

Cosminexus V9

ナビゲーション プラットフォーム 開発ガイド

手引・文法・操作書

3020-3-V42-20

■ 対象製品

P-2443-4P94 uCosminexus Navigation Platform 09-60 (適用 OS : Windows Server 2008 x86, Windows Server 2008 x64*, Windows Server 2008 R2*, Windows Server 2012*)

P-2443-4R94 uCosminexus Navigation Platform - User License 09-60 (適用 OS : Windows Server 2008 x86, Windows Server 2008 x64*, Windows Server 2008 R2*, Windows Server 2012*)

P-2443-4U94 uCosminexus Navigation Platform - Authoring License 09-60 (適用 OS : Windows Server 2008 x86, Windows Server 2008 x64*, Windows Server 2008 R2*, Windows Server 2012*)

P-2443-4V94 uCosminexus Navigation Developer 09-60 (適用 OS : Windows Vista x86, Windows XP, Windows 7 x86, Windows 7 x64*, Windows 8 x86, Windows 8 x64*)

注※ WOW64 環境だけで使用できます。

■ 輸出時の注意

本製品を輸出される場合には、外国為替及び外国貿易法の規制並びに米国輸出管理規則など外国の輸出関連法規をご確認の上、必要な手続きをお取りください。

なお、不明な場合は、弊社担当営業にお問い合わせください。

■ 商標類

GIF は、米国 CompuServe Inc.が開発したフォーマットの名称です。

Internet Explorer は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

iPad は、Apple Inc.の商標です。

Microsoft は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Microsoft Office および Excel は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Oracle と Java は、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。

Safari は、Apple Inc.の商標です。

Windows は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Windows Server は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Windows Vista は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

その他記載の会社名、製品名は、それぞれの会社の商標もしくは登録商標です。

Eclipse は、開発ツールプロバイダのオープンコミュニティである Eclipse Foundation, Inc.により構築された開発ツール統合のためのオープンプラットフォームです。

■ マイクロソフト製品の表記について

このマニュアルでは、マイクロソフト製品の名称を次のように表記しています。

表記		製品名
Excel		Microsoft(R) Office Excel
Internet Explorer	Internet Explorer 6	Microsoft(R) Internet Explorer(R) 6
	Internet Explorer 7	Windows(R) Internet Explorer(R) 7
	Internet Explorer 8	Windows(R) Internet Explorer(R) 8
	Internet Explorer 9	Windows(R) Internet Explorer(R) 9
	Internet Explorer 10	Windows(R) Internet Explorer(R) 10
Windows	Windows 7 Windows 7 x86	Microsoft(R) Windows(R) 7 Enterprise 日本語版(32 ビット版)

表記		製品名	
Windows	Windows 7	Windows 7 x86	Microsoft(R) Windows(R) 7 Professional 日本語版(32 ビット版)
			Microsoft(R) Windows(R) 7 Ultimate 日本語版(32 ビット版)
		Windows 7 x64	Microsoft(R) Windows(R) 7 Enterprise 日本語版(64 ビット版)
			Microsoft(R) Windows(R) 7 Professional 日本語版(64 ビット版)
			Microsoft(R) Windows(R) 7 Ultimate 日本語版(64 ビット版)
		Windows 8	Windows 8 x86
	Windows(R) 8 Pro 日本語版(32 ビット版)		
	Windows 8 x64		Windows(R) 8 Enterprise 日本語版(64 ビット版)
			Windows(R) 8 Pro 日本語版(64 ビット版)
	Windows Server 2008	Windows Server 2008 x86	Microsoft(R) Windows Server(R) 2008 Enterprise 32-bit 日本語版
			Microsoft(R) Windows Server(R) 2008 Standard 32-bit 日本語版
		Windows Server 2008 x64	Microsoft(R) Windows Server(R) 2008 Enterprise 日本語版
			Microsoft(R) Windows Server(R) 2008 Standard 日本語版
		Windows Server 2008 R2	Microsoft(R) Windows Server(R) 2008 R2 Enterprise 日本語版
			Microsoft(R) Windows Server(R) 2008 R2 Standard 日本語版
Windows Server 2012	Microsoft(R) Windows Server(R) 2012 Datacenter 日本語版		
	Microsoft(R) Windows Server(R) 2012 Standard 日本語版		
Windows Vista	Windows Vista x86	Microsoft(R) Windows Vista(R) Business 日本語版(32 ビット版)	
		Microsoft(R) Windows Vista(R) Enterprise 日本語版(32 ビット版)	
		Microsoft(R) Windows Vista(R) Ultimate 日本語版(32 ビット版)	

表記			製品名
Windows	Windows Vista	Windows Vista x64	Microsoft(R) Windows Vista(R) Business 日本語版(64 ビット版)
			Microsoft(R) Windows Vista(R) Enterprise 日本語版(64 ビット版)
			Microsoft(R) Windows Vista(R) Ultimate 日本語版(64 ビット版)
	Windows XP		Microsoft(R) Windows(R) XP Professional Operating System

■ 発行

2013 年 11 月 3020-3-V42-20

■ 著作権

All Rights Reserved. Copyright (C) 2012, 2013, Hitachi, Ltd.

変更内容

変更内容 (3020-3-V42-20) uCosminexus Navigation Platform 09-60, uCosminexus Navigation Platform - User License 09-60, uCosminexus Navigation Platform - Authoring License 09-60, uCosminexus Navigation Developer 09-60

追加・変更内容	変更箇所
プラグインの実行を確認するダイアログの表示有無やプラグインの実行有無が, 入出力プラグイン用 XML ファイル (ioaction.xml) で設定できるようになりました。	3.3.1
ユーザプラグインが使用するメモリ使用量の計算方法を追加しました。	3.13
ucnpOptions パラメタの値を画面 ID 単位で取得できるようになりました。	5.3.1(2), 5.3.2(2)

単なる誤字・脱字などはお断りなく訂正しました。

はじめに

このマニュアルは、次に示すプログラムプロダクトで使用するプラグインやカスタム画面の開発方法、および API リファレンスについて説明したものです。

- uCosminexus Navigation Platform
- uCosminexus Navigation Platform - User License
- uCosminexus Navigation Platform - Authoring License
- uCosminexus Navigation Developer

このマニュアルでは、uCosminexus Navigation Platform と uCosminexus Navigation Platform - User License を総称して **uCosminexus Navigation Platform** と呼びます。また、上記の 4 製品を総称して **ナビゲーション プラットフォーム** と呼びます。

■ 対象読者

このマニュアルは、ナビゲーション プラットフォームが提供する API を使用して、プラグインやカスタム画面を開発するユーザを対象としています。

なお、次に示す知識を持っていることを前提としています。

- Windows の基本的な操作方法
- Java によるプログラム開発に関する基本的な知識
- Eclipse に関する基本的な知識
- XML の基本的な知識

■ このマニュアルの画面や操作の説明で使用している記号

このマニュアルでは、次に示す記号を使用して画面や操作を説明しています。

記号	説明
< >	キーボードのキーを表します。
[]	ダイアログ名、メニュー項目名、ボタン名、および画面に表示されている項目名を表します。
[] - []	一の前に示したメニューから、一の後ろのメニューを選択することを表します。
文字列	文字列が斜体になっている項目は、可変の文字列を表します。

■ このマニュアルの文法で使用している記号

このマニュアルで使用する記号について次に示します。

記号	説明
< >	この記号で囲まれている項目は、該当する要素を指定することを示します。 (例) <プロパティ> プロパティを記述します。
△	半角スペースを示します。

目次

1	開発の概要	1
1.1	開発の範囲	2
1.2	ナビゲーション プラットフォームのプラグインとは	4
1.2.1	プラグインの実行タイミングについて	4
1.2.2	プラグインで受け渡せるデータについて	5
1.2.3	プラグインのインスタンスのライフサイクルについて	5
1.3	プラグイン開発の流れ	8
1.3.1	新規にプラグインを開発する場合の作業	9
1.3.2	プラグインを修正する場合の作業	12
1.4	カスタム画面開発の流れ	14
1.4.1	メニュー領域をカスタマイズする場合の作業	14
1.4.2	新規に画面を開発する場合の作業	15
2	開発の準備	17
2.1	WTP のセットアップ	18
2.2	システム環境変数の設定	19
2.3	pluginSDK プロジェクトのインポート	20
2.4	ライブラリの追加	21
2.4.1	ライブラリの配置場所	21
2.4.2	ライブラリの作成と設定	21
2.4.3	ライブラリ追加時の注意事項	22
3	プラグインの開発	23
3.1	ひな形プラグインの作成	24
3.1.1	プラグイン情報プロパティファイルの編集	24
3.1.2	ひな形プラグイン作成コマンドの実行	28
3.2	ひな形プラグインのプロジェクトのインポート	31
3.3	ひな形プラグインのカスタマイズ	32
3.3.1	入出力プラグインのカスタマイズ	32
3.3.2	中断再開プラグインのカスタマイズ	41
3.4	データベースとの接続処理の追加	43
3.4.1	Cosminexus のリソースアダプタの設定	43
3.4.2	web.xml へのリソース定義の追加	43
3.4.3	cosminexus.xml の編集	43
3.4.4	データベースとの接続処理の実装	44
3.5	アプリケーション属性ファイル (application.xml) の編集	45

3.6	プラグインのビルド	46
3.6.1	プラグインのビルド手順	46
3.6.2	プラグインのビルドに失敗したときの対処	47
3.7	Web サーバリダイレクトの設定	48
3.8	uCosminexus Navigation Developer の J2EE アプリケーションのインポートと開始	50
3.9	業務コンテンツへの関連づけ (入出力プラグインの場合)	52
3.9.1	マッピング線を引く (ガイドパーツとプラグインパーツをつなぐ)	52
3.9.2	プラグインに入力, または出力する値の詳細	55
3.9.3	プラグインパーツを更新する	56
3.9.4	プラグインを使用している業務コンテンツの設定情報を確認する	58
3.10	ユーザプロパティファイルの設定 (中断再開プラグインの場合)	59
3.11	プラグインのデバッグ	60
3.12	プラグインの削除	62
3.12.1	プラグインの削除 (入出力プラグインの場合)	62
3.12.2	プラグインの削除 (中断再開プラグインの場合)	62
3.13	プラグインが使用するメモリ使用量の計算	64
3.13.1	測定用の業務コンテンツの作成手順	64
3.13.2	プラグインが使用するメモリ使用量の測定手順	64
3.14	iPad 向け業務コンテンツで使用するプラグインを開発する場合の注意事項	67
3.15	09-50 より前のバージョンで開発したプラグインを使用する場合の注意事項	68
3.15.1	09-50 より前のバージョンで開発したプラグインを使用する場合の制限	68
3.15.2	09-50 より前のバージョンで開発したプラグインの実行順序とタイミング	68
3.15.3	09-50 より前のバージョンで開発したプラグインのインスタンス	69
4	カスタム画面の開発	71
4.1	カスタマイズに使用するファイルの格納先とパス	72
4.1.1	静的ファイルの格納先とパス	72
4.1.2	JSP ファイルの格納先とパス	72
4.2	メニュー領域のカスタマイズ	73
4.2.1	ナビゲーション プラットフォームで提供している JavaScript 関数	73
4.2.2	メニュー領域の表示形式を変更する	75
4.2.3	メニュー領域に検索機能を追加する	78
4.2.4	iPad 向け実行環境でメニュー領域をカスタマイズする場合の注意事項	79
4.3	新規画面の開発	81
4.3.1	検索機能を追加する	81
4.3.2	ログアウト機能を追加する	82
4.3.3	任意のデータを受け取る機能を追加する	84
4.4	画面の開発後に必要な作業	86
4.4.1	カスタム画面のビルドとデバッグ	86
4.4.2	カスタム画面の取り込み	86

4.4.3	ユーザプロパティファイルの設定	86
5	API リファレンス (プラグイン開発の場合)	89
5.1	API 一覧 (プラグイン開発の場合)	90
5.2	IloActionController (入出力アクションコントローラインタフェース) (09-50 より前のバージョンとの互換用)	91
5.2.1	execute メソッド	91
5.3	IloPluginController (サーバ処理実装インタフェース)	100
5.3.1	inputFromNode メソッド	100
5.3.2	outputToNode メソッド	106
5.3.3	プレビュー時のプラグイン処理	111
5.4	IUCNPSession (セッション情報利用インタフェース)	112
5.4.1	getLoginId メソッド	112
5.5	ISuspendActionController (中断再開アクションコントローラインタフェース)	113
5.5.1	save メソッド	113
5.5.2	load メソッド	115
5.5.3	contains メソッド	116
5.5.4	delete メソッド	117
5.5.5	deleteAll メソッド	119
5.6	ISuspendInfo (中断情報インタフェース)	121
5.7	ParamConvertUtil (入出力パラメータ変換ユーティリティクラス)	122
5.7.1	decodeHtmlPartParam メソッド	122
5.8	SuspendInfoSerializeUtil (中断情報直列化ユーティリティクラス)	126
5.8.1	serialize メソッド	126
5.8.2	deserialize メソッド	127
5.9	UCNPPluginException (入出力プラグイン例外クラス)	129
5.9.1	getMessage メソッド	129
5.9.2	getMessageId メソッド	129
5.10	UCNPPluginException (中断再開プラグイン例外クラス)	131
5.10.1	UCNPPluginException(String message)コンストラクタ	131
5.10.2	UCNPPluginException(String message, Throwable cause)コンストラクタ	131
6	API リファレンス (カスタム画面開発の場合)	133
6.1	API 一覧 (カスタム画面開発の場合)	134
6.2	IHitContentData (業務コンテンツの検索結果格納インタフェース)	135
6.2.1	getHitCount メソッド	135
6.2.2	getHitCounts メソッド	136
6.2.3	getContentId メソッド	136
6.2.4	getContentName メソッド	136
6.2.5	getLastUpdated メソッド	137

6.2.6	getHitGuideDataList メソッド	137
6.3	IHitData (検索結果格納インタフェース)	139
6.3.1	getHitContentList メソッド	139
6.3.2	getWords メソッド	139
6.4	IHitGuideData (ガイドの検索結果格納インタフェース)	141
6.4.1	getHitCount メソッド	141
6.4.2	getHitCounts メソッド	142
6.4.3	getGuideName メソッド	142
6.5	ISearcher (検索インタフェース)	143
6.5.1	search メソッド	143
6.6	LogoutActionUtil (ログアウト処理クラス)	146
6.6.1	logout メソッド	146
6.7	SearcherConstants (検索クラス)	147
6.8	SearcherProvider (検索オブジェクト取得クラス)	148
6.8.1	getSearcher メソッド	148
6.9	UCNPServerErrorException (例外クラス)	149
6.9.1	getMessageId メソッド	149

付録 151

付録 A	サンプルプラグインの使用方法	152
付録 A.1	入出力プラグイン (サンプル) の使用方法	152
付録 A.2	中断再開プラグイン (サンプル) の使用方法	153
付録 A.3	サンプルプラグインを使用するときの注意事項	155
付録 B	検索用サンプル画面の使用方法	156
付録 B.1	検索用サンプル画面の使用手順	156
付録 B.2	検索用サンプル画面を使用するときの注意事項	157
付録 C	このマニュアルの参考情報	158
付録 C.1	関連マニュアル	158
付録 C.2	このマニュアルでの表記	159
付録 C.3	英略語	159
付録 C.4	KB (キロバイト) などの単位表記について	160
付録 D	用語解説	161

索引 165

1

開発の概要

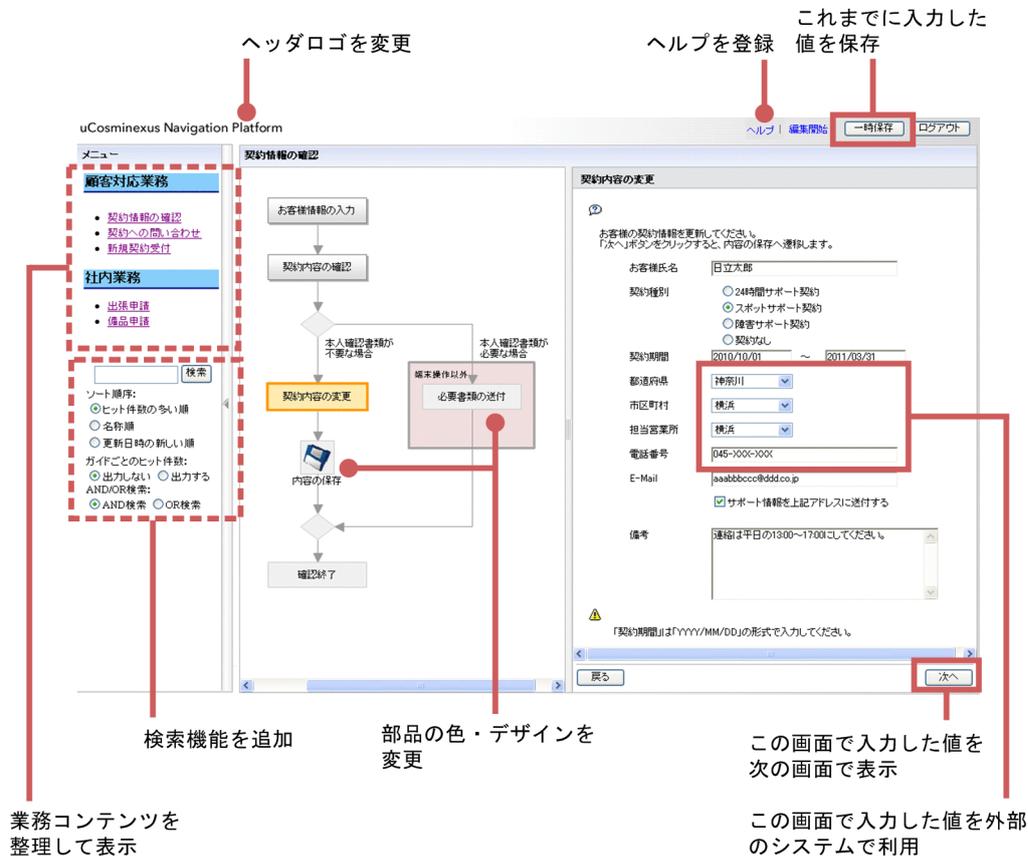
API を使用してプラグインまたはカスタム画面を開発すると、業務に合わせてナビゲーション プラットフォームを運用できます。利用者が入力したデータを利用または保存する場合、または外部プログラムと連携する業務コンテンツを作成したい場合は、プラグインを開発します。一方、業務コンテンツの内容とは関係なく、画面の見映えだけを変更したり、画面に検索機能を追加したりする場合は、カスタム画面を開発します。

この章では、ナビゲーション プラットフォームで使用するプラグインやカスタム画面の開発の概要について説明します。

1.1 開発の範囲

ナビゲーション プラットフォームの画面は、業務の内容に合わせて、例えば次のように変更できます。

図 1-1 ナビゲーション プラットフォームの画面のカスタマイズ例



(凡例)

- : プラグイン開発が必要
- : カスタム画面開発が必要

● : ユーザプロパティファイル (ucnp_user.properties) で設定

枠で示した部分は、プラグインや画面の開発が必要です。このマニュアルでは、枠で示した部分の開発について説明しています。ユーザプロパティファイル (ucnp_user.properties) の設定で実現できるカスタマイズの詳細は、マニュアル「Cosminexus V9 ナビゲーション プラットフォーム 導入から運用まで」を参照してください。

次に、枠で示した部分を説明します。

プラグイン開発

表示されている業務コンテンツの画面で利用者が入力した値を次のステップで利用したり、外部プログラムへ渡したりするには、入出力プラグインを開発してください。

利用者がこれまでに入力した値を一時保存するには、中断再開プラグインを開発してください。入力した値は中断情報として保存され、利用者は業務の途中にはほかの業務を参照したり、ログアウトしたりしても、一時保存した状態から業務を再開できるようになります。

利用者が入力した値をキャッシュといいます。プラグインはキャッシュを使って機能を実現します。

カスタム画面開発

メニュー領域の背景色の変更や画像の表示など、簡単な見映えだけを変更する場合は、HTML ファイルでカスタム画面を開発してください。業務コンテンツの数が多く、検索機能を追加するなど高度なカスタマイズが必要な場合は、JSP ファイルでカスタム画面を開発してください。

参考

- JSP ファイルを使用すると、ナビゲーション プラットフォームの画面とは別の新規画面としてカスタム画面を開発することもできます。
 - ナビゲーション プラットフォームのログイン画面は、ユーザプロパティファイル (ucnp_user.properties) の設定によってカスタマイズできます。
-

1.2 ナビゲーションプラットフォームのプラグインとは

ナビゲーションプラットフォームが提供するプラグインには、開発が不要なシステムプラグインと、開発が必要なユーザプラグインがあります。ここではユーザプラグイン（以降プラグインと呼びます）について説明します。プラグインを作成すると、利用者が画面に入力した内容や設定した内容を利用して外部プログラムと連携したり、外部プログラムから受け取ったデータを画面の各パーツの入力値として利用したりできます。利用者が画面に入力した内容や設定した内容をキャッシュといいます。

ナビゲーションプラットフォームでプラグインを使用するには、まず、開発者が開発環境でプラグインを開発して、ナビゲーションプラットフォームのEARファイルに組み込みます。さらに、システム管理者がプラグインを組み込んだEARファイルを編集環境や実行環境にデプロイする必要があります。開発環境での作業の概要は「1.3 プラグイン開発の流れ」を参照してください。

なお、09-50より前のバージョンで開発したプラグインを使用する場合は、「3.15 09-50より前のバージョンで開発したプラグインを使用する場合の注意事項」を参照してください。

プラグインには次の種類があります。

- 入出力プラグイン
業務コンテンツのノード遷移時に実行するプラグインです。業務編集画面で、プラグインを利用する業務コンテンツに入出力プラグインを関連づけます。ノードに入出力プラグインを配置する方法は「3.9 業務コンテンツへの関連づけ（入出力プラグインの場合）」を参照してください。
- 中断再開プラグイン
利用者が業務実行画面で入力した情報を一時保存して、業務を再開したときに一時保存した情報を復元するためのプラグインです。

1.2.1 プラグインの実行タイミングについて

入出力プラグインおよび中断再開プラグインが実行されるタイミングについて説明します。

(1) 入出力プラグインの実行タイミング

業務コンテンツのノード遷移時に実行する入出力プラグインの順序を次の表に示します。

表 1-1 ノード遷移時に実行する入出力プラグインとメソッド

実行順序	入出力プラグイン	入出力プラグインのメソッド
1	遷移元ノードに関連づけている入出力プラグイン	inputFromNode メソッド (IIOPluginController (サーバ処理実装インタフェース))
2	遷移先ノードに関連づけている入出力プラグイン※	outputToNode メソッド (IIOPluginController (サーバ処理実装インタフェース))

注※

ターミナルノード（開始）に接続されていない、先頭のプロセスノードに関連づけている入出力プラグインも含まれません。

なお、業務実行画面（プレビュー）上でノード遷移をした場合でも、入出力プラグインが実行されます。プレビュー時に入出力プラグインを実行させたくないときは、プレビュー時に入出力プラグインが実行されな

ように実装する必要があります。実装の詳細は「5.3.3 プレビュー時のプラグイン処理」を参照してください。

また、表 1-1 に示す入出力プラグインのメソッドの戻り値にエラー情報を設定すると、入出力プラグインの実行時に利用者にアラートを表示したり、入力誤りがある項目を強調表示してノードの遷移を抑止したりできます。

メソッドの戻り値に設定する内容については「5.3 IIoPluginController (サーバ処理実装インタフェース)」の `inputFromNode` メソッドまたは `outputToNode` メソッドの戻り値を参照してください。

(2) 中断再開プラグインの実行タイミング

中断再開プラグインは次のタイミングで実行します。

中断する場合

利用者が [一時保存] ボタンをクリックしたとき

再開する場合

- 利用者が業務実行画面で業務コンテンツを選択したとき
- カスタマイズしたメニュー領域に表示されているアンカーなどをクリックしたとき
- 基本 URL にパラメタを指定して特定の業務コンテンツを開いたとき

1.2.2 プラグインで受け渡せるデータについて

入出力プラグインで受け渡せるデータについて説明します。

(1) 入出力プラグインで受け渡せるデータ

プラグインで受け取れるデータ

入出力プラグイン実行時に、遷移元または遷移先のノードの情報、およびプラグインパーツのパラメタにマッピングしたガイドパーツのキャッシュなどのデータを受け取れます。データは IIoPluginController (サーバ処理実装インタフェース) の `inputFromNode` メソッド、または `outputToNode` メソッドのパラメタとして渡されます。

プラグインが返すデータ

プラグインが返すデータで、パラメタにマッピングしたガイドパーツのキャッシュの更新、およびエラー情報を設定し、利用者にアラートを表示したり、入力誤りがある項目を強調表示してノードの遷移を抑止したりできます。

データは IIoPluginController (サーバ処理実装インタフェース) の `inputFromNode` メソッド、または `outputToNode` メソッドの戻り値として返してください。

入出力プラグインと受け渡せるデータの詳細は「5.3 IIoPluginController (サーバ処理実装インタフェース)」の次の個所を参照してください。

- `inputFromNode` メソッドの `param` および戻り値
- `outputToNode` メソッドの `param` および戻り値

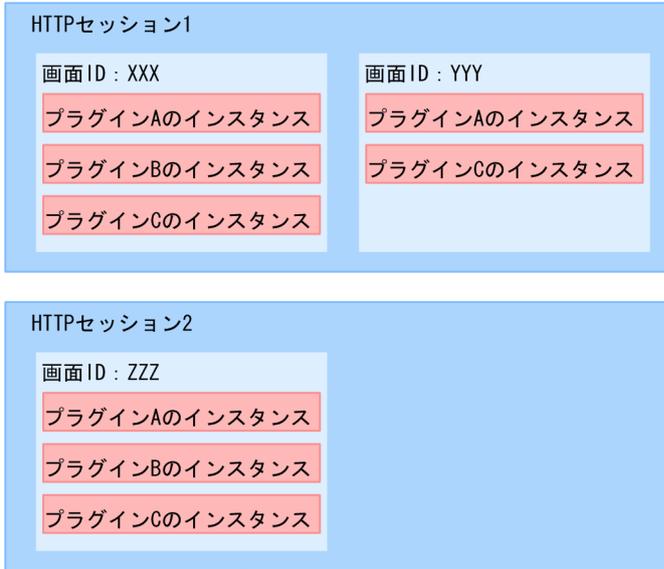
1.2.3 プラグインのインスタンスのライフサイクルについて

プラグインのインスタンスのライフサイクルについて説明します。

(1) 入出力プラグインのインスタンスの生成および保持

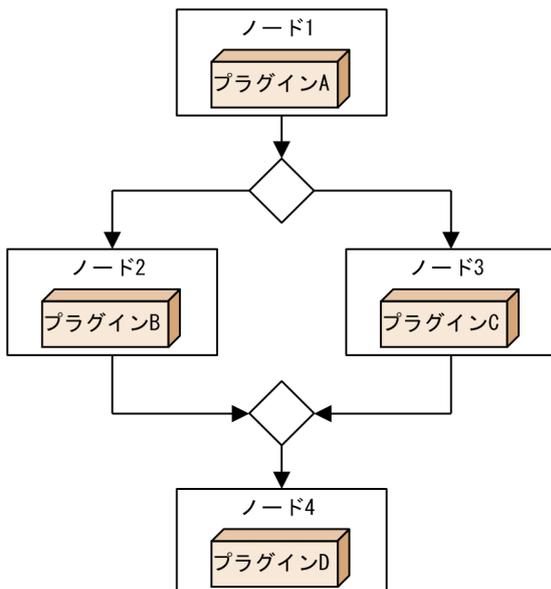
同一プラグイン ID の入出力プラグインのインスタンスは、一つの画面につき一つずつ保持されます。各インスタンスは、画面 ID をキーにして HTTP セッションに保持されます。入出力プラグインのインスタンスについて、次の図に示します。

図 1-2 入出力プラグインのインスタンス



入出力プラグインを実行するときに、HTTPセッションから画面IDとプラグインIDをキーにして、実行する入出力プラグインのインスタンスを取得します。HTTPセッションからインスタンスを取得できない場合は、新たなインスタンスを生成してHTTPセッションに追加します。このとき生成するのは実行するインスタンスだけです。例えば、次の図のように分岐ノードがある業務コンテンツを実行する際、ノード1→ノード2→ノード4の順で遷移した場合は、プラグインA、プラグインB、プラグインDだけ実行されて、プラグインCは実行されません。したがって、この遷移で業務コンテンツを実行したときは、プラグインCのインスタンスは生成しません。

図 1-3 分岐がある業務コンテンツ



(2) 入出力プラグインのインスタンスの破棄

プラグインのインスタンスが破棄されるタイミングを次の表に示します。破棄するタイミングは業務実行画面、および業務実行画面（プレビュー）のどちらで操作した場合も同じです。

表 1-2 プラグインのインスタンスが破棄されるタイミング

項番	タイミング	破棄されるプラグインのインスタンス
1	業務実行画面、または業務実行画面（プレビュー）で業務コンテンツを表示したとき ※1	操作を実行したときの HTTP セッションが、操作した画面の画面 ID をキーにして保持しているインスタンス
2	[編集開始] メニューをクリックしたとき	
3	[閉じる] ボタンをクリックしたとき※2	
4	[完了] ボタンをクリックしたとき※3	
5	[ログアウト] ボタンをクリックしたとき	破棄する HTTP セッションが保持している、すべてのインスタンス
6	セッションがタイムアウトしたとき	

注※1

メニュー領域、またはカスタマイズしたメニュー領域から業務コンテンツを選択した場合にインスタンスが破棄されます。基本 URL にパラメータを指定して特定の業務コンテンツを開いた場合は、インスタンスは破棄されません。

注※2

ユーザプロパティファイル (ucnp_user.properties) の ucnp.base.server.close.button.setting プロパティに [logout_close] が指定されている場合に [閉じる] ボタンをクリックしたときは、項番 5 の [ログアウト] ボタンをクリックしたときと同じインスタンスを破棄します。プロパティの詳細はマニュアル「Cosminexus V9 ナビゲーション プラットフォーム 導入から運用まで」の ucnp.base.server.close.button.setting プロパティ（ヘッダ領域の [閉じる] ボタンの表示有無）についての説明を参照してください。

注※3

ユーザプロパティファイル (ucnp_user.properties) の ucnp.base.client.complete.button.window.close プロパティに [true] が指定されている場合だけインスタンスを破棄します。プロパティの詳細はマニュアル「Cosminexus V9 ナビゲーション プラットフォーム 導入から運用まで」の ucnp.base.client.complete.button.window.close プロパティ（業務実行画面で [完了] ボタンをクリックしたときの Web ブラウザ動作）についての説明を参照してください。

(3) 中断再開プラグインのライフサイクル

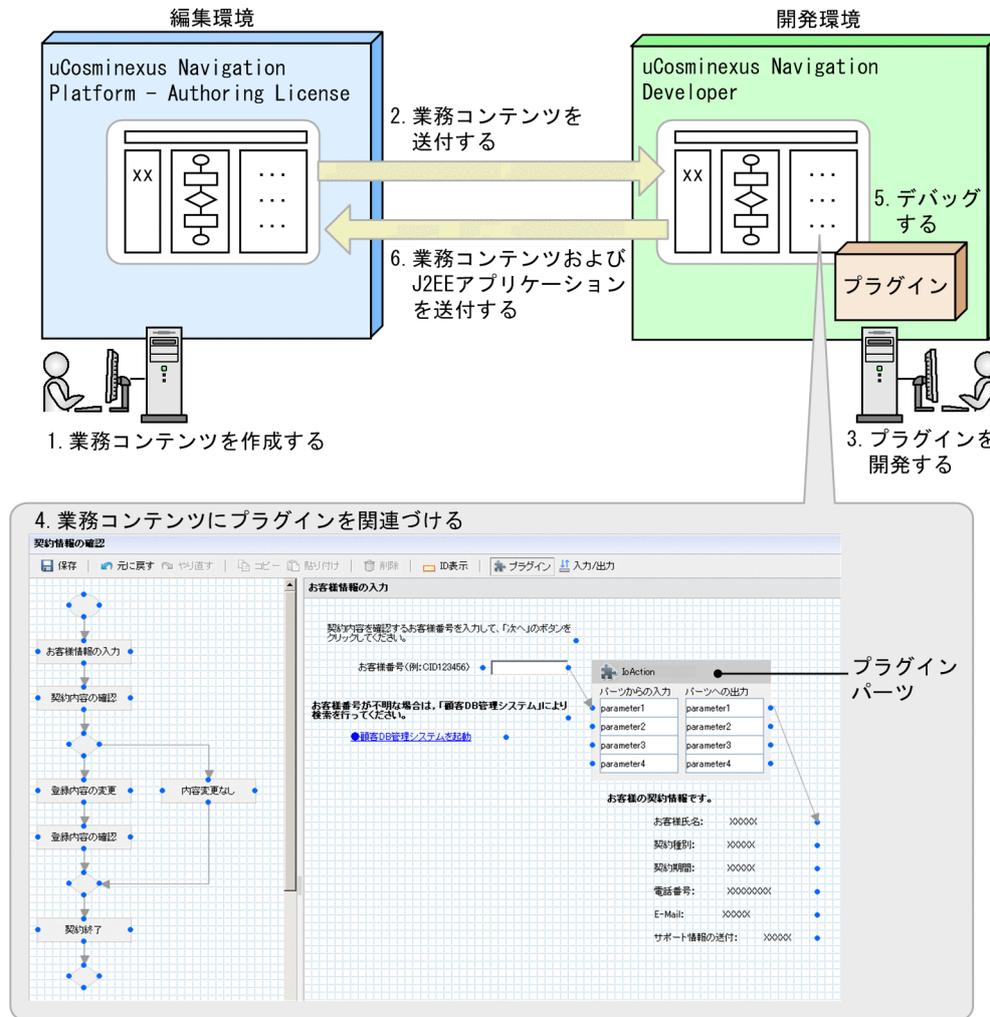
中断再開プラグインは次のタイミングで中断情報が削除されて終了します。

- ユーザが業務実行画面で [完了] ボタンをクリックしたとき
- ユーザが業務実行画面で業務を再開した際に、中断情報のチェックでエラーが発生したとき
- 業務コンテンツを作成するユーザが、業務編集画面で業務コンテンツを編集して保存したとき
- 業務コンテンツを作成するユーザが、業務編集画面で業務コンテンツを削除したとき
- 業務コンテンツを作成するユーザが、業務編集画面で業務コンテンツを選択し、CSV ファイルで業務コンテンツを更新したとき
- プラグインを開発するユーザが、業務編集画面で [プラグインパーツ更新] メニューを選択して、業務コンテンツを更新したとき

1.3 プラグイン開発の流れ

この節では、プラグイン開発の概要と作業の流れを説明します。なお、プラグイン開発は必ず Windows の管理者権限を持つユーザで実行してください。

図 1-4 プラグイン開発の概要



図中の番号は、次に示す番号と対応しています。

1. 編集環境の業務編集画面で業務コンテンツを作成します。
ただし、編集環境を使用しないシステム構成の場合は、業務コンテンツの作成も開発環境で実施します。システム構成および業務コンテンツを新規に作成する方法については、マニュアル「Cosminexus V9 ナビゲーション プラットフォーム 導入から運用まで」を参照してください。
2. 作成した業務コンテンツを編集環境から開発環境に送付します。
3. 開発環境で Eclipse を使用してプラグインを開発します。
4. 入出力プラグインの場合は、手順 2. で送付された業務コンテンツを開発環境の業務編集画面で表示し、プラグインパーツとガイドパーツの間にマッピング線を引くことで業務コンテンツにプラグインを関連づけます。これで業務コンテンツは完成です。
5. 開発環境でプラグインのデバッグを実施します。

6. プラグインのデバッグ後、完成した業務コンテンツを開発環境から編集環境に送付します。さらに、J2EE アプリケーション（EAR ファイル）も編集環境に送付します。

参考

手順 2.および手順 6.の作業は、ナビゲーション プラットフォームのシステム管理者が実施します。システム管理者の作業については、マニュアル「Cosminexus V9 ナビゲーション プラットフォーム 導入から運用まで」を参照してください。

1.3.1 新規にプラグインを開発する場合の作業

ここでは、新規にプラグインを開発する場合の作業の流れを説明します。

開発を始める前に

ここで説明する作業を始める前に、開発環境を構築してください。さらに入出力プラグインの場合は、編集環境で作成した業務コンテンツを開発環境へインポートしておく必要があります。

開発が完了したら

プラグインの開発が完了したら、J2EE アプリケーション（ucnp.ear）を編集環境と実行環境に送付します。入出力プラグインを開発した場合は、プラグインに関連づけた業務コンテンツもエクスポートして、編集環境と実行環境にインポートしてください。

各手順の詳細は、マニュアル「Cosminexus V9 ナビゲーション プラットフォーム 導入から運用まで」を参照してください。

(1) 入出力プラグインの開発

入出力プラグインを新規に開発する場合の流れを次に示します。

図 1-5 新規の入出力プラグイン開発の流れ



1 開発の概要

作業の内容と参照先を次に示します。

表 1-3 新規の入出力プラグイン開発の作業内容

図中の番号	作業の内容	参照先
1	Eclipse をセットアップし、ナビゲーション プラットフォームが提供する Eclipse のプロジェクト (pluginSDK) をインポートします。そのほかに、システム環境変数の設定、ライブラリの追加も実施します。 なお、すでに中断再開プラグインを開発している場合は、この作業は不要です。	2 章
2	プラグイン情報プロパティファイルを編集してから、ひな形プラグイン作成コマンドを実行します。 コマンドを実行すると、 ひな形プラグイン ※が Eclipse の Java プロジェクトの形式で作成されます。	3.1
3	ひな形プラグインのプロジェクトを Eclipse にインポートします。	3.2
4	Eclipse を使用して、ひな形プラグインをカスタマイズ (編集) します。データベース接続処理が必要な場合は、ここで追加します。	3.3.1, 3.4
5	アプリケーション属性ファイル (application.xml) を編集して、プラグインを有効にします。	3.5
6	Ant を使用して、カスタマイズしたひな形プラグインのプロジェクトをビルドします。	3.6
7	Cosminexus Management Server を使用して、Web サーバリダイレクトの設定を実行します。	3.7
8	Ant を使用して、開発環境の J2EE アプリケーション (ucnp.ear) をインポートして、開始します。	3.8
9	ノードのガイド領域にプラグインパーツを配置し、ガイドパーツとの間にマッピング線を引くことで、業務コンテンツにプラグインを関連づけます。	3.9
10	Eclipse のデバッグ機能を使用して、プラグインをデバッグします。	3.11

