



Popular Framework 4

製品概要

2011年12月18日

ポピュラーソフト株式会社

概要

- Popular Framework 4(以下略称PF4)は、最新の技術で新たな開発したiPhone/iPad/Androidを対応、HTML5対応する次世代総合RIA(Rich Internet Applications)WEBシステム開発基盤です。

クライアント(WEB画面)は、従来のASP.NETとJSPを利用してないため、.NETサーバとJAVAサーバに両方対応します。

PF4は、単なる開発ツールではなく、ワークフローモジュール、共通マスタと共通画面など共通機能を標準装備しています。各種業務アプリケーションや既存基幹システムとの柔軟な連携が可能です。SaaSとクラウドを対応します。

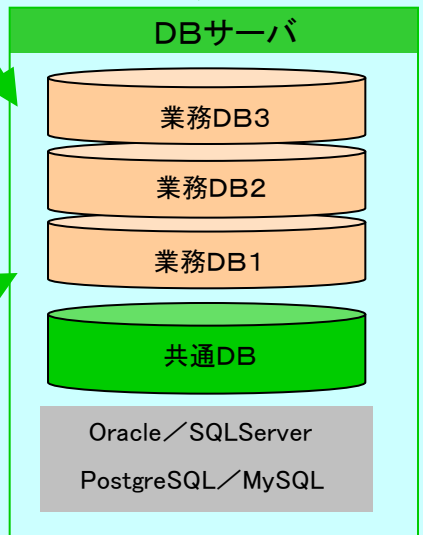
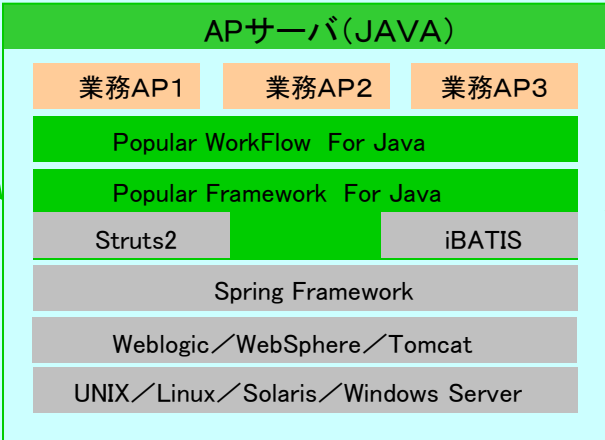
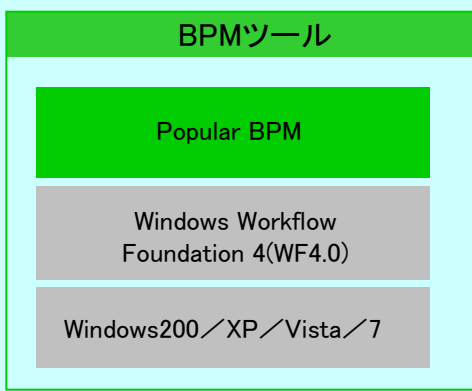
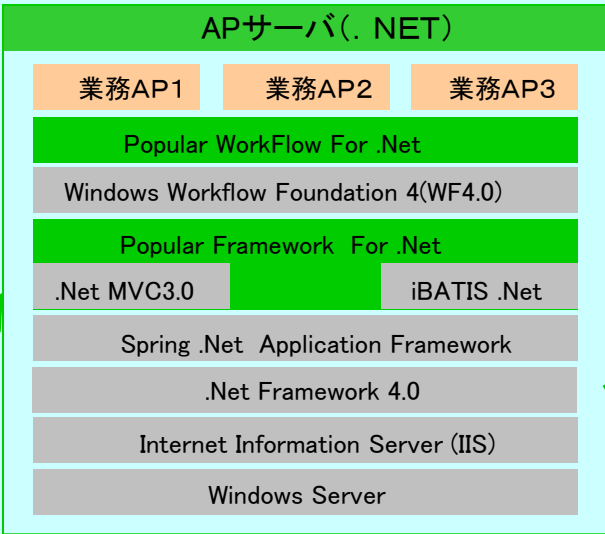
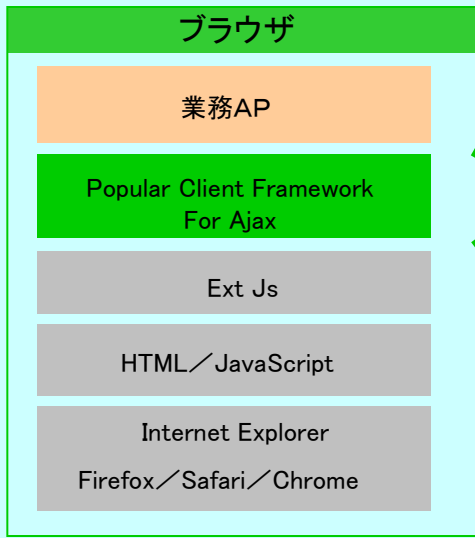
特長

