

Hitachi Navigation Platform

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

3021-3-025-30

前書き

■ 対象製品

P-2943-4PA4 Hitachi Navigation Platform 10-50 (適用 OS : Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Windows Server 2016)

P-2943-4VA4 Hitachi Navigation Platform for Developers 10-50 (適用 OS : Windows 7 x64, Windows 8 x64, Windows 8.1 x64, Windows 10 x64)

P-292C-4PBL JP1/Navigation Platform 11-50 (適用 OS : Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Windows Server 2016)

P-292C-4VBL JP1/Navigation Platform for Developers 11-50 (適用 OS : Windows 7 x64, Windows 8 x64, Windows 8.1 x64, Windows 10 x64)

■ 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

HITACHI, Cosminexus, HiRDB, JP1, uCosminexus は、株式会社 日立製作所の商標または登録商標です。

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。

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。

RSA および BSAFE は、米国 EMC コーポレーションの米国およびその他の国における商標または登録商標です。

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。

その他記載の会社名、製品名などは、それぞれの会社の商標もしくは登録商標です。

Portions of this software were developed at the National Center for Supercomputing Applications (NCSA) at the University of Illinois at Urbana-Champaign.

Regular expression support is provided by the PCRE library package, which is open source software, written by Philip Hazel, and copyright by the University of Cambridge, England. The original software is available from <ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/>

This product includes software developed by Andy Clark.

This product includes software developed by Ben Laurie for use in the Apache-SSL HTTP server project.

This product includes software developed by Daisuke Okajima and Kohsuke Kawaguchi (<http://relaxngcc.sf.net/>).

This product includes software developed by IAIK of Graz University of Technology.

This product includes software developed by Ralf S. Engelschall <rse@engelschall.com> for use in the mod_ssl project (<http://www.modssl.org/>).

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

This product includes software developed by the Java Apache Project for use in the Apache JServ servlet engine project (<http://java.apache.org/>).

This product includes software developed by the University of California, Berkeley and its contributors.

This software contains code derived from the RSA Data Security Inc. MD5 Message-Digest Algorithm, including various modifications by Spyglass Inc., Carnegie Mellon University, and Bell Communications Research, Inc (Bellcore).

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。

HITACHI
Inspire the Next

株式会社 日立製作所



本製品は、米国 EMC コーポレーションの RSA BSAFE(R)ソフトウェアを搭載しています。



■ 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 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	
	Internet Explorer 11		Windows(R) Internet Explorer(R) 11	
Windows	Windows 7	Windows 7 x64	Microsoft(R) Windows(R) 7 Enterprise (64 ビット版)	
			Microsoft(R) Windows(R) 7 Professional (64 ビット版)	
			Microsoft(R) Windows(R) 7 Ultimate (64 ビット版)	
	Windows 8	Windows 8 x64	Windows(R) 8 Enterprise (64 ビット版)	
			Windows(R) 8 Pro (64 ビット版)	
	Windows 8.1	Windows 8.1 x64	Windows(R) 8.1 Enterprise (64 ビット版)	
			Windows(R) 8.1 Pro (64 ビット版)	
	Windows 10	Windows 10 x64	Windows(R) 10 Enterprise (64 ビット版)	
			Windows(R) 10 Pro (64 ビット版)	
	Windows Server 2008 R2			Microsoft(R) Windows Server(R) 2008 R2 Datacenter
				Microsoft(R) Windows Server(R) 2008 R2 Enterprise
				Microsoft(R) Windows Server(R) 2008 R2 Standard
	Windows Server 2012			Microsoft(R) Windows Server(R) 2012 Datacenter
				Microsoft(R) Windows Server(R) 2012 Standard
Windows Server 2012 R2			Microsoft(R) Windows Server(R) 2012 R2 Datacenter	

表記		製品名
Windows	Windows Server 2012 R2	Microsoft(R) Windows Server(R) 2012 R2 Standard
	Windows Server 2016	Microsoft(R) Windows Server(R) 2016 Datacenter
		Microsoft(R) Windows Server(R) 2016 Standard

■ 発行

2017年11月 3021-3-025-30

■ 著作権

All Rights Reserved. Copyright (C) 2014, 2017, Hitachi, Ltd.

変更内容

変更内容 (3021-3-025-30) Hitachi Navigation Platform 10-50, Hitachi Navigation Platform for Developers 10-50, JP1/Navigation Platform 11-50, JP1/Navigation Platform for Developers 11-50

追加・変更内容	変更箇所
Eclipse のセットアップ時の前提となる JDK のバージョンを 8 に変更しました。	3.1
プラグインのビルドに失敗したときに、[JRE] タブをクリックして、[VM 引数] に入力する値を「-XX:MaxMetaspaceSize=256m」に変更しました。	4.6.2

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。

- Hitachi Navigation Platform
- Hitachi Navigation Platform for Developers
- JP1/Navigation Platform
- JP1/Navigation Platform for Developers

■ 表記について

このマニュアルでは、上記プログラムプロダクトの名称を次のように表記しています。

表記	プログラムプロダクト名	
ナビゲーション プラットフォーム	Navigation Platform	Hitachi Navigation Platform
		JP1/Navigation Platform
	Navigation Platform for Developers	Hitachi Navigation Platform for Developers
		JP1/Navigation Platform for Developers

■ 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。
[] - []	-の前に示したメニューから、-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

前書き	2
変更内容	6
はじめに	7

1 開発の概要 13

1.1	開発の範囲	14
1.2	プラグイン開発の流れ	15

2 プラグインの概要 18

2.1	ナビゲーションプラットフォームのプラグインとは	19
2.1.1	プラグインの初期化・終了処理について	19
2.1.2	プラグインのセッションについて	20
2.2	プラグインの種類	21
2.3	入出力プラグインとは	22
2.3.1	入出力プラグインの実行タイミング	22
2.3.2	入出力プラグインの実行順序	22
2.3.3	入出力プラグインで受け渡せるデータ	23
2.3.4	入出力プラグインのインスタンスのライフサイクル	23
2.4	中断再開プラグインとは	26
2.4.1	中断再開プラグインの実行タイミング	26
2.4.2	中断再開プラグインのライフサイクル	26
2.5	カスタム画面プラグインとは	27
2.5.1	カスタム画面プラグインの実行タイミング	27
2.5.2	カスタム画面プラグインで受け取れるデータ	27
2.5.3	カスタム画面プラグインでエラーが発生したときの処理	27

3 開発の準備 28

3.1	Eclipse のセットアップ	29
3.2	pluginSDK プロジェクトのインポート	31
3.3	ライブラリの追加	32
3.3.1	ライブラリの配置場所	32
3.3.2	ライブラリの作成と設定	32
3.3.3	ライブラリ追加時の注意事項	33

4	プラグインの開発 34
4.1	ひな形プラグインの作成 35
4.1.1	プラグイン情報プロパティファイルの編集 35
4.1.2	ひな形プラグイン作成コマンドの実行 40
4.2	ひな形プラグインのプロジェクトのインポート 44
4.3	ひな形プラグインのカスタマイズ 45
4.3.1	入出力プラグインのカスタマイズ 45
4.3.2	中断再開プラグインのカスタマイズ 55
4.3.3	カスタム画面プラグインのカスタマイズ 56
4.3.4	カスタム画面で使用する JSP ファイルの作成 56
4.4	データベースとの接続処理の追加 58
4.4.1	リソースアダプタの追加 58
4.4.2	リソースアダプタの設定変更 59
4.4.3	リソースアダプタの削除 60
4.4.4	データベースとの接続処理のためのプラグインの設定 61
4.4.5	データベースとの接続処理の実装 62
4.5	プラグイン初期化・終了時の処理の実装 64
4.6	プラグインのビルド 65
4.6.1	プラグインのビルド手順 65
4.6.2	プラグインのビルドに失敗したときの対処 66
4.7	プラグインのデプロイ 68
4.8	業務コンテンツへの関連づけ (入出力プラグインの場合) 70
4.8.1	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) 70
4.8.2	プラグインに入力, または出力する値の詳細 73
4.8.3	プラグインパーツを更新する 74
4.8.4	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 75
4.9	ユーザプロパティファイルの設定 76
4.10	プラグインのデバッグ 77
4.10.1	プラグインをデバッグするための条件 77
4.10.2	プラグインのデバッグ手順 77
4.11	プラグインの削除 79
4.11.1	プラグインの削除手順 79
4.12	プラグインが使用するメモリ使用量の計算 80
4.12.1	測定用の業務コンテンツの作成手順 80
4.12.2	プラグインが使用するメモリ使用量の測定手順 80
4.13	J2EE サーバの設定項目の変更 83
4.13.1	プラグイン開発時に変更できる J2EE サーバの設定項目 83
4.13.2	J2EE サーバの設定項目を変更するためのファイルの格納場所 83
4.13.3	J2EE サーバ用ユーザプロパティファイル (usrconf.properties) の変更手順 83

4.13.4	J2EE サーバ用オプション定義ファイル (usrconf.cfg) の変更手順	86
4.14	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	90
4.15	旧バージョンで開発したプラグインの使用について	91
4.16	ユーザプラグインが利用する Java パッケージへのアクセス権の設定	92
5	API リファレンス (入出力プラグイン開発の場合)	93
5.1	API 一覧 (入出力プラグイン開発の場合)	94
5.2	IloPluginController (サーバ処理実装インタフェース)	95
5.2.1	inputFromNode メソッド	95
5.2.2	outputToNode メソッド	104
5.2.3	プレビュー時のプラグイン処理	110
5.3	IPluginInitializer (ユーザプラグインの起動 (初期化)・終了時の処理の実装インタフェース)	111
5.3.1	init メソッド	111
5.3.2	destroy メソッド	112
5.4	UCNPPluginUserException (ユーザプラグイン例外クラス)	113
5.5	ParamConvertUtil (入出力パラメタ変換ユーティリティクラス)	114
5.5.1	decodeHtmlPartParam メソッド	114
5.6	UCNPPluginException (入出力プラグイン例外クラス)	119
5.6.1	getMessage メソッド	119
5.6.2	getMessageId メソッド	119
5.7	IUCNPSession (セッション情報利用インタフェース)	121
5.7.1	getLoginId メソッド	121
6	API リファレンス (中断再開プラグイン開発の場合)	122
6.1	API 一覧 (中断再開プラグイン開発の場合)	123
6.2	ISuspendActionController (中断再開アクションコントローラインタフェース)	124
6.2.1	save メソッド	124
6.2.2	load メソッド	126
6.2.3	contains メソッド	128
6.2.4	delete メソッド	129
6.2.5	deleteAll メソッド	131
6.3	ISuspendInfo (中断情報インタフェース)	134
6.3.1	getContentSuspendedId メソッド	134
6.3.2	getWorkId メソッド	135
6.3.3	getContentName メソッド	135
6.3.4	getGroupName メソッド	136
6.3.5	getCurrentNodeName メソッド	136
6.4	SuspendInfoSerializeUtil (中断情報直列化ユーティリティクラス)	138
6.4.1	serialize メソッド	138

- 6.4.2 deserialize メソッド 139
- 6.5 UCNPluginException (中断再開プラグイン例外クラス) 141
- 6.5.1 UCNPluginException(String message)コンストラクタ 141
- 6.5.2 UCNPluginException(String message, Throwable cause)コンストラクタ 142

7 API リファレンス (カスタム画面プラグイン開発の場合) 143

- 7.1 API 一覧 (カスタム画面プラグイン開発の場合) 144
- 7.2 CustomWindowUtil (カスタム画面 URL 取得ユーティリティクラス) 145
 - 7.2.1 getCustomWindowUrl メソッド 145
- 7.3 LogoutActionUtil (ログアウト処理クラス) 147
 - 7.3.1 logout メソッド 147

付録 149

- 付録 A サンプルプラグインの使用方法 150
 - 付録 A.1 入出力プラグイン (サンプル) の使用方法 150
 - 付録 A.2 中断再開プラグイン (サンプル) の使用方法 151
 - 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 151
- 付録 B 入出力プラグイン開発でのポイント 153
 - 付録 B.1 マッピング線の有無による, 入出力プラグインの実行抑止 153
- 付録 C 旧バージョンからの移行について 155
 - 付録 C.1 旧バージョンで開発したプラグインを移行する手順 155
 - 付録 C.2 旧バージョンで開発したメニュー領域の移行について 157
 - 付録 C.3 旧バージョンで開発したカスタム画面 (新規画面) の移行について 158
 - 付録 C.4 旧バージョンでの J2EE サーバの設定について 158
- 付録 D このマニュアルの参考情報 159
 - 付録 D.1 関連マニュアル 159
 - 付録 D.2 このマニュアルでの表記 159
 - 付録 D.3 英略語 160
 - 付録 D.4 KB (キロバイト) などの単位表記について 160
- 付録 E 用語解説 161

索引 166

1

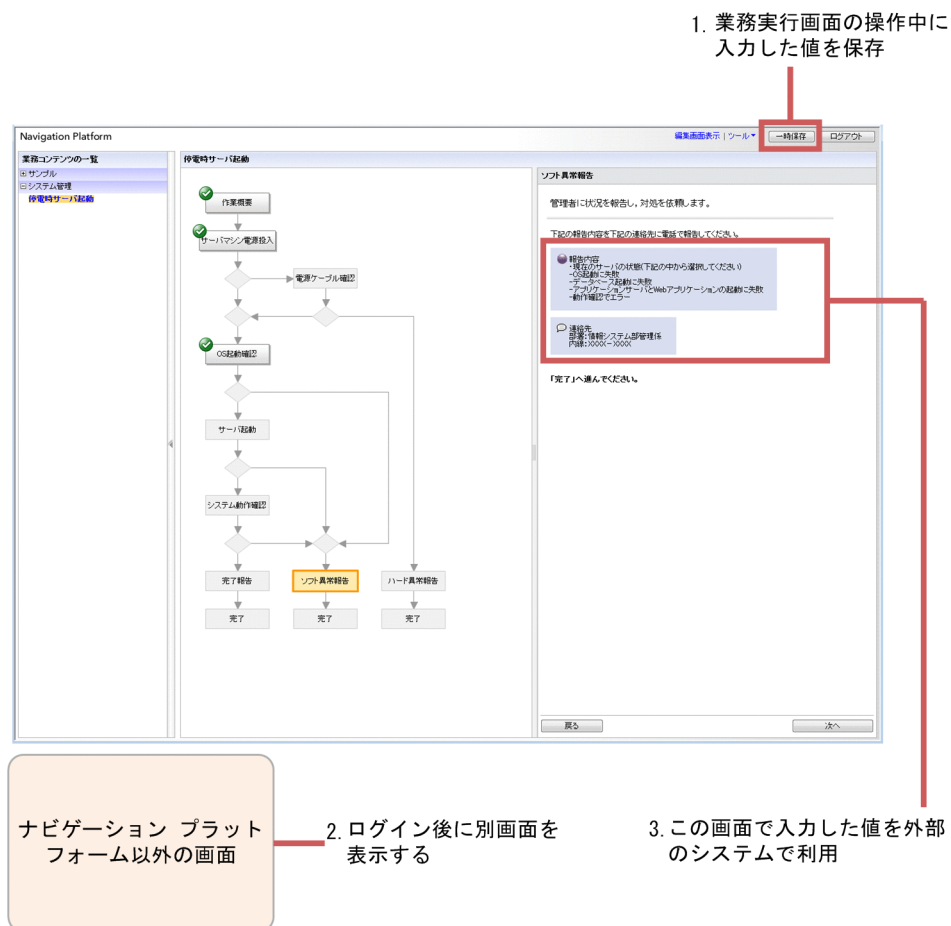
開発の概要

この章では、ナビゲーションプラットフォームで使えるプラグインの範囲と、開発作業の流れについて説明します。

1.1 開発の範囲

業務の内容に合わせたプラグインを開発すると、より便利にナビゲーションプラットフォームを利用できます。開発したプラグインによる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を次に示します。