注※

ひな形プラグインとは、プラグインの基となるファイル群です。ひな形プラグインをカスタマイズすることで、容易にプラグインを開発できます。

(2) 中断再開プラグインの開発

中断再開プラグインを新規に開発する場合の流れを次に示します。

図 1-6 新規の中断再開プラグイン開発の流れ



作業の内容と参照先を次に示します。

表 1-4 新規の中断再開プラグイン開発の作業内容

図中の番号	作業の内容	参照先
1	Eclipse をセットアップし、ナビゲーション プラットフォームが提供する Eclipse のプロジェクト (pluginSDK) をインポートします。そのほかに、システム環境変数の設定、ライブラリの追加も実施します。 なお、すでに入出力プラグインを開発している場合は、この作業は不要です。	2 章
2	プラグイン情報プロパティファイルを編集してから、ひな形プラグイン作成コマンドを実行します。 コマンドを実行すると、 ひな形プラグイン [*] が Eclipse の Java プロジェクトの形式で作成されます。	3.1
3	ひな形プラグインのプロジェクトを Eclipse にインポートします。	3.2
4	Eclipse を使用して、ひな形プラグインをカスタマイズ (編集) します。データベース接続処理が必要な場合は、ここで追加します。	3.3.2, 3.4
5	アプリケーション属性ファイル (application.xml) を編集して、プラグインを有効にします。	3.5
6	Ant を使用して、カスタマイズしたひな形プラグインのプロジェクトをビルドします。	3.6
7	Ant を使用して、開発環境の J2EE アプリケーション (ucnp.ear) をインポートして、開始します。	3.8
8	ユーザプロパティファイル (ucnp_user.properties) を編集して、次の設定をします。 • 中断再開プラグインの使用有無	3.10

図中の番号	作業の内容	参照先
9	Eclipse のデバッグ機能を使用して、プラグインをデバッグします。	3.11

注※

ひな形プラグインとは、プラグインの基となるファイル群です。ひな形プラグインをカスタマイズすることで、容易にプラグインを開発できます。

1.3.2 プラグインを修正する場合の作業

既存のプラグインを修正する場合の作業の流れを説明します。

修正が完了したら

プラグインの修正が完了したら、J2EE アプリケーション (ucnp.ear) を編集環境と実行環境に送付します。入出力プラグインを修正した場合は、プラグインを関連づけた業務コンテンツもエクスポートして、編集環境と実行環境にインポートしてください。各手順の詳細は、マニュアル「Cosminexus V9 ナビゲーション プラットフォーム 導入から運用まで」を参照してください。

(1) 入出力プラグインの修正

開発済みの入出力プラグインを修正する場合の流れを次に示します。

図 1-7 既存の入出力プラグイン修正の流れ



作業の内容と参照先を次に示します。

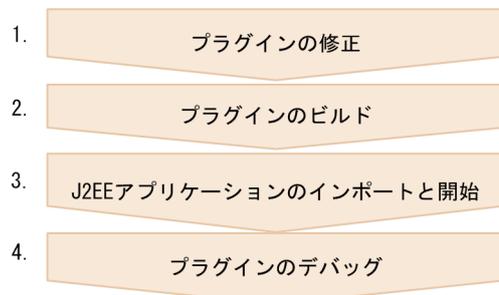
表 1-5 既存の入出力プラグイン修正の作業内容

図中の番号	作業の内容	参照先
1	Eclipse を使用して、開発済みの入出力プラグインをカスタマイズ (編集) します。	3.3.1, 3.4
2	Ant を使用して、カスタマイズしたプラグインのプロジェクトをビルドします。	3.6
3	Ant を使用して、開発環境の J2EE アプリケーション (ucnp.ear) をインポートして、開始します。	3.8
4	ガイドパーツとプラグインパーツの間にマッピング線を引いて、業務コンテンツにプラグインを関連づけます。	3.9
5	Eclipse のデバッグ機能を使用して、プラグインをデバッグします。	3.11

(2) 中断再開プラグインの修正

開発済みの中断再開プラグインを修正する場合の流れを次に示します。

図 1-8 既存の中断再開プラグイン修正の流れ



作業の内容と参照先を次に示します。

表 1-6 既存の中断再開プラグイン修正の作業内容

図中の番号	作業の内容	参照先
1	Eclipse を使用して、開発済みの中断再開プラグインをカスタマイズ（編集）します。	3.3.2, 3.4
2	Ant を使用して、カスタマイズしたプラグインのプロジェクトをビルドします。	3.6
3	Ant を使用して、開発環境の J2EE アプリケーション（ucnp.ear）をインポートして、開始します。	3.8
4	Eclipse のデバッグ機能を使用して、プラグインをデバッグします。	3.11

1.4 カスタム画面開発の流れ

この節では、カスタム画面を開発して、メニュー領域をカスタマイズする場合の作業の流れを説明します。カスタム画面には、ナビゲーションプラットフォームの業務実行画面のメニュー領域をカスタマイズしたものと、ナビゲーションプラットフォームの画面とは別に新規に開発する画面との2種類があります。

参考

カスタム画面で検索機能を使用できるようにするためには、あらかじめ検索機能用の追加セットアップを実施しておく必要があります。追加セットアップの詳細は、マニュアル「Cosminexus V9 ナビゲーションプラットフォーム 導入から運用まで」を参照してください。

1.4.1 メニュー領域をカスタマイズする場合の作業

メニュー領域をカスタマイズする場合の流れを次に示します。

図 1-9 メニュー領域カスタマイズの流れ



作業の内容と参照先を次に示します。

表 1-7 メニュー領域カスタマイズの作業内容

図中の番号	作業の内容	参照先
1*	Eclipse をセットアップし、ナビゲーションプラットフォームが提供する Eclipse のプロジェクト (pluginSDK) をインポートします。そのほかに、システム環境変数の設定、ライブラリの追加も実施します。	2 章
2	カスタマイズに使用する静的ファイル (画像ファイルや HTML ファイルなど)、または JSP ファイルを作成します。 JSP ファイルを作成する場合、ナビゲーションプラットフォームで提供している JavaScript および Java の API を利用できます。	4.2
3	作成した静的ファイルまたは JSP ファイルを決められたディレクトリに格納します。	4.1
4	ユーザプロパティファイル (ucnp_user.properties) を編集して、次の設定をします。設定後、J2EE サーバを再起動します。	4.4.3

図中の番号	作業の内容	参照先
4	<ul style="list-style-type: none"> 3で格納したファイルのパス 検索機能の使用有無 	4.4.3
5*	Ant を使用して、Eclipse のプロジェクト (pluginSDK) をビルドします。	3.6
6*	Ant を使用して、開発環境の J2EE アプリケーション (ucnp.ear) をインポートして、開始します。	3.8
7*	Eclipse のデバグ機能を使用して、開発環境の J2EE アプリケーションをデバグします。	3.11

注※

- プラグインを開発する場合と共通の作業です。
- 静的ファイルだけを使用するときは、必要ありません。なお、静的ファイルだけを使用したカスタマイズは、Eclipse での作業が不要のため、編集環境で実施することもできます。

1.4.2 新規に画面を開発する場合の作業

新規に画面を開発して、検索機能を追加する流れを説明します。

図 1-10 新規の画面開発の流れ



作業の内容と参照先を次に示します。

表 1-8 新規の画面開発の作業内容

図中の番号	作業の内容	参照先
1*	Eclipse をセットアップし、ナビゲーション プラットフォームが提供する Eclipse のプロジェクト (pluginSDK) をインポートします。そのほかに、システム環境変数の設定、ライブラリの追加も実施します。	2 章

1 開発の概要

図中の番号	作業の内容	参照先
2	新規画面用の JSP ファイルを作成します。ナビゲーションプラットフォームで提供している Java の API を利用できます。ログアウト処理やデータの受け渡し処理が必要な場合は、ここで追加します。	4.3.1, 4.3.2, 4.3.3
3	作成した JSP ファイルを決められたディレクトリに格納します。	4.1.2
4	ユーザプロパティファイル (ucnp_user.properties) を編集して、次の設定をします。 <ul style="list-style-type: none">• 検索機能の使用有無• [ログアウト] ボタンの表示有無• [閉じる] ボタンの表示有無	4.4.3
5*	Ant を使用して、Eclipse のプロジェクト (pluginSDK) をビルドします。	3.6
6*	Ant を使用して、開発環境の J2EE アプリケーション (ucnp.ear) をインポートして、開始します。	3.8
7*	Eclipse のデバッグ機能を使用して、開発環境の J2EE アプリケーションをデバッグします。	3.11

注※

プラグインを開発する場合と共通の作業です。

2

開発の準備

この章では、プラグイン開発およびカスタム画面の開発に必要なセットアップや環境変数の設定などについて説明します。

なお、開発の準備を始める前に、開発環境を構築し、さらに業務コンテンツを作成または編集環境からインポートしておく必要があります。これらの作業については、マニュアル「Cosminexus V9 ナビゲーション プラットフォーム 導入から運用まで」を参照してください。

2.1 WTP のセットアップ

WTP をセットアップします。セットアップ方法については、マニュアル「Cosminexus V9 アプリケーションサーバ アプリケーション開発ガイド」を参照してください。

なお、このマニュアルで「Eclipse」と記載しているものは、ここでセットアップした WTP を示していません。

！ 注意事項

JDK はバージョン 6 を使用してください。ただし、プラグインおよびカスタム画面から参照するライブラリについては、JDK のバージョン 5 でビルドしたライブラリを使用することもできます。

2.2 システム環境変数の設定

uCosminexus Navigation Developer では、Ant を使用してひな形プラグインの生成を実行します。

ひな形プラグインの生成を実行する前に、システム環境変数「Path」に次に示すディレクトリを追加してください。

追加するディレクトリ

*Eclipse*のインストールディレクトリ¥plugins¥org.apache.ant_で始まるディレクトリ¥bin;

注意事項

Ant のバージョンは 1.7 以降である必要があります。システム環境変数を設定したら、次のコマンドを実行して Ant のバージョンを確認してください。

```
ant -version
```

2.3 pluginSDK プロジェクトのインポート

ナビゲーション プラットフォームが提供する pluginSDK プロジェクトを利用して、プラグインまたはカスタム画面の開発に必要な環境を準備します。

pluginSDK のプロジェクトを Eclipse にインポートする手順を次に示します。

1. Eclipse を起動します。
2. Eclipse で [ファイル] - [インポート] を選択します。
[インポート] 画面が表示されます。
3. [一般] - [既存プロジェクトをワークスペースへ] を選択します。
4. [次へ] ボタンをクリックします。
5. [ルート・ディレクトリーの選択] テキストボックスに [uCosminexus Navigation Developer インストールディレクトリ\pluginSDK] と指定します。
6. [終了] ボタンをクリックします。
pluginSDK のプロジェクトが Eclipse に追加されます。

注意事項

[プロジェクトをワークスペースにコピー] はチェックしないでください。

2.4 ライブラリの追加

この節では、プラグインおよびカスタム画面で利用される Java のライブラリ (JAR ファイル) の追加について説明します。

! 注意事項

Windows Vista または Windows 7 上で開発を実施する場合、Windows の管理者権限があるユーザでライブラリを追加してください。ライブラリを OS のインストールドライブ: %Program Files ディレクトリ以下などに追加する (ファイルを新規作成またはコピーする) ときに、Windows の管理者権限がないユーザが追加すると、ファイルがユーザフォルダにリダイレクトされてしまうことがあります。

2.4.1 ライブラリの配置場所

プラグインまたはカスタム画面で利用するライブラリを格納するディレクトリを説明します。ライブラリを格納するディレクトリは、そのライブラリを J2EE サーバ全体で使用するかどうかによって異なります。

ナビゲーション プラットフォームだけでライブラリを使用する場合

アプリケーションクラスローダのレイヤにライブラリを配置します。次のディレクトリにライブラリを格納してください。

ナビゲーション プラットフォームインストールディレクトリ%usrlib%app

補足事項

すでに「ナビゲーション プラットフォームインストールディレクトリ%pluginSDK%plugin%プラグイン ID%WEB-INF%lib」にライブラリを配置してプラグインを開発している場合は、引き続きこのディレクトリのライブラリを使用することもできます。

J2EE サーバ全体でライブラリを使用する場合

J2EE サーバの共通ライブラリであるコンテナ拡張ライブラリとして、システムクラスローダのレイヤの任意の場所にライブラリを配置します。コンテナ拡張ライブラリの詳細は、マニュアル「Cosminexus V9 アプリケーションサーバ 機能解説 拡張編」を参照してください。

なお、プラグインを開発する場合は、次のディレクトリにもライブラリを格納してください。

ナビゲーション プラットフォームインストールディレクトリ%usrlib%sys

注意事項

システムクラスローダには、ナビゲーション プラットフォームが提供する API を参照するライブラリは指定できません。指定した場合、実行時にエラーが発生します。

2.4.2 ライブラリの作成と設定

ライブラリを作成するときの注意点、および作成したライブラリの設定手順を説明します。また、設定したライブラリの参照解決を Eclipse で実施する手順についても説明します。

(1) ライブラリ作成時の注意

ナビゲーション プラットフォームが提供する API を参照するライブラリを作成する場合は、次のクラスパスの設定を含める必要があります。

ナビゲーション プラットフォームインストールディレクトリ%lib%ucnpclasses.jar

(2) ライブラリの設定

ライブラリを設定する手順を次に示します。

1. 作成したライブラリを、開発環境の決められたディレクトリに格納します。
格納場所の詳細は、「2.4.1 ライブラリの配置場所」を参照してください。
2. コンテナ拡張ライブラリを使用する場合は、開発環境だけでなく、編集環境および実行環境の J2EE サーバでも設定を実施します。

(3) プロジェクトの参照解決

Eclipse のプロジェクトで参照解決する場合は、次の手順で追加したライブラリ JAR ファイルを指定してください。

1. Eclipse の [プロジェクト・エクスプローラー] のプロジェクトを右クリックします。
メニューが表示されます。
2. [プロパティ] をクリックします。
3. 左ペインの [Java のビルド・パス] をクリックします。
4. 右ペインの [ライブラリー] タブをクリックします。
5. [JAR の追加] ボタンをクリックして、追加した JAR ファイルを指定します。

2.4.3 ライブラリ追加時の注意事項

ライブラリ追加時の注意事項を次に示します。

- アプリケーションクラスローダのレイヤにライブラリを設定する場合は、次に示すファイル名をライブラリの JAR ファイル名に使用しないでください。
なお、大文字と小文字は区別されません。
 - ucnp で始まるファイル名
 - dwr_ucnp.jar
 - jxpandweb.jar
- [ナビゲーション プラットフォームインストールディレクトリ¥pluginSDK¥plugin¥プラグイン ID ¥WEB-INF¥lib] で同じ名称のファイルを配置する場合は、ファイルの内容も同じにする必要があります。
- ライブラリの配置場所には、開発に必要なライブラリ以外を格納しないでください。不要なファイルやディレクトリを格納すると、ビルド時などにエラーの原因となるおそれがあります。

3

プラグインの開発

この章では、プラグインの開発方法について説明します。

3.1 ひな形プラグインの作成

ひな形プラグインの作成手順を次に示します。

1. プラグイン情報プロパティファイルの編集
2. ひな形プラグイン作成コマンドの実行

この節では、それぞれの手順について説明します。

参考

ここで説明する作業は一つのプラグインにつき一度だけ実行してください。作成済みのプラグインを修正する場合は、この作業は不要です。

3.1.1 プラグイン情報プロパティファイルの編集

プラグイン情報プロパティファイルとは、ひな形プラグインを作成するための情報を定義したプロパティファイルです。

次に示すサンプルファイルを編集して、プラグイン情報プロパティファイルを作成してください。

`uCosminexus Navigation Developer` インストールディレクトリ¥`pluginSDK¥plugin.properties.sam`

編集後のファイル名は「任意の文字列.properties」としてください。

(1) プロパティファイル作成時の注意

サンプルファイルを編集してプラグイン情報プロパティファイルを作成する際の注意事項を次に示します。

- プロパティファイルの文字は、ISO 8859-1 (Latin1) でエンコードされます。ISO 8859-1 以外の文字コードの文字は使用できません。
- Windows の予約デバイス名 (CON, AUX, COM n (n は 1~9), LPT n (n は 1~9), PRN, NUL, CLOCK\$) は使用できません。
- 行頭文字に「#」または「!」を指定した場合、行全体がコメントと見なされます。
- 行頭に続く空白文字 (半角スペース, タブまたは改行) は無視されます。

(2) プロパティファイルの記述形式

プロパティファイルの記述形式の例を次に示します。なお、説明中の「空白文字」は半角スペース, タブまたは改行を指します。

プロパティキー=値

- プロパティキーと値の間には「:」または「=」を記述してください。プロパティキー, 「:」または「=」, 値の間に空白文字を指定しても無視されます。
- プロパティキーの次の文字 (空白文字を除く) が「:」または「=」の場合, その次の文字 (空白文字を除く) から行末文字までが値として解釈されます。
- プロパティキーの次の文字 (空白文字を除く) が「:」または「=」でない場合, その文字から行末文字までが値として解釈されます。
- 値の後ろに空白文字を付加した場合, 値の一部と解釈されます。
- 値の中に「:」, 「=」, 「#」, または「!」が使用されている場合, 値の一部と解釈されます。

(3) プロパティキーの詳細

プラグイン情報プロパティファイルに指定するプロパティキーを次に示します。

- すべてのプラグインで使用するプロパティキー
 - userplugin.id
 - userplugin.name
 - userplugin.version
 - userplugin.java.package
- 入出力プラグインで使用するプロパティキー
 - userplugin.server.controller.ioaction
 - userplugin.server.controller.ioaction.type
- 中断再開プラグインで使用するプロパティキー
 - userplugin.server.controller.suspend

各プロパティキーの詳細を次に示します。

(a) userplugin.id

プラグイン ID を指定します。プラグイン ID はシステム内でユニークになるようにしてください。なお、中断再開プラグインの場合、必ず固定値「ucnp.plugin.suspend」を指定してください。

使用できる文字

- 半角英数字
- 半角ピリオド (.)
- 半角アンダースコア (_)

! 注意事項

使用できる文字の組み合わせの制限を次に示します。

- 中断再開プラグインはシステム内で一つしか登録できないため、固定値「ucnp.plugin.suspend」で指定する必要があります。「ucnp.plugin.suspend」以外の値を設定した場合、中断再開プラグインではなく、入出力プラグインとして識別されます。
- 入出力プラグインの場合、末尾にピリオドは指定しないでください。また、「ucnp」で始まる ID は指定できません。なお、「ucnp」の大文字、小文字は区別されないため、「UCNP」や「uCnp」も指定できません。

指定できる文字列長

1~64 バイト

指定する値の例

userplugin

(b) userplugin.name

プラグイン名を指定します。プラグイン名はシステム内でユニークになるようにしてください。

使用できる文字

- 半角英数字

ただし、「ucnp」で始まるプラグイン名は指定できません。なお、「ucnp」の大文字、小文字は区別されないため、「UCNP」や「uCnp」も指定できません。また、「env」および「AppName」はプラグイン名として指定できません。

指定できる文字列長

1～31 バイト

指定する値の例

userplugin

(c) userplugin.version

プラグインのバージョンを指定します。

使用できる文字

- 半角英数字
- 半角ピリオド (.)
- 半角アンダースコア (_)
- 半角ハイフン (-)

指定できる文字列長

1～32 バイト

指定する値の例

00.01

(d) userplugin.java.package

Java パッケージ名を指定します。

使用できる文字

- 半角英数字
- 半角ピリオド (.)
- Java パッケージ名称として有効な文字列
- Windows の予約デバイス名となるディレクトリが作られないような文字列

ただし、「jp.co.hitachi.soft.ucnp」から始まるパッケージ名は指定しないでください。

指定できる文字列長

1 バイト以上

ただし、「Java パッケージ名+プラグイン ID + 入出力アクションコントローラクラス名」または「Java パッケージ名+プラグイン ID + 中断再開アクションコントローラクラス名」の合計が 128 バイト以内になるよう指定してください。

指定する値の例

sample.userplugin

(e) userplugin.server.controller.ioaction

入出力アクションコントローラクラス名を指定します。「3.1.2 ひな形プラグイン作成コマンドの実行」では、このクラス名を基にコントローラクラスが生成されます。

使用できる文字

- 半角英数字
- 半角アンダースコア (_)
- Java クラス名称として有効な文字列
- Windows の予約デバイス名を含まない文字列
- Java パッケージ名と組み合わせた値がシステム内でユニークになる文字列

指定できる文字列長

1 バイト以上

ただし、「Java パッケージ名+プラグイン ID +入出力アクションコントローラクラス名」の合計が 128 バイト以内になるよう指定してください。

指定する値の例

IoPluginController

(f) userplugin.server.controller.ioaction.type

入出力アクションコントローラクラスを識別する文字列を指定します。システム内でユニークになるようにしてください。「3.1.2 ひな形プラグイン作成コマンドの実行」では、この文字列を基に入出力プラグイン用 XML ファイル (ioaction.xml) が生成されます。なお、入出力プラグイン用 XML ファイルとは、業務編集画面に表示される [Plugins] パレットおよびプラグインパーツの表示内容を定義したファイルです。一般には、生成するひな形プラグインの ID と同じ値を指定しますが、異なる値も指定できます。

使用できる文字

- 半角英数字
- 半角ピリオド (.)
- 半角アンダースコア (_)
- 半角ハイフン (-)

指定できる文字列長

1~64 バイト

指定する値の例

userplugin

(g) userplugin.server.controller.suspend

中断再開アクションコントローラクラス名を指定します。

使用できる文字

- 半角英数字
- 半角アンダースコア (_)
- Java クラス名称として有効な文字列
- Windows の予約デバイス名を含まない文字列
- Java パッケージ名と組み合わせた値がシステム内でユニークになる文字列

指定できる文字列長

1 バイト以上

ただし、「Java パッケージ名+プラグイン ID +中断再開アクションコントローラクラス名」の合計が 128 バイト以内になるよう指定してください。

指定する値の例

SuspendActionController

(4) プロパティファイルの記述例

プラグイン情報プロパティファイルの記述例を次に示します。

入出力プラグインの場合

```
userplugin.id = userplugin
userplugin.name = userplugin
userplugin.version = 00.01
userplugin.java.package = sample.userplugin
userplugin.server.controller.ioaction = IoPluginController
userplugin.server.controller.ioaction.type = userplugin
```

中断再開プラグインの場合

```
userplugin.id = ucnp.plugin.suspend
userplugin.name = userplugin
userplugin.version = 00.01
userplugin.java.package = sample.userplugin
userplugin.server.controller.suspend = SuspendActionController
```

3.1.2 ひな形プラグイン作成コマンドの実行

ひな形プラグイン作成コマンドを実行して、ひな形プラグインを作成します。ひな形プラグインは、Eclipse の Java プロジェクトの形式で作成されます。

(1) ひな形プラグイン作成コマンドの形式

ひな形プラグイン作成コマンドの形式を次に示します。

uCosminexus Navigation Developer インストールディレクトリ $\%pluginSDK\bin\%npcreateplg.bat$ Δ プラグイン情報プロパティファイルのパス

ひな形プラグイン作成コマンドを実行すると、*uCosminexus Navigation Developer* インストールディレクトリ $\%pluginSDK\plugin$ の配下に、プラグイン情報プロパティファイルの内容に応じてディレクトリが作成されます。「(2) ひな形プラグイン作成コマンドの実行結果 (入出力プラグインの場合)」または「(3) ひな形プラグイン作成コマンドの実行結果 (中断再開プラグインの場合)」を参照してディレクトリ構成を確認し、下線で示されているファイルだけを編集してください。

なお、太字部分は、プラグイン情報プロパティファイルに指定した値になります。プラグイン情報プロパティファイルの編集方法の詳細は、「3.1.1 プラグイン情報プロパティファイルの編集」を参照してください。複数のプラグインを作成する場合は、プラグインの数だけプラグイン情報プロパティファイルの編集とひな形プラグインの作成を繰り返してください。

(2) ひな形プラグイン作成コマンドの実行結果（入出力プラグインの場合）

入出力プラグインの場合のディレクトリ構成を次に示します。

```

uCosminexus Navigation Developerインストールディレクトリ¥pluginSDK¥plugin
├─ プラグインID
│  ├─ .project
│  ├─ .classpath
│  ├─ build.xml
│  ├─ ucnp sdkversion.properties
│  ├─ js
│  ├─ dd
│  │  └─ META-INF
│  │       └─ application.xml
│  ├─ images
│  │  └─ sample icon.gif
│  └─ WEB-INF
│     ├─ plugin.xml
│     ├─ web.xml
│     └─ portlet.xml
│  └─ lib
│  └─ conf
│     ├─ ucnp_label_ プラグインID.properties
│     ├─ ucnp_message_ プラグインID.properties
│     ├─ ucnp_ プラグインID.properties
│     └─ ioaction.xml
│  └─ src
│     └─ Javaパッケージ名
│        ├─ controller
│        │  └─ 入出力アクションコントローラクラス名.java
│        └─ portlet
│           └─ PluginPortlet.java

```

(3) ひな形プラグイン作成コマンドの実行結果（中断再開プラグインの場合）

中断再開プラグインの場合のディレクトリ構成を次に示します。

```

uCosminexus Navigation Developerインストールディレクトリ¥pluginSDK¥plugin
├─ ucnp.plugin.suspend
│  ├─ .project
│  ├─ .classpath
│  ├─ build.xml
│  ├─ ucnp sdkversion.properties
│  ├─ js
│  ├─ dd
│  │  └─ META-INF
│  │       └─ application.xml
│  └─ WEB-INF
│     ├─ plugin.xml
│     ├─ web.xml
│     └─ portlet.xml
│  └─ lib
│  └─ conf
│     ├─ ucnp_label_ucnp.plugin.suspend.properties
│     ├─ ucnp_message_ucnp.plugin.suspend.properties
│     └─ ucnp_ucnp.plugin.suspend.properties
│  └─ src
│     └─ Javaパッケージ名
│        ├─ controller
│        │  └─ 中断再開アクションコントローラクラス名.java
│        └─ portlet
│           └─ PluginPortlet.java

```

(4) ひな形プラグイン作成コマンド実行時の注意事項

注意事項を次に示します。

- コマンドを複数同時に実行した場合、動作は保証されません。
- *uCosminexus Navigation Developer* インストールディレクトリ¥pluginSDK¥plugin の配下に、プラグイン情報プロパティファイルに指定したプラグイン ID と同じ名称のディレクトリが存在している場合は、上書きするかどうかを確認するメッセージが表示されます。確認メッセージに対しては次のように対処してください。
 - 上書きする場合
「Y」または「y」を指定してください。
 - 上書きしない場合
「N」または「n」を指定してください。処理が中止されます。
この場合は、プラグイン情報プロパティファイルに指定したプラグイン ID を変更してからコマンドを実行し直してください。
- Windows XP 以外でひな形プラグイン作成コマンドを実行する場合は、コマンドプロンプトを [管理者として実行] で開いてから実行してください。条件を満たさない場合、動作は保証されません。

3.2 ひな形プラグインのプロジェクトのインポート

ひな形プラグインのプロジェクトを Eclipse にインポートする手順を次に示します。

参考

この作業は一つのプラグインにつき一度だけ実行してください。作成済みのプラグインを修正する場合は、この作業は不要です。

手順

1. Eclipse を起動します。
2. Eclipse で [ファイル] - [インポート] を選択します。
[インポート] 画面が表示されます。
3. [一般] - [既存プロジェクトをワークスペースへ] を選択します。
4. [次へ] ボタンをクリックします。
5. [ルート・ディレクトリーの選択] テキストボックスに [3.1 ひな形プラグインの作成] で作成したひな形プラグインのディレクトリを指定します。
6. [終了] ボタンをクリックします。
ひな形プラグインのプロジェクトが Eclipse に追加されます。

注意事項

- [プロジェクトをワークスペースにコピー] はチェックしないでください。
- Windows XP 以外で Eclipse を起動する場合、eclipse.exe ファイルを右クリックし、[管理者として実行] を選択して起動してください。条件を満たさない場合、動作は保証されません。
- Eclipse のバージョンによっては、インポート時に「ワークスペースをリフレッシュしています」というエラーメッセージが表示される場合があります。このエラーメッセージが表示された場合、インポートしたプロジェクトをいったん削除して、再度インポートを実施してください。
エラーを防止するためには、インポート前に pluginSDK プロジェクトを閉じるかリフレッシュする必要があります。[プロジェクト・エクスプローラー] に表示される pluginSDK プロジェクトを右クリックして、[プロジェクトを閉じる] または [更新] を選択してください。

3.3 ひな形プラグインのカスタマイズ

ひな形プラグインをカスタマイズするには、Eclipse を使用します。

3.3.1 入出力プラグインのカスタマイズ

ここでは、入出力プラグインのカスタマイズ手順について説明します。

入出力プラグインとして作成したひな形プラグインをカスタマイズする手順を次に示します。

1. [Plugins] パレットおよびプラグインパーツのアイコンの作成
2. 入出力プラグイン用 XML ファイルの編集
3. パラメタ説明用ツールチップの設定
4. プラグイン実行順序の指定
5. 確認ダイアログの表示有無の指定
6. プレビュー画面での実行有無の指定
7. 実行を抑止するボタン種別の指定
8. プラグインが実行する処理の実装

以降では、それぞれの手順について説明します。

(1) [Plugins] パレットおよびプラグインパーツのアイコンの作成

必要に応じて、業務編集画面に表示される [Plugins] パレットおよびプラグインパーツのアイコンを変更します。[Plugins] パレットおよびプラグインパーツのアイコンの位置を次の図に示します。

図 3-1 [Plugins] パレットおよびプラグインパーツのアイコンの位置



ひな形プラグイン作成コマンドを実行すると、uCosminexus Navigation Developer がデフォルトで提供しているアイコンが [Plugins] パレットおよびプラグインパーツに表示されます。

複数のプラグインを開発する場合、プラグインごとに異なるアイコンを作成すると、表示されるアイコンによってプラグインを区別できるようになります。作成したアイコンは次に示すディレクトリに格納してください。

`uCosminexus Navigation Developer` インストールディレクトリ¥pluginSDK¥plugin¥プラグインID¥images

アイコンは次の表に示す形式で作成してください。

表 3-1 アイコンの作成形式

項番	項目	説明
1	ファイル名	「(2) 入出力プラグイン用 XML ファイルの編集」を参照して、ファイル名を指定してください。 なお、uCosminexus Navigation Developer がデフォルトで提供しているアイコンのファイル名は sample_icon.gif です。
2	ファイル形式	Web ブラウザで表示できる形式であればどのような形式でもかまいません。 なお、uCosminexus Navigation Developer がデフォルトで提供しているアイコンのファイル形式は GIF 形式です。
3	サイズ	24×24 ピクセルで作成してください。

注意事項

Windows XP 以外でプラグイン開発を実施する場合、アイコンを OS のインストールドライブ：
 ＊Program Files ディレクトリ以下などに作成する（ファイルを新規作成またはコピーする）ときに、
 Windows の管理者権限がないユーザが作成すると、ファイルがユーザフォルダにリダイレクトされて
 しまうことがあります。そのため、Windows の管理者権限があるユーザで追加してください。

(2) 入出力プラグイン用 XML ファイルの編集

入出力プラグイン用 XML ファイル (ioaction.xml) は、ひな形プラグインの作成時に生成されます。入出力プラグイン用 XML ファイルを編集して、次の内容を変更してください。

- 入出力プラグインの実行順序
- 入出力パラメタ定義
- 業務編集画面に表示される [Plugins] パレット、プラグインパーツ、およびツールチップの表示内容

入出力プラグイン用 XML ファイルと業務編集画面の表示内容との対応を次の図に示します。

図 3-2 入出力プラグイン用 XML ファイルと業務編集画面の表示内容との対応

■入出力プラグイン用XMLファイルの内容

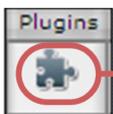
```

<?xml version="1.0" encoding="UTF-8"?>
<ioaction xmlns="http://model.xml.ioaction.navi.plugin.ucnp.soft.hitachi.co.jp"
id="sample.ioAction" name="IoAction" version="2.0">
  <iopart id="io_part_1">
    <iotype name="sample.ioAction" />
    <iconURL url="/IoAction/images/sample_icon.gif"/> (1)
    <priority value="700" />
    <execConfirm value="false" />
    <execPreview value="true" />
    <disableButtonType value="show_next_page" />
    <parameters type="title">
      <param name="IoAction" description="description of IoAction"/>
    </parameters>
    <parameters type="input"> (2)
      <param name="inputItemName1" description="description of inputItemName1"/>
      <param name="inputItemName2" description="description of inputItemName2"/>
      <param name="inputItemName3" description="description of inputItemName3"/>
    </parameters>
    <parameters type="output"> (3)
      <param name="outputItemName1" description="description of outputItemName1"/>
      <param name="outputItemName2" description="description of outputItemName2"/>
      <param name="outputItemName3" description="description of outputItemName3"/>
    </parameters>
  </iopart>
</ioaction>

```

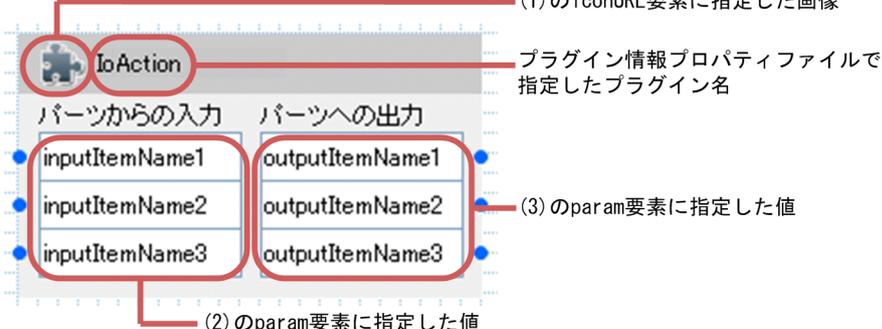
■業務編集画面の表示内容

- [Plugins] パレットの表示内容



(1)のiconURL要素に指定した画像

- プラグインパーツの表示内容



(1)のiconURL要素に指定した画像

プラグイン情報プロパティファイルで指定したプラグイン名

(3)のparam要素に指定した値

(2)のparam要素に指定した値

入出力プラグイン用 XML ファイルの編集方法を説明します。入出力プラグイン用 XML ファイル (ioaction.xml) の格納ディレクトリは、次のとおりです。

*uCosminexus Navigation Developer*インストールディレクトリ¥pluginSDK¥plugin¥プラグインID¥WEB-INF¥conf

なお、Windows XP 以外で入出力プラグイン用 XML ファイルを編集する場合、エディタを [管理者として実行] で起動して、ファイルを編集してください。

入出力プラグイン用 XML ファイル (ioaction.xml) の編集例を次に示します。太字の部分編集します。

```

<?xml version="1.0" encoding="UTF-8"?>
<ioaction xmlns="http://model.xml.ioaction.navi.plugin.ucnp.soft.hitachi.co.jp"
id="sample.ioAction" name="IoAction" version="2.0">

```

```

<iopart id="io_part_1">
  <iotype name="sample.IoAction" />
  <iconURL url="/IoAction/images/sample_icon.gif"/>
  <priority value="700" />
  <execConfirm value="false" />
  <execPreview value="true" />
  <disableButtonType value="show_next_page" />
  <parameters type="title">
    <param name="IoAction" description="description of IoAction"/>
  </parameters>
  <parameters type="input">
    <param name="inputItemName1" description="description of inputItemName1"/>
    <param name="inputItemName2" description="description of inputItemName2"/>
    <param name="inputItemName3" description="description of inputItemName3"/>
  </parameters>
  <parameters type="output">
    <param name="outputItemName1" description="description of outputItemName1"/>
    <param name="outputItemName2" description="description of outputItemName2"/>
    <param name="outputItemName3" description="description of outputItemName3"/>
  </parameters>
</iopart>
</ioaction>

```

入出力プラグイン用 XML ファイル (ioaction.xml) の編集方法を次の表に示します。次の表に示した箇所だけを修正してください。

表 3-2 入出力プラグイン用 XML ファイル (ioaction.xml) の編集方法

項番	編集する箇所	説明	属性値に指定できる文字列と意味	属性値に指定できる文字列長
1	<iconURL>タグの url 属性 (ファイル名部分)	業務編集画面に表示される [Plugins] パレットおよびプラグインパーツのアイコンのファイル名を指定します。 [(1) [Plugins] パレットおよびプラグインパーツのアイコンの作成] で作成したアイコンのファイル名をデフォルトから変更する場合だけ、url 属性のファイル名部分を変更します。	<ul style="list-style-type: none"> 半角英数字 半角アンダースコア (_) 半角ピリオド (.) 半角ハイフン (-) 	最大 1,024 バイト ([/ IoAction/images /] 部分を含む)
2	<priority>タグの value 属性	入出力プラグインの実行順序を指定します。 デフォルト値は 700 です。700 はシステムプラグインとユーザープラグインの実行順序の中間の値です。 priority 要素を省略した場合、デフォルト値が仮定されます。	次に示す半角英数字の値 <ul style="list-style-type: none"> 100~500 600~900 	—
3	<execConfirm>タグの value 属性	必要に応じて、入出力プラグインの実行時に確認ダイアログを表示するかどうかを指定します。 処理に時間が掛かることをユーザーに通知したい場合などに指定してください。 execConfirm 要素を省略した場合、false が仮定されます。	<ul style="list-style-type: none"> true 表示する false 表示しない 	—

3 プラグインの開発

項番	編集する箇所	説明	属性値に指定できる文字列と意味	属性値に指定できる文字列長
4	<execPreview>タグの value 属性	必要に応じて、プレビュー画面で入出力プラグインを実行するかどうかを指定します。*1 execPreview 要素を省略した場合、true が仮定されます。	<ul style="list-style-type: none"> • true 実行する • false 実行しない 	—
5	<disableButtonType>タグの value 属性	必要に応じて、ボタンの種別によって入出力プラグインの実行を抑止するかどうかを指定します。*2 例えば、show_next_page を指定すると、[次へ] ボタンや直接遷移で次のノードに遷移するタイミングで、入出力プラグインの実行が抑止されます。 複数指定する場合、半角コンマ区切りで指定します。 disableButtonType 要素を省略した場合、空文字が仮定されます。	<ul style="list-style-type: none"> • start <ul style="list-style-type: none"> ・ターミナルノード(開始) からプロセスノードに遷移 ・ターミナルノード(開始) に接続されていない先頭のプロセスノードを表示 • show_next_page <ul style="list-style-type: none"> ・[次へ] ボタン, [工程名へ] ボタン, または直接遷移で次のノードに遷移 ・遷移先のノードが存在しているプロセスノード, または [戻る] ボタンが表示になっているプロセスノードに遷移 • show_previous_page <ul style="list-style-type: none"> ・[戻る] ボタン ・前のノードに直接遷移 • show_complete_page <ul style="list-style-type: none"> ・[完了] ボタン ・遷移先ノードが存在しない, かつ [戻る] ボタンが非表示になっているプロセスノードに遷移 	—
6	<param>タグの description 属性	必要に応じて、入出力プラグインの説明文 (ツールチップ) を指定します。	任意の文字列*3	0~64 文字
7	<parameters type="input">の下位の<param>タグの name 属性	入力パラメタ名を指定します。 <parameters type="input">タグ内でユニークな値を指定してください。 <param>タグは 0~100 個指定できます。入力パラメタの数だけ指定してください。	<ul style="list-style-type: none"> • 半角英数字 • 半角アンダースコア (_) • 半角ピリオド (.) • 半角スペース • 半角ハイフン (-) 	1~64 バイト*4

