NTT DaTa

intra-mart

The Power for Enterprise Web Solution

ベースモジュール/フレームワーク

チュートリアルガイド

Ver. 5.0

株式会社 NTTデータ イントラマート

はじめに

当初、Webブラウザはハイパーテキストの参照システムとして世界的に急速な普及を遂げました。しかしこのWebシステムの運用管理の容易性、操作の簡便性などの利点は、最近の業務アプリケーションの構築スタイルとして一般的となったクライアントサーバー型システムの抱える諸問題（運用コスト増大、教育コストの増大等）を救う手段として注目をあびてきました。
そこでWebブラウザを利用したアプリケーション開発の試みが始まり、特にデータベースとの連携を中心とした各種技術の進展がありました。それにより、Webブラウザは単なるホームページの閲覧だけでなく、データベースに接続された業務アプリケーションのフロントエンドとしての利用が目標とされるようになってきました。
しかし、実際に業務アプリケーションとしての使用に耐えうるようなWebブラウザベースのイントラネットを開発しようとすると、これまでのクライアントサーバー型のシステム開発とはまったく異なるセキュリティ面やレスポンス面等のさまざまな障害や開発生産性の悪さに直面することになります。
さらにはECやEnterpriseレベルでのWebの活用には、24時間365日の稼動を前提とした、これまでよりも一層の信頼性のあるシステムアーキテクチャが必要となります。

intra-martの製品には、intra-martベースモジュール/フレームワークが用意されており、おのおの用途に応じて使い分けることができるようになっています。

intra-martベースモジュール/フレームワークでは、Webシステムの開発において必要になってくる各種機能を共通APIとしてモジュール化したフレームワークを用意しておりますので、それを利用することによりWebブラウザベースのシステム開発におけるさまざまな障害を意識することなく、簡単に乗り越えることができます。

これら両製品は、共にさらに親しみやすいページベースの開発（HTMLとサーバサイドJavaScriptを利用）と、J2EEベースの開発（JSP、Servlet、EJBを利用）という2つのWebアプリケーションモデルを混在させながら、本格的な信頼性のある多階層アーキテクチャに基づいたWeb業務アプリケーションを構築していくことが可能になります。
intra-martベースモジュール/フレームワークは、数多くの実績にもとづいた信頼性のあるアプリケーションサーバとして、あるいはフレームワークとして、お客様のWebシステムの基盤を構成することが可能な製品です。

本書では、intra-martベースモジュール/フレームワークの機能を説明させていただきます。

Contents

1 intra-martの全体構成 1

2 intra-martの特長 4

≪メリット1≫Webシステムの開発における各種機能をフレームワークとしてモジュール化 4
数多くのJavaコンポーネント群（im-BizAPI） 4
J2EE開発のフレームワーク（im-J2EE Framework） 5
ページベース開発も同時にサポート 6
≪メリット2≫ビジネスの変化に柔軟に対応できるSOAシステム統合基盤 8
≪メリット3≫高速で信頼性のあるWebアプリケーションサーバ機能も装備 8

3 im-BizAPI（Java業務コンポーネント群）概要 10

≪利用方法1≫J2EEベース開発モデルの場合 10
≪利用方法2≫ページベース開発モデルの場合 11
Java業務コンポーネント群（im-BizAPI） 11
ユーザインタフェース層 12
画面共通モジュール 12
帳票印刷モジュール 15
グラフ描画モジュール 15
ビジネスロジック層 16
アプリケーション共通モジュール 16
メール連携モジュール 17
外部ソフトウェア接続モジュール 18
ERP連携モジュール 18
業務基盤ツール 19
《機能UP!》アクセスセキュリティ・モジュール（[ログイングループ管理]-[ロール]フォルダ） 19
ワークフロー・モジュール （[ワークフロー]フォルダ）《機能up!》 23
ビジネスプロセスワークフローモジュール（アドバンスド版には標準装備） 25
バッチ管理モジュール （[ログイングループ管理]-[バッチ設定]メニュー） 27
ポータルモジュール （[ポータル]フォルダ、（[ユーザ設定]-[ポータル]フォルダ）《機能up!》 28
ViewCreator （[ViewCreator]フォルダ） 31
ユニット 32
アプリケーション共通マスタunit（[ログイングループ管理]-[アプリケーション共通マスタ]フォルダ）《機能up!》 32
カレンダunit （[ログイングループ管理]-[カレンダー設定]メニュー）《機能up!》 35
ファイルアップロードunit/ファイルダウンロードunit 35
ツリー表示unit （[ログイングループ管理]-[メニュー設定]メニュー） 36
iモードunit （[ユーザ設定]-[モバイルパスワード設定]メニュー） 36
アプリケーション共通マスタの履歴化《新機能!》 36
多言語対応 37
標準画面の作り方（共通画面デザイン） 37
エクステンション・モジュール 40
画面共通モジュール拡張 40
帳票印刷モジュール拡張 42
アクセスセキュリティ・モジュール拡張 43
ワークフロー・モジュール拡張 44
外部ソフトウェア連携ソリューション 48
統合検索ソリューション 48
マルチデバイスソリューション 49

4 intra-martのアプリケーション開発概要 50

ページベース開発モデルによるアプリケーション開発 52
プレゼンテーションページ 52
ファンクションコンテナ 53
J2EEベース開発モデルによるアプリケーション開発 53
J2EE開発のフレームワーク 54
Strutsの利用 56
J2EE開発モデルにおいてフレームワークを活用した際のメリット 57
intra-martのアプリケーションの動作概要 58

5 intra-martのシステムアーキテクチャー 59

intra-martベースモジュールのシステムアーキテクチャー 59
intra-martフレームワークのシステムアーキテクチャー 61

6 intra-martのその他の特徴 63

軽快なスピード環境 63
高度な信頼性 64
高いスケーラビリティ 66
運用の容易性 67
IM-Administrator 68
ASP型アプリケーションの構築 69
多言語機能を標準で提供 69
プラットフォームのオープン性 69

7 システム仕様 70

パッケージ別機能一覧 70
対応規格一覧 71
システム要件 71
サーバ要件 71
環境要件 72
クライアント要件 73
その他の要件 73
検証済み環境 73
注意事項 76

8 intra-mart「eBuilder Ver5.0」 77

豊富な開発支援機能を搭載 PageProducer 77
Eclipse(J2EE開発環境)へのプラグインも可能FrameworkProducer 78
FrameworkProducerのソースコード自動生成、ビジュアルな画面定義機能 78

9 旧バージョンとの互換性(Ver4.x→Ver5.x) 80

APIの互換性 80
データベース構造の互換性 80
データベースアクセス 80

10 Standard版とAdvance版の相違 81

Standard版 81

Advance版 81
Stabdard版とAdvance版の機能比較 81
（参考）OEMプロダクツ一覧 82

1 intra-martの全体構成

intra-martでは企業のEnterprise Web Solutionを実現するために、各種のソリューションを提供しています。

intra-martベースモジュール（フレームワークベースの次世代型Webアプリケーションサーバ）

intra-martベースモジュールは、フレームワークを利用することで、複雑なWebアプリケーションを簡単に短期間で開発することができます。フレームワークには、さまざまなJava業務コンポーネント（アクセスセキュリティ・モジュール/ワークフロー・モジュールなどを総称して「im-BizAPI」と呼びます）とJ2EEフレームワーク（im-J2EE Framework）から構成されています。さらに、J2EEベースの開発と併用して、ページベース（スクリプトベース）の開発モデルも利用できるため、Webシステム開発の敷居を大きく下げることができます。また、フェールオーバー機能やラウンドロビンによるロードバランス機能などの高度な信頼性を提供するJ2EEアプリケーションサーバの機能（intramart Application Server(Resin)）も標準搭載されています。

intra-martフレームワーク

intra-martベースモジュールのフレームワークを他社アプリケーションサーバ（IBM WebSphere、BEA WebLogicなど）でもご利用になれるフレームワークだけの製品です。J2EEアプリケーションサーバ上でのWebシステム開発の生産性を大きく向上させることができます。

intra-martアプリケーション

intra-martアプリケーションは、強力なフレームワークにより構築された統合型Web業務アプリケーションシリーズです。すべてのソースコードが公開されており、ユーザは自由にカスタマイズして機能を拡張していくことができます。さらに、各アプリケーションの会社/組織情報、顧客情報など基本的な情報が連動して動作します。これらのアプリケーションでは、既存の基幹システムと連携した全社規模の情報共有を実現することで、経営課題を短期間で達成していくことができます。
各アプリケーションのシリーズ製品は以下の3つのソリューションに分類されます。

イントラネット・ソリューション	M&A、アウトソーシングニーズの高まりにより増大する「間接部門のコスト低減」と「全社規模での情報共有（コラボレーション）」を実現するWebイントラネット構築ソリューション。全社規模のイントラネットを実現します。
イントラネット・スタートパック	掲示板、スケジュール管理、ドキュメント管理、施設予約管理、旅費精算、勤怠管理、FAQシステム、備品発注、住所録、伝言メモ、営業日報管理などで構成されるエンタープライズ向けグループウェア（iモード対応）です。
人事管理システム	グループ企業を含めてイントラネットで人材情報の共有と有効活用が可能です。また各種人事関連申請がワークフローでおこなえます。
文書管理システム	Webベースのワークフロー（電子承認）機能が確実な文書管理をサポートします。文書の起案から申請・承認・配布・廃棄にいたるライフサイクルを一元管理できます。
WebMail	移動先からWebブラウザでメール内容の確認・送信ができます。Iモードにも対応しています。
プロジェクトリソース管理システム	進捗管理をはじめとするプロジェクト管理、予算実績対比のコスト管理などプロジェクトに関するあらゆるリソースをWebで管理できます。
CRMソリューション	折衝情報、サポート情報、クレーム情報などの顧客情報の全社共有は、企業の最重要テーマとなります。CRMソリューションは、顧客情報を中心としたエンタープライズポータルを実現します。
営業支援システム	販売計画の立案から営業日報入力、予実績把握など、営業活動全般にわたるシステム化が可能です。顧客情報の一元化により、SFA/CRMが段階的に導入可能です。
販売管理システム	受注から請求または在庫管理などの販売業務に関わるすべての機能を標準装備しています。販売関連業務のWeb化が実現することで、これらの機能を社内外に広く公開して、業務効率を大幅に向上させることが可能です。
BtoBソリューション	自社のシステムとパートナー会社との間をWebにより、情報連動/共有を可能にします。また、取引先などパートナーのためのエンタープライズポータルを実現します。
Web受発注システム for SellSide	BtoB ECにおけるセルサイド企業（販売会社）側のシステムを構築するためのパッケージソリューションです。
Web受発注システム for BuySide	BtoB向け電子調達システム構築パッケージソリューションでeプロキュアメントを実現します。

intra-martでは、イントラネット、CRM、BtoB、ECへと至るお客様のe-businessへの取り組みを、これら各種のWebアプリケーション製品により、力強くトータルにバックアップしていきます。またこれらのアプリケーションシリーズはすべてソースコードが公開されているため、ソースコードに対してユーザ自身で簡単にカスタマイズできるなどの特徴があります。このマニュアルでは、intra-mart製品の中核をなす基盤ツールであるintra-martベースモジュール/フレームワークについて説明します。その他、intra-martアプリケーションの詳細については、それぞれのマニュアル等を参照してください。

2 intra-martの特長

Webベースの業務アプリケーションをintra-martベースモジュールあるいはintra-martフレームワークを利用して開発することにより、次のようなメリットが得られます。

《メリット1》Webシステムの開発における各種機能をフレームワークとしてモジュール化

数多くのJavaコンポーネント群（im-BizAPI）

J2EE対応Webアプリケーションサーバによるシステム開発では、一般的にJavaClassやEJB（Enterprise Java Beans）コンポーネントなどのソフトウェア部品を活用しながら開発生産性を高めていくことができる特徴があります。しかし、このソフトウェア部品の開発には特殊なスキルが必要とされ、依然として「しきい」が高いのが現状です。

intra-martベースモジュール/フレームワークでは、Webシステム構築でよく利用される機能を「Javaコンポーネント」として数多く提供しており、JavaClassまたはEJBコンポーネントとして再利用が可能です（アクセスセキュリティ・モジュールやワークフロー・モジュール、ポータルモジュールなど。これらを総称して「im-BizAPI」と呼んでいます）。

これにより、従来ゼロから開発したり、他社製品を組み合わせないと実現できなかった複雑な機能をフレームワークに用意されているJava業務コンポーネント群を利用することで、大規模なWebシステムを短期間かつ高品質に構築できます。

※「intra-martベースモジュール」には、intra-mart Application Serverとintra-martフレームワークが同梱されています。
「intra-martフレームワーク」は、他社アプリケーションサーバ上でご利用いただけるフレームワークのみの製品です。

J2EE開発のフレームワーク(im-J2EE Framework)

J2EEベース開発モデル*の生産性を向上させるため、J2EE開発で必要になる共通的な処理はすべてフレームワーク(im-J2EE Framework)として用意し、開発者に委ねられる箇所はコンポーネントを作成してもらう形態となります。
特に、このフレームワークを利用するコンポーネントについては、雛型となる"スケルトン"を用意しておくことで、さらに生産性を高めることが可能となります(共通部分はあらかじめ実装されているので、開発者は独自のコンポーネントを作成する場合、スケルトンをコピーし、実装されていない箇所を埋めてコンポーネントを作成することになります)。作成されたコンポーネントはプロパティ設定によりフレームワークにあてはめていくことで動作します。

スケルトンを利用したコンポーネント開発のため、J2EE開発の前提知識がなくても完成したシステムはJ2EEアーキテクチャの推奨型となり、MVCモデルの実現が容易になります(プログラム構造が統一できるため、コンポーネントの再利用性が高まり生産性および保守性が向上します)。また、コンポーネント群とプロパティファイルの設定をベースとした構造になるため、柔軟なカスタマイズや機能追加が可能なシステムとなります。

■J2EEベース開発モデル

J2EE(Java2 EnterpriseEdition)とは米サンマイクロシステムズが1999年9月に発表したプラットフォームで、Servlet、JSP(JavaServerPage)、EJB(EnterpriseJavaBeans)などから構成され、MVCモデル(Model-View-Controller)でシステム構築をおこなっていきます。特に高トランザクションが集中する処理に適し、さらに業務コンポーネントの再利用が可能になるなどのメリットがあります。また、ベースモジュールのスタンダード版ではJSPおよびServletのサポートであり、ベースモジュールアドバンスド版ではJSPおよびServletとさらにEJBまでサポートしています。

ページベース開発も同時にサポート

intra-martベースモジュール/フレームワークでは、世界で初めて2つのWebアプリケーションモデルを同時にサポートしています。「J2EE開発モデル」と「ページベース開発モデル*」という2つの開発モデルは、それぞれ特徴を持ち、用途に応じて使い分けることができます。また、2つのモデルで開発したWebアプリケーションを同一のシステム内に混在させて実行することもできます。

どちらの開発モデルもアクセスセキュリティ・モジュール、ワークフロー・モジュールなど、intra-martが標準で用意しているJavaコンポーネント群（im-BizAPI）を利用することができますので、高い生産性を誇ります。そのため、開発予算が限られ短期間開発の必要なWebシステムはページベースの開発モデルを主体に開発を進め、その中で異なるプロジェクトでも作成したコンポーネントの再利用が可能な部分を切り出してJ2EEベースで開発するといった、現実的かつ柔軟な開発スタイルをとることができます。この方式により、J2EEベースの開発モデルでの問題（クラス設計などの高度なスキルの必要性やJava技術者の確保などの面から敷居が高く開発費用が増大する）を解決することが可能となります。

■ページベース開発モデル

ColdFusionやMicrosoft ActiveServerPageに代表されるようなスクリプトを主体にした開発モデルです。intra-martでは、普段ホームページの作成に利用しているHTMLと（サーバサイド）JavaScriptを利用してデータベースと連動したWebシステムの開発が可能な作りにしています。そのため初心者でも始められる簡易性により要員のシステム教育コストを大きく削減することができます。これまで簡単なホームページしか作成したことがなかった方でも約2週間～1ヶ月ほどでマスタしてWeb業務画面を作成しています。

また、ホームページの作成更新と同様に簡易に開発ができますので、頻繁に更新されるような複雑なWebシステムにも柔軟に対応できるメリットがあります（さらに、JavaScriptからは、開発者のテクニカルスキルにあわせて、Java（Class、EJB）、C++、ストアドプロシージャなどが簡単に呼び出せます）。

