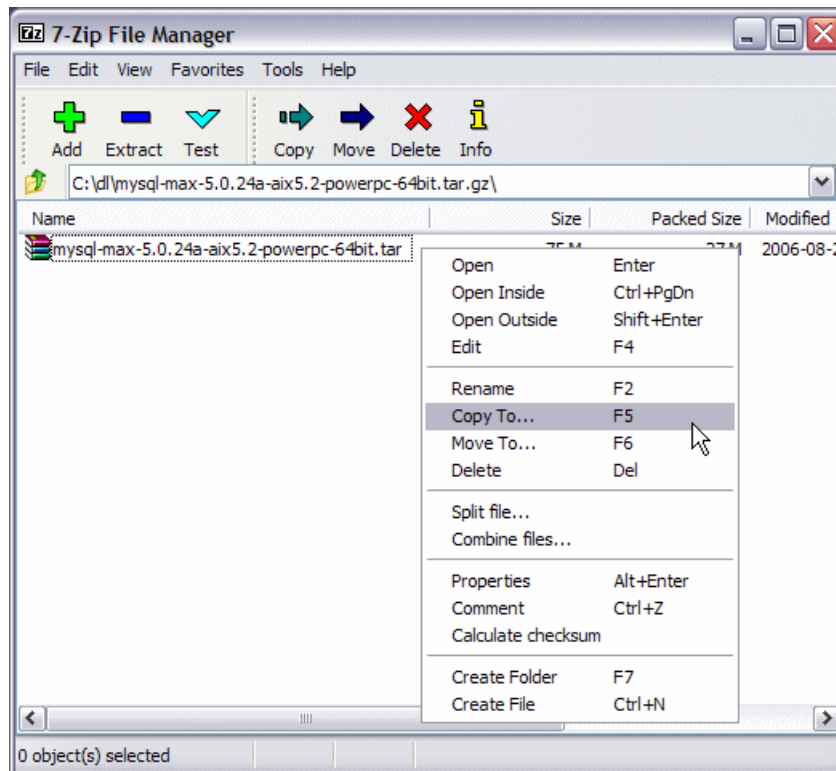


図 5-1 7-Zip を使用した TAR アーカイブの解凍

4. .tar ファイルを取得したら、i5/OS 上の /usr/local ディレクトリーに転送します（通常は FTP を使用）。
5. 5250 セッションにユーザー QSECOFR としてログオンします。
6. 次のコマンドを発行して、PASE for i5/OS 環境に移動します。
- ```
call qp2term
```
7. 次のコマンドを発行して、ディレクトリー /usr/local に移動します。
- ```
cd /usr/local
```
8. 図 5-2 に示したように、コマンド **tar xvf mysql-max-5.0.24a-aix5.2-powerpc-64bit.tar** を使用してファイルを untar します。

```

/QOpenSys/usr/bin/-sh

$
> cd /usr/local
$
> ls
Zend
mysql-max-5.0.24a-aix5.2-powerpc-64bit.tar
$
> tar xvf mysql-max-5.0.24a-aix5.2-powerpc-64bit.tar
x mysql-max-5.0.24a-aix5.2-powerpc-64bit
x mysql-max-5.0.24a-aix5.2-powerpc-64bit/bin
x mysql-max-5.0.24a-aix5.2-powerpc-64bit/bin/comp_err, 1379704 bytes, 2695 me
dia blocks.
x mysql-max-5.0.24a-aix5.2-powerpc-64bit/bin/replace, 1377369 bytes, 2691 med
ia blocks.

===>

F3= 終了   F6= 印刷   F9= コマンドの複写  F11= 切り捨て / 折り返し
F13= 消去  F17= 先頭   F18= 最後    F21=CL コマンド入力

```

図 5-2 MySQL TAR ファイルの `untar`

- ディレクトリーのナビゲーションが簡単になるように、次のコマンドを使用して新規ディレクトリー（ここでは、&ldquo;mysql-max-5.0.24a-aix5.2-powerpc-64bit"）への `mysql` という名前のシンボリック・リンクを作成します。

```
ln -s mysql-max-5.0.24a-aix5.2-powerpc-64bit mysql
```

- 次のコマンドを発行して、MySQL システム・テーブルを作成します。

```
scripts/mysql_install_db --user=mysql
```

- これで、次のコマンドを使用して MySQL サーバーを開始できます（図 5-3（74 ページ）に示したとおり）。

```
bin/mysqld_safe --user=mysql &
```

```

> bin/mysqld_safe --user=mysql &
[1] 114232
$ Starting mysqld daemon with databases from /usr/local/mysql/data

===>

F3= 終了   F6= 印刷   F9= コマンドの複写  F11= 切り捨て / 折り返し
F13= 消去  F17= 先頭   F18= 最後    F21=CL コマンド入力

```

図 5-3 MySQL の開始

- インストールが正常終了したことを検証するには、次のコマンドを発行します。

```
bin/mysqlcheck -u root mysql
```

図 5-4 に示したような出力が得られるはずですが。



```
> bin/mysqlcheck -u root mysql
mysql.columns_priv          OK
mysql.db                    OK
mysql.func                  OK
mysql.help_category         OK
mysql.help_keyword          OK
mysql.help_relation         OK
mysql.help_topic            OK
mysql.host                  OK
mysql.proc                  OK
mysql.procs_priv            OK
```

===>

F3= 終了 F6= 印刷 F9= コマンドの複写 F11= 切り捨て / 折り返し  
F13= 消去 F17= 先頭 F18= 最後 F21=CL コマンド入力

図 5-4 mysqlcheck の出力

図 5-5 に示したようなエラー・メッセージを取得した場合は、MySQL サーバーが実行されていないと考えられます。

```
> bin/mysqlcheck -u root mysql
bin/mysqlcheck:Got error:2002: Can't connect to local MySQL server through
socket '/tmp/mysql.sock' (2) when trying to connect
$
```

===>

F3= 終了 F6= 印刷 F9= コマンドの複写 F11= 切り捨て / 折り返し  
F13= 消去 F17= 先頭 F18= 最後 F21=CL コマンド入力

図 5-5 mysqlcheck 失敗の出力

13. MySQL を PHP in Zend Core とともに使用するためには、Zend Core コントロール・センターで MySQL Extensions を使用可能にする必要があります。これは、  
/usr/local/Zend/Core/etc/php.ini ファイルの以下の各行をアンコメントする (「#」文字を除去する) ことによって行うこともできます。

```
extension=mysql.so
extension=mysqli.so
```

14. MySQL サーバーを停止するには、次のコマンドを実行します (/usr/local/mysql ディレクトリーから)。

```
bin/mysqladmin -u root shutdown
```

正常にシャットダウンされると、メッセージは表示されません。

15. セキュリティーを高めるために、root MySQL ユーザーのパスワードを設定することをお勧めします。次のコマンドを入力することによって、これを行うことができます  
(/usr/local/mysql ディレクトリーから):

```
bin/mysqladmin -u root password newpass
```

このコマンドは、ユーザー「root」のパスワードを「newpass」に変更します。

**注 :** root MySQL ユーザーのパスワードを設定した後は、データベース作業中 ( ユーザー root として ) にはパスワードを指定する必要があります。例えば、サーバーをシャットダウンするには、次のコマンドを指定します。

```
bin/mysqladmin -u root -pnewpass shutdown
```

16. 次のアドレスにある「MySQL 5.0 Reference Manual」の『UNIX Post-Installation Procedures』を参照することをお勧めします。

<http://dev.mysql.com/doc/refman/5.0/en/unix-post-installation.html>

## 5.2.4 アクセス・オプション

MySQL は、以下の PHP の拡張機能とともに使用できます。

- ▶ MySQL/MySQLi
- ▶ PDO\_MYSQL
- ▶ Unified ODBC

すべての拡張機能のソース・コードは、次の PECL Web サイトからダウンロードできます。

<http://pecl.php.net/>

最良の結果とパフォーマンスを得るために、MySQL インターフェースを使用することをお勧めします。

**注 :** Zend for i5/OS にはこれらの拡張機能が初めからすべて備わっています。したがって、それらをダウンロードする必要はありません。使用するには、Zend Administration Panel で希望する拡張機能を使用可能にしてから、Web サーバーを再始動してください。

### MySQL/MySQLi

PHP の mysql および mysqli 拡張機能は、PHP から MySQL データベースに接続するインターフェースを提供します。これらの拡張機能は、アプリケーション開発者が SQL QUERY を発行し、ラージ・オブジェクトを処理し、パーシステント・コネクションを使用するメカニズムを提供します。これらの拡張機能は、バージョン 5 以前の PHP でも機能します。PDO\_ODBC とは異なり、mysql および mysqli は従来のプロシージャ型プログラミングに基づいており、Unified ODBC と比較した場合のパフォーマンスは優れています。mysql と mysqli の間の主な相違点は、mysqli は mysql のさらに高機能な ( 「改良された」 ) 拡張機能であり、MySQL 4.1 以降で提供されている MySQL の機能をサポートしているということです。

mysql 拡張機能およびその機能の詳細については、次の Web サイトを参照してください。

<http://www.php.net/manual/en/ref.mysql.php>

例 5-5 に、mysql 拡張機能の使用例を示します。

#### 例 5-5 mysql 拡張機能の例

```
<HTML>
<?php
# connect to MySQL DBMS using 'root' as username and 'kost' as password
$dbh = mysql_connect('localhost', 'root', 'kost');

# check if connected
if (!$dbh) {
    die('mysql connect failed:'.mysql_error());
}
```

```

# select database and do the query
mysql_select_db("mysql");
$sql='select Host,User from user';
$res = mysql_query($sql);
if (!$res) {
    die('Error:$sql:'.mysql_error());
}

# output the query results
while ($r = mysql_fetch_assoc($res)) {
    print $r['Host'].':';
    print $r['User'].'<br />';
}

# free the query resources
mysql_free_result($res);

# close connection to MYSQL DBMS
mysql_close($dbh);
?>

</HTML>

```

---

例 5-6 に、mysqli の使用例を示します。mysqli 拡張機能およびその機能の詳細については、次のアドレスを参照してください。

<http://www.php.net/manual/en/ref.mysqli.php>

#### 例 5-6 mysqli 拡張機能のプロシージャール構文

---

```

<HTML>
<?php
# connect to MySQL system database called 'mysql' ...
# ...using 'root' as username and 'kost' as password
$dbh = mysqli_connect('localhost', 'root', 'kost', 'mysql');

# check if connected
if (!$dbh) {
    die('mysqli connect failed:'.mysqli_connect_error());
}

# execute the query
$sql = 'select Host,User from user';
$res = mysqli_query($dbh,$sql);
if (!$res) {
    die("Error:$sql:".mysqli_error($dbh));
}

# output the query results
while ($r = mysqli_fetch_assoc($res)) {
    print $r['Host'].':';
    print $r['User'].'<br />';
}

# free the query resources
mysqli_free_result($res);

# close connection to MYSQL DBMS
mysqli_close($dbh);

```

```
?>
</HTML>
```

---

## PDO\_MYSQL

PDO (PHP Data Objects) は、オブジェクト指向の標準に基づいた PHP でのデータ・アクセス方式で、同じ方式を使用して、データベースを照会し、サポートされているデータベースからデータを取り出すことができます。PDO\_ODBC 拡張機能は、PDO 仕様のインプリメンテーションです。これは、Unified ODBC の関数と比較した場合、優れたパフォーマンスを提供します。

## Unified ODBC

Unified ODBC は、基礎メカニズムが異なっても同じ PHP メソッドを使用してさまざまなデータベースと対話します。したがって、これはユーザーのコードにデータベース独立性のレベルを提供しますが、代償として低い機能性とパフォーマンスの低下を伴います。

次の PECL Web サイトから、すべての拡張機能の無償ソース・コードをダウンロードできます。

<http://pecl.php.net/>

## 5.3 MySQL から DB2 へのマイグレーション

MySQL は非常に堅固なフル機能のデータベースへと発展してきましたが、ほとんどすべてのカテゴリにおいて DB2 for i5/OS からは大きく遅れています。

MySQL から DB2 へのマイグレーション方法に関する優れた情報については、「*Developing PHP Applications for IBM Data Servers* (SG24-7218)」を参照してください。その資料は i5/OS に固有のものではありませんが、非常に役立つガイドです。

例えば、IBM DB2 Migration Toolkit (MTK) は、

<http://www.ibm.com/software/data/db2/migration/mtk/> から入手可能なフリー・ソフトウェアです。現在、DB2 MTK は MySQL データベースから DB2 SQL 構文への変換はサポートしていますが、DB2 for i5/OS ターゲットはサポートしていません。DB2 MTK の次のバージョンは、MySQL から DB2 for i5/OS SQL 構文への変換をサポートします。『オンライン・リソース』(121 ページ) で IBM DB2 Migration Toolkit Web ページへのリンクを参照し、最新バージョン情報を調べてください。DB2 Migration Toolkit は、長い SQL スクリプトを変換する場合に最も適しています。このツールは、PHP コードに組み込まれた SQL ステートメントを直接変換するようには設計されていません。ただし、個々の SQL ステートメントを DB2 MTK Translator ユーティリティにコピーして、複雑な MySQL SQL ステートメントの変換を簡単に行うことができます。

別の役立つリソースとしては、次のアドレスの「DB2 for i5/OS - Porting Information」Web サイトがあります。

<http://www.ibm.com/servers/enable/site/db2/porting.html>

DB2 for i5/OS と MySQL の間の相違点を強調する資料が、以下にあります。

<http://www.ibm.com/developerworks/systems/library/es-path2php/>

## 5.4 MySQL または DB2 をいつ選択すべきか

DB2 for i5/OS は i5/OS オペレーティング・システムに無償で組み込まれているため、i5/OS で MySQL をわざわざ使用する理由があるとすれば、アプリケーションを別のシステムから System i に移植していて、そのアプリケーションが MySQL と密接に関連している場合などしかありません。この場合にも、DB2 を利用するようにアプリケーションを変更することを考慮できます。





