

イントラネット・スタートパック Ver6.1

カスタマイズガイド

初版

2007 年 12 月 31 日

<< 変更履歴 >>

変更年月日	変更内容
2007/12/31	初版

<< 目次 >>

1	はじめに	2
1.1	目的	2
1.2	本資料の記載内容について	2
2	カスタマイズの概要	3
2.1	カスタマイズの種類と一般的な対応手順	3
2.1.1	画面から設定できるケース	4
2.1.2	設定ファイルを修正することで解決できるケース	7
2.1.3	View層 (HTMLやJSPファイル) を修正することで解決できるケース	7
2.1.4	ロジック層 (JS/Java/DB定義) を修正することで解決できるケース	11
2.2	カスタマイズに関する規約と注意点	11
2.2.1	規定	11
2.2.2	環境移行の手順	11
2.2.3	注意事項	12
3	カスタマイズ事例集	13
3.1	スケジュール管理システムのカスタマイズ	13
3.1.1	画面上の表現をカスタマイズする	13
3.1.2	Microsoft Outlookからスケジュールデータをインポートする	14
3.2	ドキュメント管理システムのカスタマイズ	23
3.2.1	ドキュメントの管理属性を追加する	23
3.2.2	登録されているドキュメントの一覧を出力する	26
3.2.3	APIを使ってドキュメントを登録する	28
3.3	電子会議室のカスタマイズ	31
3.3.1	登録可能な文字数を拡張する	31
3.4	ポートレットの作成	33
3.4.1	ドキュメントフォルダ・ポートレットを作成する	33
3.5	ワークフロー連携	40
3.5.1	掲示板掲示前承認機能の作成	40
4	資料	50
4.1	初期化データ一覧	50
4.2	区分コード一覧	51
4.3	個人設定項目一覧	53
4.4	初期化ファイル項目一覧	60
4.4.1	startpack.ini	60
4.4.2	webmail.ini	64
5	索引	67

1 はじめに

1.1 目的

本ドキュメントは、イントラネット・スタートパック Ver6.1 をカスタマイズするために必要な設定や作業手法について説明をすることを目的とします。主に下記の内容について記載しています。

- カスタマイズの種類と一般的な対応手順
- カスタマイズに関する規定と注意点
- カスタマイズ事例集
- 資料 — 設定項目一覧

また、本資料は下記事項を想定して記載しています。

- HTML、JavaScript、JAVA 言語の概要を理解している。

1.2 本資料の記載内容について

本資料ではカスタマイズ事例として、実際のソースコードサンプルを掲載しています。本書で作成するアプリケーションはあくまでも、カスタマイズ作業の流れを理解することに主眼をおいていますので、必ずしもベストなコーディング方法とはいえない方法もあえて取っている個所があります。あくまでも、サンプルとしての位置付けでとらえるようにしてください。

本資料で紹介しているソースコードにつきましては下記URLからダウンロードすることができます。

<http://www.intra-mart.jp/download/product/index.html>

本文の説明中に記載しているソースコードは、編集の都合上、実際のソースコードと一部異なる場合がありますので、あらかじめご了承ください。

尚、本資料に記載されている内容については、今後のバージョン(リビジョン、パッチ含む)における機能追加、仕様変更に伴い、予告なく変更する場合があります。

詳細については、弊社ホームページ 製品最新情報ダウンロードページ

<<http://www.intra-mart.jp/download/product/index.html>>をご参考ください。

2 カスタマイズの概要

2.1 カスタマイズの種類と一般的な対応手順

イントラネット・スタートパックのカスタマイズは、大きく分類すると下記の4つに分類されます。

No.	ケース	主に実現できること
1	画面から設定できるケース	勤怠管理の休憩時間に関する共通設定や、スケジュールの初期表示方法に関する個人毎設定などを、プログラムを変更せずに行うことができます。
2	設定ファイルを修正することで解決できるケース	各データの登録文字数の設定などを変更できます。
3	View 層 (HTML や JSP ファイル) を修正することで解決できるケース	表示キャプションや、色・文字サイズなどの変更が可能です。
4	ロジック層 (JS/Java/DB 定義) を修正することで解決できるケース	データ項目の追加や計算方法・チェック条件などの変更が可能です。

さらに、カスタマイズの影響範囲・方法・実施者の対応は下記の図の様になります。

影響範囲	カスタマイズ方法	実施者
個人単位	個人設定 画面からの設定変更	各ユーザ
ログイングループ単位	マスタメンテナンス設定 画面からの設定変更	システム管理者
システム全体	設定ファイルの変更	
	View層プログラムの変更	システム管理者／ プログラマー
	ロジック層プログラムの変更	プログラマー

以降の章では、上記それぞれのパターンにおけるカスタマイズの実際の手順を説明します。

2.1.1 画面から設定できるケース

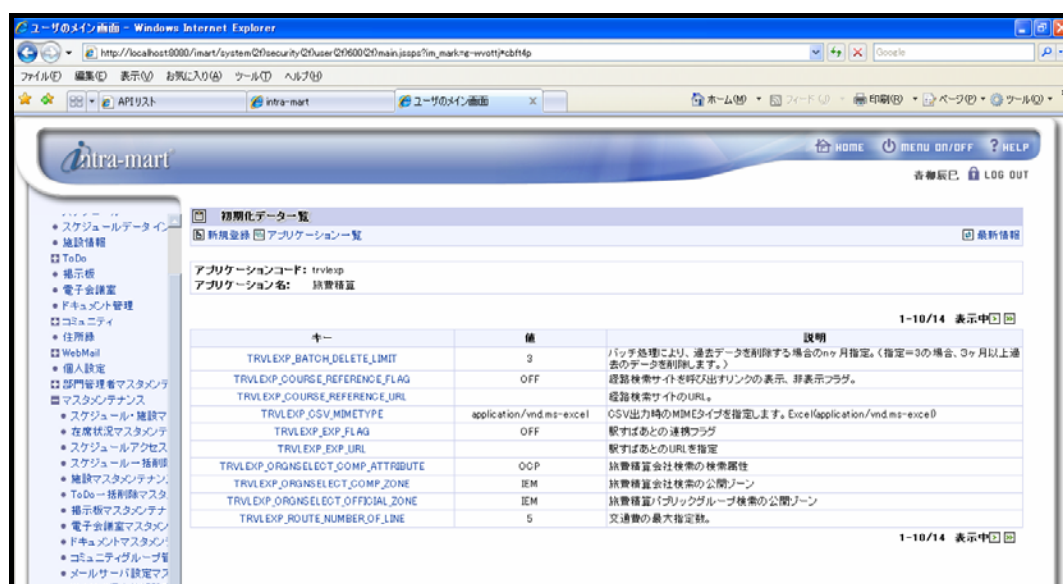
プログラムを修正しなくても、イントラマートの画面から設定が可能なケースです。

2.1.1.1 初期化データ設定

各アプリケーションで設定可能な区分コードを編集できます。

メニューから[イントラネット・スタートパック]－[環境設定]－[初期化データ設定]を選択し、各アプリケーションの初期設定値を編集します。

本設定値を修正した場合に再起動が必要か否かについては規定されていません。実際には各アプリケーション／各設定値の実装によりますので事前に確認した上で運用する必要があります。



初期化データの一覧については下記を参照してください。

⇒ [4.1初期化データ一覧](#)

2.1.1.2 区分コード設定マスタ

各アプリケーションで設定可能な区分コードを編集できます。

メニューから[イントラネット・スタートバック]―[マスタメンテナンス]―[区分コード設定マスタ]を選択し、各アプリケーションの区分設定値を編集します。本機能は、前出の初期化データ設定機能とほぼ同様の目的で使用されますが、下記の点において高機能になっています。

- グループ化した区分コードを定義できる
- ロケールを指定できる

「初期化データ設定」か「区分コード設定」のどちらを使用するかは、アプリケーションや区分コードにより異なります。また、本設定値を修正した場合に再起動が必要か否かについては規定されていません。実際には各アプリケーション／各設定値の実装によりますので事前に確認した上で運用する必要があります。



「区分コード設定マスタ」を使用したカスタマイズ例については下記を参照してください。

⇒ [3.2.1ドキュメントの管理属性を追加する](#)



区分コードの一覧は下記を参照してください。

⇒ [4.2区分コード一覧](#)

2.1.1.3 個人設定

各アプリケーションの動作仕様の内、ユーザ毎の好みに応じて設定が可能になっている項目を設定できます。

本画面は一般ユーザに開放し、各自が設定することを想定しています。

メニューから[イントラネット・スタートバック]―[個人設定]を選択し、個人設定値を編集します。

設定内容は個人毎に有効となりますので、ユーザ全員に設定したい場合はすべてのユーザでログインし、設定をする必要があります。

個人設定<スケジュール>	
ToDo 電子会議室 住所録 WebMail ポータル ブログ 最新情報 ヘルプ	
共通設定	
表示単位(日)	60 分 ※グループ表示画面は除く
表示時間帯	
先頭表示(週)	当日 ※個人週表示を除く
アクセス権	設定
ユーザスケジュール設定	
表示方法	グループ日表示
選択グループの追加	<div style="border: 1px solid #ccc; height: 100px; width: 300px;"></div> <div style="text-align: right;"> グループ検索 削除 </div>
施設スケジュール設定	
表示方法	グループ施設週表示
デフォルトグループ	施設グループ
	分類
メール即時配信設定	
登録時チェック	<input checked="" type="checkbox"/> 登録時に参加者へメールを送信する(<input checked="" type="checkbox"/> 自分を宛先に含める)
更新時チェック	<input checked="" type="checkbox"/> 登録者・更新者へメールを送信する
	<input checked="" type="checkbox"/> 参加者へメールを送信する(<input checked="" type="checkbox"/> 自分を宛先に含める)
営業日報設定	
営業日報作成チェック	<input type="checkbox"/> 表示する(<input type="checkbox"/> 外出設定と連動する)
メール定期配信設定	
配信頻度	<input checked="" type="radio"/> 配信なし <hr/> <input type="radio"/> 毎日: 指定の時間にスケジュール通知を行います 送信する時間 午前 10 時 送信するスケジュール <input type="checkbox"/> 当日分 <input type="checkbox"/> 翌日分 <hr/> <input type="radio"/> 毎週: 指定の曜日に一週間分のスケジュール通知を行います 曜日 日 時間 午前 10 時
配信先	<input checked="" type="checkbox"/> 通常アドレス (未設定) <input type="checkbox"/> モバイルアドレス (未設定) <input type="checkbox"/> ユーザ指定のメールアドレス
有効期間開始日	2007/12/14 <input type="button" value="日付選択"/>
有効期間終了日	2007/12/14 <input type="button" value="日付選択"/>
<input type="button" value="更新"/>	



個人設定項目の一覧は下記を参照してください。

⇒ [4.3.個人設定項目一覧](#)

2.1.2 設定ファイルを修正することで解決できるケース

INI ファイルや XML ファイルを修正することで解決できるケースです。
イントラネット・スタートパックには下記の設定ファイルが用意されています。

ファイル名	備考
startpack.ini	スタートパック全体の動作に関する初期設定
webmail.ini	WebMail の動作に関する初期設定
isp-input-validation.xml	入力書式チェックの動作に関する初期設定

プログラムの修正を行わなくても、ある程度の動作変更は設定ファイルの変更で対応できます。
実際の設定内容については下記を参照してください。



「startpack.ini」の設定項目一覧については下記を参照してください。

⇒ [4.4.1startpack.ini](#)



「webmail.ini」の設定項目一覧については下記を参照してください。

⇒ [4.4.2webmail.ini](#)



「isp-input-validation.xml」を使用したカスタマイズ事例については下記を参照してください。

⇒ [3.3.1登録可能な文字数を拡張する](#)

2.1.3 View 層 (HTML や JSP ファイル) を修正することで解決できるケース

画面表示上のキャプションや配色などの軽微な変更は View 層のみの修正で解決できます。

イントラネット・スタートパックはスクリプト開発モデルを採用していますので、View 層の修正はプレゼンテーションページ (HTML) の修正となります。

それでは実際にプレゼンテーションページの修正を行ってみましょう。

勤務登録

勤務日付 2007/12/26

出勤

退社

実働時間計 0.0 [H]

勤務体系(必須)

備考

項番	開始時間	終了時間	実働時間	業務1
1			[H]	
2			[H]	

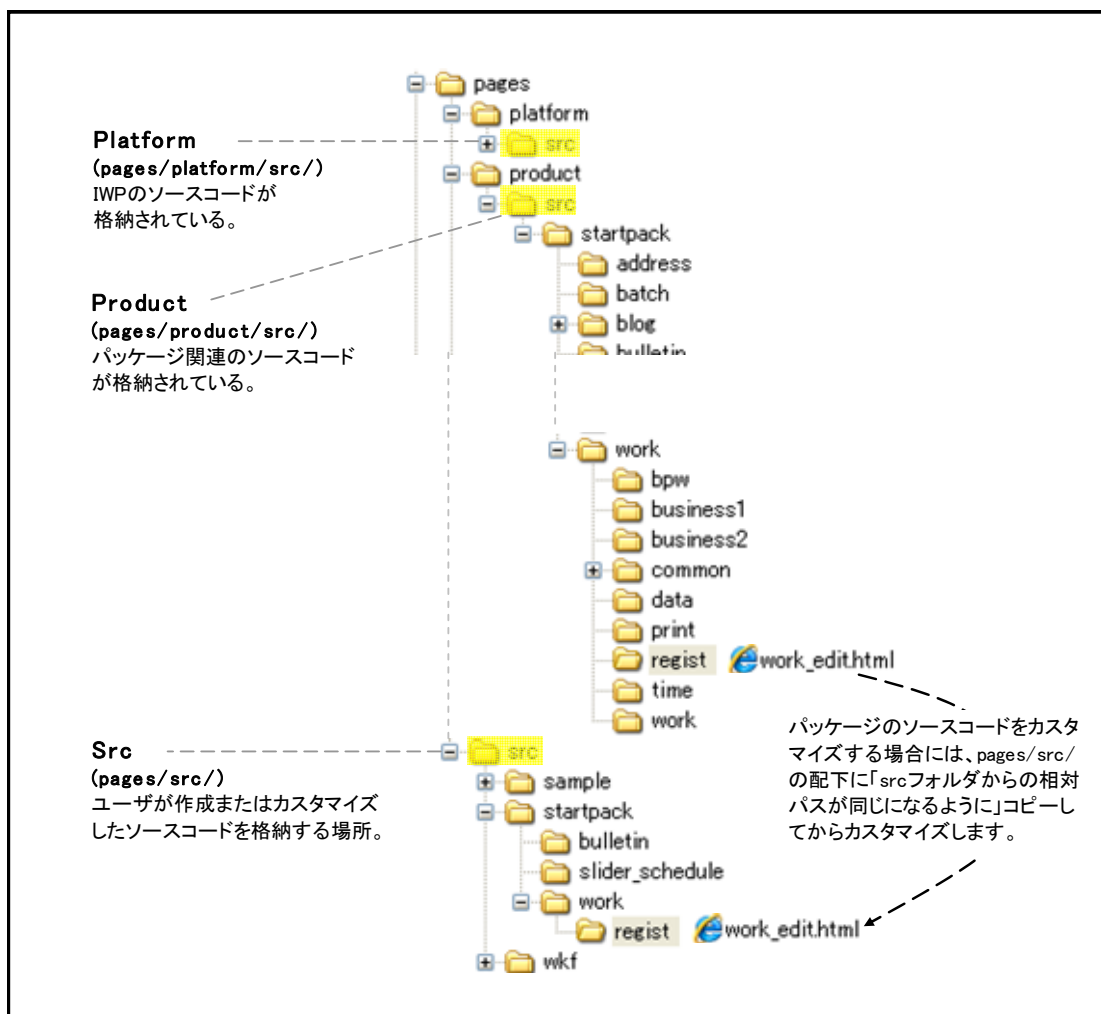
勤怠管理アプリケーションの勤務登録画面で、「勤務日付」というキャプションを「出勤日」に変更します。このページのオリジナルのプレゼンテーションページは「(ROOT)/pages/product/src/startpack/work/regist/work_edit.html」にありますので、これを直接修正しても表示を変更することができますが、通常はオリジナルを直接編集せずに、(ROOT)/pages/src/startpack/work/regist/フォルダ配下にコピーしてから、それを修正するようにします。その理由は、イントラマートがソースコードを管理するディレクトリ内で、パッケージから提供されるソースコードとユーザがカスタマイズしたソースコードを分離して管理できるようになっているからです。このルールに従うと、パッケージオリジナルのソースコードを残したまま、カスタマイズしたソースコードを配置することができます。

フォルダ	説明
pages/platform/src	IWP のソースコード
pages/product/src	イントラネット・スタートパックなどのアプリケーションシリーズのソースコード
pages/src	カスタマイズやスクラッチ開発のソースコード

このように、ソースコード保管用のフォルダが分かれており、src フォルダからの相対パスが一致するファイルに関しては pages/src フォルダが優先します。例えば、

- (A) pages/product/src/test.html
- (B) pages/src/test.html

の二つのファイルがあった場合、(A)と(B)は同じパスにあると判断され、(B)のソースコードが採用されます。



それでは、「(ROOT)/pages/product/src/startpack/work/regist/work_edit.html」を
「(ROOT)/pages/src/startpack/work/regist/work_edit.html」にコピーして、修正してみましょう。

<修正前>

```

<BR>
<IMART type="form" name="attendForm" method="POST" >
  <IMART type="hidden" nDayLine=nDayLine sRestTime=sRestTime></IMART>
  <IMART type="hidden" sTargetFullDate=sTargetFullDate></IMART>
  <IMART type="hidden" json_url=json_url></IMART>
  <TABLE width="80%" align="center" class="edit">
    <TR>
      <TD class="bottom">勤務日付</TD>
      <TD class="bottom">
        <IMART type="string" value=sTargetFullDate></IMART>
      </TD>
    </TR>
  </TABLE>

```

<修正後>

```

<BR>
<IMART type="form" name="attendForm" method="POST" >
  <IMART type="hidden" nDayLine=nDayLine sRestTime=sRestTime></IMART>
  <IMART type="hidden" sTargetFullDate=sTargetFullDate></IMART>
  <IMART type="hidden" json_url=json_url></IMART>
  <TABLE width="80%" align="center" class="edit">
    <TR>
      <TD class="bottom">出勤日</TD>
      <TD class="bottom">
        <IMART type="string" value=sTargetFullDate></IMART>
      </TD>
    </TR>
  </TABLE>