- ・ SaaS／クラウド対応
- ・ 独立性と拡張性高いワークフロー機能
- ・ 共通マスタと共通画面を提供
- ・ 自由定義できるメニュー、ツールバー、ポータル機能
- ・ WEBシステムが実現難しい権限制御機能。
- ・ ノーコーディングで画面入力チェック、排他などを実現
- ・ RIA高い操作性インタフェース
- ・ iPhone、iPad、Androidタッチ端末対応

Popular Framework 4構成図



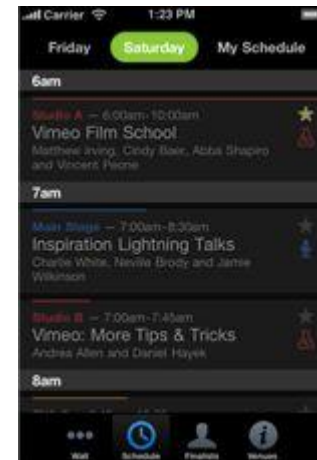
iPhone/iPad/Android対応 Windows200/XP/Vista/7対応



iPhone、iPad、Android対応

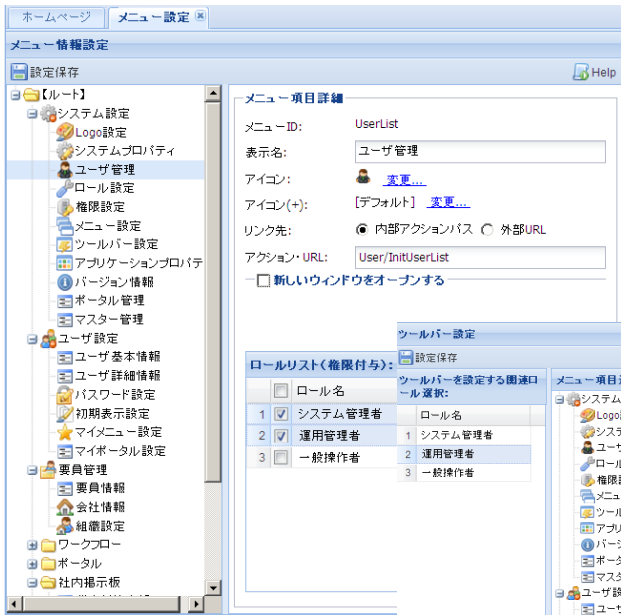
Powered By SenchaTouch

iPhoneおよびAndroidのようなタッチ端末で利用できるWebアプリ、それもネイティブアプリのような操作性を持つWebアプリを開発できます。

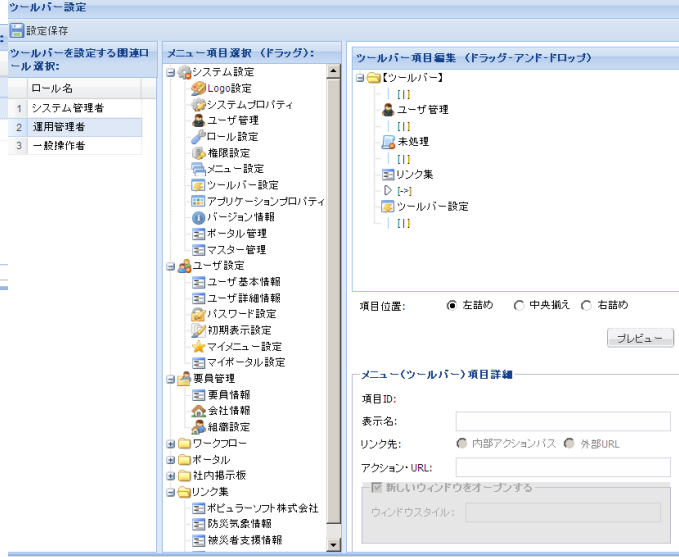
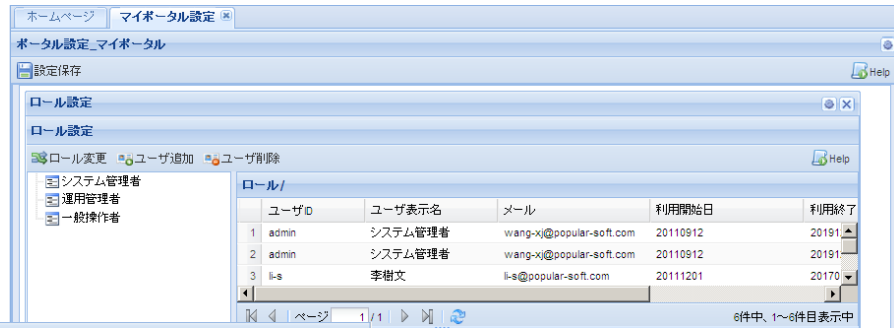


