

uCosminexus Service Coordinator Interactive  
Workflow AP 開発ガイド

3020-3-M81-90

## 前書き

### ■ 対象製品

P-2955-CJ34 uCosminexus Business Process Developer 03-11 (適用 OS : Windows Server 2016, Windows Server 2019, Windows Server 2022, Windows 10 x64, Windows 11)

P-2943-CG34 uCosminexus Service Coordinator Interactive Workflow 03-30 (適用 OS : Windows Server 2019, Windows Server 2022)

P-2955-CG34 uCosminexus Service Coordinator Interactive Workflow 03-30 (適用 OS : Windows Server 2019, Windows Server 2022, Windows 10 x64, Windows 11)

P-9W43-CG31 uCosminexus Service Coordinator Interactive Workflow 03-30 (適用 OS : Red Hat Enterprise Linux Server 7 (64-bit x86\_64), Red Hat Enterprise Linux Server 8 (64-bit x86\_64), Red Hat Enterprise Linux Server 9 (64-bit x86\_64))

これらのプログラムプロダクトのほかにもこのマニュアルをご利用になれる場合があります。詳細は「リリースノート」でご確認ください。

本製品では日立ネットワークオブジェクトプラザトレース共通ライブラリをインストールします。

### ■ 輸出時の注意

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

なお、不明な場合は、弊社担当営業にお問い合わせください。

### ■ 商標類

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

AIX は、世界の多くの国で登録された International Business Machines Corporation の商標です。

IBM は、世界の多くの国で登録された International Business Machines Corporation の商標です。

Linux は、Linus Torvalds 氏の米国およびその他の国における登録商標です。

Microsoft は、マイクロソフト 企業グループの商標です。

Oracle(R), Java, MySQL 及び NetSuite は、Oracle, その子会社及び関連会社の米国及びその他の国における登録商標です。

Red Hat, and Red Hat Enterprise Linux are registered trademarks of Red Hat, Inc. in the United States and other countries. Linux(R) is the registered trademark of Linus Torvalds in the U.S. and other countries.

Red Hat, および Red Hat Enterprise Linux は、米国およびその他の国における Red Hat, Inc. の登録商標です。Linux(R) は、米国およびその他の国における Linus Torvalds 氏の登録商標です。

SQL Server は、マイクロソフト 企業グループの商標です。

UNIX は、The Open Group の登録商標です。

Windows は、マイクロソフト 企業グループの商標です。

Windows Server は、マイクロソフト 企業グループの商標です。

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。

## ■ 発行

2024 年 3 月 3020-3-M81-90

## ■ 著作権

All Rights Reserved. Copyright (C) 2007, 2024, Hitachi, Ltd.

## 変更内容

### 変更内容(3020-3-M81-90) uCosminexus Service Coordinator Interactive Workflow 03-30

追加・変更内容	変更箇所
適用 OS に Windows Server 2022 を追加しました。	—
使用できる DBMS として PostgreSQL をサポートした。	2.11, 2.12.6, 4.2, 8.6

単なる誤字・脱字などはお断りなく訂正しました。

OS, ブラウザ, およびデータベースに関して, 新しいバージョンの追加, および古いバージョンの削除については記載していません。サポートしているバージョンの詳細については「リリースノート」でご確認ください。

## はじめに

このマニュアルは、uCosminexus Service Coordinator Interactive Workflow 上で動作する API の使用方法について説明したものです。

以降、このマニュアルでは、上記のプログラムプロダクトを **CSCIW** と称します。また、CSCIW を使用した業務システム全体の総称を **ワーク管理システム** とします。

なお、BPMN 連携機能を使用する場合に CSCIW で使用する API については、マニュアル「uCosminexus Service Coordinator Interactive Workflow BPMN 連携機能 使用の手引」を参照してください。

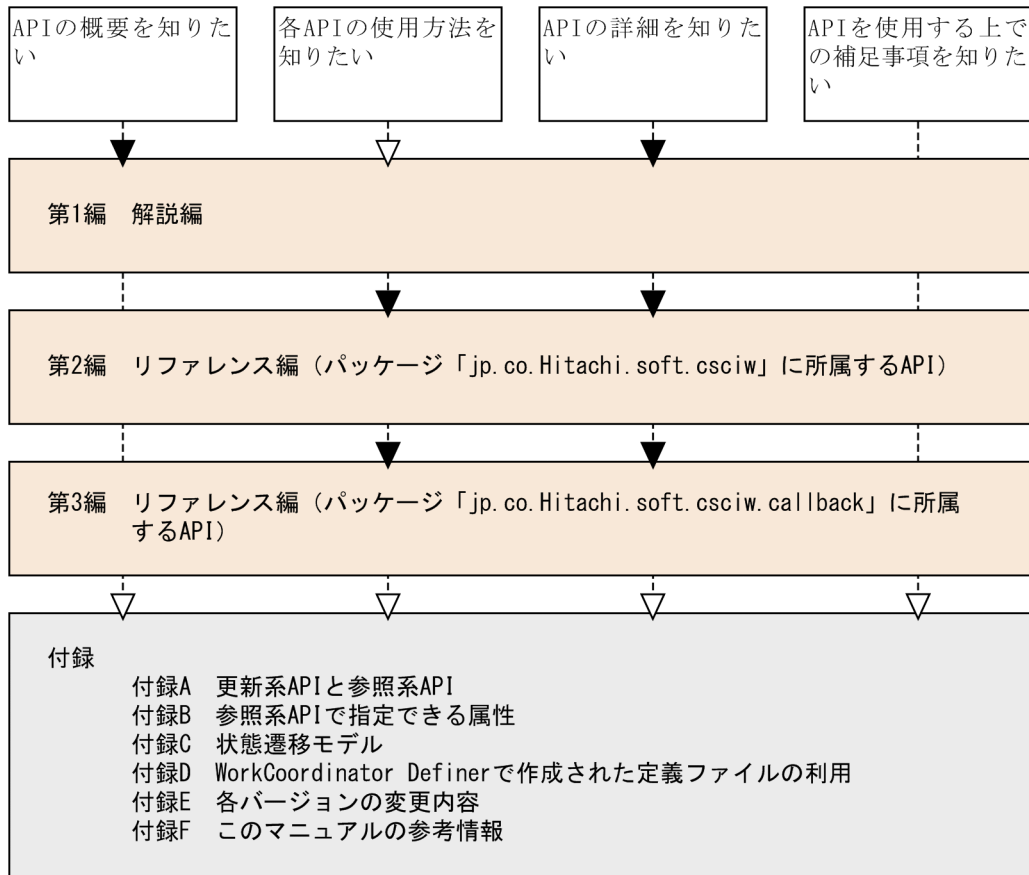
### ■ 対象読者

このマニュアルは、CSCIW を使用して対話型ワークフローを適用したシステムを開発される方を対象としています。また、次の知識をお持ちであることを前提にしています。

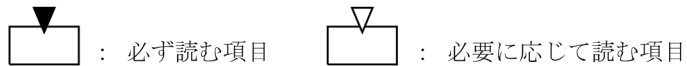
- 使用している OS に関する基本的な知識
- Cosminexus および JavaEE に関する知識
- リレーショナルデータベースの操作（SQL 文など）に関する知識

### ■ 読書手順

このマニュアルは、利用目的に合わせて章を選択して読むことができます。利用目的別に次の流れに従ってお読みいただくことをお勧めします。



(凡例)



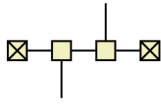
## ■ 図中で使用する記号

このマニュアルの図中で使用する記号を次のように定義します。

●工程、作業項目の流れ



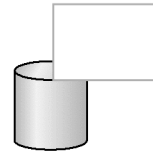
●バス型のLAN



●ファイルおよびデータベース



●ファイルの内容



●アロー



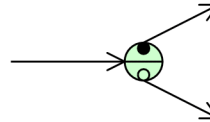
●ソースノード



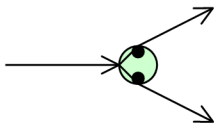
●シンクノード



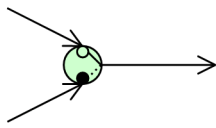
●制御ノード（分岐）



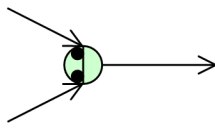
●制御ノード（分業）



●制御ノード（先着）



●制御ノード（待合）



●定義ファイル



## ■ このマニュアルで使用する記号

このマニュアルで使用する記号を次のように定義します。

記号	意味
< >	可変値を意味します。

## ■ コマンドの説明で使用する記号

コマンドの説明で使用する記号を次のように定義します。

記号	意味
< >	可変値を意味します。
(ストローク)	横に並べられた複数の項目に対し、項目間の区切りを示します。
{ }	この記号で囲まれている複数の項目のうちから1つを選択することを意味します。項目が横に並べられ、記号 で区切られている場合は、そのうちの1つを選択します。 例 {A   B}は、「Aと指定する」または「Bと指定する」ことを示します。
( )	この記号で囲まれている複数の項目のうちから1つ以上を選択することを意味します。項目が横に並べられ、記号 で区切られている場合は、そのうちの1つ以上を選択します。 例 (A   B)は、「Aと指定する」、「Bと指定する」または「AおよびBと指定する」ことを示します。
[ ]	この記号で囲まれている項目は省略してよいことを意味します。

記号	意味
	例 [A]は「何も指定しない」か「Aを指定する」ことを示します。

## ■ 操作の説明で使用する記号

操作の説明で使用する記号を次のように定義します。

記号	意味
[ ]	次のどれかを示します。 <ul style="list-style-type: none"> <li>ボタン</li> <li>キーボードのキー</li> <li>画面またはダイアログの名称</li> <li>画面またはダイアログに表示される項目</li> </ul>
[A] + [B]	+の前のキーを押したまま、後ろのキーを押すことを示します。
[AAA] - [BBB]	-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示します。

## ■ このマニュアルで使用する注記

このマニュアルで使用する注記を次に示します。

### 重要

操作を完了させるための重要な情報を示します。

### メモ

本文に対して強調したい内容または補足事項を示します。

### ヒント

操作する上で効果的な情報、指針、提案を示します。

## ■ 適用 OS の違いによる機能相違点の表記

このマニュアルで説明する機能は、適用 OS の種類（UNIX または Windows）によって、異なる場合があります。OS によって機能差がある場合、OS 名を明記しています。また、OS によってバージョン、リビジョンが異なる場合があります。バージョン、リビジョンによって操作方法などが異なる場合はそれぞれの説明に OS 名、バージョン、およびリビジョンを明記しています。



なお、UNIX および Windows の共通の説明部分に使用している「ディレクトリ」という用語は、Windows の場合は、特に断りのないかぎり、「フォルダ」と読み替えてください。また、UNIX のパスの区切り文字として使用している「/」は、Windows の場合には、特に断りのないかぎり、「¥」に読み替えてください。

# 目次

前書き	2
変更内容	4
はじめに	5

## 第1編 解説編

<b>1</b>	<b>ビジネスプロセスの開発方法</b>	<b>21</b>
1.1	CSCIW で使用するビジネスプロセス定義と振り分けルール定義の開発	22
1.2	ビジネスプロセスの変更	25
1.2.1	ビジネスプロセスの変更（定義のバージョンを更新する場合）	25
1.2.2	ビジネスプロセスの変更（バージョンを更新しない場合）	29
1.3	ビジネスプロセスを定義するときの注意事項	32
1.3.1	組み込み変数	32
1.3.2	巡回部分を含むビジネスプロセス定義	34
1.3.3	SQL文	37
<b>2</b>	<b>業務プログラムの作成方法</b>	<b>39</b>
2.1	APIの特長	40
2.2	Java 業務プログラムの作成	41
2.2.1	パッケージのインポート	41
2.2.2	業務プログラムのコンパイル	41
2.3	業務プログラムの処理の流れ	43
2.3.1	CSCIW の初期化	43
2.3.2	CIWFactory オブジェクトの取得	43
2.3.3	CIWServer オブジェクトの生成	44
2.3.4	CIWServer オブジェクトに対する Connection オブジェクトの関連づけ	45
2.3.5	CSCIW を利用した業務処理の実装	45
2.3.6	CIWServer オブジェクトに対する Connection オブジェクトの関連づけの解除	48
2.3.7	CSCIW の終了	48
2.4	条件およびルールの評価	49
2.5	Java オブジェクト呼び出し利用時の設定	50
2.6	Java オブジェクトの作成	52
2.6.1	作成時の注意事項	52
2.6.2	Java オブジェクトの実装例（条件、生成ルール、および振り分けルール）	52
2.6.3	Java オブジェクトの実装例（作業アプリケーション）	55

- 2.6.4 業務プログラムに渡される情報 56
- 2.6.5 作業アプリケーションを作成するときの注意事項 59
- 2.7 一覧情報の取得 61
  - 2.7.1 一覧情報の取得手順と基本動作 61
  - 2.7.2 一覧情報取得時のフィルター条件 63
  - 2.7.3 一覧情報取得時のソート条件 64
- 2.8 指定できるデータ 65
  - 2.8.1 オブジェクトの取得で指定できる値 65
  - 2.8.2 オブジェクトのリストの取得で指定できる値 65
  - 2.8.3 更新系 API で指定できる値 66
  - 2.8.4 その他のクラスで指定できる値 67
- 2.9 CSCIW で管理するトランザクション 69
- 2.10 業務プログラムで管理するトランザクション 70
  - 2.10.1 CIWServer とコネクションの関連づけ 70
  - 2.10.2 Java オブジェクト呼び出しとコネクションの関連づけ 71
  - 2.10.3 トランザクションの開始 72
  - 2.10.4 トランザクションの終了 72
- 2.11 業務プログラムとデータベース間での文字コード 73
- 2.12 API 利用時の注意事項 74
  - 2.12.1 スレッドからの利用について 74
  - 2.12.2 CSCIWManagementServer を停止した状態での API 実行について 74
  - 2.12.3 エラーが発生した場合のトランザクションの終了について 74
  - 2.12.4 排他およびデッドロックについて 74
  - 2.12.5 finalize()メソッド内での使用について 74
  - 2.12.6 参照系 API の使用について 75

### 3 提供 API 一覧 76

- 3.1 クラスの一覧 77
  - 3.1.1 パッケージ「jp.co.Hitachi.soft.csciw」に所属するクラス 77
  - 3.1.2 パッケージ「jp.co.Hitachi.soft.csciw.callback」に所属するクラス 79
- 3.2 クラスの構成 81
  - 3.2.1 パッケージ「jp.co.Hitachi.soft.csciw」のクラス構成 81
  - 3.2.2 パッケージ「jp.co.Hitachi.soft.csciw.callback」のクラス構成 81
- 3.3 パッケージ「jp.co.Hitachi.soft.csciw」の階層 83
  - 3.3.1 初期化・終了クラスの階層 83
  - 3.3.2 案件実行・運用クラスおよびインタフェースの階層 83
  - 3.3.3 その他の階層 83
  - 3.3.4 列挙型の階層 83
  - 3.3.5 例外クラスの階層 84

- 3.4 パッケージ「jp.co.Hitachi.soft.csciw.callback」の階層 85
- 3.4.1 インタフェースの階層 85
- 3.4.2 列挙型の階層 85
- 3.4.3 例外クラスの階層 85

## 第2編 リファレンス編 (パッケージ「jp.co.Hitachi.soft.csciw」に所属するAPI)

### 4 初期化・終了クラスが提供するメソッド 86

- 4.1 記述形式 87
  - 4.1.1 初期化・終了クラスの記述形式 87
  - 4.1.2 初期化・終了クラスが提供するメソッドの記述形式 87
- 4.2 CIWAdmin (CSCIWの初期化・終了処理をするクラス) 88
  - finalizeCIWFactory 88
  - initializeCIWFactory 89

### 5 案件実行・運用クラスおよびインタフェースが提供するメソッド 92

- 5.1 記述形式 93
  - 5.1.1 案件実行・運用クラスおよびインタフェースの記述形式 93
  - 5.1.2 案件実行・運用クラスおよびインタフェースが提供するメソッドの記述形式 93
- 5.2 CIWActivityDefinition (業務ステップ定義のインタフェース) 94
  - getDescription 94
  - getID 95
  - getName 96
  - getProcessDefinitionID 96
- 5.3 CIWActivityInstance (業務ステップのインタフェース) 98
  - adhocMakeTransition 99
  - changeState 100
  - evaluate 102
  - getActivityDefinitionID 103
  - getActivityDefinitionName 104
  - getClosedDate 105
  - getDeadline 106
  - getID 106
  - getName 107
  - getPriority 108
  - getProcessDefinitionID 109
  - getProcessInstanceID 109
  - getProcessInstanceName 110
  - getStartDate 111
  - getState 112
  - getType 113
  - makeBackwardTransition 113
  - setDeadline 115

	setPriority	116	
5.4	CIWFactory (CIWFactory と CIWServer オブジェクトの生成クラス)	118	
	createCIWServer	118	
	createCIWServer	119	
	getInstance	121	
5.5	CIWProcessDefinition (ビジネスプロセス定義のインタフェース)	122	
	getAuthor	123	
	getCreationDate	123	
	getDescription	124	
	getID	125	
	getName	125	
	getProcessInstancesList	126	
	getResponsible	128	
	getState	129	
	getValidFromDate	129	
	getValidToDate	130	
	getVersion	131	
5.6	CIWProcessInstance (案件のインタフェース)	133	
	ad hocInstantiateActivityInstance	134	
	changeState	136	
	delete	137	
	getClosedDate	138	
	getCreator	139	
	getDeadline	140	
	getID	140	
	getMovedDate	141	
	getName	142	
	getPriority	143	
	getProcessDefinition	144	
	getProcessDefinitionID	145	
	getProcessDefinitionName	145	
	getStartDate	146	
	getState	147	
	getWorkItemsList	148	
	setDeadline	149	
	setPriority	151	
	start	152	
	startAndAd hocInstantiateActivityInstanceByADName	153	
	terminate	154	
5.7	CIWServer (ワーク管理システム全般にかかわる処理を行うための機能を提供するインタフェース)	156	
	allocateWorkItemEx	157	
	createAndStartProcessInstance	159	
	createProcessInstance	161	
	getActivityDefinition	163	

- getActivityDefinitionsList 164
- getActivityInstance 166
- getActivityInstancesList 168
- getConnectionCoordinator 169
- getListCount 170
- getProcessDefinition 171
- getProcessDefinitionsList 172
- getProcessInstance 174
- getProcessInstancesList 176
- getWorkDefinition 177
- getWorkDefinitionsList 179
- getWorkItem 180
- getWorkItemsList 182
- 5.8 CIWorkDefinition (作業定義のインタフェース) 184
  - getActivityDefinitionID 184
  - getCastingRuleName 185
  - getDescription 186
  - getID 187
  - getName 187
  - getProcessDefinitionID 188
  - getType 189
- 5.9 CIWorkItem (作業のインタフェース) 190
  - changeState 191
  - complete 194
  - evaluate 195
  - free 196
  - getActivityInstanceID 197
  - getClosedDate 197
  - getCreationDate 198
  - getDeadline 199
  - getID 200
  - getName 200
  - getParentProcessInstance 201
  - getParticipant 202
  - getPriority 203
  - getProcessDefinitionID 204
  - getProcessInstanceID 205
  - getProcessInstanceName 205
  - getStartDate 206
  - getState 207
  - getType 208
  - getWorkDefinition 208
  - getWorkDefinitionID 210
  - getWorkDefinitionName 210
  - perform 211

performAndComplete 212  
reassign 213  
reassignAndPerform 214  
reevaluateCastingRule 216  
setDeadline 217  
setPriority 218

## 6 その他のクラス 220

- 6.1 記述形式 221
  - 6.1.1 その他のクラスの記述形式 221
  - 6.1.2 その他のクラスが提供するメソッドの記述形式 221
- 6.2 CIWConnectionCoordinator (CIWServer と JDBC コネクションの関連づけを管理するクラス) 222
  - attachDatabaseConnection 222
  - detachDatabaseConnection 223
- 6.3 CIWUtil (各種ユーティリティ機能を提供するクラス) 225
  - convertDateToString 225
  - convertDateToString 226

## 7 列挙型 228

- 7.1 記述形式 229
  - 7.1.1 列挙型の記述形式 229
  - 7.1.2 列挙型が提供するメソッドの記述形式 229
- 7.2 CIWActivityDefinition.AttributeName (業務ステップ定義属性の属性名の列挙型) 231
  - toAttributeName 232
  - valueOf 232
  - values 233
- 7.3 CIWActivityInstance.AttributeName (業務ステップ属性の属性名の列挙型) 235
  - toAttributeName 237
  - valueOf 237
  - values 238
- 7.4 CIWActivityInstance.State (業務ステップの状態の列挙型) 240
  - fromStateCode 242
  - isClosed 243
  - isCompleted 244
  - isExecuted 244
  - isNotRunning 245
  - isNotStarted 246
  - isOpen 246
  - isSuspended 247
  - toShortState 247
  - toState 248
  - toStateCode 248
  - toString 249

	valueOf	250
	values	251
7.5	CIWActivityInstance.Type (業務ステップの種類 of 列挙型)	252
	fromTypeCode	253
	toString	254
	toType	255
	toTypeCode	255
	valueOf	256
	values	257
7.6	CIWProcessDefinition.AttributeName (ビジネスプロセス定義属性 of 属性名 of 列挙型)	258
	toAttributeName	259
	valueOf	260
	values	261
7.7	CIWProcessDefinition.State (ビジネスプロセス定義 of 状態 of 列挙型)	262
	fromStateCode	263
	toShortState	264
	toState	265
	toStateCode	265
	toString	266
	valueOf	266
	values	267
7.8	CIWProcessInstance.AttributeName (案件属性 of 属性名 of 列挙型)	269
	toAttributeName	270
	valueOf	271
	values	272
7.9	CIWProcessInstance.State (案件 of 状態 of 列挙型)	273
	fromStateCode	275
	isClosed	276
	isNotRunning	276
	isOpen	277
	isSuspended	277
	toShortState	278
	toState	278
	toStateCode	279
	toString	280
	valueOf	280
	values	281
7.10	CIWServer.Targetkind (ワーク管理システムが管理するオブジェクト of 種別を表す列挙型)	283
	toTargetkind	284
	valueOf	284
	values	285
7.11	CIWUtil.InfinityDate (無限遠 of 日付 of 列挙型)	287
	toDate	288
	valueOf	288



	values	289	
7.12	CIWorkDefinition.AttributeName (作業定義属性の属性名の列挙型)	291	
	toAttributeName	292	
	valueOf	292	
	values	293	
7.13	CIWorkDefinition.Type (作業定義の種類列挙型)	295	
	fromTypeCode	296	
	toString	297	
	toType	298	
	toTypeCode	299	
	valueOf	299	
	values	300	
7.14	CIWorkItem.AttributeName (作業属性の属性名の列挙型)	302	
	toAttributeName	304	
	valueOf	304	
	values	305	
7.15	CIWorkItem.State (作業の状態列挙型)	307	
	fromStateCode	310	
	isClosed	310	
	isCompleted	311	
	isNotRunning	312	
	isNotStarted	312	
	isOpen	313	
	isRunning	313	
	isSuspended	314	
	toShortState	314	
	toState	315	
	toStateCode	316	
	toString	316	
	valueOf	317	
	values	318	
7.16	CIWorkItem.Type (作業の種類列挙型)	319	
	fromTypeCode	320	
	toString	321	
	toType	322	
	toTypeCode	323	
	valueOf	323	
	values	324	
<b>8</b>	<b>例外クラス</b>	<b>326</b>	
8.1	記述形式	327	
8.1.1	例外クラスの記述形式	327	
8.1.2	例外クラスが提供するメソッドの記述形式	327	
8.2	CIEntityNotExistException (処理するオブジェクトが存在しない場合)	328	

- 8.3 CIWException (例外の基底の抽象クラス) 329
- 8.4 CIWFatalException (障害発生の場合) 330
- 8.5 CIWSQLException (DBMS の障害発生の場合) 331
  - getCause 331
  - getErrorCode 332
  - getSQLState 332
- 8.6 CIWSQLTransientException (DBMS の一時的な障害発生の場合) 334
  - getCause 335
  - getErrorCode 335
  - getSQLState 336
- 8.7 CIWStateException (不正な状態の場合) 337
- 8.8 CIWTransientException (一時的な障害発生の場合) 338
- 8.9 CIWTransitionFailedException (案件処理中のエラーの場合) 339

### 第3編 リファレンス編 (パッケージ「jp.co.Hitachi.soft.csciw.callback」に所属するAPI)

## 9 Java オブジェクト呼び出し用のインタフェース 340

- 9.1 記述形式 341
  - 9.1.1 条件またはルールの評価・作業アプリケーション用クラスおよびインタフェースの記述形式 341
  - 9.1.2 条件またはルールの評価・作業アプリケーション用クラスおよびインタフェースが提供するメソッドの記述形式 341
- 9.2 CIWConditionEvaluator (条件 Java オブジェクトのインタフェース) 342
  - evaluateCondition 342
- 9.3 CIWCreationRuleEvaluator (生成ルール Java オブジェクトのインタフェース) 344
  - evaluateCreationRule 344
- 9.4 CIWResourceSelector (振り分けルール Java オブジェクトのインタフェース) 346
  - getAvailableParticipant 346
- 9.5 CIWWorkApplication (作業アプリケーション Java オブジェクトのインタフェース) 348
  - startSynchronously 348

## 10 列挙型 350

- 10.1 記述形式 351
  - 10.1.1 列挙型の記述形式 351
  - 10.1.2 列挙型が提供するメソッドの記述形式 351
- 10.2 CIWConditionEvaluator.Parameter (条件に渡されるパラメタの列挙型) 353
  - toParameterName 355
  - valueOf 356
  - values 357
- 10.3 CIWCreationRuleEvaluator.Parameter (生成ルールに渡されるパラメタの列挙型) 358
  - toParameterName 360
  - valueOf 361

	values	362	
10.4	CIWResourceSelector.Parameter (振り分けルールに渡されるパラメタの列挙型)	363	
	toParameterName	365	
	valueOf	366	
	values	366	
10.5	CIWorkApplication.Parameter (作業アプリケーションに渡されるパラメタの列挙型)	368	
	toParameterName	370	
	valueOf	371	
	values	372	
10.6	CIWorkApplication.Result (作業アプリケーションの業務処理結果の列挙型)	373	
	valueOf	374	
	values	374	

## 11 例外クラス 376

11.1	記述形式	377	
11.1.1	例外クラスの記述形式	377	
11.1.2	例外クラスが提供するメソッドの記述形式	377	
11.1.3	例外クラスが提供するコンストラクタの記述形式	377	
11.2	CIWUserException (Java オブジェクト呼び出し時にエラーが発生した場合)	379	
	CIWUserException (ユーザエラーメッセージを使用する場合)	379	
	CIWUserException (ユーザエラーメッセージおよびエラー原因を使用する場合)	380	

## 付録 382

付録 A	更新系 API と参照系 API	383	
付録 A.1	更新系 API	383	
付録 A.2	参照系 API	384	
付録 B	参照系 API で指定できる属性	386	
付録 B.1	指定できる属性一覧	386	
付録 B.2	指定できる属性値	391	
付録 C	状態遷移モデル	394	
付録 C.1	案件の状態遷移	394	
付録 C.2	業務ステップの状態遷移	398	
付録 C.3	作業の状態遷移	404	
付録 D	WorkCoordinator Definer で作成された定義ファイルの利用	411	
付録 E	各バージョンの変更内容	414	
付録 E.1	03-11 の変更内容	414	
付録 E.2	03-10 の変更内容	414	
付録 E.3	03-00 の変更内容	414	
付録 E.4	02-30 の変更内容	415	
付録 E.5	02-20 の変更内容	415	
付録 E.6	02-10 の変更内容	415	

付録 E.7	02-00 の変更内容	415
付録 E.8	01-95 の変更内容	416
付録 E.9	01-90 の変更内容	416
付録 E.10	01-70 の変更内容	416
付録 E.11	01-60 の変更内容	417
付録 F	このマニュアルの参考情報	418
付録 F.1	関連マニュアル	418
付録 F.2	このマニュアルでの表記	419
付録 F.3	マイクロソフト製品の表記について	420
付録 F.4	このマニュアルで使用している略語	421
付録 F.5	KB (キロバイト) などの単位表記について	421

## 索引 422

# 1

## ビジネスプロセスの開発方法

この章では、CSCIW で使用するビジネスプロセスの開発方法について説明します。

## 1.1 CSCIW で使用するビジネスプロセス定義と振り分けルール定義の開発

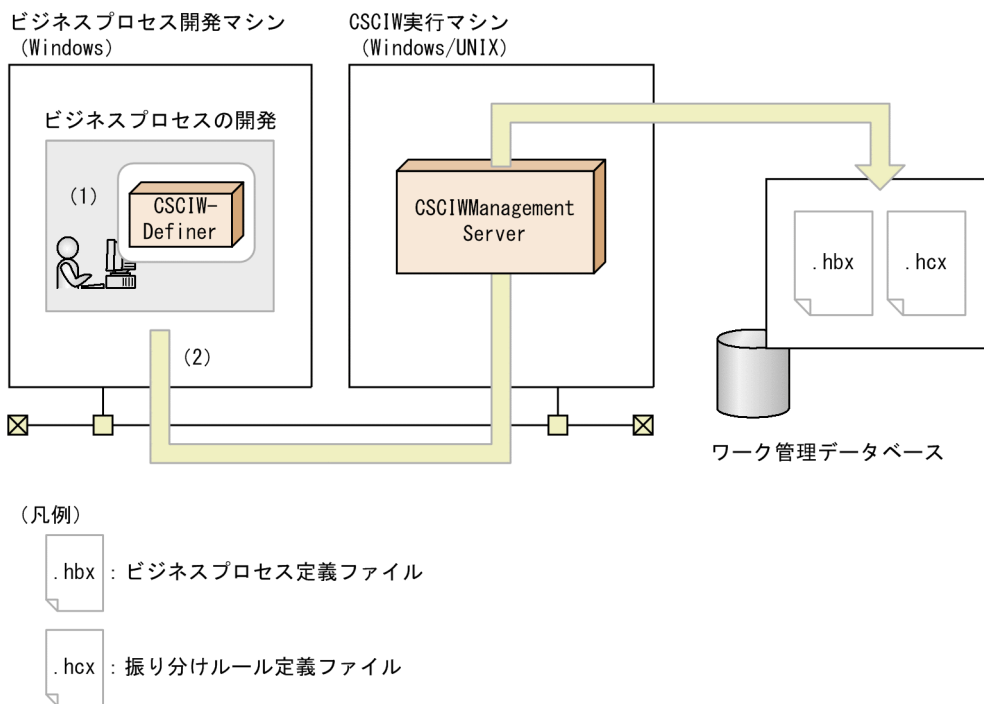
CSCIW で使用するビジネスプロセス（ビジネスプロセス定義および振り分けルール定義）は、CSCIW-Definer を使用して開発します。

開発したビジネスプロセスをチェックし、エラーがないことを確認します。確認したあと、CSCIW-Definer、またはコマンドを使用して、ビジネスプロセスを登録します。

ビジネスプロセスの登録の流れを次の図に示します。

なお、図中の数字は、説明文の項番と対応しています。

図 1-1 ビジネスプロセスの登録の流れ（CSCIW-Definer を使用する場合）

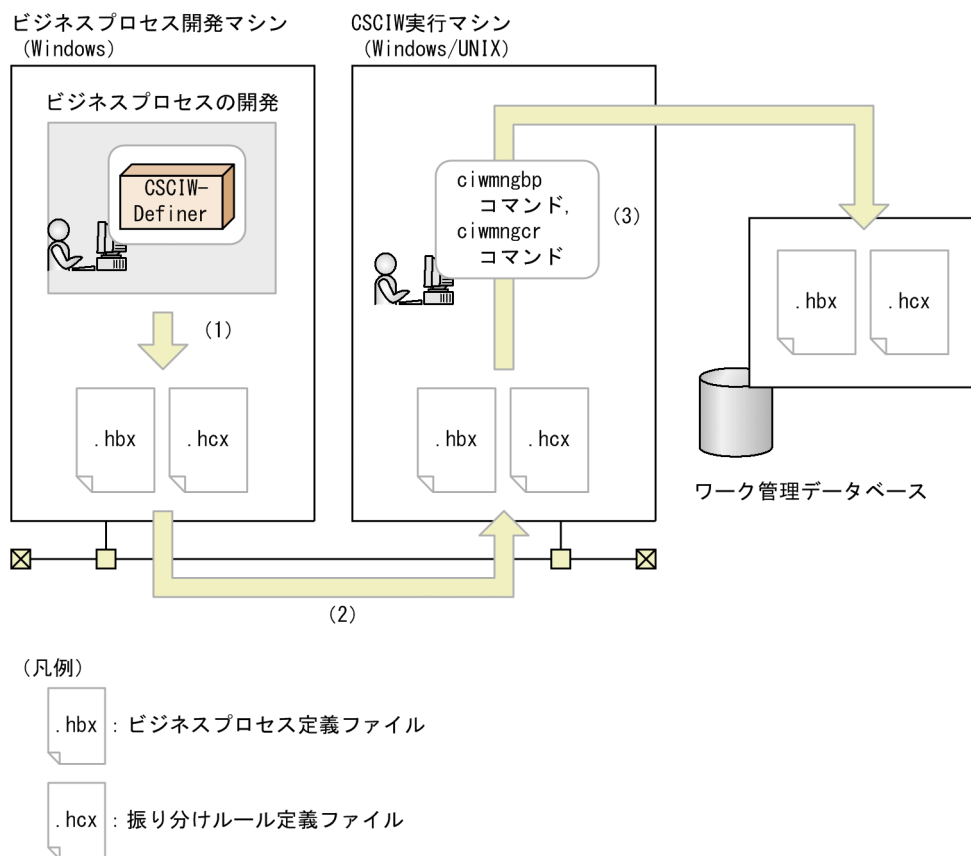


### 説明

1. 登録するビジネスプロセス定義または振り分けルール定義を指定します。
2. [登録] メニューを選択し、登録処理を実行します。

登録処理を実行すると、CSCIW Management Server を経由して、開発したビジネスプロセス定義または振り分けルール定義の内容がワーク管理データベースに登録されます。

図 1-2 ビジネスプロセスの登録の流れ（コマンドを使用する場合）



## 説明

1. 開発したビジネスプロセス定義および振り分けルール定義をローカルファイルに保存します。
2. ローカルファイルに保存したビジネスプロセス定義ファイルおよび振り分けルール定義ファイルを CSCIW 実行マシンにファイル転送プロトコル（FTP）などで転送します。なお、FTP の転送モードは、バイナリモードで転送する必要があります。また、CSCIW 実行マシンが複数ある場合は、任意の 1 台に転送してください。
3. 転送されたビジネスプロセス定義ファイルおよび振り分けルール定義ファイルを指定し、`ciwmngbp`（ビジネスプロセス定義の管理）コマンドおよび `ciwmngcr`（振り分けルール定義の管理）コマンドを実行します。

コマンドを実行すると、指定したビジネスプロセス定義ファイルおよび振り分けルール定義ファイルの内容がワーク管理データベースに登録されます。

## 注意事項

ビジネスプロセス定義および振り分けルール定義を追加、変更、または削除したあとは、すべての J2EE サーバにある CSCIWManagementServer、および CSCIW を使用している Java アプリケーションを再起動してください。再起動することで、キャッシュしているビジネスプロセス定義および振り分けルール定義が反映されます。詳細は、マニュアル「uCosminexus Service Coordinator Interactive Workflow システム構築・運用ガイド」の「ビジネスプロセス定義および振り分けルール定義の変更」を参照してください。

## 参考事項

CSCIW-Definer については、マニュアル「uCosminexus Service Coordinator Interactive Workflow ビジネスプロセス開発ガイド」を参照してください。

コマンドについては、マニュアル「uCosminexus Service Coordinator Interactive Workflow コマンド」を参照してください。



## 1.2 ビジネスプロセスの変更

業務システム運用中に、業務の流れを変更する場合は、ビジネスプロセス自体の変更が必要になります。ビジネスプロセスを変更する場合、登録済みの定義ファイルを取得し、定義を変更します。定義ファイルの内容を変更するときに、定義のバージョン番号を更新すると、すでに運用中の案件はそのままで、別の案件（同一名称でバージョンが異なる）として投入できます。定義のバージョン番号を更新しなければ、変更後の内容が案件に反映（上書き）されません。

### ❗ 重要

定義のバージョンを更新しないで上書きする場合、運用中のほかの案件に影響を与えるおそれがあるので、注意が必要です。また、変更できる範囲に制限があるため、ビジネスプロセス自体を変更する場合は、定義のバージョンを更新して登録することを推奨します。

ここでは、ビジネスプロセスを変更するときに、定義のバージョン番号を更新する場合と更新しない場合の手順を説明します。

### 1.2.1 ビジネスプロセスの変更（定義のバージョンを更新する場合）

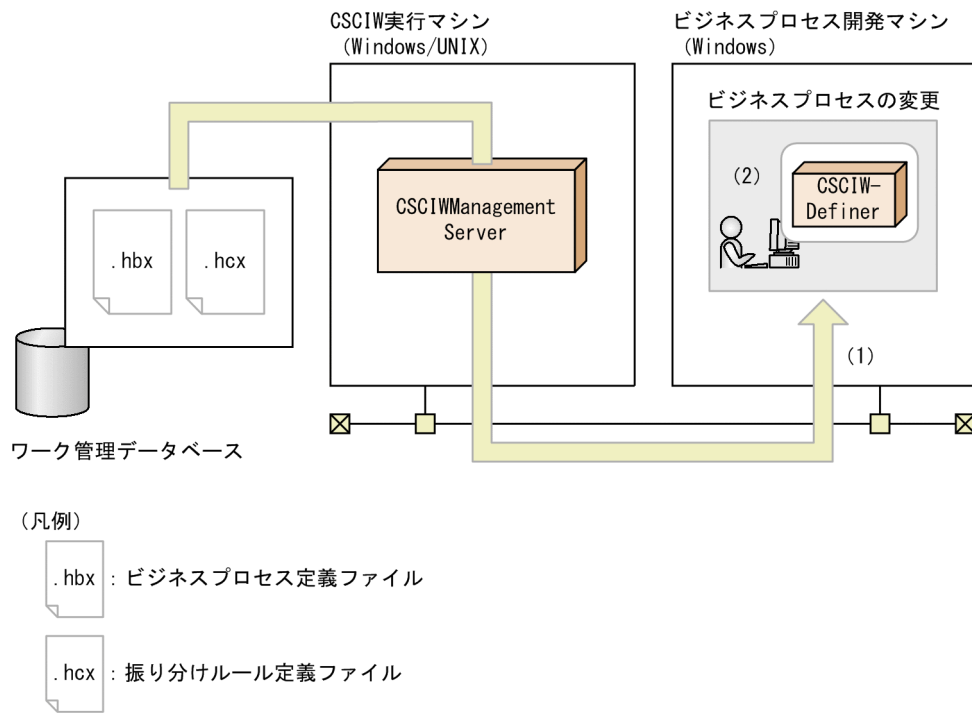
ビジネスプロセスの変更時に定義のバージョンを更新する場合の手順について説明します。

#### (1) 定義ファイルの変更

登録済みのビジネスプロセスの変更は、CSCIW-Definer、またはコマンドで実行できます。CSCIW-Definerを使用し、定義ファイルを変更するときの流れを図 1-3 に、コマンドを使用し、定義ファイルを変更するときの流れを図 1-4 に示します。

なお、図中の数字は、説明文の項番と対応しています。

図 1-3 定義ファイルの変更の流れ (CSCIW-Definer を使用する場合)



## 説明

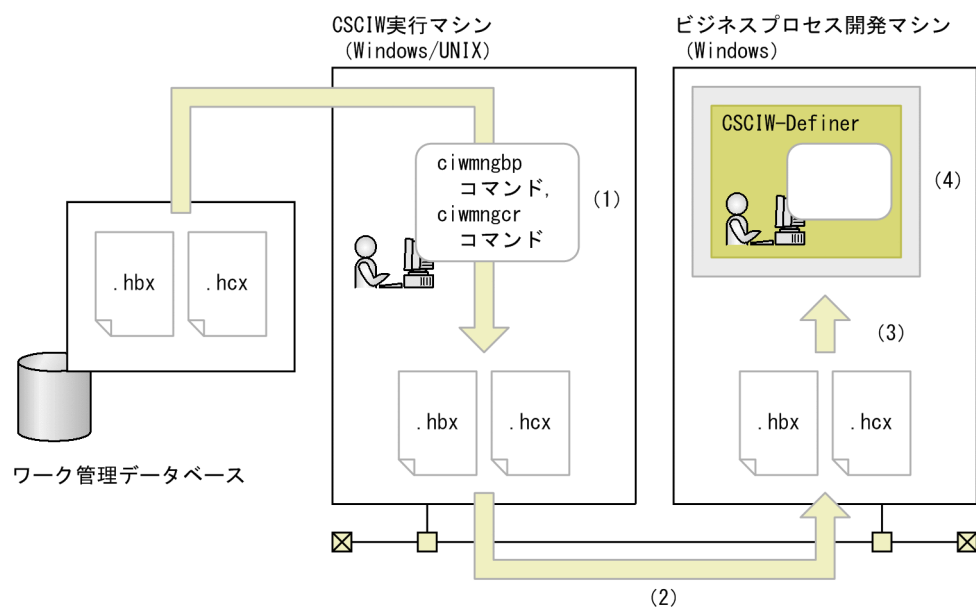
1. 登録済みのビジネスプロセス定義または振り分けルール定義を開きます。
2. 定義のバージョン番号を更新し、定義ファイルの内容を変更します。

バージョン番号は、更新前より大きな番号にしてください。



変更後のビジネスプロセスをチェックし、エラーがないことを確認します。確認したあと、CSCIW-Definer を使用してビジネスプロセスを登録します。

定義ファイルの登録の流れについては、「[1.1 CSCIW で使用するビジネスプロセス定義と振り分けルール定義の開発](#)」を参照してください。

図 1-4 定義ファイルの変更の流れ（コマンドを使用する場合）



(凡例)

-  .hbx : ビジネスプロセス定義ファイル
-  .hcx : 振り分けルール定義ファイル

## 説明

1. ciwmngbp（ビジネスプロセス定義の管理）コマンドまたはciwmngcr（振り分けルール定義の管理）コマンドを実行し、登録済みの定義ファイルを取得します。
2. 取得した定義ファイルをファイル転送プロトコル（FTP）などでビジネスプロセス定義開発マシンに転送します。
3. 転送した定義ファイルを CSCIW-Definer で開きます。
4. 定義のバージョン番号を更新し、定義ファイルの内容を変更します。

バージョン番号は、更新前より大きな番号にしてください。

変更後の定義ファイルの内容をチェックし、エラーがないことを確認します。確認したあと、コマンドを使用してビジネスプロセスを登録します。

定義ファイルの登録の流れについては、「[1.1 CSCIW で使用するビジネスプロセス定義と振り分けルール定義の開発](#)」を参照してください。

## 参考事項

CSCIW-Definer については、マニュアル「uCosminexus Service Coordinator Interactive Workflow ビジネスプロセス開発ガイド」を参照してください。

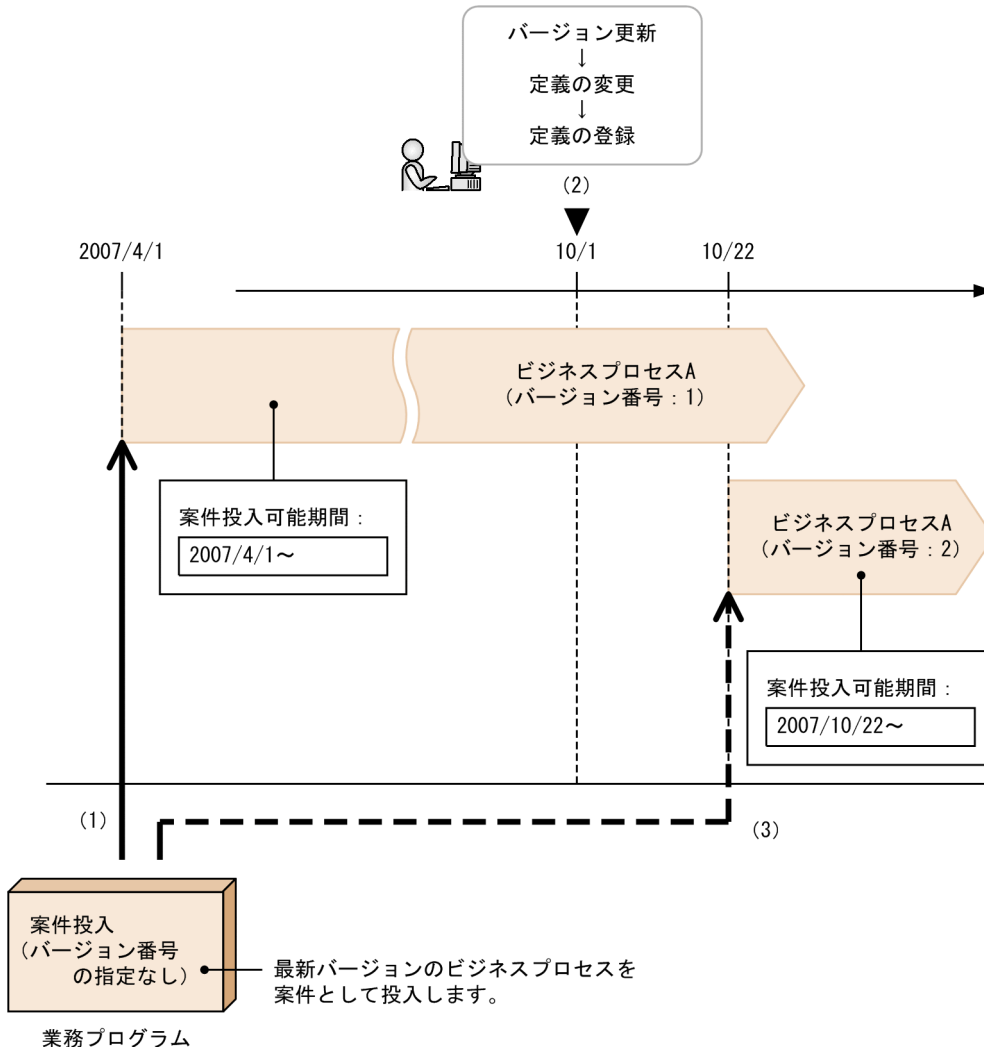
コマンドについては、マニュアル「uCosminexus Service Coordinator Interactive Workflow コマンド」を参照してください。

## (2) 運用例

定義のバージョンを更新する場合の、バージョンの更新と案件の投入のタイミングについて次に示す図を例に説明します。

なお、図中の数字は、説明文の項番と対応しています。

図 1-5 定義のバージョンを更新する場合の運用例



### 説明

1. 2007年4月1日に、業務プログラム（案件（バージョン番号の指定なし）の投入）が、ビジネスプロセスA（バージョン番号：1）を案件として投入します。
2. 2007年10月1日に、ビジネスプロセスA（バージョン番号：1）に対して定義を変更し、案件投入可能期間を2007年10月22日以降に設定して、ビジネスプロセスA（バージョン：2）として登録します。  
2007年10月21日以前にバージョン番号1として投入された案件は、そのままバージョン1で実行を続けます。

3. 2007年10月22日以降に、業務プログラム（案件（バージョン指定なし）の投入）が、変更後のビジネスプロセス A（バージョン番号：2）を案件として投入します。

ビジネスプロセス A（バージョン番号：1）の案件がすべて完了し、完了した案件をすべて削除すると、ビジネスプロセス A（バージョン番号：1）を削除できます。

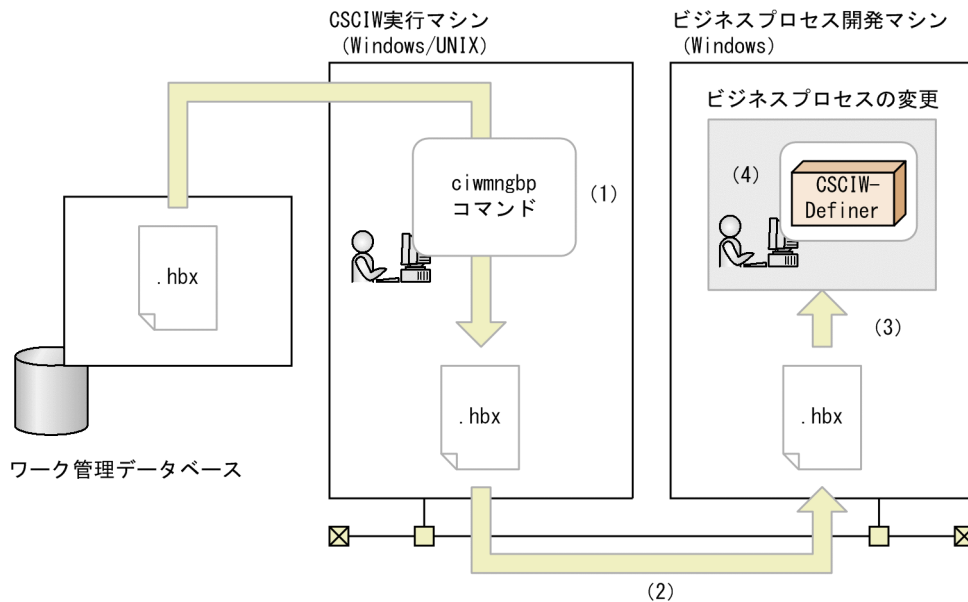
## 1.2.2 ビジネスプロセスの変更（バージョンを更新しない場合）

ビジネスプロセス定義のバージョン番号を更新しないで定義内容を変更するには、ciweditbp（ビジネスプロセス定義の変更）コマンドを使用します。

登録済みの定義ファイルの変更手順を次の図に示します。

なお、図中の数字は、説明文の項番と対応しています。

図 1-6 登録済みの定義ファイルの変更



(凡例)

.hbx : ビジネスプロセス定義ファイル

### 説明

1. ciwmngbp（ビジネスプロセス定義の管理）コマンドを実行し、変更したいビジネスプロセス定義を取得します。

ciwmngbp（ビジネスプロセス定義の管理）コマンドについては、マニュアル「uCosminexus Service Coordinator Interactive Workflow コマンド」を参照してください。