```

キャプションが変更されました。

♪ポイント♪

ファイルを修正しても表示が反映されない場合、IWP の再起動をすると反映されることがあります。「pages/product/src」フォルダ配下にはソースコードの自動コンパイル機能がデフォルトで ON となっているため、動的にスクリプトコードの変更を反映しない設定になっています。それに対して、「pages/src/」フォルダ配下は自動コンパイル機能がデフォルトで OFF になっており、インタプリタは常に最新のソースコードを実行するようになっています。これらの設定は、各 src フォルダ配下の「source-config.xml」を修正することで変更することができます。

2.1.4 ロジック層(JS/Java/DB 定義)を修正することで解決できるケース

View 層の修正で対応できない場合には内部処理(ロジック層)の修正が必要となります。

イントラネット・スタートパックはスクリプト開発モデルを採用していますので、ロジック層の修正はファンクションコンテナ(JS)の修正となります。ファンクションコンテナにおいても、プレゼンテーションページと同様、下記のようにフォルダが分かれています。

フォルダ	説明
pages/platform/src	IWP のソースコード
pages/product/src	イントラネット・スタートパックなどのアプリケーションシリーズのソースコード
pages/src	カスタマイズやスクラッチ開発のソースコード

イントラネット・スタートパックのソースコードは「pages/product/src/startpack」フォルダ配下に入っていますので、「pages/src/startpack」配下にコピーをしてからカスタマイズするようにしてください。



ファンクションコンテナをカスタマイズした事例については下記を参照してください。

⇒ [3.5.1 掲示板掲示前承認機能の作成](#)

2.2 カスタマイズに関する規約と注意点

2.2.1 規定

intra-mart では、製品として提供されたプログラムを自由にカスタマイズして利用することができます。

カスタマイズが可能なプログラムは、ソース公開として提供されているプログラムファイル(*.ini, *.html, *.js, *.properties, *.java および *.jsp など)すべてが対象となります。

但し、製品のソースに対してカスタマイズをした場合、カスタマイズをしたプログラムおよび動作に関連のあるプログラム群に関して、同パッケージの提供元は動作保証をいたしません。また、カスタマイズをしたことにより発生した不具合に関しては、サポート対象外となります。

2.2.2 環境移行の手順

カスタマイズしたプログラムを別の環境へ移行(例えば開発機から本番運用機への移行)する場合、以下の手順で移行を行って下さい。

- 移行先へ intra-mart をインストールします。
- 移行先環境の intra-mart に対してライセンス登録を行います。
- 移行先環境の intra-mart を停止します。
- カスタマイズした各プログラムソース(ini, html および js)を元環境から移行先環境へコピーします。
- 移行先の intra-mart を起動します。

カスタマイズしたプログラムを移行する場合、Resource Service の pages/ ディレクトリ内にある html および js 以外のファイルと Permanent Data Service の treasure/ ディレクトリを上書きコピーしてしまわないように注意して

下さい。万一、元環境から移行先環境に対してバイナリファイルの上書きコピーをしてしまいシステムが正常に動作しなくなった場合には、すべてのファイルを削除して移行先への intra-mart のインストールから再度行うようにして下さい。

2.2.3 注意事項

ソースが公開されているプログラムであっても、そのコード中に非公開のAPI(APIリストに記載のないもの)を利用している場合があります。これら、非公開APIは予告なく仕様が変更されることがあるので注意が必要です。

製品のソースを直接カスタマイズした場合、カスタマイズをしたソースに関してはバージョンアップ対象外となります。パッチおよびバージョンアップ版のインストールの際にはソースが自動的に上書きされてしまうことがありますので、ご注意ください。

なお、本項の詳細については、intra-mart IWP/AFW の API リスト【カスタマイズ】の項を参照してください。

3 カスタマイズ事例集

3.1 スケジュール管理システムのカスタマイズ

3.1.1 画面上の表現をカスタマイズする

イントラネット・スタートパックのスケジュール画面には、自分や他ユーザのスケジュールが色付きのバーで表現されます。ここでは、グループスケジュールのファンクションコンテナに手を加え、画面上の表現を変更するカスタマイズ手順を説明します。



今回のカスタマイズの仕様は下記の通りとします。

- (1) グループスケジュール(日単位)の画面において、自分が参加予定である正式なスケジュールを表現するバーの背景色を、初期のカラーからシアン(カラーコード #00FFFF)に変更する。

それでは、このカスタマイズの作成手順を説明します。
大まかな手順は次のようになります。

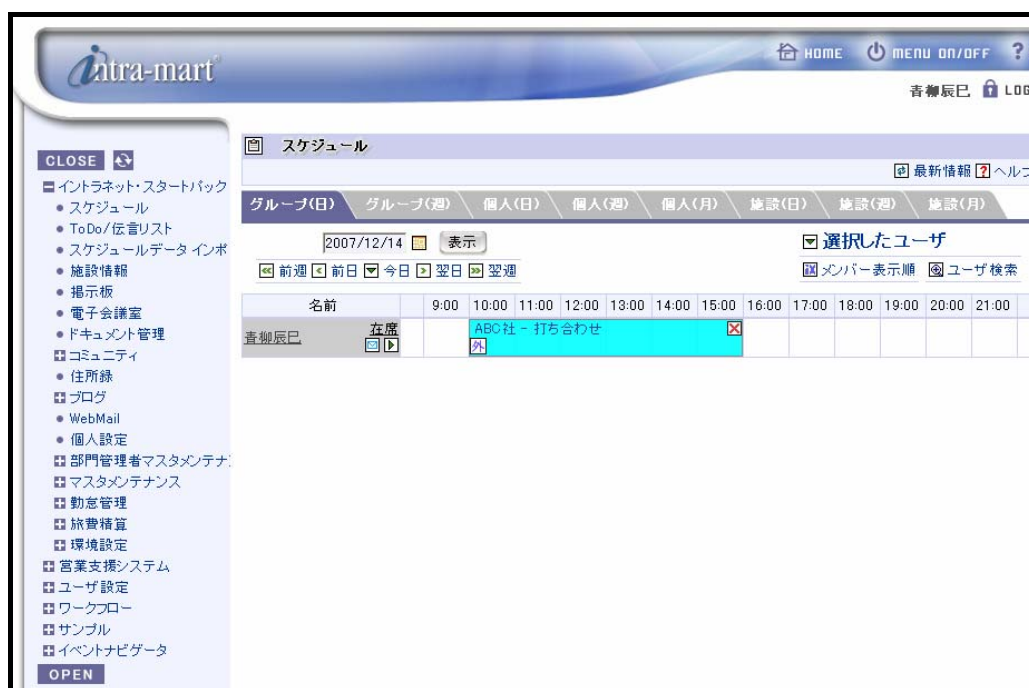
- (1) ファンクションコンテナ(JS)を編集する

編集するファイルは下記の通りです。

ファイルパス	ファイル名	説明
(ROOT) /pages/product/src/startpack/slider_schedule/ ↓ (コピー) (ROOT)/pages/src/startpack/slider_schedule/	sdl_lst_grp_day.js	ファンクションコンテナ

(ROOT) は Resource Service ルートを示します。

- (1) ファンクションコンテナを編集します。
完成イメージは次のようになります。



(白黒だとわかり辛いですが、ピンクからシアンに変更されています。)

ソースコードの修正例を示します。

```
// 正式スケジュールの色を指定している箇所をコメントアウトします。
// var COLOR_SCHE_REGULAR = "#FFEBEB";

// 新たな色コードを指定します。
var COLOR_SCHE_REGULAR = "#00FFFF";
```

3.1.2 Microsoft Outlook からスケジュールデータをインポートする

イントラネット・スタートパックにはスケジュールインポート機能が付属していますので、手動で取り込み設定をすれば Outlook のスケジュールデータを取り込むことができるようになっています。しかし、ここではさらに便利な「**極力手作業をしないで Outlook のスケジュールデータを取り込むことができる簡易ツール**」を作成します。

スケジュールデータ取り込み機能を作成するためには、イントラネット・スタートパックのデータベースにアクセスする必要がありますが、ここではイントラネット・スタートパックのAPIを利用して簡単にデータベース登録ができることを説明します。

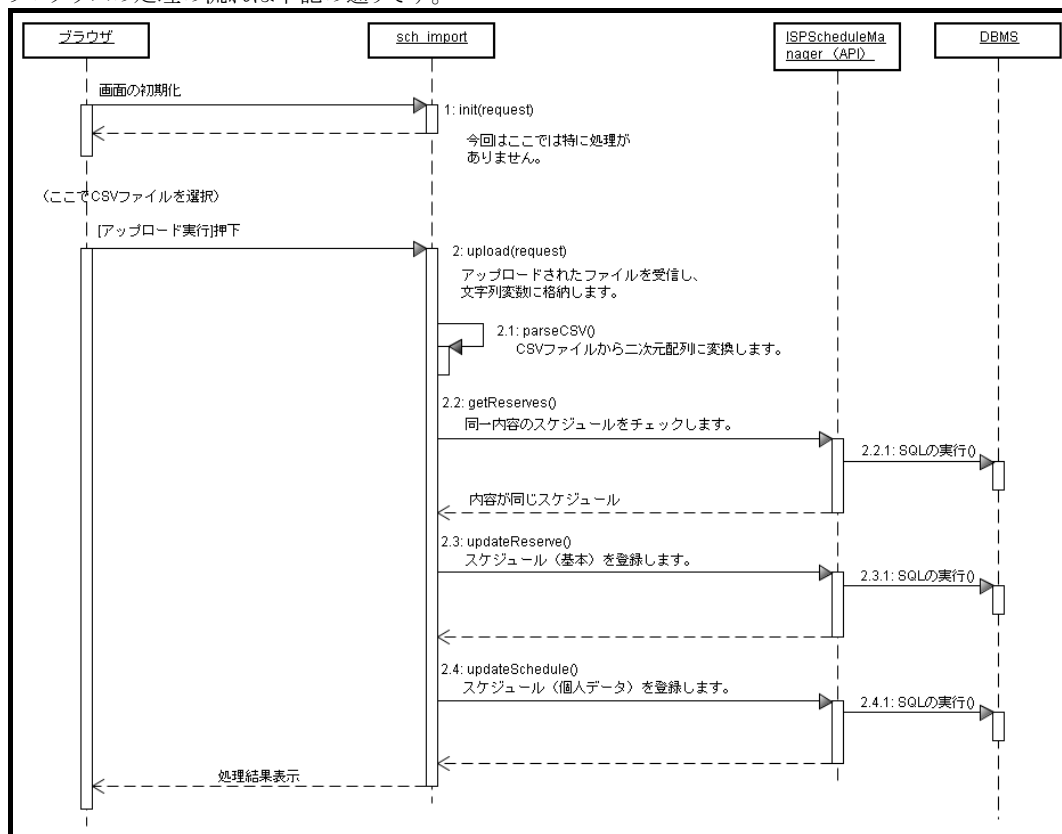
作成するツールの仕様は下記の通りとします。サンプルのため、ある程度機能を限定します。

- (1) Microsoft Outlook2003 からデフォルト設定で出力した CSV ファイルをそのまま取り込む。
- (2) 簡易ツールなので取り込む属性は下記のみとし、特に設定項目は含めない。
タイトル、開始日時、終了日時、ユーザ CD(参加者は含めない)、場所、説明
(取り込まない項目を CSV ファイルから取り除く必要はありません。)
- (3) 画面からクライアントローカルファイル(CSV)をアップロードすると即時に登録処理が実施される。
- (4) 内容に変更のないスケジュールは上書きする。

それでは、スケジュールデータインポート機能の作成手順を説明します。
大まかな手順は次のようになります。

- (1) プレゼンテーションページ(HTML)を作成する。
- (2) ファンクションコンテナ(JS)を作成する。
- (3) メニューを設定する。
- (4) 取り込みを実行する。

プログラムの処理の流れは下記の通りです。



♪ポイント♪

イントラネット・スタートパックのスケジュール関連 API「ISPScheduleManager」を利用することで、データベースアクセスのための SQL を記述する必要がありません。

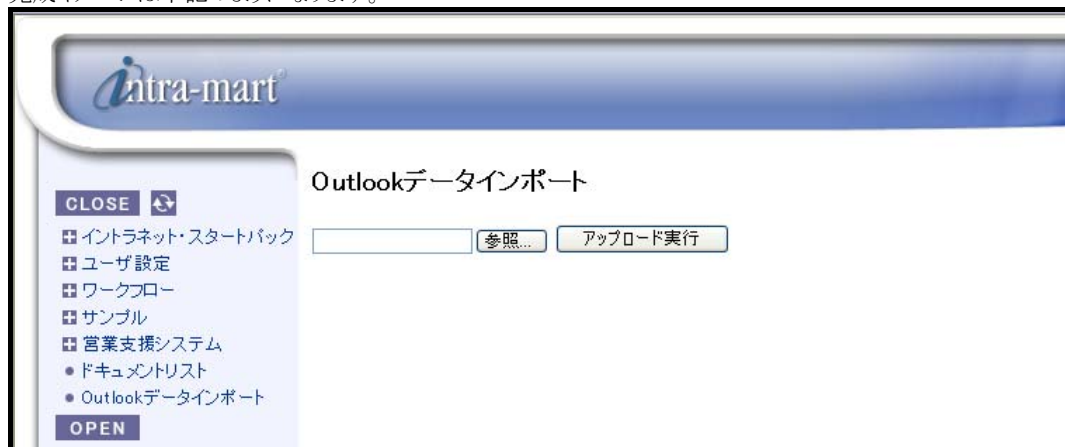
作成するファイルは下記の通りです。

ファイルパス	ファイル名	説明
(ROOT)/pages/src/sample/sch_import	sch_import.html	プレゼンテーションページ
(ROOT)/pages/src/sample/sch_import	sch_import.js	ファンクションコンテナ

(ROOT) は Resource Service ルートを示します。

- (1) プレゼンテーションページ(HTML)を作成します。

完成イメージは下記のようになります。



ソースコードの例を示します。

```
<HTML>
  <HEAD>
    <TITLE>Outlook データインポート</TITLE>
  </HEAD>
  <BODY>
    <H3>Outlook データインポート</H3>
    <IMART type="form" name="fileForm" method="post" enctype="multipart/form-data" action="upload">
      <INPUT type="file" name="upload_file">
      <INPUT type="button" value="アップロード実行" onclick="document.fileForm.submit();">
    </IMART>
  </BODY>
</HTML>
```

♪ポイント♪

ファイルをアップロードする際には、「enctype="multipart/form-data"」とする必要があります。

- (2) ファンクションコンテナ(JS)を作成します。

<2:upload(request)>

アップロードされたファイルを文字列変数として取得します。

```
// ファイルデータの取得
var sData = request.getParameter("upload_file").getValueAsStream();

// ファイルデータの文字コードを UNICODE に戻す。
sData = Unicode.from(sData,"MS932");

// 項目内部の改行はスペースに変換する。
sData = sData.replace(new RegExp("\r\n","g")," ");
```

♪ポイント♪

アップロードされたデータはクライアント上の文字コードのまま(MS932)なので、サーバ上で処理するために UNICODE に変換する必要があります。また、改行コードもここで統一します。

<2.1:parseCSV()>

CSV から二次元配列に変換します。

```
// CSV をパースし、二次元配列に格納する。
var aCSV = parseCSV(sData);
```

Outlook から出力したデータは CSV1 形式となっています。

< 2.2:getReserves() >

これから登録しようとするスケジュールと内容が同じデータが存在するか、検索します。

```
// スケジュールマネージャの生成
var oMNG = new ISPScheduleManager("default");
```

ここで、スケジュール管理システムのAPIを使うためのオブジェクトを初期化します。

以降、この oMNG を使用して処理を進めます。

```
//まったく同じスケジュールがあればスキップ
sWhere = "@startDate = ? AND @endDate = ? AND @title=? AND @scheduleUserCd=? AND @place=? AND @notes=?";
aParm = new Array();
aParm[0] = new DbParameter( DatabaseManager.dateToString(oReserve.startDate),DbParameter.TYPE_DATE );
aParm[1] = new DbParameter( DatabaseManager.dateToString(oReserve.endDate) ,DbParameter.TYPE_DATE );
aParm[2] = new DbParameter( oReserve.title,DbParameter.TYPE_STRING );
aParm[3] = new DbParameter( oSchedule.userCd,DbParameter.TYPE_STRING );
aParm[4] = new DbParameter(oReserve.place,DbParameter.TYPE_STRING );
aParm[5] = new DbParameter( oReserve.notes,DbParameter.TYPE_STRING );
oCheck = oMNG.getReserves(sWhere,aParm);

if (!oCheck.error && oCheck.data.length > 0)
{
    Debug.print("重複したスケジュールをスキップ");
    continue;
}
```

getReserves の引数に検索条件を指定します。

♪ポイント♪

検索条件は PreparedSQL の指定方式になっています。検索条件に指定するキーワード「@キーワード」の一覧は、イントラネット・スタートパックの API リストに記載されています。

< 2.3:updateReserve() >

スケジュール登録 API を呼び出します。

```
// 予定オブジェクト(Reserve)の作成
oReserve = new Object();
oReserve.reserveCd = Identifier.get();
oReserve.lumpTransactionId = Identifier.get();
oReserve.title = aCSV[i][COL_TITLE];
oReserve.startDate = getDate(aCSV[i][COL_START_DATE],aCSV[i][COL_START_TIME]);
oReserve.endDate = getDate(aCSV[i][COL_END_DATE],aCSV[i][COL_END_TIME]);
oReserve.place = aCSV[i][COL_PLACE];
oReserve.notes = aCSV[i][COL_NOTES];

// スケジュールを登録
oResult = oMNG.updateReserve(oReserve);
if (oResult.error)
{
    Debug.browse(oReserve,oResult);
} else
{
    Debug.print("スケジュール(基本)登録成功");
}
```

`updateReserve()`はスケジュールの基本情報を登録します。基本情報とは、スケジュールのタイトルや日時などの基本的な情報で、複数人が同時に参加するようなスケジュールにおいても基本情報は1レコード登録されます。