項番	編集する箇所	説明	属性値に指定できる文字列と意味	属性値に指定できる文字列長
8	<parameters type="input">の下位の<param>タグの description 属性	必要に応じて、入力パラメタの説明文（ツールチップ）を指定します。	任意の文字列※3	0～64 文字
9	<parameters type="output">の下位の<param>タグの name 属性	出力パラメタ名を指定します。 <parameters type="output">タグ内でユニークな値を指定してください。 <param>タグは0～100個指定できます。出力パラメタの数だけ指定してください。	<ul style="list-style-type: none"> 半角英数字 半角アンダースコア (_) 半角ピリオド (.) 半角スペース 半角ハイフン (-) 	1～64 バイト※4
10	<parameters type="output">の下位の<param>タグの description 属性	必要に応じて、出力パラメタの説明文（ツールチップ）を指定します。	任意の文字列※3	0～64 文字

(凡例)

－：該当しない。

注※1

プレビュー画面で入出力プラグインを実行するかどうかは、サーバ処理実装インタフェース (IioPluginController) の inputFromNode メソッド、および outputToNode メソッドの ucnp.screen.ispreview キーの値を使って実装することもできます。ただし、ucnp.screen.ispreview キーによる設定は、サーバ処理実装インタフェース (IioPluginController) で特殊な処理を設定する場合に利用します。そのため、処理をスキップするだけの場合は入出力プラグイン用 XML ファイル (ioaction.xml) で指定してください。サーバ処理実装インタフェース (IioPluginController) の詳細は、「5.3 IioPluginController (サーバ処理実装インタフェース)」を参照してください。

注※2

ボタンの種別によって入出力プラグインの実行を抑止するかどうかは、サーバ処理実装インタフェース (IioPluginController) の inputFromNode メソッド、および outputToNode メソッドの ucnp.button.type キーの値を使って実装することもできます。ただし、ucnp.button.type キーによる設定は、サーバ処理実装インタフェース (IioPluginController) で特殊な処理を設定する場合に利用します。そのため、処理をスキップするだけの場合は入出力プラグイン用 XML ファイル (ioaction.xml) で指定してください。サーバ処理実装インタフェース (IioPluginController) の詳細は、「5.3 IioPluginController (サーバ処理実装インタフェース)」を参照してください。

注※3

description に指定した文字列が、そのままツールチップを表示させるときの HTML 属性として使用されます。このためツールチップに表示する文字列を改行させたい場合は「
」を指定してください。「
」以外の制御文字を指定した場合の表示は保証されません。

注※4

長い文字列を指定した場合、プラグインパーツ上では後ろの方の文字が切れて表示されることがあります。入出力アクションコントローラクラス上では、ここで指定した文字列がそのまま使用されます。

参考

<parameters type="input">および<parameters type="output">の下位の<param>タグの name 属性に指定した値は、サーバ処理実装インタフェース (IioPluginController) の inputFromNode メソッド、および outputToNode メソッドの param パラメタと戻り値に使用されます。

name 属性に指定した値が使用される箇所を次に示します。

param パラメタ

- inputFromNode メソッドの、名前 `ucnp.current.params.map` に対応する値の Map オブジェクトの名前
- outputToNode メソッドの、名前 `ucnp.next.params.map` に対応する値の Map オブジェクトの名前

戻り値

- inputFromNode メソッドの、名前 `ucnp.current.params.map` に対応する値の Map オブジェクトの名前
- outputToNode メソッドの、名前 `ucnp.next.params.map` に対応する値の Map オブジェクトの名前

name 属性に指定した値とサーバ処理実装インタフェース (IIOPluginController) との対応を次の図に示します。

図 3-3 name 属性に指定した値とサーバ処理実装インタフェース (IloPluginController) との対応

■ 入出力プラグイン用XMLファイルの内容

```

<?xml version="1.0" encoding="UTF-8"?>
<ioaction xmlns="http://model.xml.ioaction.navi.plugin.ucnp.soft.hitachi.co.jp"
id="sample.IoAction" name="IoAction" version="2.0">
  <iopart id="io_part_1">
    <iotype name="sample.IoAction" />
    <iconURL url="/IoAction/images/sample_icon.gif"/>
    <priority value="700" />
    <execConfirm value="false" />
    <execPreview value="true" />
    <disableButtonType value="show_next_page" />
    <parameters type="title">
      <param name="IoAction" description="description of IoAction"/>
    </parameters>
    <parameters type="input">
      <param name="inputItemName1" description="description of inputItemName1"/>
      <param name="inputItemName2" description="description of inputItemName2"/>
      <param name="inputItemName3" description="description of inputItemName3"/>
    </parameters>
    <parameters type="output">
      <param name="outputItemName1" description="description of outputItemName1"/>
      <param name="outputItemName2" description="description of outputItemName2"/>
      <param name="outputItemName3" description="description of outputItemName3"/>
    </parameters>
  </iopart>
</ioaction>

```

■ サーバ処理実装インタフェース (IloPluginController) の内容 (一部抜粋)

```

/* クライアントから受信したデータをMapに展開します */
Map<?, ?> inParamMap = (Map<?, ?>) param.get("ucnp.current.params.map");

/* 受信した値をMapより取り出します */
String param1 = (String) inParamMap.get("inputItemName1");
String param2 = (String) inParamMap.get("inputItemName2");
String param3 = (String) inParamMap.get("inputItemName3");

/* クライアントに送信する値を設定します */
String outParam1 = param1;
String outParam2 = param2;
String outParam3 = param3;

Map<String, String> outParamMap = new HashMap<String, String>();

/* Mapに送信するデータを設定します */
outParamMap.put("outputItemName1", outParam1);
outParamMap.put("outputItemName2", outParam2);
outParamMap.put("outputItemName3", outParam3);

```

サーバ処理実装インタフェース (IloPluginController) の詳細は、「5.3 IloPluginController (サーバ処理実装インタフェース)」を参照してください。

(3) パラメタ説明用ツールチップの設定

入出力プラグイン用 XML ファイル (ioaction.xml) にツールチップの説明文を設定すると、業務編集画面でプラグインパーツをポイントしたときに説明が表示されます。設定できるツールチップを次に示します。

- 入出力プラグインの説明文
プラグインパーツのタイトルに表示されるツールチップです。
- 入力パラメタの説明文
プラグインパーツの入力パラメタに表示されるツールチップです。

- 出力パラメタの説明文
プラグインパーツの出力パラメタに表示されるツールチップです。

設定する方法および値の詳細は「(2) 入出力プラグイン用 XML ファイルの編集」を参照してください。

(4) プラグイン実行順序の指定

入出力プラグイン用 XML ファイル (ioaction.xml) でプラグインの実行順序を変更できます。実行順序に指定できる値を次に示します。

- 100~500
- 600~900

同じノードに複数の入出力プラグインを配置した場合、実行順序に指定した値の昇順で入出力プラグインが実行されます。また、複数の入出力プラグインに同じ実行順序が指定されている場合、実行順序は不定となります。

値の詳細は「(2) 入出力プラグイン用 XML ファイルの編集」を参照してください。

(5) 確認ダイアログの表示有無の指定

入出力プラグイン用 XML ファイル (ioaction.xml) で入出力プラグインの実行時に確認ダイアログを表示するかどうかを指定できます。指定できる値を次に示します。

- true
- false

ノードに配置しているプラグインに、「true」を指定しているプラグインが含まれている場合は、inputFromNode メソッドの実行時に入出力プラグインの確認ダイアログが表示されます。なお、設定値が「true」のプラグインが複数配置されていても、確認ダイアログは 1 回だけ表示されます。確認ダイアログで [キャンセル] を選択した場合は、KDCZ00266-Q メッセージが出力され、ノードに設定しているすべてのプラグインの処理が実行されません。

値の詳細は「(2) 入出力プラグイン用 XML ファイルの編集」を参照してください。

(6) プレビュー画面での実行有無の指定

プレビュー画面で入出力プラグインを実行するかどうかを、入出力プラグイン用 XML ファイル (ioaction.xml) で指定できます。指定できる値を次に示します。

- true
- false

ノードに複数の入出力プラグインを配置している場合、「true」を指定している入出力プラグインだけが実行されます。

値の詳細は「(2) 入出力プラグイン用 XML ファイルの編集」を参照してください。

(7) 実行を抑止するボタン種別の指定

入出力プラグインの実行を抑止するボタン種別を、入出力プラグイン用 XML ファイル (ioaction.xml) で指定できます。指定できる値を次に示します。

- start

- show_next_page
- show_previous_page
- show_complete_page

ノードに複数の入出力プラグインを配置している場合、実行を抑止する設定になっている入出力プラグインは実行されません。

値の詳細は「(2) 入出力プラグイン用 XML ファイルの編集」を参照してください。

(8) プラグインが実行する処理の実装

次に示すファイルを変更して、プラグインが実行する処理を実装します。

```
uCosminexus Navigation Developerインストールディレクトリ¥pluginSDK¥plugin¥プラグインID¥WEB-INF¥src¥Javaパッケージ階層¥controller¥入出力アクションコントローラクラス.java
```

サーバ処理実装インタフェース (IIoPluginController) の詳細は、「5.3 IIoPluginController (サーバ処理実装インタフェース)」を参照してください。

3.3.2 中断再開プラグインのカスタマイズ

ここでは、中断再開プラグインのカスタマイズ手順について説明します。

中断再開プラグインとして作成したひな形プラグインをカスタマイズする手順を次に示します。

1. ユーザプロパティファイルの設定
2. プラグインが実行する処理の実装

以降では、それぞれの手順について説明します。

(1) ユーザプロパティファイルの設定

次のプロパティについて検討し、ユーザプロパティファイルに値を設定してください。

- ucnp.base.client.suspend.enable (中断再開プラグインの使用有無)
- ucnp.base.client.viewer.confirm.discardinput.enable (入力情報の破棄確認ダイアログの表示有無)
- ucnp.base.client.suspend.confirm.load.enable (業務再開の確認ダイアログの表示有無)

中断再開プラグインを使用するには、プロパティ ucnp.base.client.suspend.enable の設定値を true (有効) にする必要があります。

ユーザプロパティファイルの設定値および設定方法については、マニュアル「Cosminexus V9 ナビゲーション プラットフォーム 導入から運用まで」を参照してください。

(2) プラグインが実行する処理の実装

次に示すファイルを変更して、プラグインが実行する処理を実装します。

```
uCosminexus Navigation Developer インストールディレクトリ¥pluginSDK¥plugin¥ucnp.plugin.suspend¥WEB-INF¥src¥Java パッケージ階層¥controller¥中断再開アクションコントローラクラス.java
```

ひな形プラグインを作成するときに生成された中断再開アクションコントローラインタフェースに、ユーザ操作に対応して各メソッドに処理を追加します。中断再開アクションコントローラインタフェースのメ

3 プラグインの開発

ソッドの詳細は、「5.5 ISuspendActionController (中断再開アクションコントローラインタフェース)」を参照してください。

中断再開アクションコントローラインタフェースの実装例については、次のサンプルファイルを参照してください。

ナビゲーション プラットフォームインストールディレクトリ¥sample¥plugin¥ucnp.plugin.suspend¥WEB-INF¥src¥jp¥co¥hitachi¥soft¥ucnp¥plugin¥sample¥suspend¥controller¥SuspendActionController.java

3.4 データベースとの接続処理の追加

ナビゲーション プラットフォームと通信する任意の処理は、ひな形プラグインをカスタマイズするときに追加します。プラグインにデータベースとの接続処理を追加するために、DB Connector を Cosminexus のリソースアダプタとしてデプロイしてください。

プラグインからデータベースを利用するための設定方法を次に示します。

1. Cosminexus のリソースアダプタの設定
2. web.xml へのリソース定義の追加
3. cosminexus.xml の編集
4. データベースとの接続処理の実装

この節では、それぞれの手順について説明します。

3.4.1 Cosminexus のリソースアダプタの設定

リソースアダプタの設定、デプロイ、および開始を実行します。詳細は、次に示すマニュアルでデータベースを使用するための設定方法について記載されている個所の、リソースアダプタに関する説明を参照してください。

- 「Cosminexus V9 アプリケーションサーバ システム構築・運用ガイド」
- 「Cosminexus V9 アプリケーションサーバ アプリケーション設定操作ガイド」
- 「Cosminexus V9 ナビゲーション プラットフォーム 導入から運用まで」

3.4.2 web.xml へのリソース定義の追加

uCosminexus Navigation Developer の web.xml にリソース定義を追加します。web.xml の格納先を次に示します。

uCosminexus Navigation Developer のユーザーセットアッププロパティファイル
(*ucnp_setup_user.properties*) の *ucnp.setup.server.ucnp.portalproject.path* プロパティの設定値
¥WEB-INF¥web.xml

リソース参照の名称を「jdbc/HiRDB_Type4」とした場合の記述例を次に示します。

```
<resource-ref>
  <res-ref-name>jdbc/HiRDB_Type4</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
```

この記述を web.xml ファイルの末尾に記述されている「</web-app>」の直前に追加してください。

注意事項

Windows XP 以外で web.xml を編集する場合、エディタを [管理者として実行] で起動して、ファイルを編集してください。

3.4.3 cosminexus.xml の編集

uCosminexus Navigation Developer の cosminexus.xml を編集します。

cosminexus.xml の格納先を次に示します。

3 プラグインの開発

*uCosminexus Navigation Developer*インストールディレクトリ¥pluginSDK¥base¥dd¥META-INF¥cosminexus.xml

リソース参照の名称を「jdbc/HiRDB_Type4」、リソースアダプタ表示名を「DB_Connector_for_HiRDB_Type4」とした場合の記述例を次に示します。

```
<resource-ref>
  <res-ref-name>jdbc/HiRDB_Type4</res-ref-name>
  <linked-to>DB_Connector_for_HiRDB_Type4</linked-to>
</resource-ref>
```

この記述を cosminexus.xml ファイルの末尾に記述されている「</war>」の直前に追加してください。

注意事項

Windows XP 以外で cosminexus.xml を編集する場合、エディタを [管理者として実行] で起動して、ファイルを編集してください。

3.4.4 データベースとの接続処理の実装

マニュアル「Cosminexus V9 アプリケーションサーバ 機能解説 互換編」のデータベースとの接続について記載されている個所を参照して、プラグインにデータベースとの接続処理を実装します。

3.5 アプリケーション属性ファイル (application.xml) の編集

アプリケーション属性ファイル (application.xml) を編集して、プラグインを有効にします。なお、この作業は一つのプラグインにつき一度だけ実行してください。作成済みのプラグインを修正する場合は、この作業は不要です。

アプリケーション属性ファイルの格納先を次に示します。

```
uCosminexus Navigation Developerインストールディレクトリ¥pluginSDK¥base¥dd¥META-INF¥application.xml
```

アプリケーション属性ファイルの<application>と</application>の間に、次に示すファイルの module 要素の内容をコピー&ペーストしてください。

```
uCosminexus Navigation Developerインストールディレクトリ¥pluginSDK¥plugin¥プラグインID¥dd¥META-INF¥application.xml
```

コピー&ペーストする内容の例を次に示します。

```
<module>
<web>
<web-uri>プラグイン名.war</web-uri>
<context-root>/プラグイン名</context-root>
</web>
</module>
```

「プラグイン名」は、プラグイン情報プロパティファイルに指定したプラグイン名となります。プラグイン情報プロパティファイルの編集方法の詳細は、「3.1.1 プラグイン情報プロパティファイルの編集」を参照してください。

また、プラグインを複数開発する場合は、開発するプラグインの数だけ module 要素を追加してください。

注意事項

Windows XP 以外でアプリケーション属性ファイルを編集する場合、エディタを [管理者として実行] で起動して、ファイルを編集してください。

3.6 プラグインのビルド

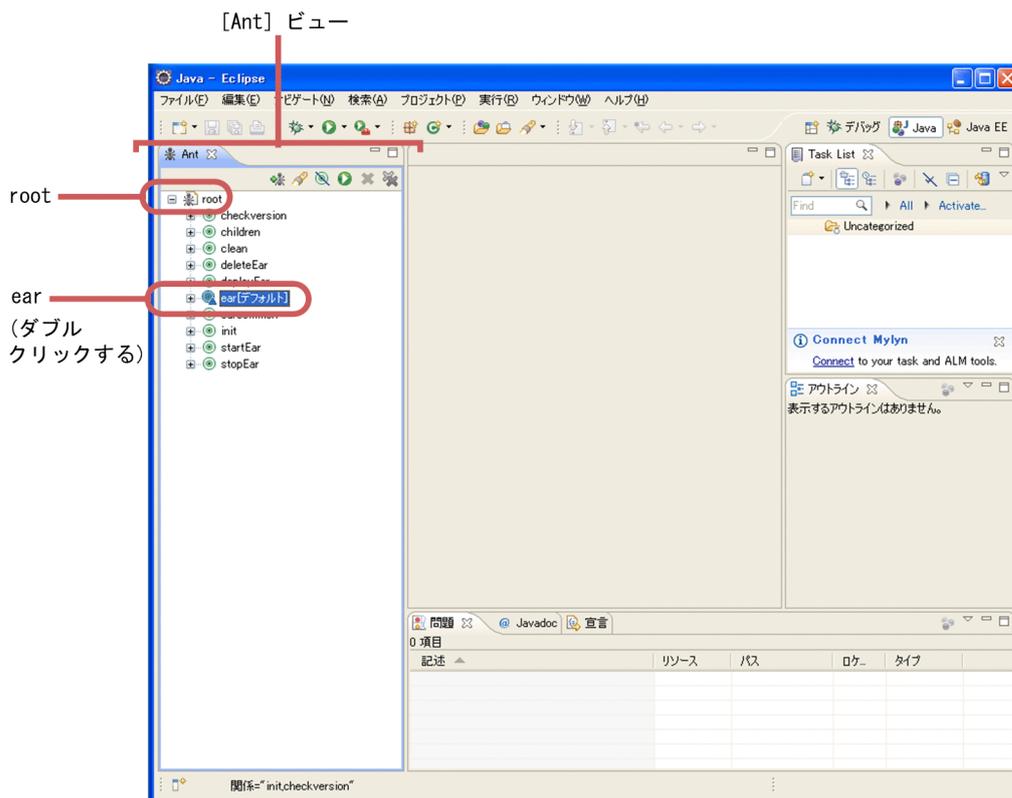
この節では、プラグインのビルド手順、およびビルドに失敗したときの対処の手順を説明します。

3.6.1 プラグインのビルド手順

Ant から pluginSDK プロジェクトの build.xml を実行して、カスタマイズしたひな形プラグインをビルドします。プラグインのビルド手順を次に示します。

手順

1. Eclipse を起動します。
2. Eclipse で [ウィンドウ] - [ビューの表示] - [その他] - [Ant] - [Ant] を選択して、[Ant] ビューを表示します。
3. [2.3 pluginSDK プロジェクトのインポート] でインポートしたプロジェクトから、build.xml を [Ant] ビューにドラッグ&ドロップします。
[Ant] ビューに「root」が追加されます。
4. [Ant] ビューの「root」の [+] アイコンをクリックして、ターゲット一覧を表示します。



5. ターゲット一覧の [ear] をダブルクリックします。

プラグインのビルドが実行され、EAR ファイルが次のディレクトリに作成されます。

`uCosminexus Navigation Developer`インストールディレクトリ \backslash pluginSDK \backslash dest

なお、2 度目以降にプラグインをビルドする場合は、ターゲット一覧の [ear] をダブルクリックする前に、必要に応じて [clean] ターゲットをダブルクリックしてください。この操作で、クラス、ライブラリ、プラグインなどの中間ファイルを削除できます。

3.6.2 プラグインのビルドに失敗したときの対処

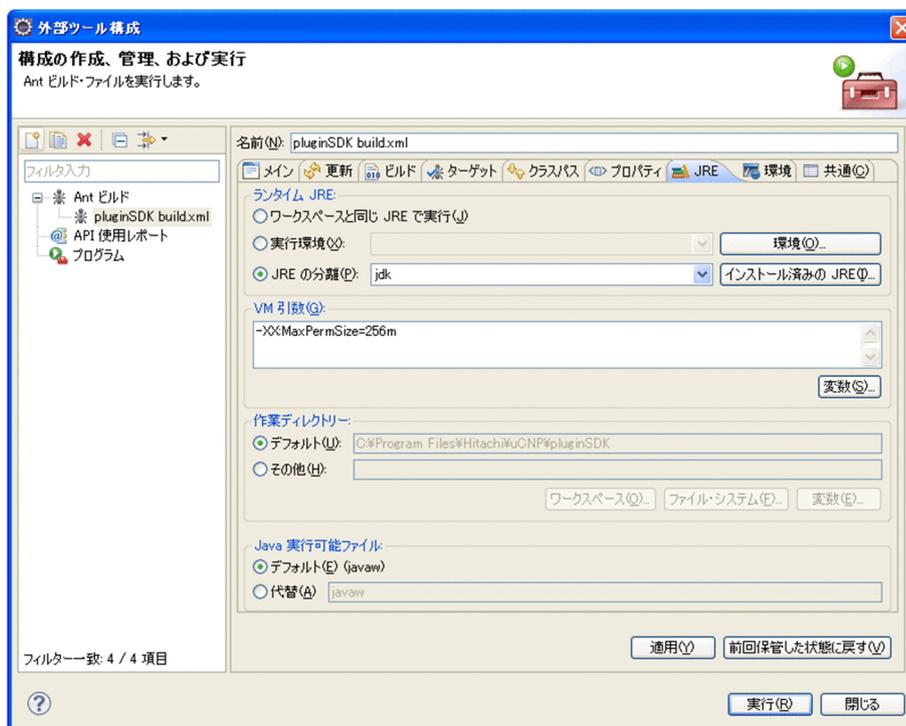
ここでは、プラグインのビルドに失敗したときの対処手順を説明します。

プラグインのビルドに失敗する要因には、`java.lang.OutOfMemoryError` のエラーが考えられます。このエラーは、プラグインが複数ある場合に発生しやすくなります。

`java.lang.OutOfMemoryError` のエラーが発生した場合は、次の手順を実施してから、再度プラグインをビルドしてください。

手順

1. [Ant] ビューの [root] を右クリックして表示されるメニューから [実行] - [外部ツールの構成] を選択します。
[外部ツール構成] ダイアログが表示されます。
2. [JRE] タブをクリックして、[VM 引数] に [-XX:MaxPermSize=256m] を入力します。



3. [適用] ボタンをクリックし、[閉じる] ボタンをクリックします。
[外部ツール構成] 画面が閉じます。

3.7 Web サーバリダイレクトの設定

Web サーバリダイレクトの設定には、Cosminexus Management Server を使用します。Cosminexus Management Server の使用方法の詳細は、マニュアル「Cosminexus V9 アプリケーションサーバ 運用管理ポータル操作ガイド」を参照してください。

注意事項

- Web サーバリダイレクトを正しく設定しないと、業務編集画面の [Plugins] パレット、およびプラグインパーツのアイコンが表示されなくなります。
- この設定は一つのプラグインにつき一度だけ実行してください。

なお、中断再開プラグインを開発する場合、および作成済みのプラグインを修正する場合は、Web サーバリダイレクトの設定は不要です。

Web サーバリダイレクトの設定手順を次に示します。

手順

1. 次の URL にアクセスして、Cosminexus Management Server を起動します。

http://ホスト名:ポート番号/mngsvr/

- **ホスト名**
開発環境の J2EE サーバマシンのホスト名または IP アドレスです。
- **ポート番号**
Cosminexus Management Server が使用するポート番号です。マニュアル「Cosminexus V9 ナビゲーション プラットフォーム 導入から運用まで」の手順どおりにセットアップした環境では、「28080」です。

2. Cosminexus Management Server の運用管理ポータルにログインします。

運用管理ポータルの画面が表示されます。

なお、マニュアル「Cosminexus V9 ナビゲーション プラットフォーム 導入から運用まで」の手順どおりにセットアップした環境では、ログイン時に入力する管理ユーザ ID が「admin」、パスワードが「ucnppass」です。

3. [論理サーバの環境設定] アンカーをクリックします。

論理サーバの環境設定画面が表示されます。

4. [サーバビュー] タブ - [論理 Web サーバ] - [Web サーバ] - [*uCosminexus Navigation Developer* の Web サーバ名称] をクリックし、[マッピング] タブをクリックします。

[マッピングの定義] 画面が表示されます。

5. 次のとおりに設定します。

- いちばん下の行の [有効] のチェックボックスをチェックする
- [URL パターン] に「/*プラグイン名*/*」と入力する
- [論理サーバ名] に *uCosminexus Navigation Developer* 用の論理サーバ名を入力する

6. [追加] ボタンをクリックします。

複数のプラグインを利用する場合は、手順 4.~5.を複数回実行してください。

7. 画面右上の [設定情報の配布] アンカーをクリックします。

[設定情報の配布] 画面が表示されます。

8. 論理サーバ名に uCosminexus Navigation Developer の Web サーバ名称が表示されている行の [配布] チェックボックスをチェックします。
論理サーバが選択されます。
9. [選択して配布] ボタンをクリックします。
手順 8. で選択した論理サーバの設定情報が, uCosminexus Navigation Developer の Web サーバに配布されます。
10. uCosminexus Navigation Developer の Web サーバを再起動します。

3.8 uCosminexus Navigation Developer の J2EE アプリケーションのインポートと開始

uCosminexus Navigation Developer の J2EE アプリケーションのインポート手順を次に示します。

手順

1. J2EE サーバを起動します。

J2EE サーバの起動方法については、マニュアル「Cosminexus V9 ナビゲーション プラットフォーム 導入から運用まで」を参照してください。

2. 中断再開プラグインを新規作成または削除した場合、ユーザプロパティファイルの `ucnp.base.client.suspend.enable` プロパティの設定を変更します。

ユーザプロパティファイルの設定方法については、マニュアル「Cosminexus V9 ナビゲーション プラットフォーム 導入から運用まで」を参照してください。

3. Eclipse を起動します。

注意事項

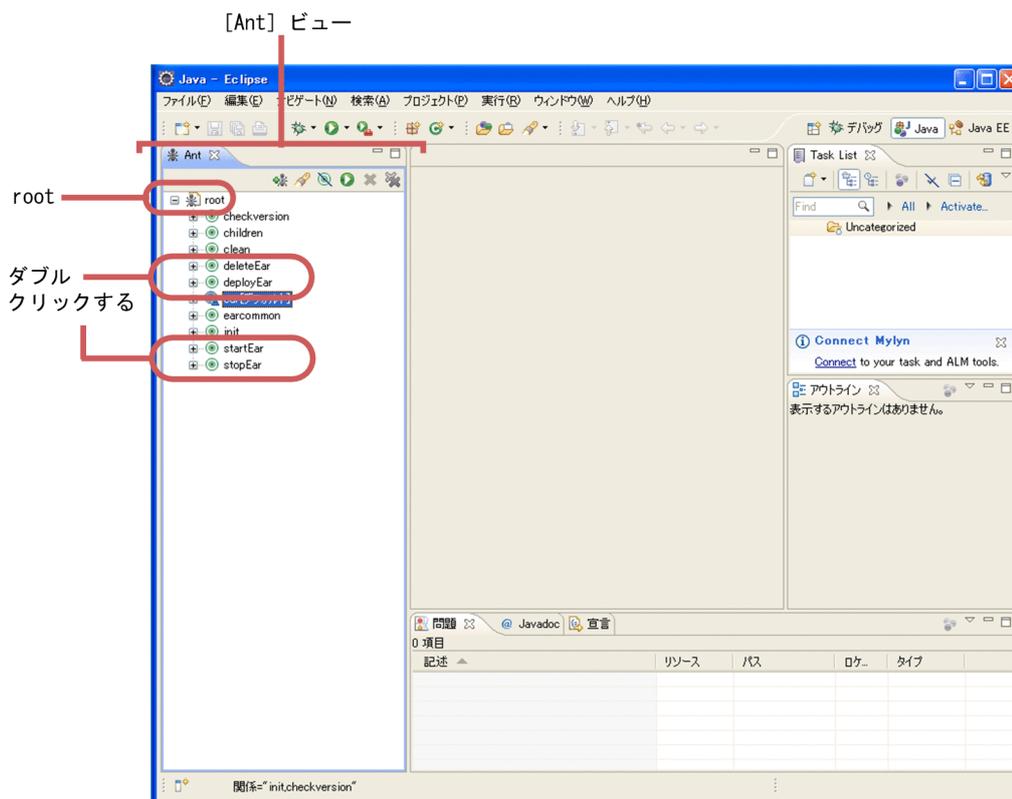
Windows XP 以外で Eclipse を起動する場合、`eclipse.exe` ファイルを右クリックし、[管理者として実行] を選択して起動してください。管理者として Eclipse を起動しない場合、動作は保証されません。

4. Eclipse で [ウィンドウ] - [ビューの表示] - [その他] - [Ant] - [Ant] を選択して、[Ant] ビューを表示します。

5. [2.3 pluginSDK プロジェクトのインポート] でインポートしたプロジェクトから、`build.xml` を [Ant] ビューにドラッグ&ドロップします。

[Ant] ビューに [root] が追加されます。

6. [Ant] ビューの [root] の [+] アイコンをクリックして、ターゲット一覧を表示します。



7. ターゲット一覧の [stopEar] をダブルクリックします。
J2EE アプリケーションが停止します。
8. ターゲット一覧の [deleteEar] をダブルクリックします。
J2EE アプリケーションが削除されます。
9. ターゲット一覧の [deployEar] をダブルクリックします。
J2EE アプリケーションがインポートされます。
10. ターゲット一覧の [startEar] をダブルクリックします。
J2EE アプリケーションが開始されます。

3.9 業務コンテンツへの関連づけ (入出力プラグインの場合)

業務編集画面で、プラグインを利用する業務コンテンツに入出力プラグインを関連づけます。中断再開プラグインの場合は、この操作は不要です。入出力プラグインを関連づけるガイドパーツにはポート (●) が表示されます。

この節では、業務編集画面で入出力プラグインとガイドパーツをマッピング線で関連づける方法について説明します。

3.9.1 マッピング線を引く (ガイドパーツとプラグインパーツをつなぐ)

プラグインパーツを配置してマッピング線を引く手順を次に示します。なお、ガイド領域がマッピングモードのときだけ、プラグインパーツおよびマッピング線を表示、操作できます。各手順の詳細については、マニュアル「Cosminexus V9 ナビゲーション プラットフォーム コンテンツ作成ガイド」を参照してください。

手順

1. Web ブラウザ上で次の URL を指定して、ログイン画面を呼び出します。

```
http://ホスト名:ポート番号/ucnpBase/portal/screen/TitlePortlet/portlet/ucnp/  
pane/-440b4408440g55S76Z2i/layout_id/default/tab_id/-440b4408440g55S76Z2i?  
start_editor=true&open_editor=true
```

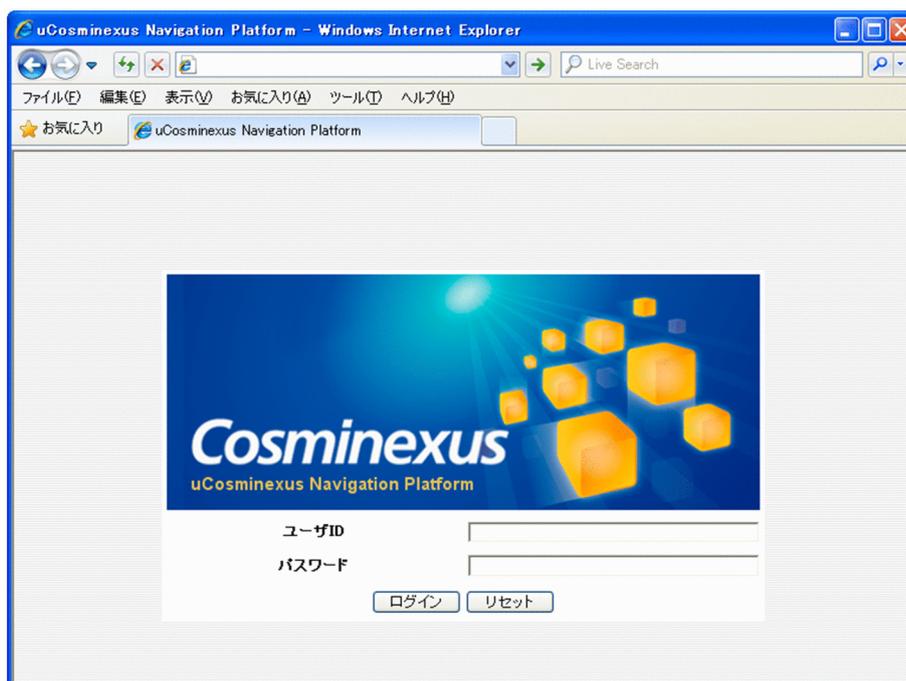
- ホスト名

開発環境の J2EE サーバマシンのホスト名または IP アドレスです。

- ポート番号

開発環境の Web サーバのポート番号です。ユーザセットアッププロパティファイル (ucnp_setup_user.properties) の ucnp.setup.server.cosminexus.hws.http.port プロパティの指定値になります。

2. ユーザ ID とパスワードを入力して、[ログイン] ボタンをクリックします。



入力を間違えた場合は、[リセット] ボタンをクリックして、入力した内容をいったん消去してから再度入力してください。

[ログイン] ボタンをクリックすると、業務編集画面が表示されます。

3. 業務フロー領域で、プラグインを設定したいプロセスノードをクリックします。

ガイド領域に業務フローに対応するガイドが表示されます。

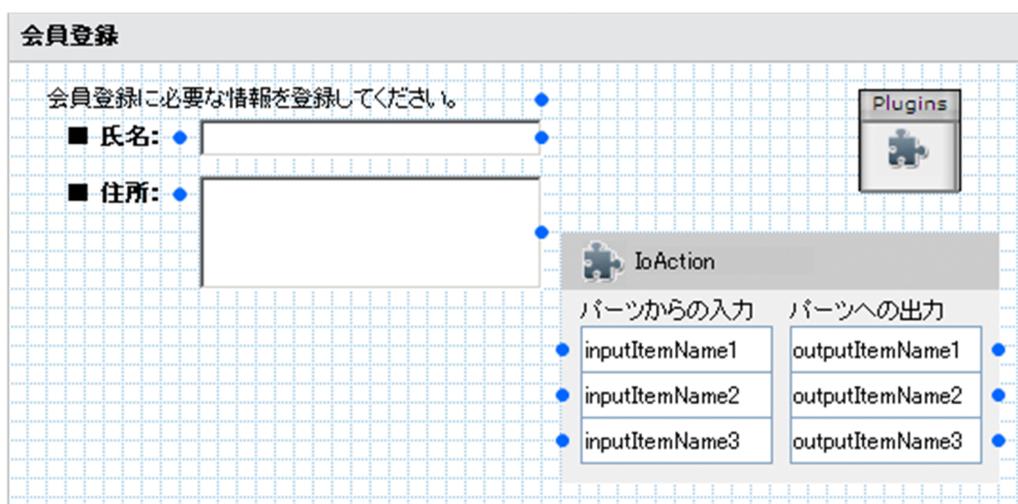
4. ツールバーの  プラグイン ([プラグイン] ボタン) をクリックします。

マッピングモードに切り替わります。

5. ガイド領域の [Plugins] パレットで  ([プラグイン] ボタン) をクリックします。

6. ガイド領域の任意の場所をクリックします。

プラグインパーツが配置されます。



ほかのプラグインを配置したい場合は、手順 5.~6.を繰り返します。

7. 遷移元のガイドパーツのポート（●）からプラグインパーツの入力パラメタ（プラグインパーツの左側のポート）にドラッグしてマッピング線を引きます。

ガイドパーツのキャッシュ値が入力パラメタとしてプラグインに渡されるようになります。

各ガイドパーツのキャッシュ値については、「5.2.1(5) プラグインで保持されるキャッシュ値」を参照してください。

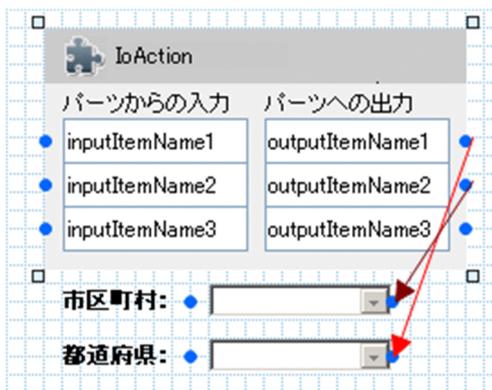
8. プラグインパーツの出力パラメタ（プラグインパーツの右側のポート）から遷移先のガイドパーツにドラッグしてマッピング線を引きます。

プラグインが設定した値がガイドパーツに渡されるようになります。

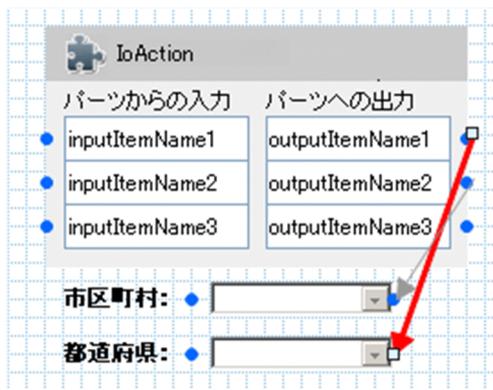
補足事項

- マッピング線は、次の図に示すように色つきの矢印で表示されます。それぞれのマッピング線は色が異なるため、マッピング線同士が交差していても区別しやすくなっています。

■プラグインパーツを選択したとき



■マッピング線を選択したとき



- 一つのガイドパーツからは、複数の入力パラメタにマッピング線を引けます。
- 複数の出力パラメタから一つのガイドパーツにマッピング線を引くことはできません。
- プラグインパーツのポートには、マッピング線を引かないこともできます。
- プラグインパーツを配置した場合は、プラグインパーツに一つもマッピング線が引かれていない状態でも、業務編集画面でプラグインを配置したノード間が遷移すると、プラグインの処理が呼び出されます。
- この作業で設定するマッピング線やプラグインパーツは、業務実行画面には表示されません。

ガイドパーツのキャッシュ値に関する注意事項

ガイドパーツのキャッシュ値は、ガイドパーツごとに一つの値が共有されます。このため、合流ノードを含む業務フローの場合、一度表示したプロセスノードを別のルートを通して再度表示すると、最初に表示したときのキャッシュ値がそのまま利用されます。