Column

intra-martベースモジュールとintra-martフレームワークの違い

intra-martベースモジュールに内包されているフレームワークは他社J2EEアプリケーションサーバ上でも同様に利用でき、開発生産性を向上できます。intra-martベースモジュールには、このフレームワークとintra-mart Application Server(Resin)が含まれています。

Column

開発支援ツール「eBuilder Ver5.0」

intra-mart「eBuilder Ver5.0」(別売)を活用することにより、ユーザアプリケーションを効率よく開発していくことができます。intra-mart「eBuilder Ver5.0」はページベース開発モデルおよびJ2EEベース開発モデルでご利用になれます。詳細は、本書「8 intra-mart eBuilder Ver5.0」を参照してください。

＊ intra-martベースモジュール/フレームワークには、intra-mart「eBuilder Ver5.0」の試用版が同梱されています。ぜひ、一度体験することをお勧めいたします。

≪メリット2≫ビジネスの変化に柔軟に対応できるSOAシステム統合基盤

M&Aや分社化、ビジネスモデルの変更など、ビジネスの変化に対して、システムも柔軟に対応できるような「変化に強いシステム」の重要性が高まっています。この考えがないと、ビジネス戦略にとって、情報システムが足かせとなってしまいます。intra-martではシステム統合基盤として、フレームワークに加え、アプリケーション共通マスタ、ポータルモジュール、シングルサインオン、ビジネスプロセスワークフロー（BPW）などの各種システム統合のための機能を用意しています。
これにより、ERPや汎用機などの基幹システムと密接に連携したWebフロントシステムの構築が既存システムを活かしながら短期間で構築できます。また、ビジネスプロセス、システム機能、DBが独立した構造となるため、業務プロセスが変わっても業務システムの対応が短期間でできます。

≪メリット3≫高速で信頼性のあるWebアプリケーションサーバ機能も装備

intra-martのフレームワーク機能は、さまざまなアプリケーションサーバに対応していますが、intra-martベースモジュールは、高速で信頼性の高いアプリケーションサーバ機能（intra-martアプリケーションサーバ）も標準装備しています。そのためWebサーバ上で実行されるビジネスロジックをWebアプリケーションサーバ上で集中的に稼動し、WebサーバやDBサーバと分割した構成にすることで負荷の分散を図ることも可能です（多階層アーキテクチャへの対応）。

* Webサーバは、intra-martベースモジュールに内包されているHTTPサーバをそのまま利用することも可能です。

また、intra-martベースモジュールでは、製品に内包されているintra-martアプリケーションサーバを使用すると「ラウンドロビン機能（*1）」を利用することができます。ラウンドロビン機能は、複数のアプリケーションサーバ間でも処理の負荷分散を図ることができるため（単一のIPアドレスで）、クライアント数の増加にも柔軟に対応できるとともに、セッションフェイルオーバー（下記注*2）などの障害対策にも優れたシステムを構築することができます。

*1：ラウンドロビン機能とは、複数台のWebサーバをクラスタ化し、負荷を複数サーバに分散させる機能です。
*2：セッションフェイルオーバー機能とは、複数台のWebサーバをクラスタ化し、その中のサーバが 何らかの障害（電源故障、ネットワーク障害等）でダウンするなど、サーバ機能を果たさない場合、そのサーバへのアクセスを他のサーバに振り替える機能です（セッション情報も他のサーバへ引き継ぐことができますので、完全にノンストップのシステムとなります）。

また、intra-martベースモジュール/フレームワークは、HTMLやJava、JavaScriptなど、インターネット業界標準のテクノロジーのみで構築されているため、ひとつのメーカに拘束されない柔軟性の高いイントラネット構築を実現します。これらはシステムの将来性があるだけでなく、他社システムまでも包含したBtoBエクストラネットシステムを構築する際や、消費者と連動するECシステムを構築する際に、特に重要なシステム要素となってきます。

3 im-BizAPI（Java業務コンポーネント群）概要

intra-martベースモジュール/フレームワークに内包されているフレームワークでは、Webシステム構築でよく利用される機能を「Javaコンポーネント」として数多く提供しており、JavaClassまたはEJBコンポーネントとして再利用が可能です（アクセスセキュリティ・モジュールやワークフロー・モジュール、ポータルモジュールなど。これらを総称して「im-BizAPI」と呼んでいます）。これにより従来ゼロから開発したり、他社製品を組み合わせないと実現できない複雑な機能を、用意されているJava業務コンポーネント群を組み合わせることで、大規模なWebシステムを短期間かつ高品質に構築できます。
このim-BizAPIは、2つのアプリケーション開発モデル（ページベース開発モデル、J2EEベース開発モデル）から同様に呼び出して活用することができます。

≪利用方法1≫J2EEベース開発モデルの場合

intra-martベースモジュール/フレームワークに内包されているフレームワーク（下図のフレームワーク部分）には、「Java業務コンポーネント群（im-BizAPI）」だけでなく、「J2EEフレームワーク（im-J2EE Framework）」もあわせて用意されています。J2EE開発で必要になる共通的な処理はすべて「J2EEフレームワーク（im-J2EE Framework）」内に隠蔽化されて実装されているため、J2EEによるWebシステムの開発生産性を大きく向上させることができます（このJ2EEフレームワーク（im-J2EE Framework）を利用せずに、Servlet/EJBやJSPなどからダイレクトにim-BizAPIを呼び出して利用することも可能です）。

特に、このJ2EEフレームワーク（im-J2EE Framework）を利用するコンポーネント開発については、雛型となる「スケルトン」を利用することで、さらに生産性を高めることが可能となります（共通部分はあらかじめ実装されているので、開発者は独自のコンポーネントを作成する場合、スケルトンをコピーし、実装されていない箇所を埋めてコンポーネントを作成することになります）。作成されたコンポーネントはプロパティ設定によりJ2EEフレームワーク（im-J2EE Framework）にあてはめていくことで動作します。

- im-J2EE Frameworkの中のサービスフレームワークの代わりにオープンソースフレームワークとして、Jakartaプロジェクトから提供されている「Struts」を利用することもできます。

≪利用方法2≫ページベース開発モデルの場合

ページベース開発モデルにおいては、あらかじめ再利用可能な“モジュール”として用意されているJava業務コンポーネント（im-BizAPI）をプレゼンテーションページ（ユーザインタフェースを担うHTMLファイル）やファンクションコンテナ（業務ロジックを記述するJavaScriptファイル）から呼び出して利用することができます。

Java業務コンポーネント群（im-BizAPI）

Java業務コンポーネント群（im-BizAPI）に用意されている“モジュール”は、それぞれ利用する形態が異なりますので、以下のように分類されています。各モジュールの詳細内容は「intra-martベースモジュールAPIリスト」を参照してください。また、プログラミングガイドJ2EEベース開発モデル編　第4章「さまざまなJavaコンポーネント群（im-BizAPI）の利用」、ページベース開発モデル編第3章「さまざまなコンポーネント群（im-BizAPI）の利用」も参照してください。また、ほとんどのモジュール群のソースコードは公開されているため、モジュール自体の機能をカスタマイズして拡張することが可能です。

*1 ：アドバンスド版のみの機能です

ユーザインタフェース層

ページベース開発モデルではプレゼンテーションページで利用される以下のモジュール群は、すべてHTML文書の中で「〈IMART〉タグ」で呼び出すことができます。J2EEベース開発モデルではJSPファイルの中で「intra-martタグライブラリ」を利用します。intra-martタグライブラリの詳細については、APIリストの「J2EEベース開発モデル」の「画面共通モジュール」を参照してください。

画面共通モジュール

WebベースでのGUI開発でよく利用される画面部品のモジュールです。それぞれのモジュールに適当なプロパティを設定して呼び出すだけで、データベースと連動したユーザインタフェースを簡単に作成できます。また、J2EEベース開発モデルには、JSPで利用できるintra-martタグライブラリを使用しています。タグライブラリの詳細は、コラム「ユーザインタフェース層の呼び出し13）」を参照してください。
ここで紹介した標準で用意されている画面共通モジュールのほかに、クライアントGUI環境を実現するエクステンション・モジュール「IM-Curl（別売）」も用意されています。

クライアントGUI環境を実現する画面共通モジュール：IM-Curl

高度かつ複雑なユーザインターフェイスを構築可能なリッチクライアントサイドWebプログラミング言語です。IM-Curlの詳細資料に関しては、下記注を参照してください。

- IM-Curlの詳細に関しては以下の資料を参照してください。
  - 製品に添付されている、Curlアプレットが相互に通信するアプリケーションを作成するために用意されているCurl連携用APIを利用するためのチュートリアルガイド
    「IM-Curl連携APIチュートリアルガイド」（im_curl_api_tutorial.pdf）
  - APIリストの「J2EEベース開発モデル」-「ビジネスロジック層」-「アプリケーション共通モジュール」内のjp.co.intra_mart.foundation.im_curlクラス

Column

ユーザインタフェース層の呼び出し

● J2EEベース開発モデルの場合

JSPで呼び出すには、intra-martタグライブラリを使用します。詳細は、APIリストのJ2EEベース開発モデルの「画面共通モジュール」を参照してください。

使用例

〈imart:imartDateFormat value="〈%= date%〉" format="yyyy/MM/dd"/〉

指定された日付データを文字列として挿入する。

・タグライブラリリスト

calendar	カレンダモジュールの画面HTMLソースを作成する
condition	タグに挟まれた部分の実行の条件分岐（真偽）
dicision	タグに挟まれた部分の実行の条件分岐（値指定）
imartDateFormat	タグの指定された場所に、指定された日付データを文字列として挿入
imartEvent	ブラウザ上で動作するイベント処理ロジックを設定
imartNumberFormat	タグの指定された場所に、指定された数値データを文字列として挿入
loop	処理の繰り返し
repeat	タグ内のネストされた部分の繰り返し処理
select	コンボボックス（リストボックス）の作成
drawer	画像ファイルを作成しHTMLに表示する

〈スプレッドシート〉

CellDataSelector　スプレッドシートのセルの選択

（サンプル）

```
〈imtag:ColumnDataSelector bgColor="white" fgColor="red"〉
…
〈imtag:SelectionSubmitter formName="myform"
                    application="taglib" service="testform" /〉
〈/imtag:ColumnDataSelector〉
```

選択時のbgColorは”white”にする、fgColorは”red”にする。
下のサブミットボタンをクリックすることで、
選択されたセルの内容がサーバへ送信される。

ID	ローマ字	漢字
001	Tokyo	東京
002	Kyoto	京都
003	Osaka	大阪
004	Koube	神戸
005	Nagoya	名古屋

送信

ColumnDataSelector	スプレッドシートの列の選択
RowDataSelector	スプレッドシートの行の選択
SelectionSubmitter	カーソル位置のスプレッドシートの行をハイライトにする
LineHighLight	カーソル位置のスプレッドシートの行をハイライトにする
Sorting	スプレッドシートの任意列で並べ替える
DynamicTable	データベースデータの表作成
DBListPaging	ベースモジュール（またはフレームワーク）のデータベース設定（またはマルチデータベース設定）で設定されたデータベースを参照し、データを読み込み、動的にテーブルの作成をします。

（サンプル）
DBListPaging/ColumnとColumnSelectorタグのネストサンプル

```
〈imtag:ColumnDataSelector bgColor="red" fgColor="white"〉
〈imtag:DBListPaging tableID="unit_table" tableName="b_m_unit" styleClass="test"
    pageSize="10" indexLocation="both" indexSize="10"〉
    〈imtag:Column attribute="unit_cd" title="Code" headerClass="header" styleClass="test"/〉
    〈imtag:Column attribute="unit_name" title="Name" headerClass="header"/〉
    〈imtag:Column attribute="notes" title="Note" headerClass="header"/〉
〈/imtag:DBListPaging〉
〈/imtag:ColumnDataSelector〉
```

〈入力制御〉

Validate	日付、英数字、数字などの入力制御
DblClickForbidden	ボタンの制御
BackForbidden	ブラウザの「戻る」ボタンの無効化

〈メニュ〉

PopMenu	ポップアップメニュー制御

〈メッセージプッシュ〉

Push	送信側で入力されたメッセージをクライアント側にプッシュする部品です。受信側のJSPで、このタグを利用することで、画面をリロードしなくても、動的に送信側のメッセージを取得し表示することができます。

● ページベース開発モデルの場合

〈IMART〉タグのtype句により、呼び出すベースモジュールオブジェクトを指定できます。詳細は、APIリストのページベース開発モデルの「画面共通モジュール」を参照してください。

使用例

〈IMART type="select"〉	：データベースの内容をコンボボックスに表示する。
〈IMART type="calendar"〉	：指定月のカレンダーを表示する

帳票印刷モジュール

これまで、帳票の印刷はイントラネットのシステム開発で一番困難と言われていました。intra-martには、2種類のエクステンション・モジュール（別売）が用意されており、印刷する帳票に合わせて選択することができます。

PDF形式の帳票印刷モジュール：IM-PDFデザイナー	PDFを採用することにより、より複雑な帳票形式への対応が可能となる、オプションのエクステンション・モジュールです。
大量帳票印刷モジュール：IM-XRFコンポーザ	数100～数1000ページにもおよぶ大量の帳票印刷をWebブラウザで処理することが可能となるオプションのエクステンション・モジュールです。

- 上記エクステンション・モジュールの詳細に関しては、以降の「エクステンション・モジュール」の項、プログラミングガイドJ2EEベース開発モデル第3章「さまざまなJavaコンポーネント群（im-BizAPI）の利用」の「9エクステンション・モジュールの組み込みと操作」、ページベース開発モデル編第3章「さまざまなコンポーネント（im-BizAPI）の利用」の「15　エクステンション・モジュールの組み込みと操作」も参照してください。

グラフ描画モジュール

グラフ描画モジュールを利用することで数値データをグラフ化してHTMLに表示することができます。グラフ描画モジュールには、次の5種類のグラフが用意されています。ページベース、J2EEベースの両方の開発モデルに対応しています。

〈利用できるグラフの種類〉

折れ線グラフ
棒グラフ
円グラフ
レーダーチャート
ポートフォリオ

〈グラフの表示例〉

ビジネスロジック層

ビジネスロジック層には、「アプリケーション共通モジュール」、「メール連携モジュール」、「外部ソフトウェア接続モジュール」、「ERP連携モジュール（アドバンスド版のみ）」の4つのモジュールが用意されています。これらのモジュールは、ページベース開発モデルではファンクションコンテナからオブジェクトの形式で呼び出すことができ、J2EEベース開発モデルでは、ServletからJavaClassやEJBコンポーネントとして利用することができます。

アプリケーション共通モジュール

アプリケーション開発に必要な処理ロジックのモジュールが、使いやすくオブジェクト化されています。これらオブジェクトをビジネスロジックに組み込んで編集していくことで、ロジックの作り込みをせずに多階層アーキテクチャに基づいたWebシステムの開発を短期間で行うことが可能になります。

■提供されるアプリケーション共通モジュールのオブジェクト概要

アプリケーション開発に必要な処理ロジックのモジュール（セッション管理やDBアクセスなど）が使いやすくオブジェクト化されて提供されています。このオブジェクトの利用により、ページをまたいだセッション管理を実現することができます。またフッター等に利用する会社名称や各種データベースへのログインユーザ名等の情報も、このオブジェクトから利用できます。これらのオブジェクトをアプリケーションロジックに組み込んで編集していくことで、余計なロジックの作りこみをせずに複雑なWebシステムの開発を短期間で行うことが可能になります。

その他、アプリケーションの環境変数を含めた各設定値へのアクセス用オブジェクト、区分コードへのアクセス用オブジェクト、データベース関連の汎用オブジェクト、日付関連オブジェクト、デバック関連オブジェクト、URL管理オブジェクトなど多数用意されています。これらにより、現在接続している社員コードや、直前に表示したHTMLページ名などさまざまな情報にアクセスできます。

また、複数DBへの同時アクセスメソッドや大量の検索データに対して、指定した件数ごとに結果を画面表示する機能の検索ストリーミング、アプリケーションロック機能、XML対応モジュールなど、高度な機能が含まれています。