## Java Bridge サポート

本章では、Zend Platform for i5/OS での Java Bridge サポートの詳細について説明します。まず Java Bridge の機能について説明します。その後で、Java Bridge を使用して、IBM Toolbox for Java 経由で i5/OS のリソースにアクセスする例を複数示します。

## 6.1 概説

Java Bridge サポートにより、PHP から Java プログラムを呼び出すことが可能になります。この結果、IBM Toolbox for Java をインターフェースとして、開発済みの既存の Java アプリケーションに直接アクセスしたり、データ・キュー、プリンターなどの i5/OS リソースそのものに直接アクセスしたりできます。この手法をとることにより、既存の機能の公開または使用に Web サービスなどのテクノロジーを使用した場合に必要な開発時間やコード関連の手間を削減できます。

Java Bridge サポートの詳細については、次の Web サイトを参照してください。  
[http://www.zend.com/products/zend\\_platform/in\\_depth/php\\_java\\_integration](http://www.zend.com/products/zend_platform/in_depth/php_java_integration)

IBM Toolbox for Java の詳細については、次の Web サイトを参照してください。  
<http://www.ibm.com/eserver/iserics/toolbox/>

## 6.2 Java Bridge の構成

Java Bridge 機能を使用するには、若干の追加の構成が必要です。

**重要：**Java Bridge 機能は、Zend Platform for i5/OS 製品がインストールされている場合にのみ使用可能です。Zend Core for i5/OS には含まれません。

### 6.2.1 主構成ファイル：javamw.rc

Java Bridge サポートの主構成ファイルは次のファイルです。

/usr/local/Zend/Platform/bin/javamw.rc

**制約事項：**メモ帳など PC ベースのエディターでこのファイルを編集しないでください。ファイルが破損します。i5/OS エディター (EDTF) を使用することを推奨します。

このファイルに変更を加える場合は、主に Java クラスパスのエントリーを追加することが必要になります。例えば、IBM Toolbox for Java に関する追加内容は次のとおりです。

```
...  
CLASSPATH=$BINDIR/javamw.jar:/j400/jt400.jar:$CLASSPATH  
...
```

### 6.2.2 Java Bridge の状況

Zend Core for i5/OS と Zend Platform for i5/OS のインストールが正常に終了したら、次の URL に Web ブラウザーでアクセスすることにより、Java Bridge サポートがアクティブかどうかを検証できます。

<http://MYSERVER:89/Zend Platform/server/index.php?frame=java>

図 6-1 (83 ページ) は、表示されるページの一例を示します。



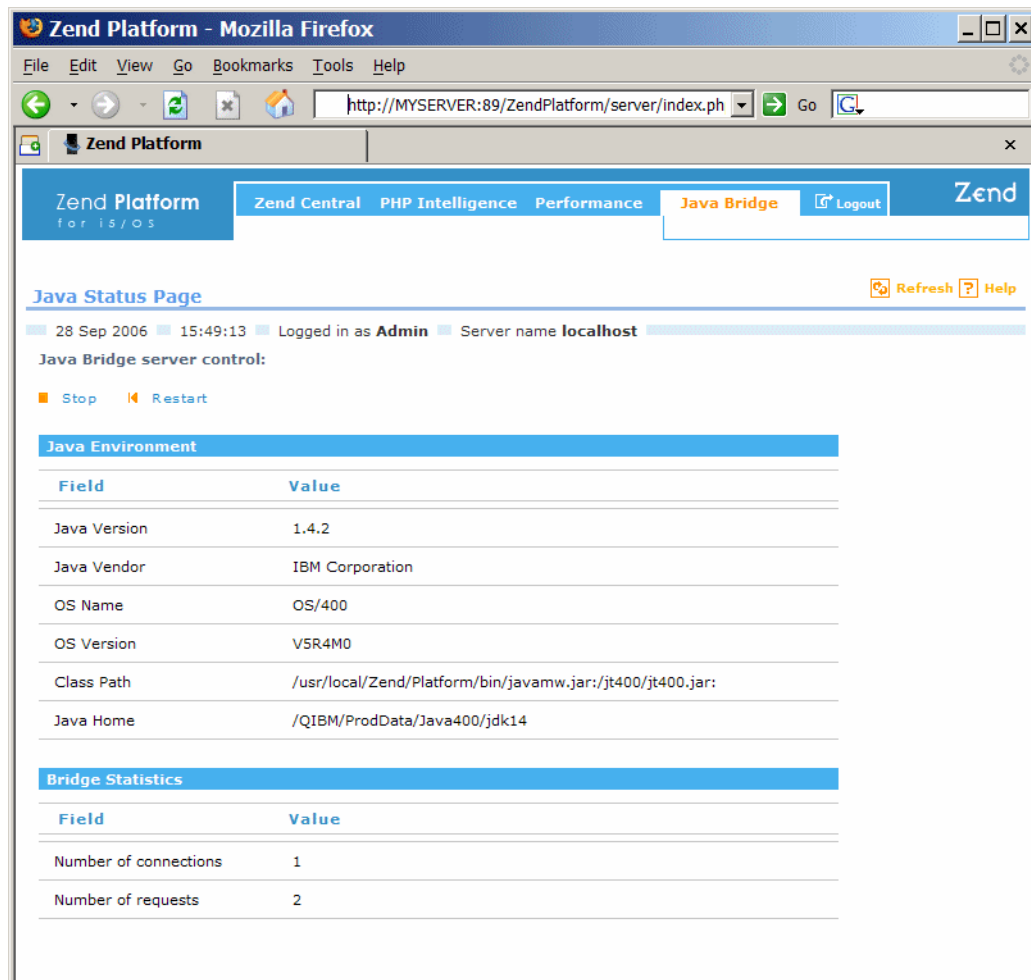


図 6-1 Java Bridge 状況ページ

変更したクラスパスが実際に機能していることをこの画面から確認できます。

この画面で、Java Bridge サポートの停止および再開を行うことができます。以下のコマンドを使用して i5/OS PASE ターミナルから以下の操作を実行することもできます (先に、`cd /usr/local/Zend/Platform` を実行して、適切なディレクトリーに移動してください)。

開始	<code>bin/javamw.rc start</code>
停止	<code>bin/javamw.rc stop</code>
再開	<code>bin/javamw.rc restart</code>
状況	<code>bin/javamw.rc status</code>

## 6.3 サンプル・プログラム

このセクションでは、簡単なプログラムを紹介して、Zend Platform for i5/OS の Java Bridge 機能をテストする方法および利用する方法を示します。

### 6.3.1 シンプル・テスト・プログラム

例 6-1 (84 ページ) のプログラムは、全 Java システム・プロパティーを取得し、これを終わりまで反復処理して、キーと値のペアをすべて画面にプリントする方法を示します。

#### 例 6-1 全 Java システム・プロパティのプリント

```
<html>
<head><title>Java System Properties</title></head>
<?php

// java.lang.System インスタンスを取得します。
$system = new Java("java.lang.System");

// プロパティを取得します。
// PHP により、プロパティは、キーと値の配列に格納されます。
$properties = $system->getProperties();

// プロパティを終わりまで反復処理して、画面に出力します。
foreach($properties as $key=>$value) {
    echo "<br>";
    echo $key . " : " . $value;
}

?>

</html>
```

この PHP プログラムを実行した出力は、図 6-2 のようになります。

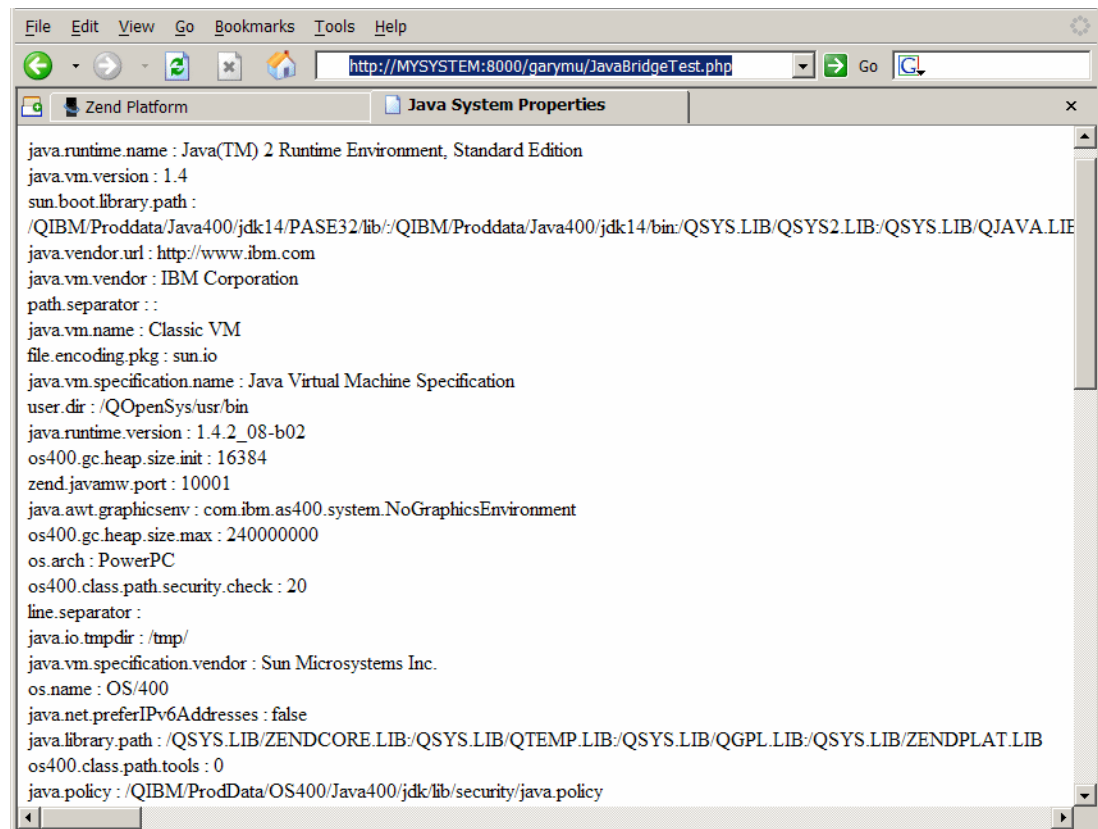


図 6-2 Java システム・プロパティ・プログラムの実行出力

## 6.3.2 IBM Toolbox for Java および PHP

このセクションでは、IBM Toolbox for Java を通じて i5/OS リソースにアクセスする 2 つのサンプル・プログラムを示します。

**制約事項：**本章の執筆時で実行されていた Toolbox 関数は一部のみでした。すべての機能の包括的なテストは、*実行されていません*。

例えば、本章の執筆時点では、i5/OS 上で `ibm.com@.as400.access.CommandCall` クラスを使用してコマンドを実行することはできませんでした。

### メッセージ・キュー・テスト・プログラム

例 6-2 は、i5/OS メッセージ・キューの内容を表示するシンプル・プログラムを示します。出力を参照するには、キューにメッセージが存在する必要があります。

例 6-2 i5/OS メッセージ・キューの全メッセージの表示

```
<html>
<head><title>Message Queue Test</title></head>
<?php

// AS400 接続オブジェクトを取得します。
$as400 = new Java('com.ibm.as400.access.AS400', 'localhost', 'myuserid', 'mypassword');

// MessageQueue オブジェクトを作成します。
$mq = new Java('com.ibm.as400.access.MessageQueue', $as400,
'/QSYS.LIB/QUSRSYS.LIB/GARYMU.MSGQ');

// キューにあるメッセージの数を取得して、プリントアウトします。
$numMessages = $mq->getLength();
echo '<br>Number of messages: ' . $numMessages ;

// メッセージが存在する場合は、それぞれをプリントします。存在しない場合は、そのことを示します。
if (0 < $numMessages) {
    $messages = $mq->getMessages(-1, 0);

    foreach($messages as $value) {
        echo '<br>Message = ' . $value;
    }
} else {
    echo '<br>No messages in queue.';
}

// すべてのサービスを切断します。
$as400->disconnectAllServices();

?>

</html>
```

キューにメッセージが存在する場合のこのプログラムの出力は、図 6-3 (86 ページ) のようになります。

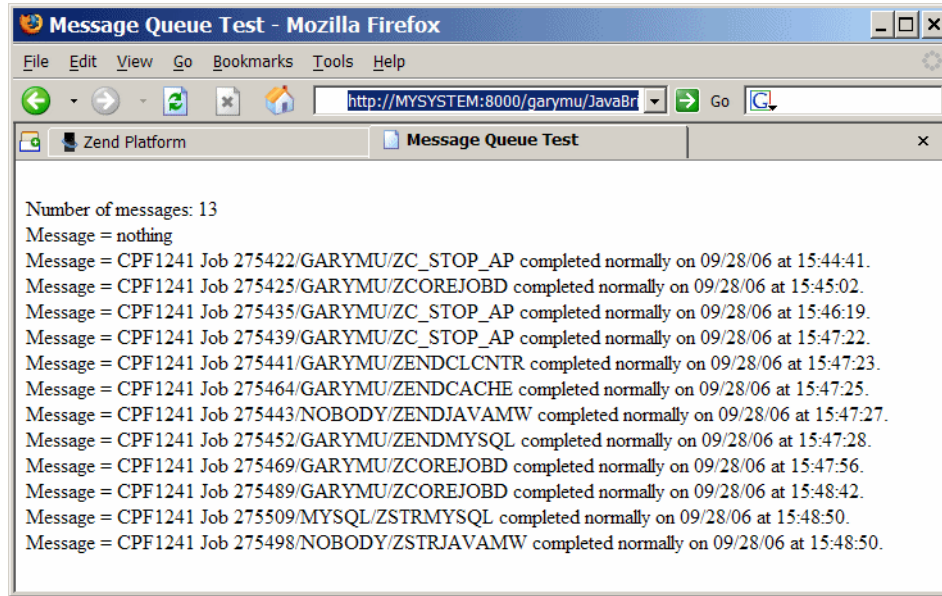


図 6-3 Java Toolbox メッセージ・キュー・プログラムの実行出力

## データ・キュー・テスト・プログラム

例 6-3 は、i5/OS データ・キューのエントリーをポストする方法および読み出す方法を示します。データ・キューが存在するかどうかをテストして、存在しない場合は作成します。存在する場合は、内容を消去します。