入出力プラグイン（サーバ処理実装インタフェース）を利用すると、条件に応じてキャッシュ値を変更できます。サーバ処理実装インタフェースについては、「5.3 IIoPluginController（サーバ処理実装インタフェース）」を参照してください。

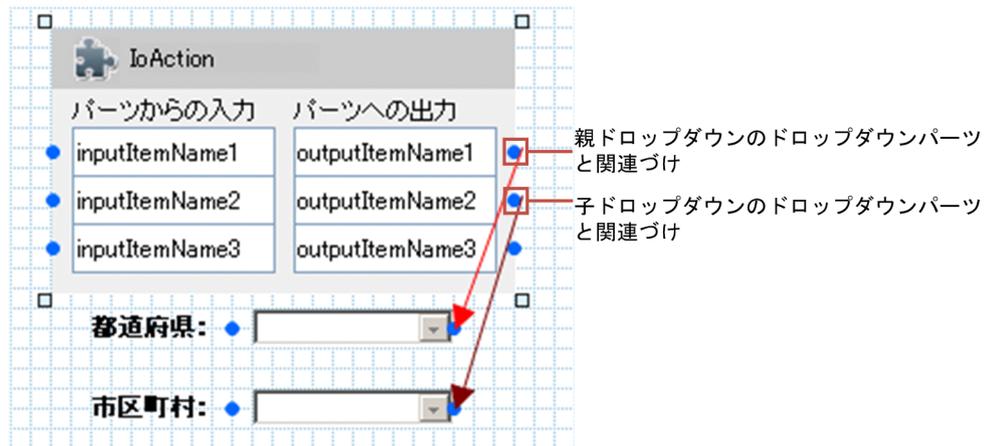
親子関係があるドロップダウンパーツと関連づける場合の注意事項

親子関係があるドロップダウンパーツとプラグインパーツを関連づける場合、親ドロップダウンに関連づけるパラメタが子ドロップダウンに関連づけるパラメタよりも上になるように設定してください。

親子ドロップダウンのパラメタは連続していなくてもかまいません。

出力パラメタ（プラグインパーツの右側のポート）を親子関係があるドロップダウンパーツと関連づける場合の例を次に示します。

図 3-4 親子関係があるドロップダウンパーツと関連づける場合の例



この例では、「outputItemName1」を親ドロップダウンに、「outputItemName2」を子ドロップダウンに関連づけています。親子ドロップダウンのパラメタが連続していなくてもよいので、例えば「outputItemName3」を子ドロップダウンに関連づけることもできます。

なお、出力パラメタだけでなく、入力パラメタを親子関係があるドロップダウンパーツと関連づける場合も、ここで説明した注意事項に従ってください。

プラグインを修正した場合の操作

関連づけが完了したあとにプラグインを修正した場合、修正後のプラグインに対応するプラグインパーツを再配置して、マッピング線を引き直す必要があります。

ただし、修正内容によっては、ヘッダ領域の「プラグインパーツ更新」メニューをクリックすることでプラグインパーツの関連づけを更新できます。詳細は、「3.9.3 プラグインパーツを更新する」を参照してください。

3.9.2 プラグインに入力、または出力する値の詳細

ガイドパーツごとに、プラグインに入力または出力する値について次に示します。

表 3-3 ガイドパーツからプラグインへの入力値とプラグインからガイドパーツへの出力結果

ガイドパーツの種類	プラグインへの入力値	ガイドパーツへの出力結果
固定テキスト	固定テキストに表示した文字列が入力される。	固定テキストに文字列が出力される。
テキストボックス	テキストボックスに入力した文字列が入力される。	テキストボックスに文字列が出力される。
テキストエリア	テキストエリアに入力した文字列が入力される。	テキストエリアに文字列が出力される。
ラジオボタン	チェックされたラジオボタンの値が入力される。	対応するラジオボタンがチェック状態になる。
チェックボックス	チェックボックスのチェック状態によって次の値が入力される。 チェック状態の場合 true 未チェック状態の場合 false	値が「true」の場合だけ、チェックボックスがチェック状態になる。 それ以外の値の場合はチェック状態にならない。

ガイドパーツの種類	プラグインへの入力値	ガイドパーツへの出力結果
ドロップダウン	選択状態のラベルに対応する値が入力される。	選択状態にしたいラベルに対応する値が出力される。
ハイパーリンク	[属性の設定] 画面で指定したアンカーの文字列と URL とを改行コード (¥r¥n) で連結した値が入力される。	アンカーの文字列と URL とを改行コード (¥r¥n) で連結した値 ^{*1} が出力される。
画像	[属性の設定] 画面で指定した画像パーツの URL と画像の説明 ([ツールチップ] テキストボックスの値) とを改行コード (¥r¥n) で連結した値 ^{*2} が入力される。	画像パーツに設定する URL と画像の説明 ([ツールチップ] テキストボックスに入力される値) とを改行コード (¥r¥n) で連結した値 ^{*3} が出力される。
インラインフレーム	インラインフレームを表示するための URL が入力される。	インラインフレームを表示するための URL が出力される。
HTML パーツ	HTML パーツの [HTML ソース文字列] に設定されている内容のうち、テキストボックス、パスワードボックス、hidden、チェックボックス、ラジオボタン、セレクトボックス、テキストエリアの name 属性および value 属性を取得してエンコードした文字列が入力される。	HTML パーツの [HTML ソース文字列] に設定する値が出力される。

注※1

出力される値については次の注意事項があります。

- ・値が null, 空文字, または改行コードだけの場合は, 空文字が出力されます。
- ・改行コードがない場合は, すべてアンカーの文字列として処理されます (URL には空文字が出力されます)。
- ・改行コードが二つ以上ある場合は, 3 行目以降の値が無視されます。

注※2

ツールチップが未設定 (空文字) の場合, 入力値は「URL + 改行コード (¥r¥n)」となります。URL だけを切り出したい場合は, 改行コードに注意してください。

注※3

出力される値については次の注意事項があります。

- ・値が null の場合は, 空文字が出力されます。
- ・改行コード (¥r¥n) がいない場合は, すべて URL として扱われます。ツールチップには空文字が出力されます。
- ・改行コードが二つ以上ある場合は, 3 行目以降の値が無視されます。

3.9.3 プラグインパーツを更新する

プラグインを変更した場合, 個々のプラグインパーツを再配置してマッピング線を引き直すよりも, [プラグインパーツ更新] メニューを使用してシステム内の全業務コンテンツに存在するプラグインパーツを一括で更新の方が効率的なことがあります。なお, 更新の対象になるプラグインパーツはユーザプラグインです。システムプラグインは更新の対象になりません。

[プラグインパーツ更新] メニューを使用する場合に更新対象となる項目と実行される更新処理を次に示します。

表 3-4 [プラグインパーツ更新] メニューで更新対象となる項目と実行される更新処理

項番	更新対象となる項目	実行される更新処理
1	パラメタの追加	パラメタだけが追加され, 新しいマッピング線は引かれません。

項番	更新対象となる項目	実行される更新処理
2	パラメタの削除	削除したパラメタに関連づけていたマッピング線が削除されます。
3	パラメタの移動（入力パラメタ同士または出力パラメタ同士を入れ替えた場合）	マッピング線を維持したまま、パラメタの位置だけが変更されます。
4	パラメタの移動（入力パラメタと出力パラメタを入れ替えた場合）	マッピング線が削除され、パラメタの位置が変更されます。
5	パラメタ名の変更	変更前の名称のパラメタに関連づけられていたマッピング線が削除されます。
6	パラメタの説明（ツールチップ）	変更後の説明に更新されます。
7	入出力アクションコントローラクラスを識別する文字列※1の変更	変更後の文字列に更新されます。
8	入出力プラグインの説明（ツールチップ）	変更後の説明に更新されます。
9	入出力プラグインの実行順序	変更後の順序に更新されます。
10	プラグインパーツのプラグインアイコンの変更※2	変更後のプラグインアイコンに更新されます。

注※1

「3.1.1(3) プロパティキーの詳細」を参照してください。

注※2

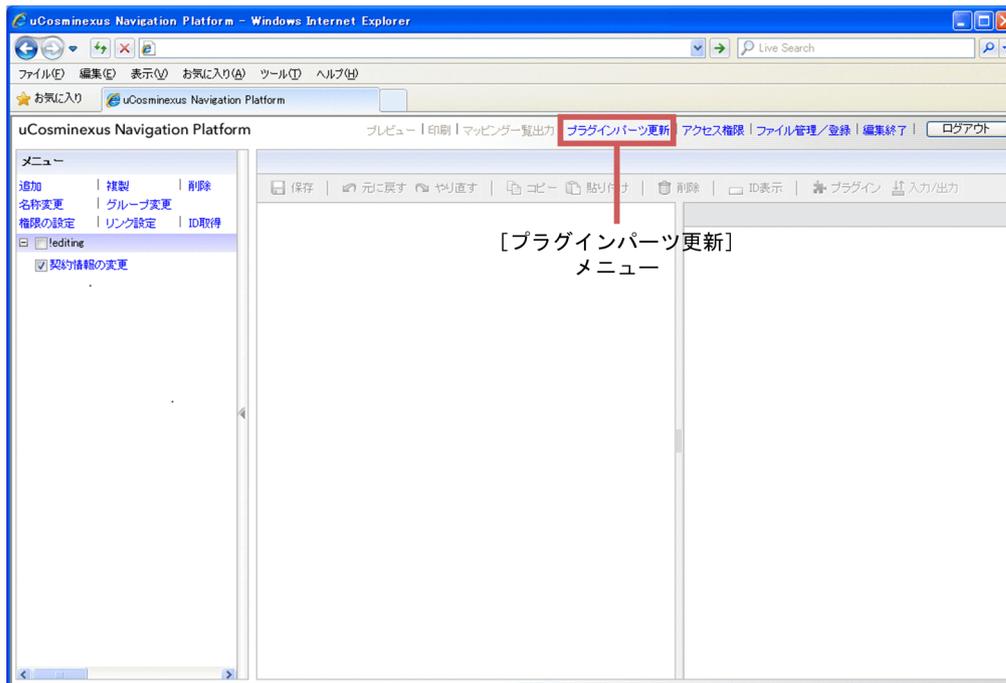
アイコンの変更方法については、「3.3.1 入出力プラグインのカスタマイズ」を参照してください。

[プラグインパーツ更新] メニューを使用する手順を次に示します。なお、ここで説明する手順を実行するには、ナビゲーション プラットフォームのシステム管理者の権限が必要です。システム管理者の権限を持つユーザ ID については、マニュアル「Cosminexus V9 ナビゲーション プラットフォーム 導入から運用まで」を参照してください。

手順

- 次のコマンドを実行して業務コンテンツのバックアップを作成し、業務コンテンツの一覧を出力します。
`npexport [-c△バックアップしたい業務コンテンツのコンテンツID]△[-l△バックアップ一覧ファイルのパス]△バックアップ先のファイルのパス`
- システム管理者のユーザ ID でナビゲーション プラットフォームにログインして、業務編集画面を表示します。
 ログインと業務編集画面の表示方法については、マニュアル「Cosminexus V9 ナビゲーション プラットフォーム コンテンツ作成ガイド」のログインおよび業務編集画面の表示に関する説明を参照してください。
- 業務編集画面の [プラグインパーツ更新] メニューを選択します。

3 プラグインの開発



変更したプラグインの内容が、システムに反映されます。このとき、変更したプラグインと関連づけがある業務コンテンツがすべて更新されます。

4. 手順 1.のコマンドを再度実行し、更新後の業務コンテンツの一覧を出力します。

業務コンテンツの一覧が出力されます。

5. 手順 1.と手順 4.で出力した業務コンテンツの一覧ファイルを比較し、どの業務コンテンツが更新されたのかを確認します。

6. 更新された業務コンテンツを業務編集画面上で表示し、必要に応じてガイドパーツとプラグインパーツの間に新しいマッピング線を引きます。

ガイドパーツとプラグインパーツの間にマッピング線を引く手順は、「3.9.1 マッピング線を引く（ガイドパーツとプラグインパーツをつなぐ）」を参照してください。

注意事項

- プラグインパーツの更新中は、業務編集画面でほかの操作をしないでください。
- プラグインパーツを更新中に業務コンテンツが操作されるとエラーが発生します。対象となる業務コンテンツが利用されていないことを確認してから、プラグインパーツを更新してください。
- プラグインパーツの更新中は、npimport コマンド（業務コンテンツのインポート）または npexport コマンド（業務コンテンツのエクスポート）を実行しないでください。

3.9.4 プラグインを使用している業務コンテンツの設定情報を確認する

[マッピング一覧] リンクでマッピング一覧ファイルを出力すると、プラグインを使用している業務コンテンツの設定情報を確認できます。出力手順や出力内容の詳細は、マニュアル「Cosminexus V9 ナビゲーションプラットフォーム コンテンツ作成ガイド」を参照してください。

3.10 ユーザプロパティファイルの設定（中断再開プラグインの場合）

開発した中断再開プラグインを有効にするには、ユーザプロパティファイル（ucnp_user.properties）の ucnp.base.client.suspend.enable プロパティに true を設定する必要があります。ユーザプロパティファイルの格納場所やプロパティの詳細は、マニュアル「Cosminexus V9 ナビゲーション プラットフォーム 導入から運用まで」を参照してください。

3.11 プラグインのデバッグ

プラグインをデバッグするには、Cosminexus Management Server と Eclipse を使用します。Cosminexus Management Server の使用方法の詳細は、マニュアル「Cosminexus V9 アプリケーションサーバ 運用管理ポータル操作ガイド」を参照してください。

Eclipse のデバッグ機能を使用したデバッグ手順の例を次に示します。

手順例

1. 次の URL にアクセスして、Cosminexus Management Server を起動します。

http://ホスト名:ポート番号/mngsvr/

- ホスト名
開発環境の J2EE サーバマシンのホスト名または IP アドレスです。
- ポート番号
Cosminexus Management Server が使用するポート番号です。マニュアル「Cosminexus V9 ナビゲーション プラットフォーム 導入から運用まで」の手順どおりにセットアップした環境では、「28080」です。

2. Cosminexus Management Server の運用管理ポータルにログインします。

運用管理ポータルの画面が表示されます。

なお、マニュアル「Cosminexus V9 ナビゲーション プラットフォーム 導入から運用まで」の手順どおりにセットアップした環境では、ログイン時に入力する管理ユーザ ID が「admin」、パスワードが「ucnppass」です。

3. [論理サーバの環境設定] アンカーをクリックします。

論理サーバの環境設定画面が表示されます。

4. [サーバビュー] タブ - [論理 J2EE サーバ] - [J2EE サーバ] - [uCosminexus Navigation Developer の J2EE サーバ名称] をクリックし、[JVM] タブをクリックします。

[起動パラメタの設定] 画面 (J2EE サーバ) が表示されます。

5. 次のとおりに設定します。

- [拡張起動パラメタ] のいちばん下の行の [有効] のチェックボックスをチェックする
- [拡張起動パラメタ] に「-agentlib:jdwp=transport=dt_socket,server=y,address=8000*,suspend=n」と入力する

注※

「address=8000」の「8000」は、Eclipse から接続するときに使用するポート番号です。Eclipse のデバッグの設定に合わせて、「address=」に指定するポート番号を変更してください。

6. [追加] ボタンをクリックします。

7. 画面右上の [設定情報の配布] アンカーをクリックします。

[設定情報の配布] 画面が表示されます。

8. 論理サーバ名に uCosminexus Navigation Developer の J2EE サーバ名が表示されている行の [配布] チェックボックスをチェックします。

論理サーバが選択されます。

9. [選択して配布] ボタンをクリックします。

手順 8. で選択した論理サーバの設定情報が、uCosminexus Navigation Developer の J2EE サーバに配布されます。

10. uCosminexus Navigation Developer の J2EE サーバを再起動します。
11. Eclipse を起動します。
12. [プロジェクト・エクスプローラー] で「3.2 ひな形プラグインのプロジェクトのインポート」でインポートしたひな形プラグインのプロジェクトをクリックします。
プロジェクトが選択されます。
13. Eclipse のメニューから [実行] - [デバッグの構成] を選択します。
[デバッグ構成] ダイアログが表示されます。
14. [リモート Java アプリケーション] を右クリックすると表示されるメニューの [新規] をクリックします。
新しいリモート Java アプリケーションが作成されます。
15. [デバッグ] ボタンをクリックします。
デバッガが実行されます。
16. 実装した Java コードにブレークポイントを設定します。
17. デバッグをします。
Web ブラウザで、プラグインが動作する操作をしてください。

3.12 プラグインの削除

この節では、プラグインを削除する方法をプラグインの種類ごとに説明します。

3.12.1 プラグインの削除（入出力プラグインの場合）

入出力プラグインの削除手順を次に示します。

手順

1. 業務コンテンツから削除対象のプラグインの関連づけを削除します。
2. 「3.7 Web サーバリダイレクトの設定」で追加した、削除対象のプラグインに対応する設定を削除します。
3. Web サーバを再起動します。
手順 2. で削除した内容が Web サーバに反映されます。
4. 「3.5 アプリケーション属性ファイル (application.xml) の編集」で追加した、削除対象のプラグインに対応する記述を削除します。
5. 削除対象のプラグインのディレクトリ (*uCosminexus Navigation Developer* インストールディレクトリ¥pluginSDK¥plugin¥プラグイン ID) を削除します。
6. build.xml の clean ターゲットを呼び出します。
clean ターゲットを呼び出す手順は、プラグインをビルドする手順と同じです。
プラグインをビルドする手順については、「3.6.1 プラグインのビルド手順」を参照してください。このとき、手順 5. の [ear] を [clean] に読み替えてください。
なお、clean ターゲットの呼び出しは、プラグイン内のライブラリやクラスの一部を削除する場合も実行してください。
7. build.xml の ear ターゲットを呼び出します。
プラグインをビルドする手順については、「3.6.1 プラグインのビルド手順」を参照してください。
8. uCosminexus Navigation Developer の再インポート・開始を実行します。
uCosminexus Navigation Developer のインポート・開始を実行する手順については、「3.8 uCosminexus Navigation Developer の J2EE アプリケーションのインポートと開始」を参照してください。

3.12.2 プラグインの削除（中断再開プラグインの場合）

中断再開プラグインの削除手順を次に示します。

手順

1. 「3.5 アプリケーション属性ファイル (application.xml) の編集」で追加した、削除対象のプラグインに対応する記述を削除します。
2. 削除対象のプラグインのディレクトリ (*uCosminexus Navigation Developer* インストールディレクトリ¥pluginSDK¥plugin¥プラグイン ID) を削除します。
3. build.xml の clean ターゲットを呼び出します。
clean ターゲットを呼び出す手順は、プラグインをビルドする手順と同じです。
プラグインをビルドする手順については、「3.6.1 プラグインのビルド手順」を参照してください。このとき、手順 5. の [ear] を [clean] に読み替えてください。

なお、clean ターゲットの呼び出しは、プラグイン内のライブラリやクラスの一部を削除する場合も実行してください。

4. build.xml の ear ターゲットを呼び出します。

プラグインをビルドする手順については、「3.6.1 プラグインのビルド手順」を参照してください。

5. uCosminexus Navigation Developer の再インポート・開始を実行します。

uCosminexus Navigation Developer のインポート・開始を実行する手順については、「3.8 uCosminexus Navigation Developer の J2EE アプリケーションのインポートと開始」を参照してください。

3.13 プラグインが使用するメモリ使用量の計算

ユーザプラグインが使用するメモリ使用量を求める式を次に示します。

メモリ使用量A^{*}-メモリ使用量B=ユーザプラグインが使用するメモリ使用量

(凡例)

メモリ使用量 A：ユーザプラグインを使用して作成した業務コンテンツを実行した場合

メモリ使用量 B：ユーザプラグインを使用しないで作成した業務コンテンツを実行した場合

注※

ucnpOptions パラメタを使用する場合、業務コンテンツを実行する際に ucnpOptions パラメタを指定して測定してください。

3.13.1 測定用の業務コンテンツの作成手順

ユーザプラグインが使用するメモリ使用量を計算するためには、同じ業務コンテンツでユーザプラグインを使用した場合と使用しない場合で実行して、それぞれのメモリ使用量を測定する必要があります。

ユーザプラグインを使用しない業務コンテンツを作成する手順を次に示します。

1. ナビゲーション プラットフォームにログインします。
2. ユーザプラグインを使用して作成した業務コンテンツをコピーします。
3. コピーした業務コンテンツからユーザプラグインを削除します。
4. 業務コンテンツを保存します。

3.13.2 プラグインが使用するメモリ使用量の測定手順

メモリ使用量の測定手順を次に示します。

(1) J2EE サーバの再起動

1. npstart コマンド (起動) でナビゲーション プラットフォームを起動します。
2. 次の URL にアクセスして、Cosminexus Management Server にログインします。
<http://ホスト名:ポート番号/mngsvr/>
 - ホスト名
開発環境の J2EE サーバマシンのホスト名または IP アドレスです。
 - ポート番号
Cosminexus Management Server が使用するポート番号です。マニュアル「Cosminexus V9 ナビゲーション プラットフォーム 導入から運用まで」の手順どおりにセットアップした環境では、「28080」です。
3. 運用管理ポータルで [論理サーバの起動/停止] アンカーをクリックし、論理サーバの起動/停止画面を表示します。
4. ツリーから [論理 J2EE サーバ] - [J2EE サーバ] - [ナビゲーション プラットフォームの J2EE サーバ名] をクリックし、[起動/停止] タブを表示します。
 ナビゲーション プラットフォームの J2EE サーバ名は、マニュアル「Cosminexus V9 ナビゲーション プラットフォーム 導入から運用まで」の手順どおりにセットアップした環境では、「MyJ2EE」です。

5. [停止] ボタンをクリックします。
6. J2EE サーバが停止したら、[起動/停止] タブを表示し直します。
7. [起動] ボタンをクリックします。
J2EE サーバの再起動が完了します。

(2) 業務コンテンツの実行

1. ナビゲーション プラットフォームにログインします。
2. メモリ使用量を測定する業務コンテンツを選択します。
3. [次へ] ボタンをクリックしながら最後まで業務コンテンツを実行します。
分岐によって複数の経路を持つ業務コンテンツの場合は、プラグインを配置したノードを通る経路で実行してください。

実行完了後の注意事項

業務コンテンツの実行後は、ほかの業務コンテンツを表示したりログアウトしたりしないでください。

(3) GC (ガーベージコレクション) の実行

1. 次のコマンドを実行してネーミングサービスのポートを使用しているプロセス ID を取得します。

```
netstat -o
```

コマンドの出力例を次に示します。

```
TCP マシンのホスト名またはIPアドレス:900 localhost:1176 ESTABLISHED
1876
```

この例では、「1876」がネーミングサービスのポートを使用しているプロセス ID となります。ネーミングサービスのポートがわからない場合は、ユーザセットアッププロパティファイル (ucnp_setup_user.properties) の ucnp.setup.server.cosminexus.ejbserver.naming.port プロパティに指定した値を確認してください。デフォルトの場合は「900」です。

2. 次のコマンドで GC を実行します。

```
"Cosminexusインストールディレクトリ¥jdk¥jre¥bin¥javagc.exe" -p ネーミングサービスのポートを使用しているプロセスID
```

コマンドを実行すると次のメッセージが表示されます。「y」を入力後< Enter >キーを押してください。

```
Force VM to execute GC: ? (y/n)
```

javagc コマンドについては、マニュアル「Cosminexus V9 アプリケーションサーバ リファレンス コマンド編」を参照してください。

(4) ナビゲーション プラットフォームの J2EE アプリケーションの停止

1. 次の URL にアクセスして、Cosminexus Management Server にログインします。

```
http://ホスト名:ポート番号/mngsvr/
```

- ホスト名

開発環境の J2EE サーバマシンのホスト名または IP アドレスです。

- ポート番号

Cosminexus Management Server が使用するポート番号です。マニュアル「Cosminexus V9 ナビゲーション プラットフォーム 導入から運用まで」の手順どおりにセットアップした環境では、「28080」です。

2. 運用管理ポータルで [論理サーバのアプリケーション管理] アンカーをクリックし、論理サーバのアプリケーション管理画面を表示します。
3. ツリーから [論理 J2EE サーバ] - [J2EE サーバ] - [ナビゲーション プラットフォームの J2EE サーバ名] - [アプリケーション] をクリックし、[開始/停止] タブを表示します。
4. インポート J2EE アプリケーション一覧で、アプリケーション名「ucnp」の [停止] アンカーをクリックします。

(5) 稼働統計の確認

次のパスに出力されている、JavaVM の稼働統計ファイルを確認します。

*Cosminexus*インストールディレクトリ¥CC¥server¥public¥ejb¥ナビゲーション プラットフォームのJ2EEサーバ名¥stats¥HJVMSstats_YYYYMMDDhhmmTZ.csv

(凡例)

YYYYMMDDhhmm：稼働統計ファイルが作成された日時

TZ：タイムゾーン

稼働統計ファイルを Excel で開いた場合、AK 列に業務コンテンツを実行したときのメモリ使用量が出力されています。メモリ使用量は、ナビゲーション プラットフォームの J2EE アプリケーションを停止してから、過去 60 秒間に Explicit ヒープから解放した 1 セッション当たりの最大メモリ使用量 (単位：バイト) です。

3.14 iPad 向け業務コンテンツで使用するプラグインを開発する場合の注意事項

Mobile Safari では、JavaScript の実行時間が 10 秒以上かかると処理が停止するおそれがあります。通信時間を含めて 10 秒以内にプラグインの処理が完了されるように開発してください。

3.15 09-50 より前のバージョンで開発したプラグインを使用する場合の注意事項

09-50 より前のバージョンの入出力プラグイン（以降、旧入出力プラグインと呼びます）と 09-50 以降の入出力プラグインは使用できる機能が異なります。

ここでは、旧入出力プラグインを 09-50 以降のバージョンで使用する場合の注意事項について説明します。詳細については弊社にお問い合わせください。

3.15.1 09-50 より前のバージョンで開発したプラグインを使用する場合の制限

旧入出力プラグインを使用する場合の制限を次に示します。

表 3-5 旧入出力プラグインを使用する場合の制限

項番	機能	説明
1	IioActionController（入出力アクションコントローラインタフェース）	IioActionController（入出力アクションコントローラインタフェース）に合わせて Java プログラムおよび定義ファイルを変更した場合は、09-50 以降のバージョンの入出力プラグインと同様に動作します。ただし、動作は保証されません。
2	IioActionController（入出力アクションコントローラインタフェース）の execute メソッド	ucnp.button.type キーの値は 09-50 で一部変更され、次のどれかの値が格納されます。 <ul style="list-style-type: none"> • "show_previous_page" • "show_next_page" • "show_complete_page" • "start" そのため、このキーを使用している場合は必要に応じて見直してください。
3	実行順序	実行順序を指定している場合の動作は保証されません。次に示す配置順序が優先されます。 <ul style="list-style-type: none"> • プロセスノード間の移動 遷移元の配置順序 • プロセスノードとターミナルノード間の移動 プロセスノードの配置順序
4	入出力プラグインの説明	指定している場合は 09-50 以降のバージョンの入出力プラグインと同様に動作します。ただし、動作は保証されません。
5	入力パラメータまたは出力パラメータの説明	

3.15.2 09-50 より前のバージョンで開発したプラグインの実行順序とタイミング

ノードを遷移するとき、遷移元または遷移先のノードに応じて実行する旧入出力プラグインを次に示します。

表 3-6 ノード遷移時に実行する旧入出力プラグイン

項番	遷移元ノード	遷移先ノード	実行する旧入出力プラグイン
1	プロセスノード	プロセスノード	遷移元ノードと遷移先ノードの両方に配置した入出力アクションコントローラのタイプ※, およびプラグインパーツ ID が同一の旧入出力プラグイン
2	ターミナルノード	プロセスノード	遷移先ノードに配置した旧入出力プラグイン
3	プロセスノード	ターミナルノード	遷移元ノードに配置した旧入出力プラグイン

注※

プラグイン情報プロパティファイル (plugin.properties.sam) の userplugin.server.controller.ioaction.type プロパティの値です。詳細は「3.1.1(3)(f) userplugin.server.controller.ioaction.type」を参照してください。

同じノードに複数の入出力プラグインを配置している場合、先に実行した execute メソッドの結果にかかわらず、表 3-6 の条件を満たすすべての旧入出力プラグインの execute メソッドを実行します。ただし、次の場合は旧入出力プラグインを実行しません。

- 旧入出力プラグインがターミナルノード（開始）に接続されていない先頭のプロセスノードに関連づけられているとき
- 遷移元と遷移先が同じノードのとき

旧入出力プラグインは 09-50 以降のバージョンの入出力プラグインの入力と出力の間に実行します。ノード遷移時に実行する入出力プラグインの順序を次の表に示します。ただし、先に実行した inputFromNode メソッドの実行結果の種別がエラーで終了したときは、すべての入出力プラグインの execute メソッドを実行しません。また、execute メソッドがエラーで終了した（戻り値に ucnp.error.message キーが指定されている）場合、すべての入出力プラグインの outputToNode メソッドを実行しません。

表 3-7 入出力プラグインを実行する順序

実行順序	入出力プラグイン	出力プラグインのメソッド
1	遷移元ノードのガイドに関連づけられている、09-50 以降のバージョンの入出力プラグイン	IioPluginController.inputFromNode
2	旧入出力プラグイン	IioActionController.execute
3	遷移先ノードのガイドに関連づけられている、09-50 以降のバージョンの入出力プラグイン	IioPluginController.outputToNode

3.15.3 09-50 より前のバージョンで開発したプラグインのインスタンス

(1) インスタンスの保持

旧入出力プラグインは、同じノードに対して同一プラグイン ID の入出力プラグインを複数配置できます。このため、同一プラグイン ID かつ同一プラグインパーツ ID の入出力プラグインのインスタンスを、一画面につき一つずつ保持して同一プラグイン ID のインスタンスを区別します。なお、インスタンスを生成するタイミングは 09-50 以降のバージョンの入出力プラグインと同じです。

(2) インスタンスの破棄

09-50 以降のバージョンの入出力プラグインと同じです。詳細は「1.2.3(2) 入出力プラグインのインスタンスの破棄」を参照してください。

(3) 09-50 より前のバージョンと 09-50 以降のバージョンのインスタンスの違い

入出力プラグインのインスタンスは、09-50 より前のバージョンと 09-50 以降のバージョンで次のように異なります。

09-50 より前のバージョンの場合

ノード遷移時に入出力プラグインを実行するたび、入出力プラグインのインスタンスが生成および破棄されます。

09-50 以降のバージョンの場合

一度生成したインスタンスは、同一の業務コンテンツを実行している間は HTTP セッションで保持されます。

注意事項

インスタンスのライフサイクルの変更に伴い、メンバ変数も入出力プラグインの実行時に破棄されずに残ります。このため、旧入出力プラグインでメンバ変数を使用している場合は、`IioActionController.execute` メソッドの実行後にメンバ変数が残っていても問題ないかどうかご注意ください。

4

カスタム画面の開発

ナビゲーションプラットフォームの画面は、メニュー領域の表示形式を変更したり、特定の業務コンテンツだけを表示させたりするなどユーザーが使いやすいようにカスタマイズできます。

この章では、ナビゲーションプラットフォームの画面のメニュー領域をカスタマイズする方法、および新規に画面を開発する方法について説明します。

4.1 カスタマイズに使用するファイルの格納先とパス

カスタマイズに使用するファイルの格納先とパスについて説明します。

4.1.1 静的ファイルの格納先とパス

HTML ファイルなどの静的ファイルは、Web サーバのコンテンツ格納ディレクトリに格納してください。デフォルトの Web サーバのコンテンツ格納ディレクトリを次に示します。

`Cosminexus`インストールディレクトリ`¥httpd¥htdocs`

Web サーバのコンテンツ格納ディレクトリの詳細は、マニュアル「[Cosminexus V9 アプリケーションサーバ 運用管理ポータル操作ガイド](#)」を参照してください。静的ファイルの格納方法に関する詳細は、マニュアル「[Cosminexus V9 ナビゲーション プラットフォーム コンテンツ作成ガイド](#)」を参照してください。

! 注意事項

Web サーバのコンテンツ格納ディレクトリをデフォルトから変更する場合は、論理サーバのディレクトリ配下以外のディレクトリを設定してください。

なお、Web サーバのコンテンツ格納ディレクトリをデフォルトから変更した場合、ユーザプロパティファイル (`ucnp_user.properties`) の `ucnp.base.server.cosminexus.httpd.documentroot.path` プロパティの指定値も変更する必要があります。プロパティの詳細は、マニュアル「[Cosminexus V9 ナビゲーション プラットフォーム 導入から運用まで](#)」を参照してください。

4.1.2 JSP ファイルの格納先とパス

カスタマイズに JSP ファイルを使用する場合、ポータルプロジェクトのディレクトリ配下で新しく作成したディレクトリに格納します。ポータルプロジェクトのディレクトリ配下にある既存のディレクトリには格納しないでください。ディレクトリは次の構成で作成し、JSP ファイルを `custom` ディレクトリに格納します。

ポータルプロジェクトディレクトリ`¥ucnp¥custom`

ポータルプロジェクトのディレクトリは、ユーザセットアッププロパティファイル (`ucnp_setup_user.properties`) の `ucnp.setup.server.ucnp.portalproject.path` プロパティに指定したパスです。

4.2 メニュー領域のカスタマイズ

ナビゲーション プラットフォームの画面は、次の点をカスタマイズできます。

- 業務実行画面のメニュー領域の表示形式を変更する
HTML ファイルを使って業務実行画面を見やすくカスタマイズしたり、JavaScript 関数を使って特定の業務コンテンツだけを業務実行画面に表示したりできます。
- メニュー領域の表示幅を変更する
ユーザプロパティファイル (ucnp_user.properties) で、メニュー領域の表示幅を変更できます。プロパティについては、マニュアル「Cosminexus V9 ナビゲーション プラットフォーム 導入から運用まで」を参照してください。
- 業務実行画面のメニュー領域の機能

注意事項

業務編集画面のメニュー領域は変更できません。

4.2.1 ナビゲーション プラットフォームで提供している JavaScript 関数

業務実行画面のメニュー領域のカスタマイズには、JavaScript 関数を使用できます。ナビゲーション プラットフォームで提供している JavaScript 関数について説明します。

業務コンテンツをプレビュー、または直接表示する場合の注意事項

ucnp_menu_onload 関数を使ってメニュー領域をカスタマイズしていないと、業務編集画面でプレビューをする場合、および業務コンテンツ ID を指定して特定の業務コンテンツを直接表示する場合に、問題が発生することがあります。これらの操作が正しく実行できない場合は、カスタマイズ用に作成したファイルの内容を見直してください。

(1) ucnp_menu_show_flow

関数名

ucnp_menu_show_flow(*id*)

機能

引数に指定した ID の業務コンテンツを表示します。

中断再開プラグインを有効にした場合は、業務コンテンツを表示するときに、中断した業務を再開します。中断再開プラグインを無効にした場合は、業務コンテンツを表示するだけです。

引数

id :

業務コンテンツ ID を指定します。業務コンテンツ ID の取得方法を次に示します。

1. 業務編集画面のメニュー領域で、ID を取得したい業務コンテンツのチェックボックスをチェックします。
2. [ID 取得] メニューをクリックします。
チェックボックスをチェックした業務コンテンツの業務コンテンツ ID がクリップボードにコピーされます。

戻り値

false

使用例

ID が 111111, 名称が「契約内容の確認」の業務コンテンツを表示する場合の記述例を次に示します。

```
<a href="javascript:void(0);" onclick="ucnp_menu_show_flow('111111');return false;">契約内容  
の確認</a>
```

(2) ucnp_menu_get_contentId (09-50 より前のバージョンとの互換用)

関数名

ucnp_menu_get_contentId()

機能

ナビゲーション プラットフォームを呼び出すときに URL に指定した業務コンテンツ ID を取得します。

メニュー領域の表示形式をカスタマイズしてナビゲーション プラットフォームを運用する場合、カスタマイズしたメニューから URL に指定した業務コンテンツ ID を取得したいときに使用します。例えば、利用者に特定の業務コンテンツを表示した状態でナビゲーション プラットフォームを使用させたいときなどに使用します。

なお、この関数は 09-50 より前のバージョンとの互換用です。09-50 以降のバージョンの開発では ucnp_menu_onload 関数の使用をお勧めします。詳細は「(3) ucnp_menu_onload」を参照してください。

引数

なし

戻り値

- ナビゲーション プラットフォームを呼び出すときに利用した URL の contentId パラメタの値
- URL に contentId パラメタを指定しなかった場合は null

(3) ucnp_menu_onload

関数名

ucnp_menu_onload(*elementId*)

機能

カスタム画面を読み込むときに、初期化処理を実行します。

この関数を body 要素の onload イベントで実行すると、次のように処理します。

- URL のリクエストパラメタ (contentId) に業務コンテンツ ID が指定されている場合
指定された業務コンテンツを開きます。
- この関数の引数に div 要素の ID を指定した場合

画面をリサイズするときに、div 要素に指定したエリアがメニュー領域の大きさに合うように調整します。また、body 要素と指定した div 要素のスタイルを変更します。

メニュー領域は iframe (HTML のインラインフレーム) で表示していますが、Mobile Safari では iframe の大きさを制御できないため、メニュー領域の内容が途中までしか表示されない場合があります。この関数を使用すると、Mobile Safari でもメニュー領域全体を適切な大きさにリサイズできます。

また、Mobile Safari ではカスタマイズしたメニュー領域の高さ (カスタマイズメニューを格納する iframe の高さ) を適切なサイズで表示できない場合があるため、iPad で業務コンテンツを参照する場合は、引数 `elementId` を指定してこの関数を実行することをお勧めします。

なお、Internet Explorer では、この関数を使わなくてもリサイズ処理を実装できます。

プレビュー時、および業務コンテンツ ID を指定して業務実行画面を開くときにこの関数が実行されます。メニュー領域の表示形式をカスタマイズしてナビゲーション プラットフォームを運用する場合は、この関数の使用をお勧めします。

引数

`elementId` :

この引数は String 型です。

カスタマイズメニューの全体を含む div 要素の ID を指定します。リサイズ処理しない場合は省略できます。

戻り値

なし

4.2.2 メニュー領域の表示形式を変更する

業務実行画面のメニュー領域の表示形式をカスタマイズする手順を次に示します。

`ucnp_menu_onload` 関数に引数を指定すると、カスタム画面の読み込み時に初期化処理が実行されるため、画面もリサイズして表示されます。iPad でカスタム画面を表示する場合、適度にリサイズされるため便利です。

1. テキストエディタで、メニュー領域に表示させる HTML ファイル、または JSP ファイルを作成します。

作成するファイルの記述形式を次に示します。

なお、太字で示した行は必ず記述してください。

- 画面をリサイズする場合

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <script type="text/javascript" src="/ucnpBase/adportlets/ucnp/js/tools/UcnpCustomTools.js"></script>
</head>
<body onload="ucnp_menu_onload('resizeArea');">