2. 取得したビジネスプロセス定義ファイルをファイル転送プロトコル（FTP）などでビジネスプロセス定義開発マシンに転送します。

3. 転送したビジネスプロセス定義ファイルを CSCIW-Definer で開きます。

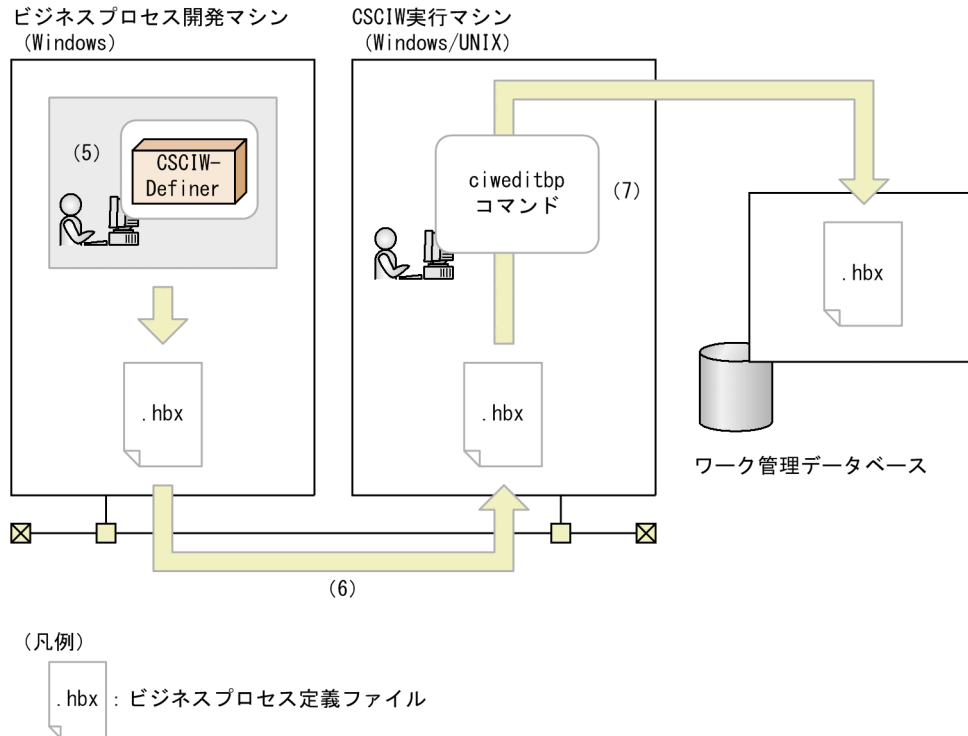
4. 開いた定義ファイルのバージョン番号はそのままにし、定義だけを変更します。

## メモ

CSCIW-Definer でビジネスプロセス定義を取得することもできます。

変更後のビジネスプロセス定義の登録手順を次の図に示します。

図 1-7 変更後のビジネスプロセス定義の登録



5. 変更後のビジネスプロセス定義をローカルファイルに保存します。

6. ローカルファイルに保存したビジネスプロセス定義ファイルを CSCIW 実行マシンに FTP など転送します。なお、FTP の転送モードはバイナリモードで転送する必要があります。また、CSCIW 実行マシンが複数ある場合は、任意の 1 台に転送してください。

7. 転送されたビジネスプロセス定義ファイルを指定し、ciweditbp (ビジネスプロセス定義の変更) コマンドを実行します。

コマンドを実行すると、転送されたビジネスプロセス定義ファイルの CSCIW に登録し、ワーク管理データベース上の指定されたビジネスプロセス定義の内容を変更します。

## 注意事項

ビジネスプロセス定義を変更したあとは、すべての J2EE サーバにある CSCIWManagementServer、および CSCIW を使用している Java アプリケーションを再起動してください。再起動することで、キャッシュしているビジネスプロセス定義が反映されます。詳細は、マニュアル「uCosminexus Service Coordinator Interactive Workflow システム構築・運用ガイド」の「ワーク管理システムの運用」を参照してください。

## 参考事項

CSCIW-Definer については、マニュアル「uCosminexus Service Coordinator Interactive Workflow ビジネスプロセス開発ガイド」を参照してください。

コマンドについては、マニュアル「uCosminexus Service Coordinator Interactive Workflow コマンド」を参照してください。

## 1.3 ビジネスプロセスを定義するときの注意事項

### 1.3.1 組み込み変数

#### (1) 指定できる組み込み変数

条件、振り分けルール、または生成ルールの SQL 文で指定できる組み込み変数を次の表に示します。なお、指定された組み込み変数は、評価時に対応する値に置き換わります。

表 1-1 指定できる組み込み変数

組み込み変数	説明	条件				振り分けルール	生成ルール
		業務ステップ	作業		制御ノード		
		完了条件	発生条件	完了条件	分岐条件		
@PIID	案件 ID	○	○	○	○	○	○
@PIName	案件名	○	○	○	○	○	○
@AIName	業務ステップ名	○	○	○	—	○	○
@WName	作業名	—	—	○	—	○	○
@PDefName	ビジネスプロセス定義名	○	○	○	○	○	○
@PDefVersion	ビジネスプロセス定義のバージョン	○	○	○	○	○	○
@ADefName	業務ステップ定義名	○	○	○	—	○	○
@WDefName	作業定義名	—	—	○	—	○	○
@CNDefName	制御ノード定義名	—	—	—	○	—	—
@PICreator	案件投入者	○	○	○	○	○	○
@PIStartDate	案件開始日時	○	○	○	○	○	○
@PIDecline	案件処理期限	○	○	○	○	○	○
@PIPRIORITY	案件優先度	○	○	○	○	○	○
@AISTartDate	業務ステップ開始日時	○	○	○	—	○	○
@AIDecline	業務ステップ処理期限	○	○	○	—	○	○
@AIPRIORITY	業務ステップ優先度	○	○	○	—	○	○
@WICreationDate	作業生成日時	—	—	○	—	○	○



組み込み変数	説明	条件				振り分け ルール	生成 ルール
		業務ス テップ	作業		制御ノー ド		
		完了条件	発生条件	完了条件	分岐条件		
@WIStartDate	作業開始日時	—	—	○	—	○	○
@WIDecline	作業処理期限	—	—	○	—	○	○
@WIPriority	作業優先度	—	—	○	—	○	○
@WIParticipant	作業者	—	—	○	—	○	○
@UserID	ユーザ識別子	○	○	○	○	○	○

(凡例)

- ：指定できます。
- ：該当しません。

## (2) 組み込み変数に関する注意事項

組み込み変数に関する注意事項を次に示します。

- 「@」で始まり、英数字以外の文字または文末までの英数字文字列は変数と見なされます。
- SQL 文中で「@」を使用する場合は、「¥@」と記述してください。

例

abc¥@hitachi.com

- SQL 文中で「¥」を使用する場合は、「¥¥」と記述してください。

例

D:¥¥work

- 組み込み変数の直後に英数字を使用する場合は、組み込み変数と文字列の間に「¥」を指定してください。

例：組み込み変数「@PIName」に文字列「XYZ」を連結する場合

@PIName¥XYZ

- 組み込み変数の直後に「@」で始まる文字列を使用する場合は、組み込み変数と文字列の間に「¥」を指定してください。

例：組み込み変数「@PIName」に文字列「@hitachi.com」を連結する場合

@PIName¥@hitachi.com

ただし、組み込み変数を続けて使用する場合は、「¥」の指定は不要です。

例：組み込み変数「@PIName」と「@PIID」を連結する場合

@PIName@PIID

- 組み込み変数の直後に記号を使用する場合は、そのまま指定できますが、「@」および「¥」を使用する場合は、組み込み変数直後の場合でも「¥」を指定してください。

例：組み込み変数「@PIName」に文字列「#abc」を連結する場合

```
@PIName#abc
```

例：文字列「C:¥work¥」に組み込み変数「@PIName」と文字列「¥data」を連結する場合

```
C:¥¥work¥¥@PIName¥¥data
```

- 名称系の組み込み変数（@PIName など）を使用した場合、案件名などの文字列にそのまま置き換えられます。そのため、案件名などに SQL 文の特殊文字として扱われる次の文字は含めないことを推奨します。
  - “%”（半角パーセント記号）
  - “\_”（半角アンダーバー）

SQL 文の LIKE 述語のパターン文字列中に組み込み変数を設定したときに、値となる文字列の中に “%” または “\_” が含まれていると、期待した検索結果が得られないおそれがあります。

- 対象となる属性の値が null の場合は、空文字列に置き換わります。  
数値として扱うとき、空文字列に置き換わると SQL 構文エラーとなるおそれがありますので、注意してください。
- 開始日時、生成日時、および処理期限は、1970/01/01 00:00:00 GMT を起点とした通算秒に置き換わります。
- 子作業生成ルールの作業生成日時、作業開始日時、作業処理期限、作業優先度、および作業者には、それぞれ生成ルールを定義した並列作業の作業生成日時、作業開始日時、作業処理期限、作業優先度、および作業者が渡されます。

## 1.3.2 巡回部分を含むビジネスプロセス定義

複数の業務ステップ間で案件遷移の整合性を保つために、巡回部分を含むビジネスプロセス定義を指定する場合は、注意が必要です。

正しい利用例と誤った利用例を次に示します。

### (1) 正しい利用例

次のような利用例であれば、業務ステップの実行中に巡回によって再度業務ステップが生成されるようなことはありません。

図 1-8 正しい利用例（巡回部分に制御ノードを含まない例）

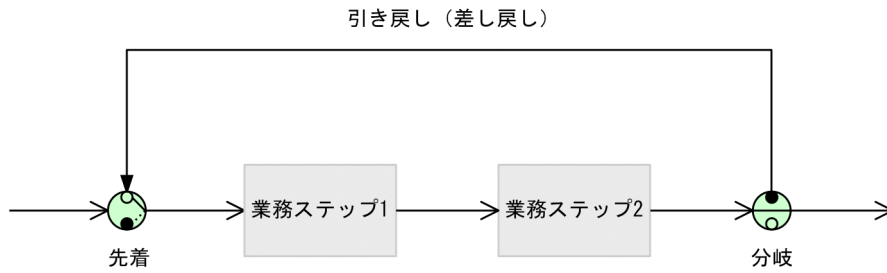
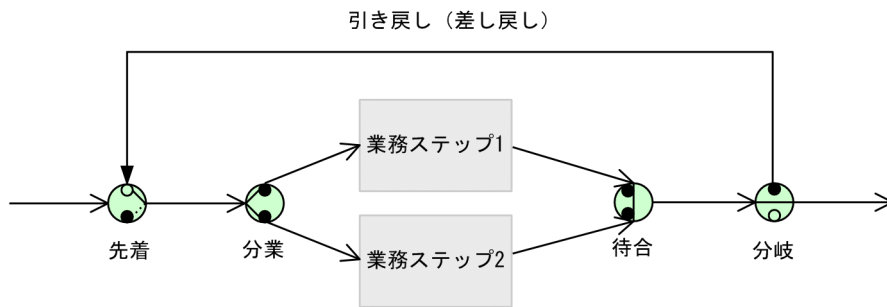


図 1-9 正しい利用例（巡回部分に一組の分業と待合だけを含む例）



## (2) 誤った利用例

次のような利用例は、業務ステップの実行中に巡回によって同一業務ステップが生成されるおそれがあります。

- 一組の分業と先着から成る巡回部分を含むビジネスプロセス定義の例  
一組の分業と先着から成る巡回部分を含むビジネスプロセス定義の誤った利用例を次に2つ示します。これらの場合は、分業ノードと先着2ノードの間の業務ステップが実行中にもかかわらず、再度同一業務ステップが生成されるおそれがあります。

図 1-10 誤った利用例（巡回部分に含まれる制御ノードに一組の分業と先着が含まれる例）  
（パターン1）

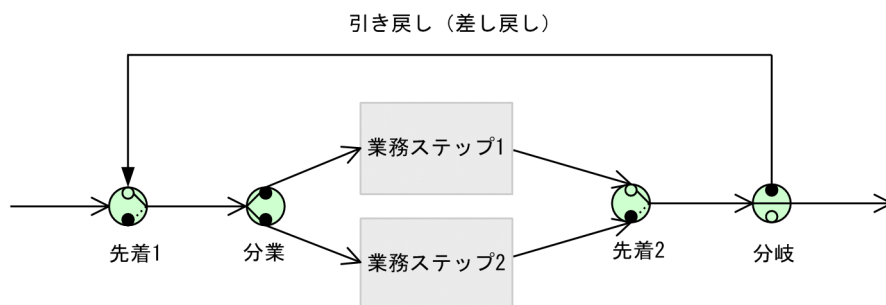
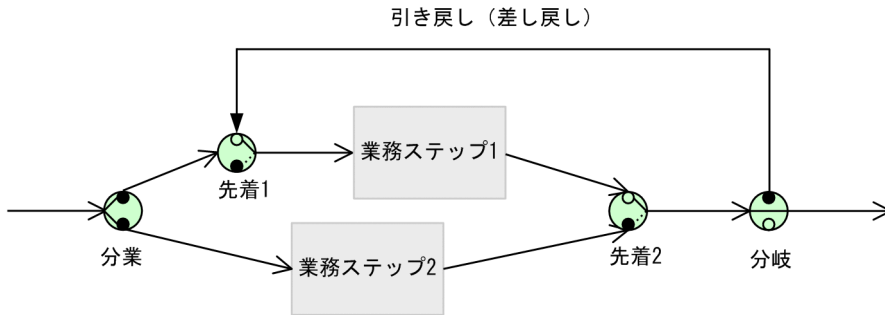
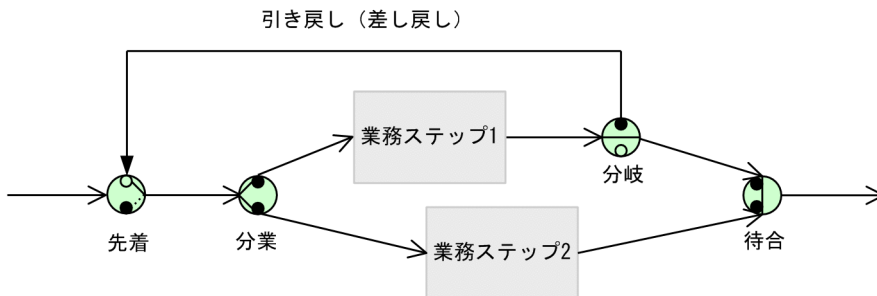


図 1-11 誤った利用例（巡回部分に含まれる制御ノードに一組の分業と先着が含まれる例）  
（パターン 2）



- 閉じていない分業ノードを含む巡回部分を含むビジネスプロセス定義の例  
閉じていない分業ノードを含む巡回部分を含むビジネスプロセス定義の誤った利用例を次に示します。  
この場合は、巡回するたびに同一の業務ステップ2が生成されます。

図 1-12 誤った利用例（巡回部分に閉じていない分業ノードが含まれている例）



### (3) 正しく動作するための条件

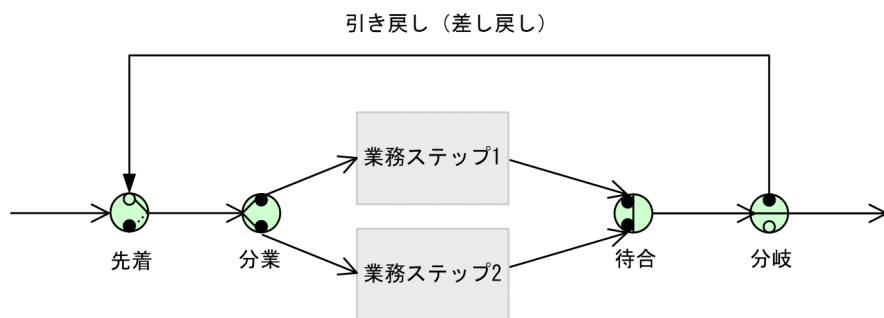
正しく動作するかどうかの判断の指針となるパターンについて、次に条件と図を示します。正しく動作させるためには、最低限次に示すどちらかのパターンの検査項目を満たしている必要があります。

#### パターン 1

次に示すすべての条件を満たしているパターンです。

- 巡回部分（巡回部分の開始点となる先着ノードと、巡回部分の終了点となる分岐ノードの対で挟まれる部分）が、先着ノードからの入力遷移と分岐ノードへの出力遷移だけを持っていること
- 巡回部分の終了点となる分岐ノードへの遷移時は、必ず、巡回部分内のすべての待合ノードが待合状態ではないこと
- 巡回部分の終了点となる分岐ノードへの遷移時は、必ず、巡回部分内のすべての業務ステップが実行中ではないこと
- 巡回部分内のすべての先着ノードには、後続停止指定がないこと
- 巡回部分の開始点となる先着ノードには、後続停止指定がないこと

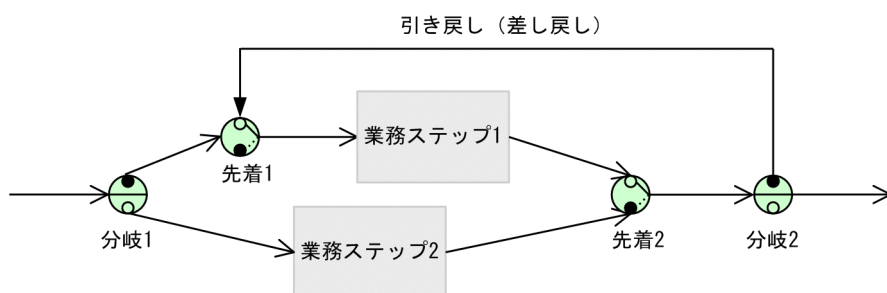
図 1-13 正しく動作するための条件を満たした例 (パターン 1)



## パターン 2

巡回部分に、業務ステップ、または後続停止指定がない先着ノードしか存在しないパターンです。

図 1-14 正しく動作するための条件を満たした例 (パターン 2)



## 1.3.3 SQL 文

### (1) 条件、生成ルールおよび振り分けルールでの SQL 文の評価規則

ビジネスプロセスに定義された SQL 文に従って、CSCIW がワーク管理データベースを検索するときの SQL 文の評価規則を説明します。ここで示す規則に従って、ビジネスプロセス定義、または振り分けルール定義の SQL 文を定義する必要があります。

表 1-2 SQL 文の評価規則

定義種別	評価対象	評価規則
ビジネスプロセス定義	条件	<p>真偽 (True または False) の結果を反対にする指定をしているかどうかで、次のようになります。</p> <p><b>指定をしていない場合</b></p> <ul style="list-style-type: none"> <li>問い合わせ結果が存在するときは、真 (True) とします。</li> <li>問い合わせ結果が存在しないときは、偽 (False) とします。</li> </ul> <p><b>指定をしている場合</b></p> <ul style="list-style-type: none"> <li>問い合わせ結果が存在するときは、偽 (False) とします。</li> <li>問い合わせ結果が存在しないときは、真 (True) とします。</li> </ul> <p>なお、COUNT(*)を指定すると常に真 (True) と見なされるため、指定しないでください。</p>

定義種別	評価対象	評価規則
	生成ルール	問い合わせ結果の先頭列のすべての行に格納された文字列を、子作業または子業務ステップの識別子とします。
振り分けルール定義	振り分けルール	問い合わせ結果の先頭行かつ先頭列に格納された文字列を作業者の ID とします。

## (2) SQL 文記述の注意

SQL 文を記述する際は、次の点に注意する。

- CSCIW-Definer では、SQL 文の文法チェックはしないため、CSCIW で使用するワーク管理データベースの DBMS 規則を意識して SQL 文を記述してください。
- 「@」で始まる半角英数字の文字列を、すべて組み込み変数と見なします。組み込み変数に関する注意事項については、「[1.3.1\(2\) 組み込み変数に関する注意事項](#)」を参照してください。

# 2

## 業務プログラムの作成方法

この章では、業務プログラムの処理の流れ、Java オブジェクトおよび業務プログラム作成時に必要な設定について説明します。また、CSCIW の API を利用するときの注意事項についても説明します。

## 2.1 API の特長

CSCIW で使用する API の特長について次に示します。

- **業務プログラムで、ワーク管理データベースと業務データベースの同期処理ができる (DB コネクション共有管理機能)**

CSCIW では、ワーク管理データベースと業務データベースのコネクションを共有できます。データベースのコネクションを共有することで、ワーク管理データベースに対する更新処理と業務データベースに対する更新処理を同一トランザクションで実行できます。

- **処理の対象や流れが理解しやすく、効率良く業務プログラムが作成できる**

Java 言語のクラスおよびインタフェースを、処理する対象ごとに案件、業務ステップ、作業などの単位で提供しています。処理の対象ごとにクラスおよびインタフェースを提供することで、処理の流れが理解しやすくなり、効率的に業務プログラムを作成できるようになります。

- **ポータビリティの高い業務プログラムが作成できる**

Java 言語および EJB の仕様に基づいたプログラミングをすることで、プログラミングコードを変更しなくても、異なるプラットフォームで動作する業務プログラムを作成できます。

### ヒント

BPMN 連携機能を使用する場合、業務アプリケーションの開発には、REST API または BPMN 連携ライブラリの Java API を使用します。Java API で開発する場合、このマニュアルで説明する更新系 API は使用しません (ただし、参照系 API は、このマニュアルで説明している API を使用します)。

REST API および BPMN 連携ライブラリの Java API については、マニュアル「uCosminexus Service Coordinator Interactive Workflow BPMN 連携機能 使用の手引」を参照してください。



## 2.2 Java 業務プログラムの作成

Java 業務プログラムの作成手順について説明します。

### 2.2.1 パッケージのインポート

CSCIW が提供するすべての Java API は、パッケージ「jp.co.Hitachi.soft.csciw」に所属します。CSCIW が提供する Java API を使用するには、次に示すパッケージのインポート文を、ソースコードに記述する必要があります。

```
import jp.co.Hitachi.soft.csciw.*;
```

また、Java オブジェクト呼び出し機能は、パッケージ「jp.co.Hitachi.soft.csciw.callback」に所属します。Java オブジェクト呼び出し機能を使用する場合は、次に示すパッケージのインポート文を、ソースコードに記述する必要があります。

```
import jp.co.Hitachi.soft.csciw.callback.*;
```

### 2.2.2 業務プログラムのコンパイル

#### (1) Java コンパイラのオプション

CSCIW では、JDK8.0 の機能を使用しています。このため、業務プログラムは、JDK8.0 以降の機能を利用したソースとしてコンパイルする必要があります。javac コマンドの-source オプションには 1.8 以降または 8 以降を指定してください。

#### (2) クラスパス

業務プログラムをコンパイルする場合は、次に示す JAR ファイルをクラスパスに設定します。

- CSCIW をインストールした環境の場合

< Windows のとき >

```
<CSCIWインストールディレクトリ>%Lib%csciw.jar  
<CSCIWインストールディレクトリ>%Lib%csciwcmn.jar
```

< UNIX のとき >

```
/opt/hitachi/CSCIW/lib/csciw.jar  
/opt/hitachi/CSCIW/lib/csciwcmn.jar
```

- uCosminexus Business Process Developer をインストールした環境の場合

```
<uCosminexus Business Process Developerインストールディレクトリ>%Lib%csciw.jar  
<uCosminexus Business Process Developerインストールディレクトリ>%Lib%csciwcmn.jar
```

## 注意事項

- 各アプリケーションの ear に csciw.jar ファイルおよび csciwcmn.jar ファイルを組み込まないでください。
- Java アプリケーションを実行する際に必要なクラスパスは、マニュアル「uCosminexus Service Coordinator Interactive Workflow システム構築・運用ガイド」の「Java アプリケーションの設定」を参照してください。

## 2.3 業務プログラムの処理の流れ

---

業務プログラムの処理の流れについて説明します。

### 2.3.1 CSCIW の初期化

プロセス開始時に、一度だけ CSCIW を初期化します。

CSCIW の初期化方法は、CSCIW を Java アプリケーションで使用する場合と、J2EE サーバで使用する場合とで異なります。

#### (1) Java アプリケーションで使用する場合

CIWAdmin クラスの initializeCIWFactory メソッドを使用して、CSCIW を初期化します。CSCIW を初期化する場合の業務プログラムのコーディング例を次に示します。なお、例中の「factory」は CIWFactory オブジェクトを、「ds」は javax.sql.DataSource オブジェクトを示します。

例

```
try {
    // データソース取得処理
    ...
    CIWAdmin.initializeCIWFactory(ds);
}
catch(CIWFatalException e) {
    // 例外処理を記述
}
```

#### (2) J2EE サーバで使用する場合

J2EE サーバ上に CSCIWManagementServer を配備し、開始すれば、CSCIW が初期化されます。

### 2.3.2 CIWFactory オブジェクトの取得

CIWFactory オブジェクトは、CIWFactory クラスの getInstance メソッドを使用して取得します。CIWFactory オブジェクトを取得する場合の業務プログラムのコーディング例を次に示します。なお、例中の「factory」は、CIWFactory オブジェクトを示します。

例

```
try {
    CIWFactory factory = CIWFactory.getInstance();
    ...
}
catch(CIWFatalException e) {
```

```
} // 例外処理を記述
```

### 2.3.3 CIWServer オブジェクトの生成

CIWServer オブジェクトは、CIWFactory クラスの createCIWServer メソッドを使用して生成します。

CIWServer オブジェクトの生成方法は、CIWServer オブジェクトをコネクションと関連づけるか、またはデータソースと関連づけるかどうかによって異なります。

#### (1) CIWConnectionCoordinator を使用してコネクションと関連づける場合

CIWServer オブジェクトは、ユーザ記述子だけを指定した createCIWServer メソッドを使用して生成します。CIWServer オブジェクトを生成する場合の業務プログラムのコーディング例を次に示します。なお、例中の「factory」は、CIWFactory オブジェクトを示します。

例

```
try {
    CIWServer server = factory.createCIWServer("User");
    ...
}
catch(CIWFatalException e) {
    // 例外処理を記述
}
```

#### (2) データソースと関連づける場合

ユーザ記述子とデータソースを指定した createCIWServer メソッドを使用して、CIWServer オブジェクトを生成します。CIWServer オブジェクトを生成する場合の業務プログラムのコーディング例を次に示します。なお、例中の「factory」は CIWFactory オブジェクトを、「ds」は javax.sql.DataSource オブジェクトを示します。

例

```
try {
    // データソース取得処理
    ...
    CIWServer server = factory.createCIWServer(
        "User",
        ds);
    ...
}
catch(CIWFatalException e) {
    // 例外処理を記述
}
```

## 2.3.4 CIWServer オブジェクトに対する Connection オブジェクトの関連づけ

CIWConnectionCoordinator クラスの attachDatabaseConnection メソッドを使用して、CIWServer オブジェクトに Connection オブジェクトを関連づけます。CIWServer オブジェクトに Connection オブジェクトを関連づける場合の業務プログラムのコーディング例を次に示します。なお、例中の「server」は CIWServer オブジェクトを、「connection」は java.aql.Connection オブジェクトを示します。

例

```
try {
    // コネクション取得および自動コミットのOFF
    ...
    CIWConnectionCoordinator coordinator = server.getConnectionCoordinator();
    coordinator.attachDatabaseConnection(connection);
}
catch(CIWFatalException e) {
    // 例外処理を記述
}
...
```

## 2.3.5 CSCIW を利用した業務処理の実装

CSCIW を利用した業務処理の実装例を示します。

### (1) 案件の投入

案件を投入する場合の業務プログラムのコーディング例を次に示します。なお、例中の「server」は、CIWServer オブジェクトを示します。

例

```
try {
    // 案件名「案件1」を指定して、ビジネスプロセス定義
    // 「ビジネスプロセス定義1」の案件を投入する。
    Map<CIWProcessInstance.AttributeName, Object> attributes
        = new HashMap<CIWProcessInstance.AttributeName, Object>();
    attributes.put(
        CIWProcessInstance.AttributeName.NAME,
        "案件1");

    CIWProcessInstance processInstance = server
        .createAndStartProcessInstance(
            attributes,
            "ビジネスプロセス定義1",
            null);
    ...
}
catch(CIWFatalException e) {
    // 例外処理を記述
}
```

```
}  
...
```

## (2) 作業一覧の取得

作業一覧を取得する場合の業務プログラムのコーディング例を次に示します。なお、例中の「server」は、CIWServer オブジェクトを示します。

例

```
try {  
    // 作業者IDが「User001」かつ「実行開始可能」状態の作業を  
    // 生成日時の降順に取得する。  
    EnumSet<CIWorkItem.AttributeName> attributeNames  
        = EnumSet.allOf(CIWorkItem.AttributeName.class);  
    List<CIWorkItem> wiList = server.getWorkItemsList(  
        "Participant = 'User001' AND StateCode = '"  
        + CIWorkItem.State.READY.toStateCode() + "'",  
        "CreationDate DESC",  
        0,  
        -1,  
        attributeNames);  
    ...  
}  
catch(CIWFatalException e) {  
    // 例外処理を記述  
}  
...
```

## (3) 作業の開始と終了

作業を開始および終了する場合の業務プログラムのコーディング例を次に示します。なお、例中の「wi」は、CIWorkItem オブジェクトを示します。

例

```
try {  
    // 作業を「実行開始可能」状態から「作業者実行」状態にして「実行済み」  
    // 状態にする。  
    // 既に「作業者実行」状態の作業の場合は、completeメソッドを使用する。  
    wi.performAndComplete();  
    ...  
}  
catch(CIWTransientException e) {  
    // 例外処理を記述  
}  
...
```

## (4) 条件に一致する作業の作業者割り当てと着手をする場合の実装例

条件に一致する作業の作業者割り当てと着手をする場合の実装について説明します。

CIWServer クラスの allocateWorkItemEx メソッドと CIWorkItem クラスの free メソッドを利用します。クラス、およびメソッドの詳細については、次の個所を参照してください。

- [5.7 CIWServer (ワーク管理システム全般にかかわる処理を行うための機能を提供するインタフェース)]
- [5.9 CIWorkItem (作業のインタフェース)]

次の条件を満たす実装例を示します。

- 作業者が「SharedWork」である
- 優先度が「1 以上」である
- 「作業実行可能」状態の作業に、作業者を割り当てて着手する
- 業務処理に成功した場合は、「実行済み」状態に遷移する
- 業務処理に失敗した場合は、作業を返却して振り分けルールを再評価する

例

```
// 条件に一致する作業に作業者を割り当てて着手する（作業者は”TargetWorker”，状態は”作業実行”になる）
// フィルター条件「作業者が”SharedWork”，優先度が1以上」
// ソート条件「作業作成日が小さい(古い日時)順」
String newParticipant = "TargetWorker";
String filter = " Participant = 'SharedWork' AND Priority >= 1 ";
String sort = " CreationDate ASC ";
EnumSet<CIWorkItem.AttributeName> attributeNames = EnumSet.allOf(CIWorkItem.AttributeName.class);

CIWorkItem wi = ciwServer.allocateWorkItemEx(
    newParticipant,
    filter,
    sort,
    attributeNames,
    20);

// トランザクションをコミットする
connection.commit();

/** 何らかの業務処理をする **/

// 処理の結果 (boolean result) で処理を分岐
if(result) {
    // 処理が成功したので、作業を実行済にする
    wi.changeState(CIWorkItem.State.EXECUTED);
} else {
    // 処理が失敗したので、作業を返却して振り分けルールを再評価する
    wi.free();
}
connection.commit();
```

## 2.3.6 CIWServer オブジェクトに対する Connection オブジェクトの関連づけの解除

CIWConnectionCoordinator クラスの detachDatabaseConnection メソッドを使用して、CIWServer オブジェクトに対する Connection オブジェクトの関連づけを解除します。CIWServer オブジェクトと Connection オブジェクトの関連づけを解除する場合の業務プログラムのコーディング例を次に示します。なお、例中の「server」は、CIWServer オブジェクトを示します。

例

```
try {
    ...
    CIWConnectionCoordinator coordinator = server.getConnectionCoordinator();
    coordinator.detachDatabaseConnection();

    // トランザクションの終了処理
    ...
}
catch(CIWFatalException e) {
    // 例外処理を記述
}
```

## 2.3.7 CSCIW の終了

プロセス終了前に、一度だけ CSCIW の終了処理をします。

CSCIW の終了処理の方法は、Java アプリケーションで使用する場合と、J2EE サーバで使用する場合とで異なります。

### (1) Java アプリケーションで使用する場合

CIWAdmin クラスの finalizeCIWFactory メソッドを使用して、CSCIW を終了します。CSCIW を終了する場合の業務プログラムのコーディング例を次に示します。

例

```
...
CIWAdmin.finalizeCIWFactory();
```

### (2) J2EE サーバで使用する場合

J2EE サーバ上に配備した CSCIWManagementServer を停止すれば、CSCIW は終了します。



## 2.4 条件およびルールの評価

---

条件定義、生成ルール定義、または振り分けルール定義に指定できる種別を次の表に示します。

表 2-1 条件定義、生成ルール定義、または振り分けルール定義に指定できる種別

種別	表示名	説明
RDB 検索	RDB 検索	SQL 文を実行し、RDB 検索を実行します。
Java オブジェクト呼び出し	Java	Java オブジェクトを呼び出し、ユーザが定義した Java オブジェクトに従って評価します。

SQL 文で指定できる組み込み変数については、「[1.3.1\(1\) 指定できる組み込み変数](#)」を参照してください。また、SQL 文の評価規則については、「[1.3.3\(1\) 条件、生成ルールおよび振り分けルールでの SQL 文の評価規則](#)」を参照してください。

Java オブジェクトの作成については「[2.6 Java オブジェクトの作成](#)」を参照してください。また、Java オブジェクトに渡される情報については、「[2.6.4 業務プログラムに渡される情報](#)」を参照してください。

## 2.5 Java オブジェクト呼び出し利用時の設定

条件、振り分けルール、生成ルール、および作業アプリケーションで、CSCIW が提供するインタフェースを実装する Java オブジェクトを呼び出すことができます。

### ● ヒント

このマニュアルでは、条件、振り分けルール、生成ルール、および作業アプリケーションの Java オブジェクト呼び出し機能について説明します。

アプリケーション呼び出しサービスの Java オブジェクト呼び出し機能については、マニュアル「uCosminexus Service Coordinator Interactive Workflow BPMN 連携機能 使用の手引」を参照してください。

Java オブジェクトを呼び出す設定手順を次に示します。

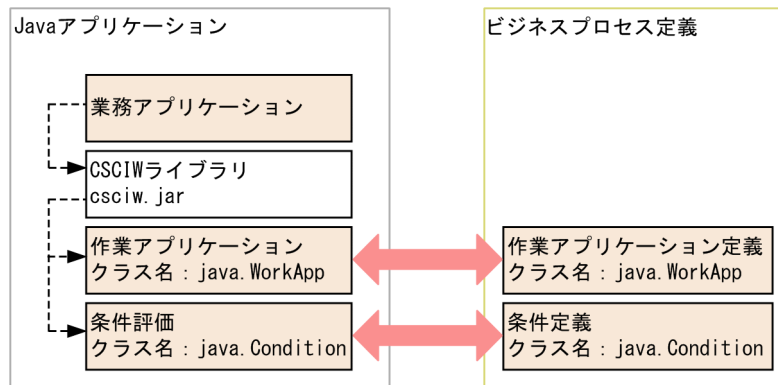
1. ビジネスプロセス定義の作業アプリケーション定義、条件またはルールの定義に、Java オブジェクトの Java クラス名をパッケージを含む形で記載する。
2. CSCIW が提供する jar ファイルおよび作成した Java オブジェクトを設定する。

Java オブジェクトの設定方法は条件によって異なります。

#### ●Java アプリケーションの場合

業務アプリケーション実行時のクラスパスに csciw.jar および Java オブジェクトを設定します。Java アプリケーションの場合の構成を次の図に示します。

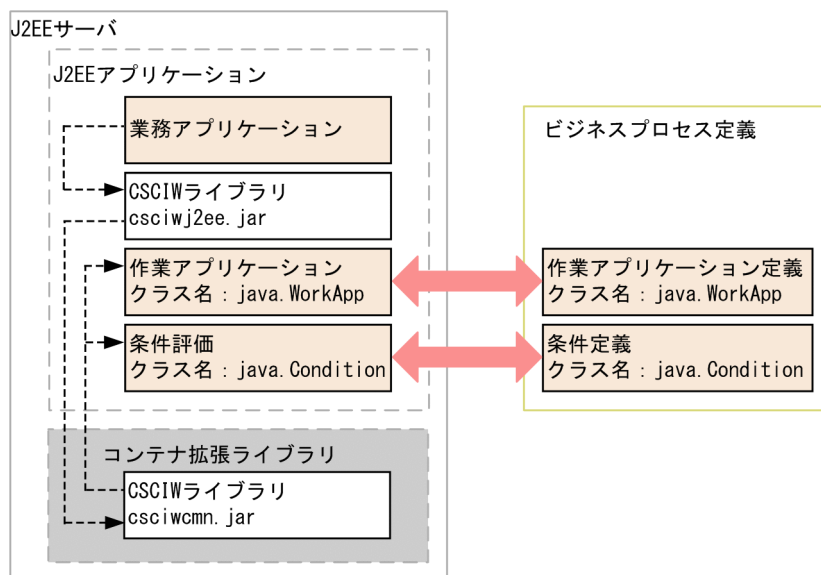
図 2-1 Java アプリケーションの場合の構成



#### ●Java オブジェクトを J2EE アプリケーションに組み込む場合

J2EE サーバのコンテナ拡張ライブラリに csciwcmn.jar を設定し、業務アプリケーション (J2EE アプリケーション) に csciwj2ee.jar および Java オブジェクトをインポートします。Java オブジェクトを J2EE アプリケーションに組み込む場合の構成を次に示します。

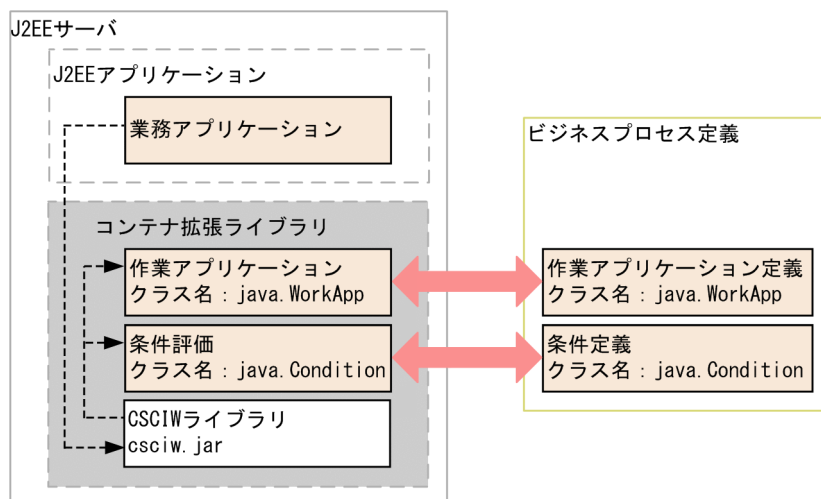
図 2-2 Java オブジェクトを J2EE アプリケーションに組み込む場合の構成



●Java オブジェクトをコンテナ拡張ライブラリに組み込む場合

J2EE サーバのコンテナ拡張ライブラリに csciw.jar および Java オブジェクトを設定します。Java オブジェクトをコンテナ拡張ライブラリに組み込む場合の構成を次の図に示します。

図 2-3 Java オブジェクトをコンテナ拡張ライブラリに組み込む場合の構成



3. CSCIW は、ビジネスプロセス定義に記載された Java クラス名で、Java オブジェクト（作業アプリケーション、条件／ルール）を呼び出す。

4. CSCIW は、3.で呼び出した Java オブジェクトの戻り値で、作業状態の変更、条件判定、子作業生成および作業者を決定する。

## 2.6 Java オブジェクトの作成

Java オブジェクト呼び出しを利用するには、CSCIW が提供するインタフェースを実装する必要があります。CSCIW が提供するインタフェースの詳細については、「[9. Java オブジェクト呼び出し用のインタフェース](#)」を参照してください。この節では、Java オブジェクトの作成方法について説明します。

### 2.6.1 作成時の注意事項

Java オブジェクト作成時の注意事項を次に示します。

- CSCIW が提供するインタフェースを必ず実装してください。  
CSCIW が提供するインタフェースの詳細については、「[9. Java オブジェクト呼び出し用のインタフェース](#)」を参照してください。
- CSCIW が提供するインタフェースを実装したクラスでは、引数のあるコンストラクタや public 以外のコンストラクタを実装しないでください。
- Java オブジェクトでは、CSCIW が提供する API を発行しないでください。
- データベースへアクセスする場合は、CSCIW が Java オブジェクトに渡す `java.sql.Connection` を利用してください。
- Java オブジェクトの処理で、CSCIW が渡す `java.sql.Connection` のコネクションを切断したり、トランザクションを終了したりしないでください。
- Java オブジェクト呼び出しでは、複数スレッドから同時に実行される場合があるため、クラス変数（スタティック変数）を使用しないでください。
- Java オブジェクトのパッケージは、「`jp.co.Hitachi.soft.csciw`」以外にしてください。
- Java オブジェクトはリクエスト（作業アプリケーションやルール評価のタイミング）ごとにインスタンス化されます。そのため、インスタンス変数（メンバ変数）とメソッドローカル変数だけを利用したコーディングを推奨します。

### 2.6.2 Java オブジェクトの実装例（条件、生成ルール、および振り分けルール）

条件定義、生成ルール定義、および振り分けルール定義の実装について説明します。条件またはルールの評価に渡される情報については、「[2.6.4\(1\) Java オブジェクトに渡される情報（条件、生成ルールおよび振り分けルールの場合）](#)」を参照してください。

#### (1) 条件評価

条件評価の実装例を示します。

条件 Java オブジェクトのインタフェースについては、「[9.2 CIWConditionEvaluator \(条件 Java オブジェクトのインタフェース\)](#)」を参照してください。

例

```
public class UserCondition implements CIWConditionEvaluator {

    /**
     * 条件評価メソッドの実装
     */
    public boolean evaluateCondition(
        Map<Parameter, String> aParameters,
        Connection aDBConnection)
        throws CIWUserException {

        // -----
        // (1) 情報の取得
        // -----

        String bpDefName = null; // ビジネスプロセス定義名
        String piName = null; // 案件名
        String piID = null; // 案件ID
        String exParameter = null; // 補助定数パラメタ
        String userId = null; // ユーザID

        bpDefName = aParameters.get(Parameter.CSCIW_PDEFNAME);
        piName = aParameters.get(Parameter.CSCIW_PINAME);
        piID = aParameters.get(Parameter.CSCIW_PIID);
        exParameter = aParameters.get(Parameter.CSCIW_EXPARAMETER);
        userId = aParameters.get(Parameter.CSCIW_USERID);

        // -----
        // (2) 条件評価処理
        // -----

        boolean result = false;
        // データベースにアクセスする場合は、コネクション (aDBConnection) を利用する。
        // ただし、コネクションを切断したりトランザクションを終了したりしないこと。
        result = true; // 評価結果を格納

        // -----
        // (3) 評価結果を戻り値に設定
        // -----

        return result;
    }
}
```

## (2) 生成ルール評価

生成ルール評価の実装例を示します。

生成ルール Java オブジェクトのインタフェースについては、「[9.3 CIWCreationRuleEvaluator \(生成ルール Java オブジェクトのインタフェース\)](#)」を参照してください。

```

public class UserCreationRule implements CIWCreationRuleEvaluator {

    /**
     * 子作業生成ルール評価メソッドの実装
     */
    public Set<String> evaluateCreationRule(
        Map<Parameter, String> aParameters,
        Connection aDBConnection)
        throws CIWUserException {

        // -----
        // (1) 情報の取得
        // -----

        String bpDefName = null; // ビジネスプロセス定義名
        String piName = null; // 案件名
        String piID = null; // 案件ID
        String exParameter = null; // 補助定数パラメタ
        String userId = null; // ユーザID

        bpDefName = aParameters.get(Parameter.CSCIW_PDEFNAME);
        piName = aParameters.get(Parameter.CSCIW_PINAME);
        piID = aParameters.get(Parameter.CSCIW_PIID);
        exParameter = aParameters.get(Parameter.CSCIW_EXPARAMETER);
        userId = aParameters.get(Parameter.CSCIW_USERID);

        // -----
        // (2) 子作業名の決定処理
        // -----

        Set<String> childWorks = new HashSet<String>();
        // データベースにアクセスする場合は、コネクション (aDBConnection) を利用する。
        // ただし、コネクションを切断したりトランザクションを終了したりしないこと。
        childWorks.add("SubWork01"); // 子作業名の文字列を格納
        childWorks.add("SubWork02"); // 子作業名の文字列を格納

        // -----
        // (3) 作業名を戻り値に設定
        // -----

        return childWorks;
    }
}

```

### (3) 振り分けルール評価

振り分けルール評価の実装例を次に示します。

振り分けルール Java オブジェクトのインタフェースについては、「[9.4 CIWResourceSelector \(振り分けルール Java オブジェクトのインタフェース\)](#)」を参照してください。

```

public class UserResourceSelector implements CIWResourceSelector {

    /**
     * 振り分けルール評価メソッドの実装
     */
    public String getAvailableParticipant(
        Map<Parameter, String> aParameters,
        Connection aDBConnection)
        throws CIWUserException {

        // -----
        // (1) 情報の取得
        // -----

        String bpDefName = null; // ビジネスプロセス定義名
        String piName = null; // 案件名
        String piID = null; // 案件ID
        String exParameter = null; // 補助定数パラメタ
        String userId = null; // ユーザID

        bpDefName = aParameters.get(Parameter.CSCIW_PDEFNAME);
        piName = aParameters.get(Parameter.CSCIW_PINAME);
        piID = aParameters.get(Parameter.CSCIW_PIID);
        exParameter = aParameters.get(Parameter.CSCIW_EXPARAMETER);
        userId = aParameters.get(Parameter.CSCIW_USERID);

        // -----
        // (2) 作業者の決定処理
        // -----

        String workerName = null;
        // データベースにアクセスする場合は、コネクション (aDBConnection) を利用する。
        // ただし、コネクションを切断したりトランザクションを終了したりしないこと。
        workerName = "User01"; // 作業者のIDを文字列に格納する。

        // -----
        // (3) 作業者を戻り値に設定
        // -----

        return workerName;
    }
}

```

### 2.6.3 Java オブジェクトの実装例 (作業アプリケーション)

作業アプリケーションの実装について説明します。Java 呼び出し時に渡される情報については、「2.6.4(2) Java オブジェクトに渡される情報 (作業アプリケーションの場合)」を参照してください。

作業アプリケーションの実装例を次に示します。

```

public class UserWorkApplication implements CIWorkApplication {

    /**
     * 作業アプリケーションの業務処理の実装
     */
    public Result startSynchronously(
        Map<Parameter, String> aParameters,
        Connection aDBConnection)
        throws CIWUserException {

        // -----
        // (1) 情報の取得
        // -----

        String piName = null; // 案件名
        String piID = null; // 案件ID
        String wiID = null; // 作業ID
        String sid = null; // システムID
        String exParameter = null; // 補助定数パラメタ
        String userId = null; // ユーザID

        piName = aParameters.get(Parameter.CSCIW_PINAME);
        piID = aParameters.get(Parameter.CSCIW_PIID);
        wiID = aParameters.get(Parameter.CSCIW_WIID);
        sid = aParameters.get(Parameter.CSCIW_SYSTEMID);
        exParameter = aParameters.get(Parameter.CSCIW_EXPARAMETER);
        userId = aParameters.get(Parameter.CSCIW_USERID);

        // -----
        // (2) 業務処理
        // -----

        Result appResult = null;
        // データベースにアクセスする場合は、コネクション (aDBConnection) を利用する。
        // ただし、コネクションを切断したりトランザクションを終了したりしないこと。
        appResult = Result.PERFORM_AND_COMPLETE; // 作業アプリケーションの完了コードを設定する。

        // -----
        // (3) 作業アプリケーションの業務処理結果を戻り値に設定
        // -----

        return appResult;
    }
}

```

## 2.6.4 業務プログラムに渡される情報

この項では、業務プログラムに渡される情報について説明します。



## (1) Java オブジェクトに渡される情報（条件，生成ルールおよび振り分けルールの場合）

CSCIW から条件，生成ルール，振り分けルールを呼び出す際に，条件，生成ルール，振り分けルールの Java オブジェクトに渡される情報を次の表に示します。

表 2-2 条件またはルールの評価に渡される情報

パラメタ	説明	条件				振り分けルール	生成ルール
		業務ステップ	作業		制御ノード		子作業
		完了条件	発生条件	完了条件	分岐条件		
CSCIWPIID	案件 ID	○	○	○	○	○	○
CSCIWPIName	案件名	○	○	○	○	○	○
CSCIWAIName	業務ステップ名	○	○	○	—	○	○
CSCIWWIName	作業名	—	—	○	—	○	○
CSCIWPDefName	ビジネスプロセス定義名	○	○	○	○	○	○
CSCIWPDefVersion	ビジネスプロセス定義のバージョン	○	○	○	○	○	○
CSCIWADefName	業務ステップ定義名	○	○	○	—	○	○
CSCIWWDefName	作業定義名	—	—	○	—	○	○
CSCIWCNDefName	制御ノード定義名	—	—	—	○	—	—
CSCIWExParameter	補助定数パラメタ	○	○	○	○	○	○
CSCIWPICreator	案件投入者	○	○	○	○	○	○
CSCIWPIStartDate	案件開始日時	○	○	○	○	○	○
CSCIWPIDecline	案件処理期限	○	○	○	○	○	○
CSCIWPIPriority	案件優先度	○	○	○	○	○	○
CSCIWAISStartDate	業務ステップ開始日時	○	○	○	—	○	○
CSCIWAIDecline	業務ステップ処理期限	○	○	○	—	○	○
CSCIWAIPriority	業務ステップ優先度	○	○	○	—	○	○
CSCIWWICreationDate	作業生成日時	—	—	○	—	○	○
CSCIWWISStartDate	作業開始日時	—	—	○	—	○	○

パラメタ	説明	条件				振り分け ルール	生成 ルール
		業務ス テツプ	作業		制御 ノード		子作業
			完了 条件	発生 条件			
CSCIWWIDealine	作業処理期限	—	—	○	—	○	○
CSCIWWIPriority	作業優先度	—	—	○	—	○	○
CSCIWWIParticipant	作業者	—	—	○	—	○	○
CSCIWUserID	ユーザ識別子	○	○	○	○	○	○

(凡例)