### 例 6-3 i5/OS データ・キューの操作

```
<html>
<head><title>Data Queue Test</title></head>
<?php

// AS400 接続オブジェクトを取得します。
$as400 = new Java('com.ibm.as400.access.AS400', 'localhost', 'myuserid', 'mypassword');

// データ・キュー・オブジェクトを取得します。
$dq = new Java('com.ibm.as400.access.DataQueue', $as400,
"/QSYS.LIB/GARYMU.LIB/GARYDATAQ.DTAQ");

// 存在しない場合はデータ・キュー・オブジェクトを i5/OS に作成し、存在する場合は内容を消去し
ます。
if (!$dq->exists()) {
    $dq->create();
} else {
    $dq->clear();
}

// 現在時刻を含むストリングをデータ・キューに書き込みます。
$dq->write("Current date and time: " . date("F j, Y, g:i a"));

// データを読み出します。データ・キューからエントリーは除去されません。
$entry = $dq->peek();
$data = $entry->getString();

// データをブラウザに書き出します。
echo '<br>Data queue data: ' . $data;
```

```
// データ・キューを削除します。
$daq->delete();

// すべてのサービスを切断します。
$as400->disconnectAllServices();
?>

</html>
```

---

図 6-4 は、例 6-3 (86 ページ) のプログラムの出力を示します。

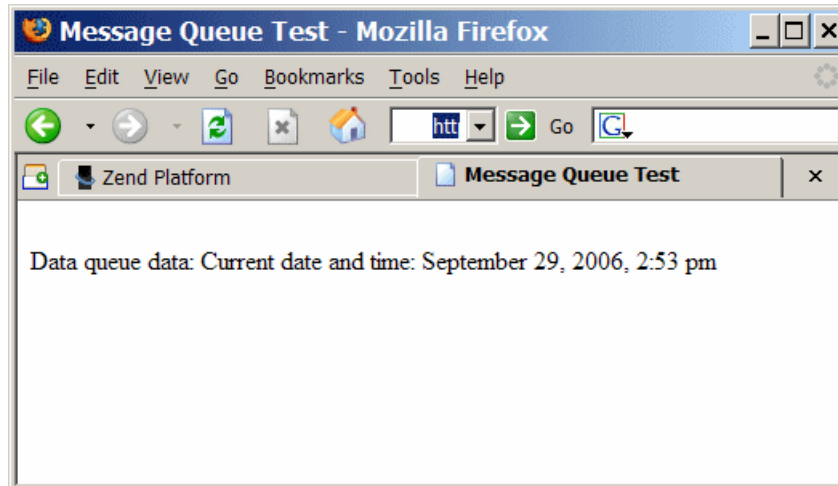


図 6-4 Java Toolbox データ・キュー・プログラムの実行出力

## 6.4 トラブルシューティングのヒント

このセクションでは、Java Bridge に関連する問題の診断を容易にする、いくつかの簡単なトラブルシューティングのヒントを紹介します。

### 6.4.1 php\_error\_log ファイル

通常は php\_error\_log ファイルを調べることからトラブルシューティングを開始します。php\_error\_log ファイルは、/usr/local/Zend/Core/logs ディレクトリーにあります。

例 6-4 は、存在しないコンストラクターを使用してオブジェクトをインスタンス化しようとしたときのエラーを示します。最後の行に、エラーが発生した PHP ファイルの行が示されています。

例 6-4 php\_error\_log ファイルに出力されたエラーのサンプル

---


```
[29-Sep-2006 14:41:19] PHP Fatal error: Uncaught exception 'JavaException' with message 'Java
Exception java.lang.InstantiationException: No matching constructor found
java.lang.InstantiationException: No matching constructor found
    at java.lang.Throwable.<init>(Throwable.java:195)
    at java.lang.Exception.<init>(Exception.java:41)
    at java.lang.InstantiationException.<init>(InstantiationException.java:37)
' in /www/ZendCore/htdocs/garymu/JavaBridgeDataQueuePost.php:9
Stack trace:
#0 /www/ZendCore/htdocs/garymu/JavaBridgeDataQueuePost.php(9): *No
Class!*->java('com.ibm.as400.a...', Object(com.ibm.as400.access.AS400))
```

```
#1 {main}  
thrown in /www/zendcore/htdocs/garymu/JavaBridgeDataQueuePost.php on line 9
```

---

### 6.4.2 var\_dump() 関数

この PHP 組み込み関数は、Java メソッド呼び出しの結果として PHP が受け取る変数の型を特定するために有用です。例えば、例 6-1 (84 ページ) に示す `$properties = $system->getProperties();` という呼び出しでは、実際の Java メソッドが `Properties` オブジェクトを返す場合でも、PHP Array を返します。



# セキュリティ

この章では、i5/OS 上の PHP に固有のセキュリティに関する考慮事項の詳細について説明します。

## 7.1 セキュリティーに関する考慮事項

i5/OS 上の PHP アプリケーションを保護するためには、デフォルト環境の変更が必要です。このセクションでは、PHP で記述されたビジネス・アプリケーションが保護されるようにするために調査する必要があるさまざまな領域について説明します。

言うまでもなく、i5/OS を含む任意のプラットフォームにおけるデータおよびアプリケーションの保護を成功させるための多くの側面があります。IBM Redbook 「*IBM System i Security Guide for IBM i5/OS Version 5 Release 4 (SG24-6668)*」は、i5/OS でセキュリティーをインプリメントするために非常に役立つ全般的なガイドです。

<http://www.redbooks.ibm.com/abstracts/sg246668.html?Open>

### 7.1.1 SSL 構成

SSL (Secure Sockets Layer) は、Web サーバーからユーザーのブラウザーに送信されるデータの暗号化に役立ちます。これにより、貴重な情報がインターネットを通過するときに第三者がデータ・ストリームをインターセプトして情報を抽出することは極めて困難 (不可能とは言えないとしても) になります。また、誰が Web サーバーのどの部分にアクセスしているかが分かるようにするため、適切な認証と権限が整っていることを確認する必要もあります。

Zend Core for i5/OS で SSL を使用する場合の主な特異性は、SSL 構成は i5/OS 内で実行している Apache サーバー (「zendcore」と呼ばれる) に対して行う必要があるということです。「内部の」Apache サーバー (PASE for i5/OS 内で実行する) を構成して SSL 作業を行うことはできません。

i5/OS で SSL を構成する方法の詳細については、次のアドレスの「*IBM HTTP Server (powered by Apache): An Integrated Solution for IBM eSeries iSeries Servers (SG24-6716)*」を参照してください。

<http://www.redbooks.ibm.com/abstracts/sg246716.html?Open>

### 7.1.2 ディレクトリー構造へのアクセス

Zend Core for i5/OS のインストール時に、いくつかのディレクトリーが作成されます。

- ▶ /usr/local/Zend
- ▶ /www/zendcore

これらのディレクトリー内の「ロックする」ことを考慮すべき重要なファイルについて説明します。

#### 構成ファイル

メイン PHP 構成ファイル (php.ini) は、ディレクトリー /usr/local/Zend/Core/etc にあります。このファイル (およびこのディレクトリー内のその他すべてのファイル) は、任意のユーザーが編集できます。特定の管理者のみがこれらのファイルを変更できるようにするためにアクセス属性を変更する必要があると考えられます。

内部 Apache サーバーの場合にも同じことが該当します。その構成ファイル (httpd.conf) は、ディレクトリー /usr/local/Zend/apache2/conf にあります。

外部 Apache サーバー構成ファイル (httpd.conf) は、/www/zendcore/conf ディレクトリーにあります。デフォルトでは、\*PUBLIC はこのファイルに対する書き込み権限を持っていません。



## PHP ソース・ファイル

デフォルトでは、Zend Core for i5/OS の文書ルートは /www/zendcore/htdocs です。すべてのユーザーが、このディレクトリー内のすべてのファイルへのフル・アクセスを持っています。したがって、デフォルトでは、任意の i5/OS ユーザーが PHP サーバーによって提供されるコンテンツを作成できます。また、ユーザーは、このディレクトリー下の任意の既存スクリプトを変更できます。このディレクトリー、およびそれに含まれるすべてのディレクトリーとファイルのデフォルトのセキュリティ属性を変更することが、非常に重要です。

### 7.1.3 リバース・プロキシー

プロキシー HTTP サーバーを使用する主要な理由は 2 つあります。

- ▶ 最初の理由は、パフォーマンスを向上させることです。

プロキシー・サーバーは、静的 Web ページをサーバー始動時に前もってメモリーのキャッシュに入れるために一般に使用されています。これにより、ファイル・システムからコンテンツを取り出すことなく要求元クライアントにページを送信することができます。

パフォーマンスを向上させる 2 番目の方法は、多数のその他の HTTP サーバーに対して要求を転送できる 1 つのプロキシー・サーバーを持つことです。それらの HTTP サーバーのそれぞれが、Web ページまたは Web アプリケーションの特定のセットについてサービスを提供するように構成されています。これは、多数の内部サーバー間で多数の要求の負荷の平衡を取るために役立ちます。

- ▶ プロキシー・サーバーの 2 番目の機能は、セキュリティを向上させることです。

プロキシー・サーバーを使用することにより、ファイアウォールおよび非武装地帯 (DMZ) 外のアクセスを制御できます。これにより、実動レベルのサーバーを保護された内部ネットワーク内に保持できます。また、これは内部サーバーを要求元クライアントから完全に隠すことができるため、要求元クライアントにはサーバーの名前、IP アドレスなどは分かりません。また、プロキシー・サーバーを使用してアクティビティをログに記録し、サービス妨害による攻撃を予防することもできます。

「リバース」プロキシーが何であるかを明確にすると、以下ようになります。リバース・プロキシーは、クライアントからの要求を受け入れて別のサーバーに転送し、その別のサーバーからの応答を受信し、結果をクライアントに戻します。リバース・プロキシーは、クライアントがインターフェースをとる唯一のサーバーです。

## ZENDCORE サーバー・インスタンス

第 3 章『管理』(11 ページ) で説明したように、Zend Core for i5/OS のインストール・プロセスによって、IBM HTTP Server for i5/OS (ZENDCORE と呼ばれる) の完全なインスタンスがデフォルトでインストールされます。これは、ポート 89 で listen し、ポート 8000 の内部 Apache サーバー (PASE for i5/OS 内で実行) に要求を転送します。

ただし、これらの 2 つの HTTP サーバーは同じハードウェア・プラットフォーム上で実行されているため、通常はリバース・プロキシー構成と関連付けられているセキュリティ上の利点 (主に DMZ) は達成できません。したがって、「*IBM HTTP Server (powered by Apache): An Integrated Solution for IBM eSeries iSeries Servers (SG24-6716)*」に記述されているように、完全な DMZ を作成することを強くお勧めします。これは以下の場所にあります。

<http://www.redbooks.ibm.com/abstracts/sg246716.html?Open>

## 7.1.4 ユーザー・プロファイル

第 3 章『管理』(11 ページ)で概要を示したように、Zend Core for the i5/OS のインストール手順によって NOBODY と呼ばれるユーザー・プロファイルが作成されます。すべての PHP ジョブはこのユーザー・プロファイルで実行され、送られてくる特定の要求をその他のユーザー・プロファイルで実行するように指定することはできません。

ibm\_db2 ドライバーと Toolkit を使用して i5/OS のデータおよびリソースにアクセスするときに、特定のユーザー・プロファイルを使用して接続することを指定できます。ユーザー ID とパスワードが指定されない場合、NOBODY プロファイルを使用して接続されます。ユーザー ID とパスワードが指定された場合、その接続で指定された操作 (例えば、SQL QUERY の実行、i5/OS プログラムの実行など) は、その特定ユーザー・プロファイルで実行されます。i5/OS のオブジェクトへのアクセスは、通常の i5/OS オブジェクト・セキュリティ規則に従います。

## 7.1.5 その他の情報

セキュリティに関する考慮事項の詳細については、以下の Web サイトにアクセスしてください。

次のサイトでは、Web セキュリティー一般が説明されています。

<http://www.w3.org/Security/Faq/>

次の情報は PHP に固有です。

<http://www.php.net/manual/en/security.php>



## パフォーマンス

この章では、i5/OS 上の PHP のパフォーマンス・チューニングの詳細について説明します。

**注：**このセクションはあまり包括的ではありません。Zend for i5/OS がマーケットプレイスに広く受け入れられるようになるにつれ、以降の年月ではるかに多くの情報を使用できるようにになります。

## 8.1 パフォーマンス・チューニング

i5/OS 上で最適なパフォーマンスを得るために PHP をチューニングすることは、サポートの提供に多数のコンポーネントや層が関与しているため、簡単なタスクではありません。このセクションでは、それぞれについて詳細に説明します。ほとんどのインスタンスにおいて、低レベルのパフォーマンス・チューニングのヒントは提供せず、その他の優れた情報源への指針を提供します。トピックが i5/OS 上の PHP サポートに固有である場合にのみ詳細に説明します。

### 8.1.1 ハードウェア

System i の機能が低い場合、PHP の実行にも支障をきたします。現在、IBM Systems Workload Estimator (WLE、<http://www-912.ibm.com/estimator/index.html>) は、i5/OS 上での PHP ワークロードの見積もりはサポートしていません。

ただし、Estimator は WebSphere ワークロードのモデル化をサポートします。PHP の場合の大まかなガイドラインとして非 EJB ワークロードを使用することをお勧めします。図 8-1 に推奨パラメーターを示します。

IBM Systems Workload Estimator - Mozilla Firefox

File Edit View Go Bookmarks Tools Help del.icio.us

http://www-912.ibm.com/wle/EstimatorServlet?postcache=11164t

IBM Systems Workload Estimator

United States [select] Terms of use

Search

Home Products Services & solutions Support & downloads My account

Version: 2006.5 fix.1 29-Nov-06 www-912

IBM Systems Workload Estimator

Workload Selection Workload Definition Help/Tutorials

WAS #1

Partition Name: Main #1  
i5/OS™ - V5R4  
No LPAR

WAS Workload Definition

A visit is a group of transactions from a given user.