- アプリケーション共通モジュールの詳細については、APIリストの「ページベース開発モデル」または「J2EEベース開発モデル」の「アプリケーション共通モジュール」を参照してください。
- J2EEベース開発モデルのアプリケーション共通モジュールの例

jp.co.intra_mart.foundation.bpw	ワークフロー・モジュールの各種操作API群
jp.co.intra_mart.foundation.acssecurity	アクセスセキュリティ・モジュールを利用するためのクラスを提供
jp.co.intra_mart.foundation.appcomn	アプリケーション共通モジュール
jp.co.intra_mart.foundation.database	データベースへアクセスするためのクラスを提供
jp.co.intra_mart.foundation.http	ブラウザとデータの送受信に利用すると便利なユーティリティクラスを提供
jp.co.intra_mart.foundation.im_curl	リッチクライアントを構築するためのエクステンションモジュールIM-Curl
jp.co.intra_mart.foundation.page	ブラウザに表示する画面の作成に利用すると便利なユーティリティクラスを提供
jp.co.intra_mart.foundation.scenario	ページベース開発モデルのプログラムと連携するAPI
jp.co.intra_mart.foundation.service.client	intra-martの各サーバを利用するための共通的機能を提供
jp.co.intra_mart.foundation.service.client.batch	Schedule Serviceを利用するためのクラスを提供
jp.co.intra_mart.foundation.service.client.information	各種Serviceを利用するためのクラスを提供
jp.co.intra_mart.foundation.smtp	メール連携モジュール
jp.co.intra_mart.foundation.utility	アプリケーションを構築する際に利用すると便利な汎用クラスを提供
jp.co.intra_mart.foundation.utility.net	ソケットによるネットワーク対応機能を実装するためのクラスを提供

メール連携モジュール

このモジュールを利用することで、他のSMTP互換のメールサーバに対するメール送信処理を行うなど、メールに連携したアプリケーションを簡単に構築することができます。たとえば掲示板にある情報を日付指定で指定ユーザに一斉にメール配信するエージェントを作成したり、売上情報を定期的に社員にメール配信するようなアプリケーションを構築したりすることができます。

〈売上情報などをメールで自動配信することができます〉

外部ソフトウェア接続モジュール

市販のアプリケーションパッケージからim-BizAPIの各種APIを呼び出して直接利用するなど、intra-martと外部ソフトウェアを簡単に連携・接続できるモジュールです。連携・接続する方法には、次の2通りの方法が用意されています。
ひとつは、この「連携APIコネクタ」がフレームワークの中のim-BizAPIと連携するためのJavaベースAPIとして提供されているので、外部ソフトウェアがJava実行環境であれば任意のプロセスとim-BizAPIを連携させる方法です。例えば、市販のポータルサーバ製品と組み合わせて、ポータル画面中にintra-martの画面を表示したり、他のアプリケーションと連携してバッチ動作する独自のJavaプロセスからユーザアカウント情報を操作したりすることが可能です。

2つ目の方法としては、外部ソフトウェアからWebサービスによりim-BizAPIの各種APIを呼び出すことも可能です。標準的なWebサービスの手続きで、外部システムなどから自由に接続できます。

- 外部ソフトウェア接続モジュールの詳細に関しましては、APIリストの、「Developer's Guide」「J2EEベース開発モデル」の「ビジネスロジック」の「外部ソフトウェア接続モジュール」を参照してください。

ERP連携モジュール

SAP JCo技術を利用し、SAP APIをライブラリ化しました。標準のJava技術を用いて、アドオン開発を低コストで行えます。※intra-martベースモジュールアドバンスド版のみの機能です。

- 現在提供しているERP連携モジュールには、SAPとの連携ができる豊富なAPIが用意されています。その他のERP用の連携モジュールも順次追加して行く予定です。
- 詳細に関しては以下の資料を参照してください。
  - 製品に添付されている、SAP R/3とリアルタイムに連携するためのチュートリアルガイド「ERP連携モジュール・チュートリアルガイド」（im_sap_api_tutorial.pdf）
  - APIリストの「J2EEベース開発モデル」-「ビジネスロジック層」-「ERP連携モジュール」
- SAP、SAP R/3、SAP JCO、製品内に記載するSAPの製品/サービス名は、すべてドイツおよびその他の国におけるSAP AGの商標または登録商標です。

業務基盤ツール

業務基盤ツールで用意されているモジュール群は、標準のマスタメンテナンス画面で各種設定を行うだけで、すぐに活用することができ、構築する業務システムの基盤として利用することができます。

《機能UP!》アクセスセキュリティ・モジュール（[ログイングループ管理]-[ロール]フォルダ）

ユーザやロールなどのログイン・セキュリティ情報を操作するためのモジュールです。ログインするユーザの認証機能はもちろん、ユーザのアクセス権に応じた固有のWebページを表示することができます。たとえば、一般社員がアクセスしたときには、アクセス権限のないページはメニューにもあらわれないため、ユーザはその存在を一切意識することはありません。しかし、管理者がアクセスした場合には、同じページでも管理者が利用できるメニューまで表示されることになります。アクセスセキュリティ・モジュールを利用することで、このようなユーザに応じて内容の異なるページ構成を組むことが可能になります。設定方法についてはアドミニストレータガイド第2章「5　アクセスセキュリティの管理」を参照してください。
標準で用意されているアクセスセキュリティ・モジュールのほかに、シングルサインオンを実現するエクステンション・モジュール「IM-SecureSignOn（別売）」も用意されています。

〈ログインするユーザごとにアクセス権限のあるページメニューが表示される〉

intra-mart Ver5.0から、ユーザごとの言語の切り替えや画面のテーマカラーの選択が可能となりました。また、これまでユーザ情報の設定はシステム管理者がすべてを行う使用でしたが、システム管理者の配下にログイングループ管理者を設定し、ログイングループ管理者がユーザ情報の設定を行えるようになりました。

さらに、アクセスセキュリティの各機能はAPI化されており、それらを利用して、独自のメニュー画面を作成することもできます。

intra-martでは完全なSecure Sockets Layer（SSL）サポートを提供します。これにより送信されるコンテンツに対して、暗号化セキュリティをかけることが可能になります。さらに、ユーザ認証はLDAPとの連携も可能です。

シングルサインオンを実現するアクセスセキュリティ・モジュール：IM-SecureSignOn	独自のエージェント型リバースプロシキ方式により適用範囲が広く、導入・運用が容易なシングルサインオンを実現するエクステンション・モジュールです。簡易ログイン認証機能により、他システムやパッケージとの連携も容易です。

- エクステンション・モジュール「IM-SecureSignOn」の詳細に関しては、以降の「エクステンション・モジュール」の項、プログラミングガイドJ2EEベース開発モデル第3章「さまざまなJavaコンポーネント群（im-BizAPI）の利用」の「9エクステンション・モジュールの組み込みと操作」、ページベース開発モデル編第3章「さまざまなコンポーネント（im-BizAPI）の利用」の「15 エクステンション・モジュールの組み込みと操作」も参照してください。

Column

intra-martのアクセスセキュリティの考え方

intra-martのユーザには、ロール、組織、役職などの属性を設定することができます。そして、intra-mart画面のフォルダやページには、ロール、組織、役職などを「アクセス権限」として設定することができます。このため、ユーザがintra-martにログインすると、そのユーザが持つロール、組織、役職と一致する属性をアクセス権限として設定したフォルダやページだけが表示されるようになります。

ロールによるアクセスセキュリティの設定例

→（[ログイングループ管理]-[ロール設定]メニュー）
→（[ログイングループ管理]-[アカウント設定]メニュー）
→（[ログイングループ管理]-[アプリケーション共通マスタ]-[組織設定]メニュー）

Column

LDAPとの連携《機能up!》

intra-martは、ディレクトリサービスへの標準的なアクセス手段であるプロトコルLDAP（Lightweight Directory Access Protocol）にも対応しています。intra-martは、認証モジュールによって、intra-mart標準の認証システムをはじめ、LDAPサーバなどに接続して認証する形式となっています。この認証モジュールを変更することによって、各社の認証システムにも対応することができます。

LDAPとの連携についてはアドミニストレータガイドの第1章「11 LDAPとの連携」を参照してください。

Column

よく利用する画面を登録できる「マイメニュー設定」

intra-martでは、頻繁に利用する画面をユーザごとに「マイメニュー」として登録することができます。登録されたメニューは、標準で表示されるメニューの最上部に表示され、メニューの階層をたどらなくてもワンタッチでアクセスできるようになります。「マイメニュー」へのページの登録は、［ユーザ設定］の［マイメニュー設定］メニューで行います。

〈表示された「マイメニュー」とマイメニュー設定の画面〉
→（［ユーザ設定］-［マイメニュー設定］メニュー）

ワークフロー・モジュール　（[ワークフロー]フォルダ）《機能up!》

Webブラウザベースのワークフローを構築するためのモジュールです。当モジュールを利用すると、イントラネット環境での多階層の承認ワークフローが構築できます。あらかじめ設定された承認ルートにしたがって、承認者宛てに承認ページURLリンクの入った電子メールが伝達されます。そこから承認者は自分の承認ページを起動し、承認行為を行います。さらに承認の進捗状況のチェックなど、きめ細かいワークフローの構築が可能になります。電子メールは、URLリンク機能およびSMTPに対応した市販のメールソフトに対応しています。
標準で用意されているワークフロー・モジュールのほかに、エクステンション・モジュールとして「IM-ワークフローデザイナー（別売）」、「IM-ビジネスプロセスワークフロー（別売）」、そして「IM-FormatCreator（別売）」の3つが用意されています。

※複数の上長に対して、上記の申請承認ルートを繰り返す

承認ルートはあらかじめ設定しておくことや、申請/承認時に次の承認者を画面から選択することができます。また、既存の人事システムと連動すると面倒な承認ルートの設定をなくすこともできます。
ワークフロー・モジュールの持つ高度な機能の例は以下のようになっています。

差戻し機能	否認した申請を申請者または直前の承認者に戻す機能
代理申請承認機能	申請者、承認者が不在時に代理者が申請承認を行う機能
一括承認機能	承認者宛の複数の申請に対して、一括で承認/否認を行う機能
根回し機能	現在依頼承認中であることを他のユーザにもメールで知らせる機能
自動催促	承認期限がせまっている申請について、処置（承認/否認）を促す通知をメールで送信する機能
自動パス機能	承認期限となったまたは承認期限を過ぎた申請について自動的に承認し、次の承認者に申請を送信する機能
承認者の設定	承認ルート上の承認者は、個人だけでなく組織や役職でも指定可能（人事異動、組織改変に強い構造のルート設定が可能）

引戻し	申請者が申請を引き戻しも可能
承認案件振替機能	特定案件に対する承認権限を第三者に委譲する機能
申請者/承認者ルート作成ワークフロー《New》	稟議書の申請に代表されるような臨機応変なプロセスを実現するワークフロー。基本のプロセスは管理者が作成し、申請者は申請時に承認者を自由に選択することができます。設定できる同期タスクはひとつのみで、それ以上の複雑なプロセスは作成できません。さらに、承認者も承認時に、それ以降の承認者を設定することができます。

途中保存機能《New》	申請前に登録内容を途中保存し、次回申請時に途中からスタートすることができる機能
アプリケーション共通マスタの履歴化《New》	アプリケーション共通マスタの履歴化に対応したルート定義ができる機能。将来変更予定の組織情報をもとに、ルート情報を作成することができます。
電子申請書を自動生成するモジュール：IM-FormatCreator	ウィザード形式で電子申請画面を自動生成するソリューションで、intra-martのワークフローと連携して、申請や承認といった一連の流れをそのままシステム化できます。
起票日の変更《New》	起票日を過去・未来に渡って選択できます。異動が発生した場合にも、前職・現職どちらの立場で申請・承認を行うか選択できます。

ワークフローのステータスに対応したダイレクトメニューの採用《New》	ワークフローのステータスをそのままメニューに採用しているので、直感的でわかりやすくなっています。代理申請・承認も独立したメニューが用意されています（上図参照）。

● ワークフローの詳細に関しては、別冊のワークフローガイドを参照してください。

ビジネスプロセスワークフローモジュール（アドバンスド版には標準装備）

申請・承認が中心となるドキュメント・ワークフロー（標準のワークフロー・モジュールやIM-ワークフローデザイナーなど）とは異なり、ビジネスプロセスワークフローモジュールでは業務処理のプロセスをあらかじめ登録しておくことで、業務処理をそのまま自動化することができます。このため、事務処理の適正化を図ることができ、作業効率を大幅に向上させることができます。

≪ブラウザ上からグラフィカルにワークフロープロセスを定義することが可能≫

グラフィカルなプロセス定義により、業務プロセスを可視化。業務自体の分析を行い、効率化・最適化を実現します。また、業務を標準化することができるため、人材の流動化への対応が可能で、作業漏れを防止する効果もあります。

≪バージョンの設定によるプロセス定義の履歴管理が可能≫

プロセス定義のバージョン管理ができます。開始日・終了日を設定することで、あらかじめ作成しておいたプロセスを指定日（開始日）から適用することができます。

≪ドキュメント・ワークフローとの連携/自動実行処理もサポート≫

ドキュメント・ワークフロー（標準のワークフロー）をビジネスプロセスワークフローの中に混在させて運用することもできます。これにより、定型化した手続きのワークフロー（ドキュメント・ワークフロー）の完了後、プロセス間制御の自動化（ビジネスプロセスワークフロー）が実行されるルートを容易に混在させて実現することができます。

コンポーネントを自動で呼び出して実行されるショートランニングプロセス（IM-Σサーブ：エクステンションモジュール）もサポート。

（他システムとの連携をさまざまな方法で実装することができます）

≪高度なワークフロー機能の実現が可能≫

ワークフローの画面は、起票用の画面、承認用の画面、そしてそれぞれに処理済の画面が用意されています。また、代理起票や承認の画面も、それぞれ独立した専用の画面によって、ステータス管理が可能となっています。また、実業務に合わせて、詳細な検索機能や絞込み表示機能が備わっているため、把握した案件をすぐに確認することができます。

プロセスのスタートは、起票画面から「起票」ボタンをクリックすることから始まります。全体プロセス中のどこまで進展しているか、履歴を含めて一目でわかります（誰が、何時、どのプロセスをおこない、その結果はどうか）。また、代理人による処理、催促メール、自動スキップ、結果通知メール、差し戻し、引き戻し、一括処理など高度な機能も用意されています。

≪ポータル機能との密な連携≫

各プロセスのタスクを処理する際に必要な関連情報やアプリケーションを、あらかじめポータル画面に設定しておき、まとめて画面で確認できます。これにより複数システムの画面を立ち上げたり開いたりする手間を省くことができ、さらに事務処理の効率化を図ることができます。

≪既存システムとのスムーズな連携による高い拡張性を実現≫

他システムとの高い連携性により、ビジネスプロセスワークフローモジュールの適用範囲は、さまざまな既存システムをまたがった全業務プロセスを対象にして広げていくことができます。
これによって、複数システムをまたがった業務プロセス全体の最適化と、社内システムの統合効果を実現することが可能になります。

- ビジネスプロセスワークフローモジュールの組み込みに関しては、下記のページを参照してください。

ページベース開発モデル
プログラミングガイド　ページベース開発モデル編　第3章「15　エクステンション・モジュールの組込と操作」の「IM-ビジネスワークフロー（p.120）」
J2EEベース開発モデル
プログラミングガイド　J2EEベース開発モデル編　第3章「9　エクステンション・モジュールの組込と操作」の「IM-ビジネスワークフロー（p.269）」

バッチ管理モジュール　（[ログイングループ管理]-[バッチ設定]メニュー）

Webサーバ自身はもともとバッチ処理を想定していません。しかし、業務アプリケーションを構築する場合は、バッチで処理を実行したい時がでてきます。バッチ管理モジュールは、そうした要求を実現するためにサーバ上のサーバサイドスクリプトプログラムやJavaプログラムをバッチ実行できるスケジューラ機能を提供しています。intra-martの設定画面から、バッチプログラム（JavaScriptまたはJava）とプログラムの実行時刻を設定していきます。

ポータルモジュール（[ポータル]フォルダ、（[ユーザ設定]-[ポータル]フォルダ）《機能up!》

ポータルモジュールは、intra-martで作成したWebアプリケーションのフロントエンドに企業内ポータル（EIP：Enterprise Information Portal）機能を実現します。
intra-martのログイン初期画面に、あらかじめ用意したページ（ポータルアプリケーションでは、ポートレットと呼びます）を表示させることができます。よく利用するアプリケーションの画面やユーザに情報を見るように喚起する画面などを一覧表示することで、業務効率をあげることができます。
さまざまなアプリケーションで作成したページをポートレットとして登録し、ポータル画面に登録されているポートレットを自由にレイアウトすることができます。ポータル画面の組み合わせは複数作成することができ、ユーザはタブを切り替えて利用します。ポータル画面には、ログイングループ全体に表示させるものから、組織やロール単位に設定できるポータル、個人単位で利用できるポータルまで複数設定することができ、画面を切り替えることで業務効率を大幅に向上させることができます。