  <div class="resizeArea" id="resizeArea">
    この部分に、HTMLタグおよびナビゲーション プラットフォームで提供している
    JavaScript関数を使用して、メニューの内容を記述します。
  </div>

</body>
</html>
```

- 画面をリサイズしない場合

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <script type="text/javascript" src="/ucnpBase/adport lets/ucnp/js/tools/UcnpCustomTools.js"></script>
</head>
<body onload="ucnp_menu_onload();">
この部分に、HTMLタグおよびナビゲーション プラットフォームで提供している
JavaScript関数を使用して、メニューの内容を記述します。
</body>
</html>
```

ナビゲーション プラットフォームで提供している JavaScript 関数の詳細は、「4.2.1 ナビゲーション プラットフォームで提供している JavaScript 関数」を参照してください。

2. 作成したファイルを格納します。

ファイルの種別ごとの、ファイルの格納先を次に示します。

- HTML ファイルの場合

実行環境の静的ファイル格納先に格納してください。静的ファイル格納先の詳細は、「4.1.1 静的ファイルの格納先とパス」を参照してください。

- JSP ファイルの場合

開発環境の JSP ファイル格納先に格納してください。JSP ファイル格納先の詳細は、「4.1.2 JSP ファイルの格納先とパス」を参照してください。

JSP ファイルを作成した場合は次の手順に進んでください。HTML ファイルを作成した場合は、手順 6.へ進んでください。

3. カスタム画面をビルドします。

カスタム画面のビルド手順はプラグインの場合と同じです。詳細は、「3.6.1 プラグインのビルド手順」を参照してください。

4. 開発環境のユーザプロパティファイル (ucnp_user.properties) を設定します。

設定するプロパティの詳細は、「4.4.3 ユーザプロパティファイルの設定」を参照してください。

5. J2EE アプリケーションをインポートして、開始します。

インポートと開始の手順は、「3.8 uCosminexus Navigation Developer の J2EE アプリケーションのインポートと開始」を参照してください。

6. J2EE アプリケーションをデバッグして、問題がないことを確認します。

デバッグ手順は、「3.11 プラグインのデバッグ」を参照してください。

7. 作成した J2EE アプリケーションのファイルを編集環境および実行環境に送付します。

開発環境から送付されたファイルを編集環境および実行環境に適用する方法の詳細は、マニュアル「Cosminexus V9 ナビゲーション プラットフォーム 導入から運用まで」を参照してください。

8. 編集環境および実行環境のユーザプロパティファイル (ucnp_user.properties) を編集します。

どのプロパティを編集するのかについては、「4.4.3 ユーザプロパティファイルの設定」を参照してください。

(1) カスタマイズしたメニュー領域のソース例

カスタマイズしたメニュー領域のソース例を次に示します。なお、この例ではカスタム画面がリサイズされます。

```

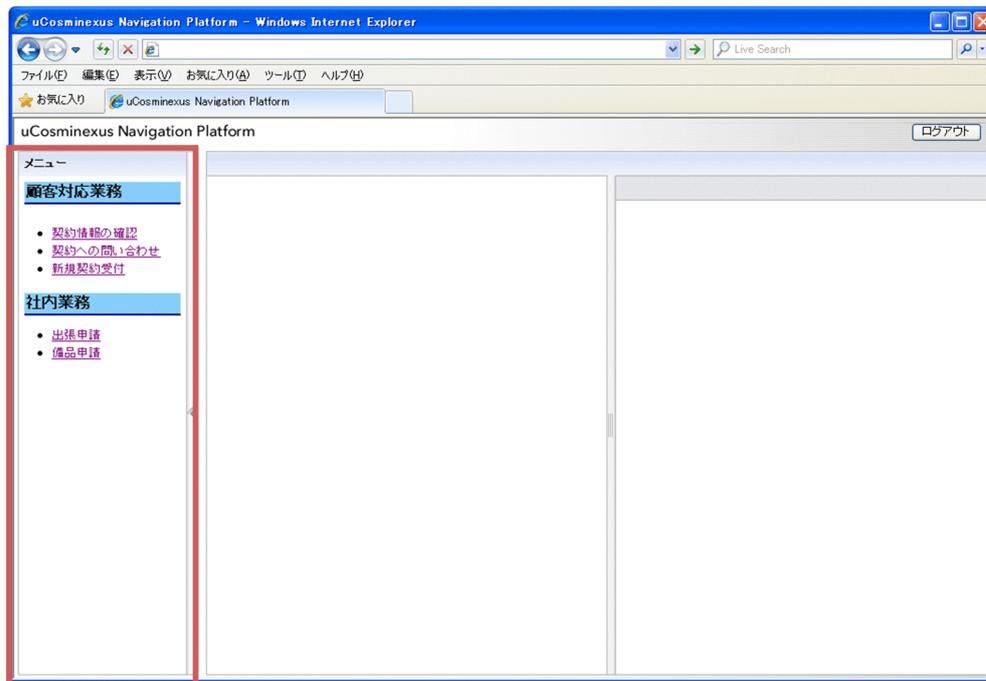
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <script type="text/javascript" src="/ucnpBase/adportlets/ucnp/js/tools/
UcnpCustomTools.js"></script>
</head>
<body onload="ucnp_menu_onload('resizeArea');">
<div id="resizeArea">
  <table border="0" cellspacing="0" cellpadding="0" style="width:100%;">
    <tr bgcolor="#00ccff"><td><font size="4"><b>顧客対応業務</b></font></td></tr>
    <tr bgcolor="#0033ff"><td></td></tr>
    <tr bgcolor="#0033ff"><td></td></tr>
    <tr><td><br></td></tr>
    <tr><td style="padding-left:14px;">
      <li><a href="javascript:void(0);" onclick="ucnp_menu_show_flow(' 契約情報の確認
の業務コンテンツID'); return false;">契約情報の確認</a></li>
    </td></tr>
    <tr><td style="padding-left:14px;">
      <li><a href="javascript:void(0);" onclick="ucnp_menu_show_flow(' 契約への問い合
せ)の業務コンテンツID'); return false;">契約への問い合わせ</a></li>
    </td></tr>
    <tr><td style="padding-left:14px;">
      <li><a href="javascript:void(0);" onclick="ucnp_menu_show_flow(' 新規契約受付)の
業務コンテンツID'); return false;">新規契約受付</a></li>
    </td></tr>
  </table>
  <br>
  <table border="0" cellspacing="0" cellpadding="0" style="width:100%;">
    <tr bgcolor="#00ccff"><td><font size="4"><b>社内業務</b></font></td></tr>
    <tr bgcolor="#0033ff"><td></td></tr>
    <tr bgcolor="#0033ff"><td></td></tr>
    <tr><td><br></td></tr>
    <tr><td style="padding-left:14px;"><li><a href=" 出張申請)社内システムのURL"
target="_blank">出張申請</a></li></td></tr>
    <tr><td style="padding-left:14px;"><li><a href=" 備品申請)社内システムのURL"
target="_blank">備品申請</a></li></td></tr>
  </table>
</div>
</body>
</html>

```

(2) カスタマイズしたメニュー領域の表示例

メニュー領域の表示形式をカスタマイズした場合の表示例を次に示します。

図 4-1 メニュー領域の表示形式をカスタマイズした場合の例



カスタマイズしたメニュー領域

4.2.3 メニュー領域に検索機能を追加する

業務実行画面のメニュー領域に検索機能を追加する手順を次に示します。

1. ナビゲーション プラットフォームで提供している API を使用して、JSP ファイルを作成します。
API の詳細は、「4.2.1 ナビゲーション プラットフォームで提供している JavaScript 関数」および「6. API リファレンス (カスタム画面開発の場合)」を参照してください。
2. 作成した JSP ファイルを格納します。
JSP ファイル格納先の詳細は、「4.1.2 JSP ファイルの格納先とパス」を参照してください。
3. カスタム画面をビルドします。
カスタム画面のビルド手順はプラグインの場合と同じです。詳細は、「3.6.1 プラグインのビルド手順」を参照してください。
4. 開発環境のユーザプロパティファイル (ucnp_user.properties) を設定します。
設定するプロパティの詳細は、「4.4.3 ユーザプロパティファイルの設定」を参照してください。
5. J2EE アプリケーションをインポートして、開始します。
インポートと開始の手順は、「3.8 uCosminexus Navigation Developer の J2EE アプリケーションのインポートと開始」を参照してください。
6. J2EE アプリケーションをデバッグして、問題がないことを確認します。
デバッグ手順は、「3.11 プラグインのデバッグ」を参照してください。
7. 作成した J2EE アプリケーションのファイルを編集環境および実行環境に送付します。
開発環境から送付されたファイルを編集環境および実行環境に適用する方法の詳細は、マニュアル「Cosminexus V9 ナビゲーション プラットフォーム 導入から運用まで」を参照してください。

8. 編集環境および実行環境のユーザプロパティファイル (ucnp_user.properties) を編集し、J2EE サーバを再起動します。

ユーザプロパティファイルのどのプロパティを編集するのかについては、「4.4.3 ユーザプロパティファイルの設定」を参照してください。

(1) 検索機能（メニュー領域）のソース例

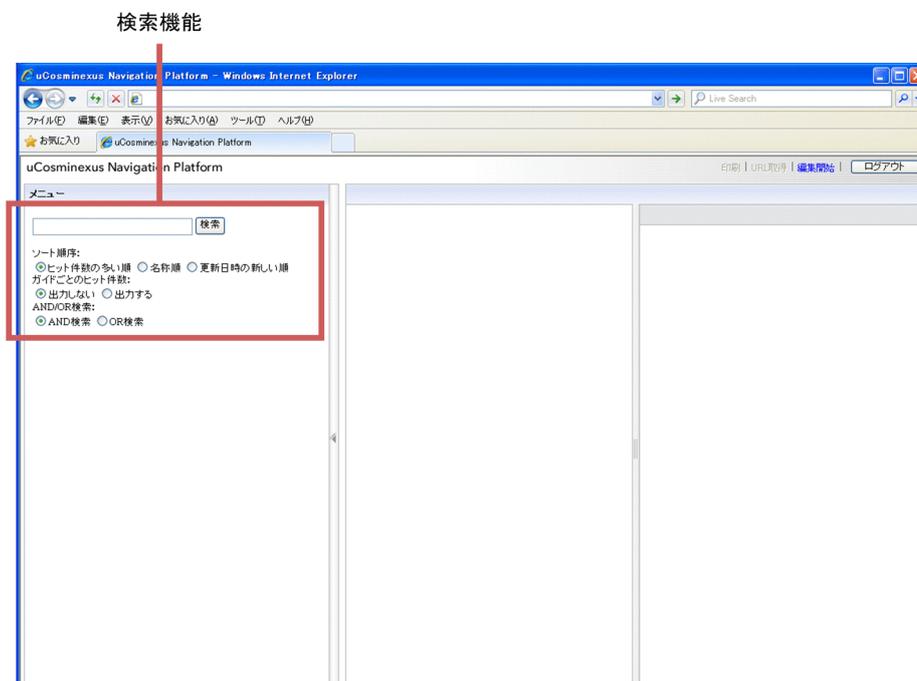
メニュー領域に検索機能を追加した場合のソース例は、次のファイルを参照してください。

ナビゲーション プラットフォームインストールディレクトリ¥sample¥search¥menu¥search_sample.jsp

(2) 検索機能（メニュー領域）の表示例

メニュー領域に検索機能を追加した場合の表示例を次に示します。

図 4-2 メニュー領域に検索機能を追加した場合の例



4.2.4 iPad 向け実行環境でメニュー領域をカスタマイズする場合の注意事項

iPad 向けの実行環境でメニュー領域をカスタマイズする場合は、次の点に注意してください。

- Mobile Safari でカスタマイズしたメニュー領域を表示させる場合は、ucnp_menu_onload 関数を使用してください。
ucnp_menu_onload 関数を使用していないと、画面の大きさが変更されたときに、カスタマイズしたメニュー領域の内容が途中でしか表示されないことがあります。
- iframe は、Mobile Safari で表示させる場合の大きさに制限があります。このため、カスタマイズしたメニュー領域内で iframe を使用することはお勧めしません。

4 カスタム画面の開発

Mobile Safari では、iframe 要素のサイズよりも src 属性に指定したページのサイズが優先されます。カスタマイズしたメニュー領域内で iframe を使用したい場合は、iframe 要素の src 属性にはメニュー領域よりもサイズが小さい画面を指定してください。

4.3 新規画面の開発

ナビゲーション プラットフォームの画面とは別に、JSP ファイルで開発した画面をナビゲーション プラットフォームの画面と連動させて使用できます。この節では、ナビゲーション プラットフォームで提供している Java の API を使用して新規画面を開発する手順を説明します。

参考

ここでは新規画面を開発し、機能を追加した場合の説明をしていますが、一度追加した機能を削除したり、画面そのものを削除したりする場合も、次の作業を実施する必要があります。

- 「4.4.1 カスタム画面のビルドとデバッグ」
- 「4.4.2 カスタム画面の取り込み」

画面そのものを削除する場合は、画面の JSP ファイルを削除します。さらに、削除した画面から参照しているライブラリがあれば、そのライブラリも削除してください。

4.3.1 検索機能を追加する

ナビゲーション プラットフォームで提供している Java の API を使用して新規画面に検索機能を追加する手順を次に示します。

1. ナビゲーション プラットフォームで提供している API を使用して、JSP ファイルを作成します。

API の詳細は、「6. API リファレンス (カスタム画面開発の場合)」を参照してください。

補足事項

カスタマイズしたナビゲーション プラットフォームのメニュー領域にデータを受け渡す場合は、`session.setAttribute()` メソッドおよび `getAttribute()` メソッドを利用します。このとき、`getAttribute()` メソッドの第 1 引数には「ucnp」で始まる値または `uCosminexus Portal Framework` の標準 API ポートレットが使用している値は指定できません。

また、新規画面を複数開発し、`form` タグなどを利用して新規画面から新規画面へ遷移したい場合は、「/ucnpBase」で始まる URL を指定する必要があります。例えば、遷移先の新規画面のファイルを `ポータルプロジェクトディレクトリ/ucnp/custom/jsp/userpage.jsp` に配置する場合は、URL に「/ucnpBase/ucnp/custom/jsp/userpage.jsp」と指定します。

2. 必要に応じて、作成した JSP ファイルにログアウト処理を追加します。

ログアウト処理の追加手順は、「4.3.2 ログアウト機能を追加する」を参照してください。

3. 必要に応じて、作成した JSP ファイルにほかのシステムからのデータを受け取る処理を追加します。

データを受け取るために必要なパラメタの詳細は、「4.3.3 任意のデータを受け取る機能を追加する」を参照してください。

4. 作成した JSP ファイルにセッション維持の処理を追加します。

通常のナビゲーション プラットフォームの画面のセッション維持は、ユーザプロパティファイル (`ucnp_user.properties`) の `ucnp.base.server.keepsession.interval` プロパティで設定しています。このプロパティの設定は、新規画面に対しては無効なため、専用の処理を追加してください。セッションタイムアウトの状態で検索機能が実行された場合、エラーが発生します。

5. 完成した JSP ファイルを格納します。

JSP ファイル格納先の詳細は、「4.1.2 JSP ファイルの格納先とパス」を参照してください。

補足事項

JSP ファイルの名称として使用できる文字は、半角英数字、半角アンダースコア (_), 半角ピリオド (.), および半角ハイフン (-) です。JSP ファイル名は URL の一部として使用されるため、そのことを考慮した名称にしてください。

6. 開発環境のユーザプロパティファイル (ucnp_user.properties) を設定します。

設定するプロパティの詳細は、「4.4.3 ユーザプロパティファイルの設定」を参照してください。

すべての手順が完了したら、次の個所で説明されている手順を実施してください。

- 「4.4.1 カスタム画面のビルドとデバッグ」
- 「4.4.2 カスタム画面の取り込み」

(1) 新規画面の表示例

検索機能を持つ新規画面の例を次に示します。

図 4-3 新規画面の例



(2) iPad 向け新規画面の注意事項

- 新規画面でログアウト後に画面を閉じる処理を実行させる JavaScript は、次のように記載してください。

```
var lWindow = window.open('', '_self');
lWindow.opener=window;
lWindow.close();
```

- 画面を閉じる処理を実行する画面が iframe などを使用して複数のフレームで構成されている場合は、最上位のフレームに実装してください。

4.3.2 ログアウト機能を追加する

ログアウト処理を追加した JSP ファイルを作成すると、ナビゲーション プラットフォームの画面以外の画面からのボタン操作で、ナビゲーション プラットフォームからログアウトできるようになります。必要に応じてログアウト処理を追加した JSP ファイルを作成してください。

！ 注意事項

新規画面にログアウト処理を追加する場合は、次のようにしてください。

- 新規画面に [ログアウト] ボタンを表示する。
- 新規画面から別画面として起動するナビゲーション プラットフォームの画面には [閉じる] ボタン (ログアウト処理を実行しない) を表示する。

ナビゲーション プラットフォームの画面からログアウト処理を実行すると、起動元の新規画面のセッションが無効となり、操作ができなくなります。

ナビゲーション プラットフォームの画面以外の画面から、ナビゲーション プラットフォームをログアウトする処理を追加する手順を次に示します。

1. ナビゲーション プラットフォームが提供している API を使用して、ログアウト用 JSP ファイルを作成します。

作成するファイルの記述形式を次に示します。

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="windows-31j"
%><%@ page import="jp.co.hitachi.soft.ucnp.base.common.LogoutActionUtil"
%><%
// ログアウトします。
LogoutActionUtil.logout(request, response);
%>
<html>
この部分に、HTMLタグおよびJavaScript関数を使用してログアウト後の処理を指定します。
</html>
```

ナビゲーション プラットフォームが提供しているログアウト用の API の詳細は、「6.6 LogoutActionUtil (ログアウト処理クラス)」を参照してください。

2. [ログアウト] ボタンをクリックしたら、作成したログアウト用 JSP ファイルに遷移するよう、新規画面の JSP ファイルをカスタマイズします。

外部システムの画面から、JavaScript 関数を使用してログアウト用 JSP ファイルに遷移するカスタマイズ例を次に示します。

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="Windows-31J" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<script type="text/javascript">
<!--
function toLogoutJSP() {
var url = "http://ホスト名:ポート番号/ucnpBase/ucnp/custom/ログアウト用JSPファイル名";
location.href = url;
}
//--></script>
</head>
<body>
<form>
<input type="button" value="ログアウト" onclick="toLogoutJSP()">
</form>
</body>
</html>
```

可変値について説明します。

- **ホスト名**
J2EE サーバマシンのホスト名または IP アドレスです。
- **ポート番号**
Web サーバのポート番号です。
- **ログアウト用 JSP ファイル名**
手順 1. で作成した JSP ファイルです。

なお、同一論理サーバ内で遷移する場合は、「http://ホスト名:ポート番号」の部分の指定は不要です。

3. 作成・編集した JSP ファイルを格納します。

JSP ファイル格納先の詳細は、「4.1.2 JSP ファイルの格納先とパス」を参照してください。

すべての手順が完了したら、次の個所で説明されている手順を実施してください。

- 「4.4.1 カスタム画面のビルドとデバッグ」
- 「4.4.2 カスタム画面の取り込み」

(1) ログアウト後にナビゲーション プラットフォームのログイン画面を表示するソース例

ログアウト後にナビゲーション プラットフォームのログイン画面を表示するソースの例を次に示します。

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="windows-31j"
%><%@ page import="jp.co.hitachi.soft.ucnp.base.common.LogoutActionUtil"
%><%
// ログアウトします。
LogoutActionUtil.logout(request, response);
// ログイン画面のURL。
String url = "ログイン画面のURL";
// ログイン画面に遷移します。
response.sendRedirect(url);
%>
```

(2) ログアウト後に画面を閉じるソース例

ログアウト後に画面を閉じるソースの例を次に示します。

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="windows-31j"
%><%@ page import="jp.co.hitachi.soft.ucnp.base.common.LogoutActionUtil"
%><%
// ログアウトします。
LogoutActionUtil.logout(request, response);
%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<!-- JavaScriptによるウィンドウクローズ --%>
<script type="text/javascript">
<!--
function closeWindow(){
var lWindow = window.open('', '_top');
lWindow.opener=window;
lWindow.close();
}
// -->
</script>
</head>
<!-- 自動的にウィンドウが閉じます --%>
<body onload="closeWindow()">
</body>
</html>
```

ただし、ナビゲーション プラットフォームにログインしていなかった場合は、ログアウト処理を実行しないで画面を閉じます。

4.3.3 任意のデータを受け取る機能を追加する

新規画面に任意のデータを送信するには、必ず ucnpUserData パラメタを使用してください。新規画面で受け取るデータが複数ある場合は、データを一つに結合して ucnpUserData パラメタに指定するようにしてください。

ucnpUserData パラメタは、GET メソッドまたは POST メソッドで送信できます。GET メソッドで URL の一部として送信する場合は、次の点に注意する必要があります。

- UTF-8 で URL エンコードした値を指定する。
- 使用する Web ブラウザの URL に使用できる最大文字数以内の値を指定する。

これらの条件が守られない場合の動作は保証されません。

なお、ucnpUserData パラメタは、ナビゲーション プラットフォームにログインして、最初に表示される新規画面に対してだけ有効です。ログイン後に新規画面からほかの画面に遷移したあとは、無効となります。

4.4 画面の開発後に必要な作業

カスタム画面を使用するには、まずカスタム画面のビルドとデバッグを実行し、そのあと編集環境と実行環境へ取り込みます。また、ユーザプロパティファイルを設定する必要があります。この節ではそれぞれの手順について説明します。

4.4.1 カスタム画面のビルドとデバッグ

1. プラグインと同じ手順で、カスタム画面をビルドします。

ビルド手順の詳細は、「3.6.1 プラグインのビルド手順」を参照してください。

2. J2EE アプリケーションをインポートして、開始します。

インポートと開始の手順は、「3.8 uCosminexus Navigation Developer の J2EE アプリケーションのインポートと開始」を参照してください。

3. Web ブラウザに次の URL を入力して、カスタム画面が表示できることを確認します。

```
http://localhost/ucnpBase/portal/action/Plain/portlet/ucnp/pane/-440b4408440g44Kw440r4408440X/layout_id/default/url/L2FkcG9ydGxldHMvdWNucC9qc3Avdmlldy9Sb290SWZyYW1lLmpzcA--?ucnpUserPage=JSPファイルのパス
```

補足事項

JSP ファイルのパスの基点は、ポータルプロジェクトのディレクトリになります。また、パスの先頭には「/」を指定してください。例えば、JSP ファイル (userpage.jsp) をポータルプロジェクトディレクトリ¥ucnp¥custom¥jsp¥userpage.jsp に配置した場合は、「/ucnp/custom/jsp/userpage.jsp」と指定してください。

4. J2EE アプリケーションをデバッグして、問題がないことを確認します。

デバッグ手順の詳細は、「3.11 プラグインのデバッグ」を参照してください。

4.4.2 カスタム画面の取り込み

1. 作成した J2EE アプリケーションのファイルを編集環境および実行環境に送付します。

開発環境から送付したファイルを編集環境および実行環境に適用する方法の詳細は、マニュアル「Cosminexus V9 ナビゲーション プラットフォーム 導入から運用まで」を参照してください。

2. 編集環境および実行環境のユーザプロパティファイル (ucnp_user.properties) を編集し、J2EE サーバを再起動します。

ユーザプロパティファイルのどのプロパティを編集するののかについては、「4.4.3 ユーザプロパティファイルの設定」を参照してください。

4.4.3 ユーザプロパティファイルの設定

ユーザプロパティファイル (ucnp_user.properties) のプロパティのうち、設定する必要があるプロパティは、メニュー領域をカスタマイズした場合と新規画面を作成した場合で次のように異なります。

各プロパティの詳細は、マニュアル「Cosminexus V9 ナビゲーション プラットフォーム 導入から運用まで」を参照してください。

メニュー領域をカスタマイズした場合

- ucnp.base.server.custom.menu プロパティ
作成した静的ファイルまたは JSP ファイルのパスを指定します。

静的ファイルのとき

パスの基点は、Web サーバのコンテンツ格納ディレクトリになります。例えば、HTML ファイル (menu.html) を Web サーバのコンテンツ格納ディレクトリ`*/custom` に配置した場合は、`[/custom/menu.html]` と指定してください。

JSP ファイルのとき

パスの基点は、ポータルプロジェクトのディレクトリになります。例えば、JSP ファイル (menu.jsp) をポータルプロジェクトディレクトリ`*/custom*/menu` に配置した場合は、`[/ucnpBase/custom/menu/menu.jsp]` と指定してください。

- `ucnp.base.server.search.enable` プロパティ (検索機能を使用するとき) `true` を指定します。
- `ucnp.base.server.search.accesscontrol.enable` プロパティ (検索機能を使用するとき) `true` を指定することを推奨します。

新規画面を作成した場合

- `ucnp.base.server.search.enable` プロパティ `true` を指定します。
- `ucnp.base.server.search.accesscontrol.enable` プロパティ `true` を指定することを推奨します。
- `ucnp.base.server.logoutbutton.display` プロパティ `false` を指定します。
- `ucnp.base.server.close.button.setting` プロパティ `close_only` を指定します。

なお、デフォルトのメニュー領域を使用しない場合は、`ucnp.base.client.menu.frame.visible` プロパティに `false` を指定すると、メニュー領域を非表示にすることもできます。

5

API リファレンス (プラグイン開発の場合)

この章では、プラグイン開発で使用する API について説明します。

5.1 API 一覧 (プラグイン開発の場合)

プラグインの開発で使用する API を次に示します。

表 5-1 API 一覧 (プラグイン開発)

分類	インタフェース名 またはクラス名	説明
全プラグイン共通	IUCNPSession	セッション情報を利用するためのインタフェースです。
入出力プラグイン	IioActionController	入出力プラグインのサーバ処理をカスタマイズするためのインタフェースです。このインタフェースは 09-50 より前のバージョンとの互換用です。09-50 以降のバージョンの開発では IioPluginController を使用してください。
	IioPluginController	入出力プラグインのサーバ処理を実装するためのインタフェースです。
	ParamConvertUtil	入出力パラメタを Map 形式に変換するユーティリティクラスです。
	UCNPPluginException	入出力プラグイン内で発生した例外を表すクラスです。
中断再開プラグイン	ISuspendActionController	中断情報を操作するためのインタフェースです。
	ISuspendInfo	保存・復元する中断情報を表すインタフェースです。
	SuspendInfoSerializeUtil	中断情報を直列化、または直列化された中断情報を復元するためのユーティリティクラスです。
	UCNPPluginException	中断再開プラグイン内で発生した例外を表すクラスです。

5.2 IIoActionController (入出力アクションコントローラインタフェース) (09-50 より前のバージョンとの互換用)

入出力プラグインのサーバ処理をカスタマイズするためのインタフェースです。

このインタフェースは 09-50 より前のバージョンとの互換用です。09-50 以降のバージョンの開発では IIoPluginController を使用してください。

パッケージ

```
package jp.co.hitachi.soft.ucnp.plugin.inputoutput.controller;
```

形式

```
public interface IIoActionController
```

メソッド

- execute メソッド
入出力プラグインのサーバ処理を実行します。

IIoActionController を継承したユーザ実装クラスのインスタンスについて

IIoActionController を継承したユーザ実装クラスのインスタンスは、execute()メソッドが呼び出されるたびに新規に作成されます。

5.2.1 execute メソッド

入出力プラグインのサーバ処理を実行します。

(1) 形式

```
public Map<String, Object> execute(HttpSession session, Map<String, Object> param);
```

(2) 引数

(a) session

現在のセッションが格納されます。このセッションに対して setAttribute()メソッドを実行する場合、setAttribute()メソッドの引数 name に次の名前を指定しないでください。

- "ucnp"で始まる名前
- uCosminexus Portal Framework の標準 API ポートレットが使用している名前
uCosminexus Portal Framework の標準 API ポートレットが使用している名前については、マニュアル「uCosminexus Portal Framework システム管理者ガイド」を参照してください。

また、URL のリクエストパラメタを利用することもできます。URL のリクエストパラメタを利用する場合は、次のキーで ucnpOptions パラメタを取得する必要があります。

ucnp.request.options キー

セッションから、URL に指定した ucnpOptions パラメタの値を取得するためのキーです。HttpSession.getAttribute()メソッドにこのキーを指定することで、URL デコードされた

ucnpOptions パラメタの値を取得できます。URL に ucnpOptions パラメタを指定していない場合は、null が返されます。

ucnp.request.options キーの指定例については、「5.3 IIoPluginController (サーバ処理実装インタフェース)」の inputFromNode メソッドの指定例を参照してください。

(b) param

クライアントから送信される次のキーが格納されます。param パラメタは更新しないでください。

表 5-2 param に格納されるキー

項番	キー名	値	説明
1	ucnp.current.params.map	Map<String,String> <ul style="list-style-type: none"> キー:1 文字以上の文字列 値:0 文字以上の文字列 	遷移元のガイドでユーザが入力したパラメタが、キーと値を対にした Map として渡されます。
2	ucnp.next.params.map	Map<String,String> <ul style="list-style-type: none"> キー:1 文字以上の文字列 値:0 文字以上の文字列 	遷移先のガイドに以前入力されていたパラメタが、キーと値を対にした Map として渡されます。
3	ucnp.button.type	String 次のどれか <ul style="list-style-type: none"> "show_previous_page" "show_next_page" "show_complete_page" "start" 	クリックされたボタンの種別、または遷移の種別を示します。 <ul style="list-style-type: none"> "show_previous_page" ・ [戻る] ボタン ・ 前のノードに直接遷移 "show_next_page" ・ [次へ] ボタン、または直接遷移で次のノードに遷移 ・ 遷移先のノードが存在しているプロセスノード、または [戻る] ボタンが表示になっているプロセスノードに遷移 "show_complete_page" ・ [完了] ボタン ・ 遷移先ノードが存在しない、かつ [戻る] ボタンが非表示になっているプロセスノードに遷移 "start" ・ ターミナルノード (開始) からプロセスノードに遷移 ・ ターミナルノード (開始) に接続されていない先頭のプロセスノードを表示
4	ucnp.current.node.name	String 0 文字以上の文字列	遷移元のガイドに対応したノードの名称が渡されます。 名称が設定されていないノードの場合は、空文字が渡されます。
5	ucnp.next.node.name	String 0 文字以上の文字列	遷移先のガイドに対応したノードの名称が渡されます。

項番	キー名	値	説明
5	ucnp.next.node.name	String 0文字以上の文字列	名称が設定されていないノードの場合は、空文字が渡されます。
6	ucnp.current.node.id	String 0文字以上の文字列	遷移元のガイドに対応したノード ID が渡されます。 ノード ID が設定されていない場合は、空文字が渡されます。
7	ucnp.next.node.id	String 0文字以上の文字列	遷移先のガイドに対応したノード ID が渡されます。 ノード ID が設定されていない場合は空文字が渡されます。
8	ucnp.flow.contents.id	String 1文字以上の文字列	選択中の業務コンテンツの業務コンテンツ ID が渡されます。
9	ucnp.flow.contents.name	String 1文字以上の文字列	選択中の業務コンテンツの業務名が渡されます。

ucnp.current.params.map キーおよび ucnp.next.params.map キーの詳細を次に示します。

ucnp.current.params.map キーの詳細

キー

- ターミナルノード (開始) から遷移した場合、および [次へ] ボタン、[工程名へ] ボタン、または [完了] ボタンをクリックした場合は、入力パラメタとなります。
入力パラメタは、入出力プラグイン用 XML ファイル (ioaction.xml) の <parameters type="input"> タグで定義されている各 <param> タグの name 属性の値です。
- [戻る] ボタンをクリックした場合は、出力パラメタとなります。
出力パラメタは、入出力プラグイン用 XML ファイル (ioaction.xml) の <parameters type="output"> タグで定義されている各 <param> タグの name 属性の値です。なお、出力パラメタを複数のガイドパーツに関連づけているときは、どのガイドパーツの値になるかは不定です。不定にならないようにするには、出力パラメタとガイドパーツを一对一で関連づける必要があります。
- パーツと関連づけられていない入力パラメタ、および出力パラメタのキーは Map 自体に含まれません。

値

入力パラメタ、および出力パラメタに関連づけられているパーツのキャッシュ値です。各パーツのキャッシュ値の詳細は、「(5) プラグインで保持されるキャッシュ値」を参照してください。
ただし、HTML パーツの入力パラメタの場合は、キャッシュ値とは異なる値が渡されます。渡されるのは HTML パーツの要素のうち、次の表に示す要素の name 属性と value 属性の値をエンコードした文字列です。エンコードした文字列をプラグインで処理できるようにするための API については、「5.7 ParamConvertUtil (入出力パラメタ変換ユーティリティクラス)」を参照してください。

表 5-3 HTML パーツの入力パラメタとして取得される要素

項番	取得される要素	要素の属性	説明
1	<INPUT>* 2 3 4 5	type="checkbox"	要素の checked 属性が true の場合に取得されます。
2		type="radio"	ラジオボタンは、name 属性でグループ化されています。このため、グループに所属するラジオボタンのうち、checked 属性が true (選択状態) の場合に取得されます。
3		type="text"	—
4		type="hidden"	—
5		type 属性の指定なし	—
6	<TEXTAREA>	—	—
7	<SELECT>	—	—

(凡例)

— : ありません。

注※

この表で示す type 属性以外の type 属性を持つ要素は取得されません。

ucnp.next.params.map キーの詳細

キー

- ターミナルノード (開始) から遷移した場合、および [次へ] ボタン、[工程名へ] ボタン、または [完了] ボタンをクリックした場合は、出力パラメタとなります。
- [戻る] ボタンをクリックした場合は、入力パラメタとなります。
- パーツと関連づけられていない入力パラメタ、および出力パラメタのキーは Map 自体に含まれません。

値

入力パラメタ、および出力パラメタに関連づけられているパーツのキャッシュ値です。各パーツのキャッシュ値の詳細は、「(5) プラグインで保持されるキャッシュ値」を参照してください。

ただし、HTML パーツの入力パラメタの場合は、キャッシュ値とは異なる値が渡されます。渡されるのは HTML パーツの要素のうち、表 5-3 に示す要素の name 属性と value 属性の値をエンコードした文字列です。エンコードした文字列をプラグインで処理できるようにするための API については、「5.7 ParamConvertUtil (入出力パラメタ変換ユーティリティクラス)」を参照してください。

(3) 戻り値

次の表に示す制御用パラメタを返します。

表 5-4 execute メソッドの戻り値

項番	キー名	値	説明
1	ucnp.next.params.map	Map<String,String> <ul style="list-style-type: none"> • キー: 1 文字以上の文字列 • 値: 0 文字以上の文字列 	遷移先のガイドに表示するパラメタを、キーと値を対にした Map として設定してください。

項番	キー名	値	説明
1	ucnp.next.params.map	Map<String,String> <ul style="list-style-type: none"> • キー: 1 文字以上の文字列 • 値: 0 文字以上の文字列 	このキーを指定しない場合、以前設定されたキャッシュ値が使用されます。各パーツのキャッシュ値については「(5) プラグインで保持されるキャッシュ値」を参照してください。
2	ucnp.error.message	String 0 文字以上の文字列	<p>ノード遷移を抑止したい場合、このキーを設定してください。このキーに設定した文字列がメッセージダイアログに表示され、ノード遷移が抑止されます。</p> <p>複数の入出力プラグインでこのキーを設定している場合、設定した文字列を改行で連結した値が一つのメッセージダイアログに表示されます。</p> <p>このキーを設定している場合、このキーと ucnp.error.params.list キーに設定しているデータ以外は無視されます。</p> <p>なお、入出力プラグインが出力するエラーメッセージが多数の改行を含む場合、JavaScript の Alert でエラーメッセージが表示されるので、メッセージダイアログの一部が画面に表示されなくなることがあります。そのため、エラーメッセージは画面に表示できる長さにしてください。</p>
3	ucnp.error.params.list	List<String> 1 文字以上の文字列	<p>エラーとなったパーツを強調表示したい場合、そのパーツに関連づけた入力パラメタまたは出力パラメタのリストを、このキーの値に設定してください。色の付いた枠で強調表示されます。</p> <p>強調表示時の枠の色のデフォルトは赤色です。この枠の色はユーザプロパティファイル (ucnp_user.properties) の ucnp.base.client.erroritem.emphasis.border.color プロパティで変更できます。</p> <p>ユーザプロパティファイルについては、マニュアル「Cosminexus V9 ナビゲーションプラットフォーム 導入から運用まで」を参照してください。</p> <p>このキーは、ucnp.error.message キーと一緒に戻り値の Map に設定してください。このキーを設定していても、ucnp.error.message キーを設定していない場合、強調表示されません。</p>

注

ucnp.next.params.map キーと ucnp.error.message キーを同時に指定すると、ucnp.error.message キーが有効になり、ucnp.next.params.map キーは無視されます。

ucnp.next.params.map キーの詳細を次に示します。

ucnp.next.params.map キーの詳細

キー

- ターミナルノード (開始) から遷移した場合、および [次へ] ボタン、[工程名へ] ボタン、または [完了] ボタンをクリックした場合は、出力パラメタを指定してください。
入出力プラグイン用 XML ファイル (ioaction.xml) の <parameters type="output"> の下位にある、<param> タグの name 属性に指定されている出力パラメタ名が、Map オブジェクトのキーに含まれないとき、その出力パラメタ名と関連づけられている遷移先のガイド領域のパーツにはキャッシュ値が使用されます。
- [戻る] ボタンをクリックした場合は、入力パラメタを指定してください。
入出力プラグイン用 XML ファイル (ioaction.xml) の <parameters type="input"> の下位にある、<param> タグの name 属性に指定されている入力パラメタ名が、Map オブジェクトのキーに含まれないとき、遷移元のガイド領域のパーツにはキャッシュ値が使用されます。
- ガイドパーツに複数の入力パラメタを関連づけている場合は、どの入力パラメタの値がキャッシュ値になるかは不定です。不定にならないようにするには、入力パラメタとガイドパーツを一対一で関連づけてください。

値

遷移後のガイド領域に表示するパラメタを指定してください。このパラメタには、パーツのキャッシュ値が設定されます。HTML パーツの場合、出力パラメタの値に動作保証外のタグや CSS が含まれているかはチェックされません。また、出力パラメタの値はサニタイズされません。

戻り値に null を指定した場合の処理を次に示します。

- インラインフレームパーツ、固定テキストパーツ、または HTML パーツの場合、空文字が使用されます。
- 画像パーツ、またはハイパーリンクパーツの場合、改行コード ("¥r¥n") が使用されます。
- これら以外のパーツの場合、何も実行されないで、すべて以前の値 (キャッシュ値) が使用されます。