1. How many visits per hour are anticipated during the **busiest** time of the day? 0.0

2. How many web requests occur during a typical visit? 0.0

3. How many active users are anticipated at the busiest time of the day? \*CALC

A typical web application consists of multiple logical layers. In this application, how are each of the following layers implemented?

4. Presentation JSPs

5. Business Logic Servlets

6. Data Access Direct JDBC

Most WebSphere applications access a database.

7. Is the database for this application:

☒ On the same partition as the WebSphere application

☐ On a separate partition or system

☐ Not used

Back Continue

About IBM Privacy Contact

Adblock Now: 23° F Wed: 31° F Thu: 22° F Fri: 32° F Sat: 29° F Sun: 24° F

図 8-1 Workload Estimator を使用した PHP ワークロード見積もりの推奨パラメーター

**重要：**このような方法で Workload Estimator を使用した PHP 動作のモデル化はかなり大まかなものであり、サイジングの開始点としてのみ使用するべきです。実際の結果は大きく異なる可能性があります。

### 8.1.2 i5/OS

「コア」i5/OS オペレーティング・システムが十分にチューニングされていることが非常に重要です。これを行う方法に関する豊富な情報があります。多くの有益な情報へのリンクについては、以下にアクセスしてください。

<http://www.ibm.com/servers/eserver/support/series/planning/perfcapacity.html>

使用可能な主なパフォーマンス分析ツールの概要をうまく説明する Redpaper を入手できます。これは「*IBM eSeries iSeries Performance Management Tools (REDP-4026)*」と呼ばれ、以下で入手できます。

<http://www.redbooks.ibm.com/abstracts/redp4026.html?Open>

### 8.1.3 i5/OS PASE

i5/OS PASE 環境に使用できるパフォーマンス・チューニングはほとんどありません。この環境のチューニングの大部分は、コア・オペレーティング・システム i5/OS がどのように構成されているかということに直接由来します。

### 8.1.4 DB2 for i5/OS

i5/OS 自体の構成に加えて、DB2 for i5/OS のチューニングが、パフォーマンスに関して注意を必要とするその他の主な領域であると考えられます。基本 i5/OS の場合と同様に、この重要なトピックに関する以下のような大量の情報が存在します。

- ▶ DB2 for i5/OS tuning: *Preparing for and Tuning the SQL Query Engine on DB2 for i5/OS* (SG24-6598)

<http://www.redbooks.ibm.com/abstracts/sg246598.html?Open>

- ▶ *SQL Performance Diagnosis on IBM DB2 Universal Database for iSeries* (SG24-6654)

<http://www.redbooks.ibm.com/abstracts/sg246654.html?Open>

### 8.1.5 Apache

i5/OS 上の Apache のパフォーマンスを最大にするチューニングに関して役立つ情報を、「*IBM HTTP Server (powered by Apache): An Integrated Solution for IBM eSeries iSeries Servers* (SG24-6716)」で入手できます。これは以下にあります。

<http://www.redbooks.ibm.com/abstracts/sg246716.html?Open>

### 8.1.6 Zend Core for i5/OS

幸いなことに、さまざまなオペレーティング・システムで Zend Core を実行させるための既存情報は大量にあり、その知識の大部分を i5/OS の場合に直接「移植」できます。

Zend Developer Zone (<http://devzone.zend.com>) には、PHP のパフォーマンスに関する多くの記事とチュートリアルがあります。Google 検索により、PHP のパフォーマンス・チューニングのヒントに関するその他の多くの最新記事が明らかになります。



## トラブルシューティング

このセクションでは、PHP 環境での問題の特定と解決に焦点を当てます。

Web テクノロジー一般に内在する困難さは、それらが通常は複数のコンポーネントから構成されているということです。どのコンポーネントで問題が発生しているかを判別することが困難である可能性があるため、これは Web 環境のトラブルシューティングのプロセスに影響を与えます。図 9-1 は、障害が発生する可能性があるいくつかの場所を示しています。

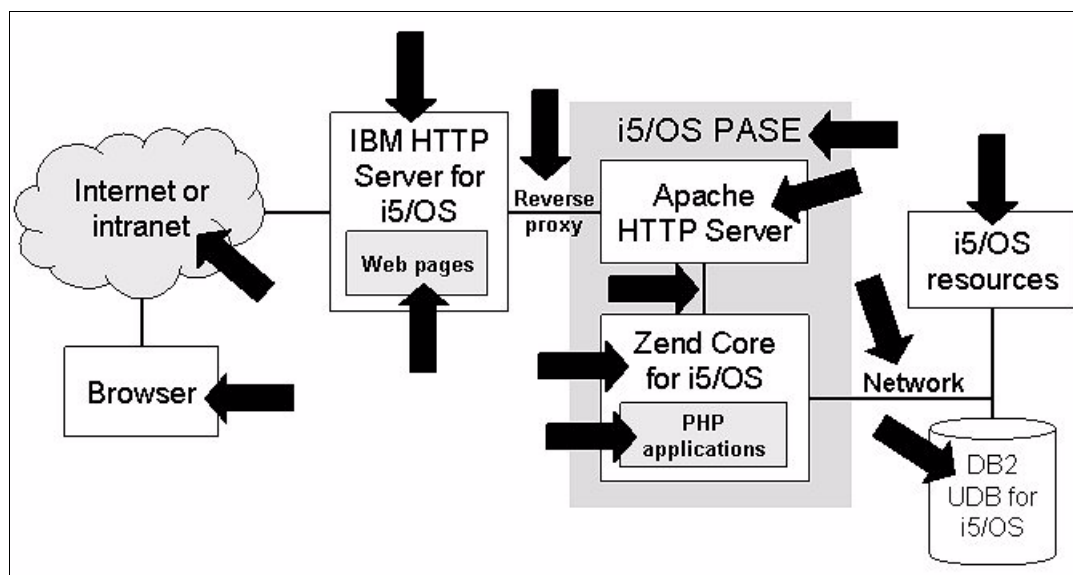


図 9-1 複数コンポーネント Web トポロジーのトラブルシューティング

**注 :** Web 環境は脆弱なように思われるかもしれませんが、環境を正しく構成してテストした場合、発生する障害の大部分はアプリケーション・エラーの結果であり、環境を構成するコンポーネントが原因ではないということに注意することが重要です。

複数コンポーネント・トポロジーのトラブルシューティングを行う場合、まず、問題を特定のコンポーネントに絞り込む必要があります。トポロジー内のどこまで要求が届いているか

を判別し、問題の調査を開始する最適な箇所を特定します。しかし、問題のほとんどの症状は最初にブラウザーに現れることを考慮すると、これを行うのは困難である可能性があります。トポロジーを最初から順番に処理してください。特定コンポーネントが正しく機能しているかどうかを判別するために役立つ一連の質問と関連テストを構成してください。

以下は、問題を見つけるための質問の例です。

- ▶ ブラウザーで同様のコンテンツを参照できますか？
- ▶ サーバーに ping できますか？
- ▶ HTTP サーバーから静的 Web ページにアクセスできますか？
- ▶ HTTP サーバーからその他の動的ページにアクセスできますか？
- ▶ 同じ Zend Core インストール済み環境からその他の PHP アプリケーションにアクセスできますか？
- ▶ アプリケーションが外部リソース ( データベースやファイルなど ) にアクセスする場合、手動でそれらにアクセスできますか？



## 9.1 始める前に

Web トポロジーを順番に処理してそれぞれのコンポーネントをテストすることは、退屈で時間がかかる可能性があります。多くの場合、最も一般的な問題を除外することにより、トラブルシューティングの時間を大幅に削減できます。

アプリケーション・コードが誤っているために問題が発生していることが最も多いため、アプリケーション・コードが正しいかどうかを確認することが役立ちます。アプリケーションが開発環境では機能するがテスト環境では機能しない場合、アプリケーション・コードは問題の原因ではないと考えられます。そうでない場合は、アプリケーションをデバッグする必要があります。Zend Studio for i5/OS は、PHP アプリケーション用の良いデバッガーを提供します。デバッグは、アプリケーション開発の章で扱われています。

次に、すべてのサーバーが開始されていることを確認します。適切なジョブが実行されていることの確認方法の詳細については、管理の章を参照してください。1 つ以上のサーバーが停止されている場合、それらを開始してください。トラブルシューティング・ルーチンをかなり進めた後で、各サーバーが実行されていることを最初に確認すれば数分間で問題を解決できたとわかった場合ほど、つまらないことはありません。

最初に行うと良い別の手順は、最新の修正がソフトウェア・スタックに適用されていることを確認することです。その問題が別のユーザーによって検出されていて、修正で対処されている可能性があります。その問題の原因を検出できないと後で判断した場合、サポート担当者が最初に尋ねる質問は、最新の修正が適用されているかどうかということです。

## 9.2 ブラウザーのトラブルシューティング

ブラウザーの問題は、実際はめったにありません。PHP などのサーバー・サイド・テクノロジーを使用する利点の 1 つは、ブラウザーは論理処理をほとんど、またはまったく行わず、応答は静的 Web ページとしてブラウザーに送信されるということです。ただし、ブラウザーが JavaScript などのクライアント・サイド・アプリケーション・コードを処理する場合は、そのコードが正しいことを確認してください。最新のほとんどのブラウザーには、スクリプト・コードを検査するユーティリティが組み込まれています。また、コードにブラウザー間の互換性がない可能性があることにも注意してください。つまり、いくつかの異なるブラウザー (Microsoft® Internet Explorer® や Mozilla Firefox など) でコードが機能することを確認してください。

アプリケーションがデフォルトのブラウザーのインストール済み環境には含まれないテクノロジー (特別なタイプのメディア用のプラグインなど) を使用する場合は、必要なソフトウェアがインストールされていることを確認してください。

ほとんどのブラウザーでは、ユーザーがクライアント・サイド・テクノロジー (JavaScript、Java アプレット、Cookie など) を使用不可に設定できます。アプリケーションがそれらに依存している場合、それらが使用不可に設定されていないことを確認してください。(アプリケーションを使用するためにはこれらのテクノロジーを使用可能に設定しておく必要があることをユーザーに通知することも役立ちます。)

ブラウザーまたはワークステーションに基づくファイアウォールの中には、ポップアップをブロックし、広告を除去し、クライアント・サイド・テクノロジーを使用不可にするために、Web ページにコードを追加するものもあります。この挿入されたコードが、Web アプリケーションで問題を発生させる可能性があります。ブラウザーでは、通常、現在ロードされているページのソース・コードを参照できます。ソースを確認し、アプリケーションに影響を与える可能性があるコードが追加されているかどうかを調べてください。

## 9.3 Web サーバーのトラブルシューティング

Web サーバーについて最初に確認することは、それらが使用する TCP/IP ポートがその他のアプリケーションと競合していないことの確認です。ポートが競合している場合、サーバーは始動するけれど継続して稼働しないのが特徴的な症状です。NETSTAT \*CNN コマンドおよび F14 を使用し、システムのポート使用状況を参照してください。

オプション 8 (ジョブの表示) を指定し、特定のポートをどのジョブが使用しているかを表示してください。図 9-2 は、I5\_COMD ジョブがポート 6077 を使用していることを示しています。デフォルト・ポートは 6078 であることに注意してください。これは ZCMENU のオプションを使用して変更できます。

Work with TCP/IP Connection Status

Type options, press Enter.  
3=Enable debug 4=End 5=Display details 6=Disable  
8=Display jobs

Display Jobs Using Connection

Connection type . . . . . : \*TCP  
Local address . . . . . : \*  
Local port . . . . . : 6077  
Remote address . . . . . : \*  
Remote port . . . . . : \*

Type options, press Enter.  
5=Work with job

Opt	Name	User	Number	Opt	Name
-	I5_COMD	QTCP	292914		
		*		*	
8	*	*		*	
-	*	*		*	
-	*	*		*	
-	*	*		*	

Local Port	Idle Time	State
515	145:05:43	Listen
657	145:05:53	Listen
2001	000:28:52	Listen
2301	145:05:59	Listen
3000	145:06:03	Listen
4111	145:04:03	Listen
5544	145:05:25	Listen
5555	145:05:14	Listen
6077	042:24:01	Listen
8000	000:10:43	Listen
8470	145:05:10	Listen
8471	145:05:09	Listen

図 9-2 ポートを使用しているジョブの表示

また、IBM HTTP Server for i5/OS の構成で、アプリケーションの呼び出しに正しい URL (ポート番号も含む) を使用していることを確認する必要があります。URL は、通常 Web サーバーの文書ルート・ディレクトリ構造からの Web ページの相対位置に基づいています。ただし、文書ルート構造を隠すために、別名およびリダイレクトを構成できます。また、Configure TCP/IP (CFGTCIP) コマンドを使用してシステムの TCP/IP 構成を確認し、URL で使用しているホスト名が正しいことを確認することもできます。

Web サーバーの構成で確認するもう 1 つの項目は、IBM HTTP Server for i5/OS サーバーが適切なポート番号 (デフォルトは 8000) を使用して PASE Apache HTTP Server に要求を転送しているということです。

LDAP Configuration  
Servlet and JSP Enablement

Server Properties

Tools  
Display Configuration File  
Edit Configuration File  
Directive Index  
Real Time Server Statistics

General Settings Forward Proxy Reverse Proxy Proxy Chaining

Reverse proxy capabilities: Enabled

Proxy requests to remote servers:

	Request type	Local virtual path	Remote server URL
Example	Client requests	/mirror/foo	http://www.myserver.com/
Example	Redirected requests	/mirror/foo	http://www.myserver.com/
<input type="radio"/>	Client requests	/	http://127.0.0.1:8000/
<input type="radio"/>	Redirected requests	/	http://127.0.0.1:8000/