メニュー設定、ポータル機能

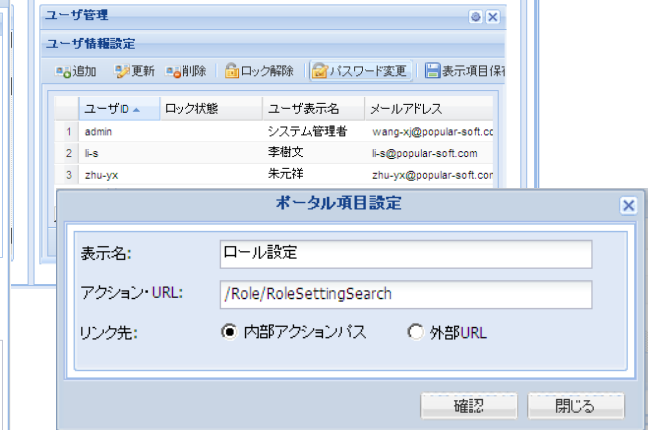
自由定義できるメニュー、マイメニュー、ツールバー、ポータル機能。



メニュー設定



ツールバー設定



ポータル設定

RIA (Rich Internet Application) と言われる高い操作性のWebブラウザ上で動作するアプリケーションの開発ができます。

The screenshot displays a web application interface with several components:

- Task List:** A table with columns for System Case ID, Receipt Number, Flow Group, Flow Definition Name, and Case Name. It lists four tasks with IDs 1 through 4.
- User Update Form:** A form titled 'ユーザ更新' (User Update) with fields for User ID, Name (Kanji/Romanji), Surname (Romanji), Name (Kana), Start/End Dates, and Contact Information (Phone, FAX, Email).
- Calendar:** A calendar for February 2011 showing the current date as the 24th.
- Navigation and Controls:** Includes buttons for '編集' (Edit), '差戻し' (Return), '取消' (Cancel), '詳細' (Details), '設定保存' (Save Settings), '出力' (Output), and '検索入力' (Search Input).

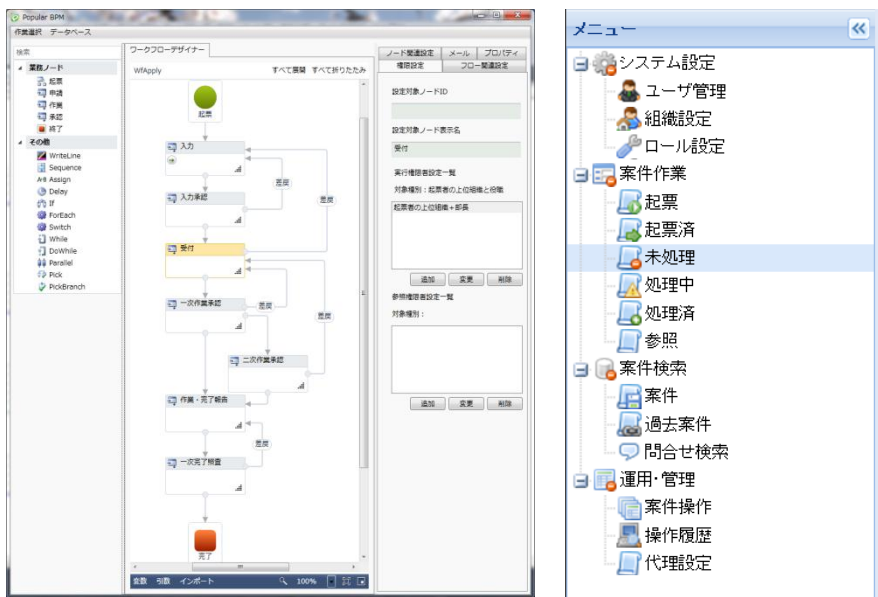
サポートブラウザ:
Internet Explorer 6+
FireFox 1.5+ (PC, Mac)
Chrome 3+
Safari 3+
Opera 9+ (PC, Mac)