組織部署単位にポータルサイトを設定したり、個人単位の設定をすることで、組織内での情報伝達のスピードアップが図れる

会社（顧客、取引先）単位にポータルサイトを設定することにより、ECサイトやマーケットプレイスにおいて、
サービスレベルを大きく向上させることができる

さらに、他社のポータル製品との連携を図る、外部ソフトウェア接続モジュールが用意されていますので、複雑なポータルサイトを構築することもできます。

■カスタムレイアウト設定《機能up!》

標準で用意されている、「分割なし」、「左右分割」、「T字分割」、「逆T字分割」、「三分割」といった定型レイアウトに加え、新たに「カスタムレイアウト」機能が追加されています。レイアウト枠（縦×横）を選択し、行結合、列結合といった機能を利用して、ポータル画面を自由にレイアウトすることができます。

■グローバルポータル《新機能!》

intra-martにログインする以前に表示されるポータル画面で、ログイングループ全員に共通の情報を提供することができます。このグローバルポータルからintra-martにログインするために、「ログインポートレットが提供されています。また、この画面は、社外用HPとしても使用することができます。

■アクセス権限のパターン追加《機能up!》

グループポータルでは、社内組織構成に応じた最適なポータル画面を表示させるために、さまざまなアクセス権限を設定できます。Ver5.xから、さらに次のようなアクセス権限が追加され、さらに柔軟に表示することができるようになりました。

- 会社・組織
- 会社・組織　＋　ロール
- パブリックグループ
- パブリックグループ　＋　ロール
- ロール

■ポータル個別のキャッシュ機能《機能up!》

Ver5.xから、各ポートレットごとにキャッシュを行うように設定することが可能となりました。

■RSSコンテンツ対応《新機能!》

RSSコンテンツを扱うAPIが提供されています。標準では、このAPIを利用したRSSポートレットが付属しています。

■簡易SSOポートレット《新機能!》

他サイトを呼び出すさいに、事前にID・パスワードを設定しておくと、毎回ID・パスワードを入力することなく画面遷移することができるようになります。

■作成できるポータル画面の種類

ポータルの種類	表示される場所	作成者	基本表示範囲	アクセス権の設定
グローバルポータル	ログイン前の画面	ログイングループ管理者	ログイングループ全員	なし
グループポータル	intra-martのHOME画面（タブとして）	ログイングループ管理者	ログイングループ全員	会社・組織 パブリックグループ ロール
組織内ポータル	intra-martのHOME画面（タブとして）	一般ユーザ 組織のポータル管理者	作成者が主所属の組織（配下の組織は含まず）	配下の組織
ユーザポータル	intra-martHOME画面（タブとして）	一般ユーザ	作成者自身	なし

〈作成できるポータルの種類〉

ViewCreator（[ViewCreator]フォルダ）

ViewCreatorは、既存のデータベースに接続して、データの参照・更新が行える画面を自動生成するツールです。データベースからデータ構造を読み取り、データを明細表示するビュー画面と詳細データの参照更新が行えるフォーム画面を自動生成します。データベースから直接データ構造を読み取るほかに、テーブル定義テキストなどの情報から、ビューやフォームを作成することもできます。表示項目を絞り込む機能や並び順の変更、項目単位でアクセス権を設定する機能などが用意されていますので、既存データベースをもとにしたEUC（エンドユーザコンピューティング）が実現できます。詳細は、アドミニストレータガイドの第2章「12 ViewCreator」を参照してください

ユニット

これらの他にアプリケーションを作成する際に、よく利用されるプログラムをユニットとして用意しています。ユニットで提供されるプログラムもソースコードが公開されていますので、自由に機能拡張することができます。ユーザが作成したアプリケーションから呼び出すだけで、部品として再利用することが可能です。次のようなユニットが用意されています。各ユニットの詳細はプログラミングガイドのページベース開発モデル　第3章「14　ユニットの組み込みと操作」、J2EEベース開発モデル　第3章「8　ユニットの組み込みと操作」を参照してください。

アプリケーション共通マスタunit ([ログイングループ管理]・[アプリケーション共通マスタ]フォルダ)《機能up!》

会社データ、組織データ、グループデータ、取引先データ、顧客データ、商品データなど、システム開発でよく利用するマスタが標準で用意されています。これらのマスタを利用することにより、設計工程まで含め短期間でのシステム開発が可能となります。また、各intra-martアプリケーションシリーズは、すべてこのアプリケーション共通マスタをベースに構築されていますので、各intra-martアプリケーションシリーズと連携したシステムが開発できます。さらに、マスタにアクセスするためのAPIなども標準で用意されています。詳細については、別冊「アプリケーション共通マスタ説明資料」を参照してください。

これらのマスタはすべてデータベース仕様も公開されており、さらに用意されているAPIを利用しながらユーザアプリケーションを作りこんでいくことができます(詳細は「アプリケーション共通マスタ説明資料」を参照してください)。また、各intra-martアプリケーションシリーズも、アプリケーション共通マスタを利用して構築されていますので、必要なデータを相互連携させながらアプリケーションを拡張していくこともできます。

Column

アプリケーション共通マスタunitで定義される組織とは

「組織」とは、会社ー組織（事業部・部・課など）ーユーザというデータの階層を指します。会社を登録し、その会社に属する組織として、事業部や部・課・係などの組織を登録します。組織を階層化したり、組織のどの階層に対しても、ユーザを紐づけたりすることができます。

※　上記図中の「グループA」と「グループB」がパブリックグループに相当します。

→　（［ログイングループ管理］-［アプリケーション共通マスタ］-［組織設定］メニュー）

Column

アプリケーション共通マスタunitで定義されるパブリックグループとは

パブリックグループは、企業内の部門など組織にとらわれないユーザグループを作成したい場合に利用します。

→　（［ログイングループ管理］-［アプリケーション共通マスタ］-［パブリックグループ設定］メニュー）

Column

アプリケーション共通マスタunitで定義されるプライベートグループとは

パブリックグループと同様に、組織に関係なく自由に作成できるユーザグループで、ユーザ自身で自由にユーザグループを作成できるように用意しています。

→　（［ユーザ設定］-［プライベートグループ設定］メニュー）

Column

アプリケーション共通マスタunitで定義される商品マスタとは

さらに標準で用意されている商品マスタに対しても、intra-martで作成する複数のアプリケーションから共同で利用することができます（intra-mart Ver5.1からご利用になれます）。

→（[ログイングループ管理]-[アプリケーション共通マスタ]-[商品マスタ設定]）

Column

アプリケーション共通マスタの履歴化《新機能!》

Ver5.xから、これらの共通マスタには、期間情報が付加されています。例えば組織マスタにおいては、下図のように未来組織を予め登録しておき、指定日から予め登録済みの組織を有効にするといった使い方や、ワークフローなどでは、過去の組織の役職で処理を行うといった使い方ができます。

- intra-mart Ver5.xから、各マスタは日付で履歴管理することができます。
- 組織の期間管理に関しては、アドミニストレータガイドの「8 アプリケーション共通マスタ」の「組織の設定」を参照してください。

カレンダunit（[ログイングループ管理]-[カレンダー設定]メニュー）《機能up!》

日付の入力に使用するユニットで、曜日、休日、営業日等を管理する機能を持っています。会社独自の休日などの管理は、「ログイングループ管理」の「カレンダー設定」で行います。なお、カレンダーの設定は複数持つことができ、本社と工場、海外の支社など状況に応じて使い分けることができます。

ファイルアップロードunit/ファイルダウンロードunit

Webサーバを経由して、ファイルをアップロードしたりダウンロードしたりできます。Storage Serviceを利用すると、各サーバを分散配置している環境においてファイルを一元的に管理することが可能となります。たとえば、アップロード文書ファイルをStorage Serviceに集中的において共有することもできます。

〈Storage Serviceでファイルを一元管理する〉

「IM-PDFデザイナー（別売）」をStorage Serviceで稼働させると、生成されたPDFファイルを各クライアントからダウンロードすることができます。また、Storage Serviceは、Schedule Serviceからも利用することができます。

〈Storage ServiceでIM-PDFデザイナーを稼働して、PDFファイルを一元管理する〉

- Storage Service上のファイルを取り扱うには、アプリケーション共通モジュールのVirtualFile オブジェクトを利用します。詳細は、「APIリスト」を参照してください。さらに、Moduleオブジェクトによりアップロード、ダウンロードが可能になります。

ツリー表示unit （[ログイングループ管理]-[メニュー設定]メニュー）

ツリー表示unitを組み込むと階層化されたデータをツリー表示することができ、階層構造の把握やメニューの選択が用意になります。ツリー表示ユニットの詳細については、APIリストの「ユニット」-「ツリー表示」を参照してください。

ツリー表示unitの表示例

iモードunit （[ユーザ設定]-[モバイルパスワード設定]メニュー）

iモードunitを組み込むと、データベースと連動したWeb業務アプリケーションをiモードからも利用できます。作成したページは、i-mode専用画面、通常のパソコン用の画面、そして両方に兼用の画面かを設定できます。また、「外出設定」の機能では、intra-martから送信するメールを、通常使用しているデスクのパソコンに送信するか、デスクのパソコンと同時に外出先のi-mode対応携帯電話に送信するかを設定できます。

アプリケーション共通マスタの履歴化《新機能!》

Ver5.xから、これらの共通マスタには、期間情報が付加されています。例えば組織マスタにおいては、下図のように未来組織を予め登録しておき、指定日から予め登録済みの組織を有効にするといった使い方や、ワークフローなどでは、過去の組織の役職で処理を行うといった使い方ができます。

- intra-mart Ver5.xから、各マスタは日付で履歴管理することができます。
- 組織の期間管理に関しては、アドミニストレータガイドの「8 アプリケーション共通マスタ」の「組織の設定」を参照してください。

多言語対応

ログインユーザごとに、使用する言語を切替えることができるアプリケーションの開発が行えます。言語を切り替えると、メニュー項目名などがユーザの指定した言語に切り替わります。

標準画面の作り方（共通画面デザイン）

intra-martの標準画面は、提供されている共通化APIを利用して、オリジナルな画面デザインでアプリケーションを開発することができます。デザインが共通化されると、他の画面と見た目や操作性が統一され、メニューから呼び出されたときに利用者が違和感なくアプリケーションを操作できるなどのメリットがありますので、アプリケーション開発の際には、デザイン統一の方法としてこのガイドラインの利用を検討してください。画面作成に関しては、以下のドキュメントが用意されています。

- 画面デザインガイドライン
- スタイルシート仕様書

画面デザインガイドラインに準じたAPIも用意されています。API仕様は、画面デザインガイドラインのドキュメントに掲載されています。

- ベースモジュール/フレームワークの持つ各画面ソースは、そのほとんどがプレーンな状態でインストールされています。画面デザインガイドラインの適用方法や、APIの使用例としてご活用ください。

Column

intra-martベースモジュール/フレームワークを利用する上での制限

intra-martベースモジュールならびにフレームワークを利用したアプリケーション構築には、いくつかの制限事項があります。

1 ユーザ設定画面の利用

intra-martへのログインユーザ管理は、「ユーザ設定」画面を使用しています。アプリケーション構築の際にも、ログインユーザの管理は、そのまま「ユーザ設定」を利用してください。　アクセスセキュリティ・モジュールやワークフロー・モジュールなどは、このユーザ設定で作成されたアカウント情報の利用が前提で稼動します（次のColumnを参照してください）。　ユーザ管理マスタの登録についてはアドミニストレータガイド第2章「8　アプリケーション共通マスタ」の「ユーザプロファイルとログインアカウントの登録」を参照してください。

2 フォルダ/ページのメニュー構成

intra-martのメニュー構成は、メニューツリーのフレーム利用が前提となっています。特に、アクセスセキュリティ・モジュールは、このメニュー構成に従うことが前提となっています。フォルダやページへの登録については、アドミニストレータガイド第2章「7　アプリケーションの登録」を参照してください。ただし、アクセスセキュリティの各機能はAPI化されており、それらを利用して独自のメニュー画面を作成することができます。

また、これらのログインユーザ管理機能やフォルダ/ページ機能を利用せずにintra-martをDBと連携した機能を利用することも可能です。詳細は、APIリストの「Developer´s　Guide」 - 「その他」 - 「アクセスセキュリティ・モジュールを利用しないで画面を構築する方法」を参照してください。

Column

intra-martで提供する各モジュール（im—BizAPI）のアカウント情報への依存について

intra-martの各モジュールには、intra-martのアカウント情報（「ユーザ設定」画面でのユーザ管理）の利用時にしか使用できないものもあります。下記表を参考にしてください。また一部モジュールには、別のモジュールの利用を前提として使用可能となるものもあります。下記の注意点を参考にしてください。

intra-martモジュール（im-BizAPI）	アカウント情報なしで： ○利用可能、△一部利用不能、×利用不能
画面共通モジュール	○
グラフ描画モジュール	○
アプリケーション共通モジュール	△*3
メール連携モジュール	○
ワークフローモジュール/ビジネスプロセスワークフロー*1	×
アクセスセキュリティモジュール	×
バッチ管理モジュール	○
ポータルモジュール*1	×
カレンダーunit	○
アプリケーション共通マスタunit*2	×
ファイルダウンロードunit	○
ファイルアップロードunit	○
i-mode unit	○
ツリー表示unit	○
エクステンションモジュール	
IM-PDFデザイナー	○
IM-ワークフローデザイナー*2	×
統合検索ソリューション	○
IM-XRFコンポーザー（大量帳票印刷オプション）	○
マルチデバイスソリューション	○

*1 アクセスセキュリティ・モジュールの利用が前提のモジュールです（intra-martのアカウント情報をプログラム内部で使用して動作するモジュールです）。

*2 IM-ワークフローデザイナーは、ワークフロー・モジュールの利用を前提として動作します（結果として、アクセス情報なしでは動作しません）。

*3 アプリケーション共通モジュールで、intra-martのアカウント情報を使わないときに利用できないものの一覧です。

アクセスセキュリティに関連した機能：jp.co.intra_mart.foundation.security.*

ワークフロー機能：jp.co.intra_mart.foundation.bpw.*

その他：jp.co.intra_mart.foundation.datastore.application.*

エクステンション・モジュール

エクステンション・モジュールは、intra-martベースモジュールやintra-martフレームワークに標準添付されているモジュールとは別に用意されている拡張モジュール群（オプション製品）です。より高機能なモジュールが必要なユーザのために用意されており、必要に応じて組み込んで標準のモジュールと同様に利用することができます。現在用意されているモジュールには、次のようなものがあります。

画面共通モジュール拡張

標準で用意されている画面共通モジュールのほかに、ユーザビリティの飛躍的な向上を実現するエクステンション・モジュール「IM-Curl（別売）」が用意されています。

IM-Curl（別売）

intra-martと連携したビジュアルなクライアントGUI環境を実現する、エクステンション・モジュールです。ユーザビリティの飛躍的な向上と効率的な開発を実現します。
IM-Curlは、高度かつ複雑なユーザーインターフェイスを構築可能なリッチクライアントサイドWebプログラミング言語です。IM-Curlは、J2EEベース開発モデルのみの提供です。

● IM-Curlは、俊敏な操作性と豊かなGUIを提供します

最近のWeb業務アプリケーションは、クライアントの表現力や操作性が求められています。
IM-Curlは、HTMLライクなテキストフォーマットから高パフォーマンス3Dグラフィックスまで、広範囲な開発記述能力を持っています。

〈IM-Curlの画面例〉

● IM-Curlは、サーバ負荷を軽減します

Webアプリケーションが高度になるほどサーバには負荷がかかりますが、IM-Curlでは、実用性に優れたアプリケーションの処理をクライアントサイドで実装することにより、サーバに負担を減少させ、通信帯域の有効活用をします。IM-Curlはアプリケーションの処理をサーバとクライアント側（各PC）に分散させることにより、より高速でサーバの負担がないスムーズな処理を実現します。

- 詳細に関しては以下の資料を参照してください。
  - 製品に添付されている、Curlアプレットが相互に通信するアプリケーションを作成するために用意されているCurl連携用APIを利用するためのチュートリアルガイド
    「IM-Curl連携APIチュートリアルガイド」（im_curl_api_tutorial.pdf）
  - APIリストの「J2EEベース開発モデル」-「ビジネスロジック層」-「アプリケーション共通モジュール」内のjp.co.intra_mart.foundation.im_curlクラス