Add

図 9-3 Web サーバーのデフォルトのポート番号

Web サーバーのログにエラー・メッセージがないか確認してください。どちらの Web サーバーも Apache HTTP Server に基づいているため、それらのログ機能は類似しています。デフォルトでは、Web サーバーは、エラー・ログとアクセス・ログという、2 つのログに情報を収集します。エラー・ログの値は明かです。アクセス・ログは、アクセスされているサーバー・リソースを記録します。両方の Web サーバーに対してアクセス・ログを使用し、両方のサーバーに要求が届いたことを確認できます。その要求が IBM HTTP Server for i5/OS アクセス・ログには記録され、PASE Apache HTTP Server アクセス・ログには記録されていない場合、問題が 2 つのサーバー間のリンケージにあるに違いないと分かります。

また、ユーザーが決定する任意の基準に基づいたカスタム・ログを使用するようにサーバーを構成することもできます。この件および Web サーバー・ローグ一般についての詳細は、以下の製品資料を参照してください。

- ▶ IBM HTTP Server for i5/OS

<http://publib.boulder.ibm.com/infocenter/iserics/v5r4/topic/rzaie/rzaieconfiglogs.htm>

- ▶ Apache HTTP Server

<http://httpd.apache.org/docs/2.0/logs.html>

## 9.4 Zend Core のトラブルシューティング

Zend Core のログ・ファイルは /usr/local/Zend/Core/logs ディレクトリーにあります。以下の 2 つの PHP ログがあります。

- ▶ php\_error\_log: このログには PHP スクリプトのエラー・メッセージが含まれています。
- ▶ ini\_modifier\_log: Zend Core の構成は php.ini ファイルに保管されています。このファイル (つまり、構成) に対する変更は ini\_modifier\_log ファイルに記録されます。このログで、アプリケーションに影響を与える可能性がある Zend Core に対する最近の変更がないかどうかを確認できます。

内部 Apache サーバーに対する次の 2 つのログが /usr/local/Zend/apache2/logs ディレクトリーにあります。

- ▶ access\_log: このログには、そのサーバーに対して行われたすべてのアクセスのリストが含まれています。これには、(構成に応じて) 要求元 IP アドレス、要求のタイプ (GET、PUT)、要求されたリソース (URL)、および戻りコードがリストされます。
- ▶ error\_log: このログには、PASE for i5/OS 内で実行している内部 Apache サーバーで発生したエラーのリストが含まれています。

エラー処理およびログギングは、Zend Core コンソールで構成されます。「**Configuration**」タブをクリックし、「**PHP**」タブをクリックします。図 9-4 に示されている『**Error Handling and Logging**』セクションを展開します。

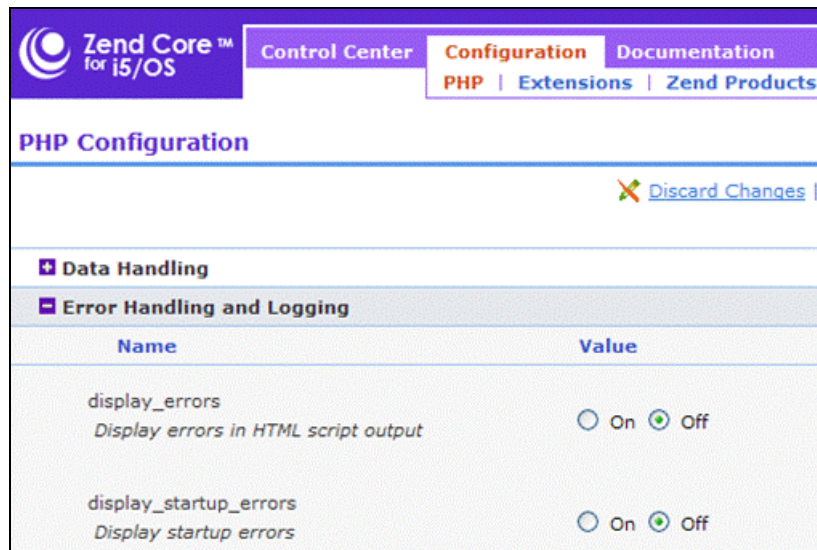


図 9-4 コンソールでのエラー処理およびロギング

コンソールにはいくつかの設定値が含まれています。以下は、知っておく必要がある重要な設定値です。

- ▶ display\_errors: 「On」に設定されている場合、エラー・メッセージはブラウザに出力されます。この機能は、アプリケーション処理中のどの時点でエラーが発生するかを判別するために非常に役立つ可能性があります。
- ▶ display\_startup\_errors: デフォルトでは、始動時のエラーは抑止されます。サーバーの始動に問題がある場合は、この設定値を「On」に変更します。
- ▶ error\_log: エラー・ログ・ファイルのパスと名前を指定します。
- ▶ error\_reporting: エラー・レポートのレベルを設定します。デフォルトでは、このプロパティは最高のレベル(「ALL」)に設定されています。実動レベルのシステムの場合、この値を下げるすることができます。
- ▶ log\_errors\_max\_len: ログ・ファイルの最大サイズを設定します。ログ・ファイルがこのサイズに達すると、新規ファイルへの記録を始めます。

PHP のログの詳細については、次のアドレスの PHP 資料の「Error Handling and Logging Functions」を参照してください。

<http://www.php.net/manual/en/ref.errorfunc.php>

アプリケーションが特定の PHP 拡張機能を使用する場合は、コンソールを使用してその拡張機能が使用可能になっていることを確認してください。「Configuration」タブをクリックしてから、「Extensions」タブをクリックします。拡張機能は、それに対応するアイコン(それらの状況を知らせる)とともにリストされます。図 9-5 に示したような赤色の感嘆符は、拡張機能にエラーがあることを示しています。

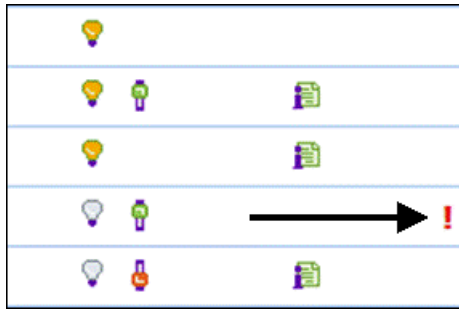


図 9-5 拡張機能の状況を表すアイコン

PASE Apache HTTP Server からの要求が Zend Core に届いていないと考えられる場合は、`/usr/local/Zend/apache2/conf` ディレクトリーにある Apache 構成ファイル `httpd.conf` の Zend ディレクティブを確認してください。Zend ディレクティブはファイルの末尾にあります。このファイルを定期的にバックアップして、比較に使用できるものがあるようにしておいてください。

## 9.5 アプリケーション・リソースのトラブルシューティング

アプリケーションがシステム・リソース (ファイルやデータベースなど) にアクセスする場合、この領域のトラブルシューティングに使用できるいくつかの技法があります。多くの場合、標準 i5/OS トラブルシューティング機能を使用します。i5/OS のログおよびリソースの詳細については、次のアドレスの iSeries Information Center の「トラブルシューティング」を参照してください。

<http://publib.boulder.ibm.com/infocenter/iseries/v5r4/topic/rzahb/rzahbrtrbshoo1.htm>

### 9.5.1 オブジェクト権限の処理

アプリケーションが使用するユーザー・プロファイルがリソースに対する十分な権限を持っていない場合にリソース・アクセスが失敗することはよくあります。これには、ライブラリー、ファイル、プログラムなどのリソースが含まれます。i5/OS オブジェクトの権限を確認する場合にはオブジェクト権限表示 (DSPOBJAUT) コマンド、Integrated File System (IFS) 内のオブジェクトの権限を確認する場合には DSPAUT (権限の表示) コマンドを使用します。アクセスに使用するユーザー・プロファイル (ユーザー指定、またはデフォルトの NOBODY のいずれか) は、通常少なくとも読み取り (\*R) 権限を必要とします。

図 9-6 は、PHPTRB ライブラリーに対する権限を示しています。\*PUBLIC ユーザー権限が \*EXCLUDE に設定されており、そのオブジェクトを所有するユーザー・プロファイルのみがそれを変更できることに注意してください。この場合 NOBODY ユーザー・プロファイル (またはアプリケーションが使用する非デフォルト・プロファイル) に \*READ 権限を付与する必要があります。

Display Object Authority		
Object . . . . .	PHPTRB	Owner . .
Library . . . . .	QSYS	Primary gr
Object type . . . . .	*LIB	ASP device
Object secured by authorization list . . . . .		
User	Group	Object Authority
*PUBLIC		*EXCLUDE
QDFTOWN		*CHANGE

図 9-6 PHPTRB.LIB に対する権限の表示

Grant Object Authority (GRTOBJAUT) コマンドを使用して、ユーザー・プロファイルに特定リソースへの権限を与えることができます。

## 9.5.2 i5/OS ジョブ・ログの処理

ほとんどの Zend Core ジョブは、i5/OS ジョブ・ログにメッセージを記録しません。エラーは製品ログ・ファイルに書き込まれます。ただし、PHP アプリケーションが独自のジョブ内で実行されるリソース・サービスにアクセスする場合は、そのジョブ・ログ内のメッセージを参照できます。

ジョブ・ログにアクセスするには、以下の手順を実行します。

1. 活動ジョブの処理 (WRKACTJOB) コマンドを使用して、実行中のジョブを表示します。サブシステム (SBS) パラメーターを使用して、特定サブシステム内で実行中のジョブのみを表示できます。
2. ジョブの前のオプション 5 (活動ジョブの処理) を指定します。
3. 「ジョブの処理」画面で、コマンド行にオプション 10 (ジョブ・ログの表示) を指定します。

**ヒント:** ジョブ・ログ表示でメッセージの詳細を参照するには、メッセージの上にカーソルを移動し、F1 を押します。

「ジョブの処理」画面には、コマンド・スタックの表示など、トラブルシューティングに役立つその他のオプションがあります。

## 9.5.3 スプール・ファイルの処理

ジョブが異常終了した場合、出力キュー QEZJOBLOG のスプール・ファイルにエラー・メッセージが書き込まれます。これらのファイルは、データベース・アクセスの問題を突き止めるために非常に役立ちます。

WRKSPLF SELECT(<user\_profile>) コマンドを実行して、既に実行中ではないジョブのメッセージを表示します (<user\_profile> はそのジョブが実行されていたユーザー・プロファイル)。表示するファイルの前のオプション 5 (表示) を指定します。

図 9-7 では、PHPUSER ユーザー・プロファイルは QSQSRVR ユーザー・データによって指定されたデータベース接続に使用されています。これらのファイル (QPJOBLOG) には、データベースへのアクセスに関するメッセージが含まれています。



Work with All Spooled Files								
Type options, press Enter.								
1=Send 2=Change 3=Hold 4=Delete 5=Display 6=Release 7=Messages								
8=Attributes 9=Work with printing status								
Opt	File	User	Device or Queue	User Data	Sts	Total Pages	Cur Page	Copy
—	QPJOBLOG	PHPUSER	QEZJOBLOG	QSQRVR	RDY	2		1
—	QPJOBLOG	PHPUSER	QEZJOBLOG	QSQRVR	RDY	2		1
—	QPJOBLOG	PHPUSER	QEZJOBLOG	QSQRVR	RDY	2		1
—	QPJOBLOG	PHPUSER	QEZJOBLOG	QSQRVR	RDY	2		1
—	QPJOBLOG	PHPUSER	QEZJOBLOG	QSQRVR	RDY	2		1
—	QPJOBLOG	PHPUSER	QEZJOBLOG	QSQRVR	RDY	2		1
5	QPJOBLOG	PHPUSER	QEZJOBLOG	QSQRVR	RDY	2		1
—	QPJOBLOG	PHPUSER	QEZJOBLOG	QSQRVR	RDY	2		1
—	QPJOBLOG	PHPUSER	QEZJOBLOG	QSQRVR	RDY	2		1

図9-7 PHPUSER 出力キューのプール・ファイル

## 9.5.4 ヒストリー・ログの処理

問題のソースを判別するためにシステム全体のビューが必要な場合もあります。i5/OS ヒストリー・ログ (QHST) には、システム上のさまざまなアクティビティからのメッセージが含まれています。ジョブがいつ開始し停止したか、および何らかのイベントによりジョブが終了したのかどうかを判別するために、このヒストリー情報が役立ちます。また、ハードウェアの問題に関するメッセージも記録されます。図 9-8 に、ヒストリー・ログのサンプルを示します。

Display History Log Contents	
Subsystem ZEND in library ZENDCORE starting.	
Subsystem ZEND started.	
Job 291064/QTCP/I5_CMD started on 09/03/06 at 20:51:54 in subsystem ZEND in	
Job 291065/ZENDADMIN/ZC_STR_ASJ started on 09/03/06 at 20:51:55 in subsystem	
Job 291066/ZENDADMIN/ZC_STR_PRN started on 09/03/06 at 20:51:55 in subsystem	
Job 291067/ZENDADMIN/ZCOREJOB started on 09/03/06 at 20:51:55 in subsystem Z	
Job 291069/QTMHHTP/ZENDCORE started on 09/03/06 at 20:51:55 in subsystem QHT	
Job 291067/ZENDADMIN/ZCOREJOB ended on 09/03/06 at 20:51:55; 1 seconds used;	
Job 291066/ZENDADMIN/ZC_STR_PRN ended on 09/03/06 at 20:51:55; 1 seconds used	
Sep 3 20:51:55 prngd[5842]: prngd 0.9.29 (12 Jul 2004) started up for user u	
Sep 3 20:51:55 prngd[5842]: have 7 out of 2000 filedescriptors open	
Job 291065/ZENDADMIN/ZC_STR_ASJ ended on 09/03/06 at 20:51:56; 1 seconds used	
Job 291077/QSECOFR/ZENDCOREAP started on 09/03/06 at 20:51:56 in subsystem ZE	
Job 291084/QTMHHTP/ZENDCORE started on 09/03/06 at 20:51:57 in subsystem QHT	
Job 291085/QTMHHTP/ZENDCORE started on 09/03/06 at 20:51:57 in subsystem QHT	
More...	
Press Enter to continue.	