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



この図に示したカスタマイズには、次に示すプラグインの開発が必要です。

1. 中断再開プラグイン
2. カスタム画面プラグイン
3. 入出力プラグイン

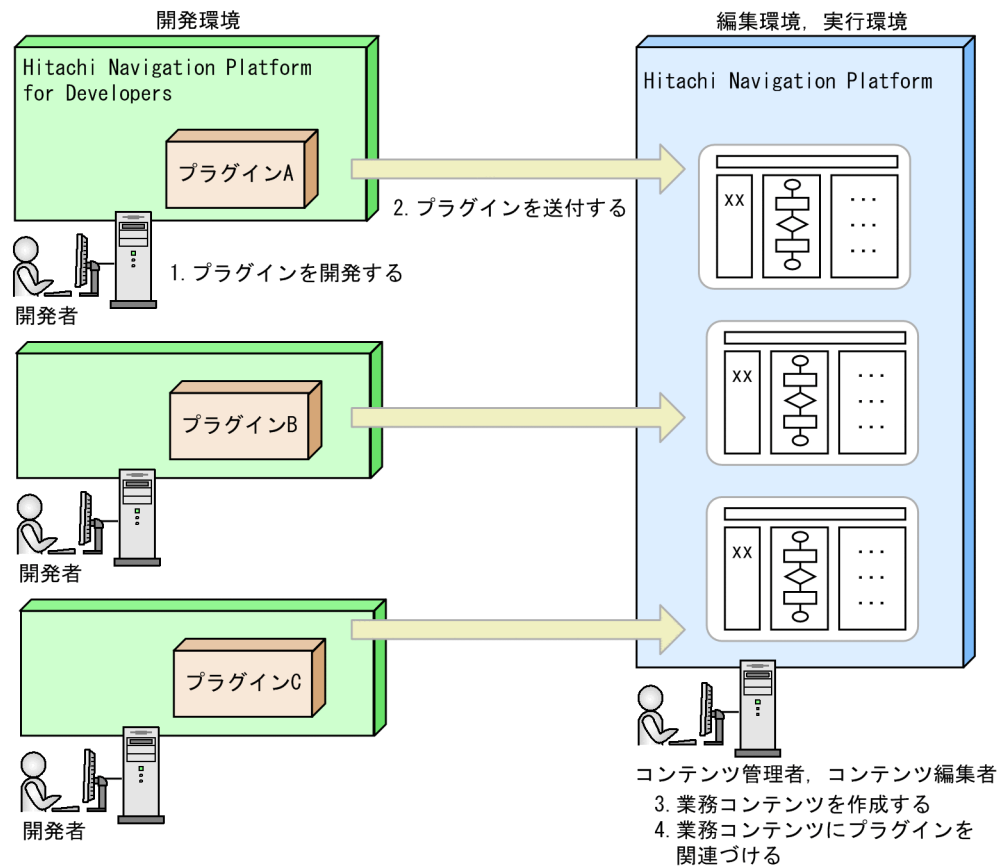
このマニュアルでは、これらのプラグインを使ったナビゲーションプラットフォームのカスタマイズについて説明します。

プラグインを開発しないで実現できるナビゲーションプラットフォームのカスタマイズについては、マニュアル「ナビゲーションプラットフォーム 導入から運用まで」のカスタマイズとユーザプロパティファイル (ucnp_user.properties) の説明を参照してください。

1.2 プラグイン開発の流れ

プラグイン開発の概要と流れを次の図に示します。

図 1-2 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. 開発者は、開発環境でプラグインを開発します。
プラグインの開発には、ナビゲーションプラットフォームが提供している Eclipse を使用します。
Eclipse を使用する場合は、事前にセットアップが必要です。
2. 開発したプラグインを編集環境と実行環境に送付します。
送付したプラグインは、システム管理者によって編集環境と実行環境に適用されます。
3. プラグイン適用後、コンテンツ管理者またはコンテンツ編集者は、編集環境で業務コンテンツを作成します。
4. 入出力プラグインの場合は、業務コンテンツを作成するときに、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いて、業務コンテンツにプラグインを関連づけます。
中断再開プラグインまたはカスタム画面プラグインの場合は、関連づけは不要です。

参考

編集環境を使用しないシステム構成の場合は、3.と4.の作業も開発者が開発環境で実施することがあります。システム構成については、マニュアル「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。業務コンテンツを作成する方法については、マニュアル「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。

次に、プラグイン開発の作業の内容と参照先を説明します。

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おく必要があります。また、ここで説明する作業は、すべて Windows の管理者権限を持つユーザで実行してください。

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（プラグイン名.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「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。

表 1-1 プラグイン開発の作業内容

順序	作業	入出力プラグイン	中断再開プラグイン	カスタム画面プラグイン	参照先
1	開発環境を準備する（Eclipse のセットアップ、ライブラリ追加など）	○	○	○	3 章
2	ひな形プラグインを作成する	○	○	○	4.1
3	ひな形プラグインのプロジェクトを Eclipse にインポートする	○	○	○	4.2
4	Eclipse を使用して、ひな形プラグインをカスタマイズ（編集）する	○	○	○	4.3.1, 4.4
5	プラグインに必要な処理を追加，実装する	△	△	△	4.4, 4.5
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する	○	○	○	4.6
7	Ant を使用して，プラグインの J2EE アプリケーション（プラグイン名.ear）をデプロイする	○	○	○	4.7
8	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る	○	×	×	4.8
9	ユーザプロパティファイルの見直し	○	○	△	4.9

順序	作業	入出力プラグイン	中断再開プラグイン	カスタム画面プラグイン	参照先
10	Eclipse のデバッガ機能を使用して、プラグインをデバッグします。	○	○	○	4.10

(凡例)

○：作業が必要

×：作業は不要

△：必要に応じて実施する

注

J2EE サーバの設定項目は、必要に応じて一部変更できます。詳細は、「[4.13 J2EE サーバの設定項目の変更](#)」を参照してください。

2

プラグインの概要

この章では、ナビゲーションプラットフォームで開発できるプラグインの種類とそれぞれの概要について説明します。

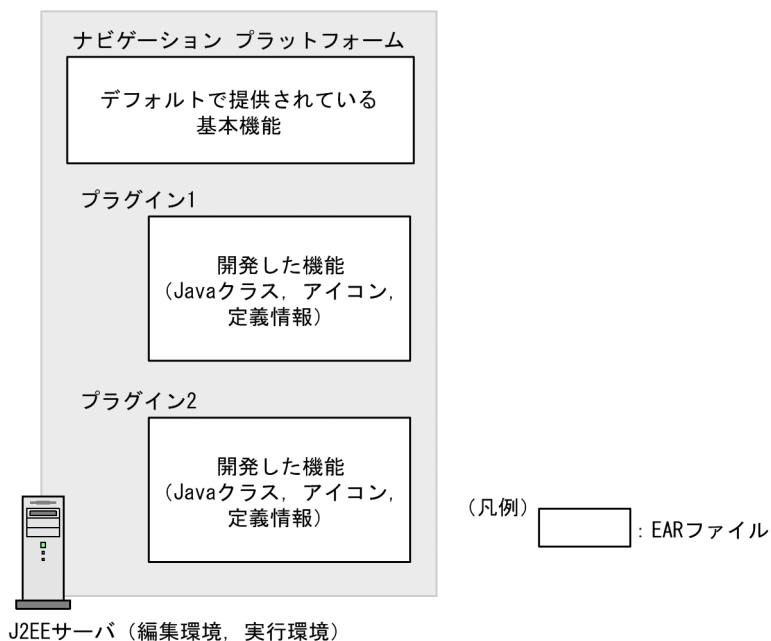
2.1 ナビゲーション プラットフォームのプラグインとは

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。このマニュアルでは、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。

プラグインの実体は、EAR ファイルです。開発者は、開発環境で Eclipse を使ってナビゲーション プラットフォームのプラグイン EAR ファイルを作成します。開発環境で作成した EAR ファイルを編集環境または実行環境のナビゲーション プラットフォーム（J2EE サーバ）にデプロイすることで、コンテンツ編集者やコンテンツ利用者がプラグインを使えるようになります。

ナビゲーション プラットフォーム（J2EE サーバ）にデプロイしたプラグインのことを、J2EE アプリケーションと呼ぶことがあります。プラグインとナビゲーション プラットフォームの EAR ファイルの関係を次に示します。

図 2-1 プラグインをナビゲーション プラットフォームに組み込んだ場合のイメージ



uCosminexus Navigation Developer 09-00 以前を使って開発したプラグインは、ナビゲーション プラットフォームへの組み込み方が異なるため、利用できません。uCosminexus Navigation Developer 09-50 以降を使って開発したプラグインは利用できますが、新しい形式のひな形プラグインに定義情報をコピーするなどの作業が必要になります。

2.1.1 プラグインの初期化・終了処理について

次のような場合は、IPluginInitializer インタフェースを使ってプラグインに初期化・終了処理を実装する必要があります。

- プラグインからデータベースに接続する

- プラグイン実行時に前処理が必要
- プラグイン停止時に後処理が必要

ナビゲーション プラットフォームが起動すると、プラグインも起動します。

(1) 初期化・終了処理クラスのインスタンス生成

プラグイン起動時に生成されます。

(2) 初期化・終了処理クラスのインスタンス破棄

サーブレット停止時に破棄されます。

2.1.2 プラグインのセッションについて

プラグイン内で取得した HttpSession オブジェクトに対して setAttribute メソッドを実行する場合、setAttribute メソッドの name パラメタに次の文字列で始まる名前を指定しないでください。

name パラメタの先頭に指定できない文字列

ucnp

java.

javax.

javax.portlet.

hptl

com.cosminexus

jp.co.hitachi.soft.portal

2.2 プラグインの種類

ナビゲーション プラットフォームで開発できるプラグインは、機能によって次のように分類されます。

- 入出力プラグイン
- 中断再開プラグイン
- カスタム画面プラグイン

入出力プラグインと中断再開プラグインは、利用者が画面で入力・選択した値（キャッシュ）を使って動作します。カスタム画面プラグインは、ナビゲーション プラットフォームから独立した別画面を表示するために使います。

2.3 入出力プラグインとは

入出力プラグインは、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。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ノードで利用したり、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。

2.3.1 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 2-1 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 (IloPluginController (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 (IloPluginController (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す。

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いように実装する必要があります。実装の詳細は「[5.2.3 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、この表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、次のことができます。

- 入出力プラグインの実行時に利用者にアラートを表示する
- 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する

メソッドの戻り値に設定する内容については「[5.2 IloPluginController \(サーバ処理実装インタフェース\)](#)」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。

2.3.2 入出力プラグインの実行順序

入出力プラグインが複数ある場合、ほかのプラグインの処理に依存しないように開発することを推奨します。

ただし、同じノードに複数の入出力プラグインを配置したときのために、入出力プラグイン用 XML ファイルで実行順序を指定しておき、指定の昇順で実行されるようにしておくことができます。実行順序はプラグインごとに指定するため、プラグイン内のメソッドの実行順序までは指定できません。

2.3.3 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ー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` メソッドのパラメー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に、パラメー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することで、次のような運用ができます。

- 利用者にアラートを表示できる
- 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できる

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「5.2 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

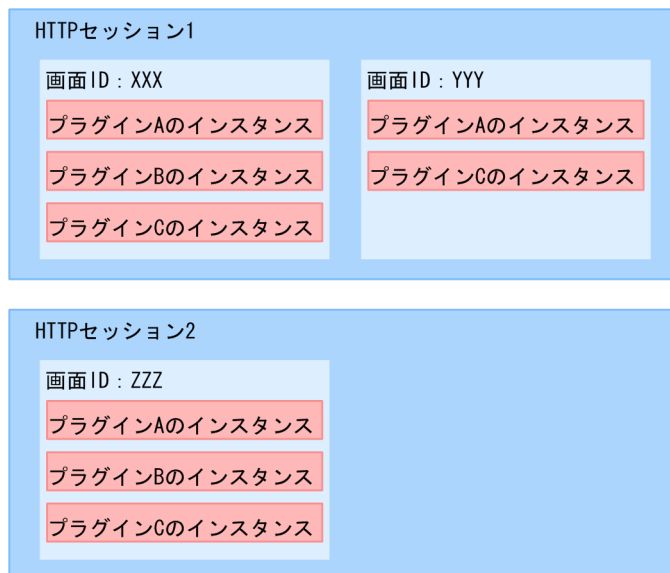
2.3.4 入出力プラグインのインスタンスのライフサイクル

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

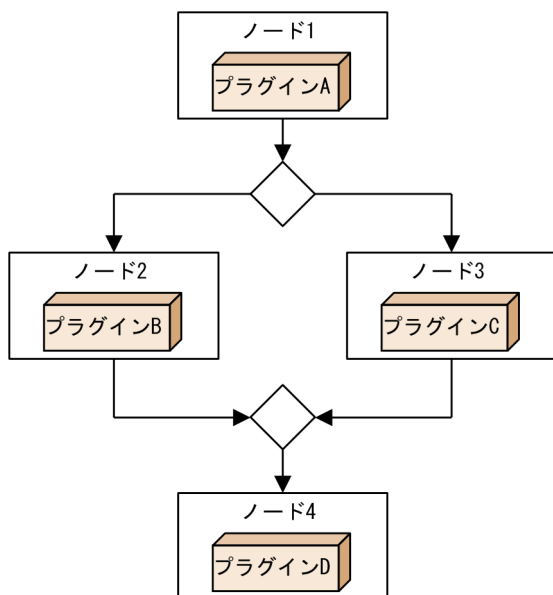
同一プラグイン ID の入出力プラグインのインスタンスは、一つの画面につき一つずつ保持されます。各インスタンスは、画面 ID をキーにして HTTP セッションに保持されます。入出力プラグインのインスタンスについて、次の図に示します。

図 2-2 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 2-3 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

入出力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 2-2 入出力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	[編集画面表示] 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「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「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。

2.4 中断再開プラグインとは

中断再開プラグインは、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。複数日にわたって実行する業務がある場合などは、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。

2.4.1 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
- 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

2.4.2 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。

- 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
- 利用者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
- 業務実行画面に表示される業務コンテンツの内容が変更されたとき
- コンテンツ管理者が、業務編集画面で業務コンテンツを削除したとき

2.5 カスタム画面プラグインとは

カスタム画面プラグインは、新規画面を開発するためのプラグインです。ログイン後にナビゲーションプラットフォームの画面のほかに、独自に作成した別画面も表示させたい場合は、カスタム画面プラグインを開発してください。

2.5.1 カスタム画面プラグインの実行タイミング

カスタム画面プラグインは、次のタイミングで実行されます。

- ucnpUserId パラメタを指定した URL でナビゲーションプラットフォームにアクセスしてログインしたとき
- CustomWindowUtil クラスの `getCustomWindowUrl` メソッドで取得した値に、ucnpUserId パラメタを指定した URL でナビゲーションプラットフォームにアクセスしてログインしたとき

2.5.2 カスタム画面プラグインで受け取れるデータ

カスタム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、最初に表示されるカスタム画面に対してだけ有効です。ログイン後にカスタム画面からほかの画面に遷移したあとは、無効となります。

2.5.3 カスタム画面プラグインでエラーが発生したときの処理

ログイン時にカスタム画面プラグインがない場合、またはカスタム画面プラグインが起動していない場合は、エラー画面に KDCZ10375-E のメッセージが表示されます。

3

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「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。

3.1 Eclipse のセットアップ

Eclipse をセットアップする手順を次に示します。

注意事項

JDK はバージョン 8 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、6 および 7 でビルドしたライブラリを使用することもできます。

次に示す手順でセットアップをすれば、Hitachi Navigation Platform for Developers に含まれているバージョン 8 の JDK を利用できるようになります。

1. Hitachi Navigation Platform for Developers, または JP1/Navigation Platform for Developers の CD-ROM から、Eclipse のアーカイブを任意のフォルダにコピーします。
2. コピーしたアーカイブを解凍して、Navigation Platform for Developers インストールディレクトリ以外のフォルダに配置します。
配置したフォルダのことを、以降の手順では「Eclipse のインストールディレクトリ」と表記します。
3. システム環境変数 Path に次のディレクトリを追加します。
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;
4. Eclipse のインストールディレクトリ直下にある eclipse.ini ファイルを編集します。

追加する内容 1

-vmargs よりも前の行に次の設定を追加してください。

-vm

Navigation Platform for Developers インストールディレクトリ¥PP¥uCPSB¥jdk¥bin
¥javaw.exe

追加する内容 2

-vmargs よりもあとの行に次の設定を追加してください。

-XX:MaxMetaspaceSize=128MB 以上の値

-Djava.endorsed.dirs=*Navigation Platform for Developers* インストールディレクトリ¥PP
¥uCPSB¥jaxp¥lib

編集例

-vm

C:¥Program Files¥Hitachi¥HNP¥PP¥uCPSB¥jdk¥bin¥javaw.exe

-vmargs

-XX:MaxMetaspaceSize=128m

-Djava.endorsed.dirs=C:¥Program Files¥Hitachi¥HNP¥PP¥uCPSB¥jaxp¥lib

5. eclipse.exe を実行して、Eclipse が起動することを確認します。

このとき、ワークスペースに Navigation Platform for Developers インストールディレクトリ以下を指定しないよう注意してください。

3.2 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. [General] - [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 [*Navigation Platform for Developers* 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。