※Ext jsはSencha Inc社登録商標です。

ワークフローソリューション

Powered By Windows Workflow Foundation

MS最新のワークフローエンジンのベースで、新たな開発した独立性が強いワークフローです。各種業務アプリケーションや既存基幹システムとの柔軟な連携、全社規模での情報連動ができます。



ワークフローの高度な機能例：
フロールート定義機能 (Popular BPM)
差戻し機能
引戻し機能
取り消し機能 (否認)
処理権限設定機能
メール通知機能 (自動催促含む)
案件操作機能
複雑条件の分岐処理機能等

システム認証

ASP.NET 4.0の認証技術で構築した堅牢なシステム認証機能を標準装備している。Active Directory & OpenLDAP認証を対応。シングルサインオンも対応している。

ログイン

Popular Framework 4

ユーザーID*: administrator

パスワード*:

ログイン パスワードの変更 リセット

[パスワードをお忘れか](#)

- 内部統制要件を満たしたパスワードポリシー適用
- パスワードをASP.NET認証DBにて暗号化
- パスワードの有効期間
- パスワードの複雑さ、強度設定
- 初期パスワード強制変更
- パスワードが間違った回数よりIDのロックアウト
- パスワード忘れに対する初期化自動メール送信
- パスワードポリシーのカスタマイズが可能

パスワードの変更

Popular Framework 4

パスワード:

新しいパスワード:

新しいパスワードの確認入力:

パスワードの変更

権限管理

権限管理はプログラムをコーディングせずに画面アクセス権限とコントロール制限(無効、非表示)を実現する事ができる。ユーザ様にて権限を自由に設定できる。

ホームページ | ロール設定

ロール設定

ロールバージョン: ST2011 | 利用開始日: 2011/04/01 | 利用終了日: 2012/04/01

権限設定

ロールバージョン	ロール	画面名	アクション...	アクション権限
1	運用管理者	メニュー	有リ	アクション権限
2	運用管理者	お知らせ	有リ	アクション権限
3	運用管理者	起票フロー一覧	有リ	アクション権限
4	運用管理者	起票済一覧	有リ	アクション権限
5	運用管理者	未処理一覧	有リ	アクション権限
6	運用管理者	処理中一覧	有リ	アクション権限
7	運用管理者	処理済一覧	有リ	アクション権限
8	運用管理者	参照一覧	有リ	アクション権限
9	運用管理者	案件操作	有リ	アクション権限
10	運用管理者	ユーザ管理詳細	有リ	アクション権限
11	運用管理者	ユーザ管理一覧	有リ	アクション権限
12	運用管理者	組織設定	有リ	アクション権限
13	運用管理者	組織設定(役職変...	有リ	アクション権限

ロール設定

画面権限設定

画面項目制限管理

ロール名: 運用管理者

画面日本語名: ユーザ管理詳細

画面項目制限一覧

物理名	型	日本語名	制限
32 URLX	textfield	URL	✓
33 MAIL1	textfield	メール1	✓
34 MAIL2	textfield	メール2	🔒
35 NOTE	textarea	備考	✓
36 btnAdd	button	登録	✗
37 btnEdit	button	更新	🔒
38 btnClose	button	閉じる	✓
39 comboRoleVersion	combo	ロールバージョン	✓
40 treRole	treep...		✓
41 btnDataConfirm	button	確定	✓

更新 | 閉じる

コントロール制限(無効、非表示)

入力チェック

入力チェックはプログラムをコーディングせずに入力項目をチェックすることができる。チェックルールとエラーメッセージはXMLファイルにて管理する。チェック内容: 必須入力チェック、桁数チェック、項目連携チェック、フォーマットチェック等。

```
<InputCheckRule>
  <ControllerActionName>UserManagement/InsertUser</ControllerActionName>
  <InputCheckItems>
    <InputCheckItem>
      <ControlID>USER_ID</ControlID>
      <ControlName>ユーザ ID</ControlName>
      <HisuNyuryoku>True</HisuNyuryoku>
      <HisuNyuryokuErrID>ME702</HisuNyuryokuErrID>
      <RegExp>^[a-zA-Z0-9]+$</RegExp>
      <RegExpErrID>ME723</RegExpErrID>
      <MaxLength>20</MaxLength>
      <MaxLengthErrID>ME703</MaxLengthErrID>
    </InputCheckItem>
    <InputCheckItem>
      <ControlID>EMPLOYEE_TYPE_ID</ControlID>
      <ControlName>従業員タイプ</ControlName>
      <HisuNyuryoku>True</HisuNyuryoku>
      <HisuNyuryokuErrID>ME702</HisuNyuryokuErrID>
    </InputCheckItem>
    <InputCheckItem>
      <ControlID>LAST_NAME_KANJI</ControlID>
      <ControlType>TextBox</ControlType>
      <ControlName>姓(漢字)</ControlName>
      <HisuNyuryoku>True</HisuNyuryoku>
      <HisuNyuryokuErrID>ME702</HisuNyuryokuErrID>
      <MaxLength>8</MaxLength>
      <MaxLengthErrID>ME703</MaxLengthErrID>
    </InputCheckItem>
    <InputCheckItem>
      <ControlID>FIRST_NAME_KANJI</ControlID>
      <ControlName>名(漢字)</ControlName>
      <HisuNyuryoku>True</HisuNyuryoku>
      <HisuNyuryokuErrID>ME702</HisuNyuryokuErrID>
      <MaxLength>8</MaxLength>
      <MaxLengthErrID>ME703</MaxLengthErrID>
    </InputCheckItem>
  </InputCheckItems>
</InputCheckRule>
```