図9-8 ヒストリー・ログ (QHST)

実動システム上のヒストリー・ログは非常に大きくなる可能性があります。ヒストリー・ログを表示する場合、特定の期間に対するメッセージのみを表示する必要があります。Display Log (DSPLOG) コマンドを入力し、F4 を押して期間を入力します。

## 9.5.5 メッセージ・キューの処理

メッセージ・キューは、i5/OS ユーザー・プロファイルと関連付けられています。これらのキューに、有益なトラブルシューティング情報が書き込まれる場合もあります。

WRKMSGQ MSGQ(<user\_profile>) で使用します。<user\_profile> は、ユーザー・プロファイルの名前です。

次のリストに、メッセージ・キューを確認すべきユーザー・プロファイルを示します。

- ▶ QSYSOPR: ここには、システム管理メッセージが記録されます。
- ▶ QTCP: ここには、TCP/IP および i5 Toolkit for PHP に関するメッセージが書き込まれます。
- ▶ NOBODY: デフォルトでは、Zend Core および PASE Apache HTTP Server はこのユーザー・プロファイルで実行されます。さらに、接続に対してユーザー・プロファイルが指定されない場合、システム・アクセスは NOBODY プロファイルで発生します。
- ▶ QTMHHTTP: デフォルトでは、IBM HTTP Server for i5/OS はこのユーザー・プロファイルで実行されます。

## 9.5.6 デバッガーの使用

デバッガーの主な目的はアプリケーション・コード内のエラーを特定することですが、リソース・アクセスを調べるために使用することもできます。戻り値をモニターすることにより、失敗するかまたは予期しないエラーを戻すアクセスを特定できます。Zend Studio for i5/OS デバッガーを使用してアプリケーション・コードをステップスルーし、リソース・アクセスを検証します。

## 9.5.7 Toolkit アクティビティーのロギング

このセットアップにより ZENDCORE ライブラリーに LOGFILE が作成され、そこに Toolkit が PHP スクリプトから受信した要求および i5/OS システムから受信したメッセージが書き込まれます。

- ▶ Toolkit ログ・ファイルのセットアップ:  
ADDLIBLE ZENDCORE  
CFGGEAC LIB(ZENDCORE) LOGLEV(4) JOBLOG(\*YES)
- ▶ Toolkit ロギングをオフにするには、次のコマンドを実行します。  
CFGGEAC LIB(ZENDCORE) LOGLEV(0) JOBLOG(\*NO)





## グローバルゼーション

本章は i5/OS で PHP を実行する際の、英語以外の言語サポートについて詳述します。現状、PHP はユニコードをサポートしないため、言語サポートは多少複雑になります。具体的な例を示しながら、キリル文字アルファベットを有効にする方法を解説します。キリル文字アルファベットが有効になることで、非定形のコード・ページや CCSID を使用できることから、興味深い内容といえます。

さらに、DB2 からの照会が可能な上、PHP で使用できる英語以外の文字データの種類についても説明します。

## 10.1 概説

本セクションでは PHP 5 の既存の国際化対応について概説し、i5/OS での PHP サポートに関連する各「レイヤー」の説明のほか、その「レイヤー」が国際化対応にもたらす影響についても取り扱います。

### 10.1.1 PHP 国際化対応

PHP 5 ではユニコードがサポートされていないため、国際化対応はより複雑になります。Zend 社によれば、ユニコード対応は次期リリース (PHP 6) で追加されるとのことですが、それまでこの機能の代替となる方法を見つける必要があります。

UTF-8 はサポート対象です ( 詳細な仕様については <http://tools.ietf.org/html/rfc3629> を参照してください )。 <http://en.wikipedia.org/wiki/Utf-8> でもわかりやすい概説を参照できます。i5/OS では CCSID 1208 が UTF-8 対応を意味します。

UTF-8 を使用するために PHP を構成するには、php.ini ファイル (\usr\local\Zend\Core\etc にあります ) を変更する必要があります。このファイルを編集し、以下の行を既存の ( コメント化されている ) エントリーの下に追加します。

```
default_charset = "UTF-8";
```

これにより、英語、スペイン語、ドイツ語、フランス語などの言語で作成されたページは、変更なしで継続して使用できます。

### 10.1.2 i5/OS のレイヤー

PHP サポートの i5/OS へのインプリメント方法を考慮に入れると、実行中にブラウザと実際の PHP ページとの間でデータを受け渡しするときに、データを直接処理するコンポーネントが複数存在します。これらのコンポーネントが各国語サポートにどのような影響を及ぼすかを理解することは極めて重要です。

#### i5/OS

従来、i5/OS は EBCDIC ベースのマシンで使用されてきました。マシン上で実行しているアプリケーションの大部分は EBCDIC コード化文字セット ID (CCSID) でエンコードされたデータを使用します。i5/OS でサポートされている CCSID のリストについては、以下の URL を参照します。

<http://publib.boulder.ibm.com/infocenter/iseres/v5r4/index.jsp?topic=/nls/rbagsccsidcdepgscharsets.htm>

i5/OS 上でアプリケーションを実行する場合、CCSID が頻繁に変換されることがあり、その変換は自動的に行われます。例えば、Java はユニコード・ベースの言語です。DB2 for i5/OS から JDBC ドライバー経由で EBCDIC データを抽出する場合、EBCDIC からユニコードへの自動変換が行われます。同様の変換は頻繁に行われ、アプリケーション開発者やユーザーが気付かないうちに実行される場合もしばしばあります。

PHP は IBM i5/OS Portable Application Solutions Environment (PASE) を備えた i5/OS 上で実行されます。i5/OS で i5/OS PASE シェルを作成すると、i5/OS ジョブのグローバルゼーション関連構成設定の多くが i5/OS PASE 環境に渡されます。

i5/OS グローバリゼーションについて詳しくは、以下の URL を参照してください。

<http://publib.boulder.ibm.com/infocenter/iseres/v5r4/index.jsp?topic=/nls/rbagsglobalmain.htm>

## i5/OS PASE

i5/OS PASE 環境は統合されたランタイムで、i5/OS から AIX アプリケーションを実行しやすくします。以下は i5/OS PASE が i5/OS および System i のハードウェアとどのように相互作用するかを簡略化したものです。

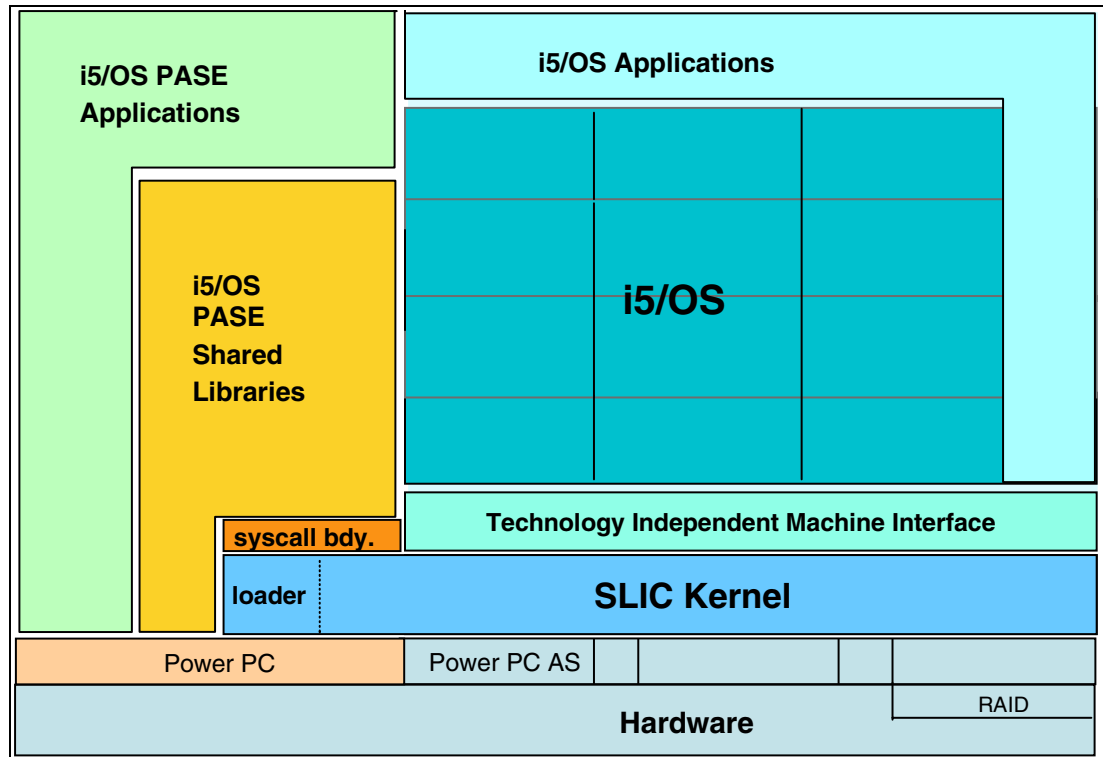


図 10-1 i5/OS PASE の簡略図

PHP は i5/OS で開始される i5/OS PASE アプリケーションとしてインプリメントされています。このことから、PHP が稼働する i5/OS PASE 環境は、その環境を開始した i5/OS ジョブのグローバリゼーションを継承します。例えば、初めに i5/OS から以下のコマンドを実行します。

```
CHGJOB LANGID(RUS) CNTRYID(RU) CCSID(1025)
```

このコマンドによりジョブの言語 ID と国別 ID のほか、ジョブの CCSID が 1025 (キリル文字の基本 EBCDIC CCSID) に変更されます。

続いて、i5/OS PASE 端末を開始します。

```
CALL PGM(QP2TERM)
```

例 10-1 で示されるように、一部のグローバリゼーション環境変数は自動的に設定されます (太字)。

### 例 10-1 新しい i5/OS PASE 環境の環境変数

```
$
> env
_=/QOpenSys/usr/bin/env
LANG=ru_RU
PASE_LANG=ru_RU
QIBM_PASE_DESCRIPTOR_STUDIO=T
PATH=/QOpenSys/usr/bin:/usr/ccs/bin:/QOpenSys/usr/bin/X11:/usr/sbin:./usr/bin
ROWS=17
```

```

QIBM_DESCRIPTOR_STDIN=CRLN=Y
COLUMNS=129
PASE_PATH=/QOpenSys/usr/bin:/usr/ccs/bin:/QOpenSys/usr/bin/X11:/usr/sbin:/usr/bin
LC__FASTMSG=true
LOGNAME=QSECOFR
LOCPATH=/usr/lib/nls/loc
PASE_LC__FASTMSG=true
QIBM_IFS_OPEN_MAX=66000
QIBM_USE_DESCRIPTOR_STDIO=I
QIBM_PASE_CCSID=915
PASE_SHELL=/QOpenSys/usr/bin/sh
SHELL=/QOpenSys/usr/bin/sh
PASE_LOCPATH=/usr/lib/nls/loc
HOME=/home/QSECOFR
PASE_TZ=UTC0
PASE_NLSPATH=/usr/lib/nls/msg/%L/%N:/usr/lib/nls/msg/%L/%N.cat:/usr/lib/nls/msg
/ru_RU/%N:/usr/lib/nls/msg/ru_RU/%N.cat
PWD=/
TZ=UTC0
NLSPATH=/usr/lib/nls/msg/%L/%N:/usr/lib/nls/msg/%L/%N.cat:/usr/lib/nls/msg/ru_RU/%
N:/usr/lib/nls/msg/ru_RU/%N.cat
$

```

===>

F3= 終了 F6= 印刷 F9= コマンドの複写 F11= 切り捨て / 折り返し  
F13= 消去 F17= 先頭 F18= 最後 F21=CL コマンド入力

---

i5/OS PASE が CCSID 915 で実行していることがわかります。偶然ですが、この値はキリル文字の主要 ASCII CCSID の 1 つです。

i5/OS PASE について詳しくは、以下の URL を参照してください。

<http://publib.boulder.ibm.com/infocenter/series/v5r4/index.jsp?topic=rzalf/rzalfintro.htm>

## PHP

言語の観点では、PHP の最も重要な構成パラメーターは default\_charset ディレクティブと言えます。DB2 および ODBC ドライバーなどの一部のプラグインは、このディレクティブの値を使用して、どのような種類のデータ変換が必要かを判別します。

このディレクティブは php.ini ファイルにあり、この ini ファイルは i5/OS のデフォルトでは /usr/local/Zend/Core/etc ディレクトリーに配置されています。このディレクティブを設定することで、ブラウザーに送信すべき構成済み文字セットが、PHP によって Content-Type ヘッダーに追加されます。

## Apache

通常、グローバル化設定を変更するために、Apache 環境側で特別な設定を行う必要はありません。HTTP サーバーは主に、PHP で生成されたデータを素通りさせるための「パイプ」としての役割を果たします。

PHP ファイル以外に HTML ファイルも使用するときは、場合によっては Content-Type ヘッダーが正しく設定されていることを確認する必要があります。ヘッダーを正しく設定するには、以下のように、HTML ファイルのヘッダー・セクションのヘッダーにコードを手動で追加する方法を取ります。

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```

また、代わりに以下のように指定することもできます。

```
<?php
    header('Content-Type:text/html; charset=utf-8');
?>
```

この構成は UTF-8 用であることに注意してください。

**重要** :PHP ファイルが表示用のデータを実際書き出す前に、必ず Content-Type ヘッダーを出力するようにしてください。つまり、この処理は BODY 要素としてではなく、HEADER 要素として記述する必要があります。

## DB2 for i5/OS

DB2 で文字データをエンコードする方法は、複雑になることが考えられます。前述のように、変換が行われるかどうかは、DB2 for i5/OS 内でデータがエンコードされる方法、およびアプリケーションでどのような入力データ形式が必要かに依存します。

表 10-1 では DB2 for i5/OS でロシア語 ( キリル文字のアルファベット ) をエンコードする 3 種類の方法を示しています。

表 10-1 DB2 for i5/OS でロシア語にエンコードする方法

エンコード	SQL のデータ・タイプ	CCSID	説明
EBCDIC	CHAR/VARCHAR	1025	EBCDIC キリル文字複数言語
UTF-8	CHAR/VARCHAR	1208	ユニコード、ユニコード標準で UTF-8 と表記、普及している
ユニコード	GRAPHIC/VARGRAPHIC	13488	ユニコード標準で UTF-16 と表記