3.3 ライブラリの追加

この節では、プラグインで利用される 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

必ず 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。

3.3.1 ライブラリの配置場所

プラグイン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。

プラグイン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¥pluginSDK¥plugin¥プラグイン ID¥WEB-INF¥lib

J2EE サーバ全体でライブラリを使用する場合

システム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。この場合、開発環境、実行環境の両方にライブラリを配置する必要があります。

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¥usrlib¥sys

3.3.2 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、次のライブラリをクラスパスに設定して、ビルドしてください。

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¥syslib¥ucnpsys.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、**[3.3.1 ライブラリの配置場所]** で説明されているディレクトリに格納します。
プラグインだけでライブラリを使用する場合は、以降の手順は不要です。
J2EE サーバ全体でライブラリを使用する場合は次の手順に進んでください。
2. J2EE サーバのシステムクラスパス指定に、ライブラリの絶対パスを追加します。
3. ナビゲーション プラットフォームを再起動します。

J2EE サーバ全体でライブラリを使用する場合は、開発環境だけでなく、実行環境の J2EE サーバでも手順 1.~手順 3.を実施してください。

(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。

3.3.3 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、ucnp で始まる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。
大文字と小文字は区別されないため、「UCNP」「uCNP」などで始まるファイル名も使用できません。
- ライブラリの配置場所には、開発に必要なライブラリだけを格納してください。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。
- ライブラリは、Windows の管理者権限があるユーザで追加してください。管理者権限がないユーザでライブラリを追加すると、ユーザフォルダにリダイレクトされてしまい、正常に動作しません。

4

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。

4.1 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。

4.1.1 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。ひな形プラグイン作成コマンドを実行すると、プラグイン情報プロパティファイルに定義した情報に基づいたファイルやフォルダが生成されます。

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。

```
Navigation Platform for Developers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) すべてのプラグインで使用する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のうち、すべてのプラグインに必要なプロパティキーの詳細を説明します。

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。
- 入出力プラグインまたはカスタム画面プラグインの場合、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

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

使用できないプラグイン名

「ucnp」で始まるプラグイン名（大文字と小文字は区別しない）

env

AppName

指定できる文字列長

1～31 バイト

指定する値の例

userplugin

userplugin.type

プラグインの種類を指定します。

使用できる文字

入出力プラグインの場合：TYPE_IO

中断再開プラグインの場合：TYPE_SUSPEND

カスタム画面プラグインの場合：TYPE_WINDOW

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

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

使用できないパッケージ名

「jp.co.hitachi.soft.ucnp」から始まるパッケージ名

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

(4) 入出力プラグインで使用するプロパティキーの詳細

userplugin.type プロパティキーに TYPE_IO を指定した場合に設定が必要なプロパティキーを次に示します。

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。「4.1.2 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

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。「4.1.2 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

(5) 中断再開プラグインで使用するプロパティキーの詳細

userplugin.type プロパティキーに TYPE_SUSPEND を指定した場合に設定が必要なプロパティキーを次に示します。

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

(6) 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 = example.inputdata
userplugin.name = InputData
userplugin.type = TYPE_IO
userplugin.version = 01.00
userplugin.java.package = com.example.inputdata
```

```
userplugin.server.controller.ioaction = InputDataController
userplugin.server.controller.ioaction.type = example.inputdata
```

中断再開プラグインの場合

```
userplugin.id = ucnplugin.suspend
userplugin.name = Suspend
userplugin.type = TYPE_SUSPEND
userplugin.version = 01.00
userplugin.java.package = com.example.suspend
userplugin.server.controller.suspend = SuspendActionController
```

カスタム画面プラグインの場合

```
userplugin.id = example.contentslist
userplugin.name = ContentsList
userplugin.type = TYPE_WINDOW
userplugin.version = 01.00
userplugin.java.package = com.example.customwindow
```

4.1.2 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。

```
Navigation Platform for Developersインストールディレクトリ¥pluginSDK¥bin¥ncreateplg.bat△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、*Navigation Platform for Developers* 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。

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。下線で示されているファイルだけを編集してください。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です。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、「4.1.1 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。

Navigation Platform for Developersインストールディレクトリ¥pluginSDK¥plugin

```
└─ プラグインID
  ├── .project
  ├── .classpath
  ├── build.xml
  ├── plugin.properties
  ├── ucnp_sdkversion.properties
  ├── js
  ├── dd
  │   ├── META-INF
  │   │   ├── application.xml
  │   │   └── cosminexus.xml
  ├── images
  │   └── sample icon.gif
  ├── WEB-INF
  │   ├── plugin.xml
  │   └── web.xml
  ├── lib
  ├── conf
  │   ├── ioaction.xml
  │   ├── 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
  └── 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
      │   │   └── PluginInitializer.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。下線で示されているファイルだけを編集してください。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です。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、「4.1.1 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。

Navigation Platform for Developersインストールディレクトリ¥pluginSDK¥plugin

```
└─ ucnp.plugin.suspend
  ├── .project
  ├── .classpath
  ├── build.xml
  ├── plugin.properties
  ├── ucnp_sdkversion.properties
  ├── js
  ├── dd
  │   ├── META-INF
  │   │   ├── application.xml
  │   │   └── cosminexus.xml
  ├── WEB-INF
  │   ├── plugin.xml
  │   └── web.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
            └─PluginInitializer.java

```

(4) ひな形プラグイン作成コマンドの実行結果（カスタム画面プラグインの場合）

カスタム画面プラグインの場合のディレクトリ構成を次に示します。下線で示されているファイルだけを編集してください。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です。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、「4.1.1 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。

```

Navigation Platform for Developersインストールディレクトリ¥pluginSDK¥plugin
└─プラグインID
   ├─.project
   ├─.classpath
   ├─build.xml
   ├─plugin.properties
   ├─ucnpsdkversion.properties
   ├─js
   ├─jsp
   │   └─sys
   │       └─ucnpCustom.jsp
   │           └─プラグイン名.jsp
   ├─dd
   │   └─META-INF
   │       └─application.xml
   │           └─cosminexus.xml
   └─WEB-INF
       ├─plugin.xml
       └─web.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
       └─lib
           └─src
               └─Javaパッケージ名
                   └─controller
                       └─PluginInitializer.java

```

(5) 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。

- *Navigation Platform for Developers* 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。
- 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。
- ひな形プラグイン作成コマンドで作成する Java ソースファイルのエンコードは、*Navigation Platform for Developers* 10-00 以前と 10-10 以降で異なります (10-00 以前は Shift_JIS, 10-10 以降は UTF-8)。そのため、10-00 以前に作成した Java ソースファイルを流用する場合は、エンコードの違いに注意してください。

4.2 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。

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. [General] - [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「[4.1 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。
- 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。

4.3 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。

4.3.1 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。

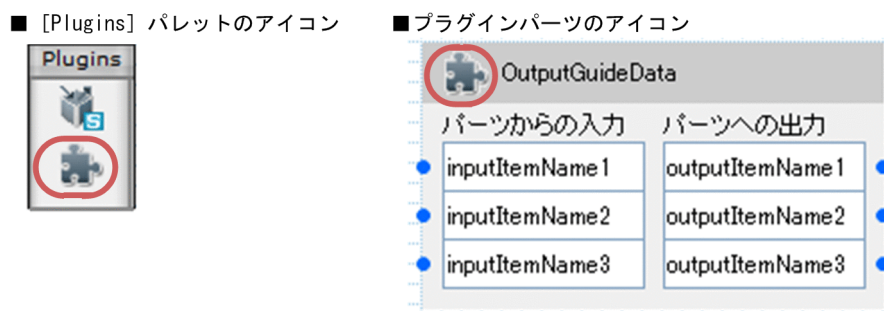
1. [Plugins] パレットおよびプラグインパーツのアイコンの作成
2. 入出力プラグイン用 XML ファイルの編集
3. パラメタ説明用ツールチップの設定
4. プラグイン実行順序の指定
5. 確認ダイアログの表示有無の指定
6. プレビュー画面での実行有無の指定
7. 実行を抑止するボタン種別の指定
8. プラグインが実行する処理の実装

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 4-1 [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、ナビゲーションプラットフォーム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。

*Navigation Platform for Developers*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 4-1 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、ナビゲーション プラットフォーム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、ナビゲーション プラットフォーム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

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 4-2 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.OutputGuideData" name="OutputGuideData" version="2.0">
  <iopart id="io_part_1">
    <iotype name="sample.OutputGuideData" />
    <iconURL url="/ucnpPlugins/OutputGuideData/images/ogdIcon.gif"/> (1)
    <priority value="700" />
    <execConfirm value="false" />
    <execPreview value="true" />
    <disableButtonType value="show_next_page" />
    <parameters type="title">
      <param name="OutputGuideData" description="description of OutputGuideData"/>
    </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

ポイント

いったん業務コンテンツに関連づけたプラグインの入出力プラグイン用 XML ファイル (ioaction.xml) を変更すると、業務コンテンツをエクスポートしてプラグインを入れ替えるなどの作業が必要になります。入出力プラグインを設計するときは、あとで入出力プラグイン用 XML ファイルを修正しなくて済むように、各項目の設定内容が業務コンテンツの運用に合ったものになるようにしてください。

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。

*Navigation Platform for Developers*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、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。太字の部分を編集します。

```
<?xml version="1.0" encoding="UTF-8"?>
<ioaction xmlns="http://model.xml.ioaction.navi.plugin.ucnp.soft.hitachi.co.jp"
id="sample.OutputGuideData" name="OutputGuideData" version="2.0">
  <iopart id="io_part_1">
    <iotype name="sample.OutputGuideData" />
    <iconURL url="/ucnpPlugins/OutputGuideData/images/ogdIcon.gif"/>
    <priority value="700" />
    <execConfirm value="false" />
    <execPreview value="true" />
    <disableButtonType legacy="false" value="show_next_page" />
    <parameters type="title">
      <param name="OutputGuideData" description="description of OutputGuideData"/>
    </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 4-2 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	<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 ([/ ucnpPlugins/プラグイン名/images/] 部分を含む)

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)	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 (「/ucnpPlugins/プラグイン名/images/」 部分を含む)
2	<priority>タグの value 属性	<p>入出力プラグインの実行順序を指定します。</p> <p>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。</p>	<p>次に示す半角英数字の値</p> <ul style="list-style-type: none"> 100~500 600~900 	—
3	<execConfirm>タグの value 属性	<p>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に通知したい場合などに指定してください。</p> <p>execConfirm 要素を省略した場合、false が仮定されます。</p>	<ul style="list-style-type: none"> true 表示する false 表示しない 	—
4	<execPreview>タグの value 属性	<p>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</p> <p>execPreview 要素を省略した場合、true が仮定されます。</p>	<ul style="list-style-type: none"> true 実行する false 実行しない 	—
5	<disableButtonType>タグの legacy 属性	<p>入出力プラグインの実行抑止について、互換性を持たせるかどうかを指定します。true を指定した場合、ナビゲーションプラットフォームのバージョン 10-10 より前と同じように抑止します。</p> <p>disableButtonType 要素を省略した場合、true が仮定されます。</p>	<ul style="list-style-type: none"> true 互換あり false 互換なし 	—
6	<disableButtonType>タグの value 属性	<p>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</p> <p>legacy 属性を false にすると、ボタンクリックで遷移する場合と、直接ノードを選択して遷移する場合とを区別して実行抑止できます。複数指定する場合、半角コンマ区切りで指定します。</p> <p>disableButtonType 要素を省略した場合、空文字が仮定されます。</p>	<ul style="list-style-type: none"> start show_next_page show_previous_page show_complete_page forward_jump legacy 属性の値が false の場合だけ back_jump legacy 属性の値が false の場合だけ 	—

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
7	<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
8	<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}
9	<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
10	<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}
11	<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、[「5.2 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、[「5.2 IIOPluginController \(サーバ処理実装インタフェース\)」](#)を参照してください。

4. プラグインの開発

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個所を次に示します。

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 4-3 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、「5.2 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。

表 4-3 入出力プラグインの実行を抑止するボタン種別

項番	互換オプション (legacy 属性の値)	実行抑止するボタン種別の設定値 (value 属性の値)	入出力プラグインの実行を抑止する操作
1	false	start	ターミナルノード (開始) からプロセスノードに遷移する。または、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する。
2		show_next_page	ボタンをクリックし、次のノードに遷移する。ただし、遷移先の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する場合は、「show_complete_page」になるため、注意すること。
3		show_previous_page	ボタンをクリックすることで前のノードに遷移する。
4		show_complete_page	[完了] ボタンが表示されているノードに遷移する。または、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する。
5		back_jump	直接ノードを選択することで次のノードに遷移する。
6		forward_jump	直接ノードを選択することで前のノードに遷移する。
7	true	start	ターミナルノード (開始) からプロセスノードに遷移する。または、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する。
8		show_next_page	ボタンをクリックまたは直接ノードを選択して遷移する。ただし、遷移先の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する場合は、「show_complete_page」になるため、注意すること。

項番	互換オプション (legacy 属性の値)	実行抑止するボタン種別の設定値 (value 属性の値)	入出力プラグインの実行を抑止する操作
9	true	show_previous_page	ボタンをクリックする、または直接ノードを選択することで前のノードに遷移する。
10		show_complete_page	[完了] ボタンが表示されているノードに遷移する。または、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する。

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

入出力プラグインにサーバ処理を実装する場合は、プラグイン情報プロパティファイルの `userplugin.server.controller.ioaction` キーに指定したクラス名のクラスに実装してください。実装する内容は、ノード遷移時の処理内容です。

プラグインの初期化終了処理が必要な場合は、「[4.5 プラグイン初期化・終了時の処理の実装](#)」を参照してください。

4.3.2 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

「[4.9 ユーザプロパティファイルの設定](#)」を参照して、ユーザプロパティファイルを編集してください。

(2) プラグインが実行する処理の実装

中断再開プラグインに中断情報の参照・更新処理を実装する場合は、プラグイン情報プロパティファイルの `userplugin.server.controller.suspend` キーに指定したクラス名のクラスに実装してください。

プラグインの初期化終了処理が必要な場合は、「4.5 プラグイン初期化・終了時の処理の実装」を参照してください。

4.3.3 カスタム画面プラグインのカスタマイズ

ここでは、カスタム画面プラグインのカスタマイズ手順について説明します。

カスタム画面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. カスタム画面用の JSP ファイルの編集
2. プラグインが実行する処理の実装

(1) カスタム画面用の JSP ファイルの編集

「4.3.4 カスタム画面で使用する JSP ファイルの作成」を参照して、カスタム画面用の JSP ファイルに必要な処理を実装してください。

(2) プラグインが実行する処理の実装

プラグインの初期化終了処理が必要な場合は、「4.5 プラグイン初期化・終了時の処理の実装」を参照してください。

4.3.4 カスタム画面で使用する JSP ファイルの作成

プラグイン情報プロパティファイルにカスタム画面プラグインの情報を定義して、ひな形プラグイン作成コマンドを実行すると、カスタム画面用の JSP ファイル (ucnpCustom.jsp) が生成されます。ucnpCustom.jsp は編集しないでください。編集用の JSP ファイルは、次のディレクトリに生成されます。

Navigation Platform for Developers インストールディレクトリ¥pluginSDK¥plugin¥プラグイン ID¥jsp ¥プラグイン名.jsp

編集用の JSP ファイル (プラグイン名.jsp) は、ucnpCustom.jsp にインクルードされます。ナビゲーションプラットフォームは ucnpCustom.jsp をロードして処理を実行するため、編集用の JSP ファイルではアプリケーションクラスローダのレイヤに配置したライブラリは使用できません。システムクラスローダのレイヤに配置したライブラリを使用してください。

編集用の JSP ファイル (プラグイン名.jsp) で指定できる項目を次に示します。

表 4-4 カスタム画面で使用する JSP ファイルで指定できる項目

項番	分類	項目	説明
1	タグ	ディレクティブ	次の形式でディレクティブを指定します。 <%@ ディレクティブ%>

項番	分類	項目	説明
2	タグ	スクリプトレット	次の形式でスクリプトレットを指定します。 <% Java コード %>
3		式	次の形式で式を指定します。 <%= 式 %>
4		コメント	次の形式でコメントを指定します。 <%-- コメント --%>
5	ディレクティブ	page	JSP ファイルのエンコーディング、Java の import 文などを定義します。カスタム画面用の JSP ファイル (ucnpCustom.jsp) と異なるエンコーディングを使用する場合は、pageEncoding 属性を指定してください。また、次の属性は指定しないでください。 contentType 属性 JSP ファイル実行時に自動的に ["text/html; charset=UTF-8"] が適用されます。 language 属性 JSP ファイル実行時に自動的に ["java"] が適用されます。
6		include	テキストファイルまたは JSP ファイルなどの別ファイルをインクルードします。
7	暗黙オブジェクト	request	javax.servlet.http.HttpServletRequest クラスのオブジェクト変数です。
8		response	javax.servlet.http.HttpServletResponse クラスのオブジェクト変数です。
9		session	javax.servlet.http.HttpSession クラスのオブジェクト変数です。

4.4 データベースとの接続処理の追加

プラグインにデータベースとの接続処理を追加するには、Cosminexusのリソースアダプタ「DB Connector」をJ2EEサーバに設定します。必要な作業を次に示します。

1. リソースアダプタの設定（追加，設定変更，削除）
2. プラグインの設定
3. データベースとの接続処理の実装

また，接続対象となるデータベースとリソースアダプタは次のとおりです。

データベース

HiRDB Single Server 08-00 以降

リソースアダプタ

DB_Connector_for_HiRDB_Type4

ナビゲーションプラットフォームをセットアップするときにインポートされているリソースアダプタです。

4.4.1 リソースアダプタの追加

リソースアダプタを追加するには，Connector 属性ファイルが必要です。手順を次に示します。

1. 次のとおりにコマンドを実行してリソースアダプタの一覧を表示し，「DB_Connector_for_HiRDB_Type4」というリソースアダプタがないことを確認します。