<2.4:updateSchedule()>

スケジュール登録(個人データ)API を呼び出します。

```
// 対象ユーザ毎オブジェクト(Schedule)の作成
oSchedule = new Object();
oSchedule.reserveCd = oReserve.reserveCd;
oSchedule.userCd = Module.client.get("id");
// Schedule オブジェクトを Reserve に格納
oReserve.Schedules = new Array();
oReserve.Schedules[0] = oSchedule;

// スケジュール(個人データ)を登録
oResult = oMNG.updateSchedule(oSchedule);
if (oResult.error)
{
    Debug.browse(oReserve,oResult);
} else
{
    Debug.print("スケジュール(個人データ)登録成功");
}
```

`updateSchedule()`はスケジュールの個人データ部分を登録します。個人データとは、スケジュールの対象者を指します。今回のプログラムではアップロード処理を実行している人のログイン ID を登録するようになっています。会議召集のように複数人が参加するスケジュールを登録する場合には、`updateSchedule()`を人数分実行する必要があります。

※処理の全体はダウンロードソースコードを参照してください。

- (3) メニューを設定します。

ログイングループ管理者でログインし、メニューから[ログイングループ管理]－[メニュー管理]－[メニュー設定]を選択してください。

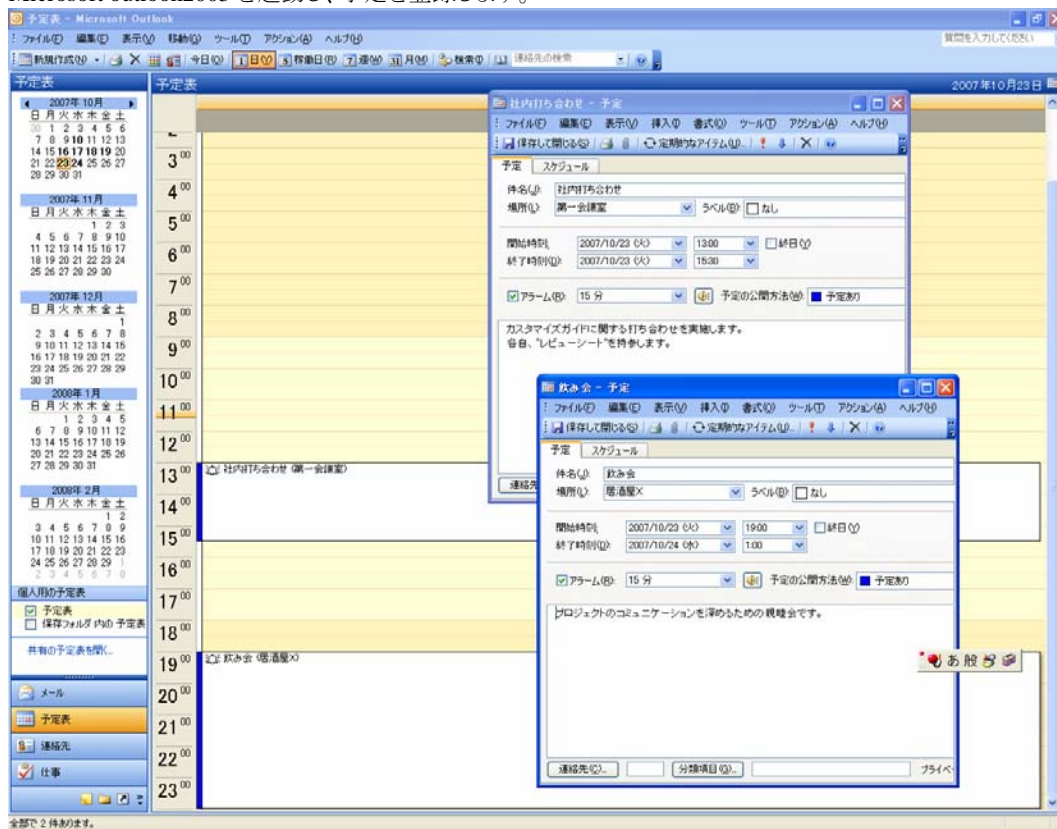
トップメニューとして下記の通り登録してください。

表示名	クライアントタイプ	URL	その他
Outlook データ インポート	パソコン	sample/sch_import/sch_import.jsp	クライアントからアクセスするのに必要なロールを設定してください。

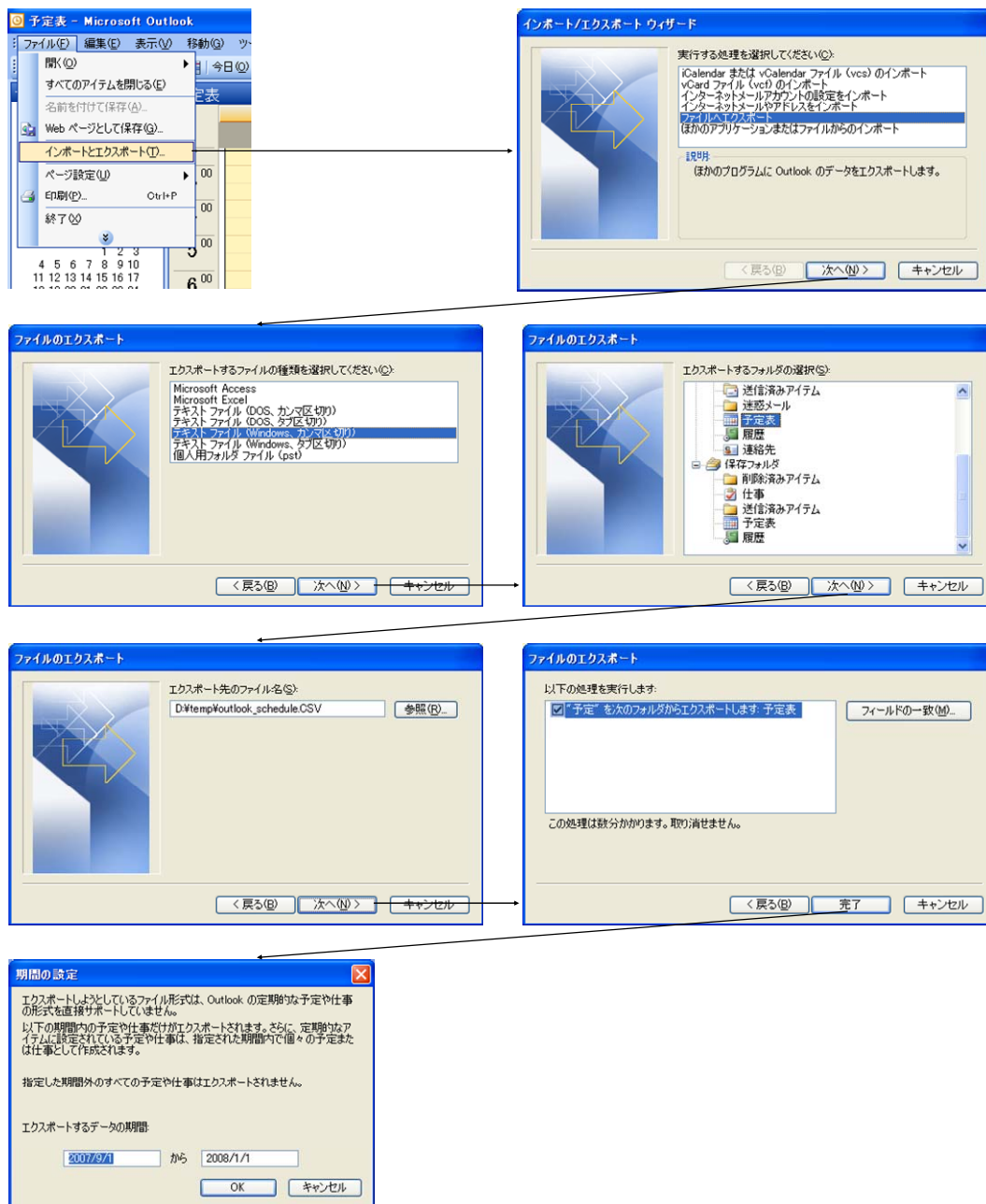
- (4) 取り込みを実行します

それでは、実際に取り込み処理を実行してみましょう。

Microsoft outlook2003 を起動し、予定を登録します。



CSVファイルにエクスポートします。



次のような CSV ファイルが出力されます。

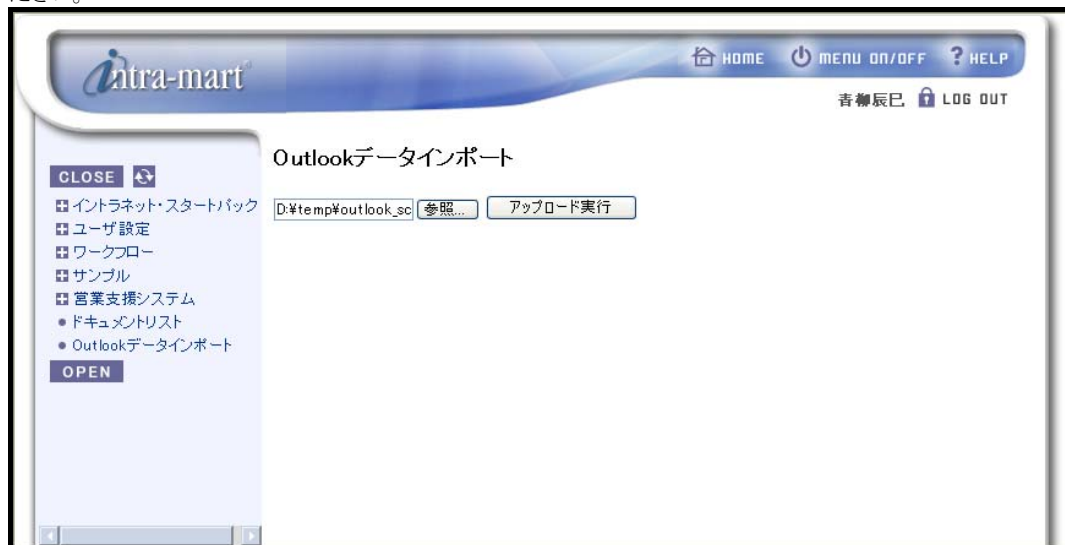
```

"件名","開始日","開始時刻","終了日","終了時刻","終日イベント","アラーム_オン/オフ","アラーム日付","アラーム時刻","会議の開催者","必須出席者","任意出席者","リ
社内打ち合わせ","2007/10/23","13:00:00","2007/10/23","15:30:00","False","True","2007/10/23","12:45:00",,,,,,"False",,"2",,"第一会議室","カスタマイズガイドに
各自,""レビュシート""を持参します。"
標準,"標準"
飲み会,"2007/10/23","19:00:00","2007/10/24","1:00:00","False","True","2007/10/23","18:45:00",,,,,,"False",,"2",,"居酒屋X","プロジェクトのコミュニケーション
標準,"標準"

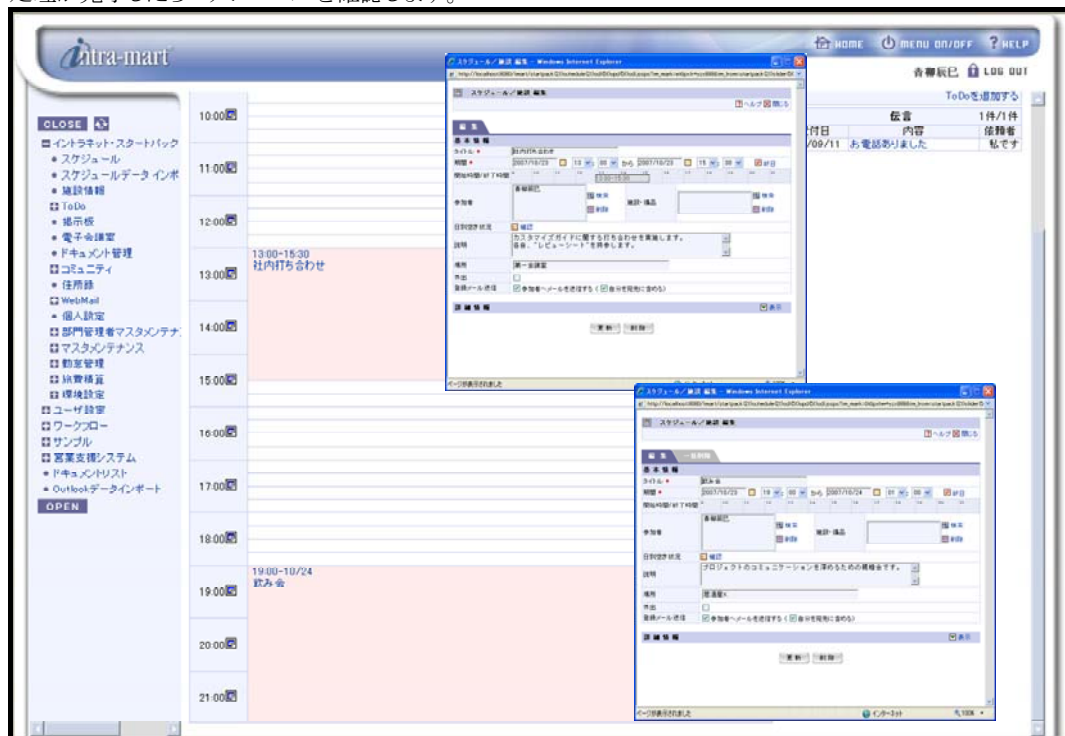
```

次に、CSV ファイルをスタートパックに取り込みます。

「参照」ボタンを押下し、ファイル選択画面で CSV ファイルを選択後、「アップロード実行」ボタンを押下してください。



処理が完了したらスケジュールを確認します。



(※ポップアップ画面は紙面の都合上、若干縮小して合成しています。)

3.2 ドキュメント管理システムのカスタマイズ

3.2.1 ドキュメントの管理属性を追加する

イントラネット・スタートパックのドキュメント管理システムは、標準では必要最低限の属性のみ登録できるようになっています。標準で登録できる属性は、

- ・ 添付ファイル名
- ・ ドキュメント名
- ・ 添付ファイル名
- ・ 登録情報(登録者・日時など)
- ・ 備考
- ・ ソート番号

に限られます。しかし実際に文書共有を実施する際には、管理上の様々な属性付与が必要になるケースがあります。例えば、内部統制のために主管部署や保存年限に関する項目を指定することなどが想定されます。ドキュメント管理システムでは、このような要件に対応するための汎用的な属性をあらかじめ用意しています。区分コード設定マスタで下記の項目を編集することで、利用が可能となります。

それでは、ドキュメント管理属性として新たに「主管部署」という属性を追加する例を示します。

ドキュメント登録

[フォルダ参照画面へ](#) [ヘルプ](#)

フォルダ: 契約書/基本契約

ドキュメント名*:

配信設定: ☒ 配信なし

添付ファイル	ファイル名	サイズ	編集	削除
追加				

関連ドキュメント	フォルダ名	ドキュメント名	関連解除
追加			

登録者: 青柳辰巳

備考:

ソート番号: 0

[登録](#)

ここに「主管部署」という入力項目を追加します。

メニューから[イントラネット・スタートパック]-[マスタメンテナンス]-[区分コード設定マスタメンテナンス]を選択してください。

区分コード設定マスタメンテナンス

新規登録 一覧

スタートバック共通
掲示板
勤怠管理
旅費精算
施設予約管理
ドキュメント管理
スケジュール管理
ToDo(伝言メモ)
住所録
営業日報管理
電子会議室
コミュニティ
WebMail

アプリケーションコード: document
アプリケーション名: ドキュメント管理
ロケール: 日本語(Windows-31J)

区分名称	区分コード	値
doc_etc_word	01	OFF 文字列 1 使用ON、未使用OFF キャプション
doc_etc_word	02	OFF 文字列 2 使用ON、未使用OFF キャプション
doc_etc_word	03	OFF 文字列 3 使用ON、未使用OFF キャプション
doc_initial	document_cb_max_record	1000 CBLコード検索(IM-統合検索オプション使用の場合)
doc_initial	document_use_cbsearch	OFF ファイル内容検索(IM-統合検索オプションが必

この中の区分名称「doc_etc_word」、区分コード「01」を選択して編集画面を開きます。

区分コード更新・削除

区分コード一覧 最新情報 ヘルプ

アプリケーションコード: document
アプリケーション名: ドキュメント管理
区分名称: doc_etc_word
区分コード: 01
ロケール: 日本語(Windows-31J)
値: OFF||文字列1
説明: 使用:ON、未使用:OFF||キャプション
ソートキー: 1

区分コード更新・削除

区分コード一覧 最新情報 ヘルプ

アプリケーションコード: document
アプリケーション名: ドキュメント管理
区分名称: doc_etc_word
区分コード: 01
ロケール: 日本語(Windows-31J)
値: ON||主管部署
説明: 使用:ON、未使用:OFF||キャプション
ソートキー: 1

値を「ON||主管部署」に更新します。

下記の表示に変更されます。

ドキュメント登録				
フォルダ参照画面へ				ヘルプ
フォルダ	契約書/基本契約			
ドキュメント名 *	<input type="text"/>			
配信設定	<input checked="" type="checkbox"/> 配信なし			
添付ファイル	ファイル名	サイズ	編集	削除
<input checked="" type="checkbox"/> 追加				
関連ドキュメント	フォルダ名	ドキュメント名	関連解除	
<input checked="" type="checkbox"/> 追加				
登録者	青柳辰巳			
備考	<input type="text"/>			
主管部署	<input type="text"/>			
ソート番号	<input type="text" value="0"/>			
登録				

このように、あらかじめ用意されている範囲で属性の表示を ON/OFF することができます。
区分設定の一覧については下記を参照してください。



区分コードの一覧は下記を参照してください。

⇒[4.2区分コード一覧](#)

3.2.2 登録されているドキュメントの一覧を出力する

ドキュメント管理システムに登録されているドキュメントの一覧を作成します。ドキュメント管理システムの API (ISPDocumentManager) を使い、ドキュメント管理システムのデータベースから簡単にデータを取得します。