例 10-2 に掲載されている DDL を使用して、これらのデータ・タイプの列を含む単一の表を作成することも可能です。

例 10-2 i5/OS で 3 種類のロシア語文字エンコードを使用した表の作成

```
CREATE TABLE MYLIBRARY/CYRILLIC (
    EBCDIC CHAR ( 100) CCSID 1025 NOT NULL WITH DEFAULT,
    UTF8 CHAR ( 100) CCSID 1208 ,
    UNICODE GRAPHIC (100 ) CCSID 13488 NOT NULL WITH DEFAULT)
```

新しい表にデータを設定する 1 つの方法として、ロシア語がインストールされている Windows ワークステーションから、データベース機能のオペレーション・ナビゲーターを使用する方法が考えられます ( 「コントロールパネル」 → 「地域と言語のオプション」 → 「言語」 → 「詳細 ...」 → 「追加」 からアクセスします ) 。

DB2 for i5/OS から PHP アプリケーションにデータを抽出するには、主に 2 つの方法があります。詳細については、第 5 章『データベース・アクセス』(59 ページ) で DB2 と ODBC ドライバーの使用方法を参照してください。いずれのドライバーも表 10-1 で解説されているようにエンコードされたデータと正常に機能しました ( 本書作成時点 ) 。次のセクションに詳述されている構成ステップを実行した場合に限り、2 つのドライバーが正常に機能することが確認されています。

## 10.2 グローバリゼーションの構成 ( シングルバイト )

このセクションでは PHP で英語以外の言語を表示し、DB2 for i5/OS から英語以外の言語を使用したデータを抽出するために必要な構成について詳述します。これまでの例と同様にロシア語を使用して、おそらく最も広く使用されている UTF-8 を使ったロシア語のサポート方法について重点的に説明します。

**制限:** 現在、以下の対応が正しく動作するには、QSECOFR ユーザー・プロファイル (\*SECOFR 権限を持つプロファイルではなく、このプロファイルそのもの) でサインオンする必要があります。これは Apache のセキュリティー・コードの記述方法に関係しています。

### 10.2.1 i5/OS

前述のように、i5/OS ジョブを変更して、ロシア語に対応するよう構成します。

CHGJOB LANGID(RUS) CNTRYID(RU) CCSID(1025)

i5/OS 側で実行する必要があるのはこれだけです。正しく動作させるためには QSECOFR でサインオンする必要がある点に注意してください。

### 10.2.2 i5/OS PASE

UTF-8 サポートを構成するために i5/OS PASE 環境の設定の一部を手動で変更する必要があります。これは、新しい i5/OS PASE が開始したときにオペレーティング・システムで 사용되는デフォルト・マッピングが、UTF-8 ではなく ASCII であることが原因です。

i5/OS PASE 端末の開始:

CALL PGM(QP2TERM)

ASCII から UTF-8 に環境を切り替えるため、以下のコマンドを発行します。

```
export QIBM_PASE_CCSID=1208
export PASE_LANG=RU_RU
export LANG=RU_RU
```

必須ではありませんが、PASE\_NLSPATH と NLSPATH を変更することもできます。

**env** コマンドを実行することで、i5/OS PASE 上で変更が反映されているかどうかを確認できます。

### 10.2.3 PHP

管理コンソールを使用するか、ファイルを手動で直接編集する方法で、メインの PHP 構成ファイルを編集します。

ここでは「メモ帳」を使用して、以下のディレクトリーにある php.ini ファイルを編集します。

/usr/local/Zend/core/etc

ファイルでは、コメント化された `default_charset` ディレクティブが見つかります。この文字列をコピーし、先頭にあるコメント (セミコロン) を除去して、値を "UTF-8" に変更します。変更後は以下のようになります。

```
default_charset = "UTF-8"
```

## 10.2.4 Apache

Apache 環境での特別な設定は必要ありません。ただし、Zend から提供されたメニュー・オプション (GO ZENDCORE/ZCMENU) では i5/OS PASE 環境が正しく構成されないため、PHP 環境の開始および停止にこれらのメニュー・オプションを使用できません。

ただし、**apachectl** と呼ばれる使いやすいコマンドを利用できます。

最初に以下のコマンドを発行します。

```
cd /usr/local/Zend/apache2/bin
```

続いて、以下のコマンドを発行します。

<b>Start</b>	<b>apachectl -k start</b>
<b>Stop</b>	<b>apachectl -k stop</b>
<b>Restart</b>	<b>apachectl -k restart</b>

ここまでの操作で、PHP を希望どおりに構成できます。

**ヒント:** 以下のコードを PHP ファイルに組み込むことで、PHP エンジンが稼働している i5/OS PASE の環境設定を参照できます。

```
<?php
    phpinfo (INFO_ENVIRONMENT);
?>
```

## 10.2.5 DB2 for i5/OS

このセクションで概説した構成ステップのすべてを実行すると、EBCDIC、UTF-8、またはユニコードでエンコードされたテキスト・データを DB2 for i5/OS から抽出し、正しく表示できるようになります (表示先のブラウザで正しいフォントが使用できることを前提とします)。







## 高度な開発トピック

この章では、さらに高度なアプリケーション開発トピックについて説明します。開発を始めたばかりの PHP 開発者には少し高度かもしれませんが、スキル、経験、およびアプリケーションの複雑化への対応として役立つ可能性があります。

## 11.1 テンプレートのインプリメンテーション

テンプレートを使用すると、表示とアプリケーション・ロジックが分離されるため、PHP コードがよりクリーンになり、軽快なアプリケーションが実現します。これにより、アプリケーション・ロジックまたはアプリケーションのルック・アンド・フィールのいずれかを、他方に影響を与えることなく変更できます。ただし、テンプレートには欠点があります。以下の簡単な例で分かるように、テンプレート・ベースのソリューションには、テンプレートを使用しないソリューションよりも多くのコードが含まれています。テンプレートは、記述するコードを減らすためではなく、表示とアプリケーション・ロジックを分離しておくために使用します。

この例は、PHP を HTML テンプレートとともに使用することにより表示をアプリケーション・ロジックから分離する方法を示すことを目的としています。例 11-1 では、Template.tpl という名前のテンプレート・ファイルを定義します。

例 11-1 HTML Template ファイル

---

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title><?php echo $template->doctitle ?></title>
</head>
<body>
<?php echo $template->contentTitle; ?>
<?php echo $template->content; ?>
</body>
</html>
```

---

次のステップは、適切な関数を使用して Template と呼ばれるクラスを作成することから構成されます (例 11-2 を参照)。この例では、メイン・メソッド show(\$doc) は表示する文書の名前を引数として受け取ります。パス (この例では空) を設定してから、PHP 関数 include() を使用してテンプレートを実行します。

その後、関数 showFooter() を呼び出します。この関数は、現在日付と現行作業ディレクトリーを含む行を表示します。

例 11-2 クラス Template (TemplateClass.php) の定義

---

```
<?php
class Template {

    public $template_dir;

    function show($doc) {
        $template = $this;
        include($this->template_dir.$doc);
        $this->showFooter();
    }

    function showFooter() {
        printf("<br><br><br>Today's date:%s | Current directory:%s", date('d. M Y'), getcwd());
    }
}
?>
```

---

この後必要なものは、メイン文書のみです。これは、最初にテンプレート・クラス・ファイルを組み込みます。次に、クラス Template の新規インスタンスを作成し、クラス属性に値を割り当てます。最後に、ファイル内の設定に従い値の解析および置換を行って、関数 show('Template.tpl') でテンプレート・ファイルを表示します。

例 11-3 TemplateMain.php、メインファイル

---

```
<?php
    include('TemplateClass.php');

    $template = new Template;
    $template->doctitle = 'PHP using an HTML template';
    $template->template_dir = "";
    $template->contentTitle = '<h3>PHP using an HTML template</h3>';
    $template->content = 'Hello, World';
    $template->show('Template.tpl');
?>
```

---

例 11-4 は、テンプレートと i5/OS スプール・ファイルの取り出しを組み合わせた、さらに複雑な例を示しています。

例 11-4 テンプレートと i5/OS スプール・ファイルの取り出しの組み合わせ

---

```
$conn = i5_connect("127.0.0.1", "${USERNAME}", "${PASSWORD}");
if ($conn === false)
{
    //echo "FAIL :Failed to connect to server <br>";
    die(i5_errormsg());
}

$spool = i5_spool_list(/*array("username"=>"${USERNAME}", "outq"=>"${OUTQ}",
"userdata"=>"${USERDATA}"),$conn*/);
if ($spool === false)
{
    //echo "Fail:failed to create a spooled file list. <br>";
    die(errormsg());
}

$spool_file = i5_spool_list_read($spool);
if ($spool_file === false)
{
    // echo "Fail:failed to get spooled file data from the queue. <br>";
    die(errormsg());
}

$data = i5_spool_get_data($spool_file['SPLFNAME'], $spool_file['JOBNAME'],
                        $spool_file['USERNAME'], $spool_file['JOBNBR'],
                        "LAST" );

if ($data === false)
{
    // echo "Fail:failed to get the data from the spooled file. <br>";
    die(i5_errormsg());
}
// echo "<pre> $data </pre><br>";

$close_spool = i5_spool_list_close($spool);
if ($close_spool === false)
{
    //print ("FAIL :Failed to free spooled file list resource. <br>");
    i5_errormsg();
}