```
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¥PP¥uCPSB¥CC¥admin¥bin¥cjlistrar
uCNP_J2EE
```

2. 次のとおりにコマンドを実行して「DB_Connector_for_HiRDB_Type4」リソースアダプタをデプロイします。

```
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¥PP¥uCPSB¥CC¥admin¥bin
¥cjdeployrar uCNP_J2EE -resname DB_Connector_for_HiRDB_Type4
```

3. リソースアダプタの一覧を再度表示して，「DB_Connector_for_HiRDB_Type4」リソースアダプタが追加されたことを確認します。
4. 次のとおりにコマンドを実行して追加したリソースアダプタのConnector 属性ファイルを取得します。

```
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¥PP¥uCPSB¥CC¥admin¥bin
¥cjgetrarprop uCNP_J2EE -resname DB_Connector_for_HiRDB_Type4 -c "C:¥temp
¥TP_Connector_for_HiRDB_Type4.xml"
```

"C:¥temp¥TP_Connector_for_HiRDB_Type4.xml"の部分はConnector 属性ファイル（拡張子は.xml）の取得先ファイルパスです。任意に変更できます。

なお、すでに一度 Connector 属性ファイルを作成している場合は、この手順は不要です。

5. Connector 属性ファイルを編集して、接続処理を追加したいデータベースの情報を指定します。

データベースの設定値に従い、次に示すタグに値を設定してください。

タグ	説明
display-name	リソースアダプタの名称
description	データベースのポート番号
DBHostName	データベースの IP アドレス/ホスト名
encodeLang	データベースの文字コードに対応する文字セット
User	データベース接続ユーザ名
Password	データベース接続ユーザ名のパスワード

なお、display-name タグを編集するときは、次の規則に従ってください。

- ucnp で始まる文字列を指定しない（大文字と小文字は区別されません）
- ほかのリソースアダプタと同じ値を指定しない

なお、すでに一度 Connector 属性ファイルを作成している場合は、この手順は不要です。

6. 次のとおりにコマンドを実行して追加したリソースアダプタに Connector 属性ファイルの編集内容を反映します。

```
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¥PP¥uCPSB¥CC¥admin¥bin  
¥cjsetrarprop uCNP_J2EE -resname リソースアダプタの表示名 -c "C:¥temp  
¥TP_Connector_for_HiRDB_Type4.xml"
```

リソースアダプタの表示名は、「DB_Connector_for_HiRDB_Type4」または Connector 属性ファイルの display-name タグに設定した名称です。

7. 次のとおりにコマンドを実行して追加したリソースアダプタでデータベースに接続できるかどうかを試します。

```
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¥PP¥uCPSB¥CC¥admin¥bin¥cjtestres  
uCNP_J2EE -type rar -resname リソースアダプタの表示名
```

8. 次のとおりにコマンドを実行して追加したリソースアダプタを開始します。

```
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¥PP¥uCPSB¥CC¥admin¥bin¥cjstartrar  
uCNP_J2EE -resname リソースアダプタの表示名
```

4.4.2 リソースアダプタの設定変更

追加済みのリソースアダプタの設定を変更するには、いったんプラグインとリソースアダプタを停止する必要があります。手順を次に示します。

1. 次のとおりにコマンドを実行してリソースアダプタの一覧を表示し、設定を変更するリソースアダプタの状態を確認します。

```
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¥PP¥uCPSB¥CC¥admin¥bin¥cjlistrar
uCNP_J2EE
```

表示名の前に「running」と表示されているリソースアダプタは、開始状態にあります。「stopped」と表示されているリソースアダプタは停止しています。

なお、表示名が「uCNP_DB_Connector_for_HiRDB_Type4」のリソースアダプタはナビゲーションプラットフォーム用のリソースアダプタです。それ以外の表示名のリソースアダプタが、プラグイン用に追加されたリソースアダプタです。

2. 設定を変更するリソースアダプタを利用するプラグインがある場合は、次のとおりコマンドを実行して停止します。

```
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¥PP¥uCPSB¥CC¥admin¥bin¥cjstopapp
uCNP_J2EE -name プラグイン名
```

3. 次のとおりコマンドを実行して、設定を変更するリソースアダプタを停止します。

```
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¥PP¥uCPSB¥CC¥admin¥bin¥cjstoprar
uCNP_J2EE -resname リソースアダプタの表示名
```

4. [\[4.4.1 リソースアダプタの追加\]](#) の手順 5.以降を実施します。

4.4.3 リソースアダプタの削除

リソースアダプタを削除する場合、そのリソースアダプタを利用しているプラグインを先に削除する必要があります。手順を次に示します。

1. [\[4.4.2 リソースアダプタの設定変更\]](#) の手順 1.と手順 3.を実施します。
2. 削除するリソースアダプタを利用するプラグインがある場合は、次のとおりコマンドを実行して削除します。

```
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¥PP¥uCPSB¥CC¥admin¥bin
¥cjdeleteapp uCNP_J2EE -name プラグイン名
```

3. 次のとおりコマンドを実行して、リソースアダプタを削除します。

```
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¥PP¥uCPSB¥CC¥admin¥bin
¥cjdeleteres uCNP_J2EE -type rar -resname リソースアダプタの表示名
```

4.4.4 データベースとの接続処理のためのプラグインの設定

データベースとの接続処理をプラグインに設定するには、web.xml と cosminexus.xml の 2 ファイルを編集して、リソースアダプタの情報を追加する必要があります。各ファイルの格納場所と編集内容を次に示します。

注意事項

web.xml と cosminexus.xml は、エディタを [管理者として実行] で起動して編集してください。

(1) web.xml の編集

ユーザプラグインの 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。

```
Navigation Platform for Developersインストールディレクトリ¥pluginSDK¥plugin¥プラグインID  
¥WEB-INF¥web.xml
```

リソース参照の名称を「jdbc/TP_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/TP_Connector_for_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。

(2) cosminexus.xml の編集

ユーザプラグインの cosminexus.xml を編集します。

cosminexus.xml の格納先を次に示します。

```
Navigation Platform for Developersインストールディレクトリ¥pluginSDK¥plugin¥プラグインID¥dd  
¥META-INF¥cosminexus.xml
```

リソース参照の名称を「jdbc/TP_Connector_for_HiRDB_Type4」、リソースアダプタ表示名を「TP_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/TP_Connector_for_HiRDB_Type4</res-ref-name>  
  <linked-to>TP_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。

4.4.5 データベースとの接続処理の実装

データベースに接続するプラグインを開発する場合の条件を次に示します。

- データソースを取得する処理は、プラグインの初期化処理（IPluginInitializer インタフェースの PluginInitializer クラス）で実装する。
- データソースのルックアップ処理は、PluginInitializer クラスの init メソッドで実装する。

次に、データベースに接続するプラグインを開発する場合に、推奨する実装を示します。

- データソースの取得に失敗した場合は、UCNPPluginUserException をスローする。
- データソースの更新処理は inputFromNode メソッドで実装する。
- データソースの参照処理は outputToNode メソッドで実装する。
- 外部システムと連携するプラグインの場合、ほかのプラグインを使って更新と参照のタイミングを合わせる。

参考

データソースのルックアップ処理の失敗を無視してプラグインの起動処理を終了させると、ナビゲーション プラットフォームを再起動するまで次のルックアップ処理は実行されません。

ルックアップ処理失敗の要因は次のどちらかであるため、UCNPPluginUserException をスローしてナビゲーション プラットフォームが起動しないようにしておく必要があります。

- リソースアダプタが起動していない。
- リソースアダプタの設定とプラグインの実装に不整合がある。

(1) プラグインの初期化処理の実装例

```
//inputFromNodeメソッドからデータソースを取得できるように、staticメンバ変数に格納する。
private static DataSource mDs;

@Override
public void init() throws UCNPPluginUserException {
    try {
        // データソースのルックアップ処理は、必ずinitメソッド内で行うこと。
        InitialContext ic = new InitialContext();
        DataSource ds = (DataSource) ic.lookup(
            "java:comp/env/jdbc/TP_Connector_for_HiRDB_Type4");
        mDs = ds;
    } catch (NamingException e) {
        // ルックアップ処理に失敗したら、UCNPPluginUserExceptionをスローすること。
        UCNPPluginUserException ue = new UCNPPluginUserException(
            "ルックアップ処理に失敗しました。", e);
        throw ue;
    }
}
```

```

public static DataSource getDataSource() {
    return mDs;
}

```

(2) outputToNode メソッドの実装例

```

public Map<String, Object> outputToNode(HttpSession session, Map<String, Object> param){
    Map<String, Object> map = new HashMap<String, Object>();

    DataSource ds = PluginInitializer.getDataSource(); // インスタンスを取得
    Connection con = null;
    PreparedStatement statement = null;
    String name = null;
    try {
        // 接続を確立する
        con = ds.getConnection();

        // SQLを実行する
        // 参照系の処理は、outputToNodeメソッドで行う
        statement = con.prepareStatement("SELECT NAME FROM TP_TBL");
        statement.setString(1, uid);
        ResultSet set = statement.executeQuery();