帳票印刷モジュール拡張

帳票印刷モジュール拡張として、よりきめ細かな帳票の印刷に対応したモジュールや、大量の帳票印刷に適したエクステンション・モジュールがオプションで用意されています。

IM-PDFデザイナー（別売）

PDFを採用することにより、より複雑な帳票形式やページ制御の対応が可能となるモジュールで、単票および連票に対応します。ビジュアルな帳票デザイン作成ツール「IOWebDoc」で帳票フォーマットを作成します。ユーザアプリケーションからのデータはCSV形式またはメモリ経由で受け渡し、PDFファイルを作成してからAcrobatを起動して印刷します。複数のアプリケーションサーバを配置した環境では、IM-PDFデザイナーをStorage Service上で稼動させることによって生成したPDFファイルを一元管理して、クライアントへダウンロードすることが可能になります。

〈IM-PDFデザイナーサンプル画面〉

IM-XRFコンポーザ（大量帳票印刷）（別売）

Webシステムの信頼性強化にともない、オフコンや汎用機の置き換え案件が急増しているなかで、従来行ってきているサーバでの大量帳票印刷は続けたいという課題があります。PDFデザイナーを利用することもできますが、生成されたPDFファイルのサイズが大きくクライアントへの配布に時間がかかという問題があります。IM-XRFコンポーザを利用することにより、大量帳票印刷時の問題を簡単に解決することができます。

- きめ細かい帳票作成が簡単に行え、開発工数を大幅に削減できます。
- 圧倒的に小さいファイルサイズと高速生成で、Web環境や大量帳票運用に適したXRFファイルと、PDFファイルの生成が可能です。
- JointBase（プリズム社）との連携により、インターネット先のプリンタにもダイレクトに印刷を行うことができます。

- ブラウザに専用プラグインのインストールが必要です。プラグインは無償配布しています。
- フォームヘルパーにより帳票フォームの作成もGUIベースで可能です。

アクセスセキュリティ・モジュール拡張

標準で用意されているアクセスセキュリティ・モジュールのほかに、シングルサインオンを実現するエクステンション・モジュールがオプションで用意されています。

■ IM-SecureSignOn（セキュア・サイン・オン）（別売）

IM-SecureSignOnは、シングルサインオンを実現するツールです。社内のさまざまなWebシステムへのログインも、このSecureSignOnに一度ログインするだけ、すべての認証が完了します。独自のエージェント型リバースプロキシ方式*により、適応範囲が広く、導入・運用が容易なシステムです。また、簡易ログイン認証機能により、他システムやパッケージとの連携も容易です。

〈図版タイトル〉

＊ エージェント型リバースプロキシ方式

シングルサインオンを実現するには、一般的にリバースプロキシ方式とエージェントモジュール方式の2つの方式が考えられます。IM-SecureSignOnでは、これら両者の長所を兼ね備えた独自のエージェント型リバースプロキシ方式を採用しています。

方式名	主な用途	特 長	システムの概念図
エージェント型リバースプロキシー方式	管理分散型 イントラネットに適したシングルサインオンの方式	・負荷が集中しない ・各エージェントが直接アクセスを受ける ・大規模ユーザに対応可能 ・Webサーバの分散配置構成に対応可能 ・Webサーバ、ホストOSの種類を問わない	

ワークフロー・モジュール拡張

標準で用意されているワークフロー・モジュール（スタンダード版）、ビジネスプロセスワークフローモジュール（アドバンスド版）のほかに、ワークフローモジュールの拡張製品として、「IM-ワークフローデザイナー（別売・アドバンスド版には同梱）」、「IM-FormatCreator（別売）」、そして「IM-Σサーブ（別売）」などのエクステンション・モジュールが用意されています。以下に、製品構成を示します。

ドキュメントワークフロー関連製品構成（スタンダード版）

ビジネスプロセスワークフロー関連製品構成（アドバンスド版）

■ IM-ワークフローデザイナー（別売・アドバンスド版には同梱）

IM-ワークフローデザイナーは、標準で用意されているワークフローモジュールに機能を追加するエクステンションモジュールとして提供されています。追加される機能は、プロセス（承認ルート）の同期、条件分岐で、より複雑な状況に対応するワークフローです。プロセス結合時や分岐する際の条件処理は、Javaプログラムによりカスタマイズして記述することができます。その他の機能や操作は、標準のワークフローに準じています。

〈IM-ワークフローデザイナーサンプル画面〉

■ IM-FormatCreator（別売）

IM-FormatCreatorは、電子申請のための申請書フォーマット画面を自動生成するソリューションです。用意されている作成ウィザード形式で、すばやく申請書作成に対応できます。「人事異動」などのイベントを選択するだけで、必要となる申請書が自動的に表示されるなど、運用に即した機能が用意されています。しかも、intra-martのワークフローと連携して、作成した申請書の申請や承認といった一連の流れをそのままシステム化できます。IM-FormatCreatorには、以下のような特長を持っています。

ノンプログラミングで申請書を自動作成	作成ウィザードにしたがってintra-mart上で動作する申請書入力画面のレイアウトを自動作成。HTML、JavaScript、XMLなどのWeb画面作成言語、およびデータベースに関する知識を必要としません。画面入力項目に対する入力必須や文字数などの基本チェックの設定もOK。画面に対する項目追加/変更も、ノンプログラミングで容易に行えます。
運用に即したイベント検索機能	「人事異動」「結婚」などのイベントを選択することにより、提出する必要のある申請書類を自動的に抽出することができます。イベント内容も自由に設定可能です。
ワークフロー機能と連携	作成した申請書は、intra-martのワークフロー機能と連携しているため、複雑なワークフローにも柔軟に対応できます。
メンテナンス性の向上	様式変更が必要になった場合も、ノンプログラミングで容易に対応できます。これにより、システム保守費用を削減できます。

レイアウト作成（管理者）

レイアウト枠設定

登録する申請書入力画面のタイトル・表のサイズを決めます。

項目設定

セルに対して以下のうち項目の内容が選択・設定できます。
文字数やチェックなどが個々に設定できます。

プレビュー

プレビュー機能で設定した内容を確認することができます。

申請書の利用（申請者）

イベント検索

申請書一覧表示

各イベントで必要な申請書を一覧表示

申請入力

申請書入力後はそのままワークフローで申請

〈申請書の作成と利用〉

■ IM-Σ Serv（別売）

IM-ΣServは企業内に分散している複数の既存システムを連携する、バックエンドシステム統合プラットフォーム(ESB)です。システム間のデータの一貫性を保証する機能も実装しており、Webサービス同士を高信頼に連携させるアプリケーションを、容易に開発することが可能です。
ビジネスプロセスワークフローモジュールと連携することで、バックエンドも含めたダイナミックかつミッションクリティカルなシステム統合が実現します。

≪ショート&ロングランニングトランザクションに対応したシステムを実現≫

intra-martのビジネスプロセスワークフロー（アドバンスド版に同梱）は、さまざまなWebシステムを業務フローに従って、柔軟に連携させることができます。いわゆるロングランニングトランザクションを実現しています。一方、IM-ΣServは、企業内のさまざまな既存システムを高信頼で結ぶ「ショートランニングトランザクション」を実現しています。そして、intra-martのビジネスプロセスワークフローからIM-ΣServと連携することにより、複数の業務プロセスや業務システムを統合・制御・自動化し、業務フロー全体を最適化するビジネスプロセスマネジメント（BPM）が実現します。

≪IM-ΣServの高信頼性　― トランザクション≫

IM-ΣServは、エラー発生時に連携システム間のデータをの一貫性を保証する機能を持っています。

≪IM-ΣServの高信頼性 ― メッセージ保証≫

IM-ΣServは、HTTP上の高信頼なメッセージを実現するためのさまざまな機能を持っています。

外部ソフトウェア連携ソリューション

各社が提供する外部ソフトウェアを利用するための連携ソリューションが用意されています。

統合検索ソリューション

IM-統合検索ソリューションは、(株)ジャストシステムの「ConceptBase」と連携することで、MS-Word、一太郎、MS-Excel、PDFファイルなどさまざまなファイル形式のドキュメントから自然文によるナレッジ検索を行うことができる文書検索システムです。

ConceptBaseサーバとintra-mart Storage Service間の接続モジュールを提供することにより、ConceptBaseの強力な文書検索機能を利用したシステムの構築が可能となります。また、検索した文書から、内容やドキュメントの要約をテキストで取得することもできます。

Conscept Base Searchは文書ファイルを対象とした検索、CB RDB GatewayはRDB内の情報を対象とした検索、CBゲートウェイfor Lotus NotesはNotes DBを対象とした検索が可能となります。

マルチデバイスソリューション

フレックスファーム社のX-Servletの利用により、マルチデバイスへのコンテンツ配信が可能になります。詳細については、X-Servlet付属のマニュアルを参照してください。

■主な特徴

シングルURL	ひとつのURLにアクセスすると、自動的にアクセスされた機種を認識します。
シングルソース	1つのコードですべての国内移動体通信会社のアクセス方式に対応します。
オートレイアウト	アクセスされた機種に対応し、画像の変換やページの自動分割を行います。

4 intra-martのアプリケーション開発概要

intra-martベースモジュールあるいはintra-martフレームワークを使ったアプリケーションの開発において、開発者はブラウザ上に表示されたユーザインタフェースと、Webサーバ上で動作するビジネスロジックを作成することになります。

ページベース開発モデルではプレゼンテーション・ページ（HTMLファイル）とファンクションコンテナ（サーバサイドJavaScriptファイル）の2つのファイルを作成します。この際、フレームワークで用意されているモジュール群（im-BizAPI）を活用することでさらに生産性を向上させることができます。

J2EEベース開発モデルでは、JSPファイルとServlet、EJBコンポーネントで開発します。
この際にも、フレームワークで用意されているモジュール群（im-BizAPI）をJ2EEフレームワーク（im-J2EE Framework）とともに利用することで、煩雑なJ2EEによるWebシステム開発をさらに効率化するとともに、業務コンポーネントの再利用を促進することができます。

Column

スクリプトとJavaのサンプルコード比較

JavaScriptにはクラスという概念がないので、その分プログラマの労力が少なくてすみますが、Javaは変数型の扱いが厳密であり、クラス定義が必須となります。JavaScriptはJavaをベースに開発された言語なので、基本部分の見た目では差違はあまり感じられませんが、細かい部分でスクリプト言語らしく簡単に扱えるように作られており、全体としてJavaほど高度な感じを受けることはありません。

同じ内容であっても下記のように記述量に違いがあります。

JavaScriptによる記述

```
var obj = new Object();
        obj.foo = "intra−mart";
        obj.age = 20;
        var foo = obj.foo;
        var age = obj.age;
```

Javaによる記述

```
public class MyObject{
        public static java.lang.String foo;
        public static int age;

        public MyObject(){
                super();
        }
}
        .
        .
        .
MyObject obj new MyObject();
obj.foo = "intra−mart";
obj.age = 20;
java.lang.String foo = obj.foo;
int age = obj.age;
```

内容が同じでも、記述量に違いがある

ページベース開発モデルによるアプリケーション開発

「ファンクションコンテナ」の中には、ビジネスロジックがJavaScriptで記述されており、「プレゼンテーション・ページ」から呼び出され実行されます。その橋渡しの機能をintra-martが実現しています。

プレゼンテーションページ

プレゼンテーションページは、ユーザインタフェース部分に相当します。拡張子は「.html」で固定となります。開発者またはエンドユーザは、“intra-mart eBuilder”を利用して、Webベースのプレゼンテーションページを作成していきます。
また、ブラウザベースのユーザインタフェースには、動画像やサウンドなどを盛り込んだマルチメディアリッチな画面を構築していくことができます。さらに、プレゼンテーションページはHTMLファイルであるため、Webシステムの開発において、ユーザインタフェース部分のみを切り出してホームページデザイナに作業を依頼することもできます。ホームページ作成ツールから生成されるHTMLファイルに〈IMART〉タグを追加していくことで、ファンクションコンテナにあるJavaScriptを関連付けて呼び出すことが可能になります。ユーザ定義関数を呼び出す〈IMART〉拡張タグも追加できます。完成したHTMLファイルは、ページ登録をするだけで、すぐにデータベースと連動して高速動作します。

① 市販のホームページ作成ツール（Microsoft FrontPageやHOMEPAGE BUILDER等）やintra-mart eBuilderを利用してユーザインタフェースとなるプレゼンテーションページのひな型を作成します。

② ホームページ作成ツールから自動生成されたHTMLソースに、ファンクションコンテナ中のJavaScript関数やベースモジュールのオブジェクトと連携する〈IMART〉タグをintra-mart eBuilderやテキストエディタを使い挿入（記述）します。

プレゼンテーションページのサンプル例を示します。
intra-martの独自拡張タグ〈IMART〉を利用して、各種モジュールを呼び出していきます。

＜社員マスタからのデータ取得用HTML（一覧表示用）ー項目を拡張＞

```
 1:  <HTML>
 2:  <BODY>
 3:  <TABLE border="1">
 4:  <TR>
 5:      <TD>社員コード</TD>
 6:      <TD>社員名</TD>
※ 7:      <TD>社員名（カナ）</TD>
 8:  </TR>
 9:  <IMART type="repeat" list=staffList item="record">
10:  <TR>
11:      <TD><IMART type="string" value=record.staff_cd></IMART></TD>
12:      <TD><IMART type="string" value=record.stf_name_kanji></IMART></TD>
※13:     <TD><IMART type="string" value=record.stf_name_kana></IMART></TD>
14:  </TR>
15:  </IMART>
16:   </TABLE>
17:  </BODY>
18:  </HTML>
```

独自拡張タグ〈IMART〉を利用して、各種モジュールを呼び出していきます。

ファンクションコンテナ

多階層アーキテクチャのうちのApplication Runtime上で稼動するビジネスロジック部分に相当します。拡張子は「.js」で固定となります。ファンクションコンテナとプレゼンテーションページはワンセットとなっているため、ファイルラベル名は同一のものを使用します。開発者は、プレゼンテーションページから呼び出されるJavaScriptを、ファンクション・コンテナの中に記述し作成していきます。具体的には、intra-martベースモジュールやintra-martフレームワークに用意されているフレームワーク機能から必要なオブジェクトや関数群を選び出し、“intra-mart eBuilder”で、それらオブジェクトや関数群を利用したサーバサイドで稼動するビジネスロジックをJavaScriptで記述し作成していきます。データベースへのSQL文もファンクション・コンテナの中に記述していきます。実際のRDBとの接続やSQL発行は、intra-martベースモジュールやintra-martフレームワークから実行されるため、細かなセッション管理やトランザクション管理を開発者は意識する必要はありません。
作成されたビジネスロジックは、プレゼンテーションページの〈IMART〉タグから呼び出され実行されることになります。intra-martベースモジュールやintra-martフレームワークに用意されているフレームワーク機能の詳細は、「intra-mart APIリスト」に一覧記述されています。これらスクリプトの記述はJavaScriptで行えるため、習得が難しいといわれるJavaやActiveXを用いることなく、これまでのホームページ作成の延長でデータベースと連動した本格的なWebシステムの開発が可能になります。

画面共通モジュール
帳票印刷
ワークフロー
アクセスセキュリティ
グラフ描画
メール連帯
SQL文

J2EEベース開発モデルによるアプリケーション開発

OSやWebアプリケーションサーバに依存しない共有プラットフォームとして、J2EEによるWebシステム開発が普及してきました。しかし、J2EEによる開発はJavaをベースにしているため、オブジェクト指向などの高度な知識と経験が要求される点や、J2EEでの前提知識が必要になる点など、敷居の高さが問題になってきています。さらに、J2EE開発の規約にさえ準拠させれば、あとはいかようにでも組める自由さが、初心者にとってはかえって負担となり、SEによってバラバラな開発スタイルとなってしまう原因ともなります。
intra-martベースモジュールやintra-martフレームワークでは、これらの問題をJ2EEフレームワーク（im-J2EE Framework）の利用により解決し、J2EEベース開発モデルの生産性を大幅に向上させています。

J2EE開発のフレームワーク

J2EEでのWebシステム開発には、構造的に共通な部分が多く、その事実を利用すると、開発生産性をさらに大きく向上させることができます。intra-martベースモジュールやintra-martフレームワークでは、J2EE開発で必要になる共通的な処理は、すべてJ2EEフレームワークとして用意し、開発者に委ねられる箇所はコンポーネントを作成してもらう形態となっています。