$close = i5_close($conn);
if ($close === false)
{
```

```
//echo "FAIL :Failed to disconnect from server :$i5_server_ip <br>";  
i5_errormsg();  
}  
${END}
```

---

テンプレートに関心があるが独自にソリューションを記述することは望まない場合は、多くの無償テンプレートを検索することが可能で、別のテンプレート・エンジンを記述することに時間を費やすよりもそれらのテンプレートをお勧めします。これらのいくつかは以下のPEARにあります。

- ▶ HTML\_Template\_Flexy ([http://pear.php.net/package/HTML\\_Template\\_Flexy](http://pear.php.net/package/HTML_Template_Flexy))
- ▶ HTML\_Template\_PHPLIB ([http://pear.php.net/package/HTML\\_Template\\_PHPLIB](http://pear.php.net/package/HTML_Template_PHPLIB))
- ▶ HTML\_Template\_IT ([http://pear.php.net/package/HTML\\_Template\\_IT](http://pear.php.net/package/HTML_Template_IT))
- ▶ HTML\_Template\_Sigma ([http://pear.php.net/package/HTML\\_Template\\_Sigma](http://pear.php.net/package/HTML_Template_Sigma))
- ▶ HTML\_Template\_Xipe ([http://pear.php.net/package/HTML\\_Template\\_Xipe](http://pear.php.net/package/HTML_Template_Xipe))

PEAR にリストされているものに加えて、以下の PHP テンプレート・ソリューションを参照することをお勧めします。

- ▶ Smarty (<http://smarty.php.net/>)
- ▶ TinyButStrong (<http://www.tinybutstrong.com/>)

コードの保守容易性およびポータビリティにさらに関心がある場合は、次のセクションを参照することをお勧めします。次のセクションでは、さらに高度な手法の説明があり、PHP プロジェクト構築時に推奨される設計実例についての理解の第一歩を踏み出すことができます。

## 11.2 Model View Controller (MVC) 設計およびフレームワーク

PHP により迅速なターンアラウンドが実現しますが、これがデメリットになることもあり、迅速に構築できるが保守不可能なアプリケーションにつながる可能性があります。設計において (特に大規模 PHP プロジェクトを構築する場合)、保守容易性、拡張性、および再使用可能性を求めて努力する必要があります。

MVC アーキテクチャーは、保守容易性、拡張性、および再使用可能性の向上に役立つ設計です。このアーキテクチャーは、アプリケーションを特定の役割を持つ 3 つの異なるコンポーネントに分割します。この技法は、次の 3 つの異なる考慮領域を確立することにより、任意の 2 つのコンポーネント間の従属関係を分離、または最小化します。

### ▶ ビュー

現行のアプリケーション状態の表示は、コントローラーから取り出されたデータを表示する小規模の PHP コードを含む HTML テンプレートによって処理されます。これらのページにはビジネス・ロジックはなく、コントローラーからのユーザー入力に基づいてデータ構造を取得するモデル・クラスへの呼び出しがあるだけです。コントローラーによって代行されるモデルまたは要求に存在する属性が、メッセージまたはフォームが表示されるかどうかを決定します。

### ▶ コントローラー

コントローラーは、アプリケーション (各表示ページを含む) への単一のエントリー・ポイントです。したがって、これは、動作およびそれぞれの PHP ビュー・ページに提供するデータを制御します。また、コントローラーは、ユーザー入力を処理対象のモデルに渡す前に妥当性検査およびクレンジングを行うなどの、セキュリティ上の予防措置もインプリメントします。

## ▶ モデル

モデルは、アプリケーションのビジネス・ロジックとオブジェクトから構成されます。これは、通常 PHP 5 クラスのコレクションとしてインプリメントされます。これは、データベース・インターフェースを呼び出す中間データ・アクセス・オブジェクトを使用します。データベース・インターフェースは、通信する実際のデータベースまたは PHP API にかかわらず、コマンドを変換しランタイム・インプリメンテーションに提供します。実際のデータベース・インターフェースとは関係なく、モデル・オブジェクトは、同じフォーマットでデータを受信し、データ層が変更しても変更する必要はありません。

実際には、これらの論理上の分離のそれぞれは、その他と少しオーバーラップします。例えば、多くの PHP アプリケーションでは、強力な MVC アプローチにもかかわらず、ビュー・ページにいくつかのコントローラー・コードがあり、表示ページにいくつかのモデル・クラス名が記述されています。

幸運なことに、将来の PHP アプリケーションの MVC 設計で役立つ MVC フレームワークが存在します。以下のような例があります。

- ▶ CakePHP (<http://www.cakephp.org/>)
- ▶ Agavi PHP MVC Framework (<http://www.agavi.org/>)
- ▶ Phrame (<http://www.phrame.org/>)
- ▶ Claw (<http://claw.tigris.org/>)
- ▶ The Ismo PHP Framework (<http://ismo.sourceforge.net/>)
- ▶ PHP on TRAX (<http://www.phpontrax.com/>)
- ▶ Symfony (<http://www.symfony-project.com/>)
- ▶ Livepipe: Pipeline Framework (<http://livepipe.net/pipeline/>)
- ▶ Zend Framework (<http://framework.zend.com>)

以下に、その他の PHP 用 MVC フレームワーク、およびそれらの比較表があります。

[http://www.phpwact.org/php/mvc\\_frameworks](http://www.phpwact.org/php/mvc_frameworks)

本書の作成時には、リストされた MVC フレームワークのほとんどは、データ・ソースとして IBM DB2 をサポートしていませんでしたが、MySQL または PostgreSQL はサポートしていました。IBM DB2 データベースを利用するためには、可能な場合は ODBC 接続を使用する必要があります。

つまり、PHP と Web アプリケーションは迅速にアプリケーションを構築する優れた方法として注目を集めるようになっていますが、柔軟で保守が容易なアプリケーションを構築するためには実績のあるソフトウェア開発原則と組み合わせる必要があります。

Web アプリケーションを設計する最良の方式について詳細が必要な場合は、PHP アプリケーション・アーキテクチャーの計画のための良いリソースとして、次の Sherri Wheeler の『Mature Design Theory in Web Development』があります。

<http://www.zend.com/php/design/mature-design.php>

Peter Eeles によって書かれた『What is a software architecture?』という、アプリケーション・アーキテクチャーについての全体的で理論的な説明が、以下の Rational® Edge にあります。

<http://www.ibm.com/developerworks/rational/library/feb06/eeles/>

## 11.3 ヒントおよび秘訣

以下の重要なヒントは、PHP コードの開発および管理に役立つ可能性があります。

### 11.3.1 事前定義配列を使用したシステム環境変数へのアクセス

デフォルトでは、Zend Core for i5/OS では `$_ENV` 配列または `$_HTTP_ENV_VARS` 配列を使用してシステム環境変数にアクセスすることはできません。示された配列にアクセスするためには、`php.ini` 内で `variable_order` オプションを構成する必要があります。または、**`getenv()`** PHP 関数を使用して特定環境変数の値を取得します。以下に、**`getenv()`** 関数に関する詳細情報があります。

<http://www.php.net/manual/en/function.getenv.php>

### 11.3.2 リバース・プロキシの問題

Zend Core for i5/OS はリバース・プロキシを使用してインプリメントされているため、`$_SERVER` (または `$_HTTP_*_VARS` に相当するもの) 内の環境変数の中には無意味になるものもあります。例えば、`SERVER_NAME` は `127.0.0.1`、`HTTP_HOST` は `127.0.0.1:8000` と表示されます。

元の有益な `SERVER_NAME` および `HTTP_HOST` (`www.example.org`) を保持するには、リバース・プロキシ・サーバーの構成 (デフォルトでは `/www/zendcore/conf/httpd.conf`) に次のディレクティブを追加します。

`ProxyPreserveHost On`

# 関連資料

このセクションにリストされた資料は、この Redbook で取り上げたトピックをさらに詳細に説明するために特に適していると考えられます。

## IBM Redbooks

これらの資料の注文方法については、『IBM Redbooks の入手方法』(122 ページ)を参照してください。ここで参照する資料の中にはソフトコピーでしか入手できないものもあることに注意してください。

- ▶ *Bringing PHP to Your IBM eSeries iSeries Server*, REDP-3639
- ▶ *Developing PHP Applications for IBM Data Servers*, SG24-7218
- ▶ *IBM HTTP Server (powered by Apache): An Integrated Solution for IBM eSeries iSeries Servers*, SG24-6716
- ▶ *IBM eServer iSeries IBM i5/OS セキュリティー・ガイド バージョン 5.4*, SG88-4015-01
- ▶ *Preparing for and Tuning the SQL Query Engine on DB2 for i5/OS*, SG24-6598
- ▶ *SQL Performance Diagnosis on IBM DB2 Universal Database for iSeries*, SG24-6654

## その他の資料

以下の資料も、その他の情報源として関係があります。

- ▶ *Beginning PHP and MySQL 5: From Novice to Professional, 2nd Edition*, W. Jason Gilmore, ISBN 1590595521
- ▶ *Learning PHP 5*, David Sklar, ISBN 0596005601
- ▶ *PHP Cookbook, 2nd Edition*, Adam Trachtenberg, et al, ISBN 0596101015

## オンライン・リソース

以下の Web サイトも、その他の情報源として関係があります。

- ▶ i5/OS 用の Zend サイト  
[http://www.zend.com/products/zend\\_core/zend\\_for\\_i5\\_os](http://www.zend.com/products/zend_core/zend_for_i5_os)
- ▶ Zend Core の IBM サイト  
<http://www-304.ibm.com/jct03002c/software/data/info/zendcore/>
- ▶ 次のサイトには、多くの有益な PHP リンクがあります。  
<http://www.ibm.com/developerworks/opensource/library/os-php-read/>
- ▶ J2EE を PHP と組み合わせて共通 Web アプリケーション・インフラストラクチャーをインプリメントする  
[http://www.ibm.com/developerworks/websphere/techjournal/0505\\_krook/0505\\_krook.html](http://www.ibm.com/developerworks/websphere/techjournal/0505_krook/0505_krook.html)

- ▶ System i 上での PHP の使用 : 概要、サンプル・アプリケーション、および 1 つのグループにおける経験  
<http://www.ibm.com/developerworks/systems/library/es-path2php/>
- ▶ DB2 for i5/OS - Porting Information  
<http://www.ibm.com/servers/enable/site/db2/porting.html>
- ▶ IBM Migration Toolkit  
<http://www.ibm.com/software/data/db2/migration/mtk/>

## IBM Redbooks の入手方法

次の Web サイトで、Redbook、Redpaper、ヒント、ドラフト資料、およびその他の資料の検索、参照、またはダウンロード、およびハードコピーの Redbook または CD-ROM の注文を行うことができます。

[ibm.com/redbooks](http://ibm.com/redbooks)

## IBM からのヘルプ

IBM サポートおよびダウンロード

[ibm.com/support](http://ibm.com/support)

IBM グローバル・サービス

[ibm.com/services](http://ibm.com/services)



# 略語および頭字語

<b>API</b>	アプリケーション・プログラミング・インターフェース (Application Programming Interface)	<b>MVC</b>	モデル・ビュー・コントローラー (Model View Controller)
<b>BDB</b>	BerkeleyDB	<b>OO</b>	オブジェクト指向 (Object Oriented)
<b>CCSID</b>	コード化文字セット ID (Coded Character Set ID)	<b>PASE</b>	Portable Application Solutions Environment
<b>CFGTCP</b>	TCP/IP の構成 (Configure TCP/IP)	<b>PEAR</b>	PHP Extension and Application Repository
<b>CLI</b>	コール・レベル・インターフェース (Call Level Interface)	<b>PECL</b>	PHP Extension Community Library
<b>CLR</b>	共通言語ランタイム (Common Language Runtime)	<b>RDBMS</b>	リレーショナル・データベース管理システム (Relational Database Management System)
<b>CMS</b>	コンテンツ・マネージメント・システム (Content Management System)	<b>RST</b>	オブジェクトの復元 (Restore Object)
<b>CPAN</b>	Comprehensive Perl Archive Network	<b>RSTLIB</b>	オブジェクトの復元 (RST) およびライブラリーの復元 (Restore Object (RST) and Restore Library)
<b>cURL</b>	クライアント URL (client URL)	<b>RSTLICPGM</b>	ライセンス・プログラムの復元 (Restore Licensed Program)
<b>DAS</b>	データ・アクセス・サービス (Data Access Service)	<b>RTVNETA</b>	ネットワーク属性検索 (Retrieve Network Attributes)
<b>DDL</b>	データ定義言語 (Data Definition Language)	<b>SAV</b>	オブジェクトの保管 (Save Object)
<b>DLTLICPGM</b>	ライセンス・プログラムの削除 (Delete Licensed Program)	<b>SAVLIB</b>	オブジェクトの保管 (SAV) およびライブラリーの保管 (Save Object (SAV) and Save Library)
<b>DML</b>	データ操作言語 (Data Manipulation Language)	<b>SDO</b>	サービス・データ・オブジェクト (Service Data Objects)
<b>DMZ</b>	非武装地帯 (Demilitarized Zone)	<b>SOA</b>	サービス指向アーキテクチャー (Service Oriented Architectures)
<b>DSPOBJAUT</b>	オブジェクト権限の表示 (Display Object Authority)	<b>SSL</b>	Secure Sockets Layer
<b>DSPPTF</b>	PTF の表示 (Display PTF)	<b>UDB</b>	Universal Database
<b>GRTOBJAUT</b>	オブジェクト権限の認可 (Grant Object Authority)	<b>V5R4</b>	バージョン 5 リリース 4 (Version 5 Release 4)
<b>GUI</b>	グラフィカル・ユーザー・インターフェース (Graphical User Interface)	<b>WRKACTJOB</b>	活動ジョブの処理 (Work with Active Jobs)
<b>IBM</b>	International Business Machines Corporation	<b>WRKAUT</b>	権限の処理 (Work with Authority)
<b>IDE</b>	統合開発環境 (Integrated Development Environment)	<b>WRKPTFGRP</b>	PTF グループの処理 (Work with PTF Groups)
<b>ISAM</b>	索引付き順次アクセス方式 (Index Sequential Access Method)	<b>WRKUSRJOB</b>	ユーザー・ジョブの処理 (Work with User Jobs)
<b>ISO</b>	国際標準化機構 (International Organization for Standardization)	<b>WRKUSRPRF</b>	ユーザー・プロファイルの処理 (Work with User Profiles)
<b>ISP</b>	インターネット・サービス・プロバイダー (Internet Service Provider)	<b>ZENDCOREAP</b>	Zend Core のジョブ (Zend Core jobs)
<b>ITSO</b>	International Technical Support Organization		
<b>LDAP</b>	Lightweight Directory Access Protocol		
<b>LICPGM</b>	ライセンス・プログラム (Licensed Program)		



# 索引

## 数字

32 ビット 60  
64 ビット 60

## あ

安定度 3

## え

永続情報 40

## お

オープン・ソース 71  
大文字 62

## か

拡張 3  
環境 60

## こ

コントローラー 119

## さ

サーバー・モード - ibm\_db2 62

## し

信頼性 3

## す

スキーマ 64, 72

## せ

セキュリティー 72  
接続 - ツールキット 51

## つ

ツールキット  
    接続 51  
    プログラム呼び出し 52  
    CL 呼び出し 51

## て

データベース・ソリューション 60

## ひ

非武装地帯 91

## ふ

ファイアウォール 99  
複数インスタンス 31

複数のインストール 31  
プログラミング言語 3  
プログラミング・インターフェース 60  
プログラム呼び出し - ツールキット 52

## ら

ライブラリー 64  
リバース・プロキシ 25, 91  
リレーショナル・データベース管理システム 60  
列名 62  
ロジック 40

## A

apachectl 35

## C

CL 呼び出し - ツールキット 51

## D

DB2 ファミリー 60  
DB2\_ATTR\_CASE 62  
DB2\_CASE\_LOWER 62  
db2\_connect 64  
DMZ 91  
DSPRDBDIRE 60

## H

httpd.conf 90

## I

i5\_command 51

## J

JavaScript 99

## M

Microsoft Internet Explorer 99  
Mozilla Firefox 99  
MVC 119  
mysql 76  
MySQL AB 71  
mysqli 76

## N

NETSTAT 100  
NOBODY ユーザー・プロファイル 70

## O

ODBC 60

## **P**

PDO\_ODBC 60, 69

Perl 8

Personal Home Page/Form Interpreter 8

PHP 5 8, 119

PHP コミュニティー 8

PHP Data Objects 69, 78

php.ini 90

PHP/FI 8

Practical Extraction and Reporting Language 8

## **Q**

QSQRVR 62, 70

## **R**

RDBMS 60

Redbooks Web サイト 122

RSS パーサー 54

## **S**

SimpleXML 拡張モジュール 54

SOAP 55, 56

SSL 90

## **V**

V5R3 4

V5R4 4

## **W**

WRKRDBDIRE 60

## **X**

XML 53

XML-RPC 55

## **Z**

z/OS 60

Zend エンジン 8

ZENDCORE 91





# PHP:Zend for i5/OS

インストールおよび  
管理方法の習得

有益な開発ヒントと  
アドバイスの発見

セキュリティ、グ  
ローバリゼーション、  
Zend Platform for  
i5/OS など

この IBM Redbook は、Zend Core for i5/OS および Zend Platform for i5/OS を使用した、System i 上の PHP のインストール、構成、および生産性の向上に役立ちます。PHP を評価している場合にも、本書は背景情報やその他のツールとの比較を提供しているため役立ちます。

以下のようなトピックが取り扱われています。

- ▶ インストール、構成、および管理
- ▶ アプリケーション開発
- ▶ PHP アプリケーションから DB2 for i5/OS および MySQL のデータにアクセスする方法
- ▶ セキュリティ、パフォーマンス、トラブルシューティングなど

本書では、すべてのプラットフォームに対応した一般的な PHP の機能よりも i5/OS 固有の関数および機能が強調されています。

## INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

実際の経験に基づいた  
テクニカル情報の構築

IBM Redbooks は、IBM International Technical Support Organization によって作成されています。IBM の専門家、世界中の顧客およびパートナーから、現実的なシナリオに基づくタイムリーなテクニカル情報が作成されます。ユーザーの環境でより効果的に IT ソリューションをインプリメントすることを助けるための、具体的な推奨事項が提供されています。

詳しくは、以下を参照  
してください。  
[ibm.com/redbooks](http://ibm.com/redbooks)

SG88-4025-00