        // 結果を取得する
        if (set.next()) {
            name = set.getString(1);
        }
        Map<String, String> rtnParm = new HashMap<String, String>();
        rtnParm.put("outputParam1", name);
        map.put("ucnp.next.params.map ", rtnParm);
    } catch (SQLException e) {
        // エラー情報を応答に格納して、遷移を中断させる。
        map.put("ucnp.error.message", "DBアクセスでエラーが発生しました。");
        map.put("ucnp.error.type", "NG");
    } finally {
        if (statement != null) {
            try {
                statement.close();
            } catch (SQLException e) {
            }
        }
        if (con != null) {
            try {
                con.close();
            } catch (SQLException e) {
            }
        }
    }
    return name;
}

```

4.5 プラグイン初期化・終了時の処理の実装

プラグインが起動または停止するタイミングで実行する必要がある処理は、IPluginInitializer インタフェースの PluginInitializer クラスに実装してください。該当する処理がない場合は、ひな形プラグイン作成コマンドで生成された PluginInitializer クラスを変更しないでそのまま使えます。

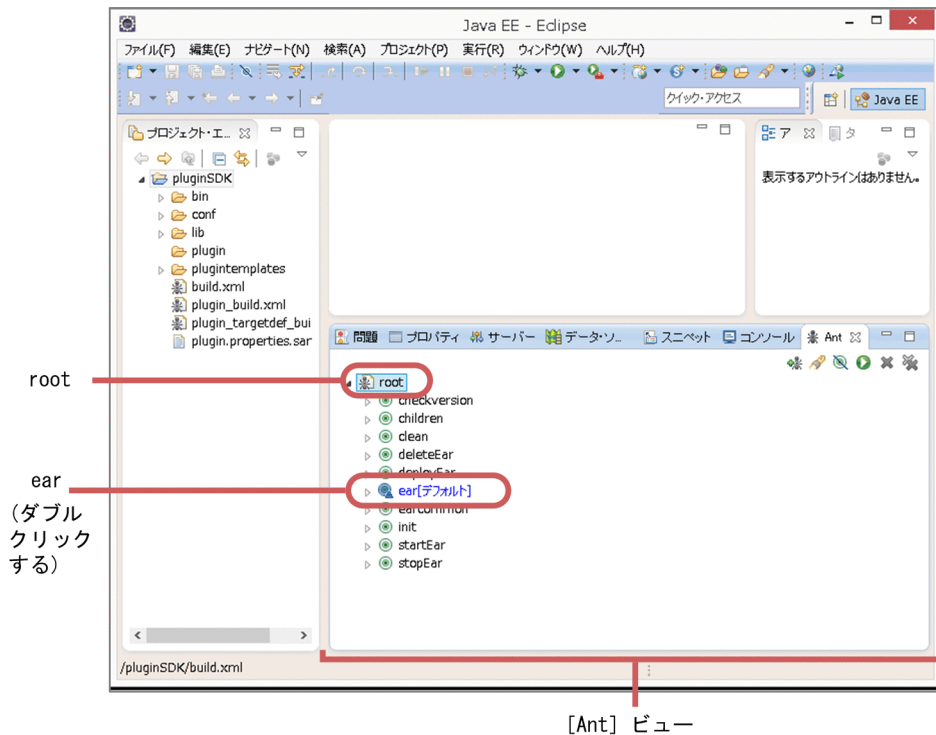
4.6 プラグインのビルド

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。

4.6.1 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。

1. Eclipse を起動します。
2. [ファイル] - [インポート] - [General] - [既存プロジェクトをワークスペースへ] を選択します。
3. [ルート・ディレクトリーの選択] テキストボックスに次の値を指定して、[終了] ボタンをクリックします。
Navigation Platform for Developers インストールディレクトリ¥pluginSDK
[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。
4. [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. [3.2 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 [ear] をダブルクリックします。

Navigation Platform for Developers インストールディレクトリ¥pluginSDK¥plugins ディレクトリ以下にあるすべての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。

*Navigation Platform for Developers*インストールディレクトリ¥pluginSDK¥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。

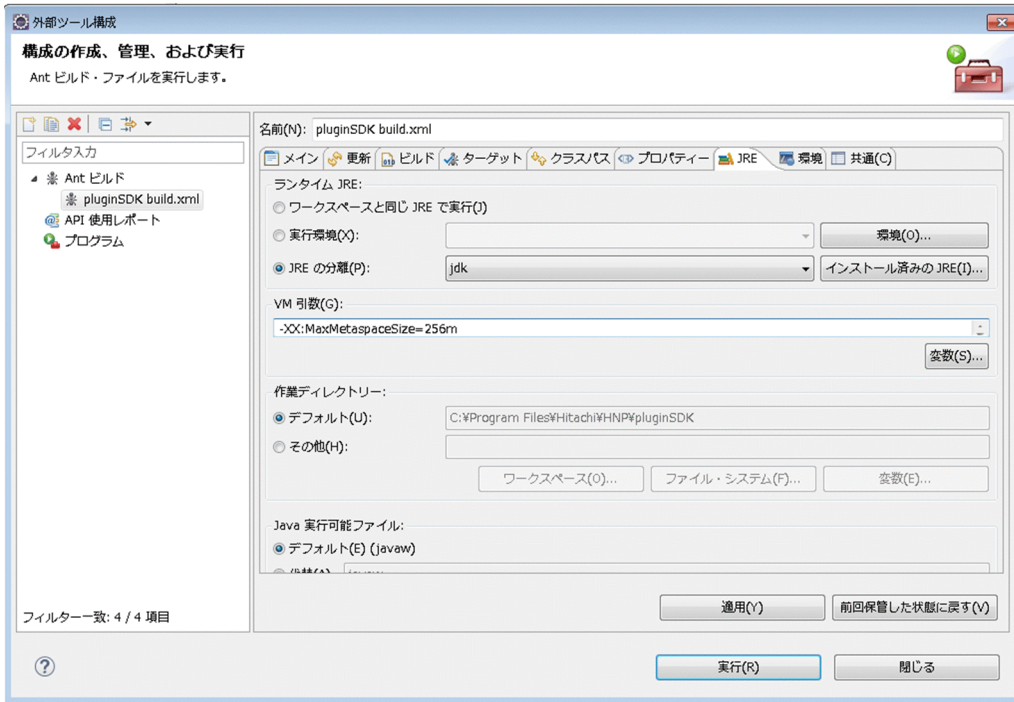
4.6.2 プラグインのビルドに失敗したときの対処

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。

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:MaxMetaspaceSize=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。

4.7 プラグインのデプロイ

開発したプラグインを利用できるようにするには、Eclipse を使ってデプロイし、開始する必要があります。手順を次に示します。

ポイント

この手順は Windows の管理者権限を持つユーザで実施してください。なお、ここで説明する手順は、Eclipse の Ant ビューに pluginSDK プロジェクトの「root」ノードが追加されていることを前提としています。

1. ナビゲーション プラットフォーム (J2EE サーバ) を起動します。

すでに起動している場合、この手順は不要です。また、ナビゲーション プラットフォームの起動方法については、マニュアル「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。

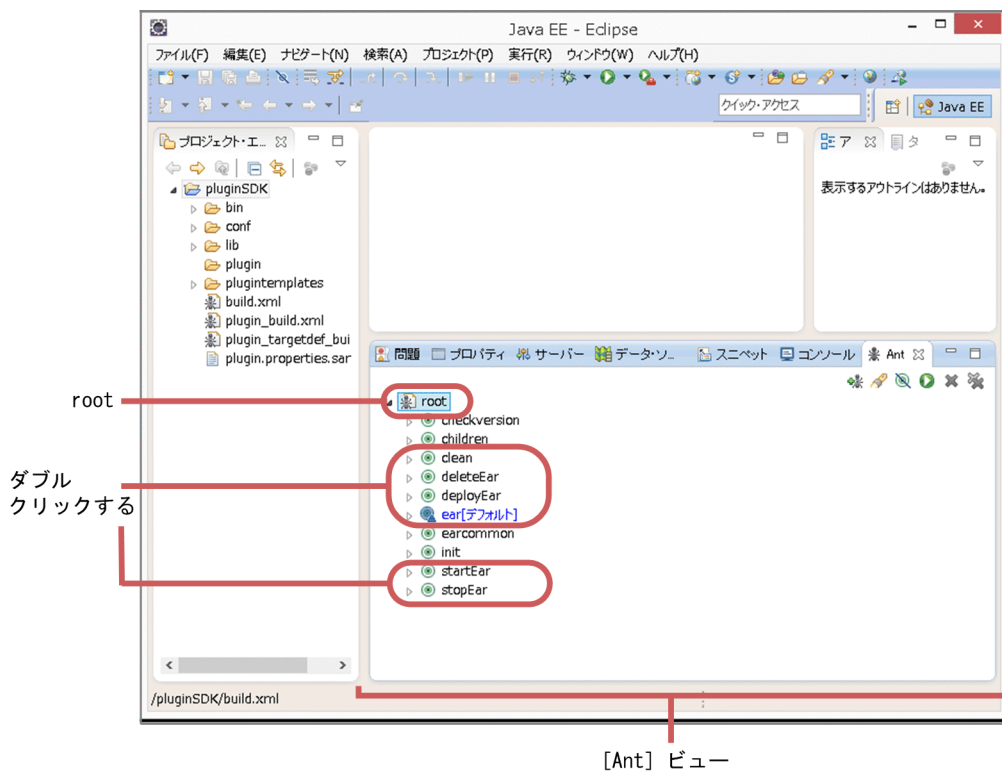
2. [管理者として実行] を選択して、Eclipse を起動します。

3. [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。

4. 「3.2 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。

5. [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。



6. J2EE サーバ上で起動しているプラグインがある場合は、ターゲット一覧の [stopEar] をダブルクリックします。
プラグインの J2EE アプリケーションが停止します。
7. J2EE サーバ上にプラグインが存在する場合は、ターゲット一覧の [deleteEar] をダブルクリックします。
プラグインの J2EE アプリケーションが削除されます。
8. ターゲット一覧の [clean] をダブルクリックしたあとに、[ear] をダブルクリックします。
プラグインがビルドされます。
9. ターゲット一覧の [deployEar] をダブルクリックします。
プラグインの 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。
プラグインの J2EE アプリケーションが開始されます。
11. ナビゲーション プラットフォーム (J2EE サーバ) を再起動します。

4.8 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またはカスタム画面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。

4.8.1 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）

入出力プラグインの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。

ポイント

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「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。

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。

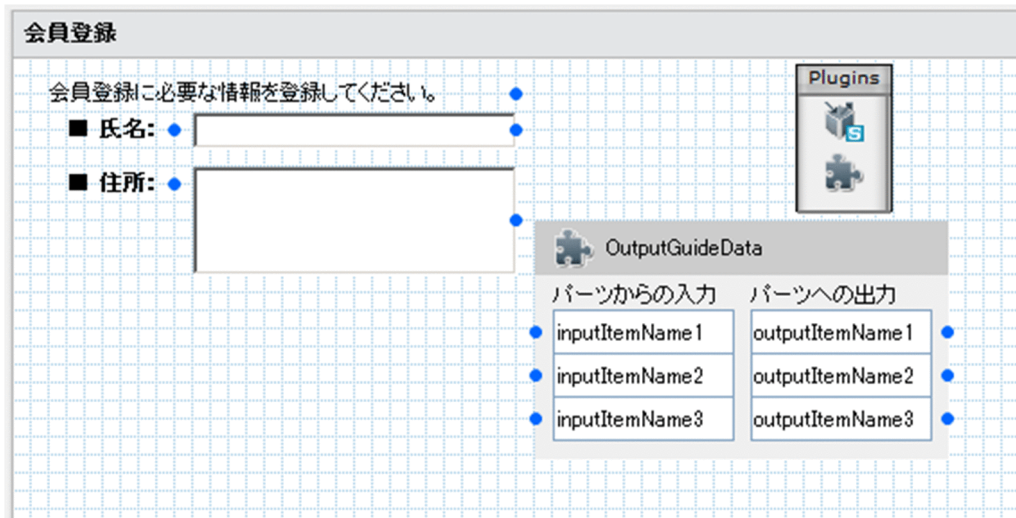
4. ツールバーの  プラグイン（[プラグイン] ボタン）をクリックします。

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) ガイドパーツのキャッシュ値」を参照してください。

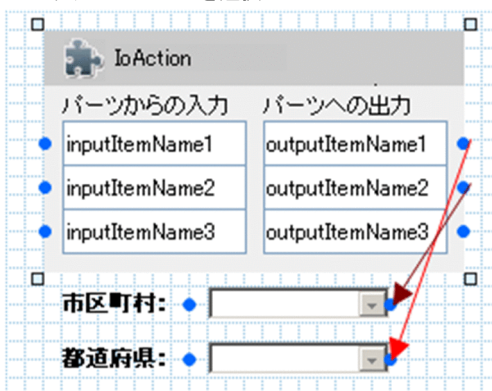
8. プラグインパーツの出力パラメタ (プラグインパーツの右側のポート) から遷移先のガイドパーツにドラッグしてマッピング線を引きます。

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。

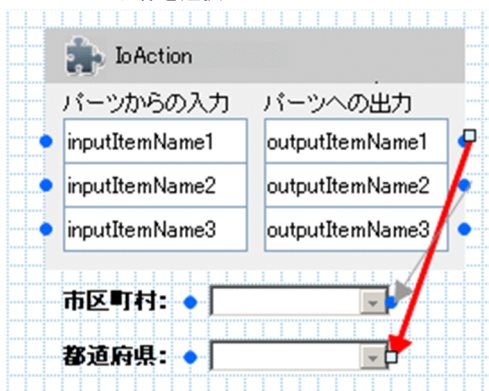
補足事項

- マッピング線は、次の図に示すように色付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。

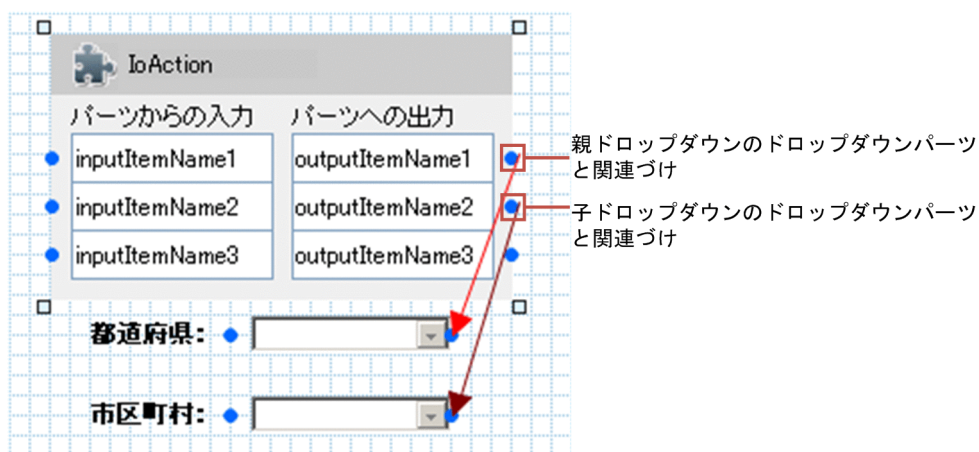
入出力プラグイン（サーバ処理実装インタフェース）を利用すると、条件に応じてキャッシュ値を変更できます。サーバ処理実装インタフェースについては、「5.2 IIoPluginController（サーバ処理実装インタフェース）」を参照してください。

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 4-4 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。

再配置とマッピング線の引き直しをしないでプラグインの修正を有効にする方法については、マニュアル「ナビゲーションプラットフォーム 導入から運用まで」のユーザプラグインの更新についての説明を参照してください。

4.8.2 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 4-5 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未チェック状態になる。
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、次の 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。 • テキストボックス	HTML パーツの [HTML ソース文字列] に設定する値が出力される。

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
HTML パーツ	<ul style="list-style-type: none"> 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 	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。

4.8.3 プラグインパーツを更新する

入出力プラグイン用 XML ファイルを編集して, 次に示す項目のどれかを変更した場合, 変更した内容を有効にするには, プラグインパーツを更新する必要があります。

- ・ ツールアイコンのファイル名
- ・ 実行順序
- ・ 実行確認ダイアログ
- ・ プレビュー画面実行フラグ
- ・ ボタン種別での実行抑止
- ・ 説明文字列
- ・ 入力パラメタ
- ・ 出力パラメタ

変更前のプラグインをすでに業務コンテンツに関連づけてある場合は, 配置済みのプラグインをいったん削除して関連づけ作業をやり直す必要があります。関連づけ作業をやり直さないで変更内容を有効にする方法については, マニュアル「ナビゲーション プラットフォーム 導入から運用まで」のユーザプラグインの更新についての説明を参照してください。

なお、プラグインの関連づけ作業のやり直し、またはユーザプラグインの更新作業は、編集環境および実行環境でも実施する必要があります。このため、入出力プラグイン用 XML ファイルを編集したときはシステム管理者にも通知してください。

4.8.4 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

業務編集画面のツールバーにある [マッピング一覧] ボタン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「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。

4.9 ユーザプロパティファイルの設定

開発したプラグインを有効にして正常に動作させるには、ユーザプロパティファイル (ucnp_user.properties) を次のように設定する必要があります。

- ucnp.base.client.directjump.enable プロパティに all を指定しない。
all を指定すると、中断再開プラグインの場合はエラーが発生して、ナビゲーションプラットフォームが起動しません。また、入出力プラグインも動作しません。
- 中断再開プラグインが存在する場合だけ、ucnp.base.client.suspend.enable プロパティに true を指定する。
true を指定すると、中断再開プラグインが動作します。また、中断再開プラグインが存在しないのに true を指定すると、ログイン後にエラーが発生して、業務実行画面を利用できません。
- カスタム画面の処理に応じて ucnp.base.server.logoutbutton.display プロパティおよび ucnp.base.server.close.button.setting プロパティの指定値をデフォルトから変更する。
カスタム画面からログアウトする場合は、ucnp.base.server.logoutbutton.display プロパティに false を、ucnp.base.server.close.button.setting プロパティには close_only を指定して、ナビゲーションプラットフォームの業務実行画面に表示するボタンを [閉じる] にしておく必要があります。

また、次に示すプロパティについても、必要に応じて指定値をデフォルトから変更してください。

- ucnp.base.client.viewer.confirm.discardinput.enable プロパティ (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 プロパティ (業務再開の確認ダイアログの表示有無)

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「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。

4.10 プラグインのデバッグ

プラグインは、Eclipse を使用してデバッグします。

4.10.1 プラグインをデバッグするための条件

プラグインをデバッグするには、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.debug.enable プロパティに true を指定して開発環境をセットアップしておく必要があります。また、デバッグ時にはナビゲーション プラットフォームを起動してください。

セットアップ手順、プロパティの詳細、ナビゲーション プラットフォームの起動手順については、マニュアル「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。

4.10.2 プラグインのデバッグ手順

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。

1. Eclipse を起動します。
2. [パッケージ・エクスプローラー] で [4.2 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。
3. メニューから [デバッグ] - [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。
4. [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。ポート番号には、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.debug.jdwp.port プロパティで設定した値を指定してください。
5. [デバッグ] ボタンをクリックします。
デバッグが実行されます。
6. 実装した Java コードにブレークポイントを設定します。
7. Web ブラウザで、プラグインが動作する操作を実行して、デバッグをします。
入出力プラグインの場合
プラグインを関連づけた業務コンテンツを操作してみて、プラグインが正しく動作するかどうかを確認します。

中断再開プラグインの場合

任意の業務コンテンツを途中まで操作し、[一時保存] ボタンをクリックしてログアウトします。再度ログインしたときに、途中まで操作した内容が保存され、業務フローの途中から操作を再開できるかどうかを確認します。

カスタム画面プラグインの場合

ログイン後に、カスタム画面が表示されるかどうかを確認します。

4.11 プラグインの削除

プラグインを削除するには、Eclipse を使用します。また、削除するプラグインの ID や、ライブラリの使用有無についてあらかじめ調べておく必要があります。

4.11.1 プラグインの削除手順

プラグインの削除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 (J2EE サーバ) を起動します。
すでに起動している場合、この手順は不要です。また、ナビゲーション プラットフォームの起動方法については、マニュアル「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. 削除するプラグインが入出力プラグインの場合は、プラグインを使用している業務コンテンツを削除するか、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。
3. [管理者として実行] を選択して、Eclipse を起動します。
4. [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. ターゲット一覧の [stopEar] と [deleteEar] を順番にダブルクリックして、いったんすべてのプラグインを削除します。
6. 削除対象のプラグインのディレクトリ (*Navigation Platform for Developers* 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。
7. 削除対象のプラグインだけがシステムクラスローダのレイヤに配置したライブラリを利用している場合は、配置したライブラリを削除し、クラスパスの設定も削除します。
8. 手順 5. で削除したプラグインに、削除したくないプラグインが含まれている場合は、[deployEar] と [startEar] を順番にダブルクリックして、再度デプロイします。
9. ナビゲーション プラットフォームを再起動します。

4.12 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。

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

(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。

4.12.1 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。

4.12.2 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) ナビゲーション プラットフォームの再起動

1. npstop コマンド (停止) でナビゲーション プラットフォームを停止します。
2. 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) 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ナビゲーション プラットフォームの RMI レジストリのポートを使用しているプロセス ID を取得します。

```
netstat -abo
```

コマンドの出力例を次に示します。

プロトコル	ローカル アドレス	外部アドレス	状態	PID
TCP	0.0.0.0:24702	navipla:0	LISTENING	1876

ローカルアドレスのポート番号が「24702」となっている行の PID を確認してください。この例では、「1876」が RMI レジストリのポートを使用しているプロセス ID となります。RMI レジストリ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.rmi.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「24702」です。

2. 次のコマンドで GC を実行します。

```
"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¥PP¥uCPSB¥jdk¥jre¥bin  
¥javagc.exe" -p RMIレジストリ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)
```

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

1. 次のコマンドを実行してプラグインの J2EE アプリケーションを停止します。

```
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¥PP¥uCPSB¥CC¥admin¥bin¥cjstopapp  
uCNP_J2EE -name ユーザプラグイン名
```

ユーザプラグインが複数ある場合は、すべてのプラグインを停止するまでこのコマンドを繰り返してください。

2. 次のコマンドを実行してナビゲーション 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¥PP¥uCPSB¥CC¥admin¥bin¥cjstopapp
uCNP_J2EE -name ucnp
```