○：Java オブジェクトに渡されます。

—：Java オブジェクトに渡されません。

## (2) Java オブジェクトに渡される情報（作業アプリケーションの場合）

CSCIW から作業アプリケーションを呼び出す際に、作業アプリケーションに渡される情報を次の表に示します。

表 2-3 作業アプリケーションに渡される情報

パラメタ	説明
CSCIWPIID	案件 ID
CSCIWWIID	作業 ID
CSCIWPIName	案件名
CSCIWAIName	業務ステップ名
CSCIWWIName	作業名
CSCIWSystemID	システム ID
CSCIWPDefName	ビジネスプロセス定義名
CSCIWPDefVersion	ビジネスプロセス定義バージョン
CSCIWADefName	業務ステップ定義名
CSCIWWDefName	作業定義名
CSCIWExParameter	補助定数パラメタ
CSCIWPICreator	案件投入者
CSCIWPIStartDate	案件開始日時
CSCIWPIDeadline	案件処理期限

パラメタ	説明
CSCIWPIPriority	案件優先度
CSCIWAIStartDate	業務ステップ開始日時
CSCIWAIDeadline	業務ステップ処理期限
CSCIWAIPriority	業務ステップ優先度
CSCIWWICreationDate	作業生成日時
CSCIWWIStartDate	作業開始日時
CSCIWWIDeadline	作業処理期限
CSCIWWIPriority	作業優先度
CSCIWWIParticipant	作業者
CSCIWUserID	ユーザ識別子

### (3) 注意事項

Java オブジェクトに渡される情報およびパラメタに関する注意事項を次に示します。

- パラメタ名は列挙型 Parameter で提供されます。
- CSCIW から Java オブジェクトに渡されるパラメタは、メソッド引数の Map aParameter 経由で渡されます。Map のキーは列挙型 Parameter になります。  
案件 ID を取得する場合は、aParameter.get(Parameter.CSCIW\_PIID)で取得します。
- 値はすべて文字列で渡されます。
- 値が未設定の場合は、null が渡されます。
- Java オブジェクトに渡されない項目はキーが存在しない状態になります。存在しないキーを指定した場合は、null が渡されます。
- 日時情報（開始日、生成日、および処理期限）を取得すると、1970/01/01 00:00:00 GMT を起点とした通算秒の文字列が渡されます。
- 補助定数パラメタを取得した場合は、ビジネスプロセス定義で指定された文字列が、空白を含めてそのまま文字列で渡されます。
- 子作業生成ルールの作業生成日時、作業開始日時、作業処理期限、作業優先度、および作業者には、それぞれ生成ルールを定義した並列作業の作業生成日時、作業開始日時、作業処理期限、作業優先度、および作業者が渡されます。

## 2.6.5 作業アプリケーションを作成するときの注意事項

作業アプリケーションを作成するときの注意事項について説明します。

- 作業の返却、および再開を行った場合、作業アプリケーションは同じ情報で呼び出されます。このため、作業アプリケーションは、複数回呼び出されても正しく一度だけ業務処理を行うように作成する必要があります。作業アプリケーションに渡される情報については、「[2.6.4\(2\) Java オブジェクトに渡される情報 \(作業アプリケーションの場合\)](#)」を参照してください。
- 作業アプリケーションの同期呼び出しで、エラーが発生し、ロールバックされたあとに再度呼び出す場合、作業 ID は異なる値になることを考慮して作成する必要があります。
- 作業アプリケーションの着手 (perform)、および完了 (performAndComplete) は、必ずメソッドの戻り値で指定してください。

## 2.7 一覧情報の取得

CSCIW の API で情報を取得する方法について説明します。

### 2.7.1 一覧情報の取得手順と基本動作

一覧情報の取得手順と基本動作について説明します。

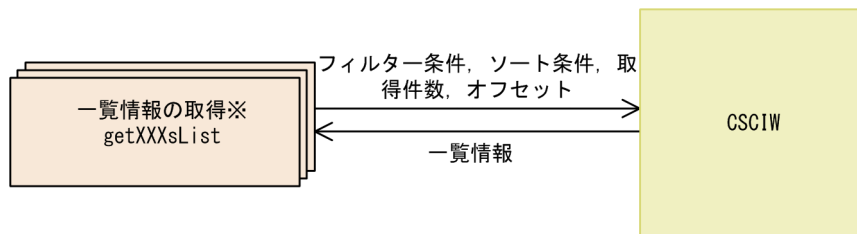
#### (1) 一覧情報の取得手順

getXXXsList メソッドで一覧情報を取得する手順について説明します。

なお、getXXXsList メソッドの「XXX」の部分は、取得する一覧情報にあわせて次の中から該当する文字列に置き換えてください。

- ActivityInstance (業務ステップオブジェクトの一覧情報を取得する場合)
- ProcessDefinition (ビジネスプロセス定義オブジェクトの一覧情報を取得する場合)
- ProcessInstance (案件オブジェクトの一覧情報を取得する場合)
- WorkDefinition (作業定義オブジェクトの一覧情報を取得する場合)
- WorkItem (作業オブジェクトの一覧情報を取得する場合)

図 2-4 getXXXsList メソッドで一覧情報を取得する手順



注※ 必要な回数だけ繰り返します。

#### 説明

フィルター条件、ソート条件、取得件数、オフセットなどを指定して getXXXsList メソッドで検索結果を取得します。

getXXXsList メソッドは一度で複数の件数を取得できますが、一度の取得単位を少なくし、getXXXsList メソッドを繰り返すことで、少しずつ順に検索結果を取得することもできます。

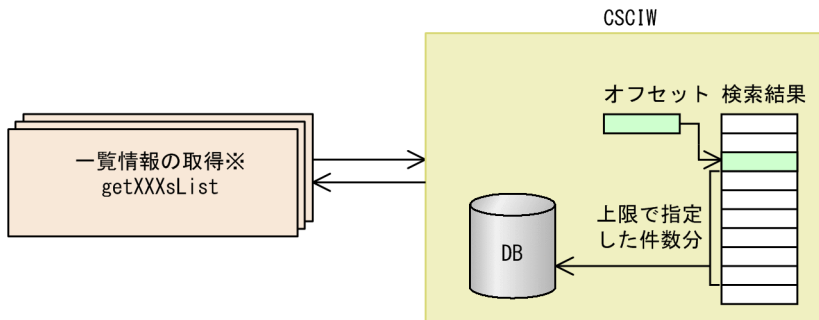
ただし、メソッドを繰り返す間、検索の集合が固定されないため、タイミングによっては、続きのオフセットを指定しても、案件情報の重複や漏れが発生します。

なお、getXXXsList メソッドは、実際には一覧取得する情報の種類によって、それぞれ個別のメソッドを利用します。

## (2) 一覧情報取得時の CSCIW 内での動作

getXXXsList メソッドで一覧情報を取得する場合の CSCIW 内での動作について説明します。

図 2-5 getXXXsList メソッドで一覧情報を取得する場合の CSCIW 内での動作



注※ 必要な回数だけ繰り返します。

### 説明

指定されたフィルター条件を満たす情報を指定されたソート順で返し、そのあと、検索結果領域を解放します。

getXXXsList メソッドでは、取得する件数の上限とオフセットを指定できます。この場合は、取得情報として検索結果の上位から指定されたオフセット分を除いて、指定された上限件数分だけ情報を返します。例えば、検索結果の 101 番目から 50 件の情報を取得する場合は、オフセットに 100 を指定し、上限に 50 を指定します。ソートを指定した場合は、全検索結果をソートしたあと、上位から指定されたオフセット分を除いて、指定された上限件数分だけ情報を返します（実際のソートはデータベースの機能で実行します）。取得件数の上限を指定すると、CSCIW 内のリソース消費量を削減できます。

### 注意事項

取得する全情報数が多くなるとリソース不足が発生するおそれがあるので、取得する全情報数に注意して使用してください。

取得する全情報数とは、取得する最大情報数の指定（aMaxCount）とは無関係です。常にフィルター条件に合致する全情報数のことです。

## (3) 一覧情報取得時の最大取得件数の指定

一覧取得する情報数については、取得する情報の最大取得件数を getXXXsList メソッドで指定できます。

最大取得件数を指定する場合、getXXXsList メソッドの呼び出しで取得する情報の最大件数を、引数（aMaxCount）で指定します。この場合は、フィルター条件を満たす情報のうち、指定された最大件数分だけ検索結果を取得します。このため、フィルター条件を満たす情報数が引数（aMaxCount）で指定した数より多い場合でも、getXXXsList メソッドで取得できる情報数の上限は引数（aMaxCount）で指定した最大件数までとなります。getXXXsList メソッドで取得できる情報数の上限を超える情報を取得しようとした場合は、全情報を取得する指定（aMaxCount の値が-1）でこれ以上情報が存在しないときに、さらに情報を取得しようとしたときと同じ動作（長さ 0 の属性値リストを返す）をします。

なお、引数 (aMaxCount) を 0 にすると、長さ 0 の属性値リストが返されます。

## 2.7.2 一覧情報取得時のフィルター条件

一覧情報取得時には情報を絞り込むためのフィルター条件を設定できます。フィルター条件は、SQL の WHERE 句の形式で指定します。実際には、一覧情報を取得できる API で発行する SQL の WHERE 句のあとに指定したフィルター条件が付加されます。

### (1) 指定方法

フィルター条件指定の構文は、使用する DBMS の SQL 構文に従います。

#### (a) 指定できる属性

フィルター条件として指定できる属性については、「付録 B 参照系 API で指定できる属性」を参照してください。

なお、属性を指定する場合の注意事項を次に示します。

- 属性名は文字列で指定してください。
- 指定する属性名は、「"」(半角ダブルクォーテーション) で囲まないでください。

#### (b) 属性値の指定方法

- 日付データを条件に指定する場合

CIWUtil クラスの convertDateToString メソッドを使用します。このメソッドは、Date オブジェクトから、フィルター条件で使用できる日付情報の値に変換した文字列を返すメソッドで、返された文字列を連結してフィルター条件を生成します。

例

```
SimpleDateFormat format = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");
Date date = format.parse("2006/08/01 00:00:00");

String filterCondition = "StartDate < " + CIWUtil.convertDateToString(date);
```

- 状態を条件に指定する場合

フィルター条件で状態の値を使用する場合、各クラスに定義された列挙型 State の toStateCode メソッドを使用します。

例

```
String filterCondition = "StateCode = '" + CIWorkItem.State.READY.toStateCode() + "'";
```

- 種類を条件に指定する場合

フィルター条件で種類の値を使用する場合、各クラスに定義された列挙型 Type の toTypeCode メソッドを使用します。

例

```
String filterCondition = "WorkTypeCode = '" + CIWorkItem.Type.NORMAL.toTypeCode() + "'";
```

## (2) 注意事項

- フィルター条件の長さは DBMS の制限にだけ依存します。したがって、フィルター条件の長さは CSCIW ではチェックしません。
- フィルター条件の文字列は CSCIW では内容をチェックしないで、SQL の WHERE 句をそのまま構成する文字列として使用します。そのため、空文字を指定すると SQL 例外になることがあります。フィルター条件を省略する場合は、null を指定してください。

### 2.7.3 一覧情報取得時のソート条件

一覧情報取得時には、ソート条件を設定できます。ソート条件は、SQL の ORDER BY 句の形式で指定します。実際には、一覧情報を取得できる API で発行する SQL の ORDER BY 句のあとに指定したソート条件が付加されます。

#### (1) 指定方法

ソート条件指定の構文は、使用する DBMS の SQL 構文に従います。

##### (a) 指定できる属性

ソート条件として指定できる属性については、「[付録 B 参照系 API で指定できる属性](#)」を参照してください。

なお、属性を指定する場合の注意事項を次に示します。

- 属性名は文字列で指定してください。
- 指定する属性名は、「"」（半角ダブルクォーテーション）で囲まないでください。

## (2) 注意事項

- ソート条件の長さは DBMS の制限にだけ依存します。したがって、ソート条件の長さは CSCIW ではチェックしません。
- ソート条件の文字列は CSCIW では内容をチェックしないで、SQL の ORDER BY 句をそのまま構成する文字列として使用します。そのため、空文字を指定すると SQL 例外になることがあります。ソート条件を省略する場合は、null を指定してください。



## 2.8 指定できるデータ

---

CSCIW の API で扱うデータについて説明します。

### 2.8.1 オブジェクトの取得で指定できる値

オブジェクトを取得する場合に、指定できる値を次に示します。

#### ID

ID はすべて `java.lang.Integer` 型で定義されています。指定できる値は、`java.lang.Integer` 型の範囲に従います。

なお、`null` は指定できません。

#### 取得する属性のセット

取得する属性のセットは、`java.util.Set` 型で定義されています。オブジェクトごとの `AttributeName` 列挙型の定数を要素に持っています。

なお、`null` を指定した場合は、要素を持たないサイズ 0 の属性のセットを指定するのと同じ動作になります。

### 2.8.2 オブジェクトのリストの取得で指定できる値

オブジェクトのリストを取得する場合に、指定できる値を次に示します。

#### フィルター条件

フィルター条件は `java.lang.String` 型で定義されています。指定できる値は、`java.lang.String` 型の範囲に従います。

なお、`null` を指定した場合は、対象となるすべてのオブジェクトが取得対象になります。

#### ソート条件

ソート条件は `java.lang.String` 型で定義されています。指定できる値は、`java.lang.String` 型の範囲に従います。

なお、`null` を指定した場合は、ソートは行われません。

#### オフセット

オフセットは `int` 型で定義されています。

なお、0 未満の値は指定できません。0 を先頭として指定してください。

#### 最大取得件数

最大取得件数は `int` 型で定義されています。

なお、-1 未満の値は指定できません。また、-1 を指定した場合は、すべてを取得します。

## 取得する属性セット

取得する属性のセットについては、「[2.8.1 オブジェクトの取得で指定できる値](#)」を参照してください。

## 2.8.3 更新系 API で指定できる値

更新系 API で指定できる値を次に示します。更新系 API の種類については、「[付録 A.1 更新系 API](#)」を参照してください。

### 日付

日付は `java.util.Date` 型で定義されています。

なお、`null` を指定した場合は、日付情報には `null` が設定されます。

注意事項を次に示します。

- 日付の設定は、ミリ秒以下については切り捨てます。
- 日付の有効範囲は、1970/01/02 00:00:00 GMT~292278993/12/31 00:00:00 GMT の範囲です。ただし、無限遠の過去および未来は、`CIWUtil.InfinityDate` で定義されているため、それぞれを表す `java.util.Date` 型を生成して指定した場合は、有効範囲外でも正常に動作します。

### 優先度

優先度は `java.lang.Integer` 型で定義されています。指定できる値は、`java.lang.Integer` 型の範囲に従います。

なお、`null` を指定した場合は、優先度には `null` が設定されます。

### 状態

状態はオブジェクトごとの State 列挙型で定義されています。指定できる値は、オブジェクトごとの State 列挙型に存在する定数です。

### ビジネスプロセス定義名

ビジネスプロセス定義名は `java.lang.String` 型で定義されています。

なお、空文字列および `null` は指定できません。

### ビジネスプロセス定義バージョン

ビジネスプロセス定義バージョンは `java.lang.Short` 型で定義されています。

なお、`null` を指定した場合は、ビジネスプロセス定義の中で最新バージョンが設定されます。

### 案件属性のマップ

案件属性のマップは `java.util.Map` 型で定義されています。案件属性のマップは、案件生成時にユーザーが指定できる属性のマップです。

なお、`null` を指定した場合は、要素を持たないサイズ 0 の案件属性のマップを指定した場合と同じ動作になります。

マップの要素として指定できる属性を次に示します。

- **案件名**  
案件名は `java.lang.String` 型で定義されています。

なお、null を指定した場合は、案件名には null が設定されます。また、空文字列は指定できません。

- **処理期限**  
処理期限については、「日付」を参照してください。
- **優先度**  
優先度については、「優先度」を参照してください。

## 作業者 ID

作業者 ID は `java.lang.String` 型で定義されています。

なお、null を指定した場合は、作業者 ID には null が設定されます。また、空文字は指定できません。注意事項を次に示します。

- 作業者 ID は、`<SYSTEMID>_WORK_ITEM` テーブルの Participant カラムのバイト数以下になるように設定してください。

## 2.8.4 その他のクラスで指定できる値

その他のクラスで指定できる値を次に示します。

### 件数を取得する対象

件数を取得する対象は、CIWServer の `getListCount` で指定できます。CIWServer.Targetkind 列挙型で定義されています。

指定できる値は、CIWServer.Targetkind 列挙型に存在する定数です。

### ユーザ記述子

ユーザ記述子は、CIWFactory の `createCIWServer` メソッドで指定できます。java.lang.String 型で定義されています。

なお、空文字列および null は指定できません。

注意事項を次に示します。

- ユーザ記述子は、`<SYSTEMID>_PROCESS_INSTANCE` テーブルの Creator カラムのバイト数以下になるように設定してください。

### データソースオブジェクト

データソースオブジェクトは、CIWAdmin#initializeCIWFactory メソッドおよび CIWFactory の `createCIWServer` メソッドで指定できます。javax.sql.DataSource 型で定義されています。

なお、null は指定できません。

CIWFactory の `createCIWServer` メソッドでは、次に示す場合だけ指定できます。

- J2EE サーバ上の EJB から利用するなど、DataSource から取得した `java.sql.Connection` が、実行スレッドで同一のトランザクションを使用してデータソースにアクセスできる場合 (DBConnector を使用する場合)

## データベースコネクション

データベースコネクションは、CIWConnectionCoordinator の attachDatabaseConnection メソッドで指定できます。java.sql.Connection 型で定義されています。

なお、null は指定できません。また、自動コミット機能は無効に設定してください。

## 2.9 CSCIW で管理するトランザクション

---

初期化処理・終了処理、および運用中の ID 採番処理におけるトランザクションは、CSCIW が管理します。そのため、コネクションは業務プログラムで使用しているものとは別に独自に取得して利用します。ID 採番処理など複数のスレッドから同時に要求される処理であっても、同時に実行するトランザクションは最大 1 つになるように制御します。コネクションオブジェクトそのものも、同時に使用するのは 1 つだけです。

業務プログラムが Java アプリケーションの場合、CIWAdmin クラスの initializeCIWFactory メソッドの引数に渡すデータソースでは、グローバルトランザクションは使用できません。初期化処理・終了処理、および運用中の ID 採番処理で使用するコネクションは、CIWAdmin クラスの initializeCIWFactory メソッドの引数で渡されたデータソースから取得します。CSCIW がトランザクションを管理するため、`java.sql.Connection` のコミットおよびロールバックメソッドを使用できる必要があります。

## 2.10 業務プログラムで管理するトランザクション

---

API を実行する場合は、業務プログラムのトランザクションと CSCIW が利用するトランザクションを、同一トランザクションで処理します。これをトランザクション共有といいます。

トランザクションは、CSCIW では管理しないため、業務プログラム側で管理する必要があります。

### 2.10.1 CIWServer とコネクションの関連づけ

CIWServer とコネクションの関連づけは、CIWConnectionCoordinator クラスを使用するか、または J2EE サーバが提供するデータソースを使用することで実行できます。

#### (1) CIWConnectionCoordinator を使用した Connection オブジェクトの関連づけ

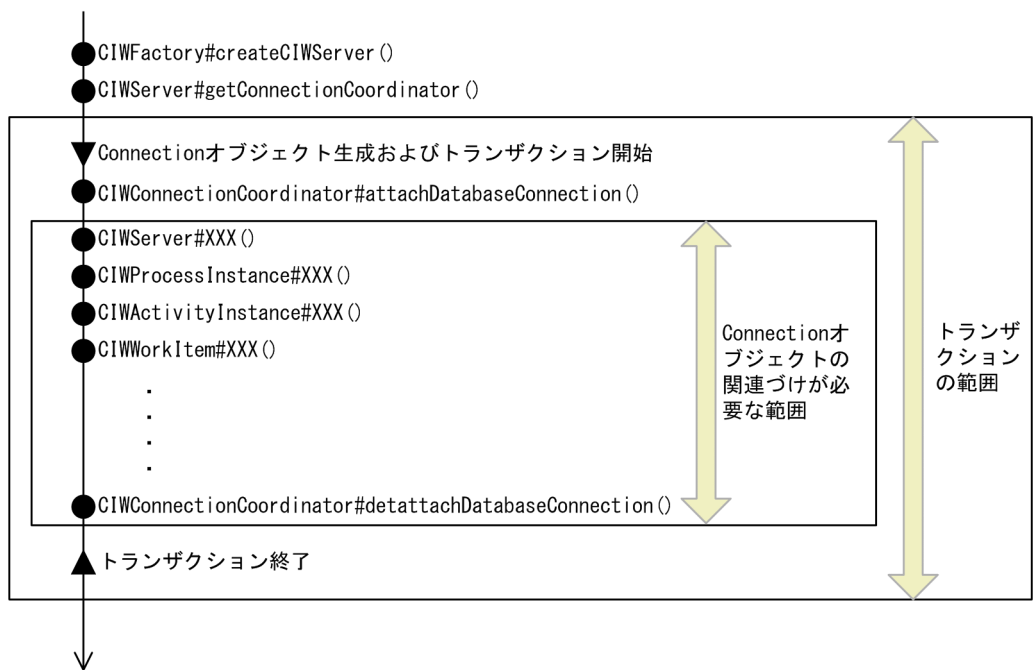
コネクションを直接共有するために、CIWConnectionCoordinator クラスを使用し、コネクションオブジェクトと CIWServer オブジェクトを関連づけます。

この関連づけは、CIWServer および CIWServer から取得するオブジェクトごとの API を使用する前に、必ず実施してください。関連づけを行わないで API を実行した場合は、例外が発生します。

また、トランザクションを終了する前に、コネクションと CIWServer との関連づけを解除する必要があります。なお、関連づけの解除時は、CSCIW ではトランザクション操作およびコネクションオブジェクトの close() は行いません。

コネクションの関連づけのタイミングとトランザクションの範囲を次の図に示します。

図 2-6 コネクションの関連づけのタイミングとトランザクションの範囲



(凡例)

↓ : 業務プログラムの処理の流れ

● : API処理

▼ : Connectionオブジェクト生成およびトランザクション開始

▲ : トランザクションの終了

関連づけの機能は、Java アプリケーションおよび J2EE サーバ上で使用する場合に使用できます。

## (2) データソースを使用した Connection オブジェクトの関連づけ

J2EE サーバ上で DBConnector を利用する場合だけ、J2EE サーバが提供するデータソース (DBConnector) オブジェクトと CIWServer を直接関連づけることができます。ただし、J2EE サーバ上で DBConnector を利用する場合でも、JDBC のデータソースを直接使用することはできません。

データソースの関連づけは、CIWServer 生成時に行います。関連づけたデータソースは、オブジェクトが消滅するまで有効です。

なお、CIWServer から生成するオブジェクトは、CIWServer に関連づけられたデータソースを参照します。したがって、CIWServer から生成されたオブジェクトよりも CIWServer が先に消滅した場合でも、関連づけられたデータソースを参照し続けます。

### 2.10.2 Java オブジェクト呼び出しとコネクションの関連づけ

条件、振り分けルール、生成ルール、または作業アプリケーションで Java オブジェクト呼び出しを利用する場合、実行される Java オブジェクトから CIWServer に関連づけられたコネクションを利用できます。

## 2.10.3 トランザクションの開始

CSCIW では、API 実行時にはトランザクションを明示的に開始しません。

業務プログラムでは、CSCIW の API を呼び出す前に、トランザクションを開始しておく必要があります。

### 注意事項

- CIWConnectionCoordinator を使用してコネクションを関連づける前に、setAutoCommit()メソッドを使用し、自動コミット機能を無効にしてください。
- J2EE サーバ使用時に、データソースと関連づける方式で実行する場合は、業務プログラム側で J2EE サーバが管理するトランザクションを開始してください。開始しないと、DBCConnector や EJB の種類および設定によっては、API と業務プログラムで発行する SQL が異なるトランザクションで実行されてしまうことがあります。この場合は、実行中に障害が発生すると、ワーク管理データベースと業務データベースの間で不整合が発生することがあります。

## 2.10.4 トランザクションの終了

CSCIW では、API で利用したトランザクションを終了しません。したがって、業務プログラムでトランザクションを終了してください。

また、CIWConnectionCoordinator を利用したコネクションの関連づけを行っている場合は、トランザクションを終了する前に関連づけを解除してください。



## 2.11 業務プログラムとデータベース間での文字コード

---

CSCIW では、文字コード変換を行いません。したがって、業務プログラムとデータベースで文字コードが異なる場合は、業務プログラム内で文字コード変換を行うか、JDBC およびデータベースの機能を利用してください。

また、CSCIW では、API 引数に指定された文字列を、データベースの文字コードに合わせてバイトコードに変換し、可変長文字列カラムに格納できる長さであるかチェックします。

### メモ

PostgreSQL の場合、可変長文字列カラムの列長はバイト数ではなく文字数を表しますが、チェックの際は列長をバイト数と見なしてチェックします。

## 2.12 API 利用時の注意事項

---

### 2.12.1 スレッドからの利用について

スレッド利用時の留意点を次に示します。

- CIWFactory は、複数のスレッドからアクセスできます。
- CIWServer および CIWServer から生成する各種オブジェクトは、1つのスレッドからだけアクセスできます。複数のスレッドから関連するオブジェクトに同時にアクセスしないでください。

### 2.12.2 CSCIWManagementServer を停止した状態での API 実行について

J2EE サーバ上で CSCIW を使用する場合に、CSCIWManagementServer を停止した状態で API を実行すると、CIWFatalException が発生します。CSCIWManagementServer を再起動することで再び使用できます。この場合は、取得済みのオブジェクトは再取得する必要はありません。取得済みのオブジェクトは、そのまま使用できます。

### 2.12.3 エラーが発生した場合のトランザクションの終了について

API 実行中にエラーが発生した場合は、必ずロールバックでトランザクションを終了してください。コミットで終了すると、データに不整合が発生する場合があります。

### 2.12.4 排他およびデッドロックについて

排他およびデッドロックについて、次に示すことに注意してください。

- 同じトランザクション内で複数の更新系のメソッドを使用すると、デッドロックが発生するおそれがあります。同じトランザクション内で複数の更新系のメソッドを使用しないでください。ただし、1つの更新系のメソッドと更新系以外のメソッドは、同一トランザクション内でも使用できます。
- トランザクションの開始から終了までの間隔は、できるだけ短くしてください。また、更新系のメソッドを呼び出した場合は、できるだけ早くコミットまたはロールバックしてください。複数のプログラムで実行しているときなどに、排他待ちが発生して処理速度が低下することがあります。

### 2.12.5 finalize()メソッド内での使用について

すべてのクラスおよびインタフェースの API は、finalize()メソッド内で使用しないでください。

## 2.12.6 参照系 API の使用について

参照系 API を使用する際には、次に示すことに注意してください。CSCIW で使用する参照系 API の種類については、「付録 A.2 参照系 API」を参照してください。

- CSCIW の参照系 API は、SQL 発行時に次に示すオプションを使用して排他（行ロック）をかけないで検索を実行しています。そのため、別トランザクションの処理で検索対象行の内容が変更されているおそれがあります。
  - HiRDB：WITHOUT LOCK NOWAIT
  - Oracle：FOR UPDATE 句など、何も指定しない
  - SQL Server：  
SQLServerReadMode<sup>※</sup> = 0 の場合、WITH(ROWLOCK, NOLOCK)  
SQLServerReadMode<sup>※</sup> = 1 の場合、WITH(READCOMMITTED)  
注※ CSCIW セットアッププロパティファイルの設定です。
  - PostgreSQL：FOR UPDATE 句など、何も指定しない

なお、排他をかけないで検索を実行した場合の動作は、データベース種別ごとに異なります。動作の詳細については、ご利用のデータベースのマニュアルを参照してください。

# 3

## 提供 API 一覧

この章では、CSCIW が提供する Java API の概要を一覧形式で説明します。また、パッケージ「jp.co.Hitachi.soft.csciw」およびパッケージ「jp.co.Hitachi.soft.csciw.callback」の階層関係も示します。

## 3.1 クラスの一覧

クラスの一覧をパッケージごとに示します。各表では、クラスを役割によって分類し、クラスおよびインタフェースごとにアルファベット順で記載しています。

### ヒント

BPMN 連携機能を使用する場合に CSCIW で使用する API については、マニュアル「uCosminexus Service Coordinator Interactive Workflow BPMN 連携機能 使用の手引」を参照してください。

### 3.1.1 パッケージ「jp.co.Hitachi.soft.csciw」に所属するクラス

パッケージ「jp.co.Hitachi.soft.csciw」に所属するクラスの一覧を表 3-1 から表 3-5 に示します。

#### • 初期化・終了クラス

表 3-1 初期化・終了クラスの一覧（パッケージ「jp.co.Hitachi.soft.csciw」）

クラス名	説明
CIWAdmin	CSCIW の初期化処理と終了処理をします。

#### • 案件実行・運用クラスおよびインタフェース

表 3-2 案件実行・運用クラスおよびインタフェースの一覧（パッケージ「jp.co.Hitachi.soft.csciw」）

クラス名およびインタフェース名	説明
CIWActivityDefinition	業務ステップ定義のインタフェースです。
CIWActivityInstance	業務ステップのインタフェースです。 業務ステップの属性取得、属性変更、状態変更、条件評価などのメソッドを提供します。
CIWFactory	CIWServer オブジェクトのファクトリクラスです。 CIWServer オブジェクトを生成します。
CIWProcessDefinition	ビジネスプロセス定義のインタフェースです。 ビジネスプロセス定義の属性取得メソッドを提供します。
CIWProcessInstance	案件のインタフェースです。 案件の属性取得、属性変更、状態変更、強制終了などのメソッドを提供します。
CIWServer	ワーク管理システム全般にかかわる処理を行うための機能を提供するインタフェースです。次に示すメソッドを提供します。 <ul style="list-style-type: none"><li>• 案件の投入</li></ul>

クラス名および インタフェース名	説明
	<ul style="list-style-type: none"> <li>案件、業務ステップ、作業、ビジネスプロセス定義などのオブジェクトおよびリストの取得</li> </ul>
CIWorkDefinition	作業定義のインタフェースです。
CIWorkItem	作業のインタフェースです。 作業の属性取得、属性変更、状態変更、条件評価、作業変更などのメソッドを提供します。

- その他のクラス

表 3-3 その他のクラスの一覧 (パッケージ [jp.co.Hitachi.soft.csciw])

クラス名	説明
CIWConnectionCoordinator	JDBC コネクションと CIWServer オブジェクトの関連づけをするクラスです。
CIWUtil	各種ユーティリティ機能を提供するクラスです。

- 列挙型

表 3-4 列挙型の一覧 (パッケージ [jp.co.Hitachi.soft.csciw])

列挙型名	説明
CIWActivityDefinition.AttributeName	業務ステップ定義属性の属性名の列挙型です。
CIWActivityInstance.AttributeName	業務ステップ属性の属性名の列挙型です。
CIWActivityInstance.State	業務ステップの状態の列挙型です。
CIWActivityInstance.Type	業務ステップの種類列挙型です。
CIWProcessDefinition.AttributeName	ビジネスプロセス定義属性の属性名の列挙型です。
CIWProcessDefinition.State	ビジネスプロセス定義の状態列挙型です。
CIWProcessInstance.AttributeName	案件属性の属性名の列挙型です。
CIWProcessInstance.State	案件の状態列挙型です。
CIWServer.Targetkind	ワーク管理システムが管理するオブジェクトの種別を表す列挙型です。
CIWUtil.InfinityDate	無限遠の日付の列挙型です。
CIWorkDefinition.AttributeName	作業定義属性の属性名の列挙型です。
CIWorkDefinition.Type	作業定義の種類列挙型です。
CIWorkItem.AttributeName	作業属性の属性名の列挙型です。
CIWorkItem.State	作業の状態列挙型です。
CIWorkItem.Type	作業の種類列挙型です。

- 例外クラス

表 3-5 例外クラスの一覧 (パッケージ「jp.co.Hitachi.soft.csciw」)

クラス名	説明
CIWEntityNotExistException	処理しようとしたオブジェクトが存在しない場合にスローされる例外クラスです。
CIWException	CSCIW で発生する例外の基底となる抽象クラスです。
CIWFatalException	CSCIW で処理を続行できない障害が発生した場合にスローされる例外クラスです。
CIWSQLException	DBMS で処理を続行できない障害が発生した場合にスローされる例外クラスです。
CIWSQLTransientException*	DBMS の一時的な要因で処理を実行できない場合にスローされる例外クラスです。
CIWStateException	不正な状態に関する例外クラスです。
CIWTransientException	一時的な要因で処理を実行できない場合にスローされる例外クラスです。
CIWTransitionFailedException	案件処理中にエラーが発生した場合にスローされる例外クラスです。

注※

デフォルトでは、すべてのデータベースアクセスのエラーが CIWSQLException になります。環境設定ファイルに一時的エラーとして扱う SQL コードを定義すると、CIWSQLException の代わりに CIWSQLTransientException をスローします。

### 3.1.2 パッケージ「jp.co.Hitachi.soft.csciw.callback」に所属するクラス

パッケージ「jp.co.Hitachi.soft.csciw.callback」に所属するクラスの一覧を表 3-6 から表 3-8 に示します。

- Java オブジェクト呼び出しのインタフェース

表 3-6 インタフェースの一覧 (パッケージ「jp.co.Hitachi.soft.csciw.callback」)

クラス名	説明
CIWConditionEvaluator	条件 Java オブジェクトのインタフェースです。条件 Java オブジェクトで実装する必要があります。
CIWCreationRuleEvaluator	生成ルール Java オブジェクトのインタフェースです。生成ルール Java オブジェクトで実装する必要があります。
CIWResourceSelector	振り分けルール Java オブジェクトのインタフェースです。振り分けルール Java オブジェクトで実装する必要があります。
CIWWorkApplication	作業アプリケーション Java オブジェクトのインタフェースです。作業アプリケーション Java オブジェクトで実装する必要があります。

- 列挙型

表 3-7 列挙型の一覧 (パッケージ [jp.co.Hitachi.soft.csciw.callback])

クラス名	説明
CIWConditionEvaluator.Parameter	条件 Java オブジェクトに渡されるパラメタキーの列挙型です。
CIWCreationRuleEvaluator.Parameter	生成ルール Java オブジェクトに渡されるパラメタキーの列挙型です。
CIWResourceSelector.Parameter	振り分けルール Java オブジェクトに渡されるパラメタキーの列挙型です。
CIWWorkApplication.Parameter	作業アプリケーション Java オブジェクトに渡されるパラメタキーの列挙型です。
CIWWorkApplication.Result	作業アプリケーションの業務処理結果の列挙型です。

- 例外クラス

表 3-8 例外クラスの一覧 (パッケージ [jp.co.Hitachi.soft.csciw.callback])

クラス名	説明
CIWUserException	Java オブジェクト呼び出し時の例外クラスです。



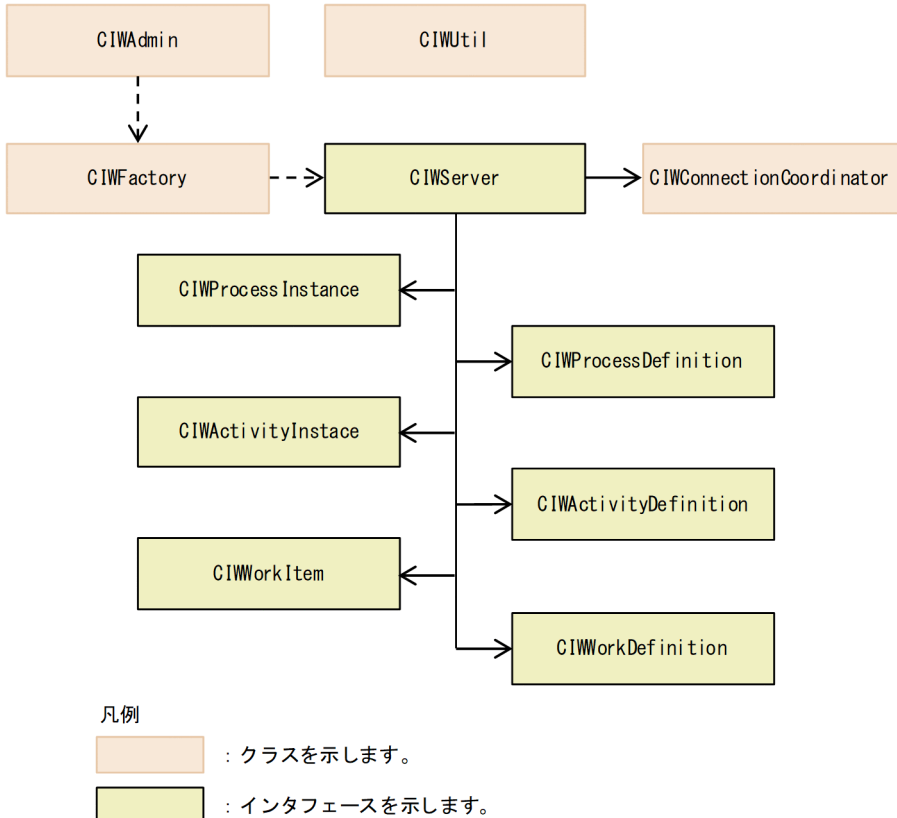
## 3.2 クラスの構成

CSCIW のクラスの構成をパッケージごとに示します。

### 3.2.1 パッケージ「jp.co.Hitachi.soft.csciw」のクラス構成

パッケージ「jp.co.Hitachi.soft.csciw」のクラス構成を次の図に示します。

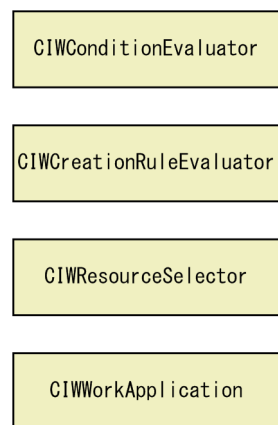
図 3-1 パッケージ「jp.co.Hitachi.soft.csciw」のクラス構成




### 3.2.2 パッケージ「jp.co.Hitachi.soft.csciw.callback」のクラス構成

パッケージ「jp.co.Hitachi.soft.csciw.callback」のクラス構成を次の図に示します。

図 3-2 パッケージ「jp.co.Hitachi.soft.csiw.callback」のクラス構成



凡例

 : インタフェースを示します。

## 3.3 パッケージ「jp.co.Hitachi.soft.csciw」の階層

パッケージ「jp.co.Hitachi.soft.csciw」の階層を分類別に示します。

### 3.3.1 初期化・終了クラスの階層

初期化・終了クラスの階層を示します。

```
java.lang.Object
└─jp.co.Hitachi.soft.csciw.CIWAdmin
```

### 3.3.2 案件実行・運用クラスおよびインタフェースの階層

案件実行・運用クラスおよびインタフェースの階層を示します。

```
java.lang.Object
└─jp.co.Hitachi.soft.csciw.CIWFactory

interface jp.co.Hitachi.soft.csciw.CIWActivityDefinition
interface jp.co.Hitachi.soft.csciw.CIWActivityInstance
interface jp.co.Hitachi.soft.csciw.CIWProcessDefinition
interface jp.co.Hitachi.soft.csciw.CIWProcessInstance
interface jp.co.Hitachi.soft.csciw.CIWServer
interface jp.co.Hitachi.soft.csciw.CIWWorkDefinition
interface jp.co.Hitachi.soft.csciw.CIWWorkItem
```

### 3.3.3 その他の階層

その他の階層を示します。

```
java.lang.Object
├─jp.co.Hitachi.soft.csciw.CIWConnectionCoordinator
└─jp.co.Hitachi.soft.csciw.CIWUtil
```

### 3.3.4 列挙型の階層

列挙型の階層を示します。

```
java.lang.Object
├─java.lang.Enum<CIWActivityDefinition.AttributeName>
│   └─jp.co.Hitachi.soft.csciw.CIWActivityDefinition.AttributeName
└─java.lang.Enum<CIWActivityInstance.AttributeName>
```

```

|   ↳jp.co.Hitachi.soft.csciw.CIWActivityInstance.AttributeName
├java.lang.Enum<CIWActivityInstance.State>
|   ↳jp.co.Hitachi.soft.csciw.CIWActivityInstance.State
├java.lang.Enum<CIWActivityInstance.Type>
|   ↳jp.co.Hitachi.soft.csciw.CIWActivityInstance.Type
├java.lang.Enum<CIWProcessDefinition.AttributeName>
|   ↳jp.co.Hitachi.soft.csciw.CIWProcessDefinition.AttributeName
├java.lang.Enum<CIWProcessDefinition.State>
|   ↳jp.co.Hitachi.soft.csciw.CIWProcessDefinition.State
├java.lang.Enum<CIWProcessInstance.AttributeName>
|   ↳jp.co.Hitachi.soft.csciw.CIWProcessInstance.AttributeName
├java.lang.Enum<CIWProcessInstance.State>
|   ↳jp.co.Hitachi.soft.csciw.CIWProcessInstance.State
├java.lang.Enum<CIWServer.Targetkind>
|   ↳jp.co.Hitachi.soft.csciw.CIWServer.Targetkind
├java.lang.Enum<CIWUtil.InfinityDate>
|   ↳jp.co.Hitachi.soft.csciw.CIWUtil.InfinityDate
├java.lang.Enum<CIWorkDefinition.AttributeName>
|   ↳jp.co.Hitachi.soft.csciw.CIWorkDefinition.AttributeName
├java.lang.Enum<CIWorkDefinition.Type>
|   ↳jp.co.Hitachi.soft.csciw.CIWorkDefinition.Type
├java.lang.Enum<CIWorkItem.AttributeName>
|   ↳jp.co.Hitachi.soft.csciw.CIWorkItem.AttributeName
├java.lang.Enum<CIWorkItem.State>
|   ↳jp.co.Hitachi.soft.csciw.CIWorkItem.State
├java.lang.Enum<CIWorkItem.Type>
|   ↳jp.co.Hitachi.soft.csciw.CIWorkItem.Type

```

### 3.3.5 例外クラスの階層

例外クラスの階層を示します。

```

java.lang.Object
├java.lang.Throwable (implements java.io.Serializable)
├java.lang.Exception
├jp.co.Hitachi.soft.csciw.CIWException
├jp.co.Hitachi.soft.csciw.CIWEntityNotExistException
├jp.co.Hitachi.soft.csciw.CIWFatalException
├jp.co.Hitachi.soft.csciw.CIWSQLException
├jp.co.Hitachi.soft.csciw.CIWStateException
├jp.co.Hitachi.soft.csciw.CIWTransientException
├jp.co.Hitachi.soft.csciw.CIWSQLTransientException
├jp.co.Hitachi.soft.csciw.CIWTransitionFailedException

```

## 3.4 パッケージ「jp.co.Hitachi.soft.csciw.callback」の階層

パッケージ「jp.co.Hitachi.soft.csciw.callback」の階層を分類別に示します。

### 3.4.1 インタフェースの階層

インタフェースの階層を示します。

```
java.lang.Object  
  
interface jp.co.Hitachi.soft.csciw.callback.CIWConditionEvaluator  
interface jp.co.Hitachi.soft.csciw.callback.CIWCreationRuleEvaluator  
interface jp.co.Hitachi.soft.csciw.callback.CIWResourceSelector  
interface jp.co.Hitachi.soft.csciw.callback.CIWWorkApplication
```

### 3.4.2 列挙型の階層

列挙型の階層を示します。

```
java.lang.Object  
└─java.lang.Enum<E> (implements java.lang.Comparable<T>, java.io.Serializable)  
    └─CIWConditionEvaluator.Parameter  
        └─CIWCreationRuleEvaluator.Parameter  
            └─CIWResourceSelector.Parameter  
                └─CIWWorkApplication.Parameter  
                    └─CIWWorkApplication.Result
```

### 3.4.3 例外クラスの階層

例外クラスの階層を示します。

```
java.lang.Object  
└─java.lang.Throwable (implements java.io.Serializable)  
    └─java.lang.Exception  
        └─CIWUserException
```

# 4

## 初期化・終了クラスが提供するメソッド

この章では、初期化・終了クラスが提供するメソッドについて説明します。

## 4.1 記述形式

---

### 4.1.1 初期化・終了クラスの記述形式

初期化・終了クラスは、次に示す形式を基本として説明しています。

#### クラス定義

クラス定義を示します。

#### メソッド

クラスのメソッドを説明します。

### 4.1.2 初期化・終了クラスが提供するメソッドの記述形式

初期化・終了クラスが提供するメソッドは、次に示す形式を基本として説明しています。

#### メソッド名

メソッド名を示します。

#### クラス名

メソッドが所属するクラス名を示します。

#### 構文

メソッドの構文を示します。

#### 機能

メソッドの機能を説明します。

#### 引数

メソッドの引数を説明します。

#### 例外

メソッドに発生する例外を説明します。

#### 注意事項

メソッドの注意事項を説明します。

## 4.2 CIWAdmin (CSCIW の初期化・終了処理をするクラス)

CSCIW の初期化および終了処理をするクラスです。

このクラスは、Java アプリケーションでだけ使用できます。J2EE アプリケーションで使用した場合の動作は保証しません。

### クラス定義

```
public final class CIWAdmin extends java.lang.Object
```

### メソッド

CIWAdmin クラスのメソッドを次の表に示します。

表 4-1 CIWAdmin クラスのメソッド

メソッド名	説明
<code>finalizeCIWFactory</code>	CSCIW の終了処理をします。
<code>initializeCIWFactory</code>	CSCIW を初期化します。

### java.lang.Object クラスから継承されたメソッド

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## finalizeCIWFactory

クラス名：CIWAdmin

### 構文

```
public static void finalizeCIWFactory()
```

### 機能

CSCIW の終了処理をします。

### 引数

ありません。

### 注意事項

- このメソッドは、プロセスで一度だけ呼び出してください。複数回呼び出した場合の動作は保証しません。



- このメソッドを呼び出したあとに、CSCIW のすべての API を呼び出さないでください。呼び出した場合の動作は保証しません。

## initializeCIWFactory

クラス名：CIWAdmin

### 構文

```
public static void initializeCIWFactory(
    javax.sql.DataSource aDataSource
)
    throws CIWFatalException
```

### 機能

CSCIW を初期化します。

### 引数

initializeCIWFactory の引数を次の表に示します。

表 4-2 initializeCIWFactory の引数

仮引数名	名称	in/out	説明
aDataSource	データソースオブジェクト	in	<p>CSCIW のワーク管理データベースにアクセスできるデータソースオブジェクトを指定します。</p> <p>データソースオブジェクトに指定できるのは、引数なしの getConnection() メソッドを実行したときに、CSCIW のワーク管理データベースにアクセスするための java.sql.Connection オブジェクトを取得できるオブジェクトです。</p> <p>指定されたデータソースを使用したトランザクションは、CSCIW が管理します。そのため、java.sql.Connection のコミットおよびロールバックメソッドを使用できる必要があります。</p> <p>グローバルトランザクションは使用できません。</p> <p>null は指定できません。</p>

### メモ

データソース実装クラスとして、oracle.jdbc.pool.OracleDataSource クラスを使用する場合は、次に示す 3 項目が設定されていれば問題ありません。

- JDBC の URL  
JDBC の URL についての詳細は使用する DBMS のマニュアルを参照してください。

- 接続ユーザ名
- 接続パスワード

データソース実装クラスとして、JP.co.Hitachi.soft.HiRDB.JDBC.PrdbDataSource クラスを使用する場合は、次に示す 4 項目が設定されていれば問題ありません。

- 接続付加情報
- ホスト名
- 認可識別子
- 接続パスワード

データソース実装クラスとして、com.microsoft.sqlserver.jdbc.SQLServerDataSource クラスを使用する場合は、次に示す 5 項目が設定されていれば問題ありません。

- ホスト名
- ポート番号
- ユーザ名
- パスワード
- データベース名

データソース実装クラスとして、org.postgresql.ds.PGSimpleDataSource クラスを使用する場合は、次に示す 5 項目が設定されていれば問題ありません。

- ホスト名
- ポート番号
- ユーザ名
- パスワード
- データベース名

## 例外

initializeCIWFactory で発生する例外を次の表に示します。

表 4-3 initializeCIWFactory で発生する例外

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合

## 注意事項

- このメソッドは、プロセスで一度だけ呼び出してください。複数回呼び出した場合の動作は保証しません。
- データソースオブジェクトに null を指定した場合は、例外 (`java.lang.IllegalArgumentException`) が発生します。

# 5

## 案件実行・運用クラスおよびインタフェースが提供するメソッド

この章では、案件実行・運用クラスおよびインタフェースが提供するメソッドについて説明します。

## 5.1 記述形式

---

### 5.1.1 案件実行・運用クラスおよびインタフェースの記述形式

案件実行・運用クラスおよびインタフェースは、次に示す形式を基本として説明しています。

#### クラス定義

クラス定義を示します。

#### 入れ子のクラス

インタフェースの入れ子のクラスを説明します。

#### メソッド

メソッドを説明します。

### 5.1.2 案件実行・運用クラスおよびインタフェースが提供するメソッドの記述形式

案件実行・運用クラスおよびインタフェースが提供するメソッドは、次に示す形式を基本として説明しています。

#### メソッド名

メソッド名を示します。

#### クラス名またはインタフェース名

メソッドが所属するクラス名またはインタフェース名を示します。

#### 機能

メソッドの機能を説明します。

#### 引数

メソッドの引数を説明します。

#### 戻り値

メソッドが返す戻り値を示します。

#### 例外

メソッドに発生する例外を説明します。

#### 注意事項

メソッドの注意事項を説明します。

## 5.2 CIWActivityDefinition (業務ステップ定義のインタフェース)

業務ステップ定義のインタフェースです。

業務ステップ定義オブジェクトに対してメソッドを発行することで、対象となる業務ステップ定義オブジェクトに対して処理が行われます。

### クラス定義

```
public interface CIWActivityDefinition
```

### 入れ子のクラス

CIWActivityDefinition インタフェースの入れ子のクラスを次の表に示します。

表 5-1 CIWActivityDefinition インタフェースの入れ子のクラス

クラス名	説明および記述形式
CIWActivityDefinition.AttributeName	業務ステップ定義属性の属性名の列挙型です。
	static class CIWActivityDefinition.AttributeName

### メソッド

CIWActivityDefinition インタフェースのメソッドを次の表に示します。

表 5-2 CIWActivityDefinition インタフェースのメソッド

API 名	説明
getDescription	対象となる業務ステップ定義の説明を取得します。
getID	対象となる業務ステップ定義の ID を取得します。
getName	対象となる業務ステップ定義の名称を取得します。
getProcessDefinitionID	対象となる業務ステップ定義が所属するビジネスプロセス定義の ID を取得します。