DBアクセス

DBアクセスはSQL文をプログラムにコーディングせずにSQL文をXMLファイルに記述する。更新系処理はSQL文の代わりにスキーマファイルを利用可能である。DB高速アクセス機能を実現しました。

```
<!-- 案件情報取得
      条件：システム案件ID
-->
<select id="COM.T_PF_MATTER.Select" parameterClass="Dictionary">
  SELECT COUNT(*) AS COUNT FROM T_PF_MATTER
  WHERE SYSTEM_MATTER_ID = #SYSTEM_MATTER_ID#
</select>

<!-- 完了案件情報取得
      条件：システム案件ID
-->
<select id="COM.T_PF_FINISH_MATTER.Select" parameterClass="Dictionary">
  SELECT COUNT(*) AS COUNT FROM T_PF_FINISH_MATTER
  WHERE SYSTEM_MATTER_ID = #SYSTEM_MATTER_ID#
</select>

<!-- 完了参照権限ロール情報取得
      条件：システム案件ID、ロールID
-->
<select id="COM.T_PF_FINISH_REFER_AUTH_ROLE.Select" parameterClass="Dictionary">
  SELECT COUNT(*) AS COUNT FROM T_PF_FINISH_REFER_AUTH_ROLE
  WHERE SYSTEM_MATTER_ID = #SYSTEM_MATTER_ID# AND
  ROLE_ID = #ROLE_ID#
</select>

<!-- 過去案件情報取得
      条件：システム案件ID
-->
<select id="COM.T_PF_PAST_MATTER.Select" parameterClass="Dictionary">
  SELECT COUNT(*) AS COUNT FROM T_PF_PAST_MATTER
  WHERE SYSTEM_MATTER_ID = #SYSTEM_MATTER_ID#
</select>
```

機能:

- ・SQL文をメモリキャッシュ機能
- ・SQLインジェクション攻撃防止機能
- ・データ型をチェック機能
- ・排他機能

EXCELファイルインポート

EXCELファイルインポートはノンプログラムでEXCELファイルをデータベースにインポートする機能である。インポート列の定義とインポートルールはXMLファイルで定義する。インポート結果を出力することができる。

EXCEL帳票出力

EXCEL帳票出力はノンプログラムで本格的な帳票設計開発機能を実現しました。EXCELテンプレートとSQL文XMLファイルを定義して、高速な多機能EXCEL帳票を出力することができる。ユーザー様にて帳票を自由にカスタマイズができる。市販ActiveReports等も利用可能です。