(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。

```
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¥PP¥uCPSB¥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。

(6) ナビゲーション プラットフォームの再起動

1. npstop コマンド（停止）でナビゲーション プラットフォームを停止します。
2. 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。

4.13 J2EE サーバの設定項目の変更

必要に応じて変更できる J2EE サーバの設定項目と、変更手順を説明します。このマニュアルで説明している項目以外は、デフォルトの設定から変更しないでください。

4.13.1 プラグイン開発時に変更できる J2EE サーバの設定項目

編集できる J2EE サーバのファイルと、変更できる設定項目を次に示します。

J2EE サーバ用ユーザプロパティファイル (usrconf.properties)

Java VM のシステムプロパティ

J2EE サーバ用オプション定義ファイル (usrconf.cfg)

システムクラスパス

Java VM の Java ヒープのメモリ使用量の最小値

Java VM の Java ヒープのメモリ使用量の最大値

Java VM の明示管理ヒープ機能における、Explicit メモリブロックの最大サイズ

このほかに、リソースアダプタも開発するプラグインに合わせて変更できます。リソースアダプタの変更については、「[4.4 データベースとの接続処理の追加](#)」を参照してください。

4.13.2 J2EE サーバの設定項目を変更するためのファイルの格納場所

J2EE サーバ用ユーザプロパティファイル (usrconf.properties) と J2EE サーバ用オプション定義ファイル (usrconf.cfg) の格納場所は次のとおりです。

```
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¥PP¥uCPSB¥CC¥server¥usrconf¥ejb
¥uCNP_J2EE
```

4.13.3 J2EE サーバ用ユーザプロパティファイル (usrconf.properties) の変更手順

J2EE サーバ用ユーザプロパティファイル (usrconf.properties) で変更できる設定項目は、Java VM のシステムプロパティです。手順を次に示します。

注意事項

この手順で説明されていない箇所は、編集しないでください。また、説明されていない設定値を追加しないでください。

1. 管理者としてエディタを起動して、J2EE サーバ用ユーザプロパティファイル (usrconf.properties) を開きます。
2. 次に示すコメント行よりも下の行に Java VM のシステムプロパティ値の指定を追加します。

```
#####  
## When you edit, please add description below. ##  
#####
```

3. npstop コマンド (停止) と npstart コマンド (起動) を実行してナビゲーション プラットフォームを再起動します。

(1) Java VM のシステムプロパティの詳細

形式

次のようにキーを指定します。

```
<キー名称> = <値>
```

指定方法

- 改行までが値になります。
- #で始まる行はコメントと見なされます。
- 値が存在しない行を定義した場合、その行は無視されます。
- 値の後ろには、空白やコメントなどの文字列は追加できません。追加した場合、不正な値と解釈されます。

システムプロパティのキー名称には、任意の文字列を指定できます。ただし、次に示す条件に該当する接頭辞は使用しないでください。

- ucnp.
- java.
- javax.
- javax.portlet.
- hptl
- com.cosminexus
- jp.co.hitachi.soft.portal
- 次に示す J2EE サーバの予約済みキー
 - com.cosminexus
 - cosminexus.jpa
 - ejbserver.application
 - ejbserver.client

- ejbserver.commonj
- ejbserver.compiler
- ejbserver.connectionpool
- ejbserver.connector
- ejbserver.container
- ejbserver.ctm
- ejbserver.deploy
- ejbserver.distributedtx
- ejbserver.DynamicStubLoading
- ejbserver.ejb
- ejbserver.ext
- ejbserver.http
- ejbserver.instrumentation
- ejbserver.jca
- ejbserver.jndi
- ejbserver.jpa
- ejbserver.jta
- ejbserver.logger
- ejbserver.management
- ejbserver.manager
- ejbserver.naming
- ejbserver.rmi
- ejbserver.security
- ejbserver.server
- ejbserver.stateful
- ejbserver.stdoutlog
- ejbserver.watch
- ejbserver.webj2ee
- https
- java
- javax
- vbj

- vbroker
- webserver.application
- webserver.connector
- webserver.container
- webserver.context
- webserver.dbsfo
- webserver.eadssfo
- webserver.errorpage
- webserver.http
- webserver.jsp
- webserver.logger
- webserver.servlet
- webserver.session
- webserver.static
- webserver.work
- webserver.xml

(2) J2EE サーバ用ユーザプロパティファイル (usrconf.properties) の編集例

J2EE サーバ用ユーザプロパティファイルに「com.example.property.key1」という名称のキーを追加した例を次に示します。

```
#####
## When you edit, please add description below.          ##
#####
com.example.property.key1=value1
```

4.13.4 J2EE サーバ用オプション定義ファイル (usrconf.cfg) の変更手順

J2EE サーバ用オプション定義ファイル (usrconf.cfg) で変更できる設定項目は、次の項目です。

- システムクラスパス
- Java VM の Java ヒープのメモリ使用量
- Java VM の明示管理ヒープ機能の Explicit メモリブロックサイズ

手順を次に示します。

注意事項

この手順で説明されていない箇所は、編集しないでください。また、説明されていない設定値を追加しないでください。

1. 管理者としてエディタを起動して、J2EE サーバ用オプション定義ファイル (usrconf.cfg) を開きます。
2. Java VM の Java ヒープのメモリ使用量、および Java VM の明示管理ヒープ機能の Explicit メモリブロックサイズの設定値を変更する場合は、次に示すコメント行よりも下の行にあるキーの値を変更します。

```
#####  
## When you change settings, please correct the following description.  ##  
#####
```

3. システムクラスパスの設定値を変更する場合は、次に示すコメント行よりも下の行に指定を追加します。

```
#####  
## When you edit, please add description below.  ##  
#####
```

4. npstop コマンド (停止) と npstart コマンド (起動) を実行してナビゲーション プラットフォームを再起動します。

(1) システムクラスパス

J2EE サーバのシステムクラスローダでロードする、ユーザプラグインやユーザライブラリといったユーザ作成のプログラムを指定します。

J2EE サーバ全体から参照できるプログラムを指定してください。また、複数指定することもできます。

キー名称

add.class.path

指定値

システムクラスパスを指定してください。複数指定したい場合は、セミコロン (;) で区切ってください。

デフォルト値

なし。

省略値

なし。

(2) Java VM の Java ヒープのメモリ使用量の詳細

J2EE サーバの Java VM の Java ヒープのメモリ使用量は、ナビゲーション プラットフォームのメモリ使用量より大きな値を指定する必要があります。また、最小値と最大値は同じ値にしてください。

注意事項

J2EE サーバの Java VM の Java ヒープのメモリ使用量の最小値・最大値の指定は省略できません。デフォルトで記載されている `add.jvm.arg` キーは削除しないでください。

キー名称

`add.jvm.arg`

指定値

`-Xmx` メモリ使用量のサイズ `m`

メモリ使用量のサイズは、1024~1434 の範囲で指定してください。1434 より大きい値を指定すると、OS のメモリ上限値を超えてしまいます。指定値末尾の「`m`」は、「MB」を意味します。

デフォルト値

`-Xmx1024m`

(3) Java VM の明示管理ヒープ機能の Explicit メモリブロックサイズの詳細

J2EE サーバの Java VM の Explicit ヒープの最大サイズは、ナビゲーション プラットフォームの Explicit メモリブロックサイズより大きな値を指定する必要があります。

注意事項

Explicit ヒープの最大サイズの指定は省略できません。デフォルトで記載されている `add.jvm.arg` キーは削除しないでください。

参考

Explicit ヒープには、ユーザプラグインのインスタンスと `ucnpOptions` パラメタの値を含むセッション情報が格納されます。そのため、「1 ユーザあたりのユーザプラグインのインスタンスのメモリ容量×最大同時実行ユーザ数」の分だけ Explicit ヒープを拡張するケースを想定しておくことを推奨します。

キー名称

`add.jvm.arg`

指定値

`-XX:HitachiExplicitHeapMaxSize=Explicit ヒープの最大サイズ` `m`

Explicit ヒープの最大サイズには、160 以上の値を指定してください。指定値末尾の「m」は、「MB」を意味します。

デフォルト値

```
-XX:HitachiExplicitHeapMaxSize=160m
```

4.14 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

- カスタム画面プラグインで作成した画面で JavaScript を使って画面を閉じる処理を実行させる場合は、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。
- 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。

4.15 旧バージョンで開発したプラグインの使用について

uCosminexus Navigation Developer 09-50 より前のバージョン（以降、09-50 より前の旧バージョン）の開発環境で開発した入出力プラグインは、ナビゲーション プラットフォームでは動作しません。09-50 より前の旧バージョンで開発した入出力プラグインが必要な場合は、ひな形プラグインを生成するところから開発し直す必要があります。

uCosminexus Navigation Developer 09-50 または uCosminexus Navigation Developer 09-60 の旧バージョン（以降、09-50 以降の旧バージョン）の開発環境で開発した入出力プラグインも、そのままではナビゲーション プラットフォームで動作しません。ただし、09-50 以降の旧バージョンで開発した入出力プラグインは、ナビゲーション プラットフォームの新しいひな形プラグインに移行できるので、開発し直す必要はありません。

移行手順については、「[付録 C.1 旧バージョンで開発したプラグインを移行する手順](#)」を参照してください。

ポイント

09-50 より前の旧バージョンで開発した入出力プラグインを関連づけている業務コンテンツは、ナビゲーション プラットフォームにインポートできません。インポートする場合は、09-50 より前の旧バージョンの環境で該当するプラグインをすべて削除しておく必要があります。

4.16 ユーザプラグインが利用する Java パッケージへのアクセス権の設定

ユーザプラグインが利用する Java パッケージの中には、個別にアクセス権の設定が必要なものがあります。アクセス権の設定手順を次に示します。

1. 管理者としてエディタを起動して、次のファイルを開きます。

```
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¥PP¥uCPSB¥CC¥server¥usrconf¥ejb
¥uCNP_J2EE¥server.policy
```

2. 次の個所に `accessClassInPackage` の設定を追加します。

```
grant codeBase "file:${ejbserver.http.root}/ejb/${ejbserver.serverName}/apps/-" {
    permission java.io.FilePermission "<<ALL FILES>>", "read, write, delete, execute";
    permission java.lang.RuntimePermission "accessDeclaredMembers";
    permission java.lang.RuntimePermission "getProtectionDomain";
    permission java.lang.RuntimePermission "setFactory";
    permission java.lang.reflect.ReflectPermission "suppressAccessChecks";
    permission java.lang.RuntimePermission "getenv.*";
    permission java.lang.RuntimePermission "getClassLoader";
    permission java.lang.RuntimePermission "accessClassInPackage.利用するJavaパッケージ名"; // 追加
};
```

3. ナビゲーション プラットフォームを再起動します。

5

API リファレンス（入出力プラグイン開発の場合）

この章では、入出力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 (入出力プラグイン開発の場合)

入出力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 (入出力プラグイン開発)

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。
	IPluginInitializer	ユーザプラグインの起動（初期化）および終了時の処理を実装するためのインタフェースです。
	UCNPPluginUserException	ユーザプラグイン例外クラスです。
入出力プラグイン	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。

5.2 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。

5.2.1 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システ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。プラグインの中で取得した HttpSession オブジェクトに対して 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
- "java." で始まる名前
- "javax." で始まる名前
- "javax.portlet." で始まる名前
- "hptl" で始まる名前
- "com.cosminexus" で始まる名前
- "jp.co.hitachi.soft.portal" で始まる名前

また、URL のリクエストパラメータを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、ucnp.request.options キーで ucnpOptions パラメータを取得する必要があります。

ucnpOptions パラメータは、セッション単位または画面 ID 単位で取得できます。ucnpOptions パラメータの値は画面 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 パラメータの値が HTTP セッションに上書きされ、前に操作した画面の ucnpOptions パラメー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 5-2 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>ガイドパーツのキャッシュ値については「(5) ガイドパーツのキャッシュ値」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" ターミナルノード（開始）からプロセスノードに遷移する。 または、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する。 "show_next_page" ボタンをクリックする、もしくは直接ノードを選択することで次のノードに遷移する。

項番	キー名	値	説明
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>または、遷移先のノードが存在しているプロセスノード、もしくは [戻る] ボタンが表示になっているプロセスノードに遷移する。</p> <ul style="list-style-type: none"> • "show_previous_page" ボタンをクリックする、または直接ノードを選択することで前のノードに遷移する。 • "show_complete_page" [完了] ボタンが表示されているノードに遷移する。 <p>または、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する。</p>
3	ucnp.isdirectjump	String 次のどちらか <ul style="list-style-type: none"> • "true" • "false" 	<p>遷移の種別が直接遷移かどうかを示します。</p> <ul style="list-style-type: none"> • "true" 直接遷移の場合 • "false" 直接遷移ではない場合
4	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>
5	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>
6	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> <p>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。</p>
7	ucnp.next.node.id	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 ID が渡されます。</p> <p>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。</p>
8	ucnp.flow.contents.id	String 1 文字以上の文字列	<p>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。</p>
9	ucnp.flow.contents.name	String 1 文字以上の文字列	<p>選択中の業務コンテンツの業務名が渡されます。</p>
10	ucnp.flow.contents.execute.id	String	<p>業務コンテンツの実行時に割り当てられる ID (業務実行 ID) です。</p>

項番	キー名	値	説明
11	ucnp.flow.contents.version.id	String	実行中の業務コンテンツのバージョン ID です。
12	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。
13	ucnp.screen.ispreview	String 次のどちらか • "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。 • "true" プレビュー画面 • "false" 業務実行画面
14	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 5-3 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。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「<code>\n</code>」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 • 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

項番	キー名	値	説明
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「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 5-4 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
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

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
9	指定あり (null 以外)	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)	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) ガイドパーツのキャッシュ値

プラグインで保持されるガイドパーツのキャッシュ値を次の表に示します。

表 5-5 各ガイド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
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

項番	パーツの種類	キャッシュ値	初期値
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。

属性の設定

項目種別: ドロップダウン
ガイドパーツID:
親のガイドパーツID:

親の選択項目一覧の各値に対応付ける選択項目一覧を設定します。
親を指定しない場合は親の値を空欄にし、選択項目一覧のみを指定してください。

親の値	選択項目一覧(コンマ区切りで指定) 上段:ラベル 下段:値	
tokyo	渋谷区.北区 shibuyaku.kitaku	追加 削除
kanagawa	横浜市.川崎市 yokohamashi.kawasakishi	削除

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。

5.2.2 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システ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。プラグインの中で取得した `HttpSession` オブジェクトに対して `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
- "java."で始まる名前
- "javax."で始まる名前
- "javax.portlet."で始まる名前
- "hptl"で始まる名前
- "com.cosminexus"で始まる名前
- "jp.co.hitachi.soft.portal"で始まる名前

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、「5.2 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 5-6 param に格納されるキー (outputToNode メソッド)

項番	キー名	値	説明
1	<code>ucnp.next.params.map</code>	<code>Map<String,String></code>	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たガイドの、遷移前の時点のキャッシュ値 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。 ガイドパーツのキャッシュ値については「5.2.1(5) ガイドパーツのキャッシュ値」を参照してください。
2	<code>ucnp.button.type</code>	<code>String</code> 次のどれか <ul style="list-style-type: none">"start""show_next_page""show_previous_page""show_complete_page"	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">"start" ターミナルノード (開始) からプロセスノードに遷移する。 または、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する。"show_next_page" ボタンをクリックする、もしくは直接ノードを選択することで次のノードに遷移する。

項番	キー名	値	説明
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>または、遷移先のノードが存在しているプロセスノード、もしくは [戻る] ボタンが表示になっているプロセスノードに遷移する。</p> <ul style="list-style-type: none"> • "show_previous_page" ボタンをクリックする、または直接ノードを選択することで前のノードに遷移する。 • "show_complete_page" [完了] ボタンが表示されているノードに遷移する。 <p>または、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する。</p>
3	ucnp.isdirectjump	String 次のどちらか <ul style="list-style-type: none"> • "true" • "false" 	<p>遷移の種別が直接遷移かどうかを示します。</p> <ul style="list-style-type: none"> • "true" 直接遷移の場合 • "false" 直接遷移ではない場合
4	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>
5	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>
6	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> <p>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。</p>
7	ucnp.next.node.id	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 ID が渡されます。</p> <p>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。</p>
8	ucnp.flow.contents.id	String 1 文字以上の文字列	<p>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。</p>
9	ucnp.flow.contents.name	String 1 文字以上の文字列	<p>選択中の業務コンテンツの業務名が渡されます。</p>
10	ucnp.flow.contents.execute.id	String	<p>業務コンテンツの実行時に割り当てられる ID (業務実行 ID) です。</p>

項番	キー名	値	説明
11	ucnp.flow.contents.version.id	String	実行中の業務コンテンツのバージョン ID です。
12	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。
13	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
14	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、「5.2 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 5-7 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

項番	キー名	値	説明
2	ucnp.error.message	String	<p>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。</p> <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 <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「<code>\n</code>」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、ノード遷移を抑止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 • 入出力プラグインに"WARNING"を一つも指定していない 遷移が抑止される

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 5-8 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
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

5.2.3 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
    }
}
```