## getDescription

インタフェース名：CIWActivityDefinition

### 構文

```
java.lang.String getDescription()  
throws CIWFatalException
```

### 機能

対象となる業務ステップ定義の説明を取得します。

## 引数

ありません。

## 戻り値

業務ステップ定義の説明を返します。

## 例外

getDescription で発生する例外を次の表に示します。

表 5-3 getDescription で発生する例外 (CIWActivityDefinition の場合)

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合

## 注意事項

- 未設定時には null が返されます。
- オブジェクト取得時に名称の取得を指定していない場合は、例外 (CIWFatalException) が発生します。

## getID

インタフェース名: CIWActivityDefinition

## 構文

```
java.lang.Integer getID()  
throws CIWFatalException
```

## 機能

対象となる業務ステップ定義の ID を取得します。

## 引数

ありません。

## 戻り値

業務ステップ定義の ID を返します。

## 例外

getID で発生する例外を次の表に示します。

表 5-4 getID で発生する例外 (CIWActivityDefinition の場合)

発生する例外	説明
<a href="#">CIWFatalException</a>	処理を続行できない障害が発生した場合

## getName

インタフェース名 : CIWActivityDefinition

### 構文

```
java.lang.String getName()  
    throws CIWFatalException
```

### 機能

対象となる業務ステップ定義の名称を取得します。

### 引数

ありません。

### 戻り値

業務ステップ定義の名称を返します。

### 例外

getName で発生する例外を次の表に示します。

表 5-5 getName で発生する例外 (CIWActivityDefinition の場合)

発生する例外	説明
<a href="#">CIWFatalException</a>	処理を続行できない障害が発生した場合

### 注意事項

オブジェクト取得時に名称の取得を指定していない場合は、例外 (CIWFatalException) が発生します。

## getProcessDefinitionID

インタフェース名 : CIWActivityDefinition



## 構文

```
java.lang.Integer getProcessDefinitionID()  
    throws CIWFatalException
```

## 機能

対象となる業務ステップ定義が所属するビジネスプロセス定義の ID を取得します。

## 引数

ありません。

## 戻り値

業務ステップ定義が所属するビジネスプロセス定義の ID を返します。

表 5-6 getProcessDefinitionID で発生する例外 (CIWActivityDefinition の場合)

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合

## 5.3 CIWActivityInstance (業務ステップのインタフェース)

業務ステップのインタフェースです。

業務ステップオブジェクトに対してメソッドを発行することで、対象となる業務ステップオブジェクトに対して処理が行われます。

### クラス定義

```
public interface CIWActivityInstance
```

### 入れ子のクラス

CIWActivityInstance インタフェースの入れ子のクラスを次の表に示します。

表 5-7 CIWActivityInstance インタフェースの入れ子のクラス

クラス名	説明および記述形式
CIWActivityInstance.AttributeName	業務ステップ属性の属性名の列挙型です。 <code>static class CIWActivityInstance.AttributeName</code>
CIWActivityInstance.State	業務ステップの状態の列挙型です。 <code>static class CIWActivityInstance.State</code>
CIWActivityInstance.Type	業務ステップの種類列挙型です。 <code>static class CIWActivityInstance.Type</code>

### メソッド

CIWActivityInstance インタフェースのメソッドを次の表に示します。

表 5-8 CIWActivityInstance インタフェースのメソッド

メソッド名	説明
adhocMakeTransition	対象となる業務ステップから指定した業務ステップに遷移します。
changeState	対象となる業務ステップの状態を変更します。
evaluate	対象となる業務ステップに対して、業務ステップに含まれる作業の開始条件、およびその業務ステップ自身の完了条件を評価します。
getActivityDefinitionID	対象となる業務ステップ定義の ID を取得します。
getActivityDefinitionName	対象となる業務ステップ定義の名称を取得します。
getClosedDate	対象となる業務ステップの終了日時を取得します。
getDeadline	対象となる業務ステップの処理期限の絶対日時を取得します。
getID	対象となる業務ステップの ID を取得します。
getName	対象となる業務ステップの名称 (業務ステップキー) を取得します。
getPriority	対象となる業務ステップの優先度を取得します。

メソッド名	説明
<code>getProcessDefinitionID</code>	対象となる業務ステップの所属するビジネスプロセス定義の ID を取得します。
<code>getProcessInstanceID</code>	対象となる業務ステップが所属する案件の ID を取得します。
<code>getProcessInstanceName</code>	対象となる業務ステップの案件名 (案件キー) を取得します。
<code>getStartDate</code>	対象となる業務ステップの開始日時を取得します。
<code>getState</code>	対象となる業務ステップの状態を取得します。
<code>getType</code>	対象となる業務ステップの種類を取得します。
<code>makeBackwardTransition</code>	対象となる業務ステップに対して、指定した業務ステップ定義に差し戻しまたは引き戻しをします。
<code>setDeadline</code>	対象となる業務ステップに対して、指定した日時の値を業務ステップの処理期限の絶対日時として設定します。
<code>setPriority</code>	対象となる業務ステップの優先度を変更します。

## adhocMakeTransition

インタフェース名：CIWActivityInstance

### 構文

```
void adhocMakeTransition(
    java.lang.Integer aActivityInstanceID
)
    throws CIWFatalException,
           CIWTransientException,
           CIWTransitionFailedException,
           CIWStateException,
           CIWEntityNotExistException
```

### 機能

対象となる業務ステップから指定した業務ステップに遷移します。

この操作によって、対象となる業務ステップの状態は「遷移済」に、遷移先の業務ステップの状態は「実行中」になります。

### 引数

`adhocMakeTransition` の引数を次の表に示します。

表 5-9 adhocMakeTransition の引数 (CIWActivityInstance の場合)

仮引数名	名称	in/out	説明
aActivityInstanceID	遷移先の業務ステップの ID	in	遷移先の業務ステップの ID を指定します。遷移先には「初期」状態の業務ステップの ID を指定する必要があります。

## 戻り値

ありません。

## 例外

adhocMakeTransition で発生する例外を次の表に示します。

表 5-10 adhocMakeTransition で発生する例外 (CIWActivityInstance の場合)

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合
CIWTransientException	一時的なエラーが発生した場合
CIWTransitionFailedException	案件処理中にエラーが発生した場合
CIWStateException	状態や属性の変更に失敗した場合
CIWEntityNotExistException	処理しようとしたオブジェクトが存在しない場合

## 注意事項

- 対象となる業務ステップが所属する案件は、「実行中」または「実行停止」状態である必要があります。そのほかの状態では、例外 (CIWStateException) が発生します。
- 対象となる業務ステップの状態は、「実行中」または「遷移可」である必要があります。そのほかの状態では、例外 (CIWStateException) が発生します。
- 遷移先の業務ステップが存在しない場合、例外 (CIWEntityNotExistException) が発生します。
- 遷移先の業務ステップの状態が「初期」の場合に遷移できます。そのほかの状態では、例外 (CIWStateException) が発生します。
- 遷移先の業務ステップの ID の指定値が null の場合は、例外 (java.lang.IllegalArgumentException) が発生します。
- 対象となる業務ステップが所属する案件のビジネスプロセス定義のプロパティ「アドホック API を使用する」にチェックが入っていない場合は、例外 (CIWFatalException) が発生します。

## changeState

インタフェース名：CIWActivityInstance

## 構文

```
void changeState(  
    CIWActivityInstance.State aNewState  
)  
    throws CIWTransientException,  
           CIWTransitionFailedException,  
           CIWFatalException,  
           CIWEntityNotExistException,  
           CIWStateException
```

## 機能

対象となる業務ステップの状態を変更します。

## 引数

changeState の引数を次の表に示します。

表 5-11 changeState の引数 (CIWActivityInstance の場合)

仮引数名	名称	in/out	説明
aNewState	業務ステップの状態	in	変更する業務ステップの状態を指定します。 指定する値として、このインタフェースで定義された CIWActivityInstance.State 列挙型の定数を指定できます。null は指定できません。

aNewState で指定できる業務ステップの状態を次の表に示します。

表 5-12 aNewState で指定できる業務ステップの状態 (CIWActivityInstance の場合)

現在の状態		指定できる状態		状態遷移の意味
定数	意味	定数	意味	
INITIAL	初期	READY	実行開始可能	活性化
RUNNING	実行中	INTERMITTED	実行停止	中断
INTERMITTED	実行停止	RUNNING	実行中	再開
—	未終了	TERMINATED	強制終了	強制終了
RUNNING	実行中	READY_FOR_TRANSITION	遷移可	業務ステップ終了

(凡例)

定数：このインタフェースに定義された CIWActivityInstance.State 列挙型の定数を表します。

状態遷移については、「付録 C 状態遷移モデル」を参照してください。

—：定数がないことを表します。

## 戻り値

ありません。

## 例外

changeState で発生する例外を次の表に示します。

表 5-13 changeState で発生する例外 (CIWActivityInstance の場合)

発生する例外	説明
CIWTransientException	一時的なエラーが発生した場合
CIWTransitionFailedException	案件処理中にエラーが発生した場合
CIWFatalException	処理を続行できない障害が発生した場合
CIWEntityNotExistException	処理しようとしたオブジェクトが存在しない場合
CIWStateException	状態の遷移に失敗した場合

## 注意事項

- 業務ステップを「実行開始可能」状態へ遷移させる場合は、対象となる業務ステップが所属する案件は、「実行中」状態または「実行停止」状態である必要があります。そのほかの状態では、例外 (CIWStateException) が発生します。
- 対象となる業務ステップが所属する案件がエラー案件の場合は、例外 (CIWStateException) が発生します。
- 指定した状態が現在の状態と同じ場合は、例外 (CIWStateException) が発生します。
- 変更する業務ステップの状態に null を指定した場合は、例外 (java.lang.IllegalArgumentException) が発生します。
- 業務ステップを強制終了した場合は、次の業務ステップへは遷移しません。そのため、ほかに「実行中」状態の業務ステップが存在しない場合は、案件の強制終了以外の操作ができなくなるので注意してください。また、実行中の業務ステップが存在した場合でも、待合ノードに続く業務ステップを強制終了すると、待合ノードから遷移しなくなります。注意してください。

## evaluate

インタフェース名 : CIWActivityInstance

## 構文

```
void evaluate()  
    throws CIWTransientException,  
           CIWTransitionFailedException,  
           CIWFatalException,
```

CIWEntityNotExistException,  
CIWStateException

## 機能

対象となる業務ステップに対して、業務ステップに含まれる作業の開始条件、およびその業務ステップ自身の完了条件を評価します。

## 引数

ありません。

## 戻り値

ありません。

## 例外

evaluate で発生する例外を次の表に示します。

表 5-14 evaluate で発生する例外 (CIWActivityInstance の場合)

発生する例外	説明
CIWTransientException	一時的なエラーが発生した場合
CIWTransitionFailedException	案件処理中にエラーが発生した場合
CIWFatalException	処理を続行できない障害が発生した場合
CIWEntityNotExistException	処理しようとしたオブジェクトが存在しない場合
CIWStateException	状態の遷移に失敗した場合

## 注意事項

- 対象となる業務ステップは、「実行中」状態である必要があります。そのほかの状態では、例外 (CIWStateException) が発生します。
- 対象となる業務ステップが所属する案件は、「実行中」状態または「実行停止」状態である必要があります。そのほかの状態では、例外 (CIWStateException) が発生します。

## getActivityDefinitionID

インタフェース名 : CIWActivityInstance

## 構文

```
java.lang.Integer getActivityDefinitionID()  
throws CIWFatalException
```

## 機能

対象となる業務ステップ定義の ID を取得します。

## 引数

ありません。

## 戻り値

業務ステップ定義の ID を返します。

## 例外

getActivityDefinitionID で発生する例外を次の表に示します。

表 5-15 getActivityDefinitionID で発生する例外 (CIWActivityInstance の場合)

発生する例外	説明
<a href="#">CIWFatalException</a>	処理を続行できない障害が発生した場合

## 注意事項

- アドホック生成された業務ステップまたはエラー回復した業務ステップの場合は、null が返されます。
- オブジェクト取得時に業務ステップ定義の ID の取得を指定していない場合は、例外 (CIWFatalException) が発生します。

## getActivityDefinitionName

インタフェース名 : CIWActivityInstance

## 構文

```
java.lang.String getActivityDefinitionName()  
    throws CIWFatalException
```

## 機能

対象となる業務ステップ定義の名称を取得します。

## 引数

ありません。

## 戻り値

業務ステップ定義の名称を返します。



## 例外

getActivityDefinitionName で発生する例外を次の表に示します。

表 5-16 getActivityDefinitionName で発生する例外

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合

## 注意事項

オブジェクト取得時に業務ステップ定義の名称の取得を指定していない場合は、例外 (CIWFatalException) が発生します。

## getClosedDate

インタフェース名：CIWActivityInstance

## 構文

```
java.util.Date getClosedDate()  
throws CIWFatalException
```

## 機能

対象となる業務ステップの終了日時を取得します。

## 引数

ありません。

## 戻り値

業務ステップの終了日時を返します。

## 例外

getClosedDate で発生する例外を次の表に示します。

表 5-17 getClosedDate で発生する例外 (CIWActivityInstance の場合)

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合

## 注意事項

- 「終了」状態への遷移が発生した場合に値が設定されます。

遷移の発生前は null が返されます。状態遷移については、「付録 C 状態遷移モデル」を参照してください。

- オブジェクト取得時に終了日時の取得を指定していない場合は、例外 (CIWFatalException) が発生します。

## getDeadline

インタフェース名 : CIWActivityInstance

### 構文

```
java.util.Date getDeadline()  
    throws CIWFatalException
```

### 機能

対象となる業務ステップの処理期限の絶対日時を取得します。

### 引数

ありません。

### 戻り値

業務ステップの処理期限の絶対日時を返します。

### 例外

getDeadline で発生する例外を次の表に示します。

表 5-18 getDeadline で発生する例外 (CIWActivityInstance の場合)

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合

### 注意事項

- 未設定時には null が返されます。
- オブジェクト取得時に処理期限の絶対日時の取得を指定していない場合は、例外 (CIWFatalException) が発生します。

## getID

インタフェース名 : CIWActivityInstance

## 構文

```
java.lang.Integer getID()  
    throws CIWFatalException
```

## 機能

対象となる業務ステップの ID を取得します。

## 引数

ありません。

## 戻り値

業務ステップの ID を返します。

## 例外

getID で発生する例外を次の表に示します。

表 5-19 getID で発生する例外 (CIWActivityInstance の場合)

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合

## getName

インタフェース名：CIWActivityInstance

## 構文

```
java.lang.String getName()  
    throws CIWFatalException
```

## 機能

対象となる業務ステップの名称（業務ステップキー）を取得します。

## 引数

ありません。

## 戻り値

業務ステップの名称（業務ステップキー）を返します。

## 例外

getName で発生する例外を次の表に示します。

表 5-20 getName で発生する例外 (CIWActivityInstance の場合)

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合

## 注意事項

- 未設定時には null が返されます。
- オブジェクト取得時に名称（業務ステップキー）の取得を指定していない場合は、例外 (CIWFatalException) が発生します。

## getPriority

インタフェース名：CIWActivityInstance

## 構文

```
java.lang.Integer getPriority()  
    throws CIWFatalException
```

## 機能

対象となる業務ステップの優先度を取得します。

## 引数

ありません。

## 戻り値

業務ステップの優先度を返します。

## 例外

getPriority で発生する例外を次の表に示します。

表 5-21 getPriority で発生する例外 (CIWActivityInstance の場合)

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合

## 注意事項

- 未設定時には null が返されます。
- オブジェクト取得時に優先度の取得を指定していない場合は、例外 (CIWFatalException) が発生します。

## getProcessDefinitionID

インタフェース名：CIWActivityInstance

### 構文

```
java.lang.Integer getProcessDefinitionID()  
    throws CIWFatalException
```

### 機能

対象となる業務ステップの所属するビジネスプロセス定義の ID を取得します。

### 引数

ありません。

### 戻り値

業務ステップの所属するビジネスプロセス定義の ID を返します。

### 例外

getProcessDefinitionID で発生する例外を次の表に示します。

表 5-22 getProcessDefinitionID で発生する例外 (CIWActivityInstance の場合)

発生する例外	説明
<a href="#">CIWFatalException</a>	処理を続行できない障害が発生した場合

## 注意事項

オブジェクト取得時に業務ステップの所属するビジネスプロセス定義の ID の取得を指定していない場合は、例外 (CIWFatalException) が発生します。

## getProcessInstanceID

インタフェース名：CIWActivityInstance

## 構文

```
java.lang.Integer getProcessInstanceID()  
throws CIWFatalException
```

## 機能

対象となる業務ステップが所属する案件の ID を取得します。

## 引数

ありません。

## 戻り値

業務ステップが所属する案件の ID を返します。

## 例外

getProcessInstanceID で発生する例外を次の表に示します。

表 5-23 getProcessInstanceID で発生する例外 (CIWActivityInstance の場合)

発生する例外	説明
<code>CIWFatalException</code>	処理を続行できない障害が発生した場合

## getProcessInstanceName

インタフェース名 : CIWActivityInstance

## 構文

```
java.lang.String getProcessInstanceName()  
throws CIWFatalException
```

## 機能

対象となる業務ステップの案件名 (案件キー) を取得します。

## 引数

ありません。

## 戻り値

業務ステップの案件名 (案件キー) を返します。

## 例外

getProcessInstanceName で発生する例外を次の表に示します。

表 5-24 getProcessInstanceName で発生する例外 (CIWActivityInstance の場合)

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合

## 注意事項

- 未設定時には null が返されます。
- オブジェクト取得時に案件名 (案件キー) の取得を指定していない場合は、例外 (CIWFatalException) が発生します。

## getStartDate

インタフェース名 : CIWActivityInstance

## 構文

```
java.util.Date getStartDate()  
    throws CIWFatalException
```

## 機能

対象となる業務ステップの開始日時を取得します。

## 引数

ありません。

## 戻り値

業務ステップの開始日時を返します。

## 例外

getStartDate で発生する例外を次の表に示します。

表 5-25 getStartDate で発生する例外 (CIWActivityInstance の場合)

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合

## 注意事項

- 対象となる業務ステップに「実行中」状態への遷移が発生した場合に、値が設定されます。「実行中」状態への遷移が発生していない場合は、null が返されます。状態遷移については、「付録 C 状態遷移モデル」を参照してください。
- オブジェクト取得時に開始日時の取得を指定していない場合は、例外 (CIWFatalException) が発生します。

## getState

インタフェース名 : CIWActivityInstance

### 構文

```
CIWActivityInstance.State getState()  
throws CIWFatalException
```

### 機能

対象となる業務ステップの状態を取得します。

### 引数

ありません。

### 戻り値

業務ステップの状態を返します。

### 例外

getState で発生する例外を次の表に示します。

表 5-26 getState で発生する例外 (CIWActivityInstance の場合)

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合

## 注意事項

オブジェクト取得時に状態の取得を指定していない場合は、例外 (CIWFatalException) が発生します。



# getType

インタフェース名：CIWActivityInstance

## 構文

```
CIWActivityInstance.Type getType()  
    throws CIWFatalException
```

## 機能

対象となる業務ステップの種類を取得します。

## 引数

ありません。

## 戻り値

業務ステップの種類を返します。

## 例外

getActivityDefinitionName で発生する例外を次の表に示します。

表 5-27 getActivityDefinitionName で発生する例外 (CIWActivityInstance の場合)

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合

## 注意事項

オブジェクト取得時に種類の取得を指定していない場合は、例外 (CIWFatalException) が発生します。

# makeBackwardTransition

インタフェース名：CIWActivityInstance

## 構文

```
void makeBackwardTransition(  
    java.lang.String aActivityDefinitionName  
)  
    throws CIWFatalException,  
           CIWTransientException,  
           CIWTransitionFailedException,
```

## 機能

対象となる業務ステップに対して、指定した業務ステップ定義に差し戻しまたは引き戻しをします。

差し戻し先には初期状態または終了状態の業務ステップが存在する業務ステップ定義を指定する必要があります。

## 引数

makeBackwardTransition の引数を次の表に示します。

表 5-28 makeBackwardTransition の引数

仮引数名	名称	in/out	説明
aActivityDefinitionName	業務ステップ定義名	in	差し戻し先または引き戻し先の業務ステップ定義名を指定します。 null は指定できません。

## 戻り値

ありません。

## 例外

makeBackwardTransition で発生する例外を次の表に示します。

表 5-29 makeBackwardTransition で発生する例外

発生する例外	説明
CIWTransientException	一時的なエラーが発生した場合
CIWStateException	状態や属性の変更に失敗した場合
CIWTransitionFailedException	案件処理中にエラーが発生した場合
CIWFatalException	処理を続行できない障害が発生した場合
CIWEntityNotExistException	処理しようとしたオブジェクトが存在しない場合

## 注意事項

- 差し戻し先または引き戻し先に指定した業務ステップ定義が存在しない場合、例外 (CIWEntityNotExistException) が発生します。
- 差し戻し先または引き戻し先に指定した業務ステップ定義に業務ステップが存在しない場合、例外 (CIWStateException) が発生します。

- 差し戻し先または引き戻し先に指定した業務ステップ定義に「初期」状態または「終了」状態のどちらでもない状態の業務ステップが存在する場合、例外 (CIWStateException) が発生します。
- 差し戻し先または引き戻し先の業務ステップ定義名に null を指定した場合、例外 (java.lang.IllegalArgumentException) が発生します。
- 対象となる業務ステップの状態が「初期」または「終了」のどちらかの場合、例外 (CIWStateException) が発生します。
- 対象となる業務ステップを含む案件の状態が「実行中」または「実行停止」のどちらでもない場合は、例外 (CIWStateException) が発生します。

## setDeadline

インタフェース名：CIWActivityInstance

### 構文

```
void setDeadline(
    java.util.Date aNewDeadline
)
    throws CIWTransientException,
           CIWStateException,
           CIWFatalException,
           CIWEntityNotExistException
```

### 機能

対象となる業務ステップに対して、指定した日時の値を業務ステップの処理期限の絶対日時として設定します。

### 引数

setDeadline の引数を次の表に示します。

表 5-30 setDeadline の引数 (CIWActivityInstance の場合)

仮引数名	名称	in/out	説明
aNewDeadline	変更後の処理期限	in	変更後の処理期限を指定します。 null を指定した場合は、処理期限は未設定となります。処理期限の絶対日時として指定できる値は、CIWUtil#convertDateToString(Date)で指定できる Date 値と同じです。 なお、処理期限はミリ秒以下を切り捨てて設定します。

### 戻り値

ありません。

## 例外

setDeadline で発生する例外を次の表に示します。

表 5-31 setDeadline で発生する例外 (CIWActivityInstance の場合)

発生する例外	説明
CIWTransientException	一時的なエラーが発生した場合
CIWStateException	状態の遷移に失敗した場合
CIWFatalException	処理を続行できない障害が発生した場合
CIWEntityNotExistException	処理しようとしたオブジェクトが存在しない場合

## 注意事項

業務ステップが「未終了」状態のときだけ変更できます。ただし、業務ステップの処理期限を変更しても、対象となる業務ステップに所属する、変更以前に生成されていた作業の処理期限は変更されません。

## setPriority

インタフェース名：CIWActivityInstance

## 構文

```
void setPriority(  
    java.lang.Integer aNewPriority  
)  
    throws CIWTransientException,  
           CIWStateException,  
           CIWFatalException,  
           CIWEntityNotExistException
```

## 機能

対象となる業務ステップの優先度を変更します。

## 引数

setPriority の引数を次の表に示します。

表 5-32 setPriority の引数 (CIWActivityInstance の場合)

仮引数名	名称	in/out	説明
aNewPriority	変更後の優先度	in	変更後の優先度を指定します。 null を指定した場合は、優先度は未設定となります。

## 戻り値

ありません。

## 例外

setPriority で発生する例外を次の表に示します。

表 5-33 setPriority で発生する例外

発生する例外	説明
<a href="#">CIWTransientException</a>	一時的なエラーが発生した場合
<a href="#">CIWStateException</a>	状態の遷移に失敗した場合
<a href="#">CIWFatalException</a>	処理を続行できない障害が発生した場合
<a href="#">CIWEntityNotExistException</a>	処理しようとしたオブジェクトが存在しない場合

## 注意事項

業務ステップが「未終了」状態のときだけ変更できます。ただし、業務ステップの優先度を変更しても、対象となる業務ステップに所属する、変更以前に生成されていた作業の優先度は変更されません。

## 5.4 CIWFactory (CIWFactory と CIWServer オブジェクトの生成クラス)

CIWFactory は、CIWServer オブジェクトのファクトリクラスです。

このクラスで次に示す処理をします。

- ファクトリオブジェクトの取得
- CIWServer オブジェクトの生成

### クラス定義

```
public final class CIWFactory extends java.lang.Object
```

### メソッド

CIWFactory クラスのメソッドを次の表に示します。

表 5-34 CIWFactory クラスのメソッド

メソッド	説明
<code>createCIWServer</code>	CIWServer オブジェクトを生成します。
<code>createCIWServer</code>	<code>javax.sql.DataSource</code> オブジェクトを設定して、CIWServer オブジェクトを生成します。
<code>getInstance</code>	ファクトリオブジェクトを取得します。

### java.lang.Object クラスから継承されたメソッド

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## createCIWServer

クラス名：CIWFactory

### 形式

```
public CIWServer createCIWServer(  
    java.lang.String aUserDescription  
)  
    throws CIWFatalException
```

### 機能

CIWServer オブジェクトを生成します。

このメソッドで生成した CIWServer オブジェクトについては、トランザクションの開始しているデータベース接続との関連づけおよび関連づけ解除を、ユーザ自身が行う必要があります。この場合、

CIWServer#getConnectionCoordinator()を実行し、コネクション関連づけオブジェクトを取得し、データベースコネクションと CIWServer オブジェクトとの関連づけおよび関連づけ解除を行ってください。

## 引数

createCIWServer の引数を次の表に示します。

表 5-35 createCIWServer の引数

仮引数名	名称	in/out	説明
aUserDescription	ユーザ記述子	in	ユーザ記述子を<SYSTEMID>_PROCESS_INSTANCE テーブルの Creator カラムのバイト数以下で指定します。空文字列および null は指定できません。

## 戻り値

CIWServer オブジェクトを返します。

## 例外

createCIWServer で発生する例外を次の表に示します。

表 5-36 createCIWServer で発生する例外

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合

## 注意事項

ユーザ記述子に次に示す文字列を指定した場合、例外 (java.lang.IllegalArgumentException) が発生します。

- ユーザ記述子に<SYSTEMID>\_PROCESS\_INSTANCE テーブルの Creator カラムのバイト数を超える文字列
- 空文字列
- null

## createCIWServer

クラス名：CIWFactory

## 形式

```
public CIWServer createCIWServer(  
    java.lang.String aUserDescription,
```

```
    javax.sql.DataSource aDataSource  
)  
    throws CIWFatalException
```

## 機能

javax.sql.DataSource オブジェクトを設定して、CIWServer オブジェクトを生成します。

このメソッドで生成した CIWServer オブジェクトについては、データベースコネクションと CIWServer オブジェクトの関連づけおよび解除は自動的に行われます。このメソッドで生成した CIWServer オブジェクトに対して、CIWServer#getConnectionCoordinator() を実行した場合は (CIWFatalException) が発生します。

## 引数

createCIWServer の引数を次の表に示します。

表 5-37 createCIWServer の引数

仮引数名	名称	in/out	説明
aUserDescription	ユーザ記述子	in	ユーザ記述子を<SYSTEMID>_PROCESS_INSTANCE テーブルの Creator カラムのバイト数以下で指定します。空文字列および null は指定できません。
aDataSource	データソースオブジェクト	in	ユーザアプリケーションで使用しているデータソースオブジェクトを指定します。null は指定できません。

## 戻り値

CIWServer オブジェクトを返します。

## 例外

createCIWServer で発生する例外を次の表に示します。

表 5-38 createCIWServer で発生する例外

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合

## 注意事項

- ユーザ記述子に次に示す文字列を指定した場合、例外 (java.lang.IllegalArgumentException) が発生します。
  - ユーザ記述子に<SYSTEMID>\_PROCESS\_INSTANCE テーブルの Creator カラムのバイト数を超える文字列



- 空文字列
- null
- ユーザアプリケーションで正在している DataSource オブジェクトに null を指定した場合は、例外 (java.lang.IllegalArgumentException) が発生します。
- DataSource から取得した java.sql.Connection が、実行スレッドで同一のトランザクションで DBMS にアクセスできる場合に、このメソッドを呼び出すことができます。例えば、J2EE サーバ上の EJB から利用する場合などです。

## getInstance

クラス名：CIWFactory

### 構文

```
public static CIWFactory getInstance()
    throws CIWFatalException
```

### 機能

ファクトリオブジェクトを取得します。

### 引数

ありません。

### 戻り値

ファクトリオブジェクトを返します。

### 例外

getInstance で発生する例外を次の表に示します。

表 5-39 getInstance で発生する例外

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合

### 注意事項

initializeCIWFactory メソッドを実行する前に、このメソッドを実行した場合、例外 (CIWFatalException) が発生します。

## 5.5 CIWProcessDefinition (ビジネスプロセス定義のインタフェース)

ビジネスプロセス定義のインタフェースです。

ビジネスプロセス定義オブジェクトに対してメソッドを発行することで、対象となるビジネスプロセス定義オブジェクトに対して処理が行われます。

### クラス定義

```
public interface CIWProcessDefinition
```

### 入れ子のクラス

CIWProcessDefinition インタフェースの入れ子のクラスを次の表に示します。

表 5-40 CIWProcessDefinition インタフェースの入れ子のクラス

クラス名	入れ子のクラスの説明および入れ子のクラスの記述形式
CIWProcessDefinition.AttributeName	ビジネスプロセス定義属性の属性名の列挙型です。
	<code>static class CIWProcessDefinition.AttributeName</code>
CIWProcessDefinition.State	ビジネスプロセス定義の状態の列挙型です。
	<code>static class CIWProcessDefinition.State</code>

### メソッド

CIWProcessDefinition インタフェースのメソッドを次の表に示します。

表 5-41 CIWProcessDefinition インタフェースのメソッド

メソッド名	説明
<code>getAuthor</code>	対象となるビジネスプロセス定義の作成者を取得します。
<code>getCreationDate</code>	対象となるビジネスプロセス定義の作成日を取得します。
<code>getDescription</code>	対象となるビジネスプロセス定義の説明を取得します。
<code>getID</code>	対象となるビジネスプロセス定義の ID を取得します。
<code>getName</code>	対象となるビジネスプロセス定義の名称を取得します。
<code>getProcessInstancesList</code>	対象となるビジネスプロセス定義内で、指定したフィルター条件を満たす案件オブジェクトのリストを取得します。
<code>getResponsible</code>	対象となるビジネスプロセス定義の管理者 ID を取得します。
<code>getState</code>	対象となるビジネスプロセス定義の状態を取得します。
<code>getValidFromDate</code>	対象となるビジネスプロセス定義の有効となる日を取得します。
<code>getValidToDate</code>	対象となるビジネスプロセス定義の無効となる日を取得します。
<code>getVersion</code>	対象となるビジネスプロセス定義のバージョンを取得します。

# getAuthor

インタフェース名：CIWProcessDefinition

## 構文

```
java.lang.String getAuthor()  
    throws CIWFatalException
```

## 機能

対象となるビジネスプロセス定義の作成者を取得します。

## 引数

ありません。

## 戻り値

ビジネスプロセス定義の作成者を返します。

## 例外

getAuthor で発生する例外を次の表に示します。

表 5-42 getAuthor で発生する例外

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合

## 注意事項

オブジェクト取得時に作成者の取得を指定していない場合は、例外（CIWFatalException）が発生します。

# getCreationDate

インタフェース名：CIWProcessDefinition

## 構文

```
java.util.Date getCreationDate()  
    throws CIWFatalException
```

## 機能

対象となるビジネスプロセス定義の作成日を取得します。

## 引数

ありません。

## 戻り値

ビジネスプロセス定義の作成日を返します。

## 例外

getCreationDate で発生する例外を次の表に示します。

表 5-43 getCreationDate で発生する例外 (CIWProcessDefinition の場合)

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合

## 注意事項

オブジェクト取得時に作成日の取得を指定していない場合は、例外 (CIWFatalException) が発生します。

## getDescription

インタフェース名 : CIWProcessDefinition

## 構文

```
java.lang.String getDescription()  
    throws CIWFatalException
```

## 機能

対象となるビジネスプロセス定義の説明を取得します。

## 引数

ありません。

## 戻り値

ビジネスプロセス定義の説明を返します。

## 例外

getDescription で発生する例外を次の表に示します。

表 5-44 getDescription で発生する例外 (CIWProcessDefinition の場合)

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合

## 注意事項

- 未設定時には null が返されます。
- オブジェクト取得時に説明の取得を指定していない場合は、例外 (CIWFatalException) が発生します。

## getID

インタフェース名 : CIWProcessDefinition

## 構文

```
java.lang.Integer getID()
    throws CIWFatalException
```

## 機能

対象となるビジネスプロセス定義の ID を取得します。

## 引数

ありません。

## 戻り値

ビジネスプロセス定義の ID を返します。

## 例外

getID で発生する例外を次の表に示します。

表 5-45 getID で発生する例外 (CIWProcessDefinition の場合)

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合

## getName

インタフェース名 : CIWProcessDefinition

## 構文

```
java.lang.String getName()  
    throws CIWFatalException
```

## 機能

対象となるビジネスプロセス定義の名称を取得します。

## 引数

ありません。

## 戻り値

ビジネスプロセス定義の名称を返します。

## 例外

getName で発生する例外を次の表に示します。

表 5-46 getName で発生する例外 (CIWProcessDefinition の場合)

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合

## 注意事項

オブジェクト取得時に名称の取得を指定していない場合は、例外 (CIWFatalException) が発生します。

## getProcessInstancesList

インタフェース名 : CIWProcessDefinition

## 構文

```
java.util.List<CIWProcessInstance> getProcessInstancesList(  
    java.lang.String aFilter,  
    java.lang.String aSort,  
    int aOffset,  
    int aMaxCount,  
    java.util.Set<CIWProcessInstance.AttributeName> aAttributeNames  
)  
    throws CIWFatalException,  
           CIWTransientException
```

## 機能

対象となるビジネスプロセス定義内で、指定したフィルター条件を満たす案件オブジェクトのリストを取得します。

## 引数

getProcessInstancesList の引数を次の表に示します。

表 5-47 getProcessInstancesList の引数 (CIWProcessDefinition の場合)

仮引数名	名称	in/out	説明
aFilter	フィルター条件	in	案件オブジェクトを取得する場合に、フィルター条件を指定します。 フィルター条件を指定しない場合は、null を指定します。 フィルター条件の指定方法については、「付録 B 参照系 API で指定できる属性」を参照してください。
aSort	ソート条件	in	案件オブジェクトを取得する場合に、ソート条件を指定します。 ソート条件を指定しない場合は、null を指定します。ソート条件の指定方法については、「付録 B 参照系 API で指定できる属性」を参照してください。
aOffset	オフセット	in	案件オブジェクトを取得する場合に、オフセットを指定します。 オフセットは先頭を 0 として指定します。0 未満の値は指定できません。
aMaxCount	最大取得数	in	案件オブジェクトを取得する場合に、最大取得数を指定します。 すべてを取得する場合は、-1 を指定します。-1 未満の値は指定できません。
aAttributeNames	属性名のセット	in	取得したい案件オブジェクトの属性名のセットを指定します。 取得属性名を指定しない場合は、サイズ 0 の属性名のセットまたは null を指定します。取得属性名に null は指定できません。取得属性名に指定されていない場合でも、案件の ID は必ず取得します。取得できる属性は、CIWProcessInstance.AttributeName 列挙型で指定できる属性です。

## 戻り値

取得した案件オブジェクトのリストを返します。

## 例外

getProcessInstancesList で発生する例外を次の表に示します。

表 5-48 getProcessInstancesList で発生する例外 (CIWProcessDefinition の場合)

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合
CIWTransientException	一時的なエラーが発生した場合

## 注意事項

- 指定したフィルター条件を満たす案件オブジェクトが存在しない場合、空のリストが返されます。
- 不正な引数を指定した場合は、例外 (java.lang.IllegalArgumentException) が発生します。
- オフセットに 0, 50, 100などを指定し、複数回に分けて取得した場合は、案件の重複や漏れが発生することがあります。

## getResponsible

インタフェース名：CIWProcessDefinition

### 構文

```
java.lang.String getResponsible()
    throws CIWFatalException
```

### 機能

対象となるビジネスプロセス定義の管理者 ID を取得します。

### 引数

ありません。

### 戻り値

ビジネスプロセス定義の管理者 ID を返します。

### 例外

getResponsible で発生する例外を次の表に示します。

表 5-49 getResponsible で発生する例外

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合



## 注意事項

オブジェクト取得時に管理者 ID の取得を指定していない場合は、例外（CIWFatalException）が発生します。

## getState

インタフェース名：CIWProcessDefinition

## 構文

```
CIWProcessDefinition.State getState()  
    throws CIWFatalException
```

## 機能

対象となるビジネスプロセス定義の状態を取得します。

## 引数

ありません。

## 戻り値

ビジネスプロセス定義の状態を返します。

## 例外

getState で発生する例外を次の表に示します。

表 5-50 getState で発生する例外（CIWProcessDefinition の場合）

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合

## 注意事項

オブジェクト取得時に状態の取得を指定していない場合は、例外（CIWFatalException）が発生します。

## getValidFromDate

インタフェース名：CIWProcessDefinition

## 構文

```
java.util.Date getValidFromDate()  
throws CIWFatalException
```

## 機能

対象となるビジネスプロセス定義の有効となる日を取得します。

## 引数

ありません。

## 戻り値

ビジネスプロセス定義の有効となる日を返します。

## 例外

getValidFromDate で発生する例外を次の表に示します。

表 5-51 getValidFromDate で発生する例外

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合

## 注意事項

- 未設定時は、無限遠の過去を表す日付情報を返します。無限遠の過去を表す日付情報については、「表 7-45 CIWUtil.InfinityDate の列挙型定数」の ORIGIN を参照してください。
- オブジェクト取得時に有効となる日の取得を指定していない場合は、例外 (CIWFatalException) が発生します。

## getValidToDate

インタフェース名：CIWProcessDefinition

## 構文

```
java.util.Date getValidToDate()  
throws CIWFatalException
```

## 機能

対象となるビジネスプロセス定義の無効となる日を取得します。

## 引数

ありません。

## 戻り値

ビジネスプロセス定義の無効となる日を返します。

## 例外

getValidToDate で発生する例外を次の表に示します。

表 5-52 getValidToDate で発生する例外

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合

## 注意事項

- 未設定時は、無限遠の未来を表す日付情報を返します。無限遠の未来を表す日付情報については、「表 7-45 CIWUtil.InfinityDate の列挙型定数」の BEYOND を参照してください。
- オブジェクト取得時に無効となる日の取得を指定していない場合は、例外 (CIWFatalException) が発生します。

## getVersion

インタフェース名：CIWProcessDefinition

## 構文

```
java.lang.Short getVersion()  
throws CIWFatalException
```

## 機能

対象となるビジネスプロセス定義のバージョンを取得します。

## 引数

ありません。

## 戻り値

ビジネスプロセス定義のバージョンを返します。

## 例外

getVersion で発生する例外を次の表に示します。

表 5-53 getVersion で発生する例外

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合

## 注意事項

オブジェクト取得時にバージョンの取得を指定していない場合は、例外（CIWFatalException）が発生します。

## 5.6 CIWProcessInstance (案件のインタフェース)

案件のインタフェースです。

案件オブジェクトにメソッドを発行することで、対象となる案件オブジェクトに対して処理が行われます。

### クラス定義

```
public interface CIWProcessInstance
```

### 入れ子のクラス

CIWProcessInstance インタフェースの入れ子のクラスを次の表に示します。

表 5-54 CIWProcessInstance インタフェースの入れ子のクラス

クラス名	説明および記述形式
CIWProcessInstance.AttributeName	案件属性の属性名の列挙型です。 <code>static class CIWProcessInstance.AttributeName</code>
CIWProcessInstance.State	案件の状態の列挙型です。 <code>static class CIWProcessInstance.State</code>

### メソッド

CIWProcessInstance インタフェースのメソッドを次の表に示します。

表 5-55 CIWProcessInstance インタフェースのメソッド

メソッド名	説明
<code>adhocInstantiateActivityInstance</code>	指定した業務ステップ定義に従って、対象となる案件内に、業務ステップを「初期」状態で生成します。
<code>changeState</code>	対象となる案件の状態を変更します。
<code>delete</code>	対象となる案件を削除します。
<code>getClosedDate</code>	対象となる案件の終了日時を取得します。
<code>getCreator</code>	対象となる案件の投入者を取得します。
<code>getDeadline</code>	対象となる案件の処理期限の絶対日時を取得します。
<code>getID</code>	対象となる案件の ID を取得します。
<code>getMovedDate</code>	対象となる案件の乗せ替え日時を取得します。
<code>getName</code>	対象となる案件の名称（案件キー）を取得します。
<code>getPriority</code>	対象となる案件の優先度を取得します。
<code>getProcessDefinition</code>	対象となる案件が所属するビジネスプロセス定義オブジェクトを取得します。
<code>getProcessDefinitionID</code>	対象となる案件が所属するビジネスプロセス定義の ID を取得します。

メソッド名	説明
<code>getProcessDefinitionName</code>	対象となる案件が所属するビジネスプロセス定義の名称を取得します。
<code>getStartDate</code>	対象となる案件の開始日時を取得します。
<code>getState</code>	対象となる案件の状態を取得します。
<code>getWorkItemsList</code>	対象となる案件内で、指定したフィルター条件を満たす作業オブジェクトのリストを取得します。
<code>setDeadline</code>	対象となる案件に対して、指定した日時の値を案件の処理期限の絶対日時として設定します。
<code>setPriority</code>	対象となる案件の優先度を変更します。
<code>start</code>	対象となる案件を開始させます。
<code>startAndAdhocInstantiateActivityInstanceByADName</code>	案件を開始し、指定した業務ステップを生成して実行状態にします。
<code>terminate</code>	対象となる案件を強制終了させます。

## adhocInstantiateActivityInstance

インタフェース名：CIWProcessInstance

### 構文

```

CIWActivityInstance adhocInstantiateActivityInstance(
    java.lang.Integer aActivityDefinitionID,
    java.lang.String aActivityInstanceName
)
    throws CIWFatalException,
           CIWTransientException,
           CIWStateException,
           CIWTransitionFailedException,
           CIWEntityNotExistException

```

### 機能

指定した業務ステップ定義に従って、対象となる案件内に、業務ステップを「初期」状態で生成します。

### 引数

`adhocInstantiateActivityInstance` の引数を次の表に示します。

表 5-56 `adhocInstantiateActivityInstance` の引数 (CIWProcessInstance の場合)

仮引数名	名称	in/out	説明
<code>aActivityDefinitionID</code>	業務ステップ定義の ID	in	業務ステップ定義の ID を指定します。

仮引数名	名称	in/out	説明
aActivityInstanceName	業務ステップ の名称	in	業務ステップの名称（業務ステップキー）を指定します。 <SYSTEMID>_ACTIVITY_INSTANCE テーブルの Name カラムのバイト数以下の文字列が指定できます。 null を指定した場合、業務ステップの名称は未設定になります。 空文字は指定できません。

## 戻り値

生成した業務ステップオブジェクトを返します。

## 例外

adhocInstantiateActivityInstance で発生する例外を次の表に示します。

表 5-57 adhocInstantiateActivityInstance で発生する例外（CIWProcessInstance の場合）

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合
CIWTransientException	一時的なエラーが発生した場合
CIWTransitionFailedException	案件処理中にエラーが発生した場合
CIWStateException	状態や属性の変更に失敗した場合
CIWEntityNotExistException	処理しようとしたオブジェクトが存在しない場合

## 注意事項

- 対象となる案件は、「未終了」状態である必要があります。それ以外の場合は、例外（CIWStateException）が発生します。
- 業務ステップ定義の ID には、遷移定義が存在しない業務ステップ定義の ID も指定できます。
- 指定した業務ステップ定義が存在しない場合、例外（CIWEntityNotExistException）が発生します。
- 業務ステップ定義の ID に null を指定した場合、例外（java.lang.IllegalArgumentException）が発生します。
- 指定する業務ステップの名称に空文字または<SYSTEMID>\_ACTIVITY\_INSTANCE テーブルの Name カラムのバイト数を超える文字列を指定した場合、例外（java.lang.IllegalArgumentException）が発生します。
- 対象となる案件のビジネスプロセス定義のプロパティ「アドホック API を使用する」にチェックが入っていない場合、例外（CIWFatalException）が発生します。

# changeState

インタフェース名：CIWProcessInstance

## 構文

```
void changeState(  
    CIWProcessInstance.State aNewState  
)  
    throws CIWFatalException,  
           CIWEntityNotExistException,  
           CIWTransientException,  
           CIWTransitionFailedException,  
           CIWStateException
```

## 機能

対象となる案件の状態を変更します。

## 引数

changeState の引数を次の表に示します。

表 5-58 changeState の引数 (CIWProcessInstance の場合)

仮引数名	名称	in/out	説明
aNewState	案件の状態	in	変更後の案件の状態を指定します。 指定する値として、このインタフェースに定義された CIWProcessInstance.State 列挙型の定数を指定できます。 null は指定できません。

aNewState で指定できる案件の状態を次の表に示します。

表 5-59 aNewState で指定できる案件の状態 (CIWProcessInstance の場合)

現在の状態		指定できる状態		状態遷移の意味
定数	意味	定数	意味	
RUNNING	実行中	INTERMITTED	実行停止	中断
INTERMITTED	実行停止	RUNNING	実行中	再開
—	未終了	TERMINATED	強制終了	強制終了

(凡例)

定数：CIWProcessInstance.State 列挙型の定数

状態遷移については、「付録 C 状態遷移モデル」を参照してください。

—：定数がないことを表します。



## 戻り値

ありません。

## 例外

changeState で発生する例外を次の表に示します。

表 5-60 changeState で発生する例外 (CIWProcessInstance の場合)

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合
CIWEntityNotExistException	処理しようとしたオブジェクトが存在しない場合
CIWTransientException	一時的なエラーが発生した場合
CIWTransitionFailedException	案件処理中にエラーが発生した場合
CIWStateException	状態の遷移に失敗した場合

## 注意事項

- エラー案件に対して実行した場合は、例外 (CIWStateException) が発生します。
- 変更する案件状態に null を指定した場合は、例外 (java.lang.IllegalArgumentException) が発生します。

## delete

インタフェース名：CIWProcessInstance

## 構文

```
void delete()  
    throws CIWTransientException,  
           CIWFatalException,  
           CIWEntityNotExistException,  
           CIWStateException
```

## 機能

対象となる案件を削除します。

案件に含まれるすべての業務ステップおよび作業も同時に削除します。

## 引数

ありません。

## 戻り値

ありません。

## 例外

delete で発生する例外を次の表に示します。

表 5-61 delete で発生する例外

発生する例外	説明
<code>CIWTransientException</code>	一時的なエラーが発生した場合
<code>CIWFatalException</code>	処理を続行できないような障害が発生した場合
<code>CIWEntityNotExistException</code>	処理しようとしたオブジェクトが存在しない場合
<code>CIWStateException</code>	状態や属性の変更に失敗した場合

## 注意事項

案件が「終了」状態の場合だけ削除できます。そのほかの状態の場合は、例外 (`CIWStateException`) が発生します。

## getClosedDate

インタフェース名：CIWProcessInstance

## 構文

```
java.util.Date getClosedDate()  
throws CIWFatalException
```

## 機能

対象となる案件の終了日時を取得します。

## 引数

ありません。

## 戻り値

案件の終了日時を返します。

## 例外

getClosedDate で発生する例外を次の表に示します。

表 5-62 getClosedDate で発生する例外 (CIWProcessInstance の場合)

発生する例外	説明
CIWFatalException	処理を続行できないような障害が発生した場合

## 注意事項

- 「終了」状態への遷移が発生した場合に値が設定されます。  
遷移の発生前は null が返されます。状態遷移については、「付録 C 状態遷移モデル」を参照してください。
- オブジェクト取得時に終了日時の取得を指定していない場合は、例外 (CIWFatalException) が発生します。

## getCreator

インタフェース名: CIWProcessInstance

### 構文

```
java.lang.String getCreator()
    throws CIWFatalException
```

### 機能

対象となる案件の投入者を取得します。

### 引数

ありません。

### 戻り値

案件の投入者を返します。

### 例外

getCreator で発生する例外を次の表に示します。

表 5-63 getCreator で発生する例外

発生する例外	説明
CIWFatalException	処理を続行できないような障害が発生した場合

## 注意事項

オブジェクト取得時に投入者の取得を指定していない場合は、例外（CIWFatalException）が発生します。

## getDeadline

インタフェース名：CIWProcessInstance

### 構文

```
java.util.Date getDeadline()  
    throws CIWFatalException
```

### 機能

対象となる案件の処理期限の絶対日時を取得します。

### 引数

ありません。

### 戻り値

案件の処理期限の絶対日時を返します。

### 例外

getDeadline で発生する例外を次の表に示します。

表 5-64 getDeadline で発生する例外（CIWProcessInstance の場合）

発生する例外	説明
<a href="#">CIWFatalException</a>	処理を続行できないような障害が発生した場合

## 注意事項

- 未設定時には null が返されます。
- オブジェクト取得時に処理期限の絶対日時の取得を指定していない場合は、例外（CIWFatalException）が発生します。

## getID

インタフェース名：CIWProcessInstance

## 構文

```
java.lang.Integer getID()  
    throws CIWFatalException
```

## 機能

対象となる案件の ID を取得します。

## 引数

ありません。

## 戻り値

案件の ID を返します。

## 例外

getID で発生する例外を次の表に示します。

表 5-65 getID で発生する例外 (CIWProcessInstance の場合)

発生する例外	説明
<a href="#">CIWFatalException</a>	処理を続行できないような障害が発生した場合

## getMovedDate

インタフェース名：CIWProcessInstance

## 構文

```
java.util.Date getMovedDate()  
    throws CIWFatalException
```

## 機能

対象となる案件の乗せ替え日時を取得します。

## 引数

ありません。

## 戻り値

案件の乗せ替え日時を返します。

## 例外

getMovedDate で発生する例外を次の表に示します。

表 5-66 getMovedDate で発生する例外

発生する例外	説明
CIWFatalException	処理を続行できないような障害が発生した場合

## 注意事項

- 未設定時には null が返されます。
- オブジェクト取得時に乗せ替え日時の取得を指定していない場合は、例外 (CIWFatalException) が発生します。