(4) 例外

なし

(5) プラグインで保持されるキャッシュ値

プラグインで保持される各パーツのキャッシュ値を次の表に示します。

表 5-5 各パーツのキャッシュ値

項番	パーツの種類	キャッシュ値	初期値
1	固定テキスト	表示文字列	[属性の設定] 画面の [表示文字列] テキストボックスに指定した値
2	画像	URL とツールチップの属性値を改行で連結した文字列 (ツールチップが空文字の場合でも、改行コードはキャッシュ値に含まれます。)	[属性の設定] 画面の [URL] テキストボックスに指定した値と [ツールチップ] テキストボックスに指定した値を改行で連結した値
3	テキストボックス	テキストボックスに入力されている値	[属性の設定] 画面の [初期値] テキストボックスに指定した値

項番	パーツの種類	キャッシュ値	初期値
4	テキストエリア	テキストエリアに入力されている値	[属性の設定] 画面の [初期値] テキストエリアに指定した値
5	ラジオボタン	選択されているラジオボタンの [属性の設定] 画面の [値] テキストボックスに指定した値 (ラジオボタンが選択されていない場合は null)	null
6	チェックボックス	[true]: チェック状態の場合 [false]: チェック状態でない場合 (どちらの場合も、文字列で半角小文字)	[属性の設定] 画面の [初期値をチェック状態にする] チェックボックスの状態によって値が異なります。 <ul style="list-style-type: none"> • true: チェック状態とした場合 • false: チェック状態でない場合
7	ドロップダウン	選択されている [選択項目一覧] の値 (値にコンマや引用符 (")) が含まれている場合でも、CSV のエスケープはしないで、そのままの値を指定します。例えば、[神"奈川"] という値をキャッシュ値とする場合も、["神"奈川"] にはなりません。)	<ul style="list-style-type: none"> • [属性の設定] 画面の [選択項目一覧] の値のうち、[親の値] に対応する最初の値 (対応する [親の値] がいない場合は、空文字) • 親ドロップダウンがない場合は、[属性の設定] 画面の [選択項目一覧] の値のうち、[親の値] が空文字となっている最初の値
8	ハイパーリンク	アンカーの文字列と URL に指定した値を改行で連結した文字列 (ツールチップは文字列に含まれません。)	<ul style="list-style-type: none"> • アンカーの文字列と URL に指定した値を改行で連結した文字列 • [属性の設定] 画面の [アンカー文字列] テキストボックスと [URL] テキストボックスに指定した値
9	インラインフレーム	表示されているページの URL (フレーム枠の有無は含まれません。)	[属性の設定] 画面の [URL] テキストボックスに指定した値
10	HTML パーツ	表示されている HTML コンテンツ	[属性の設定] 画面の [HTML ソース文字列] テキストボックスに指定した値

全パーツ共通の注意事項

- 業務実行画面で業務を進めたり戻ったりして、ガイド領域を表示したときに表示される値の優先順位を次に示します。
 1. 入出力プラグインの出力値
 2. 業務実行画面で利用者が以前入力した値 (固定テキストパーツを除く)
 3. 各パーツのキャッシュ値の初期値 (表 5-5)
- キャッシュ値のライフサイクルは、業務を開始してから、業務を終了するか、または破棄するまでとなります。中断再開プラグインを利用して一時保存した業務を再開する場合は、キャッシュ値も引き継がれます。
- 同じプロセスノードに対応するガイド領域の、同じガイドパーツのキャッシュ値は一つの値が共有されます。そのため、合流を含むフローの場合、一度表示したプロセスノードを別のルートを通して再度表

示した場合、以前表示したときのキャッシュ値がそのまま利用されます。キャッシュ値を条件に応じて変更したいときには、入出力プラグインを利用して変更してください。

- キャッシュ値の改行は「`\r\n`」です。このメソッドの戻り値で指定するキャッシュ値に改行を含める場合、「`\r\n`」を改行として指定してください。

テキストボックスの注意事項

- キャッシュ値として、プラグインから改行を含む文字列が出力された場合、改行を取り除いた文字列がキャッシュ値となります。

ドロップダウンの注意事項

- 親ドロップダウンのキャッシュ値を指定した場合、親ドロップダウンのキャッシュ値とプラグインで指定された値が異なると、子ドロップダウンのキャッシュ値はリセットされます。リセットされた子ドロップダウンのキャッシュ値は、[属性の設定] 画面の [選択項目一覧] のうち、対応する親ドロップダウンの最初の値です。

例えば、[属性の設定] 画面で次のように設定されているドロップダウンに関して、親ドロップダウンのキャッシュ値が「kanagawa」、プラグインで指定された値が「tokyo」の場合、「tokyo」に対応する最初の値である「shibuyaku」が、リセットされた子ドロップダウンのキャッシュ値となります。



なお、業務実行画面で親ドロップダウンの選択状態を変更した場合も、同様に子ドロップダウンの値がリセットされます。

- キャッシュ値は、プラグインのパラメタの順序に従って反映されます。そのため、親子関係のあるドロップダウンのキャッシュ値を更新する場合は、子ドロップダウンには親ドロップダウンに関連づけたパラメタよりもあとにあるパラメタを関連づける必要があります。
- キャッシュ値に指定した値が、[属性の設定] 画面の [親の値] に対応する [選択項目一覧] に存在しない場合、指定したキャッシュ値は無視されます。

ハイパーリンクの注意事項

- キャッシュ値に改行が含まれない場合は、ラベルだけが指定されたものと見なされます。

5.3 IIoPluginController (サーバ処理実装インタフェース)

入出力プラグインのサーバ処理を実装するためのインタフェースです。

パッケージ

```
package jp.co.hitachi.soft.ucnp.plugin.inputoutput.controller;
```

形式

```
public interface IIoPluginController
```

メソッド

- inputFromNode メソッド
入出力プラグインを配置したノードからほかのノードに遷移するときに、遷移元ノードの情報を取得して入力する処理を実行します。
- outputToNode メソッド
入出力プラグインを配置したノードに遷移するときに、遷移先ノードに情報を出力する処理を実行します。

5.3.1 inputFromNode メソッド

入出力プラグインを配置したノードからほかのノードに遷移するときに、遷移元ノードの情報を取得して入力する処理を実行します。

ノード遷移時に、遷移元ノードの情報を取得して外部プログラムと連携するなどの、データ更新系の処理をこのメソッドで実装すると効果的です。

inputFromNode メソッドの注意事項

- 同じノードに複数の入出力プラグインを配置している場合、先に実行された inputFromNode メソッドの結果に関係なく、すべての入出力プラグインの inputFromNode メソッドが実行されます。したがって、入力パラメタのチェックが必要なプラグインを複数配置するときは、すべての入出力プラグインの inputFromNode メソッドの入力パラメタをチェックする必要があります。
- 次の場合、inputFromNode メソッドが複数回実行されます。
 - あとから実行した inputFromNode メソッドの実行結果がエラーのとき
 - 遷移先ノードに入出力プラグインが配置されていて、かつ、outputToNode メソッドの実行結果がエラーのとき

したがって、inputFromNode メソッドが複数回実行されても問題がないように、更新対象となるデータの整合性を保って実装してください。なお、メソッドの実行結果はエラーのほかに、正常、警告があります。

(1) 形式

```
public Map<String, Object> inputFromNode(HttpSession session, Map<String, Object> param);
```

(2) 引数

(a) session

現在のセッションが格納されます。このセッションに対して `setAttribute()` メソッドを実行する場合、`setAttribute()` メソッドの引数 `name` に次の名前を指定しないでください。

- "ucnp"で始まる名前
- uCosminexus Portal Framework の標準 API ポートレットが使用している名前
uCosminexus Portal Framework の標準 API ポートレットが使用している名前については、マニュアル「uCosminexus Portal Framework システム管理者ガイド」を参照してください。

また、URL のリクエストパラメタを利用することもできます。URL のリクエストパラメタを利用する場合は、`ucnp.request.options` キーで `ucnpOptions` パラメタを取得する必要があります。

`ucnpOptions` パラメタは、セッション単位または画面 ID 単位で取得できます。`ucnpOptions` パラメタの値は画面 ID 単位で取得することを推奨します。画面 ID 単位で取得する方法については、「(b) param」を参照してください。

セッション単位の取得を推奨しない理由

同一セッションでナビゲーション プラットフォームの画面を複数表示した場合、あとから操作した画面の `ucnpOptions` パラメタの値が HTTP セッションに上書きされ、前に操作した画面の `ucnpOptions` パラメタの情報が削除されるためです。

`ucnp.request.options` キー

セッションから、URL に指定した `ucnpOptions` パラメタの値を取得するためのキーです。`HttpSession.getAttribute()` メソッドにこのキーを指定することで、URL デコードされた `ucnpOptions` パラメタの値を取得できます。URL に `ucnpOptions` パラメタを指定していない場合は、`null` を返します。

- `ucnp.request.options` キーの指定例 (セッション単位) (非推奨)

```
public class IoPluginController implements IIoPluginController {
    /**
     * 入出力プラグインのサーバ処理を実行します。
     *
     * @param session
     *     現在のセッションです。
     * @param param
     *     クライアントから送信されるパラメタです。
     */
    public Map<String, Object> inputFromNode(HttpSession session, Map<String, Object>
    param){
        /* 現在のセッションからucnpOptions/パラメタの値を取得します。 */
        String ucnpOptions = (String) session.getAttribute("ucnp.request.options");
        if (ucnpOptions != null) {
            /* ucnpOptionsパラメタが指定されていたら、その値に対する処理を記述します。
            *
            *     . . .
            */
        }
    }
}
```

(b) param

入出力プラグインの実行時に、クライアントから送信される情報が格納されます。詳細を次の表に示します。`param` パラメタは更新しないでください。

表 5-6 param に格納されるキー (inputFromNode メソッド)

項番	キー名	値	説明
1	ucnp.current.params.map	Map<String,String> <ul style="list-style-type: none"> • キー:1 文字以上の文字列 • 値:0 文字以上の文字列 	<p>入出力プラグインの実行時に、次に示すキーと値を対にした Map として渡されます。</p> <ul style="list-style-type: none"> • キー 入出力プラグインの入力パラメタ名 • 値 入力パラメタに関連づけられたガイドのキャッシュ値 <p>ガイドに関連づけられていないパラメタはキーに含まれません。</p> <p>キャッシュ値については「5.2.1(5) プラグインで保持されるキャッシュ値」を参照してください。</p>
2	ucnp.button.type	String 次のどれか <ul style="list-style-type: none"> • "start" • "show_next_page" • "show_previous_page" • "show_complete_page" 	<p>クリックされたボタンの種別、または遷移の種別を示します。</p> <ul style="list-style-type: none"> • "start" <ul style="list-style-type: none"> ・ターミナルノード (開始) からプロセスノードに遷移 ・ターミナルノード (開始) に接続されていない先頭のプロセスノードを表示 • "show_next_page" <ul style="list-style-type: none"> ・ [次へ] ボタン, [工程名へ] ボタン, または直接遷移で次のノードに遷移 ・ 遷移先のノードが存在しているプロセスノード, または [戻る] ボタンが表示になっているプロセスノードに遷移 • "show_previous_page" <ul style="list-style-type: none"> ・ [戻る] ボタン ・ 前のノードに直接遷移 • "show_complete_page" <ul style="list-style-type: none"> ・ [完了] ボタン ・ 遷移先ノードが存在しない, かつ [戻る] ボタンが非表示になっているプロセスノードに遷移
3	ucnp.current.node.name	String 0 文字以上の文字列	<p>遷移元のガイドに対応したノードの名称が渡されます。</p> <p>名称が設定されていないノードの場合は、空文字が渡されます。</p>
4	ucnp.next.node.name	String 0 文字以上の文字列	<p>遷移先のガイドに対応したノードの名称が渡されます。</p> <p>名称が設定されていないノードの場合は、空文字が渡されます。</p>
5	ucnp.current.node.id	String 0 文字以上の文字列	<p>遷移元のガイドに対応したノード ID が渡されます。</p>

項番	キー名	値	説明
5	ucnp.current.node.id	String 0文字以上の文字列	ノード ID が設定されていない場合は、空文字が渡されます。
6	ucnp.next.node.id	String 0文字以上の文字列	遷移先のガイドに対応したノード ID が渡されます。 ノード ID が設定されていない場合は空文字が渡されます。
7	ucnp.flow.contents.id	String 1文字以上の文字列	選択中の業務コンテンツの業務コンテンツ ID が渡されます。
8	ucnp.flow.contents.name	String 1文字以上の文字列	選択中の業務コンテンツの業務名が渡されます。
9	ucnp.screen.id	String 1文字以上の文字列	業務コンテンツを実行している画面を一意に識別する ID (画面 ID) が渡されます。
10	ucnp.screen.ispreview	String 次のどちらか <ul style="list-style-type: none"> • "true" • "false" 	業務コンテンツを実行している画面がプレビュー画面かどうかを示します。 <ul style="list-style-type: none"> • "true" プレビュー画面 • "false" 業務実行画面
11	ucnp.options.param*	String	画面 ID 単位で取得した ucnpOptions パラメタの URL デコードした値を示します。 ucnpOptions パラメタが指定されていない場合、このキーは設定されません。

注※

URL に指定した ucnpOptions パラメタの値を画面 ID 単位で取得するためのキーです。引数 param にこのキーを指定することで、URL デコードされた ucnpOptions パラメタの値を取得できます。URL に ucnpOptions パラメタを指定していない場合は、null が返されます。

- ucnp.options.param キーの指定例 (画面 ID 単位) (推奨)

```
public class IoPluginController implements IIoPluginController {
    /**
     * 入出力プラグインのサーバ処理を実行します。
     *
     * @param session
     *     現在のセッションです。
     * @param param
     *     クライアントから送信されるパラメタです。
     */
    public Map<String, Object> inputFromNode(HttpSession session, Map<String, Object>
    param) {
        /* 引数からucnpOptions/パラメタの値を取得します。 */
        String ucnpOptions = (String) param.get("ucnp.options.param");
        if (ucnpOptions != null) {
            /* ucnpOptionsパラメタが指定されていたら、その値に対する処理を記述します。
        */
        }
    }
}
```

(3) 戻り値

入出力プラグインの実行結果が Map で返されます。詳細を次の表に示します。これ以外の値は無視されます。

表 5-7 inputFromNode メソッドの戻り値

項番	キー名	値	説明
1	ucnp.error.message	String 0文字以上の文字列	<p>inputFromNode メソッドを実行したあとに利用者にメッセージを表示したい場合、このキーを設定してください。</p> <p>遷移元のノードに関連づけられたすべての入出力プラグインの inputFromNode メソッドを実行したあとに、このキーに設定した文字列がメッセージダイアログに表示されます。</p> <p>このキーを設定するときは、次の点に注意してください。</p> <ul style="list-style-type: none"> 改行は「\n」で指定してください。 複数の入出力プラグインでこのキーを設定している場合、設定した文字列を改行で連結した値が一つのメッセージダイアログに表示されます。メッセージが長過ぎると一部が画面に表示されないときがあります。メッセージは画面に表示できる長さにしてください。
2	ucnp.error.type	String 次のどちらか <ul style="list-style-type: none"> "NG" "WARNING" 	<p>ucnp.error.message で設定したメッセージダイアログが表示されたあとに、ノード遷移を抑止するかどうかを設定してください。</p> <ul style="list-style-type: none"> "NG" 遷移が抑止される "WARNING" 遷移が抑止されない <p>次の場合このキーに指定した値は無視されます。</p> <ul style="list-style-type: none"> ucnp.error.message に値を設定していない ucnp.error.message に null を指定している <p>次の場合"NG"が仮定されます。</p> <ul style="list-style-type: none"> "NG"および"WARNING"以外の文字列を指定した null を指定した 値を指定していない <p>複数の入出力プラグインの実行時は、設定によって次のように動作が異なります。</p> <ul style="list-style-type: none"> すべての入出力プラグインで"WARNING"を指定している 遷移が抑止されない

項番	キー名	値	説明
2	ucnp.error.type	String 次のどちらか <ul style="list-style-type: none"> "NG" "WARNING" 	<ul style="list-style-type: none"> 入出力プラグインに"WARNING"が一つも指定されていない 遷移が抑止される
3	ucnp.error.params.list	List<String> 1文字以上の文字列	<p>ノード遷移が抑止されているガイドパーツを強調表示したい場合、そのパーツに関連づけた入力パラメタのリストを、このキーの値に設定してください。色の付いた枠で強調表示されます。</p> <p>強調表示時の枠の色のデフォルトは赤色です。この枠の色はユーザプロパティファイル (ucnp_user.properties) の ucnp.base.client.erroritem.emphasis.border.color プロパティで変更できます。</p> <p>ユーザプロパティファイルについては、マニュアル「Cosminexus V9 ナビゲーション プラットフォーム 導入から運用まで」を参照してください。</p> <p>このキーは、ucnp.error.message キーと一緒に戻り値の Map に設定してください。ucnp.error.message キーを設定していない場合は、このキーを設定しても、強調表示されません。</p> <p>ノード遷移が抑止されていないときは、このキーに指定した値は無視されます。</p>

inputFromNode メソッドの戻り値と遷移元ノードに関連づけた入出力プラグイン実行後の動作を次の表に示します。

表 5-8 戻り値と遷移元ノードに関連づけた入出力プラグイン実行後の動作

項番	ucnp.error.message の指定値	ucnp.error.type の指定値	ucnp.error.params.list の指定値	メソッドの実行結果	ダイアログの表示の有無	ノード遷移の有無	強調表示の有無
1	指定なし、または null	NG	指定あり (null 以外)	正常	無	無	無
2			指定なし、または null	正常	無	無	無
3		WARNING	指定あり (null 以外)	正常	無	無	無
4			指定なし、または null	正常	無	無	無
5		NG および WARNING 以外	指定あり (null 以外)	正常	無	無	無
6			指定なし、または null	正常	無	無	無

項番	ucnp.error.message の指定値	ucnp.error.type の指定値	ucnp.error.params.list の指定値	メソッドの実行結果	ダイアログの表示の有無	ノード遷移の有無	強調表示の有無	
7	指定あり (null 以外)	NG	指定あり (null 以外)	エラー	有	有	有	
8			指定なし, または null	エラー	有	有	無	
9		WARNING	指定あり (null 以外)	警告	有	無	無	
10			指定なし, または null	警告	有	無	無	
11		NG および WARNING 以外	指定あり (null 以外)	指定あり (null 以外)	エラー	有	有	有
12				指定なし, または null	エラー	有	有	無

(4) 例外

なし

5.3.2 outputToNode メソッド

入出力プラグインを配置したノードに遷移するとき、遷移先ノードに情報を出力する処理を実行します。

ノード遷移時に、遷移先ノードに外部プログラムから取得したデータを反映するなど、データ参照系の処理をこのメソッドで実装すると効果的です。

合流ノードが遷移した経路ごとに入出力プラグインの処理を切り替えたい場合は、入出力プラグインの outputToNode メソッドで遷移した経路を遷移元ノードから判断し、処理を切り替えるようにしてください。

outputToNode メソッドの注意事項

- 遷移元ノードに入出力プラグインが配置されていて、かつ、先に実行した inputFromNode メソッドの実行結果がエラーの場合、すべての入出力プラグインの outputToNode メソッドは実行されません。
- 同じノードに複数の入出力プラグインが配置されている場合、先に実行した outputToNode メソッドの実行結果がエラーのときは、以降の入出力プラグインは実行されません。

なお、メソッドの実行結果はエラーのほかに、正常、警告があります。

(1) 形式

```
public Map<String, Object> outputToNode(HttpSession session, Map<String, Object> param);
```

(2) 引数

(a) session

現在のセッションが格納されます。このセッションに対して setAttribute()メソッドを実行する場合、setAttribute()メソッドの引数 name に次の名前を指定しないでください。

- "ucnp"で始まる名前
- uCosminexus Portal Framework の標準 API ポートレットが使用している名前
uCosminexus Portal Framework の標準 API ポートレットが使用している名前については、マニュアル「uCosminexus Portal Framework システム管理者ガイド」を参照してください。

また、URL のリクエストパラメタを利用することもできます。URL のリクエストパラメタを利用する場合は、ucnp.request.options キーで ucnpOptions パラメタを取得する必要があります。

ucnpOptions パラメタは、セッション単位または画面 ID 単位で取得できます。ucnpOptions パラメタの値は画面 ID 単位で取得することを推奨します。画面 ID 単位で取得する方法については、「(b) param」を参照してください。

セッション単位の取得を推奨しない理由

同一セッションでナビゲーション プラットフォームの画面を複数表示した場合、あとから操作した画面の ucnpOptions パラメタの値が HTTP セッションに上書きされ、前に操作した画面の ucnpOptions パラメタの情報が削除されるためです。

ucnp.request.options キー

セッションから、URL に指定した ucnpOptions パラメタの値を取得するためのキーです。HttpSession.getAttribute()メソッドにこのキーを指定することで、URL デコードされた ucnpOptions パラメタの値を取得できます。URL に ucnpOptions パラメタが指定されていない場合は、null を返します。

ucnp.request.options キーの指定例については、「5.3 IIoPluginController (サーバ処理実装インタフェース)」の inputFromNode メソッドの指定例を参照してください。

(b) param

入出力プラグインの実行時に、クライアントから送信される情報が格納されます。詳細を次の表に示します。param パラメタは更新しないでください。

表 5-9 param に格納されるキー (outputToNode メソッド)

項番	キー名	値	説明
1	ucnp.next.params.map	Map<String,String>	<p>入出力プラグインの実行時に、次に示すキーと値を対にした Map として渡されます。</p> <ul style="list-style-type: none"> • キー 入出力プラグインの出力パラメタ名 • 値 出力パラメタに関連づけられたガイドの、遷移前の時点のキャッシュ値 <p>ガイドに関連づけられていないパラメタはキーに含めません。</p> <p>キャッシュ値については「5.2.1(5) プラグインで保持されるキャッシュ値」を参照してください。</p>
2	ucnp.button.type	<p>String</p> <p>次のどれか</p> <ul style="list-style-type: none"> • "start" • "show_next_page" 	<p>クリックされたボタンの種別、または遷移の種別を示します。</p> <ul style="list-style-type: none"> • "start" ・ターミナルノード (開始) からプロセスノードに遷移

5 API リファレンス (プラグイン開発の場合)

項番	キー名	値	説明
2	ucnp.button.type	<ul style="list-style-type: none"> • "show_previous_page" • "show_complete_page" 	<ul style="list-style-type: none"> ・ターミナルノード (開始) に接続されていない先頭のプロセスノードを表示 • "show_next_page" <ul style="list-style-type: none"> ・ [次へ] ボタン, [工程名へ] ボタン, または直接遷移で次のノードに遷移 ・ 遷移先のノードが存在しているプロセスノード, または [戻る] ボタンが表示になっているプロセスノードに遷移 • "show_previous_page" <ul style="list-style-type: none"> ・ [戻る] ボタン ・ 前のノードに直接遷移 • "show_complete_page" <ul style="list-style-type: none"> ・ [完了] ボタン ・ 遷移先ノードが存在しない, かつ [戻る] ボタンが非表示になっているプロセスノードに遷移
3	ucnp.current.node.name	String 0 文字以上の文字列	遷移元のガイドに対応したノードの名称が渡されます。 名称が設定されていないノードの場合は, 空文字が渡されます。
4	ucnp.next.node.name	String 0 文字以上の文字列	遷移先のガイドに対応したノードの名称が渡されます。 名称が設定されていないノードの場合は, 空文字が渡されます。
5	ucnp.current.node.id	String 0 文字以上の文字列	遷移元のガイドに対応したノード ID が渡されます。 ノード ID が設定されていない場合は, 空文字が渡されます。
6	ucnp.next.node.id	String 0 文字以上の文字列	遷移先のガイドに対応したノード ID が渡されます。 ノード ID が設定されていない場合は空文字が渡されます。
7	ucnp.flow.contents.id	String 1 文字以上の文字列	選択中の業務コンテンツの業務コンテンツ ID が渡されます。
8	ucnp.flow.contents.name	String 1 文字以上の文字列	選択中の業務コンテンツの業務名が渡されます。
9	ucnp.screen.id	String 1 文字以上の文字列	業務コンテンツを実行している画面を一意に識別する ID (画面 ID) が渡されます。
10	ucnp.screen.ispreview	String 次のどちらか <ul style="list-style-type: none"> • "true" • "false" 	業務コンテンツを実行している画面がプレビュー画面かどうかを示します。 <ul style="list-style-type: none"> • "true" プレビュー画面 • "false"

項番	キー名	値	説明
10	ucnp.screen.ispreview	String 次のどちらか <ul style="list-style-type: none"> • "true" • "false" 	業務実行画面
11	ucnp.options.param*	String	画面 ID 単位で取得した ucnpOptions パラメタの URL デコードした値を示します。 ucnpOptions パラメタが指定されていない場合、このキーは設定されません。

注※

URL に指定した ucnpOptions パラメタの値を画面 ID 単位で取得するためのキーです。引数 param にこのキーを指定することで、URL デコードされた ucnpOptions パラメタの値を取得できます。URL に ucnpOptions パラメタを指定していない場合は、null が返されます。ucnp.options.param キーの指定例については、「5.3 IIoPluginController (サーバ処理実装インタフェース)」の inputFromNode メソッドの指定例を参照してください。

(3) 戻り値

入出力プラグインの実行結果が Map で返されます。詳細を次の表に示します。これ以外の値は無視されます。

表 5-10 outputToNode メソッドの戻り値

項番	キー名	値	説明
1	ucnp.next.params.map	Map<String, String>	入出力プラグインの実行時に、次に示すキーと値を対にした Map として渡されます。 <ul style="list-style-type: none"> • キー 入出力プラグインの出力パラメタ名 • 値 出力パラメタに関連づけられているガイドの、遷移前の時点のキャッシュ値
2	ucnp.error.message	String	outputToNode メソッドを実行したあとに利用者にメッセージを表示したい場合、このキーを設定してください。 遷移先ノードに関連づけられている入出力プラグインが、次のどちらかの結果で終了したとき、このキーに設定した文字列がメッセージダイアログに表示されます。 <ul style="list-style-type: none"> • どれかの入出力プラグインの outputToNode メソッドの実行結果の種別がエラー • すべての入出力プラグインの outputToNode メソッドの実行結果種別が正常、または警告 このキーを設定するときは、次の点に注意してください。 <ul style="list-style-type: none"> • 改行は「\n」で指定してください。

項番	キー名	値	説明
2	ucnp.error.message	String	<ul style="list-style-type: none"> 複数の入出力プラグインでこのキーを設定している場合、設定した文字列を改行で連結した値が一つのメッセージダイアログに表示されます。メッセージが長すぎると一部が画面に表示されないときがあります。メッセージは画面に表示できる長さにしてください。
3	ucnp.error.type	String 次のどちらか <ul style="list-style-type: none"> "NG" "WARNING" 	<p>ucnp.error.message に指定したメッセージがダイアログに表示されたあとに、ノード遷移を抑制するかどうかを設定してください。指定できる文字列を次に示します。</p> <ul style="list-style-type: none"> "NG" 遷移が抑制される "WARNING" 遷移が抑制されない <p>次の場合、このキーに指定した値は無視されます。</p> <ul style="list-style-type: none"> ucnp.error.message に値を設定していない ucnp.error.message に null を指定している <p>次の場合、"NG"が仮定されます。</p> <ul style="list-style-type: none"> "NG"および"WARNING"以外の文字列を指定した null を指定した 値を指定していない <p>複数の入出力プラグインの実行時は、設定によって次のように動作が異なります。</p> <ul style="list-style-type: none"> すべての入出力プラグインで"WARNING"を指定している 遷移が抑制されない 入出力プラグインに"WARNING"を一つも指定していない 遷移が抑制される

outputToNode メソッドの戻り値と遷移先ノードに関連づけた入出力プラグイン実行後の動作を次の表に示します。

表 5-11 戻り値と遷移先ノードに関連づけた入出力プラグイン実行後の動作

項番	ucnp.error.message の指定値	ucnp.error.type の指定値	メソッドの実行結果	ダイアログの表示の有無	ノード遷移の有無
1	指定なし, または null	NG	正常	無	無
2		WARNING	正常	無	無
3		NG および WARNING 以外	正常	無	無

項番	ucnp.error.message の指定値	ucnp.error.type の指定値	メソッドの実行結果	ダイアログの表示の有無	ノード遷移の有無
4	指定あり (null 以外)	NG	エラー	有	有
5		WARNING	警告	有	無
6		NG および WARNING 以外	エラー	有	有

(4) 例外

なし

5.3.3 プレビュー時のプラグイン処理

業務実行画面だけでなく、プレビュー時にノード遷移した場合でも、入出力プラグインが実行されます。プレビュー時に入出力プラグインを実行させたくないときは、inputFromNode/outputToNode メソッドを実装する際に、param パラメタから、プラグインを実行した画面がプレビュー画面かどうかを取得して、プラグインの処理の実行可否を判断できるようにしてください。

実装例

```
public Map<String, Object> outputToNode(HttpSession session, Map<String, Object> param) {
    boolean isPreview = Boolean.valueOf((String) param.get("ucnp.screen.ispreview"));
    if (!isPreview) {
        // プレビュー画面ではない場合だけプラグインの処理を実行する
    }
}
```

5.4 IUCNPSession (セッション情報利用インタフェース)

セッション情報を利用するためのインタフェースです。

このインスタンスは、サーバ処理実装インタフェースの `inputFromNode()` メソッドまたは `outputToNode()` メソッドの `session` パラメタから、次に示すコードで取得できます。なお、入出力アクションコントローラインタフェース (09-50 より前のバージョンとの互換用) の `execute()` メソッドの `session` パラメタからも取得できます。

コードを次に示します。

```
IUCNPSession ucnpSession = (IUCNPSession)session.getAttribute("ucnp.session");
```

パッケージ

```
package jp.co.hitachi.soft.ucnp.base.portlet;
```

形式

```
public interface IUCNPSession
```

メソッド

- `getLoginId` メソッド
ログイン中のユーザ ID を返します。

5.4.1 `getLoginId` メソッド

ログイン中のユーザ ID を返します。

(1) 形式

```
public String getLoginId();
```

(2) 引数

なし

(3) 戻り値

ユーザ ID

(4) 例外

なし

5.5 ISuspendActionController (中断再開アクションコントローラインタフェース)

中断再開プラグインで、中断情報を操作するためのインタフェースです。

パッケージ

```
package jp.co.hitachi.soft.ucnp.plugin.suspend.controller;
```

形式

```
public Interface ISuspendActionController
```

メソッド

- save メソッド
中断情報を業務コンテンツ ID に関連づけて保存します。
- load メソッド
業務コンテンツ ID に関連づいた中断情報を読み込みます。
- contains メソッド
業務コンテンツ ID に関連づいた中断情報があるかどうかを確認します。
- delete メソッド
業務コンテンツ ID に関連づいた中断情報を削除します。
- deleteAll メソッド
業務コンテンツ ID に関連づいた中断情報をすべて削除します。

ISuspendActionController を継承したユーザ実装クラスのインスタンスについて

ISuspendActionController を継承したユーザ実装クラスのインスタンスは、初回のメソッド呼び出し時に一度だけ作成されます。各メソッドの呼び出しは、同じインスタンスに対して実施します。

5.5.1 save メソッド

中断再開プラグインで、中断情報を保存するメソッドです。パラメタに指定された業務コンテンツ ID に関連づけて中断情報を保存します。

このメソッドが UCNPPluginException 例外をスローした場合、ナビゲーション プラットフォームは UCNPPluginException 例外のメッセージをダイアログに表示します。ダイアログに表示するメッセージにはメッセージ ID は付きません。

このメソッドの処理は、ひな形プラグイン作成コマンドで作成した中断再開アクションコントローラクラスに実装します。

(1) 形式

```
public void save(String contentId, ISuspendInfo suspendInfo, HttpSession session)
    throws UCNPPluginException;
```

(2) 引数

(a) contentId

業務コンテンツ ID が表示されます。

(b) suspendInfo

[一時保存] ボタンをクリックしたときに選択されていた業務コンテンツの中断情報が格納されます。

(c) session

現在のセッションが格納されます。このセッションに対して setAttribute()メソッドを実行する場合、setAttribute()メソッドの引数 name に次の名前を指定しないでください。

- "ucnp"で始まる名前
- uCosminexus Portal Framework の標準 API ポートレットが使用している名前
uCosminexus Portal Framework の標準 API ポートレットが使用している名前については、マニュアル「uCosminexus Portal Framework システム管理者ガイド」を参照してください。

また、URL のリクエストパラメタを取得して利用することもできます。URL のリクエストパラメタを利用する場合は、次のキーで ucnpOptions パラメタを取得する必要があります。

ucnp.request.options キー

セッションから、URL に指定した ucnpOptions パラメタの値を取得するためのキーです。HttpSession.getAttribute()メソッドにこのキーを指定することで、URL デコードされた ucnpOptions パラメタの値を取得できます。URL に ucnpOptions パラメタが指定されていない場合は、null を返します。

ucnp.request.options キーの指定例については、「5.3 IIoPluginController (サーバ処理実装インタフェース)」の inputFromNode メソッドの指定例を参照してください。

(3) 戻り値

なし

(4) 例外

UCNPPluginException - 中断情報の保存処理などでエラーが発生した場合

ナビゲーション プラットフォームは、このメソッドがスローした UCNPPluginException 例外の詳細メッセージを、業務実行画面のダイアログに表示します。スローする UCNPPluginException 例外には、どのようなエラーが起こったか、またどのように対処すればよいか、利用者にわかるメッセージを設定してください。なお、中断再開プラグインが出力するエラーメッセージが多数の改行を含む場合、JavaScript の Alert でエラーメッセージを表示するので、メッセージダイアログの一部が画面に表示されなくなる場合があります。そのため、エラーメッセージは画面に表示できる長さにしてください。

(5) 呼び出しタイミング

ナビゲーション プラットフォームが、このメソッドを呼び出すタイミングを次に示します。

- [一時保存] ボタンをクリックしたとき

5.5.2 load メソッド

中断再開プラグインで、中断情報を読み込んで復元するメソッドです。パラメタに指定された業務コンテンツ ID に関連づいた中断情報を読み込みます。

contains()メソッドの戻り値が true の場合だけ、このメソッドは呼び出されます。したがって、このメソッドの戻り値が null の場合、ダイアログにプラグインエラーのメッセージを表示します。中断情報がない場合は、UCNPPluginException 例外にメッセージを設定してスローしてください。

このメソッドが UCNPPluginException 例外をスローした場合、ナビゲーション プラットフォームは UCNPPluginException 例外のメッセージをダイアログに表示します。ダイアログに表示するメッセージにはメッセージ ID は付きません。

このメソッドの戻り値が不正な場合、または、このメソッドが UCNPPluginException 例外をスローした場合は、メッセージをダイアログに表示したあと、業務を最初から開始します。

(1) 形式

```
public ISuspendInfo load(String contentId, HttpSession session)
    throws UCNPPluginException;
```

(2) 引数

(a) contentId

業務コンテンツ ID が表示されます。

(b) session

現在のセッションが格納されます。このセッションに対して setAttribute()メソッドを実行する場合、setAttribute()メソッドの引数 name に次の名前を指定しないでください。

- "ucnp"で始まる名前
- uCosminexus Portal Framework の標準 API ポートレットが使用している名前
uCosminexus Portal Framework の標準 API ポートレットが使用している名前については、マニュアル「uCosminexus Portal Framework システム管理者ガイド」を参照してください。

また、URL のリクエストパラメタを取得して利用することもできます。URL のリクエストパラメタを利用する場合は、次のキーで ucnpOptions パラメタを取得する必要があります。

ucnp.request.options キー

セッションから、URL に指定した ucnpOptions パラメタの値を取得するためのキーです。

HttpSession.getAttribute()メソッドにこのキーを指定することで、URL デコードされた ucnpOptions パラメタの値を取得できます。URL に ucnpOptions パラメタが指定されていない場合は、null を返します。

ucnp.request.options キーの指定例については、「5.3 IIoPluginController (サーバ処理実装インタフェース)」の inputFromNode メソッドの指定例を参照してください。

(3) 戻り値

業務コンテンツ ID に関連づいた中断情報

(4) 例外

UCNPPluginException - 中断情報の読み込み処理などでエラーが発生した場合

ナビゲーション プラットフォームは、このメソッドがスローした UCNPPluginException 例外の詳細メッセージを、業務実行画面のダイアログに表示します。スローする UCNPPluginException 例外には、どのようなエラーが起こったか、またどのように対処すればよいか、利用者にわかるメッセージを設定してください。なお、中断再開プラグインが出力するエラーメッセージが多数の改行を含む場合、JavaScript の Alert でエラーメッセージを表示するので、メッセージダイアログの一部が画面に表示されなくなる場合があります。そのため、エラーメッセージは画面に表示できる長さにしてください。

(5) 呼び出しタイミング

ナビゲーション プラットフォームがこのメソッドを呼び出すタイミングを次に示します。

- 中断再開アクションコントローラインタフェースを継承した実装クラスの contains()メソッドが呼び出されたあと、その contains()メソッドが true を返したとき
contains()メソッドが false を返したとき、および UCNPPluginException 例外をスローしたときは呼び出されません。contains()メソッドが呼び出されるタイミングについては、「5.5 ISuspendActionController (中断再開アクションコントローラインタフェース)」の contains メソッドの説明を参照してください。

5.5.3 contains メソッド

中断再開プラグインで、中断情報の存在を確認するメソッドです。パラメタに指定された業務コンテンツ ID に関連づいた中断情報があるかどうかを確認します。関連した中断情報がある場合は true、ない場合は false を返します。

このメソッドが UCNPPluginException 例外をスローした場合、ナビゲーション プラットフォームは UCNPPluginException 例外のメッセージをダイアログに表示します。ダイアログに表示するメッセージにはメッセージ ID は付きません。

このメソッドの処理は、ひな形プラグイン作成コマンドで作成した中断再開アクションコントローラクラスに実装します。

(1) 形式

```
public boolean contains(String contentId, HttpSession session)
    throws UCNPPluginException;
```

(2) 引数

(a) contentId

業務コンテンツ ID が表示されます。

(b) session

現在のセッションが格納されます。このセッションに対して setAttribute()メソッドを実行する場合、setAttribute()メソッドの引数 name に次の名前を指定しないでください。

- "ucnp"で始まる名前
- uCosminexus Portal Framework の標準 API ポートレットが使用している名前

uCosminexus Portal Framework の標準 API ポートレットが使用している名前については、マニュアル「uCosminexus Portal Framework システム管理者ガイド」を参照してください。

また、URL のリクエストパラメタを取得して利用することもできます。URL のリクエストパラメタを利用する場合は、次のキーで ucnpOptions パラメタを取得する必要があります。

ucnp.request.options キー

セッションから、URL に指定した ucnpOptions パラメタの値を取得するためのキーです。HttpSession.getAttribute()メソッドにこのキーを指定することで、URL デコードされた ucnpOptions パラメタの値を取得できます。URL に ucnpOptions パラメタが指定されていない場合は、null を返します。

ucnp.request.options キーの指定例については、「5.3 IIoPluginController (サーバ処理実装インタフェース)」の inputFromNode メソッドの指定例を参照してください。