5.3 IPluginInitializer (ユーザプラグインの起動 (初期化)・終了時の処理の実装インタフェース)

ユーザプラグインの起動 (初期化) および終了時の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.base.pluginmng.controller;
```

形式

```
public interface IPluginInitializer
```

メソッド

- `init` メソッド
ユーザプラグインの初期化処理を実装します。
- `destroy` メソッド
ユーザプラグインの終了処理を実装します。

5.3.1 `init` メソッド

このメソッドは、ユーザプラグインの EAR が起動するときに呼び出されます。このメソッドが呼び出されると、ユーザプラグインの初期化処理が実行されます。

例えば、データベースとの接続処理をするユーザプラグインを実装する場合に、データソースをルックアップする処理をこのメソッドで実行します。データベースとの接続処理を実装する手順については、「[4.4 データベースとの接続処理の追加](#)」を参照してください。

データベースと接続するためのデータソースのルックアップは、必ずこのメソッド内で実装してください。ほかのメソッド内で実装した場合は動作が保障されません。`inputFromNode()`メソッドなど、ほかのメソッドでデータソースを利用する場合は、`init()`メソッド内でルックアップしたデータソースのインスタンスをスタティック変数などに保持し、受け渡すように実装してください。

(1) 形式

```
public void init() throws UCNPPuginUserException;
```

(2) 引数

なし

(3) 戻り値

なし

(4) 例外

このメソッドでエラーが発生した場合、ユーザプラグインの起動は中断され、UCNPPluginUserException 例外がスローされます。

5.3.2 destroy メソッド

このメソッドは、ユーザプラグインの EAR が停止するときに呼び出されます。このメソッドが呼び出されると、ユーザプラグインの終了処理が実行されます。

(1) 形式

```
public void destroy();
```

(2) 引数

なし

(3) 戻り値

なし

(4) 例外

なし

5.4 UCNPPluginUserException (ユーザプラグイン例外クラス)

ユーザプラグインからナビゲーション プラットフォームにスローする例外です。

パッケージ

```
package jp.co.hitachi.soft.ucnp.base.pluginmng.controller;
```

形式

```
public class UCNPPluginUserException extends Exception;
```

コンストラクタ

- UCNPPluginUserException(String)コンストラクタ
親クラスの Exception(String)コンストラクタを呼び出します。
- UCNPPluginUserException(String, Throwable)コンストラクタ
親クラスの Exception(String, Throwable)コンストラクタを呼び出します。

5.5 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。

5.5.1 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 UCNPluginException;
```

(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

UCNPluginException - 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、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 5-9 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

IloPluginController 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;  
  }  
}
```

5.6 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。

5.6.1 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

5.6.2 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

5.7 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、HttpSession オブジェクトから、次に示すコードで取得できます。

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

なお、IUCNPSession インタフェースのインスタンスはログイン状態のときだけ取得できます。上記のコードでは、ログインしていない状態のときは ucnpSession 変数が null になります。

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。

5.7.1 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

6

API リファレンス（中断再開プラグイン開発の場合）

この章では、中断再開プラグインの開発で使用する API について説明します。

6.1 API 一覧 (中断再開プラグイン開発の場合)

中断再開プラグインの開発で使用する API を次に示します。

表 6-1 API 一覧 (中断再開プラグイン開発)

分類	インタフェース名 またはクラス名	説明
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。

全プラグイン共通で使用する API については, 「[5.1 API 一覧 \(入出力プラグイン開発の場合\)](#)」を参照してください。

6.2 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。

6.2.1 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 UCNPPuginException;
```

(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。プラグインの中で取得した HttpSession オブジェクトに対して 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。

- "java."で始まる名前
- "javax."で始まる名前
- "javax.portlet."で始まる名前
- "hptl"で始まる名前
- "com.cosminexus"で始まる名前
- "jp.co.hitachi.soft.portal"で始まる名前

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、「5.2 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。中断再開プラグインが出力するエラーメッセージに、多数の改行を含む場合、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。これは、JavaScript の Alert でエラーメッセージを表示するためです。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

6.2.2 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。プラグインの中で取得した HttpSession オブジェクトに対して 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
- "java."で始まる名前
- "javax."で始まる名前
- "javax.portlet."で始まる名前
- "hptl"で始まる名前
- "com.cosminexus"で始まる名前
- "jp.co.hitachi.soft.portal"で始まる名前

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、「5.2 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。中断再開プラグインが出力するエラーメッセージに、多数の改行を含む場合、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。これは、JavaScript の Alert でエラーメッセージを表示するためです。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、「6.2 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。

6.2.3 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。プラグインの中で取得した `HttpSession` オブジェクトに対して `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
- "java."で始まる名前
- "javax."で始まる名前
- "javax.portlet."で始まる名前
- "hptl"で始まる名前

- "com.cosminexus"で始まる名前
- "jp.co.hitachi.soft.portal"で始まる名前

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、「5.2 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。中断再開プラグインが出力するエラーメッセージに、多数の改行を含む場合、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。これは、JavaScriptのAlertでエラーメッセージを表示するためです。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
- URLを直接指定し、業務コンテンツ ID を指定するパラメータ (contentId) で、業務コンテンツが選択されたとき

6.2.4 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。パラメー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 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。プラグインの中で取得した `HttpSession` オブジェクトに対して `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
- "java." で始まる名前
- "javax." で始まる名前
- "javax.portlet." で始まる名前
- "hptl" で始まる名前
- "com.cosminexus" で始まる名前
- "jp.co.hitachi.soft.portal" で始まる名前

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、「5.2 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。中断再開プラグインが出力するエラーメッセージに、多数の改行を含む場合、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。これは、JavaScript の Alert でエラーメッセージを表示するためです。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、「6.2 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。

6.2.5 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 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 deleteAll(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。公開予約状態の業務コンテンツの公開開始処理でこのメソッドが実行された場合、値は null になります。プラグインの中で取得した HttpSession オブジェクトに対して 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
- "java." で始まる名前
- "javax." で始まる名前
- "javax.portlet." で始まる名前
- "hptl" で始まる名前
- "com.cosminexus" で始まる名前
- "jp.co.hitachi.soft.portal" で始まる名前

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、「5.2 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が、利用者にわかるメッセージを設定してください。中断再開プラグインが出力するエラーメッセージに、多数の改行を含む場合、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。これは、JavaScript の Alert でエラーメッセージを表示するためです。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 6-2 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
Web ブラウザ上の操作により、業務実行画面に表示する業務コンテンツの内容が変化したとき	対象の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

6.3 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
```

メソッド

- [getContentSuspendedId](#) メソッド
中断 ID を取得します。監査ログに出力されたログと中断情報の突き合わせを行う場合に利用できます。
- [getWorkId](#) メソッド
業務実行 ID を取得します。操作ログに出力された各操作と、中断操作を行った業務が同じ業務であるか突き合わせを行う場合に利用できます。
- [getContentName](#) メソッド
業務中断時に実行している業務コンテンツの名称を取得します。
- [getGroupName](#) メソッド
業務中断時に実行している業務コンテンツが所属するグループの名称を取得します。
- [getCurrentNodeName](#) メソッド
業務中断時に実行しているノードの名称を取得します。

6.3.1 getContentSuspendedId メソッド

中断 ID を取得します。監査ログに出力されたログと中断情報の突き合わせを行う場合に利用できます。

(1) 形式

```
public String getContentSuspendedId();
```

(2) 引数

なし

(3) 戻り値

中断 ID

(4) 例外

なし

6.3.2 getWorkId メソッド

業務実行 ID を取得します。操作ログに出力された各操作と、中断操作を行った業務が同じ業務であるか突き合わせを行う場合に利用できます。

(1) 形式

```
public String getWorkId();
```

(2) 引数

なし

(3) 戻り値

業務実行 ID

(4) 例外

なし

6.3.3 getContentName メソッド

業務中断時に実行している業務コンテンツの名称を取得します。

(1) 形式

```
public String getContentName();
```

(2) 引数

なし

(3) 戻り値

業務中断時に実行している業務コンテンツの名称

(4) 例外

なし

6.3.4 getGroupName メソッド

業務中断時に実行している業務コンテンツが所属するグループの名称を取得します。

(1) 形式

```
public String getGroupName();
```

(2) 引数

なし

(3) 戻り値

業務中断時に実行している業務コンテンツが所属するグループの名称

(4) 例外

なし

6.3.5 getCurrentNodeName メソッド

業務中断時に実行しているノードの名称を取得します。

(1) 形式

```
public String getCurrentNodeName();
```

(2) 引数

なし

(3) 戻り値

業務中断時に実行しているノードの名称

(4) 例外

なし

6.4 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。

6.4.1 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 UCNPPluginException;
```

(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 6-3 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

6.4.2 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 UCNPluginException;
```

(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

UCNPluginException - 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, 直列化復元処理は中断され, UCNPluginException 例外がスローされます。エラーの一覧を次に示します。

表 6-4 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

6.5 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、全文が表示されないことがあります。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しまうためです。

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。

6.5.1 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

6.5.2 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 UCNPPluginException(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

7

API リファレンス（カスタム画面プラグイン開発の場合）

この章では、カスタム画面プラグインの開発で使用する API について説明します。

7.1 API 一覧 (カスタム画面プラグイン開発の場合)

カスタム画面プラグインの開発で使用する API を次に示します。

表 7-1 API 一覧 (カスタム画面プラグイン開発)

分類	インタフェース名 またはクラス名	説明
ログアウト機能	CustomWindowUrlUtil	カスタム画面の URL を取得するためのユーティリティクラスです。
	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。

全プラグイン共通で使用する API については、「[5.1 API 一覧 \(入出力プラグイン開発の場合\)](#)」を参照してください。

7.2 CustomWindowUrlUtil (カスタム画面 URL 取得ユーティリティクラス)

カスタム画面の URL を取得するためのユーティリティクラスです。

パッケージ

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

形式

```
public class CustomWindowUrlUtil
```

メソッド

- `getCustomWindowUrl` メソッド
カスタム画面の URL を取得します。

7.2.1 getCustomWindowUrl メソッド

カスタム画面の URL を取得します。このメソッドで取得した URL には、必ず `ucnpUserPageId` パラメータを指定してください。

(1) 形式

```
public static String getCustomWindowUrl(HttpServletRequest request, HttpServletResponse response);
```

(2) 引数

なし

(3) 戻り値

「ucnpBase」から始まるカスタム画面の URL が返されます。

(4) 例外

なし

(5) 使用例

カスタム画面の JSP ファイル内で `form` タグを使用する場合の例を次に示します。

```
<%  
    String url = CustomWindowUtil.getCustomWindowUrl(request, response);  
%>  
  
<form action="<%= url %>" method="post" target="_self">  
<input type="hidden" name="ucnpUserId" value="userCustomPage" />  
...略...  
</form>
```

7.3 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、JSP ファイル内で使用します。カスタム画面でログアウト処理と通常の処理を分岐させるには、ucnpUserData パラメタを使用してください。

ucnpUserData パラメタについては、「[2.5.2 カスタム画面プラグインで受け取れるデータ](#)」を参照してください。

パッケージ

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

形式

```
public class LogoutActionUtil
```

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。

7.3.1 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。

トレースファイルの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

付録

付録 A サンプルプラグインの使用法

Navigation Platform for Developers 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。

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

```
Navigation Platform for Developers 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

```
Navigation Platform for Developers インストールディレクトリ¥pluginSDK¥plugin
```

2. 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、「[4.6 プラグインのビルド](#)」を参照してください。

3. プラグインをデプロイして開始します。

プラグインのデプロイ方法については、「[4.7 プラグインのデプロイ](#)」を参照してください。

4. 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、「4.3.1(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。

5. 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、「4.8 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）の使用方法

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。

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

Navigation Platform for Developers 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

Navigation Platform for Developers インストールディレクトリ¥pluginSDK¥plugin

2. 定数「DEFAULT_SAVE_FOLDER」の指定値を編集して、中断情報を出力するフォルダを変更します。

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、「4.6 プラグインのビルド」を参照してください。

4. 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「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。