## getName

インタフェース名 : CIWProcessInstance

## 構文

```
java.lang.String getName()  
    throws CIWFatalException
```

## 機能

対象となる案件の名称 (案件キー) を取得します。

## 引数

ありません。

## 戻り値

案件の名称 (案件キー) を返します。

## 例外

getName で発生する例外を次の表に示します。

表 5-67 getName で発生する例外 (CIWProcessInstance の場合)

発生する例外	説明
CIWFatalException	処理を続行できないような障害が発生した場合

## 注意事項

- 未設定時には null が返されます。
- オブジェクト取得時に名称（案件キー）の取得を指定していない場合は、例外（CIWFatalException）が発生します。

## getPriority

インタフェース名：CIWProcessInstance

### 構文

```
java.lang.Integer getPriority()  
throws CIWFatalException
```

### 機能

対象となる案件の優先度を取得します。

### 引数

ありません。

### 戻り値

案件の優先度を返します。

### 例外

getPriority で発生する例外を次の表に示します。

表 5-68 getPriority で発生する例外（CIWProcessInstance の場合）

発生する例外	説明
<a href="#">CIWFatalException</a>	処理を続行できないような障害が発生した場合

## 注意事項

- 未設定時には null が返されます。
- オブジェクト取得時に優先度の取得を指定していない場合は、例外（CIWFatalException）が発生します。

# getProcessDefinition

インタフェース名：CIWProcessInstance

## 構文

```
CIWProcessDefinition getProcessDefinition(  
    java.util.Set<CIWProcessDefinition.AttributeName> aAttributeNames  
)  
    throws CIWFatalException,  
           CIWTransientException,  
           CIWEntityNotExistException
```

## 機能

対象となる案件が所属するビジネスプロセス定義オブジェクトを取得します。

## 引数

getProcessDefinition の引数を次の表に示します。

表 5-69 getProcessDefinition の引数 (CIWProcessInstance の場合)

仮引数名	名称	in/out	説明
aAttributeNames	属性名のセット	in	取得したいビジネスプロセス定義の属性名のセットを指定します。 取得属性名を指定しない場合は、サイズ 0 の属性名のセットまたは null を指定します。取得属性名に null は指定できません。取得属性名に指定されていない場合でも、ビジネスプロセス定義の ID は必ず取得します。取得可能な属性は、CIWProcessDefinition.AttributeName 列挙型で指定できる属性です。

## 戻り値

案件が所属するビジネスプロセス定義オブジェクトを返します。

## 例外

getProcessDefinition で発生する例外を次の表に示します。

表 5-70 getProcessDefinition で発生する例外 (CIWProcessInstance の場合)

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合
CIWTransientException	一時的なエラーが発生した場合
CIWEntityNotExistException	処理しようとしたオブジェクトが存在しない場合



## 注意事項

オブジェクト取得時に案件が所属するビジネスプロセス定義の ID の取得を指定していない場合、例外 (CIWFatalException) が発生します。

## getProcessDefinitionID

インタフェース名：CIWProcessInstance

### 構文

```
java.lang.Integer getProcessDefinitionID()  
    throws CIWFatalException
```

### 機能

対象となる案件が所属するビジネスプロセス定義の ID を取得します。

### 引数

ありません。

### 戻り値

案件が所属するビジネスプロセス定義の ID を返します。

### 例外

getProcessDefinitionID で発生する例外を次の表に示します。

表 5-71 getProcessDefinitionID で発生する例外 (CIWProcessInstance の場合)

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合

## 注意事項

オブジェクト取得時に案件が所属するビジネスプロセス定義の ID の取得を指定していない場合、例外 (CIWFatalException) が発生します。

## getProcessDefinitionName

インタフェース名：CIWProcessInstance

## 構文

```
java.lang.String getProcessDefinitionName()  
    throws CIWFatalException
```

## 機能

対象となる案件が所属するビジネスプロセス定義の名称を取得します。

## 引数

ありません。

## 戻り値

案件が所属するビジネスプロセス定義の名称を返します。

## 例外

getProcessDefinitionName で発生する例外を次の表に示します。

表 5-72 getProcessDefinitionName で発生する例外

発生する例外	説明
<a href="#">CIWFatalException</a>	処理を続行できない障害が発生した場合

## 注意事項

オブジェクト取得時に案件が所属するビジネスプロセス定義の名称の取得を指定していない場合は、例外 (CIWFatalException) が発生します。

## getStartDate

インタフェース名：CIWProcessInstance

## 構文

```
java.util.Date getStartDate()  
    throws CIWFatalException
```

## 機能

対象となる案件の開始日時を取得します。

## 引数

ありません。

## 戻り値

案件の開始日時を返します。

## 例外

getStartDate で発生する例外を次の表に示します。

表 5-73 getStartDate で発生する例外 (CIWProcessInstance の場合)

発生する例外	説明
<a href="#">CIWFatalException</a>	処理を続行できない障害が発生した場合

## 注意事項

- 「実行中」状態への遷移が発生した場合に値が設定されます。  
遷移の発生前は null が返されます。状態遷移については、「[付録 C 状態遷移モデル](#)」を参照してください。
- オブジェクト取得時に開始日時の取得を指定していない場合は、例外 (CIWFatalException) が発生します。

## getState

インタフェース名 : CIWProcessInstance

## 構文

```
CIWProcessInstance.State getState()  
    throws CIWFatalException
```

## 機能

対象となる案件の状態を取得します。

## 引数

ありません。

## 戻り値

案件の状態を返します。

## 例外

getState で発生する例外を次の表に示します。

表 5-74 getState で発生する例外 (CIWProcessInstance の場合)

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合

## 注意事項

オブジェクト取得時に状態の取得を指定していない場合は、例外 (CIWFatalException) が発生します。

## getWorkItemsList

インタフェース名 : CIWProcessInstance

## 構文

```
java.util.List<CIWorkItem> getWorkItemsList(
    java.lang.String aFilter,
    java.lang.String aSort,
    int aOffset,
    int aMaxCount,
    java.util.Set<CIWorkItem.AttributeName> aAttributeNames
)
throws CIWFatalException,
       CIWTransientException
```

## 機能

対象となる案件内で、指定したフィルター条件を満たす作業オブジェクトのリストを取得します。

## 引数

getWorkItemsList の引数を次の表に示します。

表 5-75 getWorkItemsList の引数 (CIWProcessInstance の場合)

仮引数名	名称	in/out	説明
aFilter	フィルター条件	in	作業を取得する場合のフィルター条件を指定します。 フィルター条件を指定しない場合は、null を指定します。 フィルター条件の指定方法については、「付録 B 参照系 API で指定できる属性」を参照してください。
aSort	ソート条件	in	作業を取得する場合のソート条件を指定します。 ソート条件を指定しない場合は、null を指定します。ソート条件の指定方法については、「付録 B 参照系 API で指定できる属性」を参照してください。
aOffset	オフセット	in	作業を取得する場合のオフセットを指定します。

仮引数名	名称	in/out	説明
			オフセットは先頭を0として指定します。0未満の値は指定できません。
aMaxCount	最大取得数	in	作業を取得する際の最大取得数を指定します。 すべてを取得する場合は、-1を指定します。-1未満の値は指定できません。
aAttributeNames	属性名のセット	in	取得したい作業の属性名のセットを指定します。 取得属性名を指定しない場合は、サイズ0の属性名のセットまたはnullを指定します。取得属性名にnullは指定できません。取得属性名に指定されていない場合でも、作業のIDと作業が所属する案件のIDは必ず取得します。取得可能な属性は、CIWorkItem.AttributeName 列挙型で指定できる属性です。

## 戻り値

取得した作業オブジェクトのリストを返します。

## 例外

getWorkItemsList で発生する例外を次の表に示します。

表 5-76 getWorkItemsList で発生する例外 (CIWProcessInstance の場合)

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合
CIWTransientException	一時的なエラーが発生した場合

## 注意事項

- 指定したフィルター条件を満たす作業オブジェクトが存在しない場合は、空のリストが返されます。
- 不正な引数を指定した場合は、例外 (java.lang.IllegalArgumentException) が発生します。
- オフセットに0, 50, 100などを指定し、複数回に分けて取得した場合は、作業の重複や漏れが発生することがあります。

## setDeadline

インタフェース名：CIWProcessInstance

## 構文

```
void setDeadline(
    java.util.Date aNewDeadline
)
```

```
throws CIWTransientException,  
        CIWStateException,  
        CIWFatalException,  
        CIWEntityNotExistException
```

## 機能

対象となる案件に対して、指定した日時の値を案件の処理期限の絶対日時として設定します。

## 引数

setDeadline の引数を次の表に示します。

表 5-77 setDeadline の引数 (CIWProcessInstance の場合)

仮引数名	名称	in/out	説明
aNewDeadline	変更後の処理期限	in	変更後の処理期限を指定します。 null を指定した場合は、処理期限は未設定となります。処理期限の絶対日時として指定できる値は、CIWUtil#convertDateToString(Date)で指定できる Date 値と同じです。処理期限はミリ秒以下については切り捨てて設定します。

## 戻り値

ありません。

## 例外

setDeadline で発生する例外を次の表に示します。

表 5-78 setDeadline で発生する例外 (CIWProcessInstance の場合)

発生する例外	説明
CIWTransientException	一時的なエラーが発生した場合
CIWStateException	状態の遷移に失敗した場合
CIWFatalException	処理を続行できない障害が発生した場合
CIWEntityNotExistException	処理しようとしたオブジェクトが存在しない場合

## 注意事項

案件が「未終了」状態のときだけ変更できます。ただし、案件の処理期限を変更しても、対象となる案件に所属する、変更以前に生成されていた業務ステップの処理期限は変更されません。

# setPriority

インタフェース名：CIWProcessInstance

## 構文

```
void setPriority(  
    java.lang.Integer aNewPriority  
)  
    throws CIWTransientException,  
           CIWStateException,  
           CIWFatalException,  
           CIWEntityNotExistException
```

## 機能

対象となる案件の優先度を変更します。

## 引数

setPriority の引数を次の表に示します。

表 5-79 setPriority の引数 (CIWProcessInstance の場合)

仮引数名	名称	in/out	説明
aNewPriority	変更後の優先度	in	変更後の優先度を指定します。 null を指定した場合は、優先度は未設定となります。

## 戻り値

ありません。

## 例外

setPriority で発生する例外を次の表に示します。

表 5-80 setPriority で発生する例外 (CIWProcessInstance の場合)

発生する例外	説明
<a href="#">CIWTransientException</a>	一時的なエラーが発生した場合
<a href="#">CIWStateException</a>	状態の遷移に失敗した場合
<a href="#">CIWFatalException</a>	処理を続行できない障害が発生した場合
<a href="#">CIWEntityNotExistException</a>	処理しようとしたオブジェクトが存在しない場合

## 注意事項

案件が「未終了」状態のときだけ変更できます。ただし、案件の優先度を変更しても、対象となる案件に所属する、変更以前に生成されていた業務ステップの優先度は変更されません。

## start

インタフェース名：CIWProcessInstance

## 構文

```
void start()
    throws CIWTransientException,
           CIWTransitionFailedException,
           CIWFatalException,
           CIWEntityNotExistException,
           CIWStateException
```

## 機能

対象となる案件を開始させます。

## 引数

ありません。

## 戻り値

ありません。

## 例外

start で発生する例外を次の表に示します。

表 5-81 start で発生する例外

発生する例外	説明
CIWTransientException	一時的なエラーが発生した場合
CIWTransitionFailedException	案件処理中にエラーが発生した場合
CIWFatalException	処理を続行できない障害が発生した場合
CIWEntityNotExistException	処理しようとしたオブジェクトが存在しない場合
CIWStateException	状態や属性の変更に失敗した場合



## 注意事項

案件が「未実行」状態の場合に開始できます。そのほかの状態の場合は、例外（CIWStateException）が発生します。

## startAndAdhocInstantiateActivityInstanceByADName

インタフェース名：CIWProcessInstance

## 構文

```
void startAndAdhocInstantiateActivityInstanceByADName(  
    java.util.Set  
    <java.lang.String> aActivityDefinitionNames  
)  
    throws CIWTransientException,  
           CIWTransitionFailedException,  
           CIWFatalException,  
           CIWEntityNotExistException,  
           CIWStateException
```

## 機能

指定した業務ステップを初期状態で生成して、案件の状態を「実行中」にします。また、事前生成が有効になっている業務ステップも初期状態で生成します。

生成対象の業務ステップは、業務ステップ定義名で指定します。

## 引数

startAndAdhocInstantiateActivityInstanceByADName の引数を次の表に示します。

表 5-82 startAndAdhocInstantiateActivityInstanceByADName の引数  
(CIWProcessInstance の場合)

仮引数名	名称	in/out	説明
aActivityDefinitionNames	業務ステップ定義の定義名のセット	in	業務ステップ定義の定義名のセットを指定します。

## 戻り値

ありません。

## 例外

startAndAdhocInstantiateActivityInstanceByADName で発生する例外を次の表に示します。

表 5-83 startAndAdhocInstantiateActivityInstanceByADName で発生する例外

発生する例外	説明
CIWTransientException	一時的なエラーが発生した場合
CIWTransitionFailedException	案件処理中にエラーが発生した場合
CIWFatalException	処理を続行できない障害が発生した場合
CIWEntityNotExistException	処理しようとしたオブジェクトが存在しない場合
CIWStateException	状態や属性の変更に失敗した場合

## 注意事項

- 対象となる案件は「未実行」状態である必要があります。それ以外の場合は、例外 (CIWStateException) が発生します。
- 待合ノードへ遷移する業務ステップを指定する場合は、待合ノードへ遷移するすべての業務ステップ定義名を指定する必要があります。
- 業務ステップ定義の名称には、遷移定義が存在しない業務ステップ定義の定義名も指定できます。
- 対象となる案件のビジネスプロセス定義のプロパティ「アドホック API を使用する」にチェックが入っていない場合は、例外 (CIWFatalException) が発生します。
- 指定した業務ステップ定義が存在しない場合は、例外 (CIWEntityNotExistException) が発生します。
- 引数 aActivityDefinitionNames に null または空のセットを指定した場合は、例外 (java.lang.IllegalArgumentException) が発生します。

## terminate

インタフェース名：CIWProcessInstance

## 構文

```
void terminate()
    throws CIWTransientException,
           CIWTransitionFailedException,
           CIWFatalException,
           CIWEntityNotExistException,
           CIWStateException
```

## 機能

対象となる案件を強制終了させます。

## 引数

ありません。

## 戻り値

ありません。

## 例外

terminate で発生する例外を次の表に示します。

表 5-84 terminate で発生する例外

発生する例外	説明
<code>CIWTransientException</code>	一時的なエラーが発生した場合
<code>CIWTransitionFailedException</code>	案件処理中にエラーが発生した場合
<code>CIWFatalException</code>	処理を続行できない障害が発生した場合
<code>CIWEntityNotExistException</code>	処理しようとしたオブジェクトが存在しない場合
<code>CIWStateException</code>	状態や属性の変更に失敗した場合

## 注意事項

- 案件が「未終了」状態の場合に強制終了できます。そのほかの状態の場合は、例外（`CIWStateException`）が発生します。
- エラー案件に対して実行した場合は、例外（`CIWStateException`）が発生します。

## 5.7 CIWServer (ワーク管理システム全般にかかわる処理を行うための機能を提供するインタフェース)

ワーク管理システム全般にかかわる処理を行うための機能を提供するインタフェースです。

### クラス定義

```
public interface CIWServer
```

### 入れ子のクラス

CIWServer インタフェースの入れ子のクラスを次の表に示します。

表 5-85 CIWServer インタフェースの入れ子のクラス

クラス名	説明および記述形式
CIWServer.Targetkind	ワーク管理システムが管理するオブジェクトの種別を表す列挙型です。
	<code>static class CIWServer.Targetkind</code>

### メソッド

CIWServer インタフェースのメソッドを次の表に示します。

表 5-86 CIWServer インタフェースのメソッド

メソッド名	説明
<code>allocateWorkItemEx</code>	指定したフィルター条件とソート条件を満たす作業群に対して、指定した割り当て先の作業 ID へ作業を割り当てます。最初に割り当てに成功した作業オブジェクトを返します。
<code>createAndStartProcessInstance</code>	案件属性、ビジネスプロセス定義の名称およびバージョンを指定して、案件を生成し開始します。
<code>createProcessInstance</code>	案件属性、ビジネスプロセス定義の名称およびバージョンを指定して、案件を生成します。
<code>getActivityDefinition</code>	ビジネスプロセス定義の ID および業務ステップ定義の ID を指定して、業務ステップ定義オブジェクトを取得します。
<code>getActivityDefinitionsList</code>	指定したフィルター条件を満たす業務ステップ定義オブジェクトのリストを取得します。
<code>getActivityInstance</code>	案件の ID と業務ステップの ID を指定して、業務ステップオブジェクトを取得します。
<code>getActivityInstancesList</code>	指定したフィルター条件を満たす業務ステップオブジェクトのリストを取得します。
<code>getConnectionCoordinator</code>	CIWServer オブジェクト内に保持されているコネクション関連づけオブジェクトを取得します。
<code>getListCount</code>	指定したフィルター条件を満たす各種オブジェクトのリスト件数を取得します。

メソッド名	説明
getProcessDefinition	ビジネスプロセス定義の ID を指定して、ビジネスプロセス定義オブジェクトを取得します。
getProcessDefinitionsList	指定したフィルター条件を満たすビジネスプロセス定義オブジェクトのリストを取得します。
getProcessInstance	案件の ID を指定して、案件オブジェクトを取得します。
getProcessInstancesList	指定したフィルター条件を満たす案件オブジェクトのリストを取得します。
getWorkDefinition	ビジネスプロセス定義の ID および作業定義の ID を指定して、作業定義オブジェクトを取得します。
getWorkDefinitionsList	指定したフィルター条件を満たす作業定義オブジェクトのリストを取得します。
getWorkItem	案件の ID と作業の ID を指定して、作業オブジェクトを取得します。
getWorkItemsList	指定したフィルター条件を満たす作業オブジェクトのリストを取得します。

## allocateWorkItemEx

インタフェース名：CIWServer

### 構文

```
public CIWorkItem allocateWorkItemEx(
    String aParticipant,
    String aFilter,
    String aSort,
    Set<CIWorkItem.AttributeName> aAttributeNames,
    int aFetchCount
)
```

### 機能

指定したフィルター条件とソート条件を満たす作業群に対して、指定した割り当て先の作業 ID へ作業を割り当てます。最初に割り当てに成功した作業オブジェクトを返します。

割り当てる際に、作業オブジェクトを「実行開始可能」状態から「作業実行」状態に変更します。指定したフィルター条件を満たす作業がない場合や、作業を割り当てられる作業がない場合は null を返します。

フィルター条件には次の 2 つの条件を AND 条件で自動的に追加します。

- StateCode が"j" (実行開始可能) と一致する
- 一般作業である

## 引数

allocateWorkItemEx の引数を次の表に示します。

表 5-87 allocateWorkItemEx の引数

仮引数名	名称	in/out	説明
aParticipant	作業者 ID	in	割り当てる作業者 ID を<SYSTEMID>_WORK_ITEM テーブルの Participant カラムのバイト数以下で指定します。 null は指定できません。
aFilter	フィルター条件	in	作業を取得する場合のフィルター条件を指定します。 フィルター条件を指定しない場合は、null を指定します。 フィルター条件の指定方法については、「 <a href="#">付録 B 参照系 API で指定できる属性</a> 」を参照してください。
aSort	ソート条件	in	作業を取得する場合のソート条件を指定します。 ソート条件を指定しない場合は、null を指定します。ソート条件の指定方法については、「 <a href="#">付録 B 参照系 API で指定できる属性</a> 」を参照してください。
aAttributeNames	属性名のセット	in	取得したい作業の属性名のセットを指定します。 取得属性名を指定しない場合は、サイズ 0 の属性名のセットまたは null を指定します。取得属性名に null は指定できません。取得属性名を指定しない場合でも、作業の ID、作業が所属する案件の ID、作業者の ID、作業の状態、および作業の開始日時は必ず取得します。取得可能な属性は、CIWorkItem.AttributeName 列挙型で指定できる属性です。
aFetchCount	最大取得数	in	作業を取得する場合の最大取得数を指定します。 すべてを取得する場合は、-1 を指定します。-1 未満の値は設定できません。

## 戻り値

取得した CIWorkItem（作業）オブジェクトを返します。取得した CIWorkItem（作業）オブジェクトがない場合は null を返します。

## 例外

allocateWorkItemEx で発生する例外を次の表に示します。

表 5-88 allocateWorkItemEx で発生する例外

発生する例外	説明
<a href="#">CIWTransientException</a>	一時的なエラーが発生した場合
<a href="#">CIWFatalException</a>	処理を続行できない障害が発生した場合

## 注意事項

- メソッド実行直後に、トランザクションをコミットしてください。メソッド発行からコミットまでの間に、業務テーブルの更新などを実行しないでください。
- このメソッドで取得した作業を返却する場合は、CIWorkItem クラスの free メソッドを実行します。
- 引数 aFetchCount で指定した数の作業群の中から、先頭の作業に対して実行します。失敗した場合は、それ以降ランダムに割り当てを実行します。つまり、順番に並んでいる作業のうち、2 件目から aFetchCount 分の範囲内ではソートの順番が無視されます。ソートの順番を保証するには、aFetchCount を 1 に指定してください。しかし、aFetchCount を 1 にした場合、このメソッドが同時に多数発行されると作業の割り当てが重複して、レスポンスが悪くなります。
- 取得した作業群に対して、一定回数（作業群の作業の 4 割）作業の割り当てを失敗した場合は、指定したフィルター条件とソート条件を満たす作業群を取得し直します。
- 割り当てる作業 ID に空文字列または null を指定した場合は、例外 (java.lang.IllegalArgumentException) が発生します。
- 割り当てる作業 ID に<SYSTEMID>\_WORK\_ITEM テーブルの Participant カラムのバイト数を超える文字列を指定した場合は、例外 (java.lang.IllegalArgumentException) が発生します。
- 一般作業だけが対象になります。組み込み作業は対象外です。
- 引数 aFetchCount は、同時実行する API の数より多く指定してください。少ない場合、同時実行された API 同士の衝突が多く発生し、効率が悪くなります。
- Oracle はダーティリードをサポートしていません。そのため、コミットしていないの別トランザクションの更新を読み取れません。その結果、Oracle を利用している場合は、API の更新処理が同時に複数実行できないため効率的に処理ができません。

## createAndStartProcessInstance

インタフェース名：CIWServer

### 構文

```
CIWProcessInstance createAndStartProcessInstance(  
    java.util.Map  
    <CIWProcessInstance.AttributeName, java.lang.Object> aAttributes,  
    java.lang.String aProcessDefinitionName,  
    java.lang.Short aVersion  
)  
    throws CIWFatalException,  
           CIWTransitionFailedException,  
           CIWTransientException,  
           CIWEntityNotExistException
```

## 機能

案件属性、ビジネスプロセス定義の名称およびバージョンを指定し、案件を生成して開始します。

## 引数

createAndStartProcessInstance の引数を次の表に示します。

表 5-89 createAndStartProcessInstance の引数

仮引数名	名称	in/out	説明
aAttributes	案件属性のマップ	in	<p>案件属性（案件属性名と属性値の組）のマップを指定します。指定できる属性を次に示します。</p> <ul style="list-style-type: none"><li>• 案件名 (Name)</li><li>• 処理期限 (Deadline)</li><li>• 優先度 (Priority)</li></ul> <p>案件属性を指定しない場合は、null またはサイズ 0 の案件属性のマップを指定します。各案件属性名の属性値は次に示す型で指定します。</p> <ul style="list-style-type: none"><li>• 案件名：java.lang.String 型</li><li>• 処理期限：java.util.Date 型</li><li>• 優先度：java.lang.Integer 型</li></ul> <p>属性値に null を指定した場合は、指定した案件属性は未設定になります。また、案件名の値は、空文字は指定できません。処理期限の絶対日時として指定できる値は、CIWUtil#convertDateToString(Date)で指定できる Date 値と同じです。ただし、ミリ秒以下については切り捨てられます。</p>
aProcessDefinitionName	ビジネスプロセス定義の名称	in	<p>ビジネスプロセス定義の名称を指定します。空文字列または null は指定できません。</p>
aVersion	ビジネスプロセス定義のバージョン	in	<p>ビジネスプロセス定義のバージョンを指定します。null を指定した場合、指定したビジネスプロセス定義の中で最新バージョンとなります。</p>

## 戻り値

開始した案件オブジェクトを返します。

## 例外

createAndStartProcessInstance で発生する例外を次の表に示します。

表 5-90 createAndStartProcessInstance で発生する例外

発生する例外	説明
<a href="#">CIWFatalException</a>	処理を続行できない障害が発生した場合



発生する例外	説明
<code>CIWTransitionFailedException</code>	案件処理中にエラーが発生した場合
<code>CIWTransientException</code>	一時的なエラーが発生した場合
<code>CIWEntityNotExistException</code>	処理しようとしたオブジェクトが存在しない場合

## 注意事項

- 処理期限を未設定とした場合でも、ビジネスプロセス定義の処理期限ルールを定義している場合、ビジネスプロセス定義の処理期限ルールに従い、案件開始時に案件の開始日時 (START\_DATE) を起算日として算出された値が設定されます。
- 案件名の値に空文字を指定した場合は、例外 (`java.lang.IllegalArgumentException`) が発生します。
- 指定された属性名と属性値の型が一致しない場合は、例外 (`java.lang.IllegalArgumentException`) が発生します。
- ビジネスプロセス定義の名称に空文字列または `null` を指定した場合は、例外 (`java.lang.IllegalArgumentException`) が発生します。
- 存在しないビジネスプロセス定義のバージョンを指定した場合は、例外 (`CIWEntityNotExistException`) が発生します。

## createProcessInstance

インタフェース名：CIWServer

### 構文

```
CIWProcessInstance createProcessInstance(
    java.util.Map
    <CIWProcessInstance.AttributeName, java.lang.Object> aAttributes,
    java.lang.String aProcessDefinitionName,
    java.lang.Short aVersion
)
    throws CIWFatalException,
           CIWTransientException,
           CIWEntityNotExistException
```

### 機能

案件属性、ビジネスプロセス定義の名称およびバージョンを指定して、案件を生成します。

### 引数

`createProcessInstance` の引数を次の表に示します。

表 5-91 createProcessInstance の引数

仮引数名	名称	in/out	説明
aAttributes	案件属性のマップ	in	<p>案件属性（案件属性名と属性値の組）のマップを指定します。指定できる属性を次に示します。</p> <ul style="list-style-type: none"> <li>• 案件名 (Name)</li> <li>• 処理期限 (Deadline)</li> <li>• 優先度 (Priority)</li> </ul> <p>案件属性を指定しない場合は、null またはサイズ 0 の案件属性のマップを指定します。各案件属性名の属性値は次に示す型で指定します。</p> <ul style="list-style-type: none"> <li>• 案件名：java.lang.String 型</li> <li>• 処理期限：java.util.Date 型</li> <li>• 優先度：java.lang.Integer 型</li> </ul> <p>属性値に null を指定した場合は、指定した案件属性は未設定になります。案件名の値は、空文字は指定できません。処理期限の絶対日時として指定できる値は、CIWUtil#convertDateToString(Date)で指定できる Date 値と同じです。ただし、ミリ秒以下については切り捨てられます。</p>
aProcessDefinitionName	ビジネスプロセス定義の名称	in	<p>ビジネスプロセス定義の名称を指定します。空文字列または null は指定できません。</p>
aVersion	ビジネスプロセス定義のバージョン	in	<p>ビジネスプロセス定義のバージョンを指定します。null を指定した場合、指定したビジネスプロセス定義の中で最新バージョンとなります。</p>

## 戻り値

生成した案件オブジェクトを返します。

## 例外

createProcessInstance で発生する例外を次の表に示します。

表 5-92 createProcessInstance で発生する例外

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合
CIWTransientException	一時的なエラーが発生した場合
CIWEntityNotExistException	処理しようとしたオブジェクトが存在しない場合

## 注意事項

- ビジネスプロセス定義の処理期限ルールを定義している場合は、処理期限を未設定にしてもビジネスプロセス定義の処理期限ルールに従って値が設定されます。値は案件開始時に案件の開始日時 (START\_DATE) を起算日として算出されます。
- 案件名の値に空文字を指定した場合は、例外 (java.lang.IllegalArgumentException) が発生します。
- 指定された属性名と属性値の型が一致しない場合は、例外 (java.lang.IllegalArgumentException) が発生します。
- ビジネスプロセス定義の名称に空文字列または null を指定した場合は、例外 (java.lang.IllegalArgumentException) が発生します。
- 存在しないビジネスプロセス定義のバージョンを指定した場合は、例外 (CIWEntityNotExistException) が発生します。

## getActivityDefinition

インタフェース名：CIWServer

### 構文

```
CIWActivityDefinition getActivityDefinition(  
    java.lang.Integer aProcessDefinitionID,  
    java.lang.Integer aActivityDefinitionID,  
    java.util.Set<CIWActivityDefinition.AttributeName> aAttributeNames  
)  
    throws CIWFatalException,  
           CIWTransientException,  
           CIWEntityNotExistException
```

### 機能

ビジネスプロセス定義の ID および業務ステップ定義の ID を指定して、業務ステップ定義オブジェクトを取得します。

### 引数

getActivityDefinition の引数を次の表に示します。

表 5-93 getActivityDefinition の引数

仮引数名	名称	in/out	説明
aProcessDefinitionID	ビジネスプロセス定義の ID	in	ビジネスプロセス定義の ID を指定します。 null は指定できません。
aActivityDefinitionID	業務ステップ定義の ID	in	業務ステップ定義の ID を指定します。 null は指定できません。

仮引数名	名称	in/out	説明
aAttributeNames	取得したい業務ステップ定義の属性名のセット	in	取得したい業務ステップ定義の属性名のセットを指定します。取得属性名を指定しない場合は、サイズ0の属性名のセットまたは null を指定します。取得属性名に null は指定できません。取得可能な属性は、CIWActivityDefinition.AttributeName 列挙型で指定できる属性です。

## 戻り値

指定したビジネスプロセス定義の ID と業務ステップ定義の ID を持つ業務ステップ定義オブジェクトを返します。

## 例外

getActivityDefinition で発生する例外を次の表に示します。

表 5-94 getActivityDefinition で発生する例外

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合
CIWTransientException	一時的なエラーが発生した場合
CIWEntityNotExistException	処理しようとしたオブジェクトが存在しない場合

## 注意事項

- ビジネスプロセス定義の ID に null を指定した場合は、例外 (java.lang.IllegalArgumentException) が発生します。
- 業務ステップ定義の ID に null を指定した場合は、例外 (java.lang.IllegalArgumentException) が発生します。
- 指定した条件を満たす業務ステップ定義オブジェクトが存在しない場合は、例外 (CIWEntityNotExistException) が発生します。
- 取得属性名を何も指定していない場合は、指定したビジネスプロセス定義の ID および業務ステップ定義の ID を設定しただけの業務ステップ定義オブジェクトを生成して返します。
- 指定されたビジネスプロセス定義の ID または業務ステップ定義の ID が不正な場合、このメソッドが返した業務ステップ定義オブジェクトのメソッドを呼び出したときに、エラーとなることがあります。

## getActivityDefinitionsList

インタフェース名：CIWServer

## 構文

```
java.util.List<CIWActivityDefinition> getActivityDefinitionsList(  
    java.lang.String aFilter,  
    java.lang.String aSort,  
    int aOffset,  
    int aMaxCount,  
    java.util.Set<CIWActivityDefinition.AttributeName> aAttributeNames  
)  
    throws CIWFatalException,  
    CIWTransientException
```

## 機能

指定したフィルター条件を満たす業務ステップ定義オブジェクトのリストを取得します。

## 引数

getActivityDefinitionsList の引数を次の表に示します。

表 5-95 getActivityDefinitionsList の引数

仮引数名	名称	in/out	説明
aFilter	業務ステップ定義を取得する際のフィルター条件	in	業務ステップ定義を取得する際のフィルター条件を指定します。 フィルター条件を指定しない場合は、null を指定します。 フィルター条件の指定方法については、「付録 B 参照系 API で指定できる属性」を参照してください。
aSort	業務ステップ定義を取得する際のソート条件	in	業務ステップ定義を取得する際のソート条件を指定します。 ソート条件を指定しない場合は、null を指定します。ソート条件の指定方法については、「付録 B 参照系 API で指定できる属性」を参照してください。
aOffset	業務ステップ定義を取得する際のオフセット	in	業務ステップ定義を取得する際のオフセットを指定します。 オフセットは先頭を 0 として指定します。0 未満の値は指定できません。
aMaxCount	業務ステップ定義を取得する際の最大取得数	in	業務ステップ定義を取得する際の最大取得数を指定します。 すべてを取得する場合は-1 を指定します。-1 未満の値は指定できません。
aAttributeNames	取得したい業務ステップ定義の属性名のセット	in	取得したい業務ステップ定義の属性名のセットを指定します。 取得属性名を指定しない場合は、サイズ 0 の属性名のセットまたは null を指定します。取得属性名に null は指定できません。取得属性名が指定されていない場合でも、業務ステップ定義が所属するビジネスプロセス定義の ID と業務ステップ定義の ID は必ず取得します。取得可能な属性は、CIWActivityDefinition.AttributeName 列挙型で指定できる属性です。

## 戻り値

取得した業務ステップ定義オブジェクトのリストを返します。

## 例外

getActivityDefinitionsList で発生する例外を次の表に示します。

表 5-96 getActivityDefinitionsList で発生する例外

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合
CIWTransientException	一時的なエラーが発生した場合

## 注意事項

- 指定したフィルター条件を満たす業務ステップ定義オブジェクトが存在しない場合は、空のリストが返されます。
- 不正な引数を指定した場合は、例外 (java.lang.IllegalArgumentException) が発生します。
- オフセットに 0, 50, 100 のように指定して、複数回に分けて取得した場合は、業務ステップ定義の重複や漏れが発生することがあります。

## getActivityInstance

インタフェース名：CIWServer

## 構文

```
CIWActivityInstance getActivityInstance(  
    java.lang.Integer aProcessInstanceID,  
    java.lang.Integer aActivityInstanceID,  
    java.util.Set<CIWActivityInstance.AttributeName> aAttributeNames  
)  
    throws CIWFatalException,  
           CIWTransientException,  
           CIWEntityNotExistException
```

## 機能

案件の ID と業務ステップの ID を指定して、業務ステップオブジェクトを取得します。

## 引数

getActivityInstance の引数を次の表に示します。

表 5-97 getActivityInstance の引数

仮引数名	名称	in/out	説明
aProcessInstanceID	案件の ID	in	案件の ID を指定します。 null は指定できません。
aActivityInstanceID	業務ステップの ID	in	業務ステップの ID を指定します。 null は指定できません。
aAttributeNames	業務ステップの属性名のセット	in	取得したい業務ステップの属性名のセットを指定します。 取得属性名を指定しない場合は、サイズ 0 の属性名のセットまたは null を指定します。取得属性名に null は指定できません。取得可能な属性は、CIWActivityInstance.AttributeName 列挙型で指定できる属性です。

## 戻り値

指定した案件の ID と業務ステップの ID を持つ業務ステップオブジェクトを返します。

## 例外

getActivityInstance で発生する例外を次の表に示します。

表 5-98 getActivityInstance で発生する例外

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合
CIWTransientException	一時的なエラーが発生した場合
CIWEntityNotExistException	処理しようとしたオブジェクトが存在しない場合

## 注意事項

- 案件の ID に null を指定した場合は、例外 (java.lang.IllegalArgumentException) が発生します。
- 業務ステップの ID に null を指定した場合は、例外 (java.lang.IllegalArgumentException) が発生します。
- 指定した条件を満たす業務ステップオブジェクトが存在しない場合は、例外 (CIWEntityNotExistException) が発生します。

- 取得属性名を何も指定していない場合は、指定した案件の ID および業務ステップの ID を設定しただけの業務ステップオブジェクトを生成して返します。

指定した案件の ID または業務ステップの ID が不正な場合は、業務ステップオブジェクトのメソッドを呼び出したときに、エラーになることがあります。

# getActivityInstancesList

インタフェース名：CIWServer

## 構文

```
java.util.List<CIWActivityInstance> getActivityInstancesList(  
    java.lang.String aFilter,  
    java.lang.String aSort,  
    int aOffset,  
    int aMaxCount,  
    java.util.Set<CIWActivityInstance.AttributeName> aAttributeNames  
)  
throws CIWFatalException,  
    CIWTransientException
```

## 機能

指定したフィルター条件を満たす業務ステップオブジェクトのリストを取得します。

## 引数

getActivityInstancesList の引数を次の表に示します。

表 5-99 getActivityInstancesList の引数

仮引数名	名称	in/out	説明
aFilter	フィルター条件	in	業務ステップを取得する場合のフィルター条件を指定します。フィルター条件を指定しない場合は、null を指定します。フィルター条件の指定方法については、「付録 B 参照系 API で指定できる属性」を参照してください。
aSort	ソート条件	in	業務ステップを取得する場合のソート条件を指定します。ソート条件を指定しない場合は、null を指定します。ソート条件の指定方法については、「付録 B 参照系 API で指定できる属性」を参照してください。
aOffset	オフセット	in	業務ステップを取得する場合のオフセットを指定します。オフセットは先頭を 0 として指定します。0 未満の値は指定できません。
aMaxCount	最大取得数	in	業務ステップを取得する場合の最大取得数を指定します。すべてを取得する場合は、-1 を指定します。-1 未満の値は指定できません。
aAttributeNames	属性名のセット	in	取得したい業務ステップの属性名のセットを指定します。取得属性名を指定しない場合は、サイズ 0 の属性名のセットまたは null を指定します。取得属性名に null は指定できません。取得属性名に指定されていない場合でも、業務ステップが所属する案件の ID と業務ステップの ID は必ず取得します。取得可能な属性は、



仮引数名	名称	in/out	説明
			CIWActivityInstance.AttributeName 列挙型で指定できる属性です。

## 戻り値

取得した業務ステップオブジェクトのリストを返します。

## 例外

getActivityInstancesList で発生する例外を次の表に示します。

表 5-100 getActivityInstancesList で発生する例外

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合
CIWTransientException	一時的なエラーが発生した場合

## 注意事項

- 指定したフィルター条件を満たす業務ステップオブジェクトが存在しない場合は、空のリストが返されます。
- 不正な引数を指定した場合は、例外 (java.lang.IllegalArgumentException) が発生します。
- オフセットに 0, 50, 100などを指定し、複数回に分けて取得した場合は、業務ステップの重複や漏れが発生することがあります。

## getConnectionCoordinator

インタフェース名：CIWServer

## 構文

```
CIWConnectionCoordinator getConnectionCoordinator()
    throws CIWFatalExceptionCIWServer
```

## 機能

CIWServer オブジェクト内に保持されているコネクション関連づけオブジェクトを取得します。

## 引数

ありません。

## 戻り値

コネクション関連づけオブジェクトを返します。

## 例外

getConnectionCoordinator で発生する例外を次の表に示します。

表 5-101 getConnectionCoordinator で発生する例外

発生する例外	説明
<a href="#">CIWFatalException</a>	処理を続行できない障害が発生した場合

## 注意事項

javax.sql.DataSource を設定し、CIWServer オブジェクトを生成している場合に、このメソッドを呼び出すと例外 (CIWFatalException) が発生します。

## getListCount

インタフェース名：CIWServer

## 構文

```
int getListCount(  
    CIWServer.Targetkind aTargetKind,  
    java.lang.String aFilter  
)  
    throws CIWFatalException,  
           CIWTransientException
```

## 機能

指定したフィルター条件を満たす各種オブジェクトのリスト件数を取得します。

## 引数

getListCount の引数を次の表に示します。

表 5-102 getListCount の引数

仮引数名	名称	in/out	説明
aTargetKind	対象	in	件数を取得する対象を指定します。 null は指定できません。
aFilter	フィルター条件	in	件数を取得する場合のフィルター条件を指定します。

仮引数名	名称	in/out	説明
			フィルター条件を指定しない場合は、null を指定します。フィルター条件の指定方法については、「付録 B 参照系 API で指定できる属性」を参照してください。

## 戻り値

フィルター条件と一致する件数を返します。

## 例外

getListCount で発生する例外を次の表に示します。

表 5-103 getListCount で発生する例外

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合
CIWTransientException	一時的なエラーが発生した場合

## 注意事項

取得する対象に null を指定した場合は、例外 (java.lang.IllegalArgumentException) が発生します。

## getProcessDefinition

インタフェース名：CIWServer

## 構文

```
CIWProcessDefinition getProcessDefinition(
    java.lang.Integer aProcessDefinitionID,
    java.util.Set<CIWProcessDefinition.AttributeName> aAttributeNames
)
throws CIWFatalException,
       CIWTransientException,
       CIWEntityNotExistException
```

## 機能

ビジネスプロセス定義の ID を指定して、ビジネスプロセス定義オブジェクトを取得します。

## 引数

getProcessDefinition の引数を次の表に示します。

表 5-104 getProcessDefinition の引数 (CIWServer の場合)

仮引数名	名称	in/out	説明
aProcessDefinitionID	ビジネスプロセス定義の ID	in	ビジネスプロセス定義の ID を指定します。 null は指定できません。
aAttributeNames	属性名のセット	in	取得したいビジネスプロセス定義の属性名のセットを指定します。 取得属性名を指定しない場合は、サイズ 0 の属性名のセットまたは null を指定します。取得属性名に null は指定できません。取得可能な属性は、CIWProcessDefinition.AttributeName 列挙型で指定できる属性です。

## 戻り値

指定したビジネスプロセス定義の ID を持つビジネスプロセス定義オブジェクトを返します。

## 例外

getProcessDefinition で発生する例外を次の表に示します。

表 5-105 getProcessDefinition で発生する例外 (CIWServer の場合)

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合
CIWTransientException	一時的なエラーが発生した場合
CIWEntityNotExistException	処理しようとしたオブジェクトが存在しない場合

## 注意事項

- ビジネスプロセス定義の ID に null を指定した場合は、例外 (java.lang.IllegalArgumentException) が発生します。
- 指定した条件を満たすビジネスプロセス定義オブジェクトが存在しない場合は、例外 (CIWEntityNotExistException) が発生します。
- 取得属性名を何も指定していない場合は、指定したビジネスプロセス定義の ID を設定しただけのビジネスプロセス定義オブジェクトを生成して返します。  
指定したビジネスプロセス定義の ID が不正な場合は、そのあとビジネスプロセス定義オブジェクトのメソッドを呼び出したときに、エラーとなることがあります。

## getProcessDefinitionsList

インタフェース名：CIWServer

## 構文

```
java.util.List<CIWProcessDefinition> getProcessDefinitionsList(  
    java.lang.String aFilter,  
    java.lang.String aSort,  
    int aOffset,  
    int aMaxCount,  
    java.util.Set<CIWProcessDefinition.AttributeName> aAttributeNames  
)  
    throws CIWFatalException,  
    CIWTransientException
```

## 機能

指定したフィルター条件を満たすビジネスプロセス定義オブジェクトのリストを取得します。

## 引数

getProcessDefinitionsList の引数を次の表に示します。

表 5-106 getProcessDefinitionsList の引数

仮引数名	名称	in/out	説明
aFilter	フィルター条件	in	ビジネスプロセス定義を取得する場合のフィルター条件を指定します。 フィルター条件を指定しない場合は、null を指定します。 フィルター条件の指定方法については、「付録 B 参照系 API で指定できる属性」を参照してください。
aSort	ソート条件	in	ビジネスプロセス定義を取得する場合のソート条件を指定します。 ソート条件を指定しない場合は、null を指定します。ソート条件の指定方法については、「付録 B 参照系 API で指定できる属性」を参照してください。
aOffset	オフセット	in	ビジネスプロセス定義を取得する場合のオフセットを指定します。 オフセットは先頭を 0 として指定します。0 未満の値は指定できません。
aMaxCount	最大取得数	in	ビジネスプロセス定義を取得する場合の最大取得数を指定します。 すべてを取得する場合は、-1 を指定します。-1 未満の値は指定できません。
aAttributeNames	属性名のセット	in	取得したいビジネスプロセス定義の属性名のセットを指定します。 取得属性名を指定しない場合は、サイズ 0 の属性名のセットまたは null を指定します。取得属性名に null は指定できません。取得属性名に指定されていない場合でも、ビジネスプロセス定義の ID は必ず取得します。取得可能な属性は、

仮引数名	名称	in/out	説明
			CIWProcessDefinition.AttributeName 列挙型で指定できる属性です。

## 戻り値

取得したビジネスプロセス定義オブジェクトのリストを返します。

## 例外

getProcessDefinitionsList で発生する例外を次の表に示します。

表 5-107 getProcessDefinitionsList で発生する例外

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合
CIWTransientException	一時的なエラーが発生した場合

## 注意事項

- 指定したフィルター条件を満たすビジネスプロセス定義オブジェクトが存在しない場合は、空のリストが返されます。
- 不正な引数を指定した場合は、例外 (java.lang.IllegalArgumentException) が発生します。
- オフセットに 0, 50, 100などを指定し、複数回に分けて取得した場合は、ビジネスプロセス定義の重複や漏れが発生することがあります。

## getProcessInstance

インタフェース名：CIWServer

## 構文

```
CIWProcessInstance getProcessInstance(
    java.lang.Integer aProcessInstanceID,
    java.util.Set<CIWProcessInstance.AttributeName> aAttributeNames
)
throws CIWFatalException,
        CIWTransientException,
        CIWEntityNotExistException
```

## 機能

案件の ID を指定して、案件オブジェクトを取得します。

## 引数

getProcessInstance の引数を次の表に示します。

表 5-108 getProcessInstance の引数

仮引数名	名称	in/out	説明
aProcessInstanceID	案件の ID	in	案件の ID を指定します。 null は指定できません。
aAttributeNames	属性名のセット	in	取得したい案件の属性名のセットを指定します。 取得属性名を指定しない場合は、サイズ 0 の属性名のセットまたは null を指定します。取得属性名に null は指定できません。取得できる属性は、CIWProcessInstance.AttributeName 列挙型で指定できる属性です。

## 戻り値

指定した案件の ID を持つ案件オブジェクトを返します。

## 例外

getProcessInstance で発生する例外を次の表に示します。

表 5-109 getProcessInstance で発生する例外

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合
CIWTransientException	一時的なエラーが発生した場合
CIWEntityNotExistException	処理しようとしたオブジェクトが存在しない場合

## 注意事項

- 案件の ID に null を指定した場合は、例外 (java.lang.IllegalArgumentException) が発生します。
- 指定した条件を満たす案件オブジェクトが存在しない場合は、例外 (CIWEntityNotExistException) が発生します。
- 取得属性名を何も指定していない場合、指定した案件の ID を設定しただけの案件オブジェクトを生成して返します。

指定した案件の ID が 不正な場合は、案件オブジェクトのメソッドを呼び出したときに、エラーになることがあります。

# getProcessInstancesList

インタフェース名：CIWServer

## 構文

```
java.util.List<CIWProcessInstance> getProcessInstancesList(  
    java.lang.String aFilter,  
    java.lang.String aSort,  
    int aOffset,  
    int aMaxCount,  
    java.util.Set<CIWProcessInstance.AttributeName> aAttributeNames  
)  
throws CIWFatalException,  
    CIWTransientException
```

## 機能

指定したフィルター条件を満たす案件オブジェクトのリストを取得します。

## 引数

getProcessInstancesList の引数を次の表に示します。

表 5-110 getProcessInstancesList の引数 (CIWServer の場合)

仮引数名	名称	in/out	説明
aFilter	フィルター条件	in	案件を取得する場合のフィルター条件を指定します。フィルター条件を指定しない場合は、null を指定します。フィルター条件の指定方法については、「付録 B 参照系 API で指定できる属性」を参照してください。
aSort	ソート条件	in	案件を取得する場合のソート条件を指定します。ソート条件を指定しない場合は、null を指定します。ソート条件の指定方法については、「付録 B 参照系 API で指定できる属性」を参照してください。
aOffset	オフセット	in	案件を取得する場合のオフセットを指定します。オフセットは先頭を 0 として指定します。0 未満の値は指定できません。
aMaxCount	最大取得数	in	案件を取得する場合の最大取得数を指定します。すべてを取得する場合は、-1 を指定します。-1 未満の値は指定できません。
aAttributeNames	属性名のセット	in	取得したい案件の属性名のセットを指定します。取得属性名を指定しない場合は、サイズ 0 の属性名のセットまたは null を指定します。取得属性名に null は指定できません。取得属性名に指定されていない場合でも、案件の ID は必ず取得します。取得可能な属性は、



仮引数名	名称	in/out	説明
			CIWProcessInstance.AttributeName 列挙型で指定できる属性です。

## 戻り値

取得した案件のリストを返します。

## 例外

getProcessInstancesList で発生する例外を次の表に示します。

表 5-111 getProcessInstancesList で発生する例外 (CIWServer の場合)

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合
CIWTransientException	一時的なエラーが発生した場合

## 注意事項

- 指定したフィルター条件を満たす案件オブジェクトが存在しない場合は、空のリストが返されます。
- 不正な引数を指定した場合は、例外 (java.lang.IllegalArgumentException) が発生します。
- オフセットに 0, 50, 100などを指定し、複数回に分けて取得した場合は、案件の重複や漏れが発生することがあります。

## getWorkDefinition

インタフェース名：CIWServer

## 構文

```
CIWorkDefinition getWorkDefinition(
    java.lang.Integer aProcessDefinitionID,
    java.lang.Integer aWorkDefinitionID,
    java.util.Set<CIWorkDefinition.AttributeName> aAttributeNames
)
throws CIWFatalException,
       CIWTransientException,
       CIWEntityNotExistException
```

## 機能

ビジネスプロセス定義の ID および作業定義の ID を指定して、作業定義オブジェクトを取得します。

## 引数

getWorkDefinition の引数を次の表に示します。

表 5-112 getWorkDefinition の引数 (CIWServer の場合)

仮引数名	名称	in/out	説明
aProcessDefinitionID	ビジネスプロセス定義の ID	in	ビジネスプロセス定義の ID を指定します。 null は指定できません。
aWorkDefinitionID	作業定義の ID	in	作業定義の ID を指定します。 null は指定できません。
aAttributeNames	属性名のセット	in	取得したい作業定義の属性名のセットを指定します。 取得属性名を指定しない場合は、サイズ 0 の属性名のセットまたは null を指定します。取得属性名に null は指定できません。取得できる属性は、CIWWorkDefinition.AttributeName 列挙型で指定できる属性です。