作業依頼・申請書

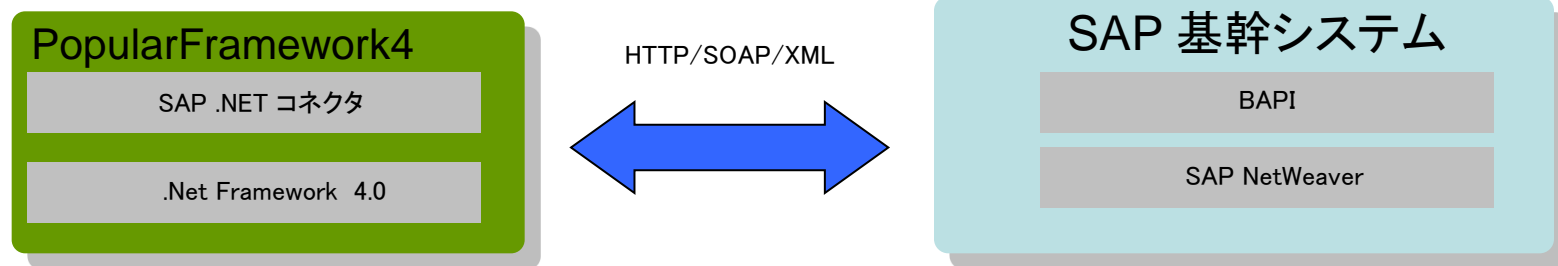
作業承認		受付番号	{\$MATTER.MATERID}		
		受付日	{\$TASK}年{\$TASK}月{\$TASK}日		
申請部署	{\$REQUEST.TABLE.APPLY.PART_HEAD}		{\$REQUEST.TABLE.APPLY.PART_ROOM}		
申請者氏名	{\$REQUEST.TABLE.APPLY.USERNAME}	申請者連絡先	{\$REQUEST.TABLE.APPLY.USER_CONTACT}		
申請者ID	{\$REQUEST.TABLE.APPLY.USERID}	完成フロア	{\$REQUEST.TABLE.SEAT_FLOOR}		
申請日	{\$TABLE.APPLY}年{\$TABLE}月{\$TABLE}日	完了希望日	{\$REQUEST.HOPE}年{\$SHU}月{\$SHU}日		
申請内容		{\$MATTER.FLOW.GROUP.NAME}	{\$MATTER.FLOW.DISPLAY.NAME}		
{\$MATTER.MATTER.NAME}					
登録区分					
希望メールアドレス名					
利用希望ドメイン					
利用目的					
管理者					
部署名	部署	従業員No.	氏名	Mailアドレス	
{\$tableNo}COMMON	{\$PART.NAME}	{\$EMPLOYEEID}	{\$NAME}	{\$MAIL_ADDRESS}	
報告 担当書: {\$WORK.INFO}					
作業完了日		{\$WORK.INFO}年{\$WORK.INFO}月{\$WORK.INFO}日		作業時間	{\$WORK.INFO.WORK.TIME} H
{\$WORK.INFO.WORK}申請内容の通り作業完了/完了/変更点・特記事項は下記の通り					
{\$WORK.INFO.WORK.CONTENT}					
完了承認		コメント			
		{\$TASK.CONTENT}			

作業依頼・申請書

作業承認		受付番号	HO-001-05-001		
		受付日	2011年05月31日		
申請部署	本社	申請者氏名	朱元理	申請者連絡先	
申請者ID	100	申請者ID	100	完成フロア	
申請日	2011年05月31日	完了希望日	2011年06月07日		
申請内容		作業依頼・アプリケーション			
{\$TEST.内容}申請内容{\$testTime}05/31/11					
{\$TEST.内容}申請内容{\$testTime}06/01/11					
コメント					
{\$TEST.内容}申請内容{\$testTime}06/01/11					
報告 担当書: test		作業時間			
作業完了日		2011年05月31日		作業時間	10:00 H
{\$TEST.内容}申請内容の通り作業完了					
{\$TEST.内容}申請内容の通り作業完了/完了/変更点・特記事項は下記の通り					
{\$TEST.内容}申請内容{\$testTime}06/01/11					
完了承認		コメント			
		{\$TASK.CONTENT}			

SAP基幹システムの相互連携

Popular Framework4は、SAP NetWeaverに相互接続性/運用性、オープン性、拡張性を提供します。このオープン性は、Webサービスによる高度なソフトウェア統合を可能にする、異機種システム環境での効果的な管理を行うための基盤になります。



SAP NetWeaverの標準サポート :

Microsoft .NETとの相互接続性/運用性

SAPとMicrosoftは10年にわたるパートナーです。実際、Microsoft WindowsはSAPのソリューションをサポートする唯一のプラットフォームです。両社より強固な製品の統合実現のために、企業の枠を越えた円滑なコラボレーションを実現しています。

SAP NetWeaverは、Microsoftの開発プラットフォームMicrosoft .NETに相互接続性/運用性、オープン性、拡張性を提供します。このオープン性は、Webサービスによる高度なソフトウェア統合を可能にするとともに、異機種システム環境での効果的な管理を行うための基盤にもなります。

Microsoft .NETとの相互接続性/運用性は、人、情報、プロセス、そしてアプリケーションレベルにまで及びます。

SAP .NET コネクタ

用途

SAP .NET コネクタは、Microsoft .NET プラットフォームと SAP システムとの間の通信 Studio .Net で Web フォーム、Windows フォーム、コンソールアプリケーションなどの動的なプログラム言語を使用することができます。