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。

- 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。
- 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 (x86)¥Hitachi
¥HNP¥lib¥ucnppluginif.jar'
Project 'OutputGuideData' is missing required library: 'C:¥Program Files (x86)¥Hitachi
¥HNP¥syslib¥ucnpnpsys.jar'
Project 'OutputGuideData' is missing required library: 'C:¥Program Files (x86)¥Hitachi
¥HNP¥lib¥ucnpclasses.jar'
Project 'OutputGuideData' is missing required library: 'C:¥Program Files (x86)¥Hitachi
¥HNP¥PP¥uCPSB¥CC¥client¥lib¥j2ee-javax.jar'
```

この場合は、メッセージに表示された JAR ファイルのパスを、次のとおりに設定し直してください。

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¥lib¥ucnppluginif.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¥syslib¥ucnpnpsys.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¥lib¥ucnpclasses.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¥PP¥uCPSB¥CC¥client¥lib¥j2ee-javax.jar

付録 B 入出力プラグイン開発でのポイント

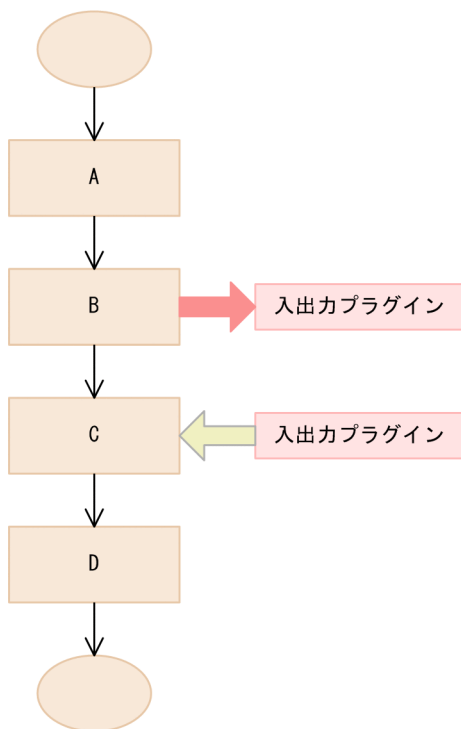
ここでは、入出力プラグインを開発でのポイントについて説明します。

付録 B.1 マッピング線の有無による、入出力プラグインの実行抑止

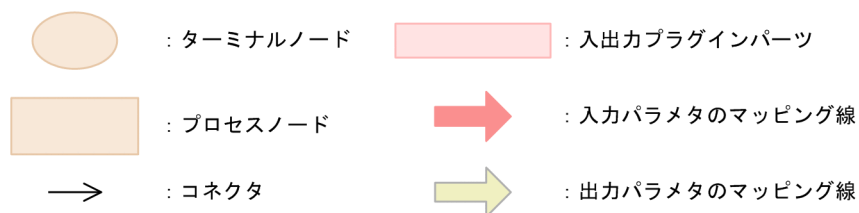
入出力プラグインの入出力パラメタと、ガイドパーツ間のマッピング線の有無によって、入出力プラグインの処理を抑止できます。

次の図に示す業務コンテンツを例に、入出力プラグインの処理の抑止について説明します。

図 B-1 業務コンテンツの例



(凡例)



この図の業務コンテンツで、プロセスノード A からプロセスノード D に遷移した場合、次に示す順番で入出力プラグインが動作します。

1. プロセスノード B に配置した入出力プラグインの `outputToNode` メソッド
2. プロセスノード C に配置した入出力プラグインの `inputFromNode` メソッド

3. プロセスノード C に配置した入出力プラグインの `outputToNode` メソッド

4. プロセスノード C に配置した入出力プラグインの `inputFromNode` メソッド

1. で実行されるメソッドの処理を抑止したい場合、次に示す内容を確認して処理を抑止してください。

- `outputToNode` メソッドの引数 `param` の `ucnp.next.params.map` キーの値が空の Map であること

4. で実行されるメソッドの処理を抑止したい場合、次に示す内容を確認して処理を抑止してください。

- `inputFromNode` メソッドの引数 `param` の `ucnp.current.params.map` キーの値が空の Map であること。

なお、4. で実行されるメソッドの処理を抑止したくない場合は、プロセスノード C の入出力プラグインの入力パラメタにマッピング線を引くことによって、抑止しないで処理できます。

付録 C 旧バージョンからの移行について

旧バージョンの開発環境の設定や開発したファイルを、ナビゲーション プラットフォームに移行する手順について説明します。また、旧バージョンから移行できない機能や移行不要な機能もあるため、それらについても説明します。

付録 C.1 旧バージョンで開発したプラグインを移行する手順

09-50 以降の旧バージョンで開発した入出力プラグインおよび中断再開プラグインは、次の手順で移行するとナビゲーション プラットフォームで使用できるようになります。

1. [4.1 ひな形プラグインの作成] を参照して、新しいひな形プラグインを作成します。

新しいひな形プラグインのプラグイン情報プロパティファイルに 09-50 以降の旧バージョンで指定していた値を反映します。

反映する値

userplugin.id キーの指定値

userplugin.name キーの指定値

userplugin.server.controller.ioaction.type キーの指定値

09-50 以降の旧バージョンのプラグインのプラグイン情報プロパティファイルの格納場所については、開発時に使用したマニュアルを参照してください。

2. 09-50 以降の旧バージョンのファイルを新しいひな形プラグインのファイルに上書きします。

表 C-1 上書きするファイル一覧 (共通)

項番	09-50 以降の旧バージョンのファイル	コピー先フォルダ	コピーが必要な場合
1	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¥プラグイン名¥WEB-INF¥lib 直下のファイル	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 ¥pluginSDK¥plugin¥プラグイン名¥WEB-INF¥lib	アプリケーションクラスロードにロードするライブラリファイルを追加している場合
2	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 直下のファイル※1		
3	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 直下のファイル※2	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	システムクラスロードにロードするライブラリファイルを追加している場合
4	J2EE サーバ用オプション定義ファイル (usrconf.cfg) で、開発者が追加した add.class.path キーに指定したパス直下のファイル※2		

項番	09-50 以降の旧バージョンのファイル	コピー先フォルダ	コピーが必要な場合
5	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 ¥base¥dd¥META-INF¥ cosminexus.xml ^{※1}	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¥プラグイン名¥dd¥META-INF	リソースアダプタの設定を追加している場合
6	ポータルプロジェクト ¥WEB-INF ¥web.xml ^{※1}	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¥プラグイン名 ¥WEB-INF	

注※1

旧バージョンではすべてのプラグインで共有されていたファイルですが、ナビゲーション プラットフォームでは共有されなくなります。そのため、すべてのプラグインに個別にファイルを配置する必要があります。

注※2

J2EE サーバ用オプション定義ファイル (usrconf.cfg) にも、クラスパスを設定する必要があります。

3. 入出力プラグインの場合は、次に示すファイルも新しいひな形プラグインのファイルに上書きします。

表 C-2 上書きするファイル一覧 (入出力プラグイン)

項番	09-50 以降の旧バージョンのファイル	コピー先フォルダ	注意事項
1	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¥プラグイン名 ¥images 直下のファイル	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 ¥pluginSDK¥plugin¥プラグイン名 ¥images	旧バージョンのファイル数の方が多い場合も、すべてのファイルをコピーしてください。
2	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¥プラグイン名 ¥WEB-INF ¥conf ¥ioaction.xml	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¥プラグイン名 ¥WEB-INF ¥conf	<p>上書きコピーではなく、コピー先フォルダにある同名のファイルを編集して、旧バージョンのファイルの内容を反映してください。</p> <p>iconURL タグに指定する URL は旧バージョンと異なるため、特に注意が必要です。</p> <p>旧バージョンの iconURL タグの指定値 "/プラグイン名/images/ファイル名"</p> <p>新しい iconURL タグの指定値 "/ucnpPlugins/プラグイン名/images/ファイル名"</p> <ul style="list-style-type: none"> 09-60 から移行する場合 diableButtonType タグで指定しているボタン種別でボタンによる遷移と直接遷移による遷移が区別できるようになっていま

項番	09-50 以降の旧バージョンのファイル	コピー先フォルダ	注意事項
2	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¥プラグイン名¥WEB-INF¥conf ¥ioaction.xml	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¥プラグイン名¥WEB-INF¥conf	す。09-60 と同じように動作させたい場合、legacy 属性に true を指定してください。
3	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¥プラグイン名¥WEB-INF¥src¥/java パッケージ¥controller¥入出力アクションコントローラ.java	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¥プラグイン名¥WEB-INF¥src¥/java パッケージ¥controller	内容を維持したまま文字コードを UTF-8 に変更し、上書きしてください。

4. 中断再開プラグインの場合は、次に示すファイルも新しいひな形プラグインのファイルに上書きします。

表 C-3 上書きするファイル一覧 (中断再開プラグイン)

項番	09-50 以降の旧バージョンのファイル	コピー先フォルダ	注意事項
1	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¥プラグイン名¥WEB-INF¥src¥/java パッケージ¥controller¥入出力アクションコントローラ.java	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¥プラグイン名¥WEB-INF¥src¥/java パッケージ¥controller	内容を維持したまま文字コードを UTF-8 に変更し、上書きしてください。

5. リソースアダプタを利用するプラグインの場合は、「4.4 データベースとの接続処理の追加」を参照してリソースアダプタの設定を追加します。

6. 新規にプラグインを作成する場合と同様に、ビルド、デプロイ、およびデバッグを実施します。

- 「4.6 プラグインのビルド」
- 「4.7 プラグインのデプロイ」
- 「4.10 プラグインのデバッグ」

付録 C.2 旧バージョンで開発したメニュー領域の移行について

旧バージョンでは、HTML または JSP ファイルを使用してメニュー領域をカスタマイズできましたが、ナビゲーション プラットフォームではメニュー領域のカスタマイズはできません。

付録 C.3 旧バージョンで開発したカスタム画面（新規画面）の移行について

旧バージョンでは、カスタム画面（新規画面）はプラグインとは別に開発していましたが、ナビゲーションプラットフォームではカスタム画面（新規画面）もプラグインとして開発し直す必要があります。

また、旧バージョンでは検索機能を利用するためにはカスタム画面が必要でしたが、ナビゲーションプラットフォームではデフォルトの機能として提供されているので、開発は不要です。

カスタム画面用プラグインの開発方法については、「[4. プラグインの開発](#)」を参照してください。

付録 C.4 旧バージョンでの J2EE サーバの設定について

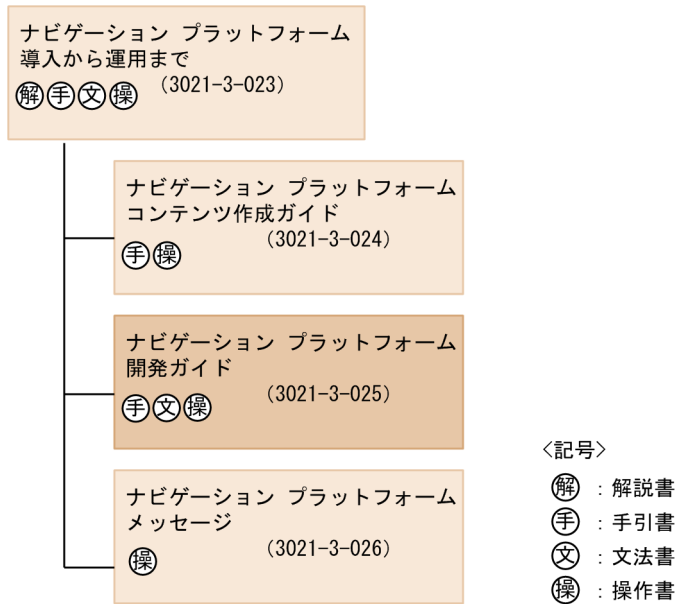
旧バージョンの開発環境で J2EE サーバの設定を変更していた場合、「[4.13.1 プラグイン開発時に変更できる J2EE サーバの設定項目](#)」に記載されている項目以外はデフォルト値に戻してください。

付録 D 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 D.1 関連マニュアル

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。なお、このマニュアルの本文中では「Hitachi Navigation Platform」を省略して表記しています。



- Hitachi Navigation Platform ナビゲーション プラットフォーム 導入から運用まで (3021-3-023)
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。
- Hitachi Navigation Platform 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 (3021-3-024)
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。
- Hitachi Navigation Platform ナビゲーション プラットフォーム 開発ガイド (3021-3-025)
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。
- Hitachi Navigation Platform ナビゲーション プラットフォーム メッセージ (3021-3-026)
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。

付録 D.2 このマニュアルでの表記

このマニュアルでは、製品および関連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 Primary Server Base
Eclipse	Eclipse Web Tools Platform

注※

uCosminexus Navigation Platform について説明している場合は、「Cosminexus」は uCosminexus Application Server と uCosminexus Developer の総称です。

付録 D.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
UTF	UCS Transformation Format
WAR	Web ARchive
XML	eXtensible Markup Language

付録 D.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, 1,024² バイト, 1,024³ バイト, 1,024⁴ バイトです。

(英字)

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。

(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。

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。Hitachi Navigation Platform for Developers, または JP1/Navigation Platform for Developers が必要です。

カスタム画面プラグイン

ナビゲーション プラットフォームの画面とは別の画面を開発するためのプラグインです。このプラグインで開発した画面をカスタム画面と呼びます。

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
- 業務コンテンツの公開・非公開
- 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ーのことです。

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ーのことです。

(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。Hitachi Navigation Platform または JP1/Integrated Management - 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

Hitachi Navigation Platform, JP1/Integrated Management - Navigation Platform, Hitachi Navigation Platform for Developers, および JP1/Navigation Platform for Developers 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。Hitachi Navigation Platform または JP1/Integrated Management - Navigation Platform 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

A

- API 一覧 (カスタム画面プラグイン開発の場合) 144
- API 一覧 (中断再開プラグイン開発の場合) 123
- API 一覧 (入出力プラグイン開発の場合) 94
- API リファレンス (カスタム画面プラグイン開発の場合) 143
- API リファレンス (中断再開プラグイン開発の場合) 122
- API リファレンス (入出力プラグイン開発の場合) 93

C

- contains メソッド 128
- CustomWindowUrlUtil 145

D

- decodeHtmlPartParam メソッド 114
- deleteAll メソッド 131
- delete メソッド 129
- deserialize メソッド 139
- destroy メソッド 112

G

- GC (ガーベージコレクション) 81
- getContentName メソッド 135
- getContentSuspendedId メソッド 134
- getCurrentNodeName メソッド 136
- getCustomWindowUrl メソッド 145
- getGroupName メソッド 136
- getLoginId メソッド 121
- getMessageId メソッド [UCNPPluginException (入出力プラグイン例外クラス)] 119
- getMessage メソッド 119
- getWorkId メソッド 135

I

- IloPluginController 95
- init メソッド 111

- inputFromNode メソッド 95
- ioaction.xml [入出力プラグイン用 XML ファイル] 46
- 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 90
- IPluginInitializer 111
- ISuspendActionController 124
- ISuspendInfo 134
- IUCNPSession 121

J

- J2EE サーバ用オプション定義ファイル 86
- J2EE サーバ用ユーザプロパティファイル 83
- Java ヒープのメモリ使用量 83

L

- load メソッド 126
- LogoutActionUtil 147
- logout メソッド 147

M

- Map 形式 114

O

- outputToNode メソッド 104

P

- ParamConvertUtil 114
- param パラメタ [inputFromNode メソッド] 97
- param パラメタ [outputToNode メソッド] 106
- pluginSDK プロジェクトのインポート 31
- [Plugins] パレットおよびプラグインパーツのアイコンの作成 45

S

- save メソッド 124
- serialize メソッド 138
- SuspendInfoSerializeUtil 138

U

ucnp.options.param キーの指定例 (画面 ID 単位)
(推奨) 99

ucnp.request.options キー 96

ucnp.request.options キーの指定例 (セッション単位) 96

ucnpCustom.jsp 56

ucnpOptions パラメタ (session) 96

UCNPPluginException(String message, Throwable cause)コンストラクタ 142

UCNPPluginException(String message)コンストラクタ 141

UCNPPluginException (中断再開プラグイン例外クラス) 141

UCNPPluginException (入出力プラグイン例外クラス) 119

UCNPPluginUserException 113

UCNPPluginUserException(String, Throwable)コンストラクタ 113

UCNPPluginUserException(String)コンストラクタ 113

ucnpUserData パラメタ 27

userplugin.id 36

userplugin.java.package 37

userplugin.name 36

userplugin.server.controller.ioaction 38

userplugin.server.controller.ioaction.type 38

userplugin.server.controller.suspend 39

userplugin.type 37

userplugin.version 37

usrconf.cfg 86

usrconf.properties 83

W

Web サーバ [用語解説] 161

あ

アイコン 45

アイコン [格納ディレクトリ] 46

アイコン [作成形式] 46

アプリケーションサーバ [用語解説] 161

い

移行 155

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

開発の概要 13

開発の準備 28

開発の範囲 14

確認ダイアログの表示有無の指定 53

カスタム画面 URL 取得ユーティリティクラス 145

カスタム画面プラグインのカスタマイズ 56

カスタム画面プラグイン [用語解説] 161

カスタム画面用の JSP ファイル 56

画面 ID [用語解説] 162

関連線 [用語解説] 162

関連づけ 70

き

キャッシュ値 102

キャッシュ [用語解説] 162

旧バージョンで開発したプラグイン 91

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

こ

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 95

サンプルプラグインの使用方法 150
サンプルプラグインを使用するときの注意事項 151

し

システムプラグイン [用語解説] 162
実行環境 [用語解説] 163
実行を抑止するボタン種別の指定 54

せ

セッション情報利用インタフェース 121

そ

測定用の業務コンテンツの作成手順 80

た

ターミナルノード [用語解説] 163

ち

中断再開アクションコントローラインタフェース 124
中断再開プラグイン [用語解説] 163
中断再開プラグイン (サンプル) の使用方法 151
中断再開プラグインのカスタマイズ 55
中断再開プラグイン例外クラス 141
中断情報 [用語解説] 163
中断情報インタフェース 134
中断情報直列化ユーティリティクラス 138

て

データベースとの接続処理の実装 62
データベースとの接続処理のためのプラグインの設定 61
データベースとの接続処理の追加 58
デバッグ 77
デバッグするための条件 77
デプロイ 68

な

ナビゲーション プラットフォームのプラグインとは 19
ナビゲーション プラットフォーム [用語解説] 163

に

入出力パラメタ変換ユーティリティクラス 114
入出力プラグイン (サンプル) の使用方法 150
入出力プラグイン開発でのポイント 153
入出力プラグインのカスタマイズ 45
入出力プラグイン用 XML ファイル [格納ディレクトリ] 48
入出力プラグイン用 XML ファイルとは 38
入出力プラグイン用 XML ファイルの編集 46
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 119

の

ノード [用語解説] 164

は

パーツ [用語解説] 164
パラメタ説明用ツールチップの設定 52

ひ

ひな形プラグイン作成コマンドの実行 40
ひな形プラグインのカスタマイズ 45
ひな形プラグインの作成 35
ひな形プラグインのプロジェクトのインポート 44
ひな形プラグイン [用語解説] 164

ふ

プラグイン開発の流れ 15
プラグインが実行する処理の実装 55
プラグインが使用するメモリ使用量の計算 80
プラグインが使用するメモリ使用量の測定手順 80
プラグイン実行順序の指定 53
プラグイン情報プロパティファイルの編集 35
プラグイン情報プロパティファイル [用語解説] 164
プラグインに入力, または出力する値の詳細 73
プラグインの開発 34
プラグインの概要 18
プラグインの削除 79

プラグインのデプロイ 68
プラグインのビルド 65
プラグインのビルド手順 65
プラグインのビルドに失敗したときの対処 66
プラグインパーツ [用語解説] 164
プラグインパーツを更新する 74
プラグイン [用語解説] 164
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 75
プレビュー画面での実行有無の指定 53
プレビュー時のプラグイン処理 110
プロセスノード [用語解説] 164
プロパティファイル作成時の注意 35
プロパティファイルの記述形式 35
分岐合流ノード [用語解説] 164

へ

編集環境 [用語解説] 164

ま

マッピング線の有無による、入出力プラグインの実行抑止 153
マッピング線 [用語解説] 165
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) 70
マニュアル [ナビゲーション プラットフォーム] 159

ゆ

ユーザプラグインが利用する Java パッケージへのアクセス権の設定 92
ユーザプラグインの起動 (初期化)・終了時の処理の実装インタフェース 111
ユーザプラグイン [用語解説] 165
ユーザプラグイン例外クラス 113
ユーザプロパティファイルの設定 76

ら

ライブラリ追加時の注意事項 33
ライブラリの作成と設定 32
ライブラリの追加 32

ライブラリの配置場所 32

り

リソースアダプタの削除 60
リソースアダプタの設定変更 59
リソースアダプタの追加 58

ろ

ログアウト処理クラス 147