作成ツールの仕様は下記の通りとします。

- (1) 登録されてから1年経過したドキュメントを検索する。
ここでは簡単に、システム日付の年部分を単純に-1します。

- (2) 検索結果を画面に一覧表示する。
表示内容も簡単にします。

ドキュメント名	登録日
ABC社基本契約	Tue Sep 11 2007 18:10:15 GMT+0900 (JST)
設計書雛形B	Thu Oct 18 2007 21:03:20 GMT+0900 (JST)

それでは、ドキュメント一覧機能の作成手順を説明します。
大まかな手順は次のようになります。

- (1) プレゼンテーションページ(HTML)を作成する。
- (2) ファンクションコンテナ(JS)を作成する。
- (3) メニューを設定する。

作成するファイルは下記の通りです。

ファイルパス	ファイル名	説明
(ROOT)/pages/src/sample/document_list/	document_list.html	プレゼンテーションページ
(ROOT)/pages/src/sample/ document_list/	document_list.js	ファンクションコンテナ

(ROOT) は Resource Service ルートを示します。

- (1) プレゼンテーションページの例

```

<html>
  <body>
    <TABLE border>
      <TR>
        <TD>ドキュメント名
        </TD>
        <TD>登録日
        </TD>
      </TR>
      <IMART type="repeat" list= oBind.documents item="record">
        <TR>
          <TD><IMART type="string" value= record.documentName></IMART>
          </TD>
          <TD><IMART type="string" value= record.updateDate></IMART>
          </TD>
        </TR>
      </IMART>
    </TABLE>
  </body>
</html>

```

(2) ファンクションコンテナの例

```

var oBind = new Object();

function init(request)
{
  // ドキュメントデータのアクセッサオブジェクトを生成する。
  var oMng = new ISPDDocumentManager("default");

  // 一年前の日付を取得する。
  var dToday = new Date();
  var dLimit = new Date(dToday.getFullYear()-1,dToday.getMonth(),dToday.getDate());
  var sLimit = DatabaseManager.dateToString(dLimit);

  // API のパラメータを作成する。
  var sWhere = "@recordDate >    ?";
  var aParm = new Array();
  aParm[0] = new DbParameter( sLimit,DbParameter.TYPE_DATE );      // Where 句パラメータの作成

  // API を利用してデータを取得する。
  var oData = oMng.getDocuments(sWhere,aParm);

  oBind.documents= oData.data;
}

```

getDocuments の引数に検索条件を指定します。

♪ポイント♪

検索条件は PreparedSQL の指定方式になっています。

(3) メニューの設定

表示名	クライアントタイプ	URL	その他
ドキュメント一覧	パソコン	sample/document_list/docum ent_list.jssp	クライアントからアクセス するのに必要なロールを 設定してください。

3.2.3 API を使ってドキュメントを登録する

ドキュメント管理システムの API を使って、プログラムのドキュメントを登録するサンプルを示します。この手法を応用すると、他のシステムからのドキュメントマイグレーションやファイルサーバからの一括インポートなどの機能を実現することができるでしょう。

ドキュメントを登録するための API には、決まった処理の手順を踏む必要があります。本資料ではドキュメントを登録するためのもっとも基本的な API 呼び出しの手順を示します。

作成ツールの仕様は下記の通りとします。

- (1) 画面から、アップロードするファイルとアップロード先のフォルダ名を指定する。
- (2) 指定されたフォルダにドキュメントを登録する。
 - (ア) フォルダがない、または同名のフォルダが複数存在する場合はエラー。

それでは、ドキュメント登録機能の作成手順を説明します。

大まかな手順は次のようになります。

- (1) プレゼンテーションページ(HTML)を作成する。
- (2) ファンクションコンテナ(JS)を作成する。
- (3) メニューに登録する。

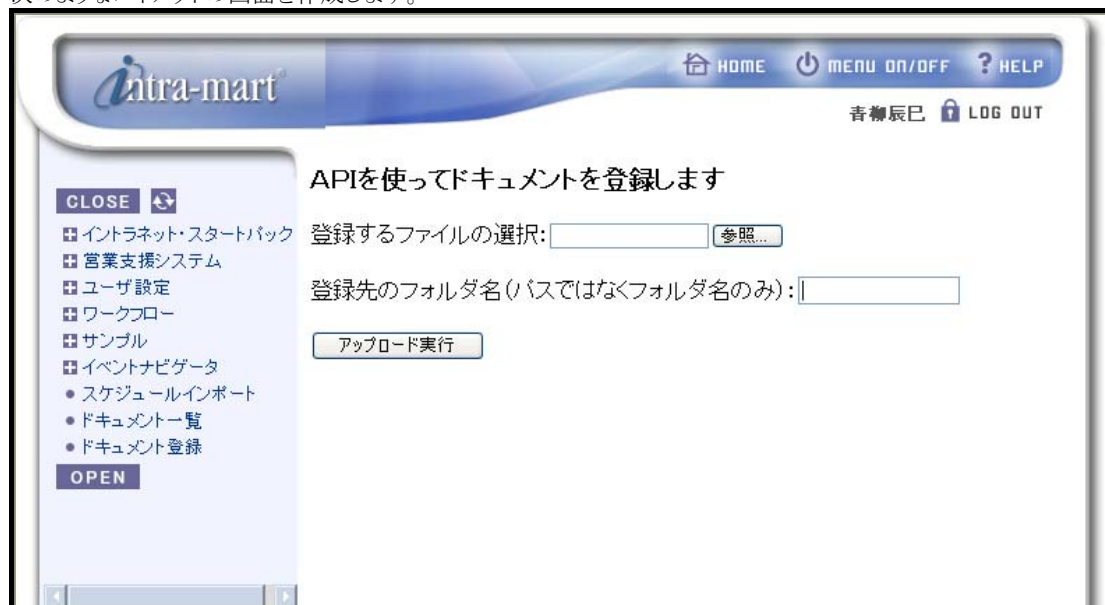
作成するファイルは下記の通りです。

ファイルパス	ファイル名	説明
(ROOT)/pages/src/sample/document_rgst/	document_rgst.html	プレゼンテーションページ
(ROOT)pages/src/sample/ document_rgst/	document_rgst.js	ファンクションコンテナ

(ROOT) は Resource Service ルートを示します。

- (1) プレゼンテーションページ(HTML)を作成する。

次のようなレイアウトの画面を作成します。



ソースコードの例を示します。

```
<HTML>
  <HEAD>
    <TITLE>APIを使ってドキュメントを登録する</TITLE>
  </HEAD>
  <BODY>
    <H3>APIを使ってドキュメントを登録します</H3>
    <IMART type="form" name="fileForm" method="post" enctype="multipart/form-data"
action="upload">
      登録するファイルの選択:<INPUT type="file" name="upload_file"><BR><BR>
      登録先のフォルダ名 (パスではなくフォルダ名のみ):<INPUT type="text" name="folder_name"
value=""><BR><BR>
      <input type="button" value="アップロード実行" onclick="document.fileForm.file_name.value =
document.fileForm.upload_file.value;document.fileForm.submit();">
      <input type="hidden" name="file_name" value="">
    </IMART>
  </BODY>
</HTML>
```

ファイルの中身は「upload_file」、フォルダ名は「folder_name」、ファイル名は「file_name」にそれぞれ格納して、ファンクションコンテナを呼び出します。

(2) ファンクションコンテナ(JS)を作成します。

ここでは、前項までの内容でも繰り返し行ってきたイントラネット・スタートパックのAPIを利用して、SQLを直接発行せずにドキュメント管理システムにデータを登録することができます。しかし、ドキュメント管理システムには添付ファイルがあるため、その登録方法には決まった手順があります。

主な注意点を挙げますと、

- ① ドキュメントを登録するオブジェクトに、あらかじめ生成しておいた添付ファイル情報オブジェクトを「ぶら下げて」登録する必要がある。(ドキュメントオブジェクトの属性として添付ファイルオブジェクトを設定する)
- ② 添付ファイルをストレージサービスに登録する時のファイル名を、「論理ファイル名(ユーザが見たままのファイル名)にする」か、「内部ID(一意のID)」として登録するか、二種類のモードがあり、システム変数から取得して適切に処理する必要がある。
- ③ APIではデータベースへのデータ登録までしかされないため、ストレージサービスへのファイル保存は VirtualFile オブジェクトを使って自分で登録する必要がある。

それでは、特徴的な部分を抜粋して紹介します。

まず、添付ファイルオブジェクトについて

```
// 添付ファイルオブジェクトの生成
// 添付ファイル情報の作成
var oAttachInfo = new Object();

// ini ファイルよりファイルの保存タイプ取得
var sSaveType = System.getValue("STARTPACK_FILE_UPLOAD_SAVE_TYPE", "id");

// 添付ファイル情報設定
oAttachInfo.fileCd = Identifier.get();

if(sSaveType == "id") {
    oAttachInfo.physicalFileName = Identifier.get();
} else {
    oAttachInfo.physicalFileName = sFileName;
}

oAttachInfo.logicalFileName = sFileName;
```

添付ファイルオブジェクトは API ではなく、「new Object()」で取得したオブジェクトに属性を付与して作成します。そして、STARTPACK_FILE_UPLOAD_SAVE_TYPE の値が”id”(ID で保存する)か空白(ユーザの見た目通り保存する)によって physicalFileName(論理ファイル名)の値を分岐しています。

添付ファイルオブジェクトは、次のようにドキュメント情報オブジェクトに「ぶら下げます」

```
// ドキュメントオブジェクトの生成と登録
var oDocumentInfo = new Object();
oDocumentInfo.folderCd = oFolderInfo.data[0].folderCd;
oDocumentInfo.documentCd = Identifier.get();
oDocumentInfo.documentName = sFileName;
oDocumentInfo.notes = "";
oDocumentInfo.attachedFiles = new Array();
oDocumentInfo.attachedFiles[0] = oAttachInfo;
```

こうしておいてからドキュメントオブジェクトを実際に登録します。

```
// ドキュメント情報登録
var oResult = oDM.updateDocument(oDocumentInfo);
```

(ここでは oDM は ISPDDocumentManager が格納されています)

これでデータベースへのデータの登録は完了したのですが、ストレージサービスへの添付ファイルの登録は完了していません。次のように VirtualFile オブジェクトを利用して添付ファイルを登録します。

```
// Storage Server の正しい場所に実ファイルを保存する。
var sReleasePath = "startpack/fsroot/" + Module.client.get("group") + "/document/"
    + oDocumentInfo.documentCd + "/" + oAttachInfo.fileCd + "/";

var oTargetDir = new VirtualFile(sReleasePath);
if(!oTargetDir.exist()) {
    if(!oTargetDir.makeDirectories()) {
        // ディレクトリ作成失敗
        Module.alert.write("SYSTEM.ERR", "ディレクトリの作成に失敗しました。");
    }
}
var oTargetFile = new VirtualFile(sReleasePath + oAttachInfo.physicalFileName);
if(!oTargetFile.save(sData)) {
    // ファイル出力失敗
    Module.alert.write("SYSTEM.ERR", "ファイルの保存に失敗しました。");
}
```

ドキュメント管理システムのフォルダ構成と、ストレージサービス内の物理ファイルの構成は決まったルールがあります。
 %ストレージサービスルート%/startpack/fsroot/%ログイングループCD%/document/%ドキュメントCD%/添付ファイルCD/

また、登録するためのフォルダもその都度作成する必要があります。

フォルダを作成 (makeDirectories) して、ファイルの保存 (save) が成功すれば、ドキュメントの登録完了です。

(3) メニューの設定

表示名	クライアントタイプ	URL	その他
ドキュメント登録	パソコン	sample/document_rgst/document _rgst.jssp	クライアントからアクセスする のに必要なロールを設定 してください。

3.3 電子会議室のカスタマイズ

3.3.1 登録可能な文字数を拡張する

電子会議室の登録文字数のデフォルト設定は 1,000 文字 (半角文字で 2,000 桁) です。1,000 文字を超える長さのデータを登録使用とした場合、下記のようなメッセージが表示されて登録することができません。

内容 最大入力桁数は1000桁です

会議室種別: 技術関連/Ajax関連
 タイトル: Ajax勉強会の告知
 内容: Ajax勉強会の開催
 2007/12/14 香柳 辰巳
 2008年1月30日の午後2時から午後4時まで
 第2会議室にてIntra-martで使用されているAjax技術の勉強会を開催
 します。
 ぜひ、時間を空けて最新の技術・トレンドを学んでください。
 目次と内容は次の通りです。
 <目次>

添付ファイル: 追加 (ファイル名, サイズ, 削除)

配信設定: ☒ 配信なし

確認 更新 削除

タイトル	投稿者	投稿日時
Ajax勉強会の告知	香柳辰巳	2007/12/14 (金) 17:45

イントラネット・スタートパックでは、このような登録可能文字数の設定は一元化されており、「isp_input_validation.xml」というファイルに記載することになっています。プログラム内部ではこの設定を参照して入力文字数のチェックを行っています。

```

2926 <table>↓
2927 <table-name>ispt_conference_info</table-name>↓
2928 <field>↓
2929 <field-name>conference_cls_cd</field-name>↓
2930 <field-type>VARCHAR</field-type>↓
2931 <max-length>20</max-length>↓
2932 </field>↓
2933 <field>↓
2934 <field-name>conference_info_cd</field-name>↓
2935 <field-type>VARCHAR</field-type>↓
2936 <max-length>20</max-length>↓
2937 </field>↓
2938 <field>↓
2939 <field-name>thread_cd</field-name>↓
2940 <field-type>VARCHAR</field-type>↓
2941 <max-length>20</max-length>↓
2942 </field>↓
2943 <field>↓
2944 <field-name>parent_cd</field-name>↓
2945 <field-type>VARCHAR</field-type>↓
2946 <max-length>20</max-length>↓
2947 </field>↓
2948 <field>↓
2949 <field-name>title</field-name>↓
2950 <field-type>VARCHAR</field-type>↓
2951 <max-length>255</max-length>↓
2952 </field>↓
2953 <field>↓
2954 <field-name>contents</field-name>↓
2955 <field-type>VARCHAR</field-type>↓
2956 <max-length>2000</max-length>↓
2957 </field>↓
2958 <field>↓
2959 <field-name>contents_html</field-name>↓
2960 <field-type>VARCHAR</field-type>↓
2961 <max-length>2000</max-length>↓
2962 </field>↓
2963 </table>↓

```

簡単には、この設定を変更すれば良いことになりますが、実際にはデータベースのカラムサイズも合わせて拡張する必要があります。

電子会議室の登録可能文字数を拡張するためには次の手順を実施します。

- (1) データベースのフィールドサイズを拡張する。
- (2) プログラム内での文字数チェック設定を変更する(isp_input_validation.xml)。
- (3) イントラマートを再起動する。

※ 注意：電子掲示板の場合には、通常テキスト(contents)と書式を保持したHTMLテキスト(contents_html)の2箇所変更が必要。また、登録可能な文字数は書式を表現するためのタグ付のHTML形式でも制限されるので、書式付のHTML(contents_html)を多少大きくしておいた方がよい。

ここでは拡張後の登録可能文字数を半角 3,000 文字、書式付で 4,000 文字として説明します。

- (1) データベースのフィールドサイズを拡張する。

次の ALTER 文を、DBMS の SQL クエリー実行ツールで実行してください。

```
alter table ispt_conference_info modify (
contents varchar2(3000);
)
```

```
alter table ispt_conference_info modify (
contents_html varchar2(4000));
```

(ALTER 文の書式は DBMS により多少異なります。詳しくは各 DBMS の説明書を参照してください。上記は ORACLE の例です。)

- (2) プログラム内での文字数チェック設定を変更する。

isp-input-validation.xml を編集し、登録可能文字数を変更します。

ファイルの場所: Resource Service インストールフォルダ/conf/isp-input-validation.xml

ノードの場所:

/root/isp-input-validation/conference/table[ispt_conference_info]/field/field-name[contents]

/root/isp-input-validation/conference/table[ispt_conference_info]/field/field-name[contents_html]

```

2953 <field>↓
2954 <field-name>contents</field-name>↓
2955 <field-type>VARCHAR</field-type>↓
2956 <max-length>2000</max-length>↓
2957 </field>↓          3000
2958 <field>↓
2959 <field-name>contents_html</field-name>↓
2960 <field-type>VARCHAR</field-type>↓
2961 <max-length>2000</max-length>↓
2962 </field>↓          4000

```

(3) イントラマートを再起動する。

起動完了後には、全角 1,000 文字を超えるテキストも正常に登録できるようになります。

3.4 ポートレットの作成

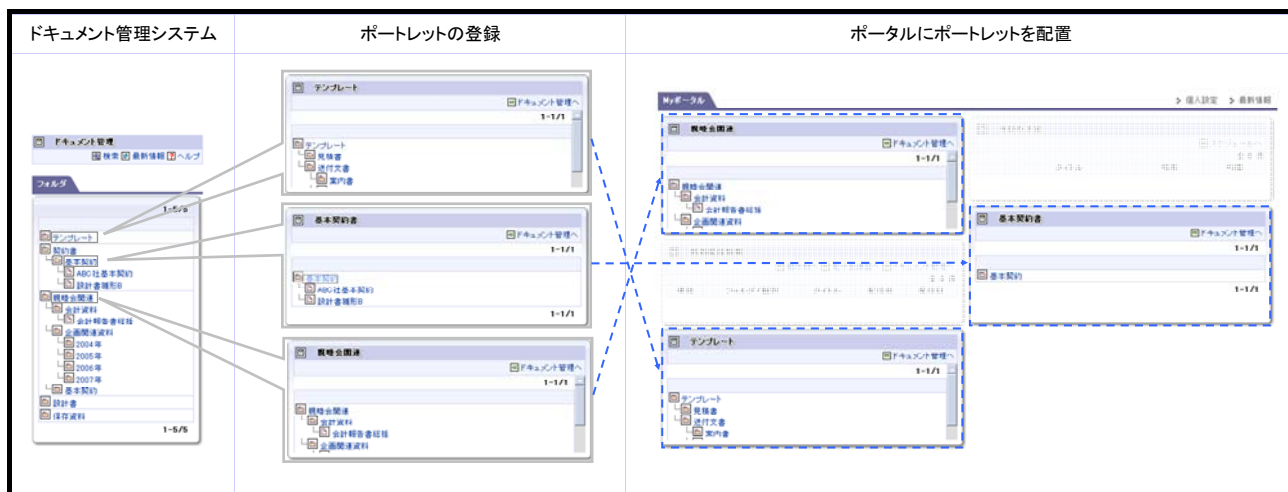
3.4.1 ドキュメントフォルダ・ポートレットを作成する

イントラネット・スタートパックのドキュメント管理システムは、社内の共有ドキュメントを統合的なフォルダ階層に保存します。ユーザは必要なドキュメントを利用する際には分かりやすく整理されたフォルダ階層にしたがって検索して行けば、目的のドキュメントにたどり着くことができます。しかし、毎日決まって使うようなドキュメントにアクセスする場合には、ドキュメント管理システムのフォルダ階層を辿るのは少々面倒な作業です。あるユーザが日常的にアクセスするフォルダは、全体のうちの一部に限られているのが普通だと考えられます。