〈intra-martフレームワークの利用概念〉

上図のフレームワーク部分にあるように、大きくは**「Java業務コンポーネント群(im-BizAPI)」**と**「J2EEフレームワーク(im-J2EE Framework)」**に分かれています。Javaコンポーネントモジュール群(ワークフロー・モジュールやアクセスセキュリティ・モジュールなど)はすべて「Java業務コンポーネント群(im-BizAPI)」に配置されます。そしてJ2EE開発で必要になる共通的な処理はすべて「J2EEフレームワーク(im-J2EE Framework)」に配置されます。
特に、このフレームワークを利用するコンポーネントについては、雛型となる「スケルトン」が用意されており、さらに生産性を高めることが可能となります(共通部分はあらかじめ実装されているので、開発者は独自のコンポーネントを作成する場合、スケルトンをコピーし、実装されていない箇所を埋めてコンポーネントを作成することになります)。作成されたコンポーネントはプロパティ設定によりフレームワークにあてはめていくことで動作します。
フレームワークもJ2EEで定義されている4層(Client層、Web層、EJB層、EIS層)を使用する共通的な機構となります(SunのBluePrintに準拠)が、開発者はそのことを一切意識する必要はありません(システム規模に合わせてあとから構成変更することも、設定ファイルの変更のみで可能です)。

intra-martのフレームワーク機能は、以下のフレームワークから構成されています。

① **サービスフレームワーク**	Client層のブラウザからの要求受付の画面遷移(Web層) (下記注を参照してください)
② **イベントフレームワーク**	主にデータアクセスなどの処理の実行(EJB層)
③ **データフレームワーク**	データストア(EIS層)へのアクセス

- **プレゼンテーションフレームワークとサービスフレームワーク**
  intra-mart Ver4.3から新たにWeb層のフレームワークとして「プレゼンテーションフレームワーク」が追加されました。Web層のフレームワークには、以前より「サービスフレームワーク」が用意されていますが、それぞれの特徴を活かして、効率的に開発を進める必要があります。
  「プレゼンテーションフレームワーク」はフロー制御の強化やリクエストの二重送信の禁止、入力チェックのアドオンなど、さまざまな点でサービスフレームワークよりも機能が強化されていて、業務システム特有の複雑な画面を作成する際に、特に威力を発揮し、生産性の向上と拡張性が見込めます。
  一方、従来からある「サービスフレーム」は、比較的簡素で一般的な画面を手軽に作成できるメリットがあります。
  開発に当たっては、これらWeb層の2つのフレームワークを、利用目的に合わせて効果的に使い分けることをお勧めします。

≪プレゼンテーションフレームワークの特徴≫

- 共通的な概念をモデル、個別の用途に適用されものをコンポーネントとして定義し、再利用性が高まります。
- 入力内容のチェック、処理、次画面遷移という一連のフローを制御します。
- コントローラコンバータ、バリデータ、サービスコントローラ、フォワードセレクタ、ビューコンバータはプロパティによって設定されます。
- プロパティを変更することで、処理や遷移先の変更が容易です(ソース修正は不要です)。
- 遷移先がひとつに決まる場合は、コントローラコンバータ、バリデータ、サービスコントローラ、フォワードセレクタの設定は不要となります(プロパティの変更のみで済みます)。
- 遷移先の画面の表示内容が固定である場合、ビューコンバータの設定は不要となります。処理についてはイベントフレームワークを利用します(「イベントフレームワーク」を参照してください)。

これらのフレームワークが、プロパティに設定された内容に従い、該当するコンポーネントを呼び出します。これにより、WebシステムにJ2EEのMVCモデルで処理が実行されることになります。

これらのフレームワーク機能を適用することで、開発するシステムを、コンポーネント群とプロパティファイルの設定をベースとした構造に分けることができます。その結果、システムは柔軟にカスタマイズや機能追加できるようになります。

Strutsの利用

プレゼンテーションフレームワークやサービスフレームワーク部分にオープンフレームワークとしてJakartaプロジェクトから提供されているStrutsを利用することもできます。Strutsでは、Web層（Servlet/JSP）が対象のフレームワークであるため、im-J2EE Frameworkのイベントフレームワーク/データフレームワークと接続して利用することで、EJB/EIS層までを対象とした完全なフレームワーク（Full J2EE Framework）として活用可能となります。

■Strutsの利用時のフレームワーク体系

Struts（＝Thin Framework）に、intra-mart J2EE Framework（イベントフレームワーク、データフレームワーク）を拡張することで、EJBコンポーネントの活用までを対象にしたFull J2EE Frameworkが実現できます。さらに、java業務コンポーネント（im-Biz-API）を用いたFull業務アプリケーション作成用のFrameworkを実現することが可能です。また、intra-martで用意された各種アプリケーション群の利用も可能です。

Strutsとim-J2EE Frameworkとの連携については、インストールCD-ROM内の「intra-mart im-J2EE Framework Struts連携ガイド」を参照してください。

J2EE開発モデルにおいてフレームワークを活用した際のメリット

J2EE開発時にJ2EEフレームワークを利用することで以下のメリットがでてきます。

J2EEベースの開発基盤

高度な知識が必要となる部分は隠蔽し、開発者はアプリケーションロジックをスケルトンの中に埋め込みます。前提知識がなくても完成したシステムはJ2EEモデルの推奨型となり、MVCモデルが容易に実現します（プログラム構造が統一できメンテナンス性向上）。

サーバ構成を意識しなくてもよい

サーバ構成が変更されても設定ファイルなどにより外部から調整が可能です。

生産性の向上

共通的なものはすべてフレームワーク中に用意されているので、コンポーネントの再利用性が高まり、生産性および保守性が向上します。

カスタマイズの容易さ

コンポーネントの新規追加時にも、元のアプリケーションを変更する必要はありません。また、機能変更時も、該当コンポーネントのみを修正すればよく、他の箇所に影響しません。

＜フレームワークの適用の有無によるカスタマイズ範囲の違い：○は、影響のある箇所＞

順番	変更内容	フレームワーク適用：無			フレームワーク適用：有		
		JSP	Servlet	Bean	JSP	Servlet	Bean
画面カスタマイズ	画面レイアウト変更	○			○		
	画面表示項目追加	○	○	○	○		○
	画面表示形式変更	○	○	○	○		
	画面表示項目削除	○	○	○	○		
	画面フレーム分割	○	○	○	○		
	画面フレーム統合	○	○	○	○		
業務処理アドオン	業務処理追加	○	○	○			○
	業務処理変更	○	○	○			○
DBカスタマイズ	DB項目追加		○	○			○
	DB項目削除		○	○			○

テスト工程期間の短縮

テスト工程でフレームワーク部分の確認が不要となります。また、問題発生時にも切り分けが可能となり、すばやく問題に対処できます。

- im-J2EE Frameworkは、im-BizAPIを使用せずに単独で利用することもできます。この場合、フレームワークで規定されるのは基本的な部分に限定されることになるため、拡張の容易なシステムとすることができます（反面、システム共通のコンポーネント群であるim-BizAPIを利用すると、多くの処理をフレームワーク側に任せることができるので、システム全体の開発生産性を向上することができるとともに、品質安定します）。フレームワークのどの範囲まで利用するかは、構築するシステムの内容に合わせて判断してください。

intra-martのアプリケーションの動作概要

ここでは、intra-martのアプリケーションの概要について、サーバサイドで作成されたアプリケーションがどのように動作するのか、その流れと仕組みをブラウザ/サーバ/データベースの多階層アーキテクチャに対応して説明します。

〈アプリケーション動作フロー〉

この図は、説明の都合上、ベースモジュールの上部が「ページベース開発モデル」、下部が「J2EEベース開発モデル」のそれぞれの流れとなっています。

1 ブラウザからリクエストの発行	ブラウザからサーバに対して、アプリケーション起動のリクエストを発行します。リクエストは、Webサーバ/Webサーバ接続モジュール(intra-martベースモジュールの場合のみ)を経由してApplication Runtimeあるいは Webアプリケーションサーバに受け渡されます。
2 アプリケーションロジックの解釈	リクエストを受け取ったApplication RuntimeあるいはWebアプリケーションサーバ上のアプリケーションロジックがintra-martエンジン上で解釈され実行されます。アプリケーションロジックは、ページベース開発モデルではファンクションコンテナ、J2EE開発モデルでは、Servlet/EJBに記述されます。
3 im-BizAPIの呼び出し	ファンクションコンテナやServlet/EJBに記述されたAPIに対応するJava業務コンポーネント(im−BizAPI)を呼び出します。
4 SQL文の発行	アプリケーションがデータベースと連携している場合には、intra-martベースモジュール/フレームワークがSQL文を発行します。
5 セッション情報の管理	ユーザ固有の情報(セッション情報)はApplication Runtimeあるいは Webアプリケーションサーバの Http Sessionで管理されます。
6 実行結果取得	intra-martは発行したSQL文に対応する実行結果をデータベースより受け取ります。
7 ユーザインタフェースの生成	プレゼンテンションページの〈IMART〉タグ、またはJSPファイルに結果が埋め込まれ、HTMLソースが動的に再構築されます。
8 結果の送信と表示	生成されたHTMLファイルがブラウザに送信され表示されます。

5 intra-martのシステムアーキテクチャー

intra-martベースモジュールあるいはintra-martフレームワークのシステムアーキテクチャは以下のようになります。

intra-martベースモジュールのシステムアーキテクチャー

以下に、intra-martベースモジュールのシステムアーキテクチャーを示します。

〈intra-martベースモジュールのシステムアーキテクチャー：4台の負荷分散構成例〉

Web Server Connector **：Webサーバ接続モジュール**	Webサーバに組み込むモジュールです。Webブラウザからのユーザのリクエストを Web サーバから Service Platform に送ったり、Service Platformの実行結果を受け取りWebサーバに渡すなど、Webサーバとintra-mart Service Platformのインタフェースとして機能します。複数のService Platformがある場合にはラウンドロビン機能により自動振り分け処理もおこないます。
Service Platform **：各種サービスのプラットフォーム**	intra-martの各種サービスを実行するプラットフォーム。
Application Runtime **：アプリケーション実行**	Web Server Connectorと連携してWebアプリケーションを実行するランタイム・サービスです。スタンドアロン型で動作させる場合、Application Runtime自身がHTTPサーバとしても機能します（Webサーバは必要ありません）。 Application Runtime機能を利用するには、Schedule Service以外のすべてのサービスが正しく動作していることが必要になります（Schedule Serviceはバッチ管理機能を利用する場合にのみ必要になります）。 ページベース開発モデルでは、プログラムファイル（プレゼンテーションページ（*.html）およびファンクションコンテナ（*.js））は、Resource Serviceに保管されており、必要時に配信され、Application Runtimeで実行されます。J2EEベース開発モデルでは、プログラムファイルはResource Serviceを利用せず、URLによりマッピングされたプログラム（*.jsp、*.xtp、*.class）をロードして実行します。 Application Runtimeは、JDBCを利用して複数のデータベースに接続

することもできます。また、構築するアプリケーションの負荷、クライアント数などに応じてApplicatio Runtimeなどを柔軟に増設することができます。複数のApplication Runtimeがある場合は、サーバトラブル発生時においても、並列に配置したApplication Runtimeが相互に補間することができます(セッションフェイルオーバー機能)。また、開発時には当サーバ内にHTTP Serverが内包されているので、別途Web Serverを用意する必要はありません。

Resource Service
:プログラム配布

intra-martシステムの動作に必要な各プレゼンテーションページ(*.html)およびファンクションコンテナ(*.js)のプログラムファイルを集中管理すると共に、Application Runtimeが必要なプログラムを自動的に配布します。
ページベース開発モデルのみで利用されるサービスです。

Shared Memory Service
:メモリへのデータ保管

データを一時的に保存しておく目的で利用される機能です。また、このサービスが提供する情報管理は、セッションレスです。タイムアウトなどによりデータが消失することはありません。これらの情報は、Shared Memory Serviceのメモリ中に保管され、プログラムの要求に応じて取り出したり削除したりします。

Permanent Data Service
:システム設定情報管理

intra-martシステムが動作するために必要な設定情報などを管理しています。
このサービスでは情報をファイル保存するので、サービスが停止した場合も、次の起動時に、以前に保存されたデータを復元することができます。したがって、簡易的なデータベースとして利用することも可能です。また、アカウントやメニューなどの、intra-martのシステム設定情報もこの機能を利用しています。
情報ファイルは、Permanent Data Serviceのインストールディレクトリ直下のtreasure/ディレクトリ内に保存されます。

Serialization Service
:システム全体の一意性保証

システム全体における統括的な利用を目的とした機能の集合体です。アプリケーションロック機能の管理が行われています。

Storage Service
:ファイルの一元管理

複数のApplication Runtimeを用いた分散システムでintra-martを利用しているときに、アップロードされたファイルやシステムで共有化したいファイル(主にデータファイル)を一元管理するときに利用します。その他、エクステンション・モジュールを用いる場合には、PDFファイルやXRFファイルの作成も、このサービスが受け持ちます。ファイルの保存場所は、Storage Serviceをインストールしたディレクトリ直下のstorage/以下になります。

Schedule Service
:バッチ実行スケジュール

バッチプログラムの実行スケジュールを管理します。
このサービスは、指定の時間になると該当するバッチプログラムの実行を指定URLを用いてApplication Runtimeに依頼します。したがって、バッチプログラムは、Application Runtimeで実行されます。

Server Manager
:動作状況のモニタリング

intra-martの各種サービス、Application Runtimeなどの動作状況を管理します。
標準で用意されている監視ツール「IM-Administrator」を経由して、intra-martの各種Serviceの状況監視や起動・停止を行うことができます。

intra-martフレームワークのシステムアーキテクチャー

以下に、intra-martフレームワークのシステムアーキテクチャーを示します。

Service Platform **：各種サービスのプラットフォーム**	intra−martの各種サービスを実行するプラットフォーム。
Web Application Server **：アプリケーション実行**	Web Applicationを実行可能なプラットフォームです。J2EEの規格に準拠しているサーバ製品をご利用ください（正式に対応している製品に関しては、製品に付属のリリースノートをご覧ください）。
Application Runtime **：アプリケーション実行**	Application Runtime機能を利用するには、Schedule Service以外のすべてのサービスが正しく動作していることが必要になります（Schedule Serviceはバッチ管理機能を利用する場合にのみ必要になります）。 ページベース開発モデルでは、プログラムファイル（プレゼンテーションページ（*.html）およびファンクションコンテナ（*.js））は、Resource Serviceに保管されており、必要時に配信され、Application Runtimeで実行されます。 J2EEベース開発モデルでは、プログラムファイルはResource Serviceを利用せず、URLによりマッピングされたプログラム（*.jsp、*.class）をロードして実行します。Application Runtimeは、JDBCを利用して複数のデータベースに接続することもできます。 また、構築するアプリケーションの負荷、クライアント数などに応じてApplicatio Runtimeなどを柔軟に増設することができます。
Resource Service **：プログラム配布**	intra-martシステムの動作に必要な各プレゼンテーションページ（*.html）およびファンクションコンテナ（*.js）のプログラムファイルを集中管理すると共に、Web Application Serverが必要なプログラムを自動的に配布します。 ページベース開発モデルのみで利用されるサービスです。
Shared Memory Service **：メモリへのデータ保管**	データを一時的に保存しておく目的で利用される機能です。 また、このサービスが提供する情報管理は、セッションレスです。タイムアウトなどによりデータが消失することはありません。これらの情報は、Shared Memory Serviceのメモリ中に保管され、プログラムの要求に応じて取り出したり削除したりします。

Permanent Data Service **：システム設定情報管理**	intra-martシステムが動作するために必要な設定情報などを管理しています。 このサービスでは情報をファイル保存するので、サービスが停止した場合も、次の起動時に、以前に保存されたデータを復元することができます。したがって、簡易的なデータベースとして利用することも可能です。また、アカウントやメニューなどの、intra-martのシステム設定情報もこの機能を利用しています。 情報ファイルは、Permanent Data Serviceのインストールディレクトリ直下のtreasure/ディレクトリ内に保存されます。
Serialization Service **：システム全体の一意性保証**	システム全体における統括的な利用を目的とした機能の集合体です。アプリケーションロック機能の管理が行われています。
Storage Service **：ファイルの一元管理**	複数のWeb Application Serverを用いた分散システムでintra-martを利用しているときに、アップロードされたファイルやシステムで共有化したいファイル（主にデータファイル）を一元管理するときに利用します。その他、エクステンション・モジュールを用いる場合には、PDFファイルやXRFファイルの作成も、このサービスが受け持ちます。ファイルの保存場所は、Storage Serviceをインストールしたディレクトリ直下のstorage/以下になります。
Schedule Service **：バッチ実行スケジュール**	バッチプログラムの実行スケジュールを管理します。 このサービスは、指定の時間になると該当するバッチプログラムの実行を指定URLを用いてWeb Application Serverに依頼します。したがって、バッチプログラムは、Web Application Serverで実行されます。
Server Manager **：動作状況のモニタリング**	intra-martの各種サービス、Application Runtimeなどの動作状況を管理します。 標準で用意されている監視ツール「IM-Administrator」を経由して、intra-martの各種Serviceの状況監視や起動・停止を行うことができます。