## 戻り値

指定したビジネスプロセス定義の ID と作業定義の ID を持つ作業定義オブジェクトを返します。

## 例外

getWorkDefinition で発生する例外を次の表に示します。

表 5-113 getWorkDefinition で発生する例外 (CIWServer の場合)

発生する例外	説明
<a href="#">CIWFatalException</a>	処理を続行できない障害が発生した場合
<a href="#">CIWTransientException</a>	一時的なエラーが発生した場合
<a href="#">CIWEntityNotExistException</a>	処理しようとしたオブジェクトが存在しない場合

## 注意事項

- ビジネスプロセス定義の ID に null を指定した場合は、例外 (java.lang.IllegalArgumentException) が発生します。
- 作業定義の ID に null を指定した場合は、例外 (java.lang.IllegalArgumentException) が発生します。
- 指定した条件を満たす作業定義オブジェクトが存在しない場合は、例外 (CIWEntityNotExistException) が発生します。
- 取得属性名を何も指定していない場合は、指定したビジネスプロセス定義の ID および作業定義の ID を設定しただけの作業定義オブジェクトを生成して返します。

指定したビジネスプロセス定義の ID または作業定義の ID が不正な場合は、作業定義オブジェクトのメソッドを呼び出したときに、エラーになることがあります。

# getWorkDefinitionsList

インタフェース名：CIWServer

## 構文

```
java.util.List<CIWorkDefinition> getWorkDefinitionsList(  
    java.lang.String aFilter,  
    java.lang.String aSort,  
    int aOffset,  
    int aMaxCount,  
    java.util.Set<CIWorkDefinition.AttributeName> aAttributeNames  
)  
throws CIWFatalException,  
    CIWTransientException
```

## 機能

指定したフィルター条件を満たす作業定義オブジェクトのリストを取得します。

## 引数

getWorkDefinitionsList の引数を次の表に示します。

表 5-114 getWorkDefinitionsList の引数

仮引数名	名称	in/out	説明
aFilter	フィルター条件	in	作業定義を取得する際のフィルター条件を指定します。フィルター条件を指定しない場合は、null を指定します。フィルター条件の指定方法については、「付録 B 参照系 API で指定できる属性」を参照してください。
aSort	ソート条件	in	作業定義を取得する際のソート条件を指定します。ソート条件を指定しない場合は、null を指定します。ソート条件の指定方法については、「付録 B 参照系 API で指定できる属性」を参照してください。
aOffset	オフセット	in	作業定義を取得する際のオフセットを指定します。オフセットは先頭を 0 として指定します。0 未満の値は指定できません。
aMaxCount	最大取得数	in	作業定義を取得する際の最大取得数を指定します。すべてを取得する場合は、-1 を指定します。-1 未満の値は指定できません。
aAttributeNames	属性名のセット	in	取得したい作業定義の属性名のセットを指定します。取得属性名を指定しない場合は、サイズ 0 の属性名のセットまたは null を指定します。取得属性名に null は指定できません。取得属性名に指定されていない場合でも、作業定義が所属するビジネスプロセス定義の ID と作業定義の ID は必ず取得します。取得可能な属性は、

仮引数名	名称	in/out	説明
			CIWorkDefinition.AttributeName 列挙型で指定できる属性です。

## 戻り値

取得した作業定義オブジェクトのリストを返します。

## 例外

getWorkDefinitionsList で発生する例外を次の表に示します。

表 5-115 getWorkDefinitionsList で発生する例外

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合
CIWTransientException	一時的なエラーが発生した場合

## 注意事項

- 指定したフィルター条件を満たす作業定義オブジェクトが存在しない場合、空のリストが返されます。
- 不正な引数を指定した場合は、例外 (java.lang.IllegalArgumentException) が発生します。
- オフセットに 0, 50, 100 のように指定して、複数回に分けて取得した場合、作業定義の重複や漏れが発生することがあります。

## getWorkItem

インタフェース名：CIWServer

## 構文

```
CIWorkItem getWorkItem(
    java.lang.Integer aProcessInstanceID,
    java.lang.Integer aWorkItemID,
    java.util.Set<CIWorkItem.AttributeName> aAttributeNames
)
throws CIWFatalException,
       CIWTransientException,
       CIWEntityNotExistException
```

## 機能

案件の ID と作業の ID を指定して、作業オブジェクトを取得します。

## 引数

getWorkItem の引数を次の表に示します。

表 5-116 getWorkItem の引数

仮引数名	名称	in/out	説明
aProcessInstanceID	案件の ID	in	案件の ID を指定します。 null は指定できません。
aWorkItemID	作業の ID	in	作業の ID を指定します。 null は指定できません。
aAttributeNames	属性名のセット	in	取得したい作業の属性名のセットを指定します。 取得属性名を指定しない場合は、サイズ 0 の属性名のセットまたは null を指定します。取得属性名に null は指定できません。取得できる属性は、CIWorkItem.AttributeName 列挙型で指定できる属性です。

## 戻り値

指定した案件の ID と作業の ID を持つ作業オブジェクトを返します。

## 例外

getWorkItem で発生する例外を次の表に示します。

表 5-117 getWorkItem で発生する例外

発生する例外	説明
<a href="#">CIWFatalException</a>	処理を続行できない障害が発生した場合
<a href="#">CIWTransientException</a>	一時的なエラーが発生した場合
<a href="#">CIWEntityNotExistException</a>	処理しようとしたオブジェクトが存在しない場合

## 注意事項

- 案件の ID に null を指定した場合は、例外 (java.lang.IllegalArgumentException) が発生します。
- 作業の ID に null を指定した場合は、例外 (java.lang.IllegalArgumentException) が発生します。
- 指定した条件を満たす作業オブジェクトが存在しない場合は、例外 (CIWEntityNotExistException) が発生します。
- 取得属性名を何も指定していない場合は、指定した案件の ID および作業の ID を設定しただけの作業オブジェクトを生成して返します。  
指定した案件の ID または作業の ID が不正な場合は、案件オブジェクトのメソッドを呼び出したときに、エラーとなることがあります。

# getWorkItemsList

インタフェース名：CIWServer

## 構文

```
java.util.List<CIWorkItem> getWorkItemsList(  
    java.lang.String aFilter,  
    java.lang.String aSort,  
    int aOffset,  
    int aMaxCount,  
    java.util.Set<CIWorkItem.AttributeName> aAttributeNames  
)  
    throws CIWFatalException,  
        CIWTransientException
```

## 機能

指定したフィルター条件を満たす作業オブジェクトのリストを取得します。

## 引数

getWorkItemsList の引数を次の表に示します。

表 5-118 getWorkItemsList の引数 (CIWServer の場合)

仮引数名	名称	in/out	説明
aFilter	フィルター条件	in	作業を取得する場合のフィルター条件を指定します。フィルター条件を指定しない場合は、null を指定します。フィルター条件の指定方法については、「 <a href="#">付録 B 参照系 API で指定できる属性</a> 」を参照してください。
aSort	ソート条件	in	作業を取得する場合のソート条件を指定します。ソート条件を指定しない場合は、null を指定します。ソート条件の指定方法については、「 <a href="#">付録 B 参照系 API で指定できる属性</a> 」を参照してください。
aOffset	オフセット	in	作業を取得する場合のオフセットを指定します。オフセットは先頭を 0 として指定します。0 未満の値は指定できません。
aMaxCount	最大取得数	in	作業を取得する場合の最大取得数を指定します。すべてを取得する場合は、-1 を指定します。-1 未満の値は指定できません。
aAttributeNames	属性名のセット	in	取得したい作業の属性名のセットを指定します。取得属性名を指定しない場合は、サイズ 0 の属性名のセットまたは null を指定します。取得属性名に null は指定できません。取得属性名に指定していない場合でも、作業の ID と作業が所属する案件の ID は必ず取得します。取得できる属

仮引数名	名称	in/out	説明
			性は、CIWorkItem.AttributeName 列挙型で指定できる属性です。

## 戻り値

取得した作業オブジェクトのリストを返します。

## 例外

getWorkItemsList で発生する例外を次の表に示します。

表 5-119 getWorkItemsList で発生する例外 (CIWServer の場合)

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合
CIWTransientException	一時的なエラーが発生した場合

## 注意事項

- 指定したフィルター条件を満たす作業オブジェクトが存在しない場合は、空のリストが返されます。
- 不正な引数を指定した場合は、例外 (java.lang.IllegalArgumentException) が発生します。
- オフセットに 0, 50, 100などを指定し、複数回に分けて取得した場合は、作業の重複や漏れが発生することがあります。

## 5.8 CIWorkDefinition (作業定義のインタフェース)

作業定義のインタフェースです。

作業定義オブジェクトに対してメソッドを発行することで、対象となる作業定義オブジェクトに対して処理が行われます。

### クラス定義

```
public interface CIWorkDefinition
```

### 入れ子のクラス

CIWorkDefinition インタフェースの入れ子のクラスを次の表に示します。

表 5-120 CIWorkDefinition インタフェースの入れ子のクラス

クラス名	説明および記述形式
CIWorkDefinition.AttributeName	作業定義属性の属性名の列挙型です。 <pre>static class CIWorkDefinition.AttributeName</pre>
CIWorkDefinition.Type	作業定義の種類の列挙型です。 <pre>static class CIWorkDefinition.Type</pre>

### メソッド

CIWorkDefinition インタフェースのメソッドを次の表に示します。

表 5-121 CIWorkDefinition インタフェースのメソッド

API 名	説明
<a href="#">getActivityDefinitionID</a>	対象となる作業定義が所属する業務ステップ定義の ID を取得します。
<a href="#">getCastingRuleName</a>	対象となる作業定義の振り分けルール名を取得します。
<a href="#">getDescription</a>	対象となる作業の説明を取得します。
<a href="#">getID</a>	対象となる作業定義の ID を取得します。
<a href="#">getName</a>	対象となる作業定義の名称を取得します。
<a href="#">getProcessDefinitionID</a>	対象となる作業定義が所属するビジネスプロセス定義の ID を取得します。
<a href="#">getType</a>	対象となる作業の種類を取得します。

## getActivityDefinitionID

インタフェース名：CIWorkDefinition



## 構文

```
java.lang.Integer getActivityDefinitionID()
    throws CIWFatalException
```

## 機能

対象となる作業定義が所属する業務ステップ定義の ID を取得します。

## 引数

ありません。

## 戻り値

作業定義が所属する業務ステップ定義の ID を返します。

## 例外

getActivityDefinitionID で発生する例外を次の表に示します。

表 5-122 getActivityDefinitionID で発生する例外 (CIWWorkDefinition の場合)

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合

## 注意事項

オブジェクト取得時に作業定義が所属する業務ステップ定義の ID の取得を指定していない場合は、例外 (CIWFatalException) が発生します。

## getCastingRuleName

インタフェース名 : CIWWorkDefinition

## 構文

```
java.lang.String getCastingRuleName()
    throws CIWFatalException
```

## 機能

対象となる作業定義の振り分けルール名を取得します。

## 引数

ありません。

## 戻り値

作業定義の振り分けルール名を返します。

## 例外

getCastingRuleName で発生する例外を次の表に示します。

表 5-123 getCastingRuleName で発生する例外

発生する例外	説明
<a href="#">CIWFatalException</a>	処理を続行できない障害が発生した場合

## 注意事項

- 未設定時には null が返されます。
- オブジェクト取得時に振り分けルール名の取得を指定していない場合は、例外 (CIWFatalException) が発生します。

## getDescription

インタフェース名：CIWWorkDefinition

## 構文

```
java.lang.String getDescription()  
throws CIWFatalException
```

## 機能

対象となる作業の説明を取得します。

## 引数

ありません。

## 戻り値

作業の説明を返します。

## 例外

getDescription で発生する例外を次の表に示します。

表 5-124 getDescription で発生する例外 (CIWorkDefinition の場合)

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合

## 注意事項

- 未設定時には null が返されます。
- オブジェクト取得時に説明の取得を指定していない場合は、例外 (CIWFatalException) が発生します。

## getID

インタフェース名 : CIWorkDefinition

## 構文

```
java.lang.Integer getID()
    throws CIWFatalException
```

## 機能

対象となる作業定義の ID を取得します。

## 引数

ありません。

## 戻り値

作業定義の ID を返します。

## 例外

getID で発生する例外を次の表に示します。

表 5-125 getID で発生する例外 (CIWorkDefinition の場合)

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合

## getName

インタフェース名 : CIWorkDefinition

## 構文

```
java.lang.String getName()  
    throws CIWFatalException
```

## 機能

対象となる作業定義の名称を取得します。

## 引数

ありません。

## 戻り値

作業定義の名称を返します。

## 例外

getName で発生する例外を次の表に示します。

表 5-126 getName で発生する例外 (CIWWorkDefinition の場合)

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合

## 注意事項

オブジェクト取得時に名称の取得を指定していない場合は、例外 (CIWFatalException) が発生します。

## getProcessDefinitionID

インタフェース名 : CIWWorkDefinition

## 構文

```
java.lang.Integer getProcessDefinitionID()  
    throws CIWFatalException
```

## 機能

対象となる作業定義が所属するビジネスプロセス定義の ID を取得します。

## 引数

ありません。

## 戻り値

作業定義が所属するビジネスプロセス定義の ID を返します。

## 例外

getProcessDefinitionID で発生する例外を次の表に示します。

表 5-127 getProcessDefinitionID で発生する例外 (CIWorkDefinition の場合)

発生する例外	説明
<a href="#">CIWFatalException</a>	処理を続行できない障害が発生した場合

## getType

インタフェース名 : CIWorkDefinition

## 構文

```
CIWorkDefinition.Type getType()  
throws CIWFatalException
```

## 機能

対象となる作業の種類を取得します。

## 引数

ありません。

## 戻り値

作業の種類を返します。

## 例外

getType で発生する例外を次の表に示します。

表 5-128 getType で発生する例外 (CIWorkDefinition の場合)

発生する例外	説明
<a href="#">CIWFatalException</a>	処理を続行できない障害が発生した場合

## 注意事項

オブジェクト取得時に種類の取得を指定していない場合は、例外 (CIWFatalException) が発生します。

## 5.9 CIWorkItem (作業のインタフェース)

作業のインタフェースです。

作業オブジェクトに対してメソッドを発行することで、対象となる作業オブジェクトに対して処理が行われます。

### クラス定義

```
public interface CIWorkItem
```

### 入れ子のクラス

CIWorkItem インタフェースの入れ子のクラスを次の表に示します。

表 5-129 CIWorkItem インタフェースの入れ子のクラス

クラス名	説明および記述形式
CIWorkItem.AttributeName	作業属性の属性名の列挙型です。 <code>static class CIWorkItem.AttributeName</code>
CIWorkItem.State	作業の状態の列挙型です。 <code>static class CIWorkItem.State</code>
CIWorkItem.Type	作業の種類列挙型です。 <code>static class CIWorkItem.Type</code>

### メソッド

CIWorkItem インタフェースのメソッドを次の表に示します。

表 5-130 CIWorkItem インタフェースのメソッド

API 名	説明
<code>changeState</code>	対象となる作業の状態を変更します。
<code>complete</code>	対象となる作業を「実行中」状態から「実行済」状態に変更したあと、対象となる作業が所属する業務ステップの終了評価を行います。
<code>evaluate</code>	対象となる作業の完了条件を評価します。
<code>free</code>	作業を返却して振り分けルールを再評価します。
<code>getActivityInstanceID</code>	対象となる作業が所属する業務ステップの ID を取得します。
<code>getClosedDate</code>	対象となる作業の終了日時を取得します。
<code>getCreationDate</code>	対象となる作業の発生日時を取得します。
<code>getDeadline</code>	対象となる作業の処理期限の絶対日時を取得します。
<code>getID</code>	対象となる作業の ID を取得します。
<code>getName</code>	対象となる作業の名称 (作業キー) を取得します。

API 名	説明
getParentProcessInstance	対象となる作業が所属する案件オブジェクトを取得します。
getParticipant	対象となる作業の作業者 ID を取得します。
getPriority	対象となる作業の優先度を取得します。
getProcessDefinitionID	対象となる作業が所属するビジネスプロセス定義の ID を取得します。
getProcessInstanceID	対象となる作業が所属する案件の ID を取得します。
getProcessInstanceName	対象となる作業の案件名（案件キー）を取得します。
getStartDate	対象となる作業の開始日時を取得します。
getState	対象となる作業の状態を取得します。
getType	対象となる作業の種類を取得します。
getWorkDefinition	対象となる作業が所属する作業定義オブジェクトを取得します。
getWorkDefinitionID	対象となる作業定義の ID を取得します。
getWorkDefinitionName	対象となる作業定義の名称を取得します。
perform	対象となる作業を「実行開始可能」状態から「作業者実行」状態にします。
performAndComplete	対象となる作業を「実行開始可能」状態から「作業者実行」状態にし「実行済」状態に変更したあと、該当作業が所属する業務ステップの終了評価を行います。
reassign	対象となる作業の作業者を、指定した別の作業者に割り当てます。
reassignAndPerform	対象となる作業を別の作業者に割り当て、「実行開始可能」状態から「作業者実行」状態にします。
reevaluateCastingRule	対象となる作業の振り分けルールを再評価し、作業者を再割り当てします。
setDeadline	対象となる作業に対して、指定した日時を作業の処理期限の絶対日時として設定します。
setPriority	対象となる作業に対して、指定した優先度を設定します。

## changeState

インタフェース名：CIWorkItem

### 構文

```
void changeState(
    CIWorkItem.State aNewState
)
    throws CIWTransientException,
           CIWTransitionFailedException,
           CIWFatalException,
```

## 機能

対象となる作業の状態を変更します。

## 引数

changeState の引数を次の表に示します。

表 5-131 changeState の引数 (CIWorkItem の場合)

仮引数名	名称	in/out	説明
aNewState	状態	in	変更する作業の状態を指定します。 指定する値には、このインタフェースに定義された CIWorkItem.State 列挙型の定数を指定できます。null は指定できません。

aNewState で指定できる作業の状態を次の表に示します。

表 5-132 aNewState で指定できる作業の状態 (CIWorkItem の場合)

現在の状態		指定できる状態		状態遷移の意味	N	B
定数	意味	定数	意味			
INITIAL	初期	READY	実行開始可能	準備済	○	○
READY	実行開始可能	PERFORMING	作業実行	着手	○	×
PERFORMING	作業実行	READY	実行開始可能	返却 <sup>※1</sup>	○	×
PERFORMING	作業実行	EXECUTED	実行済	完了通知	○	×
EXECUTING	自動実行	EXECUTED	実行済	作業完了	×	○
RUNNING	実行中	INTERMITTED	実行停止	中断	○	○
INTERMITTED	実行停止	RUNNING	実行中 <sup>※2</sup>	再開 <sup>※3</sup>	○	○
—	未終了	TERMINATED	強制終了	強制終了	○	○

### (凡例)

N：一般作業に対する指定の可否

B：組み込み作業（並列作業）に対する指定の可否

定数：CIWorkItem.State 列挙型の定数

○：指定できることを示します。

×：指定できないことを示します。

—：定数がないことを表します。



#### 注※1

アプリケーション自動起動を返却した場合は、アプリケーションを再起動します。

#### 注※2

状態には、「実行中」、それに含まれる「自動実行」および「作業実行」を指定できます。

#### 注※3

指定できる状態のどれかを指定した場合は、次に示す状態に遷移します。

- 一般作業の場合：「作業実行」
- 組み込み作業の場合：「自動実行」

状態遷移については、「付録 C 状態遷移モデル」を参照してください。

## 戻り値

ありません。

## 例外

changeState で発生する例外を次の表に示します。

表 5-133 changeState で発生する例外 (CIWorkItem の場合)

発生する例外	説明
CIWTransientException	一時的なエラーが発生した場合
CIWTransitionFailedException	案件処理中にエラーが発生した場合
CIWFatalException	処理を続行できない障害が発生した場合
CIWEntityNotExistException	処理しようとしたオブジェクトが存在しない場合
CIWStateException	状態の遷移に失敗した場合

## 注意事項

- 対象となる作業が属する案件がエラー案件の場合は、例外 (CIWStateException) が発生します。
- エラー回復作業で本メソッドを呼び出した場合は、例外 (CIWStateException) が発生します。
- 変更する作業状態に null を指定した場合は、例外 (java.lang.IllegalArgumentException) が発生します。
- 作業を強制終了した場合は、対象となる作業が所属する業務ステップの完了条件の評価は行いません。そのため、対象となる作業が所属する業務ステップを直接完了させるか、または CIWActivityInstance#evaluate() を使用して業務ステップの完了条件の評価を実行し、業務ステップの完了または作業の再作成を行う必要があります。CIWActivityInstance#evaluate() を使用して評価を実行する場合は、作業の発生条件や完了条件が適切に定義されている必要があります。

# complete

インタフェース名：CIWorkItem

## 構文

```
void complete()  
    throws CIWTransientException,  
           CIWTransitionFailedException,  
           CIWFatalException,  
           CIWEntityNotExistException,  
           CIWStateException
```

## 機能

対象となる作業を「実行中」状態から「実行済」状態に遷移させ、そのあと作業が所属する業務ステップの終了評価を行います。

## 引数

ありません。

## 戻り値

ありません。

## 例外

complete で発生する例外を次の表に示します。

表 5-134 complete で発生する例外

発生する例外	説明
CIWTransientException	一時的なエラーが発生した場合
CIWTransitionFailedException	案件処理中にエラーが発生した場合
CIWFatalException	処理を続行できない障害が発生した場合
CIWEntityNotExistException	処理しようとしたオブジェクトが存在しない場合
CIWStateException	状態の遷移に失敗した場合

## 注意事項

- 一般作業および組み込み作業で実行できます。エラー回復作業の場合は、エラーになります。
- 対象となる作業が所属する案件がエラー案件の場合は、例外（CIWStateException）が発生します。

# evaluate

インタフェース名：CIWorkItem

## 構文

```
void evaluate()  
    throws CIWTransientException,  
           CIWTransitionFailedException,  
           CIWFatalException,  
           CIWEntityNotExistException,  
           CIWStateException
```

## 機能

対象となる作業の完了条件を評価します。評価の結果、作業が「完了」状態に遷移した場合は、CIWActivityInstance インタフェースの evaluate メソッドと同様に、対象となる作業が所属する業務ステップの評価も行います。

## 引数

ありません。

## 戻り値

ありません。

## 例外

evaluate で発生する例外を次の表に示します。

表 5-135 evaluate で発生する例外 (CIWorkItem の場合)

発生する例外	説明
<a href="#">CIWTransientException</a>	一時的なエラーが発生した場合
<a href="#">CIWTransitionFailedException</a>	案件処理中にエラーが発生した場合
<a href="#">CIWFatalException</a>	処理を続行できない障害が発生した場合
<a href="#">CIWEntityNotExistException</a>	処理しようとしたオブジェクトが存在しない場合
<a href="#">CIWStateException</a>	状態の遷移に失敗した場合

## 注意事項

- 作業は「実行開始可能」状態または「実行中」状態である必要があります。そのほかの場合では例外 (CIWStateException) が発生します。
- 対象となる作業が所属する案件の状態は「実行中」状態または「実行停止」状態である必要があります。そのほかの状態では、例外 (CIWStateException) が発生します。

# free

インタフェース名：CIWorkItem

## 構文

```
public void free(  
)
```

## 機能

作業オブジェクトを「作業実行」状態から「実行開始可能」状態に変更したあと、振り分けルールを再評価して作業者を決定し直します。

## 引数

ありません。

## 戻り値

ありません。

## 例外

free で発生する例外を次の表に示します。

表 5-136 free で発生する例外

発生する例外	説明
<a href="#">CIWTransientException</a>	一時的なエラーが発生した場合
<a href="#">CIWFatalException</a>	処理を続行できない障害が発生した場合
<a href="#">CIWTransitionFailedException</a>	案件処理中にエラーが発生した場合
<a href="#">CIWEntityNotExistException</a>	処理しようとしたオブジェクトが存在しない場合
<a href="#">CIWStateException</a>	状態の遷移に失敗した場合

## 注意事項

- CIWServer クラスのメソッド `allocateWorkItemEx` で取得した `CIWorkItem`（作業）に対して、このメソッドを発行してください。
- 作業生成後に振り分けルールで評価するユーザデータを変更すると、作業生成時に振り分けルールで決定した作業者と、このメソッドの発行時に振り分けルール再評価で決定した作業者が異なる場合があります。

## getActivityInstanceID

インタフェース名：CIWWorkItem

### 構文

```
java.lang.Integer getActivityInstanceID()  
    throws CIWFatalException
```

### 機能

対象となる作業が所属する業務ステップの ID を取得します。

### 引数

ありません。

### 戻り値

作業が所属する業務ステップの ID を返します。

### 例外

getActivityInstanceID で発生する例外を次の表に示します。

表 5-137 getActivityInstanceID で発生する例外 (CIWWorkItem の場合)

発生する例外	説明
<a href="#">CIWFatalException</a>	処理を続行できない障害が発生した場合

### 注意事項

オブジェクト取得時に作業が所属する業務ステップの ID の取得を指定していない場合は、例外 (CIWFatalException) が発生します。

## getClosedDate

インタフェース名：CIWWorkItem

### 構文

```
java.util.Date getClosedDate()  
    throws CIWFatalException
```

## 機能

対象となる作業の終了日時を取得します。

## 引数

ありません。

## 戻り値

作業の終了日時を返します。

## 例外

getClosedDate で発生する例外を次の表に示します。

表 5-138 getClosedDate で発生する例外 (CIWorkItem の場合)

発生する例外	説明
<a href="#">CIWFatalException</a>	処理を続行できない障害が発生した場合

## 注意事項

- 「終了」状態への遷移が発生した場合に値が設定されます。  
遷移の発生前は null が返されます。状態遷移については、「[付録 C 状態遷移モデル](#)」を参照してください。
- オブジェクト取得時に終了日時の取得を指定していない場合は、例外 (CIWFatalException) が発生します。

## getCreationDate

インタフェース名 : CIWorkItem

## 構文

```
java.util.Date getCreationDate()  
throws CIWFatalException
```

## 機能

対象となる作業の発生日時を取得します。

## 引数

ありません。

## 戻り値

作業の発生日時を返します。

## 例外

getCreationDate で発生する例外を次の表に示します。

表 5-139 getCreationDate で発生する例外 (CIWorkItem の場合)

発生する例外	説明
<a href="#">CIWFatalException</a>	処理を続行できない障害が発生した場合

## 注意事項

- 「実行開始可能」状態への遷移が発生した場合に値が設定されます。  
遷移の発生前は null が返されます。状態遷移については、「[付録 C 状態遷移モデル](#)」を参照してください。
- オブジェクト取得時に発生日時の取得を指定していない場合は、例外 (CIWFatalException) が発生します。

## getDeadline

インタフェース名 : CIWorkItem

## 構文

```
java.util.Date getDeadline()  
    throws CIWFatalException
```

## 機能

対象となる作業の処理期限の絶対日時を取得します。

## 引数

ありません。

## 戻り値

作業の処理期限の絶対日時を返します。

## 例外

getDeadline で発生する例外を次の表に示します。

表 5-140 getDeadline で発生する例外 (CIWorkItem の場合)

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合

## 注意事項

- 未設定時には null が返されます。
- オブジェクト取得時に処理期限の絶対日時の取得を指定していない場合は、例外 (CIWFatalException) が発生します。

## getID

インタフェース名：CIWorkItem

## 構文

```
java.lang.Integer getID()
    throws CIWFatalException
```

## 機能

対象となる作業の ID を取得します。

## 引数

ありません。

## 戻り値

作業の ID を返します。

## 例外

getID で発生する例外を次の表に示します。

表 5-141 getID で発生する例外 (CIWorkItem の場合)

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合

## getName

インタフェース名：CIWorkItem



## 構文

```
java.lang.String getName()  
    throws CIWFatalException
```

## 機能

対象となる作業の名称（作業キー）を取得します。

## 引数

ありません。

## 戻り値

作業の名称（作業キー）を返します。

## 例外

getName で発生する例外を次の表に示します。

表 5-142 getName で発生する例外（CIWWorkItem の場合）

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合

## 注意事項

- 未設定時には null が返されます。
- オブジェクト取得時に名称（作業キー）の取得を指定していない場合は、例外（CIWFatalException）が発生します。

## getParentProcessInstance

インタフェース名：CIWWorkItem

## 構文

```
CIWProcessInstance getParentProcessInstance(  
    java.util.Set<CIWProcessInstance.AttributeName> aAttributeNames  
)  
    throws CIWTransientException,  
    CIWFatalException,  
    CIWEntityNotExistException
```

## 機能

対象となる作業が所属する案件オブジェクトを取得します。

## 引数

getParentProcessInstance の引数を次の表に示します。

表 5-143 getParentProcessInstance の引数

仮引数名	名称	in/out	説明
aAttributeNames	属性名のセット	in	取得したい案件の属性名のセットを指定します。 取得属性名を指定しない場合は、サイズ 0 の属性名のセットまたは null を指定します。取得属性名に null は指定できません。取得属性名に指定されていない場合でも、案件の ID は必ず取得します。取得できる属性は、CIWProcessInstance.AttributeName 列挙型で指定できる属性です。

## 戻り値

作業が所属する案件オブジェクトを返します。

## 例外

getParentProcessInstance で発生する例外を次の表に示します。

表 5-144 getParentProcessInstance で発生する例外

発生する例外	説明
CIWTransientException	一時的なエラーが発生した場合
CIWFatalException	処理を続行できないような障害が発生した場合
CIWEntityNotExistException	処理しようとしたオブジェクトが存在しない場合

## getParticipant

インタフェース名：CIWWorkItem

## 構文

```
java.lang.String getParticipant()  
    throws CIWFatalException
```

## 機能

対象となる作業の作業 ID を取得します。

## 引数

ありません。

## 戻り値

作業の作業者 ID を返します。

## 例外

getParticipant で発生する例外を次の表に示します。

表 5-145 getParticipant で発生する例外

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合

## 注意事項

- 未設定時には null が返されます。
- オブジェクト取得時に作業者 ID の取得を指定していない場合は、例外 (CIWFatalException) が発生します。

## getPriority

インタフェース名: CIWWorkItem

## 構文

```
java.lang.Integer getPriority()  
throws CIWFatalException
```

## 機能

対象となる作業の優先度を取得します。

## 引数

ありません。

## 戻り値

作業の優先度を返します。

## 例外

getPriority で発生する例外を次の表に示します。

表 5-146 getPriority で発生する例外 (CIWorkItem の場合)

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合

## 注意事項

- 未設定時には null が返されます。
- オブジェクト取得時に優先度の取得を指定していない場合は、例外 (CIWFatalException) が発生します。

## getProcessDefinitionID

インタフェース名: CIWorkItem

## 構文

```
java.lang.Integer getProcessDefinitionID()
    throws CIWFatalException
```

## 機能

対象となる作業が所属するビジネスプロセス定義の ID を取得します。

## 引数

ありません。

## 戻り値

作業が所属するビジネスプロセス定義の ID を返します。

## 例外

getProcessDefinitionID で発生する例外を次の表に示します。

表 5-147 getProcessDefinitionID で発生する例外 (CIWorkItem の場合)

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合

## 注意事項

オブジェクト取得時に作業が所属するビジネスプロセス定義の ID の取得を指定していない場合は、例外 (CIWFatalException) が発生します。

# getProcessInstanceID

インタフェース名：CIWWWorkItem

## 構文

```
java.lang.Integer getProcessInstanceID()  
throws CIWFatalException
```

## 機能

対象となる作業が所属する案件の ID を取得します。

## 引数

ありません。

## 戻り値

作業が所属する案件の ID を返します。

## 例外

getProcessInstanceID で発生する例外を次の表に示します。

表 5-148 getProcessInstanceID で発生する例外 (CIWWWorkItem の場合)

発生する例外	説明
<a href="#">CIWFatalException</a>	処理を続行できない障害が発生した場合

# getProcessInstanceName

インタフェース名：CIWWWorkItem

## 構文

```
java.lang.String getProcessInstanceName()  
throws CIWFatalException
```

## 機能

対象となる作業の案件名 (案件キー) を取得します。

## 引数

ありません。

## 戻り値

作業の案件名（案件キー）を返します。

## 例外

getProcessInstanceName で発生する例外を次の表に示します。

表 5-149 getProcessInstanceName で発生する例外（CIWWorkItem の場合）

発生する例外	説明
<a href="#">CIWFatalException</a>	処理を続行できない障害が発生した場合

## 注意事項

- 未設定時には null が返されます。
- オブジェクト取得時に案件名（案件キー）の取得を指定していない場合は、例外（CIWFatalException）が発生します。

## getStartDate

インタフェース名：CIWWorkItem

## 構文

```
java.util.Date getStartDate()  
throws CIWFatalException
```

## 機能

対象となる作業の開始日時を取得します。

## 引数

ありません。

## 戻り値

作業の開始日時を返します。

## 例外

getStartDate で発生する例外を次の表に示します。

表 5-150 `getStartDate` で発生する例外 (CIWorkItem の場合)

発生する例外	説明
<code>CIWFatalException</code>	処理を続行できない障害が発生した場合

## 注意事項

- 「実行中」状態に含まれる「作業中実行」状態、または「自動実行」状態に遷移が発生した場合に値が設定されます。  
遷移の発生前は `null` が返されます。状態遷移については、「付録 C 状態遷移モデル」を参照してください。
- オブジェクト取得時に開始日時の取得を指定していない場合は、例外 (`CIWFatalException`) が発生します。

## getState

インタフェース名: `CIWorkItem`

### 構文

```
CIWorkItem.State getState()
    throws CIWFatalException
```

### 機能

対象となる作業の状態を取得します。

### 引数

ありません。

### 戻り値

作業の状態を返します。

### 例外

`getState` で発生する例外を次の表に示します。

表 5-151 `getState` で発生する例外 (CIWorkItem の場合)

発生する例外	説明
<code>CIWFatalException</code>	処理を続行できない障害が発生した場合

## 注意事項

オブジェクト取得時に状態の取得を指定していない場合は、例外（CIWFatalException）が発生します。

## getType

インタフェース名：CIWorkItem

## 構文

```
CIWorkItem.Type getType()  
    throws CIWFatalException
```

## 機能

対象となる作業の種類を取得します。

## 引数

ありません。

## 戻り値

作業の種類を返します。

## 例外

getType で発生する例外を次の表に示します。

表 5-152 getType で発生する例外（CIWorkItem の場合）

発生する例外	説明
<a href="#">CIWFatalException</a>	処理を続行できない障害が発生した場合

## 注意事項

オブジェクト取得時に種類の取得を指定していない場合は、例外（CIWFatalException）が発生します。

## getWorkDefinition

インタフェース名：CIWorkItem



## 構文

```
CIWorkDefinition getWorkDefinition(  
    java.util.Set<CIWorkDefinition.AttributeName> aAttributeNames  
)  
    throws CIWTransientException,  
           CIWFatalException,  
           CIWEntityNotExistException
```

## 機能

対象となる作業が所属する作業定義オブジェクトを取得します。

## 引数

getWorkDefinition の引数を次の表に示します。

表 5-153 getWorkDefinition の引数 (CIWorkItem の場合)

仮引数名	名称	in/out	説明
aAttributeNames	属性名のセット	in	取得したい作業定義の属性名のセットを指定します。 取得属性名を指定しない場合は、サイズ 0 の属性名のセット または null を指定します。取得属性名に null は指定できま せん。取得属性名に指定されていない場合でも、作業が所属 するビジネスプロセス定義の ID と作業定義の ID は必ず取 得します。取得できる属性は、 CIWorkDefinition.AttributeName 列挙型で指定できる 属性です。

## 戻り値

作業が所属する作業定義オブジェクトを返します。

## 例外

getWorkDefinition で発生する例外を次の表に示します。

表 5-154 getWorkDefinition で発生する例外 (CIWorkItem の場合)

発生する例外	説明
<a href="#">CIWTransientException</a>	一時的なエラーが発生した場合
<a href="#">CIWFatalException</a>	処理を続行できない障害が発生した場合
<a href="#">CIWEntityNotExistException</a>	処理しようとしたオブジェクトが存在しない場合

## 注意事項

オブジェクト取得時に作業が所属するビジネスプロセス定義の ID と作業定義の ID の取得を指定してい  
ない場合は、例外 (CIWFatalException) が発生します。

## getWorkDefinitionID

インタフェース名：CIWWorkItem

### 構文

```
java.lang.Integer getWorkDefinitionID()  
    throws CIWFatalException
```

### 機能

対象となる作業定義の ID を取得します。

### 引数

ありません。

### 戻り値

作業定義の ID を返します。

### 例外

getWorkDefinitionID で発生する例外を次の表に示します。

表 5-155 getWorkDefinitionID で発生する例外

発生する例外	説明
<a href="#">CIWFatalException</a>	処理を続行できない障害が発生した場合

### 注意事項

- アドホック生成された作業またはエラー回復作業の場合は null が返されます。
- オブジェクト取得時に作業定義の ID の取得を指定していない場合は、例外（CIWFatalException）が発生します。

## getWorkDefinitionName

インタフェース名：CIWWorkItem

### 構文

```
java.lang.String getWorkDefinitionName()  
    throws CIWFatalException
```

## 機能

対象となる作業定義の名称を取得します。

## 引数

ありません。

## 戻り値

作業定義の名称を返します。

## 例外

getWorkDefinitionName で発生する例外を次の表に示します。

表 5-156 getWorkDefinitionName で発生する例外

発生する例外	説明
<a href="#">CIWFatalException</a>	処理を続行できない障害が発生した場合

## 注意事項

オブジェクト取得時に作業定義の名称の取得を指定していない場合は、例外 (CIWFatalException) が発生します。

## perform

インタフェース名：CIWorkItem

## 構文

```
void perform()
    throws CIWTransientException,
           CIWFatalException,
           CIWEntityNotExistException,
           CIWStateException
```

## 機能

対象となる作業を「実行開始可能」状態から「作業実行」状態にします。

## 引数

ありません。

## 戻り値

ありません。

## 例外

perform で発生する例外を次の表に示します。

表 5-157 perform で発生する例外

発生する例外	説明
<code>CIWTransientException</code>	一時的なエラーが発生した場合
<code>CIWFatalException</code>	処理を続行できない障害が発生した場合
<code>CIWEntityNotExistException</code>	処理しようとしたオブジェクトが存在しない場合
<code>CIWStateException</code>	状態の遷移に失敗した場合

## 注意事項

- 一般作業およびエラー回復作業で実行できます。
- 組み込み作業の場合はエラーになります。

## performAndComplete

インタフェース名：CIWorkItem

## 構文

```
void performAndComplete()  
    throws CIWTransientException,  
           CIWTransitionFailedException,  
           CIWFatalException,  
           CIWEntityNotExistException,  
           CIWStateException
```

## 機能

対象となる作業を「実行開始可能」状態から「作業中」状態へ遷移させて「実行済」状態に変更し、そのあと作業が所属する業務ステップの終了評価を行います。

perform メソッドと complete メソッドを連続して実行する場合と同様に処理を行います。

## 引数

ありません。

## 戻り値

ありません。

## 例外

performAndComplete で発生する例外を次の表に示します。

表 5-158 performAndComplete で発生する例外

発生する例外	説明
<a href="#">CIWTransientException</a>	一時的なエラーが発生した場合
<a href="#">CIWTransitionFailedException</a>	案件処理中にエラーが発生した場合
<a href="#">CIWFatalException</a>	処理を続行できない障害が発生した場合
<a href="#">CIWEntityNotExistException</a>	処理しようとしたオブジェクトが存在しない場合
<a href="#">CIWStateException</a>	状態の遷移に失敗した場合

## 注意事項

- 一般作業で実行可能です。エラー回復作業または組み込み作業の場合はエラーとなります。
- 対象となる作業が所属する案件がエラー案件の場合は、例外（[CIWStateException](#)）が発生します。

## reassign

インタフェース名：CIWWorkItem

## 構文

```
void reassign(  
    java.lang.String aSource,  
    java.lang.String aTarget  
)  
    throws CIWTransientException,  
           CIWFatalException,  
           CIWEntityNotExistException,  
           CIWStateException
```

## 機能

対象となる作業の作業者を、指定した別の作業者に割り当てます。

## 引数

reassign の引数を次の表に示します。

表 5-159 reassign の引数

仮引数名	名称	in/out	説明
aSource	変更する前の作業 ID	in	変更する前の作業 ID を指定します。 未設定の場合は、null を指定します。空文字列は指定できません。
aTarget	新しい作業 ID	in	新しい作業 ID を<SYSTEMID>_WORK_ITEM テーブルの Participant カラムのバイト数以下で指定します。 null を指定した場合、新しい作業 ID は未設定となります。空文字列は指定できません。

## 戻り値

ありません。

## 例外

reassign で発生する例外を次の表に示します。

表 5-160 reassign で発生する例外

発生する例外	説明
<a href="#">CIWTransientException</a>	一時的なエラーが発生した場合
<a href="#">CIWFatalException</a>	処理を続行できない障害が発生した場合
<a href="#">CIWEntityNotExistException</a>	処理しようとしたオブジェクトが存在しない場合
<a href="#">CIWStateException</a>	状態の遷移に失敗した場合

## 注意事項

- 作業が「未終了」状態の場合だけ変更できます。
- 一般作業およびエラー回復作業で実行できます。
- 組み込み作業の場合は、例外（[CIWStateException](#)）が発生します。
- 引数 aSource に指定した作業 ID と対象となる作業の作業 ID が異なる場合は、例外（[CIWStateException](#)）が発生します。
- 作業 ID に空文字列を指定した場合は、例外（[java.lang.IllegalArgumentException](#)）が発生します。
- 作業 ID に<SYSTEMID>\_WORK\_ITEM テーブルの Participant カラムのバイト数を超える文字列を指定した場合は、例外（[java.lang.IllegalArgumentException](#)）が発生します。

## reassignAndPerform

インタフェース名：CIWWorkItem

## 構文

```
void reassignAndPerform(  
    java.lang.String aSource,  
    java.lang.String aTarget  
)  
    throws CIWTransientException,  
           CIWFatalException,  
           CIWEntityNotExistException,  
           CIWStateException
```

## 機能

対象となる作業を別の作業者に割り当て、「実行開始可能」状態から「作業実行」状態へ遷移させます。

reassign メソッドと perform メソッドを連続して実行する場合と同様に処理を行います。

## 引数

reassignAndPerform の引数を次の表に示します。

表 5-161 reassignAndPerform の引数

仮引数名	名称	in/out	説明
aSource	変更する前の作業 ID	in	変更する前の作業 ID を指定します。 未設定の場合は、null を指定します。空文字列は指定できません。
aTarget	新しい作業 ID	in	新しい作業 ID を<SYSTEMID>_WORK_ITEM テーブルの Participant カラムのバイト数以下で指定します。 null を指定した場合、新しい作業 ID は未設定になります。空文字列は指定できません。

## 戻り値

ありません。

## 例外

reassignAndPerform で発生する例外を次の表に示します。

表 5-162 reassignAndPerform で発生する例外

発生する例外	説明
<a href="#">CIWTransientException</a>	一時的なエラーが発生した場合
<a href="#">CIWFatalException</a>	処理を続行できない障害が発生した場合
<a href="#">CIWEntityNotExistException</a>	処理しようとしたオブジェクトが存在しない場合
<a href="#">CIWStateException</a>	状態の遷移に失敗した場合

## 注意事項

- 対象となる作業は、「実行開始可能」状態である必要があります。
- 一般作業およびエラー回復作業の場合だけ実行できます。
- 組み込み作業の場合は、例外（CIWStateException）が発生します。
- 対象となる作業が「作業実行」状態の場合は、例外（CIWStateException）が発生します。
- 対象となる作業の状態が「実行開始可能」状態、または「作業実行」状態でない場合は、例外（CIWStateException）が発生します。
- 引数 aSource に指定した作業 ID と対象となる作業の作業 ID が異なる場合は、例外（CIWStateException）が発生します。
- 作業 ID に空文字を指定した場合は、例外（java.lang.IllegalArgumentException）が発生します。
- 作業 ID に <SYSTEMID>\_WORK\_ITEM テーブルの Participant カラムのバイト数を超える文字列を指定した場合は、例外（java.lang.IllegalArgumentException）が発生します。

## reevaluateCastingRule

インタフェース名：CIWWorkItem

### 構文

```
void reevaluateCastingRule()  
    throws CIWTransientException,  
           CIWEntityNotExistException,  
           CIWTransitionFailedException,  
           CIWFatalException,  
           CIWStateException
```

### 機能

対象となる作業の振り分けルールを再評価し、作業者を再割り当てします。

### 引数

ありません。

### 戻り値

ありません。

### 例外

reevaluateCastingRule で発生する例外を次の表に示します。



表 5-163 reevaluateCastingRule で発生する例外

発生する例外	説明
CIWTransientException	一時的なエラーが発生した場合
CIWEntityNotExistException	処理しようとしたオブジェクトが存在しない場合
CIWTransitionFailedException	案件処理中にエラーが発生した場合
CIWFatalException	処理を続行できない障害が発生した場合
CIWStateException	状態の遷移に失敗した場合

## 注意事項

- 対象となる作業は、「初期」状態以外の「未終了」状態である必要があります。そのほかの状態の場合は、例外（CIWStateException）が発生します。
- 組み込み作業およびエラー回復作業でこのメソッドを呼び出した場合は、例外（CIWStateException）が発生します。
- 振り分けルールが設定されていない一般作業は作業者を変更しません。
- 振り分けルール評価の結果、作業者候補が1人も見つからなかった場合は、例外（CIWTransitionFailedException）が発生します。
- 対象となる作業が所属する案件がエラー案件の場合は、例外（CIWStateException）が発生します。

## setDeadline

インタフェース名：CIWWorkItem

## 構文

```
void setDeadline(
    java.util.Date aNewDeadline
)
throws CIWTransientException,
       CIWStateException,
       CIWFatalException,
       CIWEntityNotExistException
```

## 機能

対象となる作業に対して、指定した日時を作業の処理期限の絶対日時として設定します。

## 引数

setDeadline の引数を次の表に示します。

表 5-164 setDeadline の引数 (CIWorkItem の場合)

仮引数名	名称	in/out	説明
aNewDeadline	変更後の処理期限	in	変更後の処理期限を指定します。 null を指定した場合は、処理期限は未設定となります。処理期限の絶対日時として指定できる値は、CIWUtil#convertDateToString(Date)で指定できる Date 値と同じです。処理期限はミリ秒以下については切り捨てて設定します。

## 戻り値

ありません。

## 例外

setDeadline で発生する例外を次の表に示します。

表 5-165 setDeadline で発生する例外 (CIWorkItem の場合)

発生する例外	説明
CIWTransientException	一時的なエラーが発生した場合
CIWStateException	状態の遷移に失敗した場合
CIWFatalException	処理を続行できない障害が発生した場合
CIWEntityNotExistException	処理しようとしたオブジェクトが存在しない場合

## 注意事項

- 作業が「未終了」状態の場合だけ変更できます。
- 作業が「未終了」状態以外の場合は、例外 (CIWStateException) が発生します。

## setPriority

インタフェース名: CIWorkItem

## 構文

```
void setPriority(
    java.lang.Integer aNewPriority
)
    throws CIWTransientException,
           CIWStateException,
           CIWFatalException,
           CIWEntityNotExistException
```

## 機能

対象となる作業に対して、指定した優先度を設定します。

## 引数

setPriority の引数を次の表に示します。

表 5-166 setPriority の引数 (CIWorkItem の場合)

仮引数名	名称	in/out	説明
aNewPriority	変更後の優先度	in	変更後の優先度を指定します。 null を指定した場合は、優先度は未設定となります。

## 戻り値

ありません。

## 例外

setPriority で発生する例外を次の表に示します。

表 5-167 setPriority で発生する例外 (CIWorkItem の場合)

発生する例外	説明
<a href="#">CIWTransientException</a>	一時的なエラーが発生した場合
<a href="#">CIWStateException</a>	状態の遷移に失敗した場合
<a href="#">CIWFatalException</a>	処理を続行できない障害が発生した場合
<a href="#">CIWEntityNotExistException</a>	処理しようとしたオブジェクトが存在しない場合

## 注意事項

- 作業が「未終了」状態の場合だけ変更できます。
- 作業が「未終了」状態以外の場合は、例外 (CIWStateException) が発生します。

# 6

## その他のクラス

この章では、その他の API について説明します。

## 6.1 記述形式

---

### 6.1.1 その他のクラスの記述形式

その他のクラスは、次に示す形式を基本として説明しています。

#### クラス定義

クラス定義を示します。

#### 入れ子のクラス

入れ子のクラスを説明します。

#### メソッド

クラスのメソッドを説明します。

### 6.1.2 その他のクラスが提供するメソッドの記述形式

その他のクラスが提供するメソッドは、次に示す形式を基本として説明しています。

#### メソッド名

メソッド名を示します。

#### クラス名

メソッドが所属するクラス名を示します。

#### 構文

メソッドの構文を示します。

#### 機能

メソッドの機能を説明します。

#### 引数

メソッドの引数を説明します。

#### 戻り値

メソッドが返す戻り値を示します。

#### 例外

メソッドに発生する例外を説明します。

#### 注意事項

メソッドの注意事項を説明します。

## 6.2 CIWConnectionCoordinator (CIWServer と JDBC コネクションの関連づけを管理するクラス)

JDBC コネクションと CIWServer オブジェクトの関連づけをするクラスです。

### クラス定義

```
public final class CIWConnectionCoordinator extends java.lang.Object
```

### メソッド

CIWConnectionCoordinator クラスのメソッドを次の表に示します。

表 6-1 CIWConnectionCoordinator クラスのメソッド

メソッド	説明
<a href="#">attachDatabaseConnection</a>	トランザクションを開始している JDBC コネクションを CIWServer オブジェクトに関連づけます。
<a href="#">detachDatabaseConnection</a>	JDBC コネクションと CIWServer オブジェクトとの関連づけを解除します。

### java.lang.Object クラスから継承されたメソッド

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## attachDatabaseConnection

クラス名 : CIWConnectionCoordinator

### 構文

```
public void attachDatabaseConnection(  
    java.sql.Connection aDBConnection  
)  
    throws CIWFatalException,  
           CIWSQLTransientException
```

### 機能

トランザクションを開始している JDBC コネクションを CIWServer オブジェクトに関連づけます。

### 引数

attachDatabaseConnection の引数を次の表に示します。

表 6-2 attachDatabaseConnection の引数

仮引数名	名称	in/out	説明
aDBConnection	JDBC コネクション	in	CIWServer オブジェクトに関連づける JDBC コネクションを指定します。 null は指定できません。