機能

- SAP .NET コネクタおよび SAP .NET プロキシウィザードを使用すると、以下のことが可能です。
- SAP ビジネスオブジェクト (BAPI) にアクセスする .NET Windows フォームアプリ
 - SAP サーバ側クライアントアプリケーションを開発します。ここでは RFC または I
 - .NET 環境で動作し、SAP システムからインストールを開始することが可能な RF

Microsoft Visual Studio .NET を使用すると、以下のことが可能です。

- 統合プロキシウィザードを使用して、プロキシオブジェクトを生成します。
- Microsoft Visual Studio .NET に完全にアクセス可能なすべての一般的なプロ
- SAP テーブルおよび構造を Windows および Web フォームコントロールにリンク
- シングルサインオン、Kerberos、Microsoft Passport (www.passport.net)

ログ出力

ログ出力は出力制御情報とフォーマットをXMLに定義して、システムログ、アクセスログ、操作ログを出力することができる。

WebAccessLog_20110608.log	2011/06/08 8:51	テキスト ドキュ...
WebAccessLog_20110606.log	2011/06/06 10:14	テキスト ドキュ...
WebAccessLog_20110602.log	2011/06/02 14:36	テキスト ドキュ...
WebAccessLog_20110531.log	2011/05/31 9:51	テキスト ドキュ...
WebAccessLog_20110530.log	2011/05/30 13:19	テキスト ドキュ...
SystemLog_20110608.log	2011/06/08 15:55	テキスト ドキュ...
SystemLog_20110606.log	2011/06/06 17:28	テキスト ドキュ...
SystemLog_20110602.log	2011/06/02 16:07	テキスト ドキュ...
SystemLog_20110531.log	2011-06-08 14:32:11.822	[4960] DEBUG - PF.Dao.IBatisNet.DataTableAccess.Update MESSAGE=Update start
SystemLog_20110530.log	2011-06-08 14:32:11.853	[4960] DEBUG - PF.Dao.IBatisNet.DataTableAccess.Update MESSAGE=Update end
OperationLog_20110608.log	2011-06-08 14:32:11.853	[4960] DEBUG - WSD.Batch.Service.UpdateVersionService.UpdateVersion MESSAGE=組織バージョンの更新が成功しました!
OperationLog_20110606.log	2011-06-08 14:32:11.853	[4960] DEBUG - WSD.Batch.Service.UpdateVersionService.UpdateVersion MESSAGE=ロールバージョンID=[LST2011]
OperationLog_20110602.log	2011-06-08 14:32:11.853	[4960] DEBUG - PF.Dao.IBatisNet.DataTableAccess.GetEmptyDataTable MESSAGE=GetEmptyDataTable start
OperationLog_20110531.log	2011-06-08 14:32:11.853	[4960] DEBUG - PF.Dao.IBatisNet.DataTableAccess.GetEmptyDataTable MESSAGE=GetEmptyDataTable end
OperationLog_20110530.log	2011-06-08 14:32:11.853	[4960] DEBUG - PF.Dao.IBatisNet.DataTableAccess.Update MESSAGE=Update start
DataAccessLog_20110608.log	2011-06-08 14:32:11.853	[4960] DEBUG - PF.Dao.IBatisNet.DataTableAccess.Update MESSAGE=Update end
DataAccessLog_20110606.log	2011-06-08 14:32:11.853	[4960] DEBUG - WSD.Batch.Service.UpdateVersionService.UpdateVersion MESSAGE=ロールバージョンIDの更新が成功しました!
DataAccessLog_20110602.log	2011-06-08 14:32:11.853	[4960] DEBUG - WSD.Batch.Service.UpdateVersionService.UpdateVersion MESSAGE=組織バージョンID=[LST2011]
DataAccessLog_20110531.log	2011-06-08 14:32:11.853	[4960] DEBUG - PF.Dao.IBatisNet.DataTableAccess.GetEmptyDataTable MESSAGE=GetEmptyDataTable start
DataAccessLog_20110530.log	2011-06-08 14:32:11.853	[4960] DEBUG - PF.Dao.IBatisNet.DataTableAccess.GetEmptyDataTable MESSAGE=GetEmptyDataTable end
	2011-06-08 14:32:11.853	[4960] DEBUG - PF.Dao.IBatisNet.DataTableAccess.Update MESSAGE=Update start
	2011-06-08 14:32:11.853	[4960] DEBUG - PF.Dao.IBatisNet.DataTableAccess.Update MESSAGE=Update end
	2011-06-08 14:32:11.853	[4960] DEBUG - WSD.Batch.Service.UpdateVersionService.UpdateVersion MESSAGE=組織バージョンIDの更新が成功しました!