6 intra-martのその他の特徴

intra-martベースモジュール/フレームワークには、次のような優位性があります。

＊ 軽快なスピード環境
＊ 高度な信頼性
＊ 高いスケーラビリティ
＊ 運用の容易性
＊ ASP型アプリケーションの構築
＊ プラットフォームのオープン性

軽快なスピード環境

intra-martを利用したWebシステムは、パフォーマンスに優れたシステムとなります。

＊ JavaVM上でintra-martは動作するために完全なマルチスレッドに対応しています。intra-martベースモジュール/フレームワークの実行エンジンは100%Javaで記述されています。
＊ ダイナミックロードコネクションプール機能（動的な負荷連動接続プール機能）などの採用により、高速かつサーバリソース負担のない環境が実現できます。
＊ ページベース開発モデルのファンクションコンテナ（JavaScriptファイル）は、実行時にJavaのクラスへ自動コンパイルされるので高速です。
＊ intra-martベースモジュールのServlet/EJB/JSPエンジンは、高速Java技術を持つ米CauchoTechnology社のResinを組み込んでいます。

intra-mart+内包HTTPサーバ ：Webサーバに内包のHTTPサーバを利用した場合
intra-mart+Apache ：WebサーバにApacheを利用した場合

高度な信頼性

intra-martベースモジュールでは、Application Runtime上にユーザごとのセッション情報が管理されます。Application Runtimeが万一ダウンしても他のApplication Runtimeがセッション情報を引き継ぐため、24時間365日のノンストップのシステムアーキテクチャを実現することができます。また、Webサーバをクラスタ化した下記のような構成にすることでさらに信頼性をあげることができます。

〈Webをベースにした多階層アーキテクチャ〉

クライアントのセッション情報がセッションフェイルオーバー機能により値が引き継がれるため、Application Runtimeがダウンした際も消失することがありません。

また、intra-martのService Platformにはフォールトトレランス機能が備わっています。これにより分散システム構成にした場合、Service Platformを複数台用意して各サービスを2重化させることにより、障害に対して堅牢なシステム構築を可能にしています。

Column

HttpSessionのフェールオーバー

Application Runtimeを複数立てた場合、HttpSession（クライアントのログインセッション情報）をフェールオーバーさせることができます。この機能は、intra-martベースモジュールで利用できる機能で、intra-martフレームワークでは利用できません（各アプリケーションサーバの機能に依存します）。
ベースモジュールでのセッションフェイルオーバーは、以下の2方式となります。
詳細は、設定ガイドの「HttpSessionのフォールトトレランス設定方法」を参照してください。

● メモリtoメモリ方式

メリット　Application Runtimeの設定だけで構築可能。メモリto RDBと比較して、処理負担がない。

デメリット　バックアップ関係にある2台が両方ともダウンするとセッションが消える。
メモリ to RDB方式に比べて設定が難。他のApplication Runtimeのメモリを保存するため、Application Runtimeのメモリ消費量が増える。

● メモリto RDB方式

メリット　Application Runtimeが何台ダウンしても最低1台残ればセッションの継続が可能。
メモリ to メモリ方式に比べて設定が容易。

デメリット　DBが必要。

市販のクラスタ製品を利用してラウンドロビン構成で運用する場合には、メモリ to RDBを推奨します。より堅牢なシステム構成にするには、メモリ to RDB方式を推奨します。

高いスケーラビリティ

intra-martは多階層分散アーキテクチャに対応しており、スケーラビリティに優れています。最初は1台のWindowsサーバからスタートし、利用者が増えるにつれてDBサーバと分割した3階層アーキテクチャに変更したり、UNIX機に置き換えたりできます。
intra-martベースモジュールはWindows版とUNIX/Linux版が用意されています。Webサーバ、アプリケーションサーバ、データベースサーバは、それぞれ複数増設することができるので、負荷に応じて拡張することができます。複数設置すると、それぞれが補間することでトラブルに対しても効果を発揮します。また、1トランザクションで複数データベースへのアクセスをサポートできるため、ユーザ既存システムとの親和性が高いシステム構築が可能になります。

● 最小構成システム例

● 各サーバを独立した構成例

● Application Runtimeを複数設置した構成例

● 最大設置可能サーバ数

クライアント	webサーバ	Application Runtime	データベースサーバ	Server Manager	etc Service
n台	n台	n台	n台	1台	各2台

〈システム構成拡張例：intra-martベースモジュールの場合〉

運用の容易性

Application Runtime（intra-martベースモジュールの場合）やWebアプリケーションサーバ（intra-martフレームワークの場合）を複数設置して稼動される大規模システムで問題となるのは、ユーザアプリケーションの配布管理です。ページベース開発モデルでは、Resource Serviceが各アプリケーションサーバへのユーザアプリケーションの配布を自動で行うため、Resource Service上でのみユーザアプリケーションを最新に保てばよいことになり、運用の負担がかかりません。ただし、J2EEベース開発モデル（JSP、Servlet、EJBコンポーネント）の場合は、intra-martベースモジュールでは各Application Runtime上に、intra-martフレームワークではWebアプリケーションサーバ上に配置する必要があります。
Application RuntimeやWebアプリケーションサーバの新規増設時にも、ネットワークにマシンを接続し、intra-martを起動するだけで最新の稼働環境になります。
また、ブラウザに送信する文字コードをintra-martアプリケーションサーバ上で指定できるため、Windows、UNIXやiモードなどの機種混在時にも文字化けがおこりません。
さらに、監視ツールであるIM-Administrator（標準装備）により、intra-mart各種サーバの稼動情報をビジュアルに表示でき、各サーバの状態監視や設定変更、自動復旧の設定が遠隔で可能になります。
詳細についてはアドミニストレータガイドの第1章「8 IM-Administrator」を参照してください。

〈システム構成拡張例：intra-martベースモジュールの場合〉

IM-Administrator

監視サーバ(Server Managerに含まれる機能)によって収集されたintra-martの各種サービスの稼動情報をビジュアルに表示する監視ツール(標準装備)です。各サーバの状態の監視や設定の変更、自動復旧の設定が遠隔で行えます。

〈IM-Administrator(監視ツール)の基本画面〉

ASP型アプリケーションの構築

ログインするユーザグループごとに、ロール、ユーザ、メニュー、リンク、メッセージそしてデータベースを異なる設定で利用することができます。この機能を利用すると、ひとつのアプリケーションを複数の会社で共同利用するようなASP型（マルチシェア型）のアプリケーションを設定だけで簡単に構築することができます。

intra-martのすべての設定は、グループや会社ごとに個別に設定することができ、それぞれのグループが個別のintra-martシステムを利用しているように使用することができます。また、マルチデータベース機能と合わせて使用すると、それぞれのグループで別のデータベースサーバを利用することができます。マルチグループ機能を設定した場合は、ログイン画面にマルチグループIDを入力する項目が表示されるようになります。

多言語機能を標準で提供

標準で日本語、英語、中国語の3ヶ国語に対応しており、ログインユーザ毎に言語の切り替えができます。海外拠点の現地社員との情報共有が容易に実現できます。

プラットフォームのオープン性

intra-martベースモジュール/フレームワークの実行エンジンは100% Javaで開発されていますので、サーバOSとして、Windows、Solaris、Linuxなど幅広く対応しています。Webサーバとしても、IIS、Apacheから選択できます。さらに、データベースも現在、Oracle、SQL Server、DB2、PostgreSQLにも対応しており、ブラウザもMicrosoft InternetExplorerとNetscapeから選択できます。
各ミドルウェアの対応バージョンなどの詳細や最新情報については、ホームページにて確認してください（http://www.intra-mart.jp）。

7 システム仕様

以下に、intra-martベースモジュール/フレームワークVer5.0の仕様を説明します。

パッケージ別機能一覧

intra-martは、パッケージの種類により使える機能が異なります。ご購入いただいたライセンスをご確認の上、次の表をご覧ください。
なお、表に掲載されていない機能に関しては、すべてのパッケージでご利用可能です。

機能	スタンダード	アドバンスド
アクセスセキュリティ機能	○	○
バッチ機能	○	○
ページベース開発モデルによるアプリケーション構築	○	○
J2EEベース開発モデルによるアプリケーション構築	○	○
ドキュメントワークフロー	○	○
ドキュメントワークフローのおける分岐ルートの作成（*1）		○
ビジネスプロセスワークフロー		○
EJBサーバ機能（*2）		○
EJB連携機能		○

*1　スタンダード版でもワークフローデザイナ（別売）を追加することにより分岐ルートを作成可能になります。

*2　EJBサーバ機能は、BaseModuleのみの機能です。また、フレームワークの場合は、ご利用中のWebApplicationサーバが提供する機能になります。

機能	スタンダード	アドバンスド	フレームワーク
im-Administrator	○	○	○
Windows環境でのサービス化（*1）	○	○	○
スタンドアロン型サーバ	○	○	○
ネットワークサーバ型による分散システム構築	○	○	○
intra-mart HTTP Server機能	○	○	
市販のWebサーバ製品を利用するためのWebServerConnector	○	○	
EJBサーバ機能（*2）		○	

*1　フレームワークの場合、Application-Runtimeが動作する環境では、ご利用のWebアプリケーションサーバの機能になります。

*2　EJBサーバ機能は、アドバンスド版のみの機能です。

*3　EJBコンテナを実行できるのは、Application-Runtimeの動作しているService-Platformのみです。他のサーバでEJBコンテナを動作させることはできません。このEJBコンテナはJ2EE1.3の仕様を実装しています。ただし以下の機能は利用できません。

・JavaMail 1.2（別途インストールすることにより利用可能）
・JAF 1.0（別途インストールすることにより利用可能）
・EJBのRMIサポート
・JCA 1.0
・earファイルによるデプロイ（J2EEアプリケーションのデプロイはマニュアル操作により可能）

対応規格一覧

本製品に含まれる各機能は、以下の規格に準拠しています。

機能	対応規格
J2EE	1.3
HTML	4.0
JavaScript(Server Side)	1.2
メール送信機能	SMTP(RFC821,2045-2049)
Document Object Model	Level2

システム要件

サーバ要件

以降に、各OS別のシステム要件を記載します。
なお、記載内容は目安であり、サーバの運用状況により常に一定のパフォーマンスを発揮できるとは限りません。

OS	Windows 2000(SP4以上を推奨) Windows Server2003Enterpirse Edition Solaris 9 Redhat Linux Advanced Server3 HP-UX11i AIX 5.2	
ハードウェア	Windows	PC/AT互換機(CPUの動作クロックが1GHz以上を推奨)
	Solaris	Solaris9の稼動可能なSparcアーキテクチャマシン機種 (Ultra SparcII 650MHz以上を推奨)
	Linux	Redhat Linux Advanced Server3の稼動可能なPC (CPUの動作クロックが1GHz以上を推奨)
	HP	HP-UX11iの稼動可能なPC (CPUの動作クロックが1GHz以上を推奨)
	AIX	AIX5.2の稼動可能なPC (CPUの動作クロックが1GHz以上を推奨)
メモリ	256MB以上のRAM(OSやその他のソフトウェアが動作するために必要なメモリが別途必要です) 512MB以上を推奨 ※注意:作成されるアプリケーションで多くのメモリを必要とする処理を想定されている場合や、想定ユーザー数等が多い場合は、より多くのメモリを必要とします。この点を御考慮いただき必要となるメモリ容量を算出してください。	
ハードディスク	200MB(intra-martベースモジュール/フレームワークのみ) ※注意:実行するシステムの規模や登録データ量により運用時に必要なハードディスク容量は変化します。また、Storage ServiceおよびPermanent-Data Serviceは、管理するデータ量に比例してハードディスク容量を必要とします。ログ機能の利用や長期間の運用も考慮し、十分な空き容量を確保してください。	
ソフトウェア	環境要件を参照してください。	

環境要件

intra-martベースモジュールver.5.0およびintra-martフレームワークver.5.0の大部分はJavaベースで開発されています。従って、JDK1.4.2が動作するプラットフォームであればどこでも動作可能です（文字エンコーディングやファイルアクセスに関してはOSその他の影響を受けることがあります）。intra-martベースモジュールver.5.0およびintra-martフレームワークver.5.0は下記環境をサポートしています。

■サーバ（Server-ManagerおよびService-Platform）

Sun JDK1.4.2相当の環境に対応しています。なお、動作可能な対応OSは下記のとおりです。

- Windows2000 Server
- Windows2003 Enterprise Edition
- Solaris 9
- Redhat Linux Advanced Server3
- HP-UX11i
- AIX 5.2

■Application Runtime（アプリケーションプログラムの実行エンジン）

下記表を参考にしてください。

Webサーバ製品	OS
intra-mart HTTP Server	Windows2000／2003 Solaris 9 Redhat Linux Advanced Server3 AIX 5. 2 HP-UX11i
IBM WebSphere Application Server 5.1	Windows2000／2003 Solaris 9
BEA Weblogic Server 8.1J	Windows2000／2003 Solaris 9 AIX 5. 2
JBoss4	Redhat Linux Advanced Server3

注意1）　intra-mart HTTP Serverは、ベースモジュールの機能です。

注意2）　IBM WebSphere Application Serverと連携できるのはintra-martフレームワークのみです。

注意3）　BEA Weblogic Serverと連携できるのはintra-martフレームワークのみです。

■Web Server Connector（ベースモジュールのみ）

WebServerConnectorはJavaベースではなく、環境に依存した実行可能形式ファイルでの運用となりますので、ご利用になられる実行環境に合わせてソースファイルをコンパイルしてご利用ください（弊社においては、下記表に示す環境で動作を確認しております）。下記表を参考にしてください。

	Webサーバ製品	接続形態
Windows2003（＊1）	IIS5.0	ISAPI
	Apache2.0.47　（DSOサポート版）	組み込みモジュール（mod）
Windows2000	Apache2.0.47　（DSOサポート版）	組み込みモジュール（mod）
Solaris9	Apache2.0.47　（DSOサポート版）	組み込みモジュール（mod）
Redhat Linux Advanced Server3	Apache2.0.47　（DSOサポート版）	組み込みモジュール（mod）

（※1）　Windows2000は、コンパイル済みのファイルが提供されています。

■im-Administratorおよび外部ソフトウェア接続モジュール（intra-martが提供する独自API）

Sun JRE1.4.1相当の環境に対応しています。ただし、im-Administratorはウィンドウを有する対話型インタフェースとなっていますので、ウィンドウを表示できる環境に限られます。

クライアント要件

クライアントとして対応する主なユーザインタフェースはWebブラウザです。なお、製品標準の画面を利用せずに独自のアプリケーションを構築した場合は、この限りではありません(ただし、一部機能にブラウザの制限を受けるものがあります)。

・ Netscape 7.1以上
・ Microsoft Internet Explorer 6sp1以上のWindows 2000/XP版ブラウザ
＊ Netscape 6.xは対応しておりません。

但し、以下のJava Appletを利用した画面はMicrosoft Internet Explorer 6sp1以上または、Netscape 7.1以上である必要があります。Microsoft Internet Explorerをご使用の場合、Java Plug-in 1.4.1(Sun JRE1.4.1)以上が必要です。

弊社動作環境済み(推奨)Java Plug-inのVersionは1.4.1_05です。

・[ワークフロー]-[マスタメンテナンス]-[プロセスメンテナンス]のプロセス編集画面
・[ワークフロー]-[マスタメンテナンス]-[プロセス一覧]のプロセス確認画面
・[ワークフロー]-[タスク一覧]のプロセス確認画面
・[ワークフロー]-[代理設定一覧]のプロセス確認画面
・[ワークフロー]-[タスク一覧]のドキュメントワークフロー承認処理時の未承認案件一覧プロセス確認画面

その他の要件

特殊な用途として利用される一部のモジュールは上記とは別に下記要件となります。

モジュール	対応環境
Windows用サービス化プログラム	Windows2000/ Windows2003
外部ソフトウェア接続モジュール	JRE1.4.2

検証済み環境

弊社動作検証済み(推奨)のintra-mart製品と他社ミドルウェア製品との組み合わせを記します。弊社では以下の環境をサポートしますが、intra-mart製品以外(OS、JDK、DB、Mailサーバ等)に起因する問題が発生した場合は保証致しかねます。