## 戻り値

ありません。

## 例外

attachDatabaseConnection で発生する例外を次の表に示します。

表 6-3 attachDatabaseConnection で発生する例外

クラス名	説明
CIWFatalException	処理を続行できない障害が発生した場合
CIWSQLTransientException	DB アクセスで一時的エラーが発生した場合

## 注意事項

- 次に示す場合は、例外 (CIWFatalException) が発生します。
  - 引数に渡す JDBC コネクションに null を指定した場合
  - 空文字列引数に渡す JDBC コネクションの自動コミットモードが有効になっている場合
  - このメソッドを実行する前に CIWServer オブジェクトの API を発行した場合
- このメソッドを実行したあとは、関連づけを解除するまで JDBC コネクションを切断したり、トランザクションを終了したりしないでください。

## detachDatabaseConnection

クラス名：CIWConnectionCoordinator

## 構文

```
public void detachDatabaseConnection()
```

## 機能

JDBC コネクションと CIWServer オブジェクトとの関連づけを解除します。

すでに関連づけを解除していた場合は何もしません。

## 引数

ありません。

## 戻り値

ありません。

## 例外

ありません。

## 注意事項

トランザクションを終了する前に、このメソッドを実行し、JDBC コネクションの関連づけを解除してください。このメソッドを実行しなかった場合の動作は保証しません。



## 6.3 CIWUtil (各種ユーティリティ機能を提供するクラス)

各種ユーティリティ機能を提供するクラスです。

### クラス定義

```
public final class CIWUtil extends java.lang.Object
```

### 入れ子のクラス

CIWUtil の入れ子のクラスを次の表に示します。

表 6-4 CIWUtil の入れ子のクラス

クラス名	説明
CIWUtil.InfinityDate	無限遠の日付の列挙型です。

### メソッド

CIWUtil クラスのメソッドを次の表に示します。

表 6-5 CIWUtil クラスのメソッド

メソッド	説明
convertDateToString	引数で渡された CIWUtil.InfinityDate 列挙型の定数を、フィルター条件の指定用 java.lang.String 型の文字列へ変換します。
convertDateToString	引数で渡された Date 型の日付情報を、フィルター条件の指定用 java.lang.String 型の文字列へ変換します。

### java.lang.Object クラスから継承されたメソッド

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## convertDateToString

クラス名：CIWUtil

### 構文

```
public static java.lang.String convertDateToString(  
    CIWUtil.InfinityDate aInfinityDate  
)
```

### 機能

引数で渡された CIWUtil.InfinityDate 列挙型の定数を、フィルター条件の指定用 java.lang.String 型の文字列に変換します。

## 引数

convertDateToString の引数を次の表に示します。

表 6-6 convertDateToString の引数

仮引数名	名称	in/out	説明
aInfinityDate	CIWUtil.InfinityDate 列挙型の定数	in	CIWUtil.InfinityDate 列挙型の定数を指定します。 null は指定できません。

## 戻り値

フィルター条件指定用の文字列を返します。

## 例外

ありません。

## 注意事項

列挙型の定数に null を指定した場合は、例外 (java.lang.IllegalArgumentException) が発生します。

## convertDateToString

クラス名 : CIWUtil

## 構文

```
public static java.lang.String convertDateToString(  
    java.util.Date aDate  
)
```

## 機能

引数で渡された Date 型の日付情報を、フィルター条件の指定用 java.lang.String 型の文字列に変換します。

## 引数

convertDateToString の引数を次の表に示します。

表 6-7 convertDateToString の引数

仮引数名	名称	in/out	説明
aDate	Date 型の日付情報	in	Date 型の日付情報を指定します。 null は指定できません。

## 戻り値

フィルター条件指定用の文字列を返します。

## 例外

ありません。

## 注意事項

- 無限遠の過去および無限遠の未来を表す Date 型オブジェクトを指定することもできます。無限遠の過去および無限遠の未来については、CIWUtil.InfinityDate を参照してください。
- 指定した日付が 1970/01/02 00:00:00 GMT から 292278993/12/31 00:00:00 GMT の範囲外であった場合は、例外 (java.lang.IllegalArgumentException) が発生します。
- 日付情報に null を指定した場合は、例外 (java.lang.IllegalArgumentException) が発生します。

# 7

## 列挙型

この章では、パッケージ「jp.co.Hitachi.soft.csciw」が提供する列挙型について説明します。

## 7.1 記述形式

---

### 7.1.1 列挙型の記述形式

列挙型は、次に示す形式を基本として説明しています。

#### 概要

列挙型の概要を説明します。

#### 定義

列挙型の定義を示します。

#### すべての実装されたインタフェース

列挙型のすべての実装されたインタフェースを示します。

#### 含まれているインタフェース

列挙型の含まれているインタフェースを示します。

#### 定数

列挙型の定数を説明します。

#### メソッド

列挙型のメソッドを説明します。

### 7.1.2 列挙型が提供するメソッドの記述形式

列挙型が提供するメソッドは、次に示す形式を基本として説明しています。

#### メソッド名

メソッド名を示します。

#### 列挙型

メソッドが所属する列挙型を示します。

#### 構文

メソッドの構文を示します。

#### 機能

メソッドの機能を説明します。

#### 引数

メソッドの引数を説明します。

#### オーバーライド

メソッドのオーバーライドを示します。

## 戻り値

メソッドが返す戻り値を示します。

## 例外

メソッドに発生する例外を説明します。

## 注意事項

メソッドの注意事項を説明します。

## 7.2 CIWActivityDefinition.AttributeName (業務ステップ定義属性の属性名の列挙型)

### 概要

業務ステップ定義属性の属性名の列挙型です。

### 定義

```
public static enum CIWActivityDefinition.AttributeName
    extends java.lang.Enum<CIWActivityDefinition.AttributeName>
```

### すべての実装されたインタフェース

```
java.io.Serializable,
java.lang.Comparable<CIWActivityDefinition.AttributeName>
```

### 含まれているインタフェース

```
CIWActivityDefinition
```

### 定数

CIWActivityDefinition.AttributeName の列挙型定数を次の表に示します。

表 7-1 CIWActivityDefinition.AttributeName の列挙型定数

定数名	説明および記述形式
DESCRIPTION	「説明」を表す定数です。
	<code>public static final CIWActivityDefinition.AttributeName DESCRIPTION</code>
ID	「業務ステップ定義 ID」を表す定数です。
	<code>public static final CIWActivityDefinition.AttributeName ID</code>
NAME	「業務ステップ定義名」を表す定数です。
	<code>public static final CIWActivityDefinition.AttributeName NAME</code>
PROCESS_DEFINITION_ID	「ビジネスプロセス定義 ID」を表す定数です。
	<code>public static final CIWActivityDefinition.AttributeName PROCESS_DEFINITION_ID</code>

### メソッド

CIWActivityDefinition.AttributeName のメソッドを次の表に示します。

表 7-2 CIWActivityDefinition.AttributeName のメソッド

メソッド	説明
<code>toAttributeName</code>	業務ステップ定義属性の属性名（属性を格納するテーブルのカラム名）を取得します。
<code>valueOf</code>	指定した名前を持つこの型の列挙型定数を返します。

メソッド	説明
values	この列挙型の定数が含まれている配列を宣言されている順序で返します。

#### java.lang.Enum クラスから継承されたメソッド

clone, compareTo, equals, finalize, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

#### java.lang.Object クラスから継承されたメソッド

getClass, notify, notifyAll, wait, wait, wait

## toAttributeName

列挙型：CIWActivityDefinition.AttributeName

### 構文

```
public final java.lang.String  
    toAttributeName()
```

### 機能

業務ステップ定義属性の属性名（属性を格納するテーブルのカラム名）を取得します。

### 引数

ありません。

### 戻り値

業務ステップ定義属性の属性名を返します。

### 例外

ありません。

## valueOf

列挙型：CIWActivityDefinition.AttributeName

### 構文

```
public static CIWActivityDefinition.AttributeName valueOf(  
    java.lang.String name  
)
```



## 機能

指定した名前を持つこの型の列挙型定数を返します。

文字列は、この型の列挙型定数を宣言するのに使用した識別子と完全に一致している必要があります。余分な空白文字を含めることはできません。

## 引数

valueOf の引数を次の表に示します。

表 7-3 valueOf の引数 (CIWActivityDefinition.AttributeName の場合)

仮引数名	名称	in/out	説明
name	列挙型定数の名前	in	返される列挙型定数の名前を指定します。

## 戻り値

指定された名前を持つ列挙型定数を返します。

## 例外

valueOf で発生する例外を次の表に示します。

表 7-4 valueOf で発生する例外 (CIWActivityDefinition.AttributeName の場合)

発生する例外	説明
java.lang.IllegalArgumentException	指定された名前を持つ定数をこの列挙型が持っていない場合
java.lang.NullPointerException	引数が null の場合

## values

列挙型：CIWActivityDefinition.AttributeName

## 構文

```
public static CIWActivityDefinition.AttributeName[]  
    values()
```

## 機能

この列挙型の定数が含まれている配列を宣言されている順序で返します。

このメソッドは次のようにして定数を繰り返すために使用できます。

```
for(CIWActivityDefinition.AttributeName c : CIWActivityDefinition.AttributeName.values())  
    System.out.println(c);
```

## 引数

ありません。

## 戻り値

この列挙型の定数が含まれている配列を宣言されている順序で返します。

## 例外

ありません。

## 7.3 CIWActivityInstance.AttributeName (業務ステップ属性の属性名の列挙型)

### 概要

業務ステップ属性の属性名の列挙型です。

### 定義

```
public static enum CIWActivityInstance.AttributeName
    extends java.lang.Enum<CIWActivityInstance.AttributeName>
```

### すべての実装されたインタフェース

```
java.io.Serializable,
java.lang.Comparable<CIWActivityInstance.AttributeName>
```

### 含まれているインタフェース

```
CIWActivityInstance
```

### 定数

CIWActivityInstance.AttributeName の列挙型定数を次の表に示します。

表 7-5 CIWActivityInstance.AttributeName の列挙型定数

定数名	説明および記述形式
ACTIVITY_DEFINITION_ID	「業務ステップ定義の ID」を表す定数です。
	<pre>public static final CIWActivityInstance.AttributeName ACTIVITY_DEFINITION_ID</pre>
ACTIVITY_DEFINITION_NAME	「業務ステップ定義の名称」を表す定数です。
	<pre>public static final CIWActivityInstance.AttributeName ACTIVITY_DEFINITION_NAME</pre>
CLOSED_DATE	「業務ステップの終了日時」を表す定数です。
	<pre>public static final CIWActivityInstance.AttributeName CLOSED_DATE</pre>
DEADLINE	「業務ステップの処理期限の絶対日時」を表す定数です。 業務ステップ開始時に、業務ステップ定義の処理期限ルールに従って案件の開始日時 (START_DATE) を起算日として算出された値が設定されます。ただし、すでに API によって明示的に値を設定している場合は、処理期限ルールは適用されません。 未設定時は、null が設定されています。
	<pre>public static final CIWActivityInstance.AttributeName DEADLINE</pre>
ID	「業務ステップの ID」を表す定数です。
	<pre>public static final CIWActivityInstance.AttributeName ID</pre>
NAME	「業務ステップの名称 (業務ステップキー)」を表す定数です。

定数名	説明および記述形式
	<code>public static final CIWActivityInstance.AttributeName NAME</code>
PRIORITY	「業務ステップの優先度」を表す定数です。
	<code>public static final CIWActivityInstance.AttributeName PRIORITY</code>
PROCESS_DEFINITION_ID	「業務ステップが所属するビジネスプロセス定義の ID」を表す定数です。
	<code>public static final CIWActivityInstance.AttributeName PROCESS_DEFINITION_ID</code>
PROCESS_INSTANCE_ID	「業務ステップが所属する案件の ID」を表す定数です。
	<code>public static final CIWActivityInstance.AttributeName PROCESS_INSTANCE_ID</code>
PROCESS_INSTANCE_NAME	「案件名（案件キー）」を表す定数です。
	<code>public static final CIWActivityInstance.AttributeName PROCESS_INSTANCE_NAME</code>
START_DATE	「業務ステップの開始日時」を表す定数です。
	<code>public static final CIWActivityInstance.AttributeName START_DATE</code>
STATE	「業務ステップの状態」を表す定数です。
	<code>public static final CIWActivityInstance.AttributeName STATE</code>
TYPE	「業務ステップの種類」を表す定数です。
	<code>public static final CIWActivityInstance.AttributeName TYPE</code>

## メソッド

CIWActivityInstance.AttributeName のメソッドを次の表に示します。

表 7-6 CIWActivityInstance.AttributeName のメソッド

メソッド	説明
<code>toAttributeName</code>	業務ステップ属性の属性名（属性を格納するテーブルのカラム名）を取得します。
<code>valueOf</code>	指定した名前を持つこの型の列挙型定数を返します。
<code>values</code>	この列挙型の定数が含まれている配列を宣言されている順序で返します。

### java.lang.Enum クラスから継承されたメソッド

`clone`, `compareTo`, `equals`, `getDeclaringClass`, `hashCode`, `name`, `ordinal`, `toString`, `valueOf`

### java.lang.Object クラスから継承されたメソッド

`finalize`, `getClass`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

## toAttributeName

列挙型：CIWActivityInstance.AttributeName

### 構文

```
public final java.lang.String  
    toAttributeName()
```

### 機能

業務ステップ属性の属性名（属性を格納するテーブルのカラム名）を取得します。

### 引数

ありません。

### 戻り値

業務ステップ属性の属性名を返します。

### 例外

ありません。

## valueOf

列挙型：CIWActivityInstance.AttributeName

### 構文

```
public static CIWActivityInstance.AttributeName valueOf(  
    java.lang.String name  
)
```

### 機能

指定した名前を持つこの型の列挙型定数を返します。

文字列は、この型の列挙型定数を宣言するのに使用した識別子と完全に一致している必要があります。余分な空白文字を含めることはできません。

### 引数

valueOf の引数を次の表に示します。

表 7-7 valueOf の引数 (CIWActivityInstance.AttributeName の場合)

仮引数名	名称	in/out	説明
name	列挙型定数の名前	in	返される列挙型定数の名前を指定します。

## 戻り値

指定された名前を持つ列挙型定数を返します。

## 例外

valueOf で発生する例外を次の表に示します。

表 7-8 valueOf で発生する例外 (CIWActivityInstance.AttributeName の場合)

発生する例外	説明
java.lang.IllegalArgumentException	指定された名前を持つ定数をこの列挙型が持っていない場合

## values

列挙型 : CIWActivityInstance.AttributeName

## 構文

```
public static CIWActivityInstance.AttributeName[]
    values()
```

## 機能

この列挙型の定数が含まれている配列を宣言されている順序で返します。

このメソッドは次のようにして定数を繰り返すために使用できます。

```
for(CIWActivityInstance.AttributeName c : CIWActivityInstance.AttributeName.values())
    System.out.println(c);
```

## 引数

ありません。

## 戻り値

この列挙型の定数が含まれている配列を宣言されている順序で返します。

## 例外

ありません。

## 7.4 CIWActivityInstance.State (業務ステップの状態の列挙型)

### 概要

業務ステップの状態の列挙型です。

### 定義

```
public static enum CIWActivityInstance.State
    extends java.lang.Enum<CIWActivityInstance.State>
```

### すべての実装されたインタフェース

```
java.io.Serializable, java.lang.Comparable<CIWActivityInstance.State>
```

### 含まれているインタフェース

```
CIWActivityInstance
```

### 定数

CIWActivityInstance.State の列挙型定数を次の表に示します。

表 7-9 CIWActivityInstance.State の列挙型定数

定数名	説明および記述形式
ABORTED*	<p>「異常終了」を表す定数です。</p> <ul style="list-style-type: none"><li>短縮表記形式：aborted</li><li>完全表記形式：closed.aborted</li><li>コード値：v</li></ul> <pre>public static final CIWActivityInstance.State ABORTED</pre>
DISABLED	<p>「実行開始不可」を表す定数です。</p> <ul style="list-style-type: none"><li>短縮表記形式：disabled</li><li>完全表記形式：open.notRunning.suspended.disabled</li><li>コード値：l</li></ul> <pre>public static final CIWActivityInstance.State DISABLED</pre>
INITIAL	<p>「初期」を表す定数です。</p> <ul style="list-style-type: none"><li>短縮表記形式：initial</li><li>完全表記形式：open.notRunning.notStarted.initial</li><li>コード値：i</li></ul> <pre>public static final CIWActivityInstance.State INITIAL</pre>
INTERMITTED	<p>「実行停止」を表す定数です。</p> <ul style="list-style-type: none"><li>短縮表記形式：intermitted</li><li>完全表記形式：open.notRunning.suspended.intermitted</li><li>コード値：m</li></ul> <pre>public static final CIWActivityInstance.State INTERMITTED</pre>



定数名	説明および記述形式
NOT_EXECUTED	<p>「実行省略」を表す定数です。</p> <ul style="list-style-type: none"> <li>短縮表記形式：notExecuted</li> <li>完全表記形式：closed.completed.notExecuted</li> <li>コード値：p</li> </ul>
	<code>public static final CIWActivityInstance.State NOT_EXECUTED</code>
READY	<p>「実行開始可能」を表す定数です。</p> <ul style="list-style-type: none"> <li>短縮表記形式：ready</li> <li>完全表記形式：open.notRunning.notStarted.ready</li> <li>コード値：j</li> </ul>
	<code>public static final CIWActivityInstance.State READY</code>
READY_FOR_TRANSITION	<p>「遷移可」を表す定数です。</p> <ul style="list-style-type: none"> <li>短縮表記形式：readyForTransition</li> <li>完全表記形式：closed.completed.executed.readyForTransition</li> <li>コード値：s</li> </ul>
	<code>public static final CIWActivityInstance.State READY_FOR_TRANSITION</code>
RUNNING	<p>「実行中」を表す定数です。</p> <ul style="list-style-type: none"> <li>短縮表記形式：running</li> <li>完全表記形式：open.running</li> <li>コード値：d</li> </ul>
	<code>public static final CIWActivityInstance.State RUNNING</code>
TERMINATED	<p>「強制終了」を表す定数です。</p> <ul style="list-style-type: none"> <li>短縮表記形式：terminated</li> <li>完全表記形式：closed.terminated</li> <li>コード値：u</li> </ul>
	<code>public static final CIWActivityInstance.State TERMINATED</code>
TRANSITION_COMPLETED	<p>「遷移済」を表す定数です。</p> <ul style="list-style-type: none"> <li>短縮表記形式：transitionCompleted</li> <li>完全表記形式：closed.completed.executed.transitionCompleted</li> <li>コード値：t</li> </ul>
	<code>public static final CIWActivityInstance.State TRANSITION_COMPLETED</code>
UNDEFINED	<p>「未定義」を表す定数です。</p> <ul style="list-style-type: none"> <li>短縮表記形式：undefined</li> <li>完全表記形式：undefined</li> <li>コード値：z</li> </ul>
	<code>public static final CIWActivityInstance.State UNDEFINED</code>

注※

CSCIW ではサポートしていません。

## メソッド

CIWActivityInstance.State のメソッドを次の表に示します。

表 7-10 CIWActivityInstance.State のメソッド

メソッド	説明
fromStateCode	指定した State のコード値に対応した業務ステップの状態を定数に持つ、State 列挙型を返します。
isClosed	業務ステップが「終了」状態かどうかを判定します。
isCompleted	業務ステップが「完了」状態かどうかを判定します。
isExecuted	業務ステップが「実行済」状態かどうかを判定します。
isNotRunning	業務ステップが「停止中」状態かどうかを判定します。
isNotStarted	業務ステップが「未実行」状態かどうかを判定します。
isOpen	業務ステップが「未終了」状態かどうかを判定します。
isSuspended	業務ステップが「中断」状態かどうかを判定します。
toShortState	業務ステップの状態を示す State の短縮表記形式を取得します。
toState	業務ステップの状態を示す State の完全表記形式を取得します。
toStateCode	業務ステップの状態を示す State のコード値を取得します。
toString	業務ステップの状態を示す文字列表現を取得します。
valueOf	指定した名前を持つこの型の列挙型定数を返します。
values	この列挙型の定数が含まれている配列を宣言されている順序で返します。

### java.lang.Enum クラスから継承されたメソッド

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, valueOf

### java.lang.Object クラスから継承されたメソッド

finalize, getClass, notify, notifyAll, wait, wait, wait

## fromStateCode

列挙型：CIWActivityInstance.State

### 構文

```
public static CIWActivityInstance.State fromStateCode(  
    java.lang.String aCode  
)  
    throws CIWFatalException
```

## 機能

指定した State のコード値に対応した業務ステップの状態を定数に持つ、State 列挙型を返します。

## 引数

fromStateCode の引数を次の表に示します。

表 7-11 fromStateCode の引数 (CIWActivityInstance.State の場合)

仮引数名	名称	in/out	説明
aCode	State 列挙型の StateCode の値	in	取得したい State 列挙型の StateCode の値を指定します。

## 戻り値

State のコード値に対応する状態を定数に持つ State 列挙型を返します。

## 例外

fromStateCode で発生する例外を次の表に示します。

表 7-12 fromStateCode で発生する例外 (CIWActivityInstance.State の場合)

発生する例外	説明
<a href="#">CIWFatalException</a>	処理を続行できない障害が発生した場合

## 注意事項

- State のコード値に「z」または存在しないコードを指定した場合は、例外 (CIWFatalException) が発生します。
- State のコード値については、「[付録 B 参照系 API で指定できる属性](#)」を参照してください。

## isClosed

列挙型：CIWActivityInstance.State

## 構文

```
public boolean isClosed()
```

## 機能

業務ステップが「終了」状態かどうかを判定します。

## 引数

ありません。

## 戻り値

- 業務ステップが「終了」状態の場合は true を返します。
- 業務ステップが「終了」状態以外の場合は false を返します。

## 例外

ありません。

## isCompleted

列挙型：CIWActivityInstance.State

## 構文

```
public boolean isCompleted()
```

## 機能

業務ステップが「完了」状態かどうかを判定します。

## 引数

ありません。

## 戻り値

- 業務ステップが「完了」状態の場合は true を返します。
- 業務ステップが「完了」状態以外の場合は false を返します。

## 例外

ありません。

## isExecuted

列挙型：CIWActivityInstance.State

## 構文

```
public boolean isExecuted()
```

## 機能

業務ステップが「実行済」状態かどうかを判定します。

## 引数

ありません。

## 戻り値

- 業務ステップが「実行済」状態の場合は true を返します。
- 業務ステップが「実行済」状態以外の場合は false を返します。

## 例外

ありません。

## isNotRunning

列挙型：CIWActivityInstance.State

## 構文

```
public boolean isNotRunning()
```

## 機能

業務ステップが「停止中」状態かどうかを判定します。

## 引数

ありません。

## 戻り値

- 業務ステップが「停止中」状態の場合は true を返します。
- 業務ステップが「停止中」状態以外の場合は false を返します。

## 例外

ありません。

## isNotStarted

列挙型：CIWActivityInstance.State

### 構文

```
public boolean isNotStarted()
```

### 機能

業務ステップが「未実行」状態かどうかを判定します。

### 引数

ありません。

### 戻り値

- 業務ステップが「未実行」状態の場合は true を返します。
- 業務ステップが「未実行」状態以外の場合は false を返します。

### 例外

ありません。

## isOpen

列挙型：CIWActivityInstance.State

### 構文

```
public boolean isOpen()
```

### 機能

業務ステップが「未終了」状態かどうかを判定します。

### 引数

ありません。

### 戻り値

- 業務ステップが「未終了」状態の場合は true を返します。
- 業務ステップが「未終了」状態以外の場合は false を返します。

## 例外

ありません。

## isSuspended

列挙型：CIWActivityInstance.State

## 構文

```
public boolean isSuspended()
```

## 機能

業務ステップが「中断」状態かどうかを判定します。

## 引数

ありません。

## 戻り値

- 業務ステップが「中断」状態の場合は true を返します。
- 業務ステップが「中断」状態以外の場合は false を返します。

## 例外

ありません。

## toShortState

列挙型：CIWActivityInstance.State

## 構文

```
public final java.lang.String toShortState()
```

## 機能

業務ステップの状態を示す State の短縮表記形式を取得します。

## 引数

ありません。

## 戻り値

State の短縮表記形式を返します。

## 例外

ありません。

## toState

列挙型：CIWActivityInstance.State

## 構文

```
public final java.lang.String toState()
```

## 機能

業務ステップの状態を示す State の完全表記形式を取得します。

## 引数

ありません。

## 戻り値

State の完全表記形式を返します。

## 例外

ありません。

## 注意事項

State の完全表記形式については、「[付録 B 参照系 API で指定できる属性](#)」を参照してください。

## toStateCode

列挙型：CIWActivityInstance.State

## 構文

```
public final java.lang.String toStateCode()
```



## 機能

業務ステップの状態を示す State のコード値を取得します。

## 引数

ありません。

## 戻り値

State のコード値を返します。

## 例外

ありません。

## 注意事項

State のコード値については、「[付録 B 参照系 API で指定できる属性](#)」を参照してください。

## toString

列挙型：CIWActivityInstance.State

## 構文

```
public final java.lang.String toString()
```

## 機能

業務ステップの状態を示す文字列表現を取得します。

表示形式は、「状態の定数（状態のコード値）」となります。

## 引数

ありません。

## オーバーライド

java.lang.Enum<CIWActivityInstance.State>クラス内の toString

## 戻り値

業務ステップの状態の文字列表現を返します。

## 例外

ありません。

## 注意事項

業務ステップの状態の文字列表現については、「付録 B 参照系 API で指定できる属性」を参照してください。

## valueOf

列挙型：CIWActivityInstance.State

## 構文

```
public static CIWActivityInstance.State valueOf(  
    java.lang.String name  
)
```

## 機能

指定した名前を持つこの型の列挙型定数を返します。

文字列は、この型の列挙型定数を宣言するのに使用した識別子と完全に一致している必要があります。余分な空白文字を含めることはできません。

## 引数

valueOf の引数を次の表に示します。

表 7-13 valueOf の引数 (CIWActivityInstance.State の場合)

仮引数名	名称	in/out	説明
name	列挙型定数の名前	in	返される列挙型定数の名前を指定します。

## 戻り値

指定された名前を持つ列挙型定数を返します。

## 例外

valueOf で発生する例外を次の表に示します。

表 7-14 valueOf で発生する例外 (CIWActivityInstance.State の場合)

発生する例外	説明
java.lang.IllegalArgumentException	指定された名前を持つ定数をこの列挙型が持っていない場合

## values

列挙型 : CIWActivityInstance.State

### 構文

```
public static CIWActivityInstance.State[]  
    values()
```

### 機能

この列挙型の定数が含まれている配列を宣言されている順序で返します。

このメソッドは次のようにして定数を繰り返すために使用できます。

```
for(CIWActivityInstance.State c :  
    CIWActivityInstance.State.values())  
    System.out.println(c);
```

### 引数

ありません。

### 戻り値

この列挙型の定数が含まれている配列を宣言されている順序で返します。

### 例外

ありません。

## 7.5 CIWActivityInstance.Type (業務ステップの種類)の列挙型

### 概要

業務ステップの種類)の列挙型です。

### 定義

```
public static enum CIWActivityInstance.Type
    extends java.lang.Enum<CIWActivityInstance.Type>
```

### すべての実装されたインタフェース

```
java.io.Serializable,
java.lang.Comparable<CIWActivityInstance.Type>
```

### 含まれているインタフェース

```
CIWActivityInstance
```

### 定数

CIWActivityInstance.Type の列挙型定数を次の表に示します。

表 7-15 CIWActivityInstance.Type の列挙型定数

定数名	説明および記述形式
ERROR_NOTICE*	「エラー回復業務ステップ」を表す定数です。 <ul style="list-style-type: none"><li>完全表記形式：errorNotice</li><li>コード値：9</li></ul>
	<pre>public static final CIWActivityInstance.Type ERROR_NOTICE</pre>
NORMAL	「一般業務ステップ」を表す定数です。 <ul style="list-style-type: none"><li>完全表記形式：normal</li><li>コード値：0</li></ul>
	<pre>public static final CIWActivityInstance.Type NORMAL</pre>
UNDEFINED	「未定義」を表す定数です。 <ul style="list-style-type: none"><li>完全表記形式：undefined</li><li>コード値：z</li></ul>
	<pre>public static final CIWActivityInstance.Type UNDEFINED</pre>

注※

CSCIW ではサポートしていません。

### メソッド

CIWActivityInstance.Type のメソッドを次の表に示します。

表 7-16 CIWActivityInstance.Type のメソッド

メソッド	説明
fromTypeCode	指定した Type のコード値に対応した業務ステップの種類を定数を持つ、Type 列挙型を返します。
toString	業務ステップの種類 of 文字列表現を取得します。
toType	業務ステップの種類を示す Type の完全表記形式を取得します。
toTypeCode	業務ステップの種類を示す Type のコード値を取得します。
valueOf	指定した名前を持つこの型の列挙型定数を返します。
values	この列挙型の定数が含まれている配列を宣言されている順序で返します。

java.lang.Enum クラスから継承されたメソッド

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, valueOf

java.lang.Object クラスから継承されたメソッド

finalize, getClass, notify, notifyAll, wait, wait, wait

## fromTypeCode

列挙型 : CIWActivityInstance.Type

### 構文

```
public static CIWActivityInstance.Type fromTypeCode(
    java.lang.String aCode
)
    throws CIWFatalException
```

### 機能

指定した Type のコード値に対応した業務ステップの種類を定数を持つ、Type 列挙型を返します。

### 引数

fromTypeCode の引数を次の表に示します。

表 7-17 fromTypeCode の引数 (CIWActivityInstance.Type の場合)

仮引数名	名称	in/out	説明
aCode	Type のコード値	in	取得したい Type のコード値を指定します。

### 戻り値

Type のコード値に対応する種類を定数を持つ Type 列挙型を返します。

## 例外

fromTypeCode で発生する例外を次の表に示します。

表 7-18 fromTypeCode で発生する例外 (CIWActivityInstance.Type の場合)

発生する例外	説明
CIWFatalException	処理を続行できない障害が発生した場合

## 注意事項

- Type のコード値に「z」または存在しないコードを指定した場合は、例外 (CIWFatalException) が発生します。
- Type のコード値については、「[付録 B 参照系 API で指定できる属性](#)」を参照してください。

## toString

列挙型：CIWActivityInstance.Type

## 構文

```
public final java.lang.String toString()
```

## 機能

業務ステップの種類の種類文字列表現を取得します。

表示形式は、「種類の数 (種類の種類)」になります。

## 引数

ありません。

## オーバーライド

java.lang.Enum<CIWActivityInstance.Type>クラス内の toString

## 戻り値

業務ステップの種類の種類文字列表現を返します。

## 例外

ありません。

## 注意事項

業務ステップの種類の文字列表現については、「[付録 B 参照系 API で指定できる属性](#)」を参照してください。

## toType

列挙型：CIWActivityInstance.Type

### 構文

```
public final java.lang.String toType()
```

### 機能

業務ステップの種類を示す Type の完全表記形式を取得します。

### 引数

ありません。

### 戻り値

Type の完全表記形式を返します。

### 例外

ありません。

## 注意事項

Type の完全表記形式については、「[付録 B 参照系 API で指定できる属性](#)」を参照してください。

## toTypeCode

列挙型：CIWActivityInstance.Type

### 構文

```
public final java.lang.String toTypeCode()
```

### 機能

業務ステップの種類を示す Type のコード値を取得します。

## 引数

ありません。

## 戻り値

Type のコード値を返します。

## 例外

ありません。

## 注意事項

Type のコード値については、「[付録 B 参照系 API で指定できる属性](#)」を参照してください。

## valueOf

列挙型：CIWActivityInstance.Type

## 構文

```
public static CIWActivityInstance.Type valueOf(  
    java.lang.String name  
)
```

## 機能

指定した名前を持つこの型の列挙型定数を返します。

文字列は、この型の列挙型定数を宣言するのに使用した識別子と完全に一致している必要があります。余分な空白文字を含めることはできません。

## 引数

valueOf の引数を次の表に示します。

表 7-19 valueOf の引数 (CIWActivityInstance.Type の場合)

仮引数名	名称	in/out	説明
name	列挙型定数の名前	in	返される列挙型定数の名前を指定します。

## 戻り値

指定された名前を持つ列挙型定数を返します。



## 例外

valueOf で発生する例外を次の表に示します。

表 7-20 valueOf で発生する例外 (CIWActivityInstance.Type の場合)

発生する例外	説明
java.lang.IllegalArgumentException	指定された名前を持つ定数をこの列挙型が持っていない場合

## values

列挙型 : CIWActivityInstance.Type

### 構文

```
public static CIWActivityInstance.Type[]  
    values()
```

### 機能

この列挙型の定数が含まれている配列を宣言されている順序で返します。

このメソッドは次のようにして定数を繰り返すために使用できます。

```
for(CIWActivityInstance.AttributeName c :  
    CIWActivityInstance.Type.values())  
    System.out.println(c);
```

### 引数

ありません。

### 戻り値

この列挙型の定数が含まれている配列を宣言されている順序で返します。

### 例外

ありません。

## 7.6 CIWProcessDefinition.AttributeName (ビジネスプロセス定義属性の属性名の列挙型)

### 概要

ビジネスプロセス定義属性の属性名の列挙型です。

### 定義

```
public static enum CIWProcessDefinition.AttributeName
    extends java.lang.Enum<CIWProcessDefinition.AttributeName>
```

### すべての実装されたインタフェース

```
java.io.Serializable,
java.lang.Comparable<CIWProcessDefinition.AttributeName>
```

### 含まれているインタフェース

```
CIWProcessDefinition
```

### 定数

CIWProcessDefinition.AttributeName の列挙型定数を次の表に示します。

表 7-21 CIWProcessDefinition.AttributeName の列挙型定数

定数名	説明および記述形式
AUTHOR	「ビジネスプロセス定義の作成者（上書き時は更新者）」を表す定数です。 <pre>public static final CIWProcessDefinition.AttributeName AUTHOR</pre>
CREATION_DATE	「ビジネスプロセス定義の作成日（上書き時は更新日）」を表す定数です。 <pre>public static final CIWProcessDefinition.AttributeName CREATION_DATE</pre>
DESCRIPTION	「ビジネスプロセス定義の説明」を表す定数です。 <pre>public static final CIWProcessDefinition.AttributeName DESCRIPTION</pre>
ID	「ビジネスプロセス定義 ID」を表す定数です。 <pre>public static final CIWProcessDefinition.AttributeName ID</pre>
NAME	「ビジネスプロセス定義名称」を表す定数です。 <pre>public static final CIWProcessDefinition.AttributeName NAME</pre>
RESPONSIBLE	「ビジネスプロセス定義の管理者 ID」を表す定数です。 <pre>public static final CIWProcessDefinition.AttributeName RESPONSIBLE</pre>
STATE	「ビジネスプロセス定義の状態」を表す定数です。 <pre>public static final CIWProcessDefinition.AttributeName STATE</pre>
VALID_FROM_DATE	「ビジネスプロセス定義が有効となる日（投入可能開始日）」を表す定数です。

定数名	説明および記述形式
	特に指定していない場合は、無限遠の過去を表す日付情報が設定されます。無限遠の過去を表す日付情報については、CIWUtil.InfinityDate.ORIGIN を参照してください。
	<code>public static final CIWProcessDefinition.AttributeName VALID_FROM_DATE</code>
VALID_TO_DATE	「ビジネスプロセス定義が無効となる日（投入可能終了日）」を表す定数です。特に指定していない場合は、無限遠の未来を表す日付情報が設定されます。無限遠の未来を表す日付情報については、CIWUtil.InfinityDate.BEYOND を参照してください。
	<code>public static final CIWProcessDefinition.AttributeName VALID_TO_DATE</code>
VERSION	「ビジネスプロセス定義のバージョン」を表す定数です。
	<code>public static final CIWProcessDefinition.AttributeName VERSION</code>

## メソッド

CIWProcessDefinition.AttributeName のメソッドを次の表に示します。

表 7-22 CIWProcessDefinition.AttributeName のメソッド

メソッド	説明
<code>toAttributeName</code>	ビジネスプロセス定義属性の属性名（属性を格納するテーブルのカラム名）を取得します。
<code>valueOf</code>	指定した名前を持つこの型の列挙型定数を返します。
<code>values</code>	この列挙型の定数が含まれている配列を宣言されている順序で返します。

### java.lang.Enum クラスから継承されたメソッド

`clone`, `compareTo`, `equals`, `getDeclaringClass`, `hashCode`, `name`, `ordinal`, `toString`, `valueOf`

### java.lang.Object クラスから継承されたメソッド

`finalize`, `getClass`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

## toAttributeName

列挙型：CIWProcessDefinition.AttributeName

## 構文

```
public final java.lang.String toAttributeName()
```

## 機能

ビジネスプロセス定義属性の属性名（属性を格納するテーブルのカラム名）を取得します。

## 引数

ありません。

## 戻り値

ビジネスプロセス定義属性の属性名を返します。

## 例外

ありません。

## valueOf

列挙型：CIWProcessDefinition.AttributeName

## 構文

```
public static CIWProcessDefinition.AttributeName valueOf(  
    java.lang.String name  
)
```

## 機能

指定した名前を持つこの型の列挙型定数を返します。

文字列は、この型の列挙型定数を宣言するのに使用した識別子と完全に一致している必要があります。余分な空白文字を含めることはできません。

## 引数

valueOf の引数を次の表に示します。

表 7-23 valueOf の引数 (CIWProcessDefinition.AttributeName の場合)

仮引数名	名称	in/out	説明
name	列挙型定数の名前	in	返される列挙型定数の名前を指定します。

## 戻り値

指定された名前を持つ列挙型定数を返します。

## 例外

valueOf で発生する例外を次の表に示します。

表 7-24 valueOf で発生する例外 (CIWProcessDefinition.AttributeName の場合)

発生する例外	説明
java.lang.IllegalArgumentException	指定された名前を持つ定数をこの列挙型が持っていない場合

## values

列挙型 : CIWProcessDefinition.AttributeName

## 構文

```
public static CIWProcessDefinition.AttributeName[]  
    values()
```

## 機能

この列挙型の定数が含まれている配列を宣言されている順序で返します。

このメソッドは次のようにして定数を繰り返すために使用できます。

```
for(CIWProcessDefinition.AttributeName c :  
    CIWProcessDefinition.AttributeName.values())  
    System.out.println(c);
```

## 引数

ありません。

## 戻り値

この列挙型の定数が含まれている配列を宣言されている順序で返します。

## 例外

ありません。

## 7.7 CIWProcessDefinition.State (ビジネスプロセス定義の状態の列挙型)

### 概要

ビジネスプロセス定義の状態の列挙型です。

### 定義

```
public static enum CIWProcessDefinition.State
    extends java.lang.Enum<CIWProcessDefinition.State>
```

### すべての実装されたインタフェース

```
java.io.Serializable,
java.lang.Comparable<CIWProcessDefinition.State>
```

### 含まれているインタフェース

```
CIWProcessDefinition
```

### 定数

CIWProcessDefinition.State の列挙型定数を次の表に示します。

表 7-25 CIWProcessDefinition.State の列挙型定数

定数名	説明および記述形式
ACTIVE	「活性」を表す定数です。 <ul style="list-style-type: none"><li>短縮表記形式：active</li><li>完全表記形式：active</li><li>コード値：b</li></ul>
	<pre>public static final CIWProcessDefinition.State ACTIVE</pre>
INACTIVE	「非活性」を表す定数です。 <ul style="list-style-type: none"><li>短縮表記形式：inactive</li><li>完全表記形式：inactive</li><li>コード値：a</li></ul>
	<pre>public static final CIWProcessDefinition.State INACTIVE</pre>
UNDEFINED	「未定義」を表す定数です。 <ul style="list-style-type: none"><li>短縮表記形式：undefined</li><li>完全表記形式：undefined</li><li>コード値：z</li></ul>
	<pre>public static final CIWProcessDefinition.State UNDEFINED</pre>

### メソッド

CIWProcessDefinition.State のメソッドを次の表に示します。

表 7-26 CIWProcessDefinition.State のメソッド

メソッド	説明
fromStateCode	指定した State のコード値に対応したビジネスプロセス定義の状態を定数に持つ、State 列挙型を返します。
toShortState	ビジネスプロセス定義の状態を示す State の短縮表記形式を取得します。
toState	ビジネスプロセス定義の状態を示す State の完全表記形式を取得します。
toStateCode	ビジネスプロセス定義の状態を示す State のコード値を取得します。
toString	ビジネスプロセス定義の状態の文字列表現を取得します。
valueOf	指定した名前を持つこの型の列挙型定数を返します。
values	この列挙型の定数が含まれている配列を宣言されている順序で返します。

java.lang.Enum クラスから継承されたメソッド

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, valueOf

java.lang.Object クラスから継承されたメソッド

finalize, getClass, notify, notifyAll, wait, wait, wait

## fromStateCode

列挙型：CIWProcessDefinition.State

### 構文

```
public static CIWProcessDefinition.State fromStateCode(
    java.lang.String aCode
)
    throws CIWFatalException
```

### 機能

指定した State のコード値に対応したビジネスプロセス定義の状態を定数に持つ、State 列挙型を返します。

### 引数

fromStateCode の引数を次の表に示します。

表 7-27 fromStateCode の引数 (CIWProcessDefinition.State の場合)

仮引数名	名称	in/out	説明
aCode	State のコード値	in	取得したい State のコード値を指定します。

## 戻り値

State のコード値に対応する状態を定数に持つ State 列挙型を返します。

## 例外

fromStateCode で発生する例外を次の表に示します。

表 7-28 fromStateCode で発生する例外 (CIWProcessDefinition.State の場合)

発生する例外	説明
<a href="#">CIWFatalException</a>	処理を続行できない障害が発生した場合

## 注意事項

- State のコード値に「z」または存在しないコードを指定した場合は、例外 (CIWFatalException) が発生します。
- State のコード値については、「[付録 B 参照系 API で指定できる属性](#)」を参照してください。

## toShortState

列挙型：CIWProcessDefinition.State

## 構文

```
public final java.lang.String toShortState()
```

## 機能

ビジネスプロセス定義の状態を示す State の短縮表記形式を取得します。

## 引数

ありません。

## 戻り値

State の短縮表記形式を返します。

## 例外

ありません。

## 注意事項

State の短縮表記形式については、「[付録 B 参照系 API で指定できる属性](#)」を参照してください。



## toState

列挙型：CIWProcessDefinition.State

### 構文

```
public final java.lang.String toState()
```

### 機能

ビジネスプロセス定義の状態を示す State の完全表記形式を取得します。

### 引数

ありません。

### 戻り値

State の完全表記形式を返します。

### 例外

ありません。

### 注意事項

State の完全表記形式については、「[付録 B 参照系 API で指定できる属性](#)」を参照してください。

## toStateCode

列挙型：CIWProcessDefinition.State

### 構文

```
public final java.lang.String toStateCode()
```

### 機能

ビジネスプロセス定義の状態を示す State のコード値を取得します。

### 引数

ありません。

### 戻り値

State のコード値を返します。

## 例外

ありません。

## 注意事項

State のコード値については、「付録 B 参照系 API で指定できる属性」を参照してください。

## toString

列挙型：CIWProcessDefinition.State

## 構文

```
public final java.lang.String toString()
```

## 機能

ビジネスプロセス定義の状態の文字列表現を取得します。

表示形式は、「状態の定数（状態のコード値）」になります。

## 引数

ありません。

## オーバーライド

java.lang.Enum<CIWProcessDefinition.State>クラス内の toString

## 戻り値

ビジネスプロセス定義の状態の文字列表現を返します。

## 例外

ありません。

## 注意事項

ビジネスプロセス定義の状態の文字列表現については、「付録 B 参照系 API で指定できる属性」を参照してください。

## valueOf

列挙型：CIWProcessDefinition.State

## 構文

```
public static CIWProcessDefinition.State valueOf(  
    java.lang.String name  
)
```

## 機能

指定した名前を持つこの型の列挙型定数を返します。

文字列は、この型の列挙型定数を宣言するのに使用した識別子と完全に一致している必要があります。余分な空白文字を含めることはできません。

## 引数

valueOf の引数を次の表に示します。

表 7-29 valueOf の引数 (CIWProcessDefinition.State の場合)

仮引数名	名称	in/out	説明
name	列挙型定数の名前	in	返される列挙型定数の名前を指定します。

## 戻り値

指定された名前を持つ列挙型定数を返します。

## 例外

valueOf で発生する例外を次の表に示します。

表 7-30 valueOf で発生する例外 (CIWProcessDefinition.State の場合)

発生する例外	説明
java.lang.IllegalArgumentException	指定された名前を持つ定数をこの列挙型が持っていない場合

## values

列挙型 : CIWProcessDefinition.State

## 構文

```
public static CIWProcessDefinition.State[]  
    values()
```

## 機能

この列挙型の定数が含まれている配列を宣言されている順序で返します。

このメソッドは次のようにして定数を繰り返すために使用できます。

```
for(CIWProcessDefinition.State c :  
    CIWProcessDefinition.State.values())  
    System.out.println(c);
```

## 引数

ありません。

## 戻り値

この列挙型の定数が含まれている配列を宣言されている順序で返します。

## 例外

ありません。

## 7.8 CIWProcessInstance.AttributeName (案件属性の属性名の列挙型)

### 概要

案件属性の属性名の列挙型です。

### 定義

```
public static enum CIWProcessInstance.AttributeName
    extends java.lang.Enum<CIWProcessInstance.AttributeName>
```

### すべての実装されたインタフェース

```
java.io.Serializable,
java.lang.Comparable<CIWProcessInstance.AttributeName>
```

### 含まれているインタフェース

```
CIWProcessInstance
```

### 定数

CIWProcessInstance.AttributeName の列挙型定数を次の表に示します。

表 7-31 CIWProcessInstance.AttributeName の列挙型定数

定数名	説明および記述形式
CLOSED_DATE	「案件の終了日時」を表す定数です。
	<code>public static final CIWProcessInstance.AttributeName CLOSED_DATE</code>
CREATOR	「案件の投入者」を表す定数です。
	<code>public static final CIWProcessInstance.AttributeName CREATOR</code>
DEADLINE	「案件の処理期限の絶対日時」を表す定数です。 案件開始時に、ビジネスプロセス定義の処理期限ルールに従い、案件の開始日時 (START_DATE) を起算日として算出された値が設定されます。ただし、すでに API によって明示的に値を設定している場合は、処理期限ルールは適用されません。値を設定していない場合は、null が設定されています。
	<code>public static final CIWProcessInstance.AttributeName DEADLINE</code>
ID	「案件の ID」を表す定数です。
	<code>public static final CIWProcessInstance.AttributeName ID</code>
MOVED_DATE	「案件の乗せ替え日時」を表す定数です。
	<code>public static final CIWProcessInstance.AttributeName MOVED_DATE</code>
NAME	「案件の名称 (案件キー)」を表す定数です。
	<code>public static final CIWProcessInstance.AttributeName NAME</code>
PRIORITY	「案件の優先度」を表す定数です。
	<code>public static final CIWProcessInstance.AttributeName PRIORITY</code>

定数名	説明および記述形式
PROCESS_DEFINITION_ID	「案件が所属するビジネスプロセス定義の ID」を表す定数です。  public static final CIWProcessInstance.AttributeName PROCESS_DEFINITION_ID
PROCESS_DEFINITION_NAME	「案件が所属するビジネスプロセス定義の名称」を表す定数です。  public static final CIWProcessInstance.AttributeName PROCESS_DEFINITION_NAME
START_DATE	「案件の開始日時」を表す定数です。  public static final CIWProcessInstance.AttributeName START_DATE
STATE	「案件の状態」を表す定数です。  public static final CIWProcessInstance.AttributeName STATE

## メソッド

CIWProcessInstance.AttributeName のメソッドを次の表に示します。

表 7-32 CIWProcessInstance.AttributeName のメソッド

メソッド	説明
<a href="#">toAttributeName</a>	案件属性の属性名（属性を格納するテーブルのカラム名）を取得します。
<a href="#">valueOf</a>	指定した名前を持つこの型の列挙型定数を返します。
<a href="#">values</a>	この列挙型の定数が含まれている配列を宣言されている順序で返します。

### java.lang.Enum クラスから継承されたメソッド

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

### java.lang.Object クラスから継承されたメソッド

finalize, getClass, notify, notifyAll, wait, wait, wait

## toAttributeName

列挙型：CIWProcessInstance.AttributeName

## 構文

```
public final java.lang.String toAttributeName()
```

## 機能

案件属性の属性名（属性を格納するテーブルのカラム名）を取得します。

## 引数

ありません。

## 戻り値

案件属性の属性名を返します。

## 例外

ありません。

## valueOf

列挙型：CIWProcessInstance.AttributeName

## 構文

```
public static CIWProcessInstance.AttributeName valueOf(  
    java.lang.String name  
)
```

## 機能

指定した名前を持つこの型の列挙型定数を返します。

文字列は、この型の列挙型定数を宣言するのに使用した識別子と完全に一致している必要があります。余分な空白文字を含めることはできません。

## 引数

valueOf の引数を次の表に示します。

表 7-33 valueOf の引数 (CIWProcessInstance.AttributeName の場合)

仮引数名	名称	in/out	説明
name	列挙型定数の名前	in	返される列挙型定数の名前を指定します。

## 戻り値

指定された名前を持つ列挙型定数を返します。

## 例外

valueOf で発生する例外を次の表に示します。

表 7-34 valueOf で発生する例外 (CIWProcessInstance.AttributeName の場合)

発生する例外	説明
java.lang.IllegalArgumentException	指定された名前を持つ定数をこの列挙型が持っていない場合

## values

列挙型 : CIWProcessInstance.AttributeName

## 構文

```
public static CIWProcessInstance.AttributeName[]  
    values()
```

## 機能

この列挙型の定数が含まれている配列を宣言されている順序で返します。

このメソッドは次のようにして定数を繰り返すために使用できます。

```
for(CIWProcessDefinition.AttributeName c :  
    CIWProcessDefinition.AttributeName.values())  
    System.out.println(c);
```