(3) 戻り値

- true : 業務コンテンツ ID に関連づいた中断情報がある
- false : 業務コンテンツ ID に関連づいた中断情報がない

(4) 例外

UCNPPluginException - 中断情報の読み込み処理などでエラーが発生した場合

ナビゲーション プラットフォームは、このメソッドがスローした UCNPPluginException 例外の詳細メッセージを、業務実行画面のダイアログに表示します。スローする UCNPPluginException 例外には、どのようなエラーが起こったか、またどのように対処すればよいか、利用者にわかるメッセージを設定してください。なお、中断再開プラグインが出力するエラーメッセージが多数の改行を含む場合、JavaScript の Alert でエラーメッセージを表示するので、メッセージダイアログの一部が画面に表示されなくなる場合があります。そのため、エラーメッセージは画面に表示できる長さにしてください。

(5) 呼び出しタイミング

ナビゲーション プラットフォームが、このメソッドを呼び出すタイミングを次に示します。

- ユーザが業務実行画面で業務コンテンツを選択したとき
- カスタム画面用の JavaScript 関数を呼び出したとき (詳細は「4.2.1 ナビゲーション プラットフォームで提供している JavaScript 関数」を参照)
- URL を直接指定し、業務コンテンツ ID を指定するパラメタ (contentId) で、業務コンテンツが選択されたとき

5.5.4 delete メソッド

中断再開プラグインで、中断情報を削除するメソッドです。パラメタに指定された業務コンテンツ ID に関連づいた中断情報を削除します。

このメソッドが UCNPPluginException 例外をスローした場合、ナビゲーション プラットフォームは UCNPPluginException 例外のメッセージをダイアログに表示します。ダイアログに表示するメッセージにはメッセージ ID は付きません。

このメソッドの処理は、ひな形プラグイン作成コマンドで作成した中断再開アクションコントローラクラスに実装します。

(1) 形式

```
public void delete(String contentId, HttpSession session)
                    throws UCNPPuginException;
```

(2) 引数

(a) contentId

業務コンテンツ ID が表示されます。

(b) session

現在のセッションが格納されます。このセッションに対して setAttribute()メソッドを実行する場合、setAttribute()メソッドの引数 name に次の名前を指定しないでください。

- "ucnp"で始まる名前
- uCosminexus Portal Framework の標準 API ポートレットが使用している名前
uCosminexus Portal Framework の標準 API ポートレットが使用している名前については、マニュアル「uCosminexus Portal Framework システム管理者ガイド」を参照してください。

また、URL のリクエストパラメータを取得して利用することもできます。URL のリクエストパラメータを利用する場合は、次のキーで ucnpOptions パラメータを取得する必要があります。

ucnp.request.options キー

セッションから、URL に指定した ucnpOptions パラメータの値を取得するためのキーです。

HttpSession.getAttribute()メソッドにこのキーを指定することで、URL デコードされた ucnpOptions パラメータの値を取得できます。URL に ucnpOptions パラメータが指定されていない場合は、null を返します。

ucnp.request.options キーの指定例については、「5.3 IIoPluginController (サーバ処理実装インタフェース)」の inputFromNode メソッドの指定例を参照してください。

(3) 戻り値

なし

(4) 例外

UCNPPuginException - 中断情報の削除処理などでエラーが発生した場合

ナビゲーション プラットフォームは、このメソッドがスローした UCNPPuginException 例外の詳細メッセージを、業務実行画面のダイアログに表示します。スローする UCNPPuginException 例外には、どのようなエラーが起こったか、またどのように対処すればよいか、利用者にわかるメッセージを設定してください。なお、中断再開プラグインが出力するエラーメッセージが多数の改行を含む場合、JavaScript の Alert でエラーメッセージを表示するので、メッセージダイアログの一部が画面に表示されなくなる場合があります。そのため、エラーメッセージは画面に表示できる長さにしてください。

(5) 呼び出しタイミング

ナビゲーション プラットフォームがこのメソッドを呼び出すタイミングを次に示します。

- 利用者が、業務実行画面で [完了] ボタンをクリックしたとき
- 中断再開アクションコントローラインタフェースを継承した実装クラスの load メソッドが呼び出されたあと、その load メソッドが返した中断情報のチェックでエラーが発生したとき

load メソッドが返した中断情報のチェックでは、中断情報を保存した時点から業務コンテンツが更新されているかどうかをチェックし、更新されている場合にエラーとなります。load メソッドが呼び出されるタイミングについては、「5.5 ISuspendActionController (中断再開アクションコントローラインタフェース)」の load メソッドの説明を参照してください。

注意事項

この機能は、中断再開プラグインがシステムで有効な場合だけ呼び出されます。中断再開プラグインがシステムで無効な場合は、上記のタイミングでも呼び出されません。

5.5.5 deleteAll メソッド

中断再開プラグインで、業務コンテンツに関連づいたすべての中断情報を削除するメソッドです。パラメータに指定された業務コンテンツ ID に関連づいたすべての中断情報を削除します。

業務コンテンツを作成するユーザが業務コンテンツを編集または削除した場合に、関連づいた中断情報をすべて削除するためにこのメソッドを呼び出します。業務コンテンツが編集または削除されると、すでに保存されている中断情報は無効になるため、関連するすべての中断情報を削除してください。

このメソッドが UCNPluginException 例外をスローした場合、ナビゲーション プラットフォームは UCNPluginException 例外のメッセージをダイアログに表示します。ダイアログに表示するメッセージにはメッセージ ID は付きません。

このメソッドの処理は、ひな形プラグイン作成コマンドで作成した中断再開アクションコントローラクラスに実装します。

(1) 形式

```
public void deleteAll(String contentId, HttpSession session)
                    throws UCNPluginException;
```

(2) 引数

(a) contentId

業務コンテンツ ID が表示されます。

(b) session

現在のセッションが格納されます。このセッションに対して setAttribute()メソッドを実行する場合、setAttribute()メソッドの引数 name に次の名前を指定しないでください。

- "ucnp"で始まる名前
- uCosminexus Portal Framework の標準 API ポートレットが使用している名前
uCosminexus Portal Framework の標準 API ポートレットが使用している名前については、マニュアル「uCosminexus Portal Framework システム管理者ガイド」を参照してください。

また、URL のリクエストパラメータを取得して利用することもできます。URL のリクエストパラメータを利用する場合は、次のキーで ucnpOptions パラメータを取得する必要があります。

ucnp.request.options キー

セッションから、URL に指定した ucnpOptions パラメータの値を取得するためのキーです。HttpSession.getAttribute()メソッドにこのキーを指定することで、URL デコードされた ucnpOptions パラメータの値を取得できます。URL に ucnpOptions パラメータが指定されていない場合は、null を返します。

ucnp.request.options キーの指定例については、「5.3 IIoPluginController (サーバ処理実装インタフェース)」の inputFromNode メソッドの指定例を参照してください。

(3) 戻り値

なし

(4) 例外

UCNPPluginException - 中断情報の削除処理などでエラーが発生した場合

ナビゲーション プラットフォームは、このメソッドがスローした UCNPPluginException 例外の詳細メッセージを、業務実行画面のダイアログに表示します。スローする UCNPPluginException 例外には、どのようなエラーが起こったか、またどのように対処すればよいか、利用者にわかるメッセージを設定してください。なお、中断再開プラグインが出力するエラーメッセージが多数の改行を含む場合、JavaScript の Alert でエラーメッセージを表示するので、メッセージダイアログの一部が画面に表示されなくなる場合があります。そのため、エラーメッセージは画面に表示できる長さにしてください。

(5) 呼び出しタイミング

ナビゲーション プラットフォームがこのメソッドを呼び出すタイミングと、削除対象となる中断情報を次に示します。

表 5-12 deleteAll メソッドの呼び出しタイミングと削除対象の中断情報

呼び出しタイミング	削除対象の中断情報
業務コンテンツを作成するユーザが、業務編集画面で業務コンテンツを編集して保存したとき	編集した業務コンテンツに関する、すべての中断情報
業務コンテンツを作成するユーザが、業務編集画面で業務コンテンツを削除したとき	削除した業務コンテンツに関する、すべての中断情報
業務コンテンツを作成するユーザが、ドロップダウンの選択項目を一括設定する CSV ファイルを更新して、業務コンテンツを更新したとき	選択した業務コンテンツに関する、すべての中断情報
業務コンテンツを作成するユーザが、業務編集画面で業務コンテンツの属性を変更したとき	選択した業務コンテンツに関する、すべての中断情報
プラグインを開発するユーザが、業務編集画面で [プラグインパーツ更新] メニューを選択して、業務コンテンツを更新したとき	更新した業務コンテンツに関する、すべての中断情報

5.6 ISuspendInfo (中断情報インタフェース)

中断再開プラグインで、中断情報を表すインタフェースです。ISuspendActionController インタフェースの各メソッドで、保存または復元する中断情報の受け渡しに使用するインタフェースです。このインタフェースのオブジェクトはシステムが生成するため、プラグインで生成する必要はありません。

パッケージ

```
package jp.co.hitachi.soft.ucnp.plugin.suspend.model;
```

形式

```
public interface ISuspendInfo
```

5.7 ParamConvertUtil (入出力パラメタ変換ユーティリティクラス)

入出力パラメタを Map 形式に変換するユーティリティクラスです。

パッケージ

```
package jp.co.hitachi.soft.ucnp.plugin.inputoutput.util;
```

形式

```
public class ParamConvertUtil
```

メソッド

- decodeHtmlPartParam メソッド
HTML パーツから入力されたエンコード文字列を引数として受け取り、Map 形式に変換します。

5.7.1 decodeHtmlPartParam メソッド

HTML パーツから入力されたエンコード文字列を引数として受け取り、Map 形式に変換します。

(1) 形式

```
public static Map<String, String> decodeHtmlPartParam(String param)
    throws UCNPPuginException;
```

(2) 引数

(a) param

HTML パーツから入力されたエンコード文字列が格納されます。

(3) 戻り値

Map<String, String>

HTML パーツ内の各入力項目に対応した値が Map で返されます。入力項目の name 属性をキー、value 属性を値とする Map です。引数 param が空文字の場合、空の Map が返されます。ユーザは戻り値の Map から、HTML パーツで設定した要素の name 属性をキーとして、値を取得してください。

(4) 例外

UCNPPuginException - エンコード文字列の変換に失敗した場合

このメソッドでエラーが発生した場合、Map 形式への変換処理は中断され、UCNPPuginException 例外がスローされます。エラーの一覧を次に示します。

表 5-13 decodeHtmlPartParam メソッドで発生するエラー一覧

エラー	メッセージ ID
HTML パーツの入力パラメタ以外のデータに対して、このメソッドを実行した場合	KDCZ10205-E

(5) 使用例

IIOPluginController インタフェースの inputFromNode メソッド内で、decodeHtmlPartParam メソッドを使用する例を説明します。

条件

この例は次の条件に基づいています。

- 入出力プラグインのサンプルを利用する。
入出力プラグインのサンプルについては、「付録 A.1 入出力プラグイン (サンプル) の使用方法」を参照してください。

- HTML パーツ用に次の外部 CSS ファイルを作成している。

```
table.sample {
  border:1px solid #777777;
  border-collapse:collapse;
  border-spacing:0;
  background-color:#ffffff;
}
th.sample {
  border-right:1px solid #777777;
  border-bottom:1px solid #777777;
  background-color:#e3e5e7;
  padding:0.3em 1em;
  text-align:center;
}
td.sample {
  border-right:1px solid #777777;
  border-bottom:1px solid #777777;
  padding:0.3em 1em;
}
```

- 入力パラメタ [inputItemName1] に、次のソースコードを入力した HTML パーツを関連づけている。

```
<TABLE class="sample">
<TBODY>
<TR>
<TH class="sample"></TH>
<TH class="sample">注文番号</TH>
</TR>
<TR>
<TD class="sample"><input type="radio" name="order" value="0001-20100801-00001" /></TD>
<TD class="sample"> 0001-20100801-00001</TD>
</TR>
<TR>
<TD class="sample"><input type="radio" name="order" value="0001-20100801-00002" /></TD>
<TD class="sample"> 0001-20100801-00002</TD>
</TR>
</TBODY>
</TABLE>
```

HTML パーツの画面上の表示は次のとおりです。

	注文番号
<input type="radio"/>	0001-20100801-00001
<input type="radio"/>	0001-20100801-00002

- 出力パラメタ [outputItemName1] と固定テキストパーツを関連づけている。
- 利用者が選択したラジオボタンに応じて、次の工程のガイド領域に表示される注文番号が変わるようにする。

入出力プラグインの実装例

条件に示した HTML パーツから入力パラメタを受け取って処理する実装例を次に示します。
decodeHtmlPartParam メソッドを使用しているのは、太字で示した部分です。

```
package jp.co.hitachi.soft.ucnp.plugin.sample.ioaction.controller;

import java.util.HashMap;
import java.util.Map;
import javax.servlet.http.HttpSession;
import jp.co.hitachi.soft.ucnp.plugin.inputoutput.controller.IIoPluginController;
import jp.co.hitachi.soft.ucnp.plugin.inputoutput.common.UCNPPluginException;
import jp.co.hitachi.soft.ucnp.plugin.inputoutput.util.ParamConvertUtil;

public class IoPluginController implements IIoPluginController {
    /* クライアントから受信した入力パラメタの値を保持するメンバ変数です。 */
    private String param1 = null;
    private String param2 = null;
    private String param3 = null;

    public Map<String, Object> inputFromNode(HttpSession session,
        Map<String, Object> param) {

        /* クライアントに処理結果を送るためのMapを生成します。 */
        Map<String, Object> map = new HashMap<String, Object>();

        /* ボタンの種別で処理を分けます。 */
        String buttonType = (String) param.get("ucnp.button.type");
        if ("show_next_page".equals(buttonType)) {
            /* [次へ] ボタンの場合は、入力の値を取得してメンバ変数で保持します。 */

            /* クライアントから受信したデータをMapに展開します。 */
            Map<?, ?> inParamMap = (Map<?, ?>) param.get("ucnp.current.params.map");

            /* 変換する値をMapより取り出します。 */
            String inParam1 = (String) inParamMap.get("inputItemName1");

            Map<String, String> decodedMap = null;
            /* APIを利用するため、try-catchで囲みます。 */
            try {
                decodedMap = ParamConvertUtil.decodeHtmlPartParam(inParam1);
            } catch (UCNPPluginException e) {
                /* 例外処理を行います。 */
                String errMsg = e.getMessage();
                /* プラグインのログ出力などの処理をします。 */
                /* エラーメッセージをマップに追加して返します。 */
                map.put("ucnp.error.message", errMsg);
                return map;
            }

            /* name属性が「order」のvalue属性の値を取得します。 */
            String value = decodedMap.get("order");

            /* メンバ変数に入力パラメタの値を設定します。 */
            param1 = value;
            param2 = (String) inParamMap.get("inputItemName2");
            param3 = (String) inParamMap.get("inputItemName3");

        } else if ("show_previous_page".equals(buttonType)) {
            /* 戻るボタンの場合は、何もしません。 */
        }

        return map;
    }

    public Map<String, Object> outputToNode(HttpSession session,
        Map<String, Object> param) {

        /* クライアントへ送るMapを生成します。 */
        Map<String, Object> map = new HashMap<String, Object>();

        /* ボタンの種別で処理を分けます。 */
    }
}
```

```
String buttonType = (String) param.get("ucnp.button.type");
if ("show_next_page".equals(buttonType)) {
    /* 次へボタンの場合は、入力の値を出力項目へマッピングします。 */

    Map<String, String> outParamMap = new HashMap<String, String>();

    /* Mapに送信するデータを設定します。 */
    /* outputItemName1に固定テキストパーツを設定しているので、
       この固定テキストパーツの値としてHTMLパーツから取得した値を設定します。 */
    outParamMap.put("outputItemName1", param1);
    outParamMap.put("outputItemName2", param2);
    outParamMap.put("outputItemName3", param3);

    map.put("ucnp.next.params.map", outParamMap);
} else if ("show_previous_page".equals(buttonType)) {
    /* 戻るボタンの場合は、何もしません。 */
}

return map;
}
}
```

5.8 SuspendInfoSerializeUtil (中断情報直列化ユーティリティクラス)

中断情報 (ISuspendInfo) を直列化, 直列化復元するためのメソッドを提供するユーティリティクラスです。

このクラスの `serialize` メソッドを使用すると, 中断情報 (ISuspendInfo) を XML 形式に直列化できます。また, `deserialize` メソッドを使用すると, XML 形式に直列化された中断情報 (ISuspendInfo) を復元できます。

パッケージ

```
package jp.co.hitachi.soft.ucnp.plugin.suspend.util;
```

形式

```
public class SuspendInfoSerializeUtil
```

メソッド

- `serialize` メソッド
中断情報を XML 形式に直列化します。
- `deserialize` メソッド
直列化された中断情報を読み込み, 中断情報を復元します。

5.8.1 serialize メソッド

中断再開プラグインで, 中断情報を直列化するためのメソッドです。パラメタに指定された中断情報 (`suspendInfo`) を XML 形式に直列化し, 結果を出力ストリーム `out` に書き込みます。このメソッドでは, 出力ストリーム `out` のクローズ処理はしません。呼び出し側でクローズ処理を実施してください。

中断情報の直列化, およびストリームへの出力に失敗した場合は, `UCNPPluginException` 例外をスローします。このメソッドがスローする `UCNPPluginException` 例外の詳細メッセージには, ナビゲーションプラットフォームのメッセージ ID が付きます。スローされた例外をハンドリングして, ダイアログに表示するメッセージを設定した `UCNPPluginException` 例外をスローしてください。

(1) 形式

```
public static void serialize(ISuspendInfo suspendInfo, OutputStream out)
    throws UCNPluginException;
```

(2) 引数

(a) `suspendInfo`

中断情報が格納されます。

(b) `out`

直列化した中断情報を書き込む出力ストリームが格納されます。

(3) 戻り値

なし

(4) 例外

UCNPPluginException - 中断情報の直列化, および出力に失敗した場合

このメソッドでエラーが発生した場合, 直列化処理は中断され, UCNPPluginException 例外がスローされます。エラーの一覧を次に示します。

表 5-14 serialize メソッドで発生するエラー一覧

エラー	メッセージ ID
suspendInfo パラメタに null を指定した場合	KDCZ10083-E
out パラメタに null を指定した場合	KDCZ10083-E
suspendInfo パラメタに不正なオブジェクト (ナビゲーション プラットフォームで生成していないオブジェクト) を指定した場合	KDCZ10084-E
アクセス権, ディスク容量などの問題でパラメタに指定された出力ストリームに書き込みができなかった場合	KDCZ10085-E

5.8.2 deserialize メソッド

中断再開プラグインで, 中断情報を直列化復元するためのメソッドです。パラメタに指定された入力ストリーム in から直列化した中断情報を読み込み, 中断情報を復元します。このメソッドでは, 入力ストリーム in のクローズ処理はしません。呼び出し側でクローズ処理を実施してください。

中断情報の直列化復元, およびストリームからの入力に失敗した場合は, UCNPPluginException 例外をスローします。このメソッドがスローする例外のメッセージにはシステムのメッセージ ID が付きます。スローされた例外をハンドリングして, ダイアログに表示するメッセージを設定した UCNPPluginException 例外をスローしてください。

(1) 形式

```
public static ISuspendInfo deserialize(InputStream in) throws UCNPPluginException;
```

(2) 引数

(a) in

直列化した中断情報を読み込む入力ストリームが格納されます。

(3) 戻り値

入力ストリームから直列化復元した中断情報

(4) 例外

UCNPPluginException - 中断情報の復元, および入力に失敗した場合

このメソッドでエラーが発生した場合, 直列化処理は中断され, UCNPPluginException 例外がスローされます。エラーの一覧を次に示します。

表 5-15 deserialize メソッドで発生するエラー一覧

エラー	メッセージ ID
in パラメタに null を指定した場合	KDCZ10083-E
in パラメタに指定された入力ストリームが中断情報 (ISuspendInfo) を復元するストリームでない場合	KDCZ10086-E
アクセス権などの問題でパラメタに指定された入力ストリームから読み込みができなかった場合	KDCZ10093-E

5.9 UCNPPluginException (入出力プラグイン例外クラス)

入出力プラグイン内で発生した例外を表すクラスです。

パッケージ

```
package jp.co.hitachi.soft.ucnp.plugin.inputoutput.common;
```

形式

```
public class UCNPPluginException extends Exception
```

メソッド

- getMessage メソッド
エラーの原因を記述する詳細メッセージを取得します。
- getMessageId メソッド
メッセージ ID を取得します。

5.9.1 getMessage メソッド

エラーの原因を記述する詳細メッセージを取得します。メッセージ ID は含まれません。

(1) 形式

```
public String getMessage();
```

(2) 引数

なし

(3) 戻り値

エラーの原因を記述する詳細メッセージ

(4) 例外

なし

5.9.2 getMessageId メソッド

メッセージ ID を取得します。

(1) 形式

```
public String getMessageId();
```

(2) 引数

なし

(3) 戻り値

メッセージ ID

(4) 例外

なし

5.10 UCNPPluginException (中断再開プラグイン例外クラス)

中断再開プラグイン内で発生した例外を表すクラスです。

ISuspendActionController の各メソッドでエラーが発生した場合、この例外にメッセージを設定してスローします。例外のメッセージは、ダイアログに表示されます。

中断再開プラグインのエラー内容をメッセージダイアログに表示する場合、エラー内容が長文になると、Web ブラウザによってはメッセージダイアログに表示される文字列が途切れて、全文が表示されないときがあります。

パッケージ

```
package jp.co.hitachi.soft.ucnp.plugin.suspend.common;
```

形式

```
public class UCNPPluginException extends Exception
```

コンストラクタ

- UCNPPluginException(String message)コンストラクタ
指定された詳細メッセージを使用して、新規の例外を作成します。
- UCNPPluginException(String message, Throwable cause)コンストラクタ
指定された詳細メッセージおよび原因を使用して新規の例外を作成します。

5.10.1 UCNPPluginException(String message)コンストラクタ

指定された詳細メッセージを使用して、例外を新規に作成します。

(1) 形式

```
public UCNPPluginException(String message);
```

(2) 引数

(a) message

エラーの原因を記述する詳細メッセージが表示されます。

このメッセージは、業務実行画面のダイアログに表示されるため、エラー内容および対処方法が利用者にわかるようなメッセージを指定してください。

(3) 例外

なし

5.10.2 UCNPPluginException(String message, Throwable cause)コンストラクタ

指定された詳細メッセージとエラーの原因を使用して、例外を新規に作成します。

(1) 形式

```
public UCNPPuginException(String message, Throwable cause);
```

(2) 引数

(a) message

エラーの原因を記述する詳細メッセージが表示されます。

このメッセージは、業務実行画面のダイアログに表示されるため、エラー内容および対処方法が利用者にわかるようなメッセージを指定してください。

(b) cause

エラーの原因となった Throwable オブジェクトが格納されます。

(3) 例外

なし

6

API リファレンス (カスタム画面開発の場合)

この章では、カスタム画面の開発で使用する API について説明します。

6.1 API 一覧 (カスタム画面開発の場合)

カスタム画面の開発で使用する API を次に示します。

表 6-1 API 一覧 (カスタム画面開発)

分類	インタフェース名 またはクラス名	説明
検索機能	IHitContentData	業務コンテンツの検索結果を格納するデータモデルのインタフェースです。
	IHitData	検索ワードとして指定された要素の検索結果を格納するデータモデルのインタフェースです。
	IHitGuideData	検索ワードとして指定された要素の検索結果を、ガイドごとに格納するデータモデルのインタフェースです。
	ISearcher	検索ワードとして指定された要素を検索するためのインタフェースです。
	SearcherConstants	検索機能で利用する定数を格納するユーティリティクラスです。
	SearcherProvider	検索用オブジェクトを取得するためのクラスです。
	UCNPServerErrorException	ナビゲーション プラットフォームで発生した例外を表すクラスです。
ログアウト機能	LogoutActionUtil	ナビゲーション プラットフォームからログアウトするためのユーティリティクラスです。

注

検索ワードは複数指定できます。検索ワードを複数指定した場合、各検索ワードのことを要素といいます。

6.2 IHitContentData (業務コンテンツの検索結果格納 インタフェース)

業務コンテンツの検索結果を格納するデータモデルのインタフェースです。

パッケージ

```
package jp.co.hitachi.soft.ucnp.base.search.model.ifc;
```

形式

```
public interface IHitContentData
```

メソッド

- getHitCount メソッド
業務コンテンツ内のすべての検索ワードのヒット数の合計を取得します。
- getHitCounts メソッド
業務コンテンツ内のヒット数の配列を、検索ワードとして指定された要素ごとに取得します。
- getContentId メソッド
業務コンテンツの ID を取得します。
- getContentName メソッド
業務コンテンツの名称を取得します。
- getLastUpdated メソッド
業務コンテンツの更新日時を取得します。
- getHitGuideDataList メソッド
ヒットしたガイドごとの検索結果のリストを取得します。

IHitContentData インタフェースの定数フィールドについて

このインタフェースは、次に示す定数フィールドを持っています。

表 6-2 IHitContentData インタフェースの定数フィールド

項番	フィールド名	形式	値	説明
1	LAST_UPDATED_EMPTY	public static final long	-1L	更新日時が存在しません。

6.2.1 getHitCount メソッド

業務コンテンツ内のすべての検索ワードのヒット数の合計を取得します。

(1) 形式

```
int getHitCount();
```

(2) 引数

なし

(3) 戻り値

業務コンテンツ内のすべての検索ワードのヒット数の合計

(4) 例外

なし

6.2.2 getHitCounts メソッド

業務コンテンツ内のヒット数の配列を、検索ワードとして指定された要素ごとに取得します。

配列の順序は、IHitData インタフェースの getWords メソッドで取得する要素の順序と同じです。詳細は、「6.3 IHitData (検索結果格納インタフェース)」の getWords メソッドを参照してください。

(1) 形式

```
int[] getHitCounts();
```

(2) 引数

なし

(3) 戻り値

業務コンテンツ内の検索ワードのヒット数の要素ごとの配列

(4) 例外

なし

6.2.3 getContentId メソッド

業務コンテンツの ID を取得します。

(1) 形式

```
java.lang.String getContentId();
```

(2) 引数

なし

(3) 戻り値

業務コンテンツの ID

(4) 例外

なし

6.2.4 getContentName メソッド

業務コンテンツの名称を取得します。

(1) 形式

```
java.lang.String getContentName();
```

(2) 引数

なし

(3) 戻り値

業務コンテンツの名称

(4) 例外

なし

6.2.5 getLastUpdated メソッド

標準時間から業務コンテンツの更新日時までのミリ秒数を取得します。なお、更新日時の取得の精度は 1,000 ミリ秒です。

(1) 形式

```
long getLastUpdated();
```

(2) 引数

なし

(3) 戻り値

業務コンテンツの更新日時 (標準時間から更新日時までのミリ秒数)

ナビゲーション プラットフォームのバージョン 08-01 以前で作成された業務コンテンツには、更新日時のデータがありません。そのため、バージョン 08-01 以前で作成された業務コンテンツの更新日時は、すべて定数フィールド LAST_UPDATED_EMPTY になります。

(4) 例外

なし

6.2.6 getHitGuideDataList メソッド

ヒットしたガイドごとの検索結果のリストを取得します。

業務コンテンツ名称などガイド外に検索ワードがヒットした場合、そのデータがリストの 1 番目の要素になります。それ以外の要素の順序は不定です。必要に応じてソートしてください。

(1) 形式

```
java.util.List<IHitGuideData> getHitGuideDataList();
```

(2) 引数

なし

(3) 戻り値

ヒットしたガイドごとの検索結果のリスト

ただし、ISearcher インタフェースの search メソッドの guideFlag パラメタに false を指定して、検索でガイドごとのヒット数を出力しない指定をしている場合は、null を返します。詳細は、「6.5 ISearcher (検索インタフェース)」の search メソッドを参照してください。

(4) 例外

なし

6.3 IHitData (検索結果格納インタフェース)

検索ワードとして指定された要素の検索結果を格納するデータモデルのインタフェースです。

パッケージ

```
package jp.co.hitachi.soft.ucnp.base.search.model.ifc;
```

形式

```
public interface IHitData
```

メソッド

- `getHitContentList` メソッド
検索ワードにヒットした検索結果のリストを業務コンテンツごとに取得します。
- `getWords` メソッド
検索ワードとして指定された要素の配列を取得します。

6.3.1 `getHitContentList` メソッド

検索ワードにヒットした検索結果のリストを業務コンテンツごとに取得します。

(1) 形式

```
java.util.List<IHitContentData> getHitContentList();
```

(2) 引数

なし

(3) 戻り値

ヒットした業務コンテンツごとの検索結果のリスト

検索でヒットした業務コンテンツがない場合は、空のリストが返されます。

(4) 例外

なし

6.3.2 `getWords` メソッド

検索ワードとして指定された要素の配列を取得します。複数の要素が指定された場合は、スペース区切りごとに一つの要素と見なします。

(1) 形式

```
java.lang.String[] getWords();
```

(2) 引数

なし

(3) 戻り値

検索ワードとして指定された要素の配列

(4) 例外

なし

6.4 IHitGuideData (ガイドの検索結果格納インターフェース)

検索ワードとして指定された要素の検索結果を、ガイドごとに格納するデータモデルのインターフェースです。

検索ワードの検索対象は次のように分けられます。

- ガイド内の検索対象
各種ガイドパーツの属性およびプロセスノードの名称が含まれます。
- ガイド外の検索対象
ガイド内の検索対象以外がまとめられて、一つのインスタンスとなります。業務コンテンツの名称、ターミナルノードの名称などが含まれます。

パッケージ

```
package jp.co.hitachi.soft.ucnp.base.search.model.ifc;
```

形式

```
public interface IHitGuideData
```

メソッド

- getHitCount メソッド
ガイドのすべての検索ワードのヒット数の合計を取得します。
- getHitCounts メソッド
ガイドの検索ワードのヒット数の配列を要素ごとに取得します。
- getGuideName メソッド
ガイドの名称 (プロセスノードの名称) を取得します。

6.4.1 getHitCount メソッド

ガイドのすべての検索ワードのヒット数の合計を取得します。

(1) 形式

```
int getHitCount();
```

(2) 引数

なし

(3) 戻り値

- 検索対象がガイド内の場合、ガイド内のすべての検索ワードのヒット数の合計
- 検索対象がガイド外の場合、ガイド外のすべての検索ワードのヒット数の合計

(4) 例外

なし

6.4.2 getHitCounts メソッド

ガイドの検索ワードのヒット数の配列を要素ごとに取得します。

配列の順序は、IHitData インタフェースの getWords メソッドで取得する要素の順序と同じです。詳細は、「6.3 IHitData (検索結果格納インタフェース)」の getWords メソッドを参照してください。

(1) 形式

```
int[] getHitCounts();
```

(2) 引数

なし

(3) 戻り値

- 検索対象がガイド内の場合、ガイド内の検索ワードの要素ごとのヒット数の配列
- 検索対象がガイド外の場合、ガイド外の検索ワードの要素ごとのヒット数の配列

(4) 例外

なし

6.4.3 getGuideName メソッド

ガイドの名称 (プロセスノードの名称) を取得します。

(1) 形式

```
java.lang.String getGuideName();
```

(2) 引数

なし

(3) 戻り値

- 検索対象がガイド内の場合、ガイドの名称 (プロセスノードの名称)
- 検索対象がガイド外の場合、null

(4) 例外

なし

6.5 ISearcher (検索インタフェース)

検索ワードとして指定された要素を検索するためのインタフェースです。

インスタンスは SearcherProvider クラスを利用して取得してください。SearcherProvider クラスの詳細は、「6.8 SearcherProvider (検索オブジェクト取得クラス)」を参照してください。

パッケージ

```
package jp.co.hitachi.soft.ucnp.base.search.searcher;
```

形式

```
public interface ISearcher
```

メソッド

- search メソッド
検索を実行します。

6.5.1 search メソッド

検索を実行します。

(1) 形式

```
IHitData search(boolean guideFlag, boolean andFlag, int sortFlag, java.lang.String word)
    throws UCNPServerException;
```

(2) 引数

(a) guideFlag

ガイドごとの検索ヒット数の出力有無を指定します。

- true : 出力する
- false : 出力しない

false を指定した場合、戻り値に含まれる IHitContentData オブジェクトの getHitGuideDataList メソッドは常に null を返します。

(b) andFlag

複数の要素を検索ワードとして指定するときの AND 検索の有無を指定します。

- true : AND 検索
- false : OR 検索

(c) sortFlag

検索結果の業務コンテンツのソート順序を指定します。

業務コンテンツのソート順序には、SearcherConstants クラスの次の定数のどれかを指定します。

- SORT_TYPE_NONE
- SORT_TYPE_HITCOUNT

- SORT_TYPE_NAME
- SORT_TYPE_LASTUPDATED

SearcherConstants クラスの定数の詳細は、「6.7 SearcherConstants (検索クラス)」を参照してください。

なお、不正な値を指定した場合はソートされません。

(d) word

検索ワードを指定します。検索を実行するとき、次に示す検索ワードの違いは区別されないので注意してください。

- 英字の大文字・小文字
- 英数字およびカタカナ (JIS X 0201 0xA1~0xDF) の全角・半角

なお、指定した検索ワードは、部分一致で検索されます。例えば、「京都」という検索ワードを指定すると、「東京都」「京都市」などもヒットします。前方一致または後方一致の指定はできません。

指定できる検索ワードの条件を次に示します。

- 半角スペースまたは全角スペースで区切ることによって、複数の要素を検索ワードとして指定できます。同時に指定できる要素は、最大 10 個です。10 個を超えた場合、例外がスローされます。
- この引数に指定できる文字数は最大 500 文字です。500 文字を超えた場合、例外がスローされます。
- 制御文字 (ASCII コードの 0x00~0x1F と 0x7F) は検索ワードとして指定できません。制御文字を指定した場合、例外がスローされます。
- スペースを含む検索ワードを指定する場合は、検索ワードを半角の引用符 (") で囲んでください。ただし、半角の引用符 (") を検索ワードとして指定することはできません。検索ワードとして半角の引用符 (") を指定した場合、例外がスローされます。
- 検索ワードの先頭または末尾にスペースがある場合、スペースは自動的に除去されます。
- null, 空文字, またはスペースだけの検索ワードを指定した場合は、検索は実行されません。
- 同じパターンの繰り返しが含まれる文字列を検索し、かつパーツにも同じパターンを繰り返す文字列が含まれる場合、一度ヒットした文字は重複して検索の対象となることはないため、注意が必要です。
例えば、検索ワードに「市川市」(いちかわし) を指定して「市川市川市町」(いちかわし かわいちちょう) という固定テキストパーツを検索した場合、ヒット数は 1 です (1~3 文字目の「市川市」が最初にヒットの対象となるため、3~5 文字目の「市川市」はヒットの対象外)。
- 検索ワードに、同じ単語を大文字と小文字でそれぞれ一つずつ指定した場合、大文字と小文字それぞれで検索した結果を表示します。
- 同じ検索ワードを複数指定して検索した場合は、それぞれの検索ワードに対する検索結果が表示されるため、ヒット数が多くなります。
例えば、「Japan」という固定テキストパーツが一つしかない場合も、検索ワードに「Japan Japan」と「Japan」を 2 回指定して検索すると、ヒット数は 2 になります。また、同様の条件で「Japan」という固定テキストパーツに対して、「JAPAN Japan」と検索した場合も、ヒット数は 2 になります。

(3) 戻り値

検索結果のオブジェクト

word パラメタに null, 空文字, またはスペースだけの文字列を指定した場合は null を返します。

(4) 例外

UCNPServerErrorException - 検索でエラーが発生した場合

このメソッドでエラーが発生した場合、検索は中断され、UCNPServerErrorException 例外がスローされます。エラーの一覧を次に示します。

表 6-3 search メソッドで発生するエラー一覧

エラー	メッセージ ID
word パラメタの文字数が 500 文字を超えている場合	KDCZ10206-E
word パラメタに制御文字 (ASCII コードの 0x00~0x1F と 0x7F) が含まれている場合	KDCZ10207-E
word パラメタに含まれる半角の引用符 (") の使い方に誤りがある場合	KDCZ10208-E
word パラメタの検索ワードの要素数が 10 個を超えている場合	KDCZ10209-E
ナビゲーション プラットフォームにログインしていない状態でこのメソッドを実行した場合	KDCZ10066-E
リソースアダプタが見つからない場合	KDCZ18028-E
リソースアダプタのホスト名が正しくない場合 データベースに検索用テーブルが存在しない場合	KDCZ18029-E
その他、環境不正などの場合	上記以外

6.6 LogoutActionUtil (ログアウト処理クラス)

ナビゲーション プラットフォームからログアウトするためのユーティリティクラスです。このクラスは、新規画面の開発で使用します。メニュー領域のカスタマイズには使用できないので注意してください。

パッケージ

```
package jp.co.hitachi.soft.ucnp.base.common;
```

形式

```
public class LogoutActionUtil extends java.lang.Object
```

メソッド

- logout メソッド
ナビゲーション プラットフォームからログアウトします。

6.6.1 logout メソッド

ナビゲーション プラットフォームからログアウトします。ナビゲーション プラットフォームにログインしていない場合は、何も実行しません。

ログアウトの実行結果はトレースファイルを確認してください。また、ログアウトに失敗した場合、トレースファイルに KDCZ18033-E のメッセージが出力されます。

トレースファイルは、ユーザプロパティファイル (ucnp_user.properties) の ucnp.base.server.trace.path プロパティに設定されているディレクトリに格納されます。デフォルトの格納ディレクトリは次のとおりです。

格納ディレクトリ

ナビゲーション プラットフォームインストールディレクトリ¥logs¥ucnp_trace[*N*].log (*N*はファイル面数)

(1) 形式

```
public static void logout(HttpServletRequest request, HttpServletResponse response);
```

(2) 引数

(a) request

JSP の暗黙オブジェクト request を指定します。これ以外の値を指定した場合は、エラーになります。

(b) response

JSP の暗黙オブジェクト response を指定します。これ以外の値を指定した場合は、エラーになります。

(3) 戻り値

なし

(4) 例外

なし

6.7 SearcherConstants (検索クラス)

検索機能で利用する定数を格納するユーティリティクラスです。

パッケージ

```
package jp.co.hitachi.soft.ucnp.base.search.constants;
```

形式

```
public class SearcherConstants extends java.lang.Object
```

SearcherConstants クラスの定数フィールドについて

このクラスは、次に示す定数フィールドを持っています。

表 6-4 SearcherConstants クラスの定数フィールド

項番	フィールド名	形式	値	説明
1	SORT_TYPE_HITCOUNT	public static final int	1	ヒット数の多い順でソートします。 ソートの優先順位を次に示します。 1. ヒット件数の多い順 2. コンテンツ名称の昇順 3. 不定
2	SORT_TYPE_LASTUPDATED	public static final int	3	更新日時の新しい順でソートします。 ソートの優先順位を次に示します。 1. 更新日時の新しい順 2. コンテンツ名称の昇順 3. 不定
3	SORT_TYPE_NAME	public static final int	2	業務コンテンツ名称の昇順でソートします。 ソートの優先順位を次に示します。 1. コンテンツ名称の昇順 2. ヒット件数の多い順 3. 不定
4	SORT_TYPE_NONE	public static final int	0	ソートしません。ソートの優先順位も不定です。