■Service PlatformおよびServer Manager

サーバOS	ミドルウェア
Windows2000Server(SP4)	SUN JDK1.4.2_08
Windows2003 Enterprise Edition	
Solaris 9(SPARC版)	
Redhat Linux Advanced Server3.2.3-47	SUN JDK1.4.2_07
HP_UX11i	HP-UX for the Java2 Platform SDK 1.4.2.01
AIX 5.2	IBM 32-bit SDK for AIX, Java 2 Technology Edition, Version 1.4.2

注意1) intra-mart フレームワークの場合、Application Runtimeの動作する環境は、Webアプリケーションサーバ製品の仕様に依存します。

■Web Server Connector（ベースモジュールのみ）

サーバOS（*1）	ミドルウェア
Windows2003	Microsoft IIS 6.0（*）
	Apache2.0.47（DSOサポート版）
Windows2000 Server（SP4）	Apache2.0.47（DSOサポート版）
Solaris 9	Apache2.0.47（DSOサポート版） Apache2.0.47（DSOサポート版）
Redhat Linux Advanced Server3.2.3-47	Apache2.0.47（DSOサポート版）

■Application Runtime

intra-martベースモジュールVer5.0およびintra-martフレームワークVer5.0のApplication Runtimeは、以下のWeb Applicationサーバ上で動作します。

intra-mart製品	Web Application サーバ
intra-martフレームワーク	IBM WebSphere Application Server Ver.6.0.0.3
	BEA Weblogic Server 8.1J（SP4）
	JBoss 4.0.1Sp1
intra-martベースモジュール	intra-mart HTTP serverVer5.0 （Application Runtimeの動作するService Platformに付属）

■データベース

データベースに関しては、以下の環境にて動作が確認されています。

No	Web Application サーバ	サーバOS	データベース（*1）	サーバOS
1	intra-mart HTTP server Ver.5.0（Application Runtime の動作する Service Platformに付属）	Windows 2003 Enterprise Edition	MicrosoftSQL Server 2000	Windows2000（SP4）
2	intra-mart HTTP server Ver.5.0（Application Runtime の動作する Service Platformに付属）	Windows2000（SP4）	IBM DB2 ver8.1	Windows2000（SP4）
3	intra-mart HTTP server Ver.5.0（Application Runtime の動作する Service Platformに付属）	HP_UX11i	Oracle10g	Windows2000（SP4）
4	intra-mart HTTP server Ver.5.0（Application Runtime の動作する Service Platformに付属）	Redhat Linux Advanced Server 3.2.3-47	PostgreSQL8.0	Redhat Linux Advanced Server 3.2.3-47
5	IBM WebSphere Application Server Ver.6.0.0.3	AIX 5.2	IBM DB2 ver8.1	Windows2000（SP4）

6	IBM WebSphere Application Server Ver.6.0.0.3	Windows 2003 Enterprise Edition	Oracle10g	Windows2000 (SP4)
7	BEA Weblogic Server 8.1J(SP4)	Solaris9	Oracle10g	Windows2000 (SP4)
8	JBoss 4.0.1SP1	Redhat Linux Advanced Server 3.2.3-47	PostgreSQL8.0	Redhat Linux Advanced Server 3.2.3-47

データベースへの接続方法に関しては、下表をご覧ください。

データベース製品	接続方法	接続に必要なソフトウェア
Oracle 8.1.7以上	JDBC(Type4 非XA)	Oracle JDBC Driver (version9.2) WebLogic Type 4 JDBC ドライバ(intra-mart フレームワーク+WebLogicの場合のみ)
MS-SQL Server 2000	JDBC(Type4 非XA)	Microsoft SQL Server 2000 Driver for JDBC Version 2.2.0037 WebLogic Type 4 JDBC ドライバ(intra-mart フレームワーク+WebLogicの場合のみ)
IBM DB2 UDB	JDBC(Type2 非XA)	DB2 JDBC Driver (IBM DB2 ver8.1に付属)
postgresql-8.0.0	JDBC(Type4 非XA)	postgresql-8.0-310.jdbc3

※1 IBM DB2へは、IBM DB2 Ver8.1に付属するJDBCドライバで接続します。
※2 intra-martの提供する画面等は、XAドライバには対応しておりません。
※3 IBM DB2へは、IBM DB2 Ver8.1に付属するJDBCドライバで接続します。
※4 IBM DB2をご利用の際、IBMがLinuxにおいてサポートするJDKは、SUN JDKではなく、IBM JDK 1.3.1以降となります。
http://www.ibm.com/software/data/db2/udb/ad/v8/bldg/r0006776.htm(2005年5月31日現在)

■クライアント

クライアント関しては、以下の環境にて動作が確認されています（動作検証済バージョンのブラウザを御推奨いたします）。

- Windows2000 ＋ Netscape 7.1
- Windows2000 ＋ Microsoft Internet Explorer 6.0(SP1)
- WindowsXP　＋ Netscape 7.1
- WindowsXP　＋ Microsoft Internet Explorer 6.0(SP1)

なお、ワークフローについては、下記環境にて動作が確認されています（Microsoft Internet Explorerをクライアントとして利用した場合）。

- Windows2000 ＋ Microsoft Internet Explorer 6.0(SP1) ＋ Java Plug-in 1.4.1_05
- WindowsXP　＋ Microsoft Internet Explorer 6.0(SP1) ＋ Java Plug-in 1.4.1_05

■アクセスセキュリティ認証用LDAPサーバ

- Sun Java System Directory Server 5.2
- Microsoft Windows Server 2003 Active Directory
- Microsoft Windows Server 2000 Active Directory

注意事項

- 弊社では、この本ドキュメントに記載されている『検証済み環境』にて試験を実施しております。
- Web サーバおよびデータベース、その他のモジュール稼動に必要な稼働環境情報（リソースやメモリ等）につきましては、各製品（各アプリケーション）の Web サイト等でご確認ください。
- intra-mart ベースモジュールまたは intra-mart フレームワークを稼動させるために必要なメモリ量をインストール時に適切に設定してください。なお、インストール後に設定を変更することもできますので、環境及び利用状況に合わせて適宜チューニングを実施してください。
- intra-mart の各サーバ（Server Manager および Service Platform）、im-Administrator が稼動するために必要な JavaVM は別途入手してください。
- データベース接続をするために必要となる JDBC ドライバは各ベンダより提供されているものをご利用ください。

● システム仕様の詳細、制限事項に関しては、製品に同梱されているリリースノートを必ず参照してください。

8 intra-mart「eBuilder Ver5.0」

J2EEベースでの開発に最適な開発ツールには、オープンソースの統合開発環境である「eclipse」に対するプラグインとしても利用できるようになった「eBuilder Framework Producer」も用意されていて、煩雑なJ2EE開発の効率を大きく向上させることができます。既存のデータベースからSQLを発行する段階から、画面および画面遷移の作成に至るまで、J2EEフレームワークを利用したトータルな開発環境を提供しています。これにより、高い生産性を得ることができるようになりました。

- intra-martベースモジュール/フレームワークには、intra-mart 「eBuilder Ver5.0」の試用版が同梱されています。ぜひ、一度体験することをお勧めいたします。

豊富な開発支援機能を搭載 PageProducer

ページベース開発モデル用のPageProducerには、さまざまな開発支援機能が搭載されており、HTML、JavaScriptを利用したプレゼンテーションページとファンクションコンテナによる開発効率を大きく向上させるツールです。

- ＜IMART＞タグや intra-martAPI、タグライブラリなどを呼び込んで、ソースに貼り込める。
- スペルチェックや文法チェック、デバックツールも提供。
- テンプレートやウィザードも提供。
- ブレークポイントによる実行時とレース機能やインスペクト機能も搭載。（変数の内容参照）

Eclipse（J2EE開発環境）へのプラグインも可能FrameworkProducer

J2EEベースでの開発に最適な開発ツールです。オープンソースのIDE（統合開発環境）である“Eclipse”にプラグインでインストールして、intra-mart J2EE Frameworkで使用される設定ファイル（プロパティファイル群）をGUIツールによって自動的に作成したり、スケルトンコードを自動生成することができます。J2EEフレームワークの編集がビジュアルに可能となるオーサリングツールが提供されることで、煩雑なJ2EE開発の効率を大きく向上させることができます。

FrameworkProducerのソースコード自動生成、ビジュアルな画面定義機能

FrameworkProducerには、画面関連のコンポーネント、業務ロジック、DB関連のコンポーネントなどを自動生成する機能が標準で提供されています。各編集画面から必要な項目などを入力するだけで、J2EEのソースコードを生成することが可能です。生成されたソースコードはそのまま intra-martのim-J2EE Framework上で稼動するコンポーネントとして利用することができるようになるため、Javaのコーディング経験の浅い開発者でも、Webシステム開発を容易に、効率的に進めていくことができます。

- 「eBuilder Framework Producer」に関しては、「Framework Producer 操作ガイド」がインストールCDの下記に用意されています。
  document/Application Development/presentation_framework_tutorial.pdf

Column

ソースコードの自動生成～実行までの利用手順

1

コンポーネントの作成

■SQLビルダによるDAOの作成

データフレームワーク上で動作するコンポーネントを作成します。こちらではデータベース(DB)構造を定義するもので、市販のDB設計ツールからのデータ取り込みも可能です。

用意された編集画面上で、DBの項目名やタイプなどのテーブル情報を登録し、データベース・クエリをビジュアルに定義することで、DBへのアクセスを行うクラスを自動生成できます。

■業務ロジックの作成

イベントフレームワーク上で動作するコンポーネントを作成します。システム内で利用される業務ロジックをeBuilder上で作成し、部品化していきます。

2

画面、画面遷移の作成とコンポーネントとの関連付け(フロー)

上記で作成された各コンポーネントを編集画面からビジュアルに呼び出し、コンポーネント同士の呼び出しを定義していきます。ここからプロパティファイルを自動出力します。

3

ソースコード完成したコンポーネントをJ2EEフレームワーク上に配置・実行編集・追加

eBuilderのソースコード自動生成機能を利用することで、開発すべき全体ソースコードの70～80%が自動生成されるため、従来よりも開発効率が大幅に向上いたします。

4

クライアント

クライアントサイドはHTMLやリッチクライアントツールとの連携など、自由度の高い組み合わせが可能です。

9 旧バージョンとの互換性（Ver4.x→Ver5.x）

intra-mart Ver5.xには、Ver4.xで作成したアプリケーションとの互換性を保つための仕組みが用意されています。

APIの互換性

APIの互換性に関しては、一部のAPIを除き両Ver間での互換性が保たれています。

- pp共通Mのトリガ機能APIは別のAPIに置き換わりますのでご注意ください。
- ドキュメントワークフローおよびビジネスプロセスワークフローのAPIについて、一部互換されないものがあります。
- リンクセキュリティ機能（<IMART type="secure">）は互換されません。
- 詳しくは、リリースノートおよびAPIリストを参照してください。

データベース構造の互換性

データベース構造は一部変更があるため、「データ移行ツール」が提供されています。

データベースアクセス

アプリケーション共通マスタでは、Ver4.3互換アクセス機能を用意しています。これにより、既存アプリケーションからのデータベースアクセスが可能となります。

10 Standard版とAdvance版の相違

intra-martベースモジュールには、Standard版とAdvance版の2種類が用意されています。

Standard版

Advance版

Stabdard版とAdvance版の機能比較

intra-mart Ver5.xでは、standard版とadvance版とで以下のように相違があります。

No.	区分	備考	Std.	Adv.
1	他社フレームワークへの対応	WebSphere、Weblogic、JBOSS対応	○	○
2	リッチクライアント対応（Curl）	リッチクライアントへの対応	○	○
3	オープンソース対応	ソース公開	○	○
4	ワークフロー	ただしワークフローデザイナーはStandard版にはオプション	○	○
5	ビジネスプロセスワークフロー	Advance版にバンドル	－	○
6	ERP連携モジュール	SAPのリアルタイムAPI	－	○
7	SOAアーキテクチャー	IM-Σサーブ	－	○
8	EJBコンテナ	EJBコンテナの組み込み	－	○

（参考）OEMプロダクツ一覧

intra-martベースモジュール/フレームワーク上で動作するアプリケーション群が、intra-mart特約店・パートナー各社からリリースされていますので、参考にしてください。

製品名称	補足	開発元特約店名
らくらく稟議Express	文書管理	NTTデータ四国
Web購買 EPICUS	電子調達	西華産業
Property Manager	不動産業界向けERP	三井情報開発
プロジェクト別稼動管理	プロジェクト原価集計システム	NTTデータ関西
ITAMワークフロー	資産管理ワークフロー	コア
評価三昧	成果型人事評価	アスネット
超勤怠	外食・小売チェーンなどのリアル勤怠	NSD
Webワークフロー	ワークフロー申請帳票集	ダイヤモンドコンピュータサービス
Kizuki Enterprise Portal	企業ポータル	エー・アンド・アイ システム
JointBase	インターネット帳票配信	プリズム
営業PROナビ	コーチング型営業支援	アスクラボ
ナレッジクリエータ	ナレッジマネジメントシステム	システムクリエートセンター
CyberVine	Eラーニング	TIS
SCAW Frontizum	ワークフロー申請帳票集	NTTデータシステムズ
BusinessView	CTIコールセンター	NECネクサソリューションズ
QuickBinder	汎用文書管理システム	日本アドバンテージ
FX証憑ソリューション	複合機連携ソリューション	富士ゼロックス
三菱総合研究所 個人情報管理システム	個人情報管理ソリューション	NTTデータイントラマート

- 詳細および最新の情報に関しては、弊社Webサイトの「Partner´s Solution」のページをご覧ください。
  http://www.intra-mart.jp/products/solution/solution.html

B

BtoB ソリューション 2

C

CRM ソリューション 2

E

eBuilder 68
ERP 連携モジュール 16

I

IM-Administrator 60
im-BizAPI 8
IM-Curl 30
IM-FComponents 31
IM-FormatCreator 38
IM-PDF デザイナー 32
IM-XRF コンポーザ 33
IM-SecureSignOn 33
IM-ビジネスプロセスワークフロー 35
IM-ワークフローデザイナー 35
intra-mart の特長 3
i モード unit 28

J

J2EE ベース開発モデル 4
Java 業務コンポーネント群 9
Java コンポーネント 8

S

Struts 48

V

ViewCreator 24

W

WebMail 2
Web 受発注システム for SellSide 2
Web 受発注システム for SellSide 2

あ

アクセスセキュリティ・モジュール 17
アプリケーション共通マスタ unit 24
アプリケーション共通モジュール 14
イントラネットスタートパック 2
イントラネットソリューション 1
営業支援システム 2
エクステンションモジュール 30
パブリックグループ 25

か

外部ソフトウェア接続モジュール 16
画面共通モジュール 12
カレンダ unit 26
業務基盤ツール 17
グラフ描画モジュール 13

さ

システムアーキテクチャー 51
システム仕様 62
システム仕様 62
人事管理システム 2
申請者ルート作成ワークフロー 34
全体構成 1

た

帳票印刷モジュール 13
ツリー表示 unit 27
統合検索ソリューション 40

は

バッチ管理モジュール 22
ビジネスロジック層 14
ファイルアップロード unit 26
ファイルダウンロード unit 26
ファンクションコンテナ 45
プライベートグループ 25
フレームワーク 1
プレゼンテーションページ 44
プロジェクトリソース管理システム 2
プロトタイプウイザード 23
文書管理システム 2
ページベース開発モデル 5
ベースモジュール 1
ポータルモジュール 22

ま

マイメニュー 19
マルチデバイスソリューション 41
メール連携モジュール 15

や

ユーザインタフェース層 10
ユニット 24

わ

ワークフロー・モジュール 20

intra-mart The Power for Enterprise Web Solution

ベースモジュール/フレームワーク・マニュアル　Ver. 5.0　チュートリアル・ガイド

2005年 11月 2版

株式会社 NTTデータ イントラマート

〒107-0052　東京都港区赤坂3-21-16 SKI赤坂ビル
TEL(03)5549-2821　FAX(03)5549-2816
E-mail：info@intra-mart.jp
ホームページ：http://www.intra-mart.jp
Copyright 2000-2005 株式会社NTTデータ イントラマート All rights Reserved.

※本マニュアルに記載されている社名および商品名は、一般に各社の商標および登録商標です。