そこで、「ドキュメント管理システム内の指定したフォルダをポータルに表示するためのポートレット」を作成します。

ここでも、既存のソースコードをうまく流用しながらカスタマイズしてこの機能を実装していきます。

概念図は下記の通りです。詳細は以降説明をしていきます。



作成するポートレットの仕様は下記の通りとします。

- (1) 表示対象のフォルダをページ引数で指定する
ドキュメント管理システムで管理しているフォルダの中から、ページ引数で指定したフォルダをトップ階層とする、フォルダ・ドキュメント階層ツリーを表示する。
- (2) ページ引数の仕様
「target_folder=[階層表現],[フォルダ名]」という表現とする。
階層表現はトップを「/」とし、以下フォルダ階層ごとに「フォルダ名+」を加える。
(例:「親睦会/企画画関連資料/2004 年」なら「target_folder=/親睦会/企画画関連資料/2004 年」)
- (3) 動作の仕様
ツリーの開閉動作はドキュメント管理システムと同様とする。
ドキュメントを選択すると、ポップアップ画面を開いてドキュメント情報の詳細を表示する。
- (4) 制限事項
フォルダ階層に「,」が含まれると正しく動作しないが、今回は制限事項とする。

※フォルダの指定にページ引数を用いることにより、1つのポートレットプログラムを、異なるフォルダ・ドキュメント階層ツリーを表示するポートレットとして用いることができます。

ポートレットの見た目の完成イメージは次のようになります。

(「親睦会関連」フォルダを表示フォルダに指定した例)



ポートレットの作成に関するルールと詳細な手引きについては下記の資料に記載されていますので割愛します。
記載先:「intra-mart Ver6.1 APIリスト」のJavaEE 開発モデル、スクリプト開発モデルそれぞれの中の「業務基盤ツール／ポータルモジュール」の項

手順についても＜プレゼンテーションページの作成＞→＜ファンクションコンテナの作成＞→＜メニュー／ポートレット登録＞という基本的な流れは同様です。

本項では、画面上の各要素を既存のソースコードから流用する手法を中心に説明します。

今回作成する機能を効率よく開発するために、下記のように既存機能を流用します。
(詳細はダウンロードソースコードを参照してください。ここでは考え方に絞って説明します。)

参照ウィンドウはコピーせず、
既存のものをそのままコールする。

流用のポイント

- ①ポートレットのヘッダ部分のレイアウトと、ドキュメント管理に遷移する仕組みを流用
- ②フォルダの階層表示をする仕組みを流用
- ③ドキュメント名をクリックするとドキュメント参照ウィンドウを開く仕組みを流用

また今回のポートレットは、ひとつのポータルの中に複数配置される可能性があるため、フォーム名やスクリプト名等がポータル画面内で干渉しないようにする必要があります。干渉を避ける方法としては IFRAME を利用するのが簡単です。そこで、今回はポートレットの中身を別ファイルとして用意し、IFRAME の中に取り込むようにしています。

カスタマイズの手順は下記の様になります。

- (1) プレゼンテーションページ(HTML)を作成する。
- (2) ファンクションコンテナ(JS)を作成する。
- (3) ポートレットとしてプログラムを登録する。
- (4) ポータルにポートレットを配置する。

作成するファイルと流用元ファイルは下記の通りです。

ファイルパス	ファイル名	説明
(ROOT)/pages/src/sample/custom_portlet/	iframe.html iframe.js	IFRAME 用 また、このページをポートレットに指定する。
(ROOT)/pages/src/sample/custom_portlet /	custom_portlet.html custom_portlet.js	ポートレットの中身 (流用元) startpack/portal/ptl_memo
(ROOT)/pages/src/sample/custom_portlet /	tree.html tree.js	ツリーの表示、制御 (流用元) startpack/common/tree/cm_tree_app
(ROOT)/pages/src/sample/custom_portlet /	tree_data.html tree_data.js	ツリーのデータ取得(AJAX の非同期応答用プログラム) (流用元) startpack/common/tree/cm_tree_appData

(ROOT) は Resource Service ルートを示します。

① ポートレットのヘッダ部分

ポートレットのヘッダ部分については、「新着配信機能」を流用することになります。

ポートレットの枠とタイトルについてはポータルシステムが自動的に生成します。作成しなければならないのはツールバー部分(①の点線の部分)となります。「ドキュメント管理へ」をクリックするとドキュメント管理システムへ遷移します。この部分も「新着配信機能」のソースコードがそのまま流用できます。

掲示板や電子会議室に関連するコードは削除します。

ツールバー部分のソースコードは次のようになります。

```
<IMART type="imDesignCss" ></IMART>
<IMART type = "imToolBarFrame">
  <IMART type = "imToolBarRight">
    <IMART type = "imIcon"
      name = "ドキュメント管理へ"
      icon = "images/standard/arrow_left.gif"
      href = "JavaScript:onToDocument100;">
    </IMART>
  </IMART>
</IMART>
```

なお、クリックイベントに対応するため「onToDocument100()」「openWindow7()」「docDetailSubmit7()」の各メソッドも合わせてコピーする必要があります。

② フォルダ階層を表示する仕組み

ドキュメント管理システム標準のフォルダ階層表示機能を流用します。実は、このフォルダ階層表示機

能の中では『アクセス権がないフォルダは見えなくする機能』が組み込まれていますので、そのまま流用します。流用元のプログラムでは、フォルダのトップ階層をすべて取得し、その配下のフォルダ階層を表示するようになっているため、「フォルダのトップ階層をすべて取得」している部分を「指定したフォルダを1つ取得」するように動作変更します。

具体的には、既存の `getTopFolder()` の代わりに新しく `getTargetFolder()` を追加し、差し替えるようにします。

```
tree.js 55 行目付近
// oData          = getTopFolder(session, target_process, acl_flg, page_pos, view_max,
community_flg, community_manager_flg);

oData = getTargetFolder(request.target_folder_path,request.target_folder);
```

`request.target_folder_path` と `request.target_folder` は、ポートレットのページ引数として `iframe.js` に渡されますので、`iframe.js` から `tree.js` へ順次値を引き継ぐ必要があります。

`getTargetFolder()` の中身では、

```
tree.js 270 行目付近
oMNG = new ISPDDocumentManager();
sWhere = "@folderName = ? ";
aParm[0] = new DbParameter( sTargetFolder,DbParameter.TYPE_STRING );
oSelectData = oMNG.getDocumentFolders(sWhere,aParm);
```

のように、スタートバックの API を利用してドキュメント管理システムのデータベースにアクセスしています。ドキュメントのフォルダのフルパスを指定した検索は API ではサポートされていないので、まずフォルダ名のみ的一致で検索し、検索結果から親フォルダの情報が取得できますのでパスの一致をチェックする処理を加えています。

③ ドキュメント名をクリックするとドキュメント参照ウィンドウを開く仕組み

この機能は新着配信情報のポートレットで実現されていますので、そのままソースコードを流用します。ただし、クリックイベントを発生する場所がフォルダ階層ツリーの中になるため、その部分のイベント処理を加える必要があります。

フォルダ階層をクリックした時のイベントハンドラに、対象がドキュメントかフォルダかを示すフラグを追加します。

```
tree_data.js 313 行目付近
strList = strList + "<a href=¥"javascript:¥" onclick=¥"serchList("
+ "" + oData[cnt].id + "," +
+ oData[cnt].info_cd + "," +
+ oData[cnt].parent_cd + "," +
+ oData[cnt].category_depth + "," +
+ oData[cnt].category_cls + "," +
// + oData[cnt].wkf_flag + "," +
+ oData[cnt].wkf_flag + "," + "" + ((oData[cnt].category_cls != "cate" &&
reqData.target_process == "DC") ? "1":"0") + ""¥">"
+ strSpace + strImg
+ Module.string.browse(oData[cnt].name) + "</a>"
```

そして、フォルダ階層ツリーの `onClick` イベントハンドラ内でそのフラグを参照して、もし対象がフォルダではなくドキュメントであれば、流用してきたメソッドをコールするようにします。

```
tree.html 36行目付近
// ドキュメントならウィンドウを開く
if (isDocument == "1") docDetailSubmit7(parent_cd, info_cd);
```

これで、ドキュメントをクリックした時に新しいポップアップウィンドウが開き、その中にドキュメントの詳細を表示できるようになります。

次に、完成したプログラムを利用できるように設定します。

① ポートレットの登録

グループ管理者でログインし、メニューから[ポータル]―[ポートレット]を選択してください。
下記の方法で必要な分だけポートレットを登録します。

ポートレット編集
[一覧へ戻る](#)

基本設定

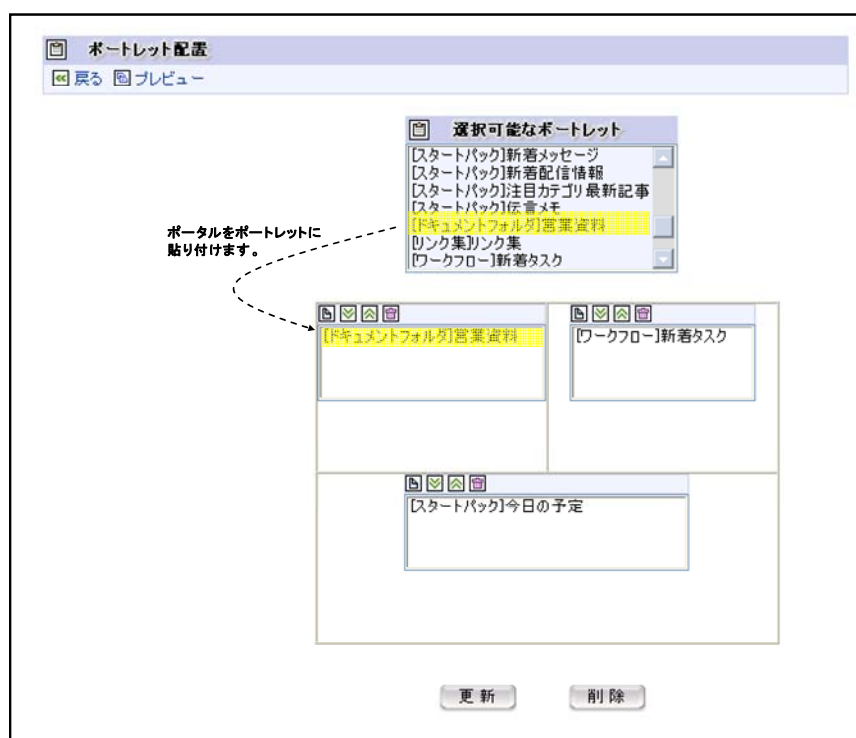
アプリケーション	ドキュメントフォルダ	自由に指定してください。
名称(必須)	営業資料	ここも自由で良いですが、フォルダ名がわかるようにつけてください。
ページ種別(必須)	Presentation Page	
画面パスあるいはURL種類	sample/custom_portlet/iframe	固定です。
ページ引数	target_folder=/営業資料	対象フォルダを指定してください。 (指定方法は前述の通り)

オプション設定

タイトルの表示	<input checked="" type="radio"/> する <input type="radio"/> 使用しない	自由に指定してください。
キャッシュの設定	なし	
ポートレットの説明		
表示領域(高さ: 幅)	単位 % :	

② ポータルを作成して、ポートレットを貼り付ける

グループポータルやユーザポータルの設定画面で、ポートレットを貼り付けます。



- ③ 動作を確認する。
設定したポータルが参照できるユーザでログインします。



ポータルが表示されれば完成です。

3.5 ワークフロー連携

3.5.1 掲示板掲示前承認機能の作成

intra-mart WebPlatform の利用において、ワークフローエンジンとの連携は大きな魅力のひとつです。本項では、掲示板システムとワークフローエンジンを連携する方法を通して、ワークフローAPI の使用方法を説明します。実際に、アプリケーションをワークフローエンジンと連携させるためには検討箇所が多岐に渡ります(「起票方法」「承認時の確認機能」「差し戻し時の編集機能」「最終承認時の処理」等)。ここでは紙面の都合上「起票方法」と「最終承認時の処理」に絞って説明します。

今回カスタマイズする仕様は下記の通りです。

- (1) 掲示板種別を「未承認」と「承認済み」の2つ作成し、「未承認」掲示板に登録した情報が承認されると「承認済み」掲示板に移動することにする。また、それぞれの掲示板のアクセス権を設定する。
例)
「未承認」→ 特定部門のみ編集可能(特定部門は例えば、総務部や管理部などを想定する。)
「承認済み」→ 全公開(参照のみ)
- (2) 掲示板参照画面に「申請」ボタンを追加し、押下するとワークフローAPIを使って申請する。
(今回はワークフローAPI でダイレクトに申請しますので、ワークフローモジュール標準の申請画面からの申請はサポートしないこととします。)
- (3) 承認時の確認画面には、掲示板参照画面を流用する。
- (4) 最終承認時の後処理で、掲示板種別を「承認前」から「承認後」に移動する。

業務で運用するには足りない機能もありますが、ワークフローAPIの利用イメージを示すために必要最低限な機能の記述にとどめます。

カスタマイズの手順は下記のようになります。

- (1) 掲示板種別を作成する。(「未承認」と「承認済み」)
- (2) ワークフローのプロセス定義を作成する。
- (3) 掲示板参照画面をカスタマイズし、申請機能を実装する。
- (4) 掲示板参照画面をカスタマイズし、承認時確認画面を実装する。
- (5) 最終承認後処理を実装する。

カスタマイズまたは作成するファイルは下記の通りとなります。

ファイルパス	ファイル名	説明
(ROOT)/pages/product/src/startpack/bulletin/ ↓ (コピー) (ROOT)/pages/src/startpack/bulletin/	ntc_dtl_cmnt.html ntc_dtl_cmnt.js	掲示板参照画面。 product から src にコピーして、カスタマイズする。通常の掲示板参照画面と、ワークフローの詳細画面、配信設定からのポップアップ表示を1つのプログラムで分岐処理している。
(ROOT)/pages/src/ startpack/bulletin/	postproc	承認後処理プログラム

(ROOT) は Resource Service ルートを示します。

(1) 掲示板種別の作成

次のように掲示板種別を作成します。

未承認(サンプル課22のみアクセス可能)

掲示板種別-更新

情報の一括削除 最新情報 ヘルプ

掲示板種別名称: 未承認

有効期限開始日:

有効期限終了日:

アクセス権: 設定

ソートキー: 0

更新 削除

アクセス権設定 - Windows Internet Explorer

アクセス権設定

公開方法: ☐ 全公開 ☒ 詳細設定を行う

個別設定

グループ

権限なし 権限変更 解除 下位組織にもアクセス権 グループ追加

グループ名	アクセス権
サンプル会社 / サンプル部門02 / サンプル課22	登録可

承認済み(全公開)

掲示板種別-更新

情報の一括削除 最新情報 ヘルプ

掲示板種別名称: 承認済み

有効期限開始日:

有効期限終了日:

アクセス権: 設定

ソートキー: 0

更新 削除

アクセス権設定 - Windows Internet Explorer

アクセス権設定

公開方法: ☒ 全公開 ☐ 詳細設定を行う

更新

「未承認」は「サンプル課22」のメンバーでのみアクセスが可能になっています。「承認済み」は全公開とします。

これから「未承認」の掲示板種別に登録したデータをワークフローに申請し、承認することで「承認済み」のフォルダに移動するように作りこんでいきます。(実際にはどのフォルダに登録したものでも申請でき、承認されるとすべて「承認済み」フォルダに移動するように作ります。)

(2) ワークフローのプロセス定義を作成します。

メニューから[ワークフロー]－[マスタメンテナンス]－[プロセス定義]を開きます。

ワークフローのプロセス定義を新規登録します。基本情報は下記のようになります。

プロセスマスタメンテナンス/プロセス定義編集

一覧に戻る

最新情報

更新
 新規バージョン
 削除
 プロセスの複製

掲示板申請

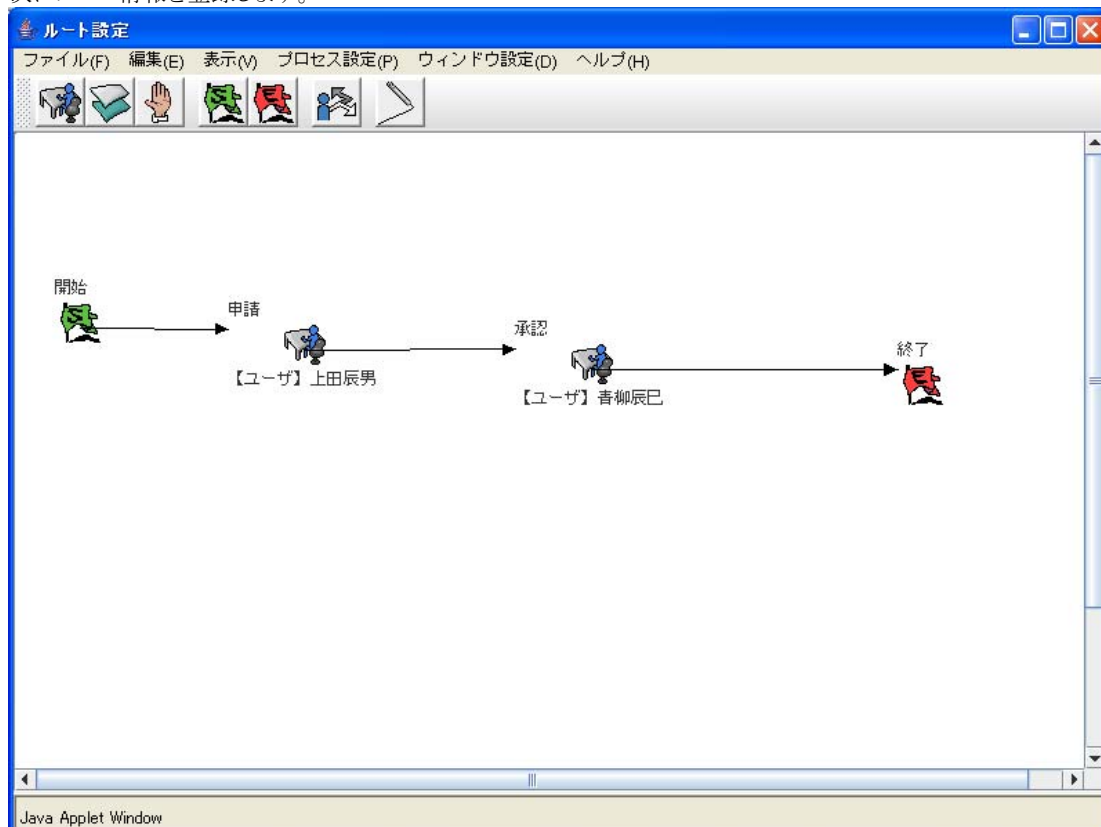
バージョン名	編集	フロー
1		

カテゴリ スクリプト開発モデル
 プロセス定義 掲示板申請
 バージョン 1

備考

プロセス数 0
 アクティブ 0
 作成日時 2007/10/29|15:51:34
 作成者 青柳辰巳
 最終更新日 2007/10/29|16:53:38
 最終更新者 青柳辰巳
開始日(必須) 2007/10/29
 終了日