6.8 SearcherProvider (検索オブジェクト取得クラス)

検索用オブジェクトを取得するためのクラスです。

パッケージ

```
package jp.co.hitachi.soft.ucnp.base.search.searcher;
```

形式

```
public class SearcherProvider extends java.lang.Object
```

メソッド

- `getSearcher` メソッド
検索用オブジェクトを取得します。

6.8.1 `getSearcher` メソッド

検索用オブジェクトを取得します。

このメソッドを同じプロセスで複数回実行した場合、同じインスタンスを返します。

(1) 形式

```
public static ISearcher getSearcher();
```

(2) 引数

なし

(3) 戻り値

検索用オブジェクト

検索機能を利用する設定がされていない場合は `null` を返します。

(4) 例外

なし

6.9 UCNPServerException (例外クラス)

ナビゲーション プラットフォームで発生した例外を表すクラスです。

パッケージ

```
package jp.co.hitachi.soft.ucnp.base.common;
```

形式

```
public class UCNPServerException extends java.lang.Exception
```

メソッド

- getMessageId メソッド
メッセージ ID を取得します。

補足事項

java.lang.Throwable クラスから継承された getMessage()メソッドでは、「メッセージ ID + "(半角スペース) + メッセージ」の文字列を取得します。

6.9.1 getMessageId メソッド

メッセージ ID を取得します。

(1) 形式

```
public java.lang.String getMessageId();
```

(2) 引数

なし

(3) 戻り値

メッセージ ID

(4) 例外

なし

付録

付録 A サンプルプラグインの使用方法

uCosminexus Navigation Developer では、次に示すサンプルプラグインを提供しています。

- 入出力プラグイン (サンプル)
ノードを遷移するときに、プラグインパーツのパラメタに関連づけたガイドパーツの入力値を保持します。業務を完了したときに、保持している入力値をファイルに出力します。
- 中断再開プラグイン (サンプル)
複数の業務コンテンツを並行して利用する場合に便利です。操作の途中にはほかの業務コンテンツを参照したり、ログアウトしたりしても、一時保存した状態から業務コンテンツの操作を再開できるようにします。
中断情報はサーバ上のファイルシステムに保存され、操作再開時には、サーバ上のファイルシステムから中断情報が読み込まれます。

ここでは、サンプルプラグインの使用方法について説明します。なお、サンプルプラグインを使用するには注意事項があります。詳細は「付録 A.3 サンプルプラグインを使用するときの注意事項」を参照してください。

付録 A.1 入出力プラグイン (サンプル) の使用方法

入出力プラグイン (サンプル) の使用手順を次に示します。

手順

1. サンプルプラグインをプラグイン開発の作業ディレクトリにコピーします。
コピーするファイルとコピー先のディレクトリを次に示します。
コピーするファイル
`uCosminexus Navigation Developer` インストールディレクトリ¥sample¥plugin
¥sample.OutputGuideData
コピー先ディレクトリ
`uCosminexus Navigation Developer` インストールディレクトリ¥pluginSDK¥plugin
2. アプリケーション属性ファイルを編集します。
アプリケーション属性ファイルの編集方法については、次の個所を参照してください。
 - 「3.5 アプリケーション属性ファイル (application.xml) の編集」
3. プラグインをビルドします。
プラグインのビルド方法については、「3.6 プラグインのビルド」を参照してください。
4. Web サーバリダイレクトを設定します。
Web サーバリダイレクトの設定手順については、「3.7 Web サーバリダイレクトの設定」を参照してください。なお、手順で説明されているプラグイン名には、「OutputGuideData」を指定してください。
5. J2EE アプリケーションのインポートと開始を実行します。
J2EE アプリケーションのインポートと開始の実行方法については、「3.8 uCosminexus Navigation Developer の J2EE アプリケーションのインポートと開始」を参照してください。
6. 業務完了時に入力値を出力するファイルの出力先フォルダを作成します。

サンプルプラグインをそのまま利用する場合

C:*ucnpwork

出力先フォルダを変更する場合

プラグインをカスタマイズしてください。プラグインをカスタマイズする方法については、「3.3.1(8) プラグインが実行する処理の実装」を参照してください。

7. サンプルプラグインを業務コンテンツに関連づけて、動作確認を実施します。

サンプルプラグインを業務コンテンツに関連づける方法については、「3.9 業務コンテンツへの関連づけ（入出力プラグインの場合）」を参照してください。なお、サンプルプラグインを利用するには、次のとおり業務フローにプラグインパーツおよびターミナルノードを配置する必要があります。

- 開始のプロセスノードにプラグインパーツを配置する
- 業務フローの終点にターミナルノードを配置し、そのノードの前のプロセスノードにプラグインパーツを配置する

付録 A.2 中断再開プラグイン（サンプル）の使用法

中断再開プラグイン（サンプル）の使用手順を次に示します。

手順

1. サンプルプラグインをプラグイン開発の作業ディレクトリにコピーします。

コピーするファイルとコピー先のディレクトリを次に示します。

コピーするファイル

uCosminexus Navigation Developer インストールディレクトリ*sample*plugin
*ucnp.plugin.suspend

コピー先ディレクトリ

uCosminexus Navigation Developer インストールディレクトリ*pluginSDK*plugin

2. アプリケーション属性ファイルを編集します。

アプリケーション属性ファイルの編集方法については、次の個所を参照してください。

- 「3.5 アプリケーション属性ファイル（application.xml）の編集」

3. プラグインをビルドします。

プラグインのビルド方法については、「3.6 プラグインのビルド」を参照してください。

4. 次の URL にアクセスして、Cosminexus Management Server を起動します。

http://ホスト名:ポート番号/mngsvr/

- ホスト名
開発環境の J2EE サーバマシンのホスト名または IP アドレスです。
- ポート番号
Cosminexus Management Server が使用するポート番号です。マニュアル「Cosminexus V9 ナビゲーション プラットフォーム 導入から運用まで」の手順どおりにセットアップした環境では、「28080」です。

5. Cosminexus Management Server にログインします。

運用管理ポータルが表示されます。

なお、マニュアル「Cosminexus V9 ナビゲーション プラットフォーム 導入から運用まで」の手順どおりにセットアップした環境では、ログイン時に入力する管理ユーザ ID が「admin」、パスワードが「ucnppass」です。

6. [論理サーバの環境設定] アンカーをクリックします。

論理サーバの環境設定画面が表示されます。

7. [サーバビュー] タブ - [論理 J2EE サーバ] - [J2EE サーバ] - [uCosminexus Navigation Developer の J2EE サーバ名称] をクリックし、[JVM] タブをクリックし、さらに [プロパティ] タブをクリックします。

[システムプロパティの設定] 画面が表示されます。

8. 次のとおりに設定します。

- [システムプロパティの定義] のいちばん下の行の [有効] のチェックボックスをチェックする
- [システムプロパティ] に次の値を設定する
ucnp.plugin.suspend.output.folder = サンプルの中断情報出力ディレクトリ

9. [追加] ボタンをクリックします。

有効	システムプロパティ	
<input checked="" type="checkbox"/>	java.security.auth.login.config=C:/Program Files/Hitachi...	削除
<input checked="" type="checkbox"/>	com.cosminexus.admin.auth.config=C:/Program Files/H...	削除
<input checked="" type="checkbox"/>	ucnp.plugin.suspend.output.folder=D:/uCNPWork	削除
<input type="checkbox"/>		削除
<input type="checkbox"/>		追加

J2EEサーバでは、次に示す接頭辞のキーを内部的に利用します。そのため、アプリケーションで利用するキーには、これらを接頭辞として利用しないで下さい。

- *ejbserver
- *webserver
- *appclient
- *j2eeserver
- *vbj

適用 リセット

10. [設定情報の配布] アンカーをクリックします。

[設定情報の配布] 画面が表示されます。

11. 論理サーバ名に uCosminexus Navigation Developer の J2EE サーバ名が表示されている行の [配布] チェックボックスをチェックします。

論理サーバが選択されます。

12. [選択して配布] ボタンをクリックします。

論理サーバの設定情報が、uCosminexus Navigation Developer の J2EE サーバに配布されます。

13. Cosminexus Management Server 経由で J2EE サーバを再起動します。

14. 中断再開プラグインをシステムで有効にします。

ユーザプロパティファイル (ucnp_user.properties) の ucnp.base.client.suspend.enable キーで、true に設定してください。ユーザプロパティファイルの設定方法については、マニュアル「Cosminexus V9 ナビゲーション プラットフォーム 導入から運用まで」を参照してください。

15. J2EE アプリケーションのインポートと開始を実行します。

J2EE アプリケーションのインポートと開始の実行方法については、「3.8 uCosminexus Navigation Developer の J2EE アプリケーションのインポートと開始」を参照してください。

付録 A.3 サンプルプラグインを使用するときの注意事項

サンプルプラグインを使用するときには、次の注意事項があります。

- サンプルプラグインにはエラー処理が含まれていません。必要に応じてエラー処理を追加してください。
- 十分な動作確認を実施して、エラーが発生しないことを確認してから使用してください。
- Windows XP 以外でプラグイン開発をする場合、サンプルプラグインを OS のインストールドライブ: ¥Program Files ディレクトリ以下などの作業ディレクトリにコピーするときに、Windows の管理者権限がないユーザが追加すると、ファイルがユーザフォルダにリダイレクトされてしまうことがあります。そのため、Windows の管理者権限があるユーザで追加してください。
- uCosminexus Navigation Developer のインストールディレクトリがデフォルトと異なる場合、サンプルプラグインのプロジェクトを Eclipse にインポートしようとするとき、Eclipse の問題ビューに次のエラーが表示されてプロジェクトのビルドに失敗します。

```
Project 'OutputGuideData' is missing required library: 'C:¥Program Files¥HITACHI¥uCNP¥Lib¥ucnpclasses.jar'
```

この場合は、ucnpclasses.jar のパスを設定し直してください。uCosminexus Developer, および uCosminexus Portal Framework のインストールパスがデフォルトと異なる場合は、同様の手順で j2ee-javax.jar, および portlet.jar のパスも編集してください。

パスを設定する手順を次に示します。

手順

1. Eclipse のプロジェクト・エクスプローラーでプロジェクトを選択します。
2. メニューから [ファイル] - [プロパティ] - [プラグイン名] を選択します。
プロパティ ダイアログが表示されます。
3. ダイアログで [Java のビルド・パス] を選択します。
4. [ライブラリ] タグを選択します。
5. ucnpclasses.jar を選択して、[編集] ボタンをクリックします。
Windows のファイル選択ダイアログが表示されます。
6. Windows のファイル選択ダイアログから uCosminexus Navigation Developer インストールディレクトリ¥lib に格納されている ucnpclasses.jar を選択して、[開く] ボタンをクリックします。
7. プロパティ ダイアログで [OK] ボタンをクリックします。

付録 B 検索用サンプル画面の使用法

ここでは、ナビゲーション プラットフォームが提供する検索用サンプル画面 (search_sample.jsp) の使用方法について説明します。なお、検索用サンプル画面を使用するには注意事項があります。詳細は「付録 B.2 検索用サンプル画面を使用するときの注意事項」を参照してください。

付録 B.1 検索用サンプル画面の使用手順

サンプル画面 (search_sample.jsp) を使用してメニュー領域に検索機能を追加する手順を次に示します。

手順

1. サンプル画面をカスタム画面用 JSP ファイルの格納ディレクトリにコピーします。

コピーするファイルとコピー先のディレクトリを次に示します。

コピーするファイル

ナビゲーション プラットフォームインストールディレクトリ¥sample¥search¥menu
¥search_sample.jsp

コピー先ディレクトリ

ポータルプロジェクトディレクトリ¥ucnp¥custom¥menu

2. ユーザプロパティファイル (ucnp_user.properties) を次のとおりに設定します。

- ucnp.base.server.custom.menu プロパティに「/ucnpBase/ucnp/custom/menu/search_sample.jsp」を指定する。
- ucnp.base.client.menu.frame.width プロパティに「400」を指定する。

ユーザプロパティファイルの詳細は、マニュアル「Cosminexus V9 ナビゲーション プラットフォーム 導入から運用まで」を参照してください。

3. Eclipse を起動して、カスタム画面をビルドします。

ビルド手順はプラグインの場合と同じです。手順の詳細は、「3.6.1 プラグインのビルド手順」を参照してください。

4. Eclipse で [ウィンドウ] - [ビューの表示] - [その他] - [Ant] - [Ant] を選択します。

[Ant] ビューが表示されます。

5. 「2.3 pluginSDK プロジェクトのインポート」でインポートしたプロジェクトから、build.xml を [Ant] ビューにドラッグ&ドロップします。

[Ant] ビューに [root] が追加されます。

6. [Ant] ビューの [+] アイコンをクリックします。

ターゲット一覧が表示されます。

7. ターゲット一覧の [stopEar] をダブルクリックします。

J2EE アプリケーションが停止します。

8. ターゲット一覧の [deleteEar] をダブルクリックします。

J2EE アプリケーションが削除されます。

9. 次の URL にアクセスして、Cosminexus Management Server を起動します。

http://ホスト名:ポート番号/mngsvr/

- ホスト名
開発環境の J2EE サーバマシンのホスト名または IP アドレスです。

- ポート番号

Cosminexus Management Server が使用するポート番号です。マニュアル「Cosminexus V9 ナビゲーション プラットフォーム 導入から運用まで」の手順どおりにセットアップした環境では、「28080」です。

10. Cosminexus Management Server にログインします。

運用管理ポータル画面が表示されます。

なお、マニュアル「Cosminexus V9 ナビゲーション プラットフォーム 導入から運用まで」の手順どおりにセットアップした環境では、ログイン時に入力する管理ユーザ ID が「admin」、パスワードが「ucnppass」です。

11. Cosminexus Management Server 経由で J2EE サーバを再起動します。

12. ターゲット一覧の [deployEar] をダブルクリックします。

J2EE アプリケーションがインポートされます。

13. ターゲット一覧の [startEar] をダブルクリックします。

J2EE アプリケーションが開始されます。

14. ナビゲーション プラットフォームにログインして、業務実行画面のメニュー領域に検索用のメニューが表示されていることを確認します。

付録 B.2 検索用サンプル画面を使用するときの注意事項

検索用のサンプル画面 (search_sample.jsp) を使用するときには、次の注意事項があります。

- 検索用サンプル画面にはエラー処理が含まれていません。必要に応じてエラー処理を追加してください。
- 十分な動作確認を実施して、エラーが発生しないことを確認してから使用してください。
- 検索用サンプル画面で検索を実行するためには、検索機能のセットアップが完了している必要があります。検索機能の追加セットアップ方法の詳細は、マニュアル「Cosminexus V9 ナビゲーション プラットフォーム 導入から運用まで」を参照してください。
- サンプル画面 (search_sample.jsp) では、特殊文字のエスケープ処理が省略されています。そのため、サンプル画面を流用して検索機能を追加する場合は、そのまま使用するのではなく、特殊文字のエスケープ処理を追加してから使用してください。
- サンプル画面 (search_sample.jsp) は、メニュー領域に検索機能を追加するためのサンプルです。このサンプル画面を流用して、新規画面に検索機能を追加する場合は、「4.2.1 ナビゲーション プラットフォームで提供している JavaScript 関数」で紹介している関数は使用できないため、該当箇所を削除・変更する必要があります。編集が必要な箇所を次に示します。
 - src="/ucnpBase/adportlets/ucnp/js/tools/UcnpCustomTools.js"を削除する。
 - <a>タグの href 属性の参照先を、ログイン用の URL などに置き換える。

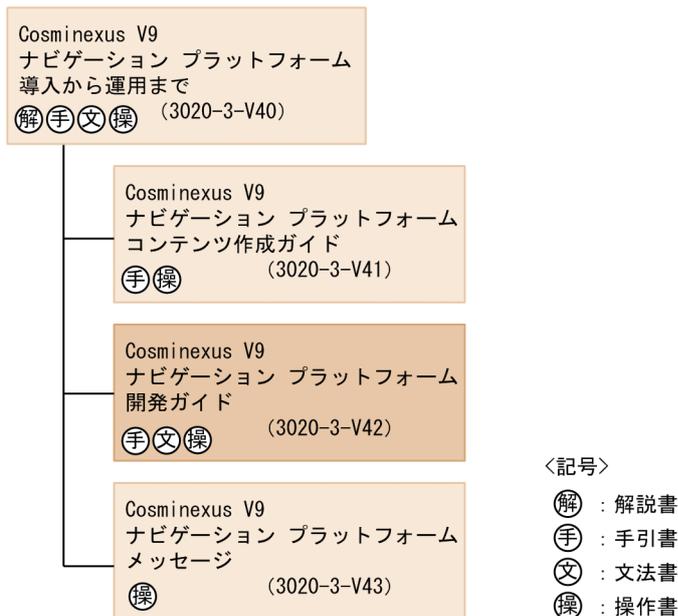
付録 C このマニュアルの参考情報

このマニュアルを読むに当たっての参考情報を示します。

付録 C.1 関連マニュアル

(1) ナビゲーション プラットフォームのマニュアル

ナビゲーション プラットフォームのマニュアルを次に示します。



- Cosminexus V9 ナビゲーション プラットフォーム 導入から運用まで (3020-3-V40)
 ナビゲーション プラットフォームの概要、機能、構築、および運用について説明しています。
 ナビゲーション プラットフォームの使用に当たっては、このマニュアルを最初にお読みください。
- Cosminexus V9 ナビゲーション プラットフォーム コンテンツ作成ガイド (3020-3-V41)
 ナビゲーション プラットフォームを使用して業務コンテンツを作成、変更、削除したり、コンテンツのアクセス権を管理したりする方法について説明しています。
- Cosminexus V9 ナビゲーション プラットフォーム 開発ガイド (3020-3-V42)
 このマニュアルです。ナビゲーション プラットフォームのプラグインおよびカスタム画面を開発する方法について説明しています。
- Cosminexus V9 ナビゲーション プラットフォーム メッセージ (3020-3-V43)
 ナビゲーション プラットフォームを使用しているときに出力されるメッセージについて説明しています。

(2) 構成ソフトウェアのマニュアル

ナビゲーション プラットフォームの構成ソフトウェアのマニュアルを次に示します。

- Cosminexus V9 アプリケーションサーバ システム構築・運用ガイド (3020-3-Y02)
 アプリケーションサーバを導入してシステムを構築・運用する方法について説明しています。

- **Cosminexus V9 アプリケーションサーバ機能解説 拡張編 (3020-3-Y08)**
アプリケーションサーバの機能の詳細について、アプリケーションの実装方法や実行環境に必要な設定などを含めて解説しています。このマニュアルでは、アプリケーションサーバ独自の拡張機能について説明しています。
- **Cosminexus V9 アプリケーションサーバ 機能解説 互換編 (3020-3-Y12)**
アプリケーションサーバの機能の詳細について、アプリケーションの実装方法や実行環境に必要な設定などを含めて解説しています。このマニュアルでは、旧バージョンのアプリケーションサーバで提供していた互換用の機能について説明しています。
- **Cosminexus V9 アプリケーションサーバ アプリケーション設定操作ガイド (3020-3-Y13)**
サーバ管理コマンドを使用した J2EE アプリケーションおよびリソースの操作について説明しています。
- **Cosminexus V9 アプリケーションサーバ 運用管理ポータル操作ガイド (3020-3-Y14)**
運用管理ポータルの画面および操作について説明しています。
- **Cosminexus V9 アプリケーションサーバ リファレンス コマンド編 (3020-3-Y15)**
システムを構築・運用するとき、またはアプリケーションを開発するとき使用するコマンドについて説明しています。
- **Cosminexus V9 アプリケーションサーバ アプリケーション開発ガイド (3020-3-Y20)**
アプリケーションの開発方法について説明しています。また、開発環境のセットアップ方法についても説明しています。
- **uCosminexus Portal Framework システム管理者ガイド (3020-3-H71)**
uCosminexus Portal Framework 全体の機能概要について説明しています。また、ポータル全体の管理者（システム管理者）が実施する作業（ポータルの構築方法、ポータルのカスタマイズ方法、ポートレットの登録方法、およびポータルの起動方法）について説明しています。

付録 C.2 このマニュアルでの表記

このマニュアルでは、製品および構成ソフトウェアの正式名称を次に示す略称で表記します。

表記	正式名称
Cosminexus	uCosminexus Application Server
	uCosminexus Developer
Eclipse	Eclipse Web Tools Platform

付録 C.3 英略語

このマニュアルでは、次に示す英略語を使用します。

表記	正式名称
EAR	Enterprise ARchive
ISO	International Organization for Standardization
J2EE	Java 2 Platform, Enterprise Edition
JavaVM	Java Virtual Machine

表記	正式名称
UTF	UCS Transformation Format
WAR	Web ARchive
WOW64	Windows On Windows 64
WTP	Eclipse Web Tools Platform
XML	eXtensible Markup Language

付録 C.4 KB (キロバイト) などの単位表記について

1KB (キロバイト), 1MB (メガバイト), 1GB (ギガバイト), 1TB (テラバイト) はそれぞれ $1,024$ バイト, $1,024^2$ バイト, $1,024^3$ バイト, $1,024^4$ バイトです。

付録 D 用語解説

(英字)

Cosminexus Management Server

構成ソフトウェアである uCosminexus Application Server または uCosminexus Developer の機能です。アプリケーションサーバや Web サーバの運用を管理します。

Web サーバ

Web ブラウザからのリクエスト受信、および Web ブラウザへのデータ送信に関連する処理を実行するサーバです。ナビゲーション プラットフォームの構成ソフトウェアである uCosminexus Application Server または uCosminexus Developer が、Web サーバの役割を持ちます。

WTP (Eclipse Web Tools Platform)

Java アプリケーションの開発環境です。

(ア行)

アプリケーションサーバ

Web ブラウザからのユーザの操作を処理し、データベースなどの業務システムに反映させるためのサーバです。ナビゲーション プラットフォームの構成ソフトウェアである uCosminexus Application Server または uCosminexus Developer が、アプリケーションサーバの役割を持ちます。

インポート

エクスポートされたデータを自環境に取り込むことです。
業務コンテンツ、およびアクセス権の設定情報をインポートできます。

エクスポート

業務コンテンツ、およびアクセス権の設定情報を ZIP 形式で出力することです。エクスポートしたデータは、ほかの環境にインポートできます。

(カ行)

ガイド

業務コンテンツのうち、業務で実施する個々の作業の説明のことです。

ガイドパーツ

ガイドを作成するための部品です。[Guide] パレットから選択して、ガイド領域に配置します。

開発環境

プラグインを開発するための環境です。uCosminexus Navigation Developer が必要です。

画面 ID

業務コンテンツを実行している画面を一意に識別する ID です。

関連線

ノードからノードへの遷移を示す矢印のことです。

キャッシュ

利用者が業務コンテンツのガイドで入力または設定した値のデータです。キャッシュを利用することで、プラグインは入力途中の状態を保存したり、入力した値をほかの画面やシステムに受け渡したりできるようになります。

業務コンテンツ

業務を遂行するために必要な情報のことです。業務コンテンツは、ナビゲーション プラットフォームの画面で作成、表示できます。

業務フロー

業務コンテンツのうち、業務の流れをフローチャート化した図のことです。

業務編集画面

ナビゲーション プラットフォームで、次の操作を実施するための画面です。

- 業務コンテンツの作成・変更・複製・削除
- 入出力プラグインの関連づけ
- 業務コンテンツに対するアクセス権の設定

コンテンツ管理者

業務コンテンツに対して管理者のアクセス権を持つユーザのことです。

コンテンツ編集者

業務コンテンツに対して編集者のアクセス権を持つユーザのことです。

(サ行)

システムプラグイン

外部プログラムまたはガイドパーツ間の連携に必要なプラグインのうち、ナビゲーション プラットフォームが提供しているプラグインのことです。

実行環境

業務コンテンツを利用して業務を実行する際に、利用者のアクセス先となる環境です。uCosminexus Navigation Platform が必要です。

(タ行)

ターミナルノード

業務の開始または終了を表すノードです。業務フローの最初と最後に配置します。ナビゲーション プラットフォームの画面では、丸で表示されます。

中断再開プラグイン

利用者が業務の途中にほかの業務を参照したり、ログアウトしたりしても、一時保存した状態から業務を再開できるようにするために必要なプラグインです。

中断情報

中断再開プラグインを使用している環境で、利用者が画面操作を中断したときに、一時保存される情報です。画面操作を再開すると、中断情報が復元されます。

(ナ行)

ナビゲーション プラットフォーム

uCosminexus Navigation Platform, uCosminexus Navigation Platform - User License, uCosminexus Navigation Platform - Authoring License, uCosminexus Navigation Developer の総称です。

入出力プラグイン

ナビゲーション プラットフォームの画面に入力された情報を処理したり、画面に出力する情報を決定したりするためのプラグインです。入出力プラグインを使用すると、ガイドに入力された値の整合性をチェックしたり、外部プログラムへ出力したり、ログとして記録したりできます。

入出力プラグイン用 XML ファイル

業務コンテンツと入出力プラグインを関連づけるために必要な情報（入力・出力パラメタ名）を定義するファイルです。このファイルで定義した内容は、業務編集画面のプラグインパーツに表示されます。

また、業務編集画面に表示される [Plugins] パレットのアイコンを任意のファイルに設定できます。

ノード

業務の工程を表す部品であり、業務フローに表示されます。ノードには、ターミナルノード、プロセスノード、および分岐合流ノードの 3 種類があります。

(ハ行)

パーツ

ナビゲーション プラットフォームの画面に配置する部品です。

ひな形プラグイン

プラグインの基となるファイルです。ひな形プラグインを編集して、プラグインを作成します。

ひな形プラグインは、Eclipse の Java プロジェクトの形式で作成されます。

プラグイン

ナビゲーション プラットフォームと外部プログラムとを連携したり、ナビゲーション プラットフォームの拡張機能を使用したりするためのユーザプログラムです。ナビゲーション プラットフォームでは、プラグインを開発するために必要なインタフェースを提供しています。

プラグイン情報プロパティファイル

ひな形プラグインを作成するための情報を定義するファイルです。

プラグインパーツ

業務コンテンツにプラグインを関連づけるための部品です。[Plugins] パレットから選択して、ガイド領域に配置します。

プロセスノード

業務の途中の工程を表すノードです。ターミナルノードの間には 1 個以上のプロセスノードが必要です。画面上では長方形で表示されます。

分岐合流ノード

業務の工程が分岐する場合に必要なノードです。ナビゲーション プラットフォームの画面では、ひし形で表示されます。

編集環境

業務コンテンツを作成、編集したり、業務コンテンツに対するアクセス権を設定したりするための環境です。uCosminexus Navigation Platform - Authoring License が必要です。

(マ行)

マッピング線

ガイドパーツとプラグインパーツとの入出力の関係を表す矢印のことです。

(ヤ行)

ユーザプラグイン

ナビゲーションプラットフォームが提供する API を使用して、開発者が開発したプラグインのことです。

索引

数字

- 09-50 より前のバージョンで開発したプラグインのインスタンス 69
- 09-50 より前のバージョンで開発したプラグインの実行順序とタイミング 68
- 09-50 より前のバージョンで開発したプラグインを使用する場合の制限 68
- 09-50 より前のバージョンで開発したプラグインを使用する場合の注意事項 68

A

- AND 検索 143
- API 一覧 (カスタム画面開発の場合) 134
- API 一覧 (プラグイン開発の場合) 90
- API リファレンス (カスタム画面開発の場合) 133
- API リファレンス (プラグイン開発の場合) 89
- application.xml 45

C

- contains メソッド 116
- cosminexus.xml の編集 43
- Cosminexus Management Server [用語解説] 161

D

- decodeHtmlPartParam メソッド 122
- deleteAll メソッド 119
- delete メソッド 117
- deserialize メソッド 127

E

- Eclipse 18
- execute メソッド 91

G

- GC (ガーベージコレクション) 65
- getContentId メソッド 136
- getContentName メソッド 136
- getGuideName メソッド 142
- getHitContentList メソッド 139
- getHitCounts メソッド [IHitContentData インタフェース] 136
- getHitCounts メソッド [IHitGuideData インタフェース] 142

- getHitCount メソッド [IHitContentData インタフェース] 135
- getHitCount メソッド [IHitGuideData インタフェース] 141
- getHitGuideDataList メソッド 137
- getLastUpdated メソッド 137
- getLoginId メソッド 112
- getMessageId メソッド [UCNPPluginException (入出力プラグイン例外クラス)] 129
- getMessageId メソッド [UCNPServerException (例外クラス)] 149
- getMessage メソッド 129
- getSearcher メソッド 148
- getWords メソッド 139

I

- IHitContentData 135
- IHitData 139
- IHitGuideData 141
- IioActionController (09-50 より前のバージョンとの互換用) 91
- IioPluginController 100
- inputFromNode メソッド 100
- ioaction.xml [入出力プラグイン用 XML ファイル] 33
- iPad 向け業務コンテンツで使用するプラグインを開発する場合の注意事項 67
- iPad 向け実行環境でメニュー領域をカスタマイズする場合の注意事項 79
- ISearcher 143
- ISuspendActionController 113
- ISuspendInfo 121
- IUCNPSession 112

J

- JavaScript 関数 73

L

- load メソッド 115
- LogoutActionUtil 146
- logout メソッド 146

M

- Map 形式 122

O

outputToNode メソッド 106

P

ParamConvertUtil 122
 param パラメタ [execute メソッド] 92
 param パラメタ [inputFromNode メソッド] 101
 param パラメタ [outputToNode メソッド] 107
 pluginSDK プロジェクトのインポート 20
 [Plugins] パレットおよびプラグインパーツのアイコンの作成 32

S

save メソッド 113
 SearcherConstants 147
 SearcherProvider 148
 search メソッド 143
 serialize メソッド 126
 SuspendInfoSerializeUtil 126

U

ucnp_menu_get_contentId (09-50 より前のバージョンとの互換用) [JavaScript 関数] 74
 ucnp_menu_onload [JavaScript 関数] 74
 ucnp_menu_show_flow [JavaScript 関数] 73
 ucnp.base.client.menu.frame.visible プロパティ 87
 ucnp.base.server.close.button.setting プロパティ 87
 ucnp.base.server.custom.menu プロパティ 86
 ucnp.base.server.logoutbutton.display プロパティ 87
 ucnp.base.server.search.accesscontrol.enable [新規画面] 87
 ucnp.base.server.search.accesscontrol.enable プロパティ [メニュー領域] 87
 ucnp.base.server.search.enable [新規画面] 87
 ucnp.base.server.search.enable プロパティ [メニュー領域] 87
 ucnp.options.param キーの指定例 (画面 ID 単位) (推奨) 103
 ucnp.request.options キー 101
 ucnp.request.options キーの指定例 (セッション単位) 101
 ucnpOptions パラメタ (session) 101
 UCNPPuginException(String message, Throwable cause)コンストラクタ 131

UCNPPuginException(String message)コンストラクタ 131
 UCNPPuginException (中断再開プラグイン例外クラス) 131
 UCNPPuginException (入出力プラグイン例外クラス) 129
 UCNPServerException 149
 ucnpUserData パラメタ 84
 uCosminexus Navigation Developer の J2EE アプリケーションのインポートと開始 50
 userplugin.id 25
 userplugin.java.package 25
 userplugin.name 25
 userplugin.server.controller.ioaction 25
 userplugin.server.controller.ioaction.type 25
 userplugin.server.controller.suspend 25
 userplugin.version 25

W

web.xml へのリソース定義の追加 43
 Web サーバ [用語解説] 161
 Web サーバリダイレクトの設定 48
 WTP (Eclipse Web Tools Platform) [用語解説] 161
 WTP のセットアップ 18

あ

アイコン 32
 アイコン [格納ディレクトリ] 32
 アイコン [作成形式] 32
 アプリケーションサーバ [用語解説] 161
 アプリケーション属性ファイル (application.xml) の編集 45

い

インポート [用語解説] 161

え

エクスポート [用語解説] 161

か

ガイドの検索結果格納インタフェース 141
 ガイドパーツ [用語解説] 161
 ガイド [用語解説] 161
 開発環境 [用語解説] 161
 開発 [新規画面] 81
 開発の概要 1

開発の準備 17
 開発の範囲 2
 確認ダイアログの表示有無の指定 40
 格納先 [JSP ファイル] 72
 格納先 [静的ファイル] 72
 カスタマイズ [業務実行画面のメニュー領域] 73
 カスタマイズに使用するファイルの格納先とパス 72
 カスタム画面開発の流れ 14
 カスタム画面の開発 71
 カスタム画面の取り込み 86
 カスタム画面のビルドとデバッグ 86
 画面 ID [用語解説] 161
 画面の開発後に必要な作業 86
 関連線 [用語解説] 161
 関連づけ 52

き

キャッシュ 3
 キャッシュ値 96
 キャッシュ [用語解説] 162
 旧入出力プラグイン 68
 業務コンテンツの検索結果格納インタフェース 135
 業務コンテンツ [用語解説] 162
 業務フロー [用語解説] 162
 業務編集画面 [用語解説] 162

け

検索インタフェース 143
 検索オブジェクト取得クラス 148
 検索機能を追加 [新規画面] 81
 検索機能を追加 [メニュー領域] 78
 検索クラス 147
 検索結果格納インタフェース 139
 検索対象 141
 検索ワード 134
 検索ワードの条件 144

こ

コンテンツ格納ディレクトリ 72
 コンテンツ管理者 [用語解説] 162
 コンテンツ編集者 [用語解説] 162

さ

サーバ処理実装インタフェース 100
 サンプルプラグインの使用法 152
 サンプルプラグインを使用するときの注意事項 155

し

システム環境変数の設定 19
 システム管理者 57
 システムプラグイン [用語解説] 162
 実行環境 [用語解説] 162
 実行を抑制するボタン種別の指定 40
 新規に画面を開発する場合の作業 15
 新規にプラグインを開発する場合の作業 9

せ

セッション情報利用インタフェース 112

そ

ソース例 [検索機能 (メニュー領域)] 79
 ソース例 [メニュー領域の表示形式] 76
 ソース例 [ログアウト後に画面を閉じる] 84
 ソース例 [ログアウト後にログイン画面を表示] 84
 ソート順序 143
 測定用の業務コンテンツの作成手順 64

た

ターミナルノード [用語解説] 162

ち

中断再開アクションコントローライタフェース 113
 中断再開プラグイン [用語解説] 162
 中断再開プラグイン (サンプル) の使用方法 153
 中断再開プラグインのカスタマイズ 41
 中断再開プラグイン例外クラス 131
 中断情報 [用語解説] 162
 中断情報インタフェース 121
 中断情報直列化ユーティリティクラス 126

て

定数フィールド [IHitContentData インタフェース] 135
 定数フィールド [SearcherConstants クラス] 147
 データベースとの接続処理の実装 44
 データベースとの接続処理の追加 43
 デバッグ 60

な

ナビゲーション プラットフォームのプラグインとは4
 ナビゲーション プラットフォーム [用語解説] 163

に

- 入出力アクションコントローラインタフェース
(09-50 より前のバージョンとの互換用) 91
- 入出力パラメタ変換ユーティリティクラス 122
- 入出力プラグイン (サンプル) の使用方法 152
- 入出力プラグインのカスタマイズ 32
- 入出力プラグイン用 XML ファイル [格納ディレクトリ] 34
- 入出力プラグイン用 XML ファイルとは 27
- 入出力プラグイン用 XML ファイルの編集 33
- 入出力プラグイン用 XML ファイル [用語解説] 163
- 入出力プラグイン [用語解説] 163
- 入出力プラグイン例外クラス 129
- 任意のデータを受け取る機能を追加 [新規画面] 84

の

- ノード [用語解説] 163

は

- パーツ [用語解説] 163
- パスの基点 87
- パラメタ説明用ツールチップの設定 39

ひ

- ひな形プラグイン作成コマンドの実行 28
- ひな形プラグインのカスタマイズ 32
- ひな形プラグインの作成 24
- ひな形プラグインのプロジェクトのインポート 31
- ひな形プラグイン [用語解説] 163

ふ

- プラグイン開発の流れ 8
- プラグインが実行する処理の実装 41
- プラグインが使用するメモリ使用量の計算 64
- プラグインが使用するメモリ使用量の測定手順 64
- プラグイン実行順序の指定 40
- プラグイン情報プロパティファイルの編集 24
- プラグイン情報プロパティファイル [用語解説] 163
- プラグインで受け渡せるデータについて 5
- プラグインに入力, または出力する値の詳細 55
- プラグインのインスタンスのライフサイクルについて
5
- プラグインの開発 23
- プラグインの削除 62
- プラグインの削除 (中断再開プラグインの場合) 62
- プラグインの削除 (入出力プラグインの場合) 62
- プラグインの実行タイミングについて 4

- プラグインのビルド 46
- プラグインのビルド手順 46
- プラグインのビルドに失敗したときの対処 47
- プラグインパーツ [用語解説] 163
- プラグインパーツを更新する 56
- プラグイン [用語解説] 163
- プラグインを修正する場合の作業 12
- プラグインを使用している業務コンテンツの設定情報を確認する 58
- プレビュー画面での実行有無の指定 40
- プレビュー時のプラグイン処理 111
- プロセスノード [用語解説] 163
- プロパティキーの詳細 25
- プロパティファイル作成時の注意 24
- プロパティファイルの記述形式 24
- 分岐合流ノード [用語解説] 163

へ

- 編集環境 [用語解説] 163

ほ

- ポータルプロジェクトのディレクトリ 72

ま

- マッピング線 [用語解説] 164
- マッピング線を引く (ガイドパーツとプラグインパーツをつなぐ) 52
- マニュアル [ナビゲーション プラットフォーム] 158
- マニュアル [ナビゲーション プラットフォーム以外] 158

め

- メニュー領域の表示形式を変更する 75
- メニュー領域をカスタマイズする場合の作業 14

ゆ

- ユーザプラグイン [用語解説] 164
- ユーザプロパティファイル 41
- ユーザプロパティファイルの設定 86
- ユーザプロパティファイルの設定 (中断再開プラグインの場合) 59

ら

- ライブラリ追加時の注意事項 22
- ライブラリの作成と設定 21
- ライブラリの追加 21
- ライブラリの配置場所 21

り

リソースアダプタの設定 43

れ

例外クラス 149

ろ

ログアウト機能を追加 [新規画面] 82

ログアウト処理クラス 146