	2011-06-08 14:32:11.853	[4960] DEBUG - WSD.Batch.Service.UpdateVersionService.UpdateVersion MESSAGE=ロールバージョンID=[LST2011]
	2011-06-08 14:32:11.853	[4960] DEBUG - PF.Dao.IBatisNet.DataTableAccess.GetEmptyDataTable MESSAGE=GetEmptyDataTable start
	2011-06-08 14:32:11.853	[4960] DEBUG - PF.Dao.IBatisNet.DataTableAccess.GetEmptyDataTable MESSAGE=GetEmptyDataTable end
	2011-06-08 14:32:11.853	[4960] DEBUG - PF.Dao.IBatisNet.DataTableAccess.Update MESSAGE=Update start
	2011-06-08 14:32:11.853	[4960] DEBUG - PF.Dao.IBatisNet.DataTableAccess.Update MESSAGE=Update end
	2011-06-08 14:32:11.853	[4960] DEBUG - WSD.Batch.Service.UpdateVersionService.UpdateVersion MESSAGE=組織バージョンIDの更新が成功しました!
	2011-06-08 14:32:11.853	[4960] DEBUG - WSD.Batch.Service.UpdateVersionService.UpdateVersion MESSAGE=ロールバージョンID=[LST2011]
	2011-06-08 14:32:11.853	[4960] INFO - WSD.Batch.Service.FinishDataMoveService.DataMove MESSAGE=データの転移(完了->過去)を開始しました。
	2011-06-08 14:32:12.072	[4960] DEBUG - WSD.Batch.Service.FinishDataMoveService.DataMove MESSAGE=完了案件保存期限日=[20100608]
	2011-06-08 14:32:12.088	[4960] DEBUG - WSD.Batch.Service.FinishDataMoveService.DataMove MESSAGE=過期完了案件件数=[0]
	2011-06-08 14:32:12.088	[4960] DEBUG - WSD.Batch.Service.FinishDataMoveService.DataMove MESSAGE=過期完了案件情報が存在しないので、データの転移が不要
	2011-06-08 14:32:12.088	[4960] INFO - WSD.Batch.Service.FinishDataMoveService.DataMove MESSAGE=データの転移(完了->過去)を正常に終了しました。
	2011-06-08 14:32:12.103	[4960] INFO - WSD.Batch.Service.PastDataDeleteService.DataDelete MESSAGE=データの削除(過去)を開始しました。
	2011-06-08 14:32:12.103	[4960] DEBUG - WSD.Batch.Service.PastDataDeleteService.DataDelete MESSAGE=過去案件保存期限日=[20080608]
	2011-06-08 14:32:12.103	[4960] DEBUG - WSD.Batch.Service.PastDataDeleteService.DataDelete MESSAGE=過期過去案件件数=[0]
	2011-06-08 14:32:12.103	[4960] DEBUG - WSD.Batch.Service.PastDataDeleteService.DataDelete MESSAGE=過期過去案件情報が存在しないので、データの削除が不
	2011-06-08 14:32:12.103	[4960] INFO - WSD.Batch.Service.PastDataDeleteService.DataDelete MESSAGE=データの削除(過去)を正常に終了しました。
	2011-06-08 14:32:12.103	[4960] INFO - WSD.Batch.Service.AttachmentFileDelete.FileDelete MESSAGE=添付ファイルの削除を開始しました。
	2011-06-08 14:32:12.103	[4960] DEBUG - WSD.Batch.Service.AttachmentFileDelete.FileDelete MESSAGE=未完了添付ファイルを削除 start
	2011-06-08 14:32:12.119	[4960] DEBUG - WSD.Batch.Service.AttachmentFileDelete.DeleteFolderAndFile MESSAGE=添付ファイルのConfig定義ディレクトリで、削
	2011-06-08 14:32:12.119	[4960] DEBUG - WSD.Batch.Service.AttachmentFileDelete.FileDelete MESSAGE=未完了添付ファイルを削除 end
	2011-06-08 14:32:12.119	[4960] DEBUG - WSD.Batch.Service.AttachmentFileDelete.FileDelete MESSAGE=完了添付ファイルを削除 start
	2011-06-08 14:32:12.119	[4960] DEBUG - WSD.Batch.Service.AttachmentFileDelete.DeleteFolderAndFile MESSAGE=添付ファイルのConfig定義ディレクトリ内で
	2011-06-08 14:32:12.119	[4960] DEBUG - WSD.Batch.Service.AttachmentFileDelete.FileDelete MESSAGE=完了添付ファイルを削除 end