次にフロー情報を登録します。



<申請>

対象者を指定します。(ここでは上田辰男とします。)

コンテンツパスはすべて「startpack/bulletin/ntc_dtl_cmnt」としてください。

<承認>

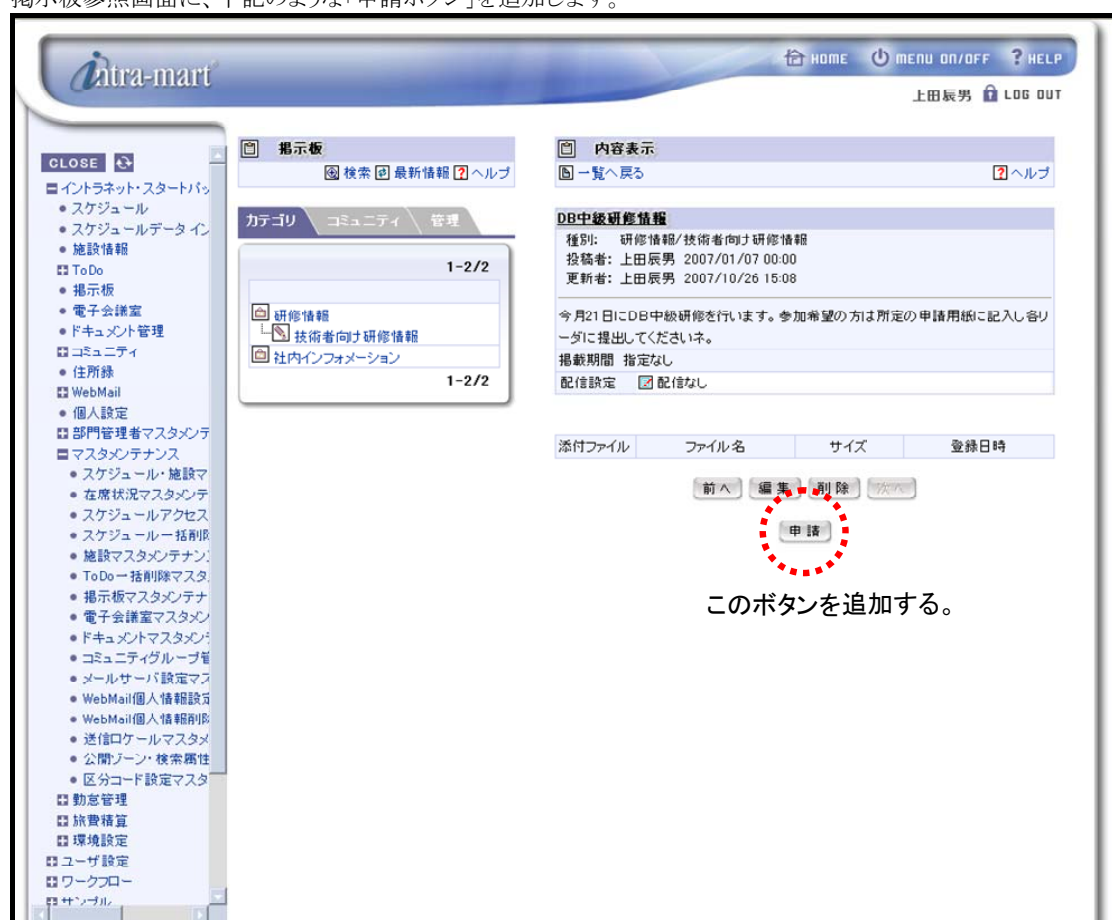
対象者を指定します。(ここでは青柳辰巳とします。)

コンテンツパスはすべて「startpack/bulletin/ntc_dtl_cmnt」としてください。

後処理「使用する」とし、処理プログラムに「startpack/bulletin/postproc」としてください。

- (3) 掲示板参照画面に「申請」ボタンを追加し、押下するとワークフローAPIを使って申請します。

掲示板参照画面に、下記のような「申請ボタン」を追加します。



このボタンを使用できる権限は、「編集ボタン」に対する権限と同様でよいと考えられますので、「編集ボタン」のソースコードを流用して作成します。

ntc_dtl_cmnt.html 523 行目付近、下記ソースコードの追加

```
<TABLE border="0" cellspacing="0" cellpadding="0" align="center">
  <TR>
    <TD class="button_padding"><IMG src="images/standard/button_left.gif"></TD>
    <TD class="button_bg">
      <INPUT type="button" class="button_bg" id="applyBtn"
        onClick="javascript:document.applyForm.submit();" value=" 申 請 "><IMART
type="condition" validity=oBulletinBind.editable negative> disabled="disabled"></IMART>>
    </TD>
    <TD class="button_padding"><IMG src="images/standard/button_right.gif"></TD>
  </TR>
</TABLE>
<!-- 申請処理実行 -->
<IMART type="form" name="applyForm" action="applyWF">
  <IMART type="hidden" bulletinCd = oBulletinBind.bulletinInfoData.bulletinCd></IMART>
  <IMART type="hidden" title = oBulletinBind.bulletinInfoData.title></IMART>
</IMART>
```

「編集ボタン」から変更しているのは、ボタンのキャプションと、onClick 時の動作です。それから、申請処理を呼び出すための Form を追加しています。「申請ボタン」を押下すると、applyForm が submit されます。ソース中に記載されている Condition タグは、タグに挟まれた範囲の実行を制御する IMART タグです。IMART タグについては、intra-mart の API リストを参照してください。

applyForm を submit すると下記のサーバメソッドを実行します。


```

ntc_dtl_cmnt.js 776 行目付近、下記ソースコードの追加

function applyWF(request)
{
    var sUserCd = Module.client.get("id");
    var sGroupCd = Module.client.get("group");

    var aApKey = new Array();
    aApKey[0] = request.bulletinCd;

    // BPW への起票処理
    var oWkfDraftObject = new WkfDrafter( sUserCd, sGroupCd );

    //起票メソッド

    var PROCESS_DEF_CD = "21";
    var VERSION_CD = "00000000000000000001";

    var oResult = oWkfDraftObject.draft(
        PROCESS_DEF_CD, /*プロセス定義 CD*/
        VERSION_CD,    /*バージョン CD*/
        "",             /*一時プロセス ID (任意)*/
        request.title,  /*プロセス名*/
        aApKey,         /*アプリケーションキー*/
        DatabaseManager.dateToString(new Date()), /* 起票指定日 */
        "",             /*会社コード*/
        "",             /*組織コード*/
        "",             /*起票コメント(任意)*/
        null,           /*根回しオブジェクト(任意)*/
        null,           /*次承認者オブジェクト(任意)*/
        "",             /*次アクティビティ CD(任意)*/
        "",             /*ロケール(任意)*/
    );

    Debug.browse(result);
}

```

上記のようにワークフローAPIを使用して申請処理を実行します。

PROCESS_DEF_CD,VERSION_CD は、ワークフローのマスタメンテナンス画面で知ることができます。
VERSION_CD については 20 桁で「0 バインド」して指定してください。

実際に運用システムを作成する場合には、起票時点の最新の VERSION_CD を取得する処理を記述する必要があります。

ワークフローAPI の詳細については「API リスト」および「ワークフロー仕様書」に詳しく記載されています。

- (4) 掲示板参照画面をカスタマイズし、承認時確認機能を実装します。

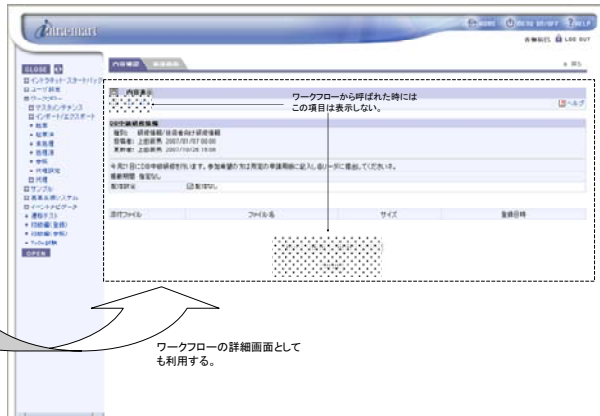
申請されたデータを承認者が確認する際にも掲示板情報を参照します。ワークフローモジュールから掲示板参照画面を呼び出すために、プロセス定義画面で下記のようにコンテンツパスを設定します。

ここではすべてのフェーズで掲示板参照画面のパスを入力しています。こうすると、掲示板参照画面を承認時の確認画面にも流用することができるようになります。しかし、承認時の確認画面に「申請」ボタンや他の掲示板情報へのリンクなどが表示されていると不都合です。そこで、下図のように不要なコントロールは表示しないようにします。

掲示板参照画面



ワークフロー承認時確認画面



同じプログラムを流用しますので、条件により表示と非表示を切り替える仕組みが必要です。ここでは<IMART type = "condition">タグを使用し、ワークフローモジュールから呼ばれた場合にはコントロールを表示しないようにします。

```
ntc_dtl_cmnt.html 317 行目付近
<IMART type="condition" validity=oBulletinBind.isWF negative>
  <IMART type="imToolBarLeft">
    <IMART type="imIcon"
      name="戻る"
      icon="images/standard/new.gif"
      href="javascript:onClickGoBackClick();"></IMART>
    </IMART>
  </IMART>
</IMART>
```

ワークフローモジュールから呼ばれたかどうかを判定するフラグ「oBulletinBind.isWF」は

```

ntc_dtl_cmnt.html.js 78行目付近

oBulletinBind.isWF = !isUndefined(request.bpw_parameter_cds);

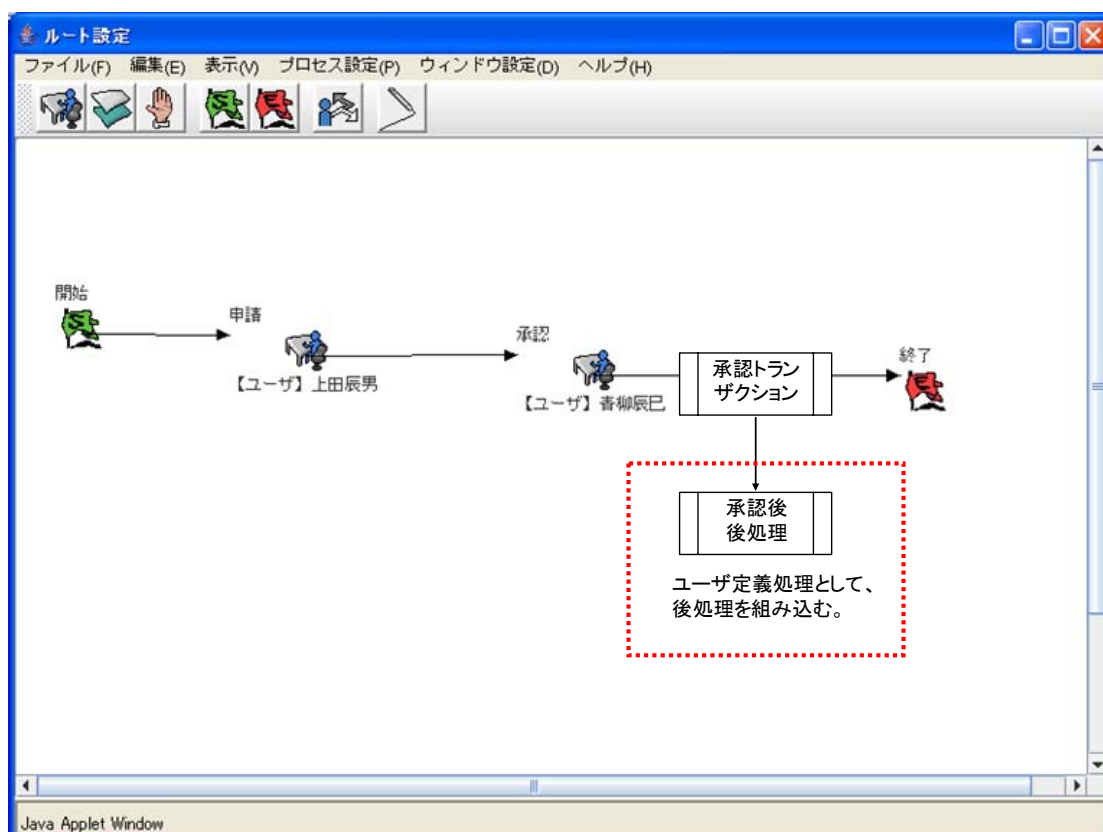
if (oBulletinBind.isWF)
{
    request.bulletinClsCd = request.bpw_parameter_cds.split(",")[0];
    request.bulletinCd = request.bpw_parameter_cds.split(",")[1];
}

```

のように初期化しています。ワークフローモジュールから画面が呼ばれる場合には、bpw_parameter_cds (ワークフローのアプリケーションキー) が渡されますので、この変数の有無で判断することができます。また、ワークフローモジュールから呼ばれた場合には、掲示板種別 CD と掲示板情報 CD が bpw_parameter_cds (ワークフローのアプリケーションキー) に入っているため、この値を取り出す処理を行っています。

(5) 承認後処理を実装します。

承認者が承認処理を実施したタイミングで、掲示板情報を「未承認」フォルダから「承認済み」フォルダに移動します。このような「承認したタイミング」に任意の処理を実施したい場合には、ワークフローモジュールの「後処理プログラム」として組み込みます。イメージとしては下記のようになります。



青柳辰巳さんが承認処理を実行すると、ワークフローモジュールが「承認トランザクション」を実行します。そのトランザクションの中で、ユーザが独自に定義した承認後の後処理を実行します。後処理を組み込む設定は下図のようにします。

基本 対象者 コンテンツパス 自動パス 自動催促

キャプション 承認

終了可能

☐ 使用する ☒ 使用しない

直前の承認者へ差戻し・引戻し

☒ 使用する ☐ 使用しない

後処理

☒ 使用する ☐ 使用しない

処理プログラム /startpack/bulletin/postproc

コンテンツフレーム種別

☒ 標準 ☐ カスタム

☐ タスク追加時のコンテンツフレーム種別の初期値に設定します。

OK(O) Cancel(C)

Java Applet Window

承認アクティビティの後処理を「使用する」にし、処理プログラムの欄には組み込みたい後処理プログラムのパスを指定します。そして、後処理プログラムの中に「`proceed()`」というメソッドを実装し、そこに「承認時の後処理」を記載します。「`proceed()`」には「承認時の後処理を記載する」と決まっています、他にも「否認時の後処理」を記載したければ「`rejection()`」、「起票者への差し戻し」なら「`returnDraft()`」というメソッドに記載することになっています。各処理毎に呼び出されるメソッド名の一覧については「ワークフロー仕様書」を参照してください。

それでは、`proceed.js` の内容を見ていきます。ワークフローの後処理プログラムは、引数と戻り値のI/Fが決まっている点が特徴的です。

```

proceed.js

function proceed(sProcessDefCd,sVersionCd,sProcessCd,sActivityCd,sLoginGrpCd,sUserCd)
{
    // 戻り値の設定
    var STATUS_COMPLETED =
    Packages.jp.co.intra_mart.foundation.bpw.model.data.EISModelStaticIF.STATUS_COMPLETED + "";
    var STATUS_FAULT =
    Packages.jp.co.intra_mart.foundation.bpw.model.data.EISModelStaticIF.STATUS_FAULT + "";

    ///////////////////////////////////////////////////
    // 申請対象のプロセス情報を取得する。

    var oPro = new WkfProcess (sProcessDefCd ,sVersionCd ,sProcessCd ,sLoginGrpCd);
    var oPInf = oPro.getProcessInfo();

    Debug.print(oPInf.app_key[0]); /* 掲示板種別 CD */
    Debug.print(oPInf.app_key[1]); /* 掲示板情報 CD */

    ～ (中略) ～

    // 掲示板種別を更新する。
    oApp.data[0].bulletinClsCd = sAprClsCd;
    var oResult = oBulletinMng.updateBulletin(oApp.data[0]);

    if (oResult.err)
    {
        Debug.print("[ERR] 掲示板情報の更新でエラーが発生");
        return STATUS_FAULT;
    }

    // 正常終了で戻り値を返す
    Debug.print("[SUCCESS] 掲示板情報の申請後処理が正常終了");

    return STATUS_COMPLETED;
}

```

引数は、processDefCd、versionCd、processCd、activityCd、groupId、userId(プロセス定義 CD、バージョン CD、プロセス CD、アクティビティ CD、ログイングループ ID、ユーザ ID)となっています。今回の後処理プログラムではワークフローのアプリケーションキーから掲示板種別 CD や掲示板情報 CD を取得する必要がありますので、WkfProcess.getProcessInfo() で詳細なワークフローのプロセス情報を取得し、そこからアプリケーションキーを取り出しています。

掲示板種別を変更する処理については説明を割愛します。ここでも同様に掲示板の API を利用して処理を行っています。

最後に、成功か失敗かを返却します。返却値はあらかじめ規定されているものを返却しています。

♪ポイント♪

後処理プログラムの中ではトランザクション制御を行うことができません。呼び出し元の承認トランザクションに依存します。

4 資料

4.1 初期化データ一覧

勤怠管理(work)

項目名称	説明
WORK_BATCH_DELETE_LIMIT	バッチ処理により、過去データを削除する場合のnヶ月指定。(指定=3の場合、3ヶ月以上過去のデータを削除します。)
WORK_CSV_MIMETYPE	CSV出力時のMIMEタイプを指定します。 Excel(application/vnd.ms-excel)
WORK_LIST_PER_DAY	1-N(N:整数)を指定。1日の勤務入力数を決定する。
ORK_ORGNSELECT_COMP_ATTRIBUTE	勤怠管理会社検索の検索属性
WORK_ORGNSELECT_COMP_ZONE	勤怠管理会社検索の公開ゾーン
WORK_ORGNSELECT_OFFICIAL_ZONE	勤怠管理パブリックグループ検索の公開ゾーン
WORK_REST_TIME	休憩時間の指定(複数可、 区切りで指定)。時間を重複しての指定はできません。例 12.50-13.50 20.00-20.50(12:30-13:30と20:00-20:30の指定)
WORK_USERSELECT_COMP_ATTRIBUTE	勤怠管理ユーザ検索の検索属性
WORK_USERSELECT_COMP_ZONE	勤怠管理ユーザ検索の公開ゾーン
WORK_USERSELECT_OFFICIAL_ZONE	勤怠管理ユーザ検索のパブリックグループ公開ゾーン
WORK_WKFBATCH_DELETE_LIMIT	バッチ処理により、過去データを削除する場合のn日指定。(指定=3の場合、3日以上過去のデータを削除します。)

旅費清算(trvlexp)

項目名称	説明
TRVLEXP_BATCH_DELETE_LIMIT	バッチ処理により、過去データを削除する場合のnヶ月指定。(指定=3の場合、3ヶ月以上過去のデータを削除します。)
TRVLEXP_COURSE_REFERENCE_FLAG	経路検索サイトを呼び出すリンクの表示、非表示フラグ。
TRVLEXP_COURSE_REFERENCE_URL	経路検索サイトのURL。
TRVLEXP_CSV_MIMETYPE	CSV出力時のMIMEタイプを指定します。 Excel(application/vnd.ms-excel)
TRVLEXP_EXP_FLAG	駅すばあとの連携フラグ
TRVLEXP_EXP_URL	駅すばあとのURLを指定
TRVLEXP_ORGNSELECT_COMP_ATTRIBUTE	旅費精算会社検索の検索属性
TRVLEXP_ORGNSELECT_COMP_ZONE	旅費精算会社検索の公開ゾーン

TRVLEXP_ORGNSELECT_OFFICIAL_ZONE	旅費精算パブリックグループ検索の公開ゾーン
TRVLEXP_ROUTE_NUMBER_OF_LINE	交通費の最大指定数。
TRVLEXP_USERSELECT_COMP_ATTRIBUT E	旅費精算ユーザ検索の検索属性
TRVLEXP_USERSELECT_COMP_ZONE	旅費精算ユーザ検索の公開ゾーン
TRVLEXP_USERSELECT_OFFICIAL_ZONE	旅費精算ユーザ検索のパブリックグループ公開ゾーン
TRVLEXP_WKFBATCH_DELETE_LIMIT	バッチ処理により、過去データを削除する場合のn日指定。(指定＝3の場合、3日以上過去のデータを削除します。)

4.2 区分コード一覧

アプリケーションの動作を設定する、区分コードの一覧は下記の通りです。

(メニュー[イントラネット・スタートパック]-[マスタメンテナンス]-[区分コード設定マスタ]で設定できます。)

掲示板(notice)

区分名称	区分コード	説明
NTC_INITIAL	NOTICE_CB_MAX_RECORD	CBレコード検索(IM-統合検索オプション使用)の検索数上限値指定(掲示板添付ファイル内容検索機能を利用の場合)
NTC_INITIAL	NOTICE_SEND_MAIL_ADDRESS	エラーメール送信者アドレス(掲示板メール投稿機能を利用の場合)投稿エラーの場合に配信するメールの送信元メールアドレスを指定する。
NTC_INITIAL	NOTICE_USE_CBSEARCH	ファイル内容検索(IM-統合検索オプションが必要)の使用不使用設定使用(ON)、不使用(OFF)を指定する。
NOTICE_BATCH_DELETE_LIMIT	NOTICE_BATCH_DELETE_LIMIT_01	バッチ処理により、過去データを削除する場合のnヶ月指定。(指定＝3の場合、3ヶ月以上過去のデータを削除します。)

旅費精算(trvlexp)

区分名称	区分コード	説明
TRVEXP_CLASS	TRVL_CLASS01～	旅行区分の選択肢
TRVLEXP_DAILY_ALLOWANCE	TRVL_ALLOWANCE01～	日当の選択肢
TRVLEXP_TRANSPORTATION	TRVL_TRANSPORTATION01～	交通手段の選択肢

ドキュメント管理 (document)

区分名称	区分コード	説明
DOC_ETC_WORD	01～	ドキュメントデータの予備項目 使用:ON、未使用:OFF キャプション
DOC_INITIAL	DOCUMENT_CB_MAX_RECORD	CBレコード検索 (IM-統合検索オプション 使用) の検索数上限値指定 (ドキュメント ファイル内容検索機能を利用の場合)
DOC_INITIAL	DOCUMENT_USE_CBSEARCH	ファイル内容検索 (IM-統合検索オプション が必要) の使用不使用設定使用 (ON)、 不使用 (OFF) を指定する。

スケジュール管理 (schdl)

区分名称	区分コード	説明
SCHEDULE_BATCH_DELETE_LI MIT	SCHEDULE_BATCH_DELETE_LIMIT_0 1	バッチ処理により、過去データを削除す る場合のnヶ月指定。(指定=3の場合、 3ヶ月以上過去のデータを削除します。)

ToDo 管理 (todo)

区分名称	区分コード	説明
TODO_RECEPTION_STATUS	01～	対応状況選択肢
TODO_CONTENTS	01～	ToDo内容選択肢

住所録 (address)

区分名称	区分コード	説明
ADDRESS_ITEM	01～	※この値は変更しないでください。

4.3 個人設定項目一覧

イントラネット・スタートパックのユーザ単位に設定可能な項目です。

メニューから[イントラネット・スタートパック]－[個人設定]を選択してください。

＜スケジュール・施設予約＞

項目名称	説明
共通設定	
表示単位(日)	個人スケジュール画面の、表示単位を設定します。
表示時間帯	スケジュール画面の表示時間帯を設定します。
先頭表示(週)	グループスケジュール画面に表示する週の先頭を設定します。
アクセス権	各アクセス権を設定します。
ユーザスケジュール設定	
表示方法	ユーザスケジュールの表示方法を設定します。
選択グループの追加	スケジュールの選択できるグループを追加します
施設スケジュール設定	
表示方法	施設スケジュールの表示方法を設定します。
デフォルトグループ	施設スケジュール表示時のグループのデフォルト選択を設定します。

メール即時配信設定	
登録時チェック	配信設定後、登録時にメールを送信します。
更新時チェック	配信設定後、更新時にメールを送信します。
営業日報設定	
営業日報作成チェック	スケジュール登録と同時に営業日報の下書きを作成します。
メール定期配信設定	
配信頻度	メール定期配信の配信頻度を設定します。
配信先	メール定期配信の配信先を設定します。
有効期間開始日	メール定期配信の期間開始日を設定します。
有効期間終了日	メール定期配信の期間終了日を設定します。

< 電子会議室 >

個人設定(電子会議室)

[スケジュール](#)
[ToDo](#)
[住所録](#)
[WebMail](#)
[ポータル](#)
[ブログ](#)

[最新情報](#)
[ヘルプ](#)

共通設定

表示方法

☐ ツリー表示
☒ 未読一覧

更新

項目名称	説明
共通設定	
表示方法	電子会議室のデフォルト表示方法を設定します。

< 住所録 >

住所録個人設定

個人設定(住所録)

[スケジュール](#)
[ToDo](#)
[電子会議室](#)
[WebMail](#)
[ポータル](#)
[ブログ](#)

[最新情報](#)
[ヘルプ](#)

個人住所録表示

個人住所録グループ

共有住所録表示

住所録一覧に表示する項目、及びその順番を指定します。
表示する項目リストの上から順に住所録の項目表示されます。

表示しない項目

☐ 個人住所録ユーザコード
☐ FAX番号
☐ 郵便番号
☐ 住所
☐ メールアドレス2

☒ 全選択
☐ 選択
☐ 解除
☒ 全解除

表示する項目

氏名

メールアドレス1

電話番号

会社名

組織名

☒ 先頭へ
☐ 上へ
☐ 下へ
☒ 最後へ

更新

項目名称	説明
表示しない項目	個人住所録一覧に表示しない属性の一覧です。
表示する項目	個人住所録一覧に表示する属性の一覧です。

住所録個人グループ設定

項目名称	説明
名前	個人住所録グループの一覧です。

住所録共有設定

項目名称	説明
表示しない項目	共有住所録一覧に表示しない項目の一覧です。
表示する項目	共有住所録一覧に表示する項目の一覧です。

< WebMail >

個人情報設定

個人設定(WebMail)
 To Do
 電子会議室
 住所録
 WebMail
 ポータル
 ブログ
 最新情報
 ヘルプ

アカウント設定

アカウント *
 氏名 *
 メールアドレス *
 送信サーバ *
 ポート番号 *
 受信サーバ *
 ポート番号 *
 サーバの種類 * ☒ POP3サーバ ☐ IMAP4サーバ
 アカウント名 *
 パスワード *

注)ポート番号はシステム管理者から特に指定されていない場合は、変更しないでください

登録

引用符 >
 署名
 受信メール表示設定 10 件毎にメールを表示する
 開封確認

開封確認を行うかどうか、その都度問い合わせる(推奨)
開封確認を無視する
自動で開封確認を送る

 送信オプション ☒ 自分自身にも送信する
 ユーザ認証 ☐ SMTP Auth を使用する ☐ POP before SMTP を使用する [詳細設定](#)

項目名称	説明
アカウント	アカウントを識別する任意の名前です。
氏名	氏名です。
メールアドレス	メールアドレスです。
送信サーバ	メール送信サーバのアドレスです。
送信サーバ - ポート番号	メール送信サーバのポート番号です。
受信サーバ	メール受信サーバのアドレスです。
受信サーバ - ポート番号	メール受信サーバのポート番号です。
サーバの種類	サーバの種類を設定します。
アカウント名	メールサーバへのアカウントです。
パスワード	アカウントと組になるパスワードです。
引用符	引用符です。
署名	署名です。
受信メール表示設定	受信メールのページあたりの表示件数を設定します。
開封確認	開封確認の設定です。
送信オプション	自分への送信の有無を設定します。
ユーザ認証	ユーザ認証の種類を設定します。

個人設定

アカウント設定 初期設定 振り分け設定

メール受信設定

表示列名   重要度
添付
件名
差出人
送信日時
サイズ

プレビューウィンドウ ☒ メッセージの下(W) ☐ メッセージの横(S)

ハイライトの色の変更 

クイック検索設定 ☒ 表示する ☐ 表示しない

メール送信設定

メールを送信形式 ☒ HTML ☐ TXT

既定のエンコーディング 日本語(ISO-2022-JP) ▼

新しいアドレス保存 ☒ 保存する ☐ 保存しない

CCとBCC詳細 ☒ 表示する ☐ 表示しない

詳細オプション ☒ 表示する ☐ 表示しない

登録

項目名称	説明
メール受信設定	
表示列名	メール一覧の表示内容を設定します。
プレビューウィンドウ	メール詳細のフレームの位置を設定します。
ハイライトの色の変更	メール選択時のハイライトの色を変更します。
クイック検索設定	クイック検索の表示/非表示を設定します。
メール送信設定	
メールの送信形式	メールの送信形式を設定します。
既存のエンコーディング	メールのエンコーディングを設定します。
新しいアドレス保存	過去に送受信したことのないアドレスを保存するかどうかを設定します。
CCとBCC詳細	CCとBCCの表示/非表示を設定します。
詳細オプション	メール詳細オプションの表示/非表示を設定します。

振り分け設定

アカウント設定 初期設定 振り分け設定

ルール名	フィルター	ソートキー	編集
<input type="checkbox"/> 日報振り分け	Fromに「営業日報」を含む場合、非表示にする	1	

項目名称	説明
ルール名	振り分けルールの名称です。
フィルター	振り分けの内容です。
ソートキー	振り分けルールの表示順です。

個人設定(ToDo) | スケジュール | 電子会議室 | 住所録 | WebMail | ポータル | ブログ | 最新情報 | ヘルプ

新規登録		1-1/1
カテゴリ	説明	
開発グループ		

1-1/1

項目名称	説明
カテゴリ	ToDoカテゴリの名称です。
説明	説明です。

個人設定(ポータル)	
スケジュール ToDo 電子会議室 住所録 WebMail ブログ 最新情報 ヘルプ	
お気に入り配信設定<選択したお気に入りの情報が新着順にポータル画面に表示されます>	
配信対象	<div> 掲示板(カテゴリ) <div> 1-2/2 <div> 選択 解除 </div> </div> </div> <div> 掲示板(カテゴリ)検索 <input type="text"/> </div> <div> <input checked="" type="checkbox"/> 研修情報 <input checked="" type="checkbox"/> 社内インフォメーション </div> <div> 1-2/2 </div>
最大表示行数 *	<input type="text" value="1"/> ※最大行数は20行
その他設定	
新着営業日報一覧表示行数 *	<input type="text" value="1"/> ※最大行数は20行
今日の予定表示行数 *	<input type="text" value="1"/> ※最大行数は20行
ToDo通知表示行数 *	<input type="text" value="1"/> ※最大行数は20行
新着上司コメント表示行数 *	<input type="text" value="1"/> ※最大行数は20行
新着配信情報表示行数 *	<input type="text" value="1"/> ※最大行数は20行
新着コミュニティ情報表示行数 *	<input type="text" value="1"/> ※最大行数は20行
ToDo一覧表示行数 *	<input type="text" value="1"/> ※最大行数は20行
伝言メモ一覧表示行数 *	<input type="text" value="1"/> ※最大行数は20行
注目カテゴリ最新記事表示行数 *	<input type="text" value="1"/> ※最大行数は20行
フレンド最新記事表示行数 *	<input type="text" value="1"/> ※最大行数は20行
新着メッセージ表示行数 *	<input type="text" value="1"/> ※最大行数は20行
更新	

項目名称	説明
お気に入り配信設定	
配信対象	配信対象を設定します。
最大表示行数	設定した配信情報の最大表示行数を設定します。
その他設定	
新着営業日報一覧表示行数	新着営業日報一覧ポートレットの最大表示行数を設定します。
今日の予定行数	今日の予定ポートレットの最大表示行数を設定します。
ToDo通知表示行数	ToDo通知ポートレットの最大表示行数を設定します。
新着上司コメント表示行数	新着上司コメントポートレットの最大表示行数を設定します。
新着配信情報表示行数	新着配信情報ポートレットの最大表示行数を設定します。

新着コミュニティ情報表示行数	新着コミュニティ情報ポートレットの最大表示行数を設定します。
ToDo一覧表示行数	ToDo一覧ポートレットの最大表示行数を設定します。
伝言メモ一覧表示行数	伝言メモ一覧ポートレットの最大表示行数を設定します。
注目カテゴリ最新記事表示行数	注目カテゴリ最新記事ポートレットの最大表示行数を設定します。
フレンド最新記事表示行数	フレンド最新記事ポートレットの最大表示行数を設定します。
新着メッセージ表示行数	新着メッセージポートレットの最大表示行数を設定します。

<ブログ>

スケジュール
TODO
電子会議室
住所録
WebMail
ポータル
最新情報

注目カテゴリ設定

注目カテゴリ

ファッション

メンズ

レディース

アクセサリ

選択

解除

注目カテゴリ

先頭へ

上へ

下へ

最後へ

メール通知設定

メール通知設定(随時)

☐ 注目カテゴリに記事が投稿されたとき
☐ マイフレンドが記事を投稿したとき
☐ 自分の記事に対してコメントが投稿されたとき
☐ 自分宛のメッセージが届いたとき

メール通知設定(定期)

☒ 通知なし
☐ 毎日:1日分のサマリを受け取る

通知先

☐ 通常アドレス (未設定)
☐ モバイルアドレス (未設定)
☐ ユーザ指定のメールアドレス

有効期間開始日

2007/12/14 日付選択

有効期間終了日

2008/01/14 日付選択

記事登録初期状態設定

デフォルトカテゴリ/分類

更新

項目名称	説明
注目カテゴリ設定	
注目カテゴリ	注目カテゴリを設定します。
メール通知設定	
メール通知設定(随時)	メールの通知タイミングを設定します。
メール通知設定(定期)	定期メールの通知タイミングを設定します。
通知先	通知先を設定します。
有効期間開始日	有効期間開始日を設定します。
有効期間終了日	有効期間終了日を設定します。
記事登録初期状態設定	
デフォルトカテゴリ/分類	記事登録時のデフォルトのカテゴリと分類を設定します。

4.4 初期化ファイル項目一覧

ISP では以下の項目を初期化ファイル(startpack.ini,webmail.ini)で設定することが可能です。運用開始前に設定を行ってください。初期化ファイルを変更した場合には、intra-mart システムの再起動が必要です。

4.4.1 startpack.ini

ファイルの場所:Resource Service インストールフォルダ/pages/product/src/startpack/startpack.ini

項目名称	初期値	説明
スタートパック共通		
CSV_EXPLAIN_MARK	空欄	CSV出力時の説明データの改行コード変換文字列 空欄または任意の文字列を指定します。
CATEGORY_NUM_MAX	30	カテゴリ数の上限値
TREE_CATEGORY_NUM_MAX	30	組織検索カテゴリ数の上限値
CSV_SEPARATOR	,(コロン)	CSV出力時の区切り文字
SEARCH_TYPE	1	検索方法の環境設定 ・1:前方一致(デフォルト) ・2:部分一致
STARTPACK_FILE_UPLOAD_SAVE_TYPE	id	ファイルのサーバへの保存タイプ 仮想ファイル名(ID)で保存する場合:id ファイルサーバに実ファイル名で保存する場合:空欄
MAIL_URL_USE	OFF	メールにURLを表示 (ON=URLを表示する,OFF=URLを表示しない)
PAGING_BEFORE_NUMBER_OF_LINE	5	1画面中のページング最大表示数(表示ページより前方)
PAGING_AFTER_NUMBER_OF_LINE	5	1画面中のページング最大表示数(表示ページより後方)
SAVE_SAME_NAME_FILE	OFF	ファイル添付機能で同名ファイルで保存可能/不可能の設定 (ON=可能,OFF=不可能)
FILE_SORT_TYPE	DESC	ファイル添付機能で降順/昇順の設定(ASC=昇順,DESC=降順)
EDIT_PROFILE_FLG	OFF	ユーザプロファイル情報の変更をユーザ自身が変更可能/変更不可能かの設定(ON=可能,OFF=不可能)
スケジュール管理		
SCHEDULE_BATCH_DELETE_LIMIT	3	過去データ削除対象設定(指定=3の場合、3ヶ月以上過去のデータを削除)
SCHEDULE_DISP_TODO	15	個人スケジュール画面のToDo表示件数
SCHEDULE_DISP_SEAT	10	個人スケジュール画面の伝言メモ表示件数
SCHEDULE_SENDMAIL	ON	スケジュール機能でメール送信機能を使用する/しないの指定 ON=メールを送信する,OFF=送信しない
SCHEDULE_USER_MAIL	ON	スケジュール毎メール通知設定 ON=有効
SCHEDULE_APPLICATION_LOCK_TIME_OUT	10	スケジュール/施設 新規登録で登録更新処理時に行うアプリケーションロックのセッションタイムアウト秒数
ToDo管理		
TODO_BATCH_DELETE_LIMIT	3	過去データ削除対象設定(指定=3の場合、3ヶ月以上過去のデータを削除)
TODO_NUMBER_OF_LINE	10	ToDoリストの1ページ表示件数
SND_NUMBER_OF_LINE	10	依頼リストの1ページ表示件数

MSG_NUMBER_OF_LINE	10	ToDo通知リストの1ページ表示件数
掲示板		
NOTICE_BATCH_DELETE_LIMIT	3	過去データ削除対象設定(指定=3の場合、3ヶ月以上過去のデータを削除)
NOTICE_NUMBER_OF_LINE2	10	掲示板情報検索一覧の1ページ表示件数
NOTICE_INSERT_MAIL_HOST	空欄	投稿用メールサーバホストの指定(掲示板メール投稿機能を利用の場合)
NOTICE_INSERT_MAIL_PORT	空欄	投稿用メールサーバポートの指定(掲示板メール投稿機能を利用の場合)
NOTICE_INSERT_MAIL_PROTOCOL	空欄	投稿用メールプロトコルの指定(掲示板メール投稿機能を利用の場合) POP3またはIMAPを指定する。
NOTICE_USE_BY_MAIL	OFF	メールによる掲示板投稿機能の使用不使用設定 ON=使用() OFF=不使用
NOTICE_SEND_MAIL_ADDRESS	空欄	メールによる掲示板投稿時、エラーメールを送信するアドレス
NOTICE_CONTENT_OF_LINE	250	掲示板情報内容の表示行数
電子会議室※		
CONFERENCE_BATCH_DELETE_LIMIT	3	過去データ削除対象設定(指定=3の場合、3ヶ月以上過去のデータを削除)
MEETING_NUMBER_OF_LINE	20	電子会議室ツリー表示、1ページ表示スレッド件数
MEETING_NUMBER_OF_LINE2	20	電子会議室新着順表示、1ページ表示件数
MEETING_CONTENT_OF_LINE	3	電子会議室情報内容の表示行数
住所録		
ADDRESS_NUMBER_OF_LINE	10	一覧の1ページ表示件数
配信設定		
DELIVERIED_MEMBER_MAX	30	未読表示ユーザ上限数
コミュニティ※		
COMMUNITY_NUMBER_OF_LINE	10	一覧の1ページ表示件数
COMMUNITY_NUMBER_OF_HORIZONTAL	4	マイコミュニティ横方向表示件数
COMMUNITY_MAX_PICTURE_SIZE	500000	コミュニティ画像の最大サイズ(500K)
COMMUNITY_PICTURE_EXTENSION	gif jpg jpeg png	コミュニティ画像許容拡張子
COMMUNITY_NO_IMAGE	startpack/froot/noimage/community.gif	コミュニティ画像指定なし時の表示画像
マスタメンテナンス		
MASTER_SDLDEL_NUMBER_OF_LINE	10	スケジュール一括削除マスタメンテナンス一覧の1ページ表示件数
MASTER_TODODEL_NUMBER_OF_LINE	10	ToDo一括削除マスタメンテナンス一覧の1ページ表示件数
MASTER_SFA_NUMBER_OF_LINE	10	顧客名(SFA)マスタメンテナンス一覧の1ページ表示件数
営業日報		
BSN_SDL_NUMBER_OF_LINE	10	営業日報一覧の1ページ表示件数

BSN_RCV_NUMBER_OF_LINE	10	受信日報一覧の1ページ表示件数
BSN_MNT_NUMBER_OF_LINE	10	管理日報一覧の1ページ表示件数
BSN_NEW_NUMBER_OF_LINE	10	営業日報下書き一覧の1ページ表示件数
ドキュメント管理※		
DOCUMENT_SEARCH_OF_LINE	20	ドキュメント管理 検索結果一覧の1ページ表示件数
ポートレット		
PORTLET_FAVORITE_DELIVERY_LINE_DEFAULT	10	お気に入り配信 最大表示件数
PORTLET_TODAY_SCHEDULE_LINE_DEFAULT	10	今日の予定 最大表示件数
PORTLET_TODO_LINE_DEFAULT	10	ToDo通知 最大表示件数
PORTLET_BUSINESS_COMMENT_LINE_DEFAULT	10	新着上司コメント 最大表示件数
PORTLET_BUSINESS_DELIVERY_LINE_DEFAULT	10	新着営業日報一覧 最大表示件数
PORTLET_NEW_DELIVERY_LINE_DEFAULT	10	新着配信情報 最大表示件数
PORTLET_NEW_COMMUNITY_LINE_DEFAULT※	10	新着コミュニティ情報 最大表示件数
PORTLET_TODO_LIST_LINE_DEFAULT	10	ポートレットのToDo一覧 最大表示件数
PORTLET_MEMO_LINE_DEFAULT	10	ポートレットの伝言メモ一覧 最大表示件数
PORTLET_CATEGORY_ATCL_LINE_DEFAULT	10	ポートレットのカテゴリ最新記事 最大表示件数
PORTLET_FRIEND_ATCL_LINE_DEFAULT	10	ポートレットのフレンド最新記事 最大表示件数
PORTLET_NEW_MESSAGE_LINE_DEFAULT	10	ポートレットの新着メッセージ 最大表示件数
ブログ		
BLOG_BUILDING_FRIENDLY_RELATIONS_AUTOMATICALLY	OFF	プロフィール登録時の自動フレンド登録設定 ON :プロフィール登録を行った時点で全てのユーザがマイフレンドに追加される OFF:フレンド登録には個人間の申請・承認/否認行為が必要
BLOG_BATCH_DELETE_ACCESS_EACH_DAY_LIMIT	10	ブログ記事日別アクセス件数削除対象設定(単位:日) (例 指定=10、バッチ起動日=9/11の場合:9/1より前のデータを削除) バッチ「ブログ記事日別アクセス件数削除」設定を有効にする必要があります。 ※「アクセスランキング(週間ランキング)」画面では、7日間前までの日別アクセス数を使用しているため、 当値は7以上を設定してください。
BLOG_BATCH_DELETE_ARTICLE_LIMIT	12	ブログ記事削除対象設定(単位:月) (例 指定=12、バッチ起動日=2007/9/11の場合:記事更新日が2006/9/11より前の記事データ(付属情報含む)を削除) バッチ「ブログ記事削除」設定を有効にする必要があります。
BLOG_BATCH_DELETE_MESSAGE_LIMIT	3	ブログメッセージ削除対象設定(単位:月)

MIT		(例 指定=3、バッチ起動日=9/11の場合:6/11より前に送信されたメッセージを削除) バッチ「ブログメッセージ削除」設定を有効にする必要があります。
BLOG_SYSTEM_MAIL_SENDER	intra-mart@system	ブログ機能からユーザに対してメール通知(随時/定期)が行われる際のメールアドレス(FROM)
BLOG_PICTURE_MAX_SIZE	500	全般 写真・画像 登録可能最大サイズ(単位:KB)
BLOG_PROFILE_DEFAULT_PICTURE	startpack/f sroot/noimage/blog_profile.gif	ブログプロフィール画像指定なしの場合のデフォルト表示画像
BLOG_PROFILE_WORD_BREAK	OFF	ブログプロフィール得意分野/自己紹介の英単語強制改行 ON: 英単語のみの、枠をはみ出すような長い単語を強制的に改行 OFF: ブラウザの動作(長い英単語は改行しない)のまま動作
BLOG_TIME_UNTIL_ACCESS_HISTORY_UPDATE	5	ブログ記事のアクセス後、次にアクセス履歴が更新されるまでの必要時間(単位:分)
BLOG_MAX_NUMBER_OF_LATEST_ARTICLES	3	ブログホーム画面 最新記事 最大表示件数
BLOG_MAX_NUMBER_OF_LATEST_CATEGORIZED_ARTICLES	2	ブログホーム画面 注目カテゴリ記事 最大表示件数
BLOG_MAX_NUMBER_OF_LATEST_FRIEND_ARTICLES	3	ブログホーム画面 フレンド最新記事 最大表示件数
BLOG_MAX_LENGTH_OF_ARTICLE_TITLE	20	ブログホーム画面 記事タイトル 最大表示文字数
BLOG_MAX_LENGTH_OF_SELF_INTRODUCTION	100	一覧画面共通 自己紹介 最大表示文字数
BLOG_MAX_NUMBER_OF_MESSAGES	20	メッセージ一覧画面 メッセージ 最大表示件数
BLOG_MAX_NUMBER_OF_RANKING	30	アクセスランキング画面 記事 最大表示件数
BLOG_MAX_NUMBER_OF_ACCESS_HISTORY	30	記事アクセス履歴画面 履歴 最大表示件数
BLOG_KEEPING_DAYS_OF_ACCESS_HISTORY	10	記事アクセス履歴 履歴保持日数(バッチ処理で使用)
BLOG_PERIOD_OF_ARTICLE_ARCHIVES	12	記事一覧画面 過去の記事 表示期間(システム月を含む指定ヶ月分を表示)
BLOG_MAX_NUMBER_OF_LATEST_COMMENTS	10	記事一覧画面 最新コメント 最大表示件数
BLOG_MAX_NUMBER_OF_ARTICLES	20	記事一覧画面 記事 最大表示件数
BLOG_MAX_LENGTH_OF_ARTICLE_CONTENTS	100	記事一覧画面 記事内容 最大表示文字数
BLOG_MAX_LENGTH_OF_LATEST_COMMENTS	10	記事一覧画面 最新コメント 最大表示文字数
BLOG_MAX_NUMBER_OF BLOG MEMBERS	30	ブログメンバー一覧表示 メンバー 最大表示数
BLOG_MAX_NUMBER_OF_SEARCHES	30	記事検索結果一覧画面 記事 最大表示件数

D_ARTICLES		
BLOG_MAX_NUMBER_OF_MANAGED_ARTICLES	30	記事管理画面 記事 最大表示件数
BLOG_MAX_NUMBER_OF_MASTER_CATEGORY	10	ブログカテゴリマスタメンテナンス画面 カテゴリ 最大表示件数
モバイル		
MOBILE_DISPLAY_MAX_WIDTH	18	モバイル画面 表示文字数(半角)
MOBILE_SCHEDULE_MAX_NUMBER_OF_RESERVES	15	スケジュール/施設 一覧画面 最大表示件数
MOBILE_SCHEDULE_MAX_NUMBER_OF_GROUP_USERS	30	スケジュール グループの予定画面 最大表示ユーザ数(ログインユーザを除く)
MOBILE_SCHEDULE_MAX_NUMBER_OF_FACILITIES	20	施設情報 施設情報画面 最大表示施設数
MOBILE_TODO_MAX_NUMBER_OF_RESERVES	15	ToDo 一覧画面 最大表示件数
MOBILE_TODO_MAX_NUMBER_OF_MEMOS	15	伝言メモ 一覧画面 最大表示件数
MOBILE_ADDRESS_MAX_NUMBER_OF_PERSONAL_ADDRESSES	30	個人住所録 検索結果画面 最大表示件数
MOBILE_WEBMAIL_MAX_NUMBER_OF_RECEIVE_MAILS	30	WebMail メール一覧画面 最大表示件数
MOBILE_WEBMAIL_MAX_NUMBER_OF_DESTINATIONS	30	WebMail 送信先一覧画面 最大表示件数

4.4.2 webmail.ini

ファイルの場所: Resource Service インストールフォルダ/pages/product/src/startpack/webmail/webmail.ini

項目名称	初期値	説明
WML_ATTACH_COUNT	3	添付ファイルの個数制限
WML_ATTACH_MAX	3145728	添付ファイルの容量制限(byte) 半角数字を指定してください。
WBM_ADV_SEARCH_NUM	3	絞り込み検索の画面用キー 絞り込み検索の日付デフォルト値の設定 デフォルト開始日は終了日より3ヶ月前の日付を設定する。 4ヶ月前の日付を設定すれば下記の3が4に変わる。
WM_MAX_ADDRESS_HISTORY	20	送信画面用キー 送信している時にアドレスの履歴の最大設定 アドレスを入力する時に表示用アドレス履歴の最大数
WM_MAX_PERSONAL_ADDRESS	20	送信画面用キー 送信している時にアドレスの提示の最大設定 アドレスを入力する時に表示用アドレス提示の最大数
WEBMAIL_SEARCH_AMOUNT	50	メール一覧画面用キー メール検索後の表示数

WEBMAIL_DISP_AMOUNT_MAIL	10,20,30	個人設定ページ用キー メール一覧画面にページングのページメール表示数
WEBMAIL_ACCOUNT_NUM_MAX	10	管理者設定画面のWebMail個人情報設定に検索したユーザ数用キー 検索したユーザの表示数
WM_IMAP_SOCKET_TIMEOUT WM_POP3_SOCKET_TIMEOUT	10000 10000	IMAP,POP3サーバに対するソケットタイムアウト時間 WM_IMAP_SOCKET_TIMEOUT WM_POP3_SOCKET_TIMEOUT それぞれのサーバに対するソケットタイムアウト時間(ミリ秒) 省略時はタイムアウトになりません 半角数字以外の値を設定したときも、タイムアウトにはなりません また、1000ミリ秒未満の値を設定しても無視されます
WM_IMAP_CONNECTION_TIMEOUT WM_POP3_CONNECTION_TIMEOUT	10000 10000	IMAP,POPサーバに対するコネクションタイムアウト時間 WM_IMAP_SOCKET_TIMEOUT WM_POP3_SOCKET_TIMEOUT それぞれのサーバに対するコネクションタイムアウト時間(ミリ秒) 省略時はタイムアウトになりません 半角数字以外の値を設定したときも、タイムアウトにはなりません また、1000ミリ秒未満の値を設定しても無視されます
WM_MAX_CONNECTION_ENABLE WM_MAX_CONNECTION	OFF 50	受信サーバに対する最大同時接続数 WM_MAX_CONNECTION 受信サーバに対する最大同時接続数 省略時は50 半角数字以外の値を設定したときも、50になります
WM_ATTACHMENT_YIELD_ENABLE WM_ATTACHMENT_YIELD_COUNT	false 1048576	添付ファイル受信時に大きなファイルを受け取った場合 受信処理を一時中断して、CPUリソースを解放するかどうかの フラグおよび設定 WM_ATTACHMENT_YIELD_ENABLE true : 有効 false: 無効 WM_ATTACHMENT_YIELD_COUNT WM_ATTACHMENT_YIELD_ENABLEがONの時に有効。 指定されたバイト数受け取ったときにいったん、処理を一時中断する。
WEBMAIL_DEBUG_MODE	OFF	メールサーバのデバッグモード ONの場合、Javamailが詳しい情報を出力し、デバッグが簡単になる OFFの場合、Javamailのデバッグ情報が出力されない
WM_DRAFT_NUMBER_OF_LINE	20	下書き一覧ページ用キー 下書き一覧画面にページングのページメール表示数
WEBMAIL_HTMLENCODING_UTF8	ON	HTMLファイルのencodingをUTF-8設定 ONの場合、HTMLのHTMLファイルのencodingをUTF-8設定

		する OFF の場合、HTMLファイルの encodingをインストーラの encodingで表示する
--	--	---

5 索引

H

HTML ・3

J

JAVA ・ii

JavaScript ・ii

M

Microsoft Outlook ・14

V

View 層 ・3

い

イントラネット・スタートパック ・3

き

キャプション ・3

勤怠管理 ・3

く

区分コード ・51

け

掲示板 ・40

こ

個人設定 ・53

し

初期化データ ・50

初期化ファイル ・60

初期表示 ・3

す

スケジュール管理システム ・13

そ

属性を追加 ・23

て

電子会議室 ・31

と

ドキュメント ・26

ドキュメント管理システム ・23

ほ

ポートレット ・33

ろ

ロジック層 ・3

わ

ワークフロー ・40

イントラネット・スタートパック Ver6.1
カスタマイズガイド

初版 : December 31, 2007

Copyright 2007 (株)NTT データイントラマート

TEL: 03-5549-2821

FAX: 03-5549-2816

URL: <http://www.intra-mart.jp/>