## 引数

ありません。

## 戻り値

この列挙型の定数が含まれている配列を宣言されている順序で返します。

## 例外

ありません。



## 7.9 CIWProcessInstance.State (案件の状態の列挙型)

### 概要

案件の状態の列挙型です。

### 定義

```
public static enum CIWProcessInstance.State
    extends java.lang.Enum<CIWProcessInstance.State>
```

### すべての実装されたインタフェース

```
java.io.Serializable,
java.lang.Comparable<CIWProcessInstance.State>
```

### 含まれているインタフェース

```
CIWProcessInstance
```

### 定数

CIWProcessInstance.State の列挙型定数を次の表に示します。

表 7-35 CIWProcessInstance.State の列挙型定数

定数名	説明および記述形式
ABORTED*	<p>「異常終了」を表す定数です。</p> <ul style="list-style-type: none"><li>短縮表記形式：aborted</li><li>完全表記形式：closed.aborted</li><li>コード値：v</li></ul> <pre>public static final CIWProcessInstance.State ABORTED</pre>
COMPLETED	<p>「完了」を表す定数です。</p> <ul style="list-style-type: none"><li>短縮表記形式：completed</li><li>完全表記形式：closed.completed</li><li>コード値：o</li></ul> <pre>public static final CIWProcessInstance.State COMPLETED</pre>
INTERMITTED	<p>「実行停止」を表す定数です。</p> <ul style="list-style-type: none"><li>短縮表記形式：intermitted</li><li>完全表記形式：open.notRunning.suspended.intermitted</li><li>コード値：m</li></ul> <pre>public static final CIWProcessInstance.State INTERMITTED</pre>
NOT_STARTED	<p>「未実行」を表す定数です。</p> <ul style="list-style-type: none"><li>短縮表記形式：notStarted</li><li>完全表記形式：open.notRunning.notStarted</li><li>コード値：h</li></ul> <pre>public static final CIWProcessInstance.State NOT_STARTED</pre>

定数名	説明および記述形式
RUNNING	<p>「実行中」を表す定数です。</p> <ul style="list-style-type: none"> <li>短縮表記形式：running</li> <li>完全表記形式：open.running</li> <li>コード値：d</li> </ul> <pre>public static final CIWProcessInstance.State RUNNING</pre>
TERMINATED	<p>「強制終了」を表す定数です。</p> <ul style="list-style-type: none"> <li>短縮表記形式：terminated</li> <li>完全表記形式：closed.terminated</li> <li>コード値：u</li> </ul> <pre>public static final CIWProcessInstance.State TERMINATED</pre>
UNDEFINED	<p>「未定義」を表す定数です。</p> <ul style="list-style-type: none"> <li>短縮表記形式：undefined</li> <li>完全表記形式：undefined</li> <li>コード値：z</li> </ul> <pre>public static final CIWProcessInstance.State UNDEFINED</pre>

注※

CSCIW ではサポートしていません。

## メソッド

CIWProcessInstance.State のメソッドを次の表に示します。

表 7-36 CIWProcessInstance.State のメソッド

メソッド	説明
<code>fromStateCode</code>	指定した State のコード値に対応した案件の状態を定数に持つ、State 列挙型を返します。
<code>isClosed</code>	案件が「終了」状態かどうかを判定します。
<code>isNotRunning</code>	案件が「停止中」状態かどうかを判定します。
<code>isOpen</code>	案件が「未終了」状態かどうかを判定します。
<code>isSuspended</code>	案件が「中断」状態かどうかを判定します。
<code>toShortState</code>	案件の状態を示す State の短縮表記形式を取得します。
<code>toState</code>	案件の状態を示す State の完全表記形式を取得します。
<code>toStateCode</code>	案件の状態を示す State のコード値を取得します。
<code>toString</code>	案件の状態の文字列表現を取得します。
<code>valueOf</code>	指定した名前を持つこの型の列挙型定数を返します。
<code>values</code>	この列挙型の定数が含まれている配列を宣言されている順序で返します。

## java.lang.Enum クラスから継承されたメソッド

`clone`, `compareTo`, `equals`, `getDeclaringClass`, `hashCode`, `name`, `ordinal`, `valueOf`

java.lang.Object クラスから継承されたメソッド  
finalize, getClass, notify, notifyAll, wait, wait, wait

## fromStateCode

列挙型：CIWProcessInstance.State

### 構文

```
public static CIWProcessInstance.State fromStateCode(  
    java.lang.String aCode  
)  
    throws CIWFatalException
```

### 機能

指定した State のコード値に対応した案件の状態を定数を持つ、State 列挙型を返します。

### 引数

fromStateCode の引数を次の表に示します。

表 7-37 fromStateCode の引数 (CIWProcessInstance.State の場合)

仮引数名	名称	in/out	説明
aCode	State のコード値	in	取得したい State のコード値を指定します。

### 戻り値

State のコード値に対応する状態を定数を持つ State 列挙型を返します。

### 例外

fromStateCode で発生する例外を次の表に示します。

表 7-38 fromStateCode で発生する例外 (CIWProcessInstance.State の場合)

発生する例外	説明
<a href="#">CIWFatalException</a>	処理を続行できない障害が発生した場合

### 注意事項

- State のコード値に「z」または存在しないコードを指定した場合は、例外 (CIWFatalException) が発生します。
- State のコード値については、「[付録 B 参照系 API で指定できる属性](#)」を参照してください。

## isClosed

列挙型：CIWProcessInstance.State

### 構文

```
public boolean isClosed()
```

### 機能

案件が「終了」状態かどうかを判定します。

### 引数

ありません。

### 戻り値

- 案件が「終了」状態の場合は true を返します。
- 案件が「終了」状態以外の場合は false を返します。

### 例外

ありません。

## isNotRunning

列挙型：CIWProcessInstance.State

### 構文

```
public boolean isNotRunning()
```

### 機能

案件が「停止中」状態かどうかを判定します。

### 引数

ありません。

### 戻り値

- 案件が「停止中」状態の場合は true を返します。
- 案件が「停止中」状態以外の場合は false を返します。

## 例外

ありません。

## isOpen

列挙型：CIWProcessInstance.State

## 構文

```
public boolean isOpen()
```

## 機能

案件が「未終了」状態かどうかを判定します。

## 引数

ありません。

## 戻り値

- 案件が「未終了」状態の場合は true を返します。
- 案件が「未終了」状態以外の場合は false を返します。

## 例外

ありません。

## isSuspended

列挙型：CIWProcessInstance.State

## 構文

```
public boolean isSuspended()
```

## 機能

案件が「中断」状態かどうかを判定します。

## 引数

ありません。

## 戻り値

- 案件が「中断」状態の場合は true を返します。
- 案件が「中断」状態以外の場合は false を返します。

## 例外

ありません。

## toShortState

列挙型：CIWProcessInstance.State

## 構文

```
public final java.lang.String toShortState()
```

## 機能

案件の状態を示す State の短縮表記形式を取得します。

## 引数

ありません。

## 戻り値

State の短縮表記形式を返します。

## 例外

ありません。

## 注意事項

State の短縮表記形式については、「[付録 B 参照系 API で指定できる属性](#)」を参照してください。

## toState

列挙型：CIWProcessInstance.State

## 構文

```
public final java.lang.String toState()
```

## 機能

案件の状態を示す State の完全表記形式を取得します。

## 引数

ありません。

## 戻り値

State の完全表記形式を返します。

## 例外

ありません。

## 注意事項

State の完全表記形式については、「付録 B 参照系 API で指定できる属性」を参照してください。

## toStateCode

列挙型：CIWProcessInstance.State

## 構文

```
public final java.lang.String toStateCode()
```

## 機能

案件の状態を示す State のコード値を取得します。

## 引数

ありません。

## 戻り値

StateCode の値を返します。

## 例外

ありません。

## 注意事項

State のコード値については、「付録 B 参照系 API で指定できる属性」を参照してください。

# toString

列挙型：CIWProcessInstance.State

## 構文

```
public final java.lang.String toString()
```

## 機能

案件の状態の文字列表現を取得します。

表示形式は、「状態の定数（状態のコード値）」になります。

## 引数

ありません。

## オーバーライド

java.lang.Enum<CIWProcessInstance.State>クラス内の toString

## 戻り値

案件の状態の文字列表現を返します。

## 例外

ありません。

## 注意事項

案件の状態の文字列表現については、「[付録 B 参照系 API で指定できる属性](#)」を参照してください。

# valueOf

列挙型：CIWProcessInstance.State

## 構文

```
public static CIWProcessInstance.State valueOf(  
    java.lang.String name  
)
```

## 機能

指定した名前を持つこの型の列挙型定数を返します。



文字列は、この型の列挙型定数を宣言するのに使用した識別子と完全に一致している必要があります。余分な空白文字を含めることはできません。

## 引数

valueOf の引数を次の表に示します。

表 7-39 valueOf の引数 (CIWProcessInstance.State の場合)

仮引数名	名称	in/out	説明
name	列挙型定数の名前	in	返される列挙型定数の名前を指定します。

## 戻り値

指定された名前を持つ列挙型定数を返します。

## 例外

valueOf で発生する例外を次の表に示します。

表 7-40 valueOf で発生する例外 (CIWProcessInstance.State の場合)

発生する例外	説明
java.lang.IllegalArgumentException	指定された名前を持つ定数をこの列挙型が持っていない場合

## values

列挙型 : CIWProcessInstance.State

## 構文

```
public static CIWProcessInstance.State[]  
    values()
```

## 機能

この列挙型の定数が含まれている配列を宣言されている順序で返します。

このメソッドは次のようにして定数を繰り返すために使用できます。

```
for(CIWProcessInstance.State c :  
    CIWProcessInstance.State.values())  
    System.out.println(c);
```

## 引数

ありません。

## 戻り値

この列挙型の定数が含まれている配列を宣言されている順序で返します。

## 例外

ありません。

## 7.10 CIWServer.Targetkind (ワーク管理システムが管理するオブジェクトの種別を表す列挙型)

### 概要

ワーク管理システムが管理するオブジェクトの種別を表す列挙型です。

### 定義

```
public static enum CIWServer.Targetkind
    extends java.lang.Enum<CIWServer.Targetkind>
```

### すべての実装されたインタフェース

```
java.io.Serializable,
java.lang.Comparable<CIWServer.Targetkind>
```

### 含まれているインタフェース

```
CIWServer
```

### 定数

CIWServer.Targetkind の列挙型定数を次の表に示します。

表 7-41 CIWServer.Targetkind の列挙型定数

定数名	説明および記述形式
ACTIVITY_DEFINITION	「業務ステップ定義」を表す定数です。
	<code>public static final CIWServer.Targetkind ACTIVITY_DEFINITION</code>
ACTIVITY_INSTANCE	「業務ステップ」を表す定数です。
	<code>public static final CIWServer.Targetkind ACTIVITY_INSTANCE</code>
PROCESS_DEFINITION	「ビジネスプロセス定義」を表す定数です。
	<code>public static final CIWServer.Targetkind PROCESS_DEFINITION</code>
PROCESS_INSTANCE	「案件」を表す定数です。
	<code>public static final CIWServer.Targetkind PROCESS_INSTANCE</code>
WORK_DEFINITION	「作業定義」を表す定数です。
	<code>public static final CIWServer.Targetkind WORK_DEFINITION</code>
WORK_ITEM	「作業」を表す定数です。
	<code>public static final CIWServer.Targetkind WORK_ITEM</code>

### メソッド

CIWServer.Targetkind のメソッドを次の表に示します。

表 7-42 CIWServer.Targetkind のメソッド

メソッド	説明
toTargetkind	ワーク管理システムが管理するオブジェクトの種別の名称を取得します。
valueOf	指定した名前を持つこの型の列挙型定数を返します。
values	この列挙型の定数が含まれている配列を宣言されている順序で返します。

#### java.lang.Enum クラスから継承されたメソッド

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

#### java.lang.Object クラスから継承されたメソッド

finalize, getClass, notify, notifyAll, wait, wait, wait

## toTargetkind

列挙型：CIWServer.Targetkind

### 構文

```
public final java.lang.String toTargetkind()
```

### 機能

ワーク管理システムが管理するオブジェクトの種別の名称を取得します。

### 引数

ありません。

### 戻り値

ワーク管理システムが管理するオブジェクトの種別の名称を返します。

### 例外

ありません。

## valueOf

列挙型：CIWServer.Targetkind

## 構文

```
public static CIWServer.Targetkind valueOf(  
    java.lang.String name  
)
```

## 機能

指定した名前を持つこの型の列挙型定数を返します。

文字列は、この型の列挙型定数を宣言するのに使用した識別子と完全に一致している必要があります。余分な空白文字を含めることはできません。

## 引数

valueOf の引数を次の表に示します。

表 7-43 valueOf の引数 (CIWServer.Targetkind の場合)

仮引数名	名称	in/out	説明
name	列挙型定数の名前	in	返される列挙型定数の名前を指定します。

## 戻り値

指定された名前を持つ列挙型定数を返します。

## 例外

valueOf で発生する例外を次の表に示します。

表 7-44 valueOf で発生する例外 valueOf の引数 (CIWServer.Targetkind の場合)

発生する例外	説明
java.lang.IllegalArgumentException	指定された名前を持つ定数をこの列挙型が持っていない場合

## values

列挙型 : CIWServer.Targetkind

## 構文

```
public static CIWServer.Targetkind[]  
    values()
```

## 機能

この列挙型の定数が含まれている配列を宣言されている順序で返します。

このメソッドは次のようにして定数を繰り返すために使用できます。

```
for(CIWServer.Targetkind c :  
    CIWServer.Targetkind.values())  
    System.out.println(c);
```

## 引数

ありません。

## 戻り値

この列挙型の定数が含まれている配列を宣言されている順序で返します。

## 例外

ありません。

## 7.11 CIWUtil.InfinityDate (無限遠の日付の列挙型)

### 概要

無限遠の日付の列挙型です。

### 定義

```
public static enum CIWUtil.InfinityDate
    extends java.lang.Enum<CIWUtil.InfinityDate>
```

### すべての実装されたインタフェース

```
java.io.Serializable,
java.lang.Comparable<CIWUtil.InfinityDate>
```

### 含まれているインタフェース

```
CIWUtil
```

### 定数

CIWUtil.InfinityDate の列挙型定数を次の表に示します。

表 7-45 CIWUtil.InfinityDate の列挙型定数

定数名	説明および記述形式
BEYOND	<p>「無限遠の未来」を表す定数です。</p> <ul style="list-style-type: none"><li>日付情報を取得するメソッドによって、無限遠の未来を表す日付情報を取得した場合、292278994/08/17 07:12:55 GMT を表す java.util.Date 型オブジェクトが取得されます。</li><li>日付情報を取得するメソッドによって取得される、無限遠の未来を表す日付情報の値は、将来変更される場合があります。</li></ul>
	<pre>public static final CIWUtil.InfinityDate BEYOND</pre>
ORIGIN	<p>「無限遠の過去」を表す定数です。</p> <ul style="list-style-type: none"><li>日付情報を取得するメソッドによって、無限遠の過去を表す日付情報を取得した場合、1970/01/01 00:00:00 GMT を表す java.util.Date 型オブジェクトが取得されます。</li><li>日付情報を取得するメソッドによって取得される、無限遠の過去を表す日付情報の値は、将来変更される場合があります。</li></ul>
	<pre>public static final CIWUtil.InfinityDate ORIGIN</pre>

### メソッド

CIWUtil.InfinityDate のメソッドを次の表に示します。

表 7-46 CIWUtil.InfinityDate のメソッド

メソッド	説明
toDate	定数の表す日付情報を取得します。
valueOf	指定した名前を持つこの型の列挙型定数を返します。
values	この列挙型の定数が含まれている配列を宣言されている順序で返します。

java.lang.Enum クラスから継承されたメソッド

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

java.lang.Object クラスから継承されたメソッド

finalize, getClass, notify, notifyAll, wait, wait, wait

## toDate

列挙型：CIWUtil.InfinityDate

### 構文

```
public final java.util.Date toDate()
```

### 機能

定数の表す日付情報を取得します。

### 引数

ありません。

### 戻り値

定数の表す日付情報を返します。

### 例外

ありません。

## valueOf

列挙型：CIWUtil.InfinityDate

### 構文

```
public static CIWUtil.InfinityDate valueOf(  
    java.lang.String name  
)
```

### 機能

指定した名前を持つこの型の列挙型定数を返します。



文字列は、この型の列挙型定数を宣言するのに使用した識別子と完全に一致している必要があります。余分な空白文字を含めることはできません。

## 引数

valueOf の引数を次の表に示します。

表 7-47 valueOf の引数 (CIWUtil.InfinityDate の場合)

仮引数名	名称	in/out	説明
name	列挙型定数の名前	in	返される列挙型定数の名前を指定します。

## 戻り値

指定された名前を持つ列挙型定数を返します。

## 例外

valueOf で発生する例外を次の表に示します。

表 7-48 valueOf で発生する例外 (CIWUtil.InfinityDate の場合)

発生する例外	説明
java.lang.IllegalArgumentException	指定された名前を持つ定数をこの列挙型が持っていない場合

## values

列挙型 : CIWUtil.InfinityDate

## 構文

```
public static CIWUtil.InfinityDate[]  
    values()
```

## 機能

この列挙型の定数が含まれている配列を宣言されている順序で返します。

このメソッドは次のようにして定数を繰り返すために使用できます。

```
for(CIWUtil.InfinityDate c :  
    CIWUtil.InfinityDate.values())  
    System.out.println(c);
```

## 引数

ありません。

## 戻り値

この列挙型の定数が含まれている配列を宣言されている順序で返します。

## 例外

ありません。

## 7.12 CIWorkDefinition.AttributeName (作業定義属性の属性名の列挙型)

### 概要

作業定義属性の属性名の列挙型です。

### 定義

```
public static enum CIWorkDefinition.AttributeName
    extends java.lang.Enum<CIWorkDefinition.AttributeName>
```

### すべての実装されたインタフェース

```
java.io.Serializable,
    java.lang.Comparable<CIWorkDefinition.AttributeName>
```

### 含まれているインタフェース

```
CIWorkDefinition
```

### 定数

CIWorkDefinition.AttributeName の列挙型定数を次の表に示します。

表 7-49 CIWorkDefinition.AttributeName の列挙型定数

定数名	説明および記述形式
ACTIVITY_DEFINITION_ID	「作業定義が所属する業務ステップ定義の ID」を表す定数です。 <pre>public static final CIWorkDefinition.AttributeName ACTIVITY_DEFINITION_ID</pre>
CASTING_RULE_NAME	「振り分けルール名」を表す定数です。 <pre>public static final CIWorkDefinition.AttributeName CASTING_RULE_NAME</pre>
DESCRIPTION	「作業の説明」を表す定数です。 <pre>public static final CIWorkDefinition.AttributeName DESCRIPTION</pre>
ID	「作業定義の ID」を表す定数です。 <pre>public static final CIWorkDefinition.AttributeName ID</pre>
NAME	「作業定義の名称」を表す定数です。 <pre>public static final CIWorkDefinition.AttributeName NAME</pre>
PROCESS_DEFINITION_ID	「作業定義が所属するビジネスプロセス定義の ID」を表す定数です。 <pre>public static final CIWorkDefinition.AttributeName PROCESS_DEFINITION_ID</pre>
TYPE	「作業の種類」を表す定数です。 <pre>public static final CIWorkDefinition.AttributeName TYPE</pre>

## メソッド

CIWWWorkDefinition.AttributeName のメソッドを次の表に示します。

表 7-50 CIWWWorkDefinition.AttributeName のメソッド

メソッド	説明
<code>toAttributeName</code>	作業定義属性の属性名（属性を格納するテーブルのカラム名）を取得します。
<code>valueOf</code>	指定した名前を持つこの型の列挙型定数を返します。
<code>values</code>	この列挙型の定数が含まれている配列を宣言されている順序で返します。

### java.lang.Enum クラスから継承されたメソッド

`clone`, `compareTo`, `equals`, `getDeclaringClass`, `hashCode`, `name`, `ordinal`, `toString`, `valueOf`

### java.lang.Object クラスから継承されたメソッド

`finalize`, `getClass`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

## toAttributeName

列挙型：CIWWWorkDefinition.AttributeName

### 構文

```
public final java.lang.String toAttributeName()
```

### 機能

作業定義属性の属性名（属性を格納するテーブルのカラム名）を取得します。

### 引数

ありません。

### 戻り値

作業定義属性の属性名を返します。

### 例外

ありません。

## valueOf

列挙型：CIWWWorkDefinition.AttributeName

## 構文

```
public static CIWorkDefinition.AttributeName valueOf(  
    java.lang.String name  
)
```

## 機能

指定した名前を持つこの型の列挙型定数を返します。

文字列は、この型の列挙型定数を宣言するのに使用した識別子と完全に一致している必要があります。余分な空白文字を含めることはできません。

## 引数

valueOf の引数を次の表に示します。

表 7-51 valueOf の引数 (CIWorkDefinition.AttributeName の場合)

仮引数名	名称	in/out	説明
name	列挙型定数の名前	in	返される列挙型定数の名前を指定します。

## 戻り値

指定された名前を持つ列挙型定数を返します。

## 例外

valueOf で発生する例外を次の表に示します。

表 7-52 valueOf で発生する例外 (CIWorkDefinition.AttributeName の場合)

発生する例外	説明
java.lang.IllegalArgumentException	指定された名前を持つ定数をこの列挙型が持っていない場合

## values

列挙型 : CIWorkDefinition.AttributeName

## 構文

```
public static CIWorkDefinition.AttributeName[]  
    values()
```

## 機能

この列挙型の定数が含まれている配列を宣言されている順序で返します。

このメソッドは次のようにして定数を繰り返すために使用できます。

```
for(CIWorkDefinition.AttributeName c :  
    CIWorkDefinition.AttributeName.values())  
    System.out.println(c);
```

## 引数

ありません。

## 戻り値

この列挙型の定数が含まれている配列を宣言されている順序で返します。

## 例外

ありません。

## 7.13 CIWorkDefinition.Type (作業定義の種類の種類列挙型)

### 概要

作業定義の種類の種類列挙型です。

### 定義

```
public static enum CIWorkDefinition.Type
    extends java.lang.Enum<CIWorkDefinition.Type>
```

### すべての実装されたインタフェース

```
java.io.Serializable,
java.lang.Comparable<CIWorkDefinition.Type>
```

### 含まれているインタフェース

```
CIWorkDefinition
```

### 定数

CIWorkDefinition.Type の列挙型定数を次の表に示します。

表 7-53 CIWorkDefinition.Type の列挙型定数

定数名	説明および記述形式
BUILTIN_CONCURRENT_ACTIVITY*	「並列業務ステップ作業」を表す定数です。 <ul style="list-style-type: none"><li>完全表記形式: builtin.concurrentActivity</li><li>コード値: a</li></ul>
	<pre>public static final CIWorkDefinition.Type BUILTIN_CONCURRENT_ACTIVITY</pre>
BUILTIN_CONCURRENT_WORK	「並列作業」を表す定数です。 <ul style="list-style-type: none"><li>完全表記形式: builtin.concurrentWork</li><li>コード値: b</li></ul>
	<pre>public static final CIWorkDefinition.Type BUILTIN_CONCURRENT_WORK</pre>
BUILTIN_FOLLOWING_ACTIVITY*	「先手通知作業」を表す定数です。 <ul style="list-style-type: none"><li>完全表記形式: builtin.followingActivity</li><li>コード値: d</li></ul>
	<pre>public static final CIWorkDefinition.Type BUILTIN_FOLLOWING_ACTIVITY</pre>
BUILTIN_TIMER*	「計時作業」を表す定数です。 <ul style="list-style-type: none"><li>完全表記形式: builtin.timer</li><li>コード値: c</li></ul>
	<pre>public static final CIWorkDefinition.Type BUILTIN_TIMER</pre>
NORMAL	「一般作業」を表す定数です。

定数名	説明および記述形式
	<ul style="list-style-type: none"> <li>完全表記形式：normal</li> <li>コード値：0</li> </ul> <pre>public static final CIWorkDefinition.Type NORMAL</pre>
UNDEFINED	「未定義」を表す定数です。 <ul style="list-style-type: none"> <li>完全表記形式：undefined</li> <li>コード値：z</li> </ul> <pre>public static final CIWorkDefinition.Type UNDEFINED</pre>

注※

CSCIW ではサポートしていません。

## メソッド

CIWorkDefinition.Type のメソッドを次の表に示します。

表 7-54 CIWorkDefinition.Type のメソッド

メソッド	説明
fromTypeCode	指定した Type のコード値に対応した作業定義の種類を定数に持つ、Type 列挙型を返します。
toString	作業定義の種類 of 文字列表現を取得します。
toType	作業定義の種類を示す Type の完全表記形式を取得します。
toTypeCode	作業定義の種類を示す Type のコード値を取得します。
valueOf	指定した名前を持つこの型の列挙型定数を返します。
values	この列挙型の定数が含まれている配列を宣言されている順序で返します。

### java.lang.Enum クラスから継承されたメソッド

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, valueOf

### java.lang.Object クラスから継承されたメソッド

finalize, getClass, notify, notifyAll, wait, wait, wait

## fromTypeCode

列挙型：CIWorkDefinition.Type

### 構文

```
public static CIWorkDefinition.Type fromTypeCode(
    java.lang.String aCode
)
    throws CIWFatalException
```



## 機能

指定した Type のコード値に対応した作業定義の種類を定数を持つ、Type 列挙型を返します。

## 引数

fromTypeCode の引数を次の表に示します。

表 7-55 fromTypeCode の引数 (CIWWWorkDefinition.Type の場合)

仮引数名	名称	in/out	説明
aCode	Type のコード値	in	取得したい Type のコード値を指定します。

## 戻り値

Type のコード値に対応する種類を定数を持つ Type 列挙型を返します。

## 例外

fromTypeCode で発生する例外を次の表に示します。

表 7-56 fromTypeCode で発生する例外 (CIWWWorkDefinition.Type の場合)

発生する例外	説明
CIWFFatalException	処理を続行できない障害が発生した場合

## 注意事項

- Type のコード値に「z」または存在しないコードを指定した場合は、例外 (CIWFFatalException) が発生します。
- Type のコード値については、「付録 B 参照系 API で指定できる属性」を参照してください。

## toString

列挙型 : CIWWWorkDefinition.Type

## 構文

```
public final java.lang.String toString()
```

## 機能

作業定義の種類 of 文字列表現を取得します。

表示形式は、「種類の定数 (種類のコード値)」になります。

## 引数

ありません。

## オーバーライド

java.lang.Enum<CIWorkDefinition.Type>クラス内の toString

## 戻り値

作業定義の種類 of 文字列表現を返します。

## 例外

ありません。

## 注意事項

作業定義の種類 of 文字列表現については、「付録 B 参照系 API で指定できる属性」を参照してください。

## toType

列挙型：CIWorkDefinition.Type

## 構文

```
public final java.lang.String toType()
```

## 機能

作業定義の種類を示す Type の完全表記形式を取得します。

## 引数

ありません。

## 戻り値

Type の完全表記形式を返します。

## 例外

ありません。

## 注意事項

Type の完全表記形式については、「付録 B 参照系 API で指定できる属性」を参照してください。

## toTypeCode

列挙型：CIWorkDefinition.Type

### 構文

```
public final java.lang.String toTypeCode()
```

### 機能

作業定義の種類を示す Type のコード値を取得します。

### 引数

ありません。

### 戻り値

Type のコード値を返します。

### 例外

ありません。

### 注意事項

Type のコード値については、「[付録 B 参照系 API で指定できる属性](#)」を参照してください。

## valueOf

列挙型：CIWorkDefinition.Type

### 構文

```
public static CIWorkDefinition.Type valueOf(  
    java.lang.String name  
)
```

### 機能

指定した名前を持つこの型の列挙型定数を返します。

文字列は、この型の列挙型定数を宣言するのに使用した識別子と完全に一致している必要があります。余分な空白文字を含めることはできません。

## 引数

valueOf の引数を次の表に示します。

表 7-57 valueOf の引数 (CIWorkDefinition.Type の場合)

仮引数名	名称	in/out	説明
name	列挙型定数の名前	in	返される列挙型定数の名前を指定します。

## 戻り値

指定された名前を持つ列挙型定数を返します。

## 例外

valueOf で発生する例外を次の表に示します。

表 7-58 valueOf で発生する例外 (CIWorkDefinition.Type の場合)

発生する例外	説明
java.lang.IllegalArgumentException	指定された名前を持つ定数をこの列挙型が持っていない場合

## values

列挙型 : CIWorkDefinition.Type

## 構文

```
public static CIWorkDefinition.Type[]  
    values()
```

## 機能

この列挙型の定数が含まれている配列を宣言されている順序で返します。

このメソッドは次のようにして定数を繰り返すために使用できます。

```
for(CIWorkDefinition.Type c :  
    CIWorkDefinition.Type.values())  
    System.out.println(c);
```

## 引数

ありません。

## 戻り値

この列挙型の定数が含まれている配列を宣言されている順序で返します。

## 例外

ありません。

## 7.14 CIWorkItem.AttributeName (作業属性の属性名の列挙型)

### 概要

作業属性の属性名の列挙型です。

### 定義

```
public static enum CIWorkItem.AttributeName
    extends java.lang.Enum<CIWorkItem.AttributeName>
```

### すべての実装されたインタフェース

```
java.io.Serializable,
java.lang.Comparable<CIWorkItem.AttributeName>
```

### 含まれているインタフェース

```
CIWorkItem
```

### 定数

CIWorkItem.AttributeName の列挙型定数を次の表に示します。

表 7-59 CIWorkItem.AttributeName の列挙型定数

定数名	説明および記述形式
ACTIVITY_INSTANCE_ID	「作業が所属する業務ステップの ID」を表す定数です。
	<code>public static final CIWorkItem.AttributeName ACTIVITY_INSTANCE_ID</code>
CLOSED_DATE	「作業の終了日時」を表す定数です。
	<code>public static final CIWorkItem.AttributeName CLOSED_DATE</code>
CREATION_DATE	「作業の発生日時」を表す定数です。
	<code>public static final CIWorkItem.AttributeName CREATION_DATE</code>
DEADLINE	「作業の処理期限の絶対日時」を表す定数です。 作業生成時に、所属する業務ステップの処理期限に設定されている値が、作業の処理期限として設定されます。 未設定時は、null が設定されています。
	<code>public static final CIWorkItem.AttributeName DEADLINE</code>
ID	「作業の ID」を表す定数です。
	<code>public static final CIWorkItem.AttributeName ID</code>
NAME	「作業の名称 (作業キー)」を表す定数です。
	<code>public static final CIWorkItem.AttributeName NAME</code>
PARTICIPANT	「作業者の ID」を表す定数です。
	<code>public static final CIWorkItem.AttributeName PARTICIPANT</code>

定数名	説明および記述形式
PRIORITY	「作業の優先度」を表す定数です。 <code>public static final CIWorkItem.AttributeName PRIORITY</code>
PROCESS_DEFINITION_ID	「作業が所属するビジネスプロセス定義の ID」を表す定数です。 <code>public static final CIWorkItem.AttributeName PROCESS_DEFINITION_ID</code>
PROCESS_INSTANCE_ID	「作業が所属する案件の ID」を表す定数です。 <code>public static final CIWorkItem.AttributeName PROCESS_INSTANCE_ID</code>
PROCESS_INSTANCE_NAME	「案件名 (案件キー)」を表す定数です。 <code>public static final CIWorkItem.AttributeName PROCESS_INSTANCE_NAME</code>
START_DATE	「作業の開始日時」を表す定数です。 <code>public static final CIWorkItem.AttributeName START_DATE</code>
STATE	「作業の状態」を表す定数です。 <code>public static final CIWorkItem.AttributeName STATE</code>
TYPE	「作業の種類」を表す定数です。 <code>public static final CIWorkItem.AttributeName TYPE</code>
WORK_DEFINITION_ID	「作業定義の ID」を表す定数です。 <code>public static final CIWorkItem.AttributeName WORK_DEFINITION_ID</code>
WORK_DEFINITION_NAME	「作業定義の名称」を表す定数です。 <code>public static final CIWorkItem.AttributeName WORK_DEFINITION_NAME</code>

## メソッド

CIWorkItem.AttributeName のメソッドを次の表に示します。

表 7-60 CIWorkItem.AttributeName のメソッド

メソッド	説明
<code>toAttributeName</code>	作業属性の属性名 (属性を格納するテーブルのカラム名) を取得します。
<code>valueOf</code>	指定した名前を持つこの型の列挙型定数を返します。
<code>values</code>	この列挙型の定数が含まれている配列を宣言されている順序で返します。

### java.lang.Enum クラスから継承されたメソッド

`clone`, `compareTo`, `equals`, `getDeclaringClass`, `hashCode`, `name`, `ordinal`, `toString`, `valueOf`

### java.lang.Object クラスから継承されたメソッド

`finalize`, `getClass`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

## toAttributeName

列挙型：CIWorkItem.AttributeName

### 構文

```
public final java.lang.String toAttributeName()
```

### 機能

作業属性の属性名（属性を格納するテーブルのカラム名）を取得します。

### 引数

ありません。

### 戻り値

作業属性の属性名を返します。

### 例外

ありません。

## valueOf

列挙型：CIWorkItem.AttributeName

### 構文

```
public static CIWorkItem.AttributeName valueOf(  
    java.lang.String name  
)
```

### 機能

指定した名前を持つこの型の列挙型定数を返します。

文字列は、この型の列挙型定数を宣言するのに使用した識別子と完全に一致している必要があります。余分な空白文字を含めることはできません。

### 引数

valueOf の引数を次の表に示します。



表 7-61 valueOf の引数 (CIWorkItem.AttributeName の場合)

仮引数名	名称	in/out	説明
name	列挙型定数の名前	in	返される列挙型定数の名前を指定します。

## 戻り値

指定された名前を持つ列挙型定数を返します。

## 例外

valueOf で発生する例外を次の表に示します。

表 7-62 valueOf で発生する例外 (CIWorkItem.AttributeName の場合)

発生する例外	説明
java.lang.IllegalArgumentException	指定された名前を持つ定数をこの列挙型が持っていない場合

## values

列挙型 : CIWorkItem.AttributeName

## 構文

この列挙型の定数が含まれている配列を宣言されている順序で返します。

このメソッドは次のようにして定数を繰り返すために使用できます。

```
public static CIWorkItem.AttributeName[]
    values()
```

## 機能

```
for(CIWorkItem.AttributeName c :
    CIWorkItem.AttributeName.values())
    System.out.println(c);
```

## 引数

ありません。

## 戻り値

この列挙型の定数が含まれている配列を宣言されている順序で返します。

## 例外

ありません。

## 7.15 CIWorkItem.State (作業の状態の列挙型)

### 概要

作業の状態の列挙型です。

### 定義

```
public static enum CIWorkItem.State
    extends java.lang.Enum<CIWorkItem.State>
```

### すべての実装されたインタフェース

```
java.io.Serializable,
java.lang.Comparable<CIWorkItem.State>
```

### 含まれているインタフェース

```
CIWorkItem
```

### 定数

CIWorkItem.State の列挙型定数を次の表に示します。

表 7-63 CIWorkItem.State の列挙型定数

定数名	説明および記述形式
ABORTED*	「異常終了」を表す定数です。 <ul style="list-style-type: none"><li>短縮表記形式：aborted</li><li>完全表記形式：closed.aborted</li><li>コード値：v</li></ul>
	<pre>public static final CIWorkItem.State ABORTED</pre>
CANCELED	「実行取消」を表す定数です。 <ul style="list-style-type: none"><li>短縮表記形式：canceled</li><li>完全表記形式：closed.completed.canceled</li><li>コード値：q</li></ul>
	<pre>public static final CIWorkItem.State CANCELED</pre>
DISABLED	「実行開始不可」を表す定数です。 <ul style="list-style-type: none"><li>短縮表記形式：disabled</li><li>完全表記形式：open.notRunning.suspended.disabled</li><li>コード値：l</li></ul>
	<pre>public static final CIWorkItem.State DISABLED</pre>
EXECUTED	「実行済」を表す定数です。 <ul style="list-style-type: none"><li>短縮表記形式：executed</li><li>完全表記形式：closed.completed.executed</li><li>コード値：r</li></ul>
	<pre>public static final CIWorkItem.State EXECUTED</pre>

定数名	説明および記述形式
EXECUTING	<p>「自動実行」を表す定数です。</p> <ul style="list-style-type: none"> <li>短縮表記形式：executing</li> <li>完全表記形式：open.running.executing</li> <li>コード値：e</li> </ul> <pre>public static final CIWorkItem.State EXECUTING</pre>
INITIAL	<p>「初期」を表す定数です。</p> <ul style="list-style-type: none"> <li>短縮表記形式：initial</li> <li>完全表記形式：open.notRunning.notStarted.initial</li> <li>コード値：i</li> </ul> <pre>public static final CIWorkItem.State INITIAL</pre>
INTERMITTED	<p>「実行停止」を表す定数です。</p> <ul style="list-style-type: none"> <li>短縮表記形式：intermitted</li> <li>完全表記形式：open.notRunning.suspended.intermitted</li> <li>コード値：m</li> </ul> <pre>public static final CIWorkItem.State INTERMITTED</pre>
NOT_EXECUTED	<p>「実行省略」を表す定数です。</p> <ul style="list-style-type: none"> <li>短縮表記形式：notExecuted</li> <li>完全表記形式：closed.completed.notExecuted</li> <li>コード値：p</li> </ul> <pre>public static final CIWorkItem.State NOT_EXECUTED</pre>
PERFORMING	<p>「作業実行」を表す定数です。</p> <ul style="list-style-type: none"> <li>短縮表記形式：performing</li> <li>完全表記形式：open.running.performing</li> <li>コード値：f</li> </ul> <pre>public static final CIWorkItem.State PERFORMING</pre>
READY	<p>「実行開始可能」を表す定数です。</p> <ul style="list-style-type: none"> <li>短縮表記形式：ready</li> <li>完全表記形式：open.notRunning.notStarted.ready</li> <li>コード値：j</li> </ul> <pre>public static final CIWorkItem.State READY</pre>
RUNNING	<p>「実行中」を表す定数です。</p> <ul style="list-style-type: none"> <li>短縮表記形式：running</li> <li>完全表記形式：open.running</li> <li>コード値：d</li> </ul> <pre>public static final CIWorkItem.State RUNNING</pre>
TERMINATED	<p>「強制終了」を表す定数です。</p> <ul style="list-style-type: none"> <li>短縮表記形式：terminated</li> </ul>

定数名	説明および記述形式
	<ul style="list-style-type: none"> <li>完全表記形式：closed.terminated</li> <li>コード値：u</li> </ul> <pre>public static final CIWorkItem.State TERMINATED</pre>
UNDEFINED	「未定義」を表す定数です。 <ul style="list-style-type: none"> <li>短縮表記形式：undefined</li> <li>完全表記形式：undefined</li> <li>コード値：z</li> </ul> <pre>public static final CIWorkItem.State UNDEFINED</pre>

注※

CSCIW ではサポートしていません。

## メソッド

CIWorkItem.State のメソッドを次の表に示します。

表 7-64 CIWorkItem.State のメソッド

メソッド	説明
<code>fromStateCode</code>	指定した State のコード値に対応した作業の状態を定数に持つ、State 列挙型を返します。
<code>isClosed</code>	作業が「終了」状態かどうかを判定します。
<code>isCompleted</code>	作業が「完了」状態かどうかを判定します。
<code>isNotRunning</code>	作業が「停止中」状態かどうかを判定します。
<code>isNotStarted</code>	作業が「未実行」状態かどうかを判定します。
<code>isOpen</code>	作業が「未終了」状態かどうかを判定します。
<code>isRunning</code>	作業が「実行中」状態かどうかを判定します。
<code>isSuspended</code>	作業が「中断」状態かどうかを判定します。
<code>toShortState</code>	作業の状態を示す State の短縮表記形式を取得します。
<code>toState</code>	作業の状態を示す State の完全表記形式を取得します。
<code>toStateCode</code>	作業の状態を示す State のコード値を取得します。
<code>toString</code>	作業の状態の文字列表現を取得します。
<code>valueOf</code>	指定した名前を持つこの型の列挙型定数を返します。
<code>values</code>	この列挙型の定数が含まれている配列を宣言されている順序で返します。

### java.lang.Enum クラスから継承されたメソッド

`clone`, `compareTo`, `equals`, `getDeclaringClass`, `hashCode`, `name`, `ordinal`, `valueOf`

### java.lang.Object クラスから継承されたメソッド

`finalize`, `getClass`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

## fromStateCode

列挙型：CIWorkItem.State

### 構文

```
public static CIWorkItem.State fromStateCode(  
    java.lang.String aCode  
)  
    throws CIWFatalException
```

### 機能

指定した State のコード値に対応した作業の状態を定数を持つ、State 列挙型を返します。

### 引数

fromStateCode の引数を次の表に示します。

表 7-65 fromStateCode の引数 (CIWorkItem.State の場合)

仮引数名	名称	in/out	説明
aCode	State のコード値	in	取得したい State のコード値を指定します。

### 戻り値

State のコード値に対応する状態を定数を持つ State 列挙型を返します。

### 例外

fromStateCode で発生する例外を次の表に示します。

表 7-66 fromStateCode で発生する例外 (CIWorkItem.State の場合)

発生する例外	説明
<a href="#">CIWFatalException</a>	処理を続行できない障害が発生した場合

### 注意事項

- State のコード値に「z」または存在しないコードを指定した場合は、例外 (CIWFatalException) が発生します。
- State のコード値については、「[付録 B 参照系 API で指定できる属性](#)」を参照してください。

## isClosed

列挙型：CIWorkItem.State

## 構文

```
public boolean isClosed()
```

## 機能

作業が「終了」状態かどうかを判定します。

## 引数

ありません。

## 戻り値

- 作業が「終了」状態の場合は true を返します。
- 作業が「終了」状態以外の場合は false を返します。

## 例外

ありません。

## isCompleted

列挙型：CIWorkItem.State

## 構文

```
public boolean isCompleted()
```

## 機能

作業が「完了」状態かどうかを判定します。

## 引数

ありません。

## 戻り値

- 作業が「完了」状態の場合は true を返します。
- 作業が「完了」状態以外の場合は false を返します。

## 例外

ありません。

## isNotRunning

列挙型：CIWorkItem.State

### 構文

```
public boolean isNotRunning()
```

### 機能

作業が「停止中」状態かどうかを判定します。

### 引数

ありません。

### 戻り値

- 作業が「停止中」状態の場合は true を返します。
- 作業が「停止中」状態以外の場合は false を返します。

### 例外

ありません。

## isNotStarted

列挙型：CIWorkItem.State

### 構文

```
public boolean isNotStarted()
```

### 機能

作業が「未実行」状態かどうかを判定します。

### 引数

ありません。

### 戻り値

- 作業が「未実行」状態の場合は true を返します。
- 作業が「未実行」状態以外の場合は false を返します。



## 例外

ありません。

## isOpen

列挙型：CIWorkItem.State

## 構文

```
public boolean isOpen()
```

## 機能

作業が「未終了」状態かどうかを判定します。

## 引数

ありません。

## 戻り値

- 作業が「未終了」状態の場合は true を返します。
- 作業が「未終了」状態以外の場合は false を返します。

## 例外

ありません。

## isRunning

列挙型：CIWorkItem.State

## 構文

```
public boolean isRunning()
```

## 機能

作業が「実行中」状態かどうかを判定します。

## 引数

ありません。

## 戻り値

- 作業が「実行中」状態の場合は true を返します。
- 作業が「実行中」状態以外の場合は false を返します。

## 例外

ありません。

## isSuspended

列挙型：CIWorkItem.State

## 構文

```
public boolean isSuspended()
```

## 機能

作業が「中断」状態かどうかを判定します。

## 引数

ありません。

## 戻り値

- 作業が「中断」状態の場合は true を返します。
- 作業が「中断」状態以外の場合は false を返します。

## 例外

ありません。

## toShortState

列挙型：CIWorkItem.State

## 構文

```
public final java.lang.String toShortState()
```

## 機能

作業の状態を示す State の短縮表記形式を取得します。

## 引数

ありません。

## 戻り値

State の短縮表記形式を返します。

## 例外

ありません。

## 注意事項

State の短縮表記形式については、「[付録 B 参照系 API で指定できる属性](#)」を参照してください。

## toState

列挙型：CIWorkItem.State

## 構文

```
public final java.lang.String toState()
```

## 機能

作業の状態を示す State の完全表記形式を取得します。

## 引数

ありません。

## 戻り値

State の完全表記形式を返します。

## 例外

ありません。

## 注意事項

State の完全表記形式については、「[付録 B 参照系 API で指定できる属性](#)」を参照してください。

## toStateCode

列挙型：CIWorkItem.State

### 構文

```
public final java.lang.String toStateCode()
```

### 機能

作業の状態を示す State のコード値を取得します。

### 引数

ありません。

### 戻り値

State のコード値を返します。

### 例外

ありません。

### 注意事項

State のコード値については、「[付録 B 参照系 API で指定できる属性](#)」を参照してください。

## toString

列挙型：CIWorkItem.State

### 構文

```
public final java.lang.String toString()
```

### 機能

作業の状態の文字列表現を取得します。

表示形式は、「状態の定数（状態の StateCode の値）」となります。

### 引数

ありません。

## オーバーライド

java.lang.Enum<CIWWWorkItem.State>クラス内の toString

## 戻り値

作業の状態の文字列表現を返します。

## 例外

ありません。

## 注意事項

作業の状態の文字列表現については、「付録 B 参照系 API で指定できる属性」を参照してください。

## valueOf

列挙型：CIWWWorkItem.State

## 構文

```
public static CIWWWorkItem.State valueOf(  
    java.lang.String name  
)
```

## 機能

指定した名前を持つこの型の列挙型定数を返します。

文字列は、この型の列挙型定数を宣言するのに使用した識別子と完全に一致している必要があります。余分な空白文字を含めることはできません。

## 引数

valueOf の引数を次の表に示します。

表 7-67 valueOf の引数 (CIWWWorkItem.State の場合)

仮引数名	名称	in/out	説明
name	列挙型定数の名前	in	返される列挙型定数の名前を指定します。

## 戻り値

指定された名前を持つ列挙型定数を返します。

## 例外

valueOf で発生する例外を次の表に示します。

表 7-68 valueOf で発生する例外 (CIWorkItem.State の場合)

発生する例外	説明
java.lang.IllegalArgumentException	指定された名前を持つ定数をこの列挙型が持っていない場合

## values

列挙型 : CIWorkItem.State

## 構文

```
public static CIWorkItem.State[]  
    values()
```

## 機能

この列挙型の定数が含まれている配列を宣言されている順序で返します。

このメソッドは次のようにして定数を繰り返すために使用できます。

```
for(CIWorkItem.State c :  
    CIWorkItem.State.values())  
    System.out.println(c);
```

## 引数

ありません。

## 戻り値

この列挙型の定数が含まれている配列を宣言されている順序で返します。

## 例外

ありません。

## 7.16 CIWorkItem.Type (作業の種類)の列挙型

### 概要

作業の種類)の列挙型です。

### 定義

```
public static enum CIWorkItem.Type
    extends java.lang.Enum<CIWorkItem.Type>
```

### すべての実装されたインタフェース

```
java.io.Serializable,
java.lang.Comparable<CIWorkItem.Type>
```

### 含まれているインタフェース

```
CIWorkItem
```

### 定数

CIWorkItem.Type の列挙型定数を次の表に示します。

表 7-69 CIWorkItem.Type の列挙型定数

定数名	説明および記述形式
BUILTIN_CONCURRENT_ACTIVITY*	「並列業務ステップ作業」を表す定数です。 <ul style="list-style-type: none"><li>完全表記形式: builtin.concurrentActivity</li><li>コード値: a</li></ul>
	public static final CIWorkItem.Type BUILTIN_CONCURRENT_ACTIVITY
BUILTIN_CONCURRENT_WORK	「並列作業」を表す定数です。 <ul style="list-style-type: none"><li>完全表記形式: builtin.concurrentWork</li><li>コード値: b</li></ul>
	public static final CIWorkItem.Type BUILTIN_CONCURRENT_WORK
BUILTIN_FOLLOWING_ACTIVITY*	「先手通知作業」を表す定数です。 <ul style="list-style-type: none"><li>完全表記形式: builtin.followingActivity</li><li>コード値: d</li></ul>
	public static final CIWorkItem.Type BUILTIN_FOLLOWING_ACTIVITY
BUILTIN_TIMER*	「計時作業」を表す定数です。 <ul style="list-style-type: none"><li>完全表記形式: builtin.timer</li><li>コード値: c</li></ul>
	public static final CIWorkItem.Type BUILTIN_TIMER
ERROR_NOTICE*	「エラー回復作業」を表す定数です。 <ul style="list-style-type: none"><li>完全表記形式: errorNotice</li><li>コード値: 9</li></ul>

定数名	説明および記述形式
	<code>public static final CIWorkItem.Type ERROR_NOTICE</code>
NORMAL	「一般作業」を表す定数です。 <ul style="list-style-type: none"> <li>完全表記形式：normal</li> <li>コード値：0</li> </ul>
	<code>public static final CIWorkItem.Type NORMAL</code>
UNDEFINED	「未定義」を表す定数です。 <ul style="list-style-type: none"> <li>完全表記形式：undefined</li> <li>コード値：z</li> </ul>
	<code>public static final CIWorkItem.Type UNDEFINED</code>

注※

CSCIW ではサポートしていません。

## メソッド

CIWorkItem.Type のメソッドを次の表に示します。

表 7-70 CIWorkItem.Type のメソッド

メソッド	説明
<code>fromTypeCode</code>	指定した Type のコード値に対応した作業の種類を定数に持つ、Type 列挙型を返します。
<code>toString</code>	作業の種類 of 文字列表現を取得します。
<code>toType</code>	作業の種類を示す Type の完全表記形式を取得します。
<code>toTypeCode</code>	作業の種類を示す Type のコード値を取得します。
<code>valueOf</code>	指定した名前を持つこの型の列挙型定数を返します。
<code>values</code>	この列挙型の定数が含まれている配列を宣言されている順序で返します。

### java.lang.Enum クラスから継承されたメソッド

`clone`, `compareTo`, `equals`, `getDeclaringClass`, `hashCode`, `name`, `ordinal`, `valueOf`

### java.lang.Object クラスから継承されたメソッド

`finalize`, `getClass`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

## fromTypeCode

列挙型：CIWorkItem.Type

### 構文

```
public static CIWorkItem.Type fromTypeCode(
    java.lang.String aCode
```



```
)  
throws CIWFatalException
```

## 機能

指定した Type のコード値に対応した作業の種類を定数を持つ、Type 列挙型を返します。

## 引数

fromTypeCode の引数を次の表に示します。

表 7-71 fromTypeCode の引数 (CIWorkItem.Type の場合)

仮引数名	名称	in/out	説明
aCode	Type 列挙型の Type のコード値	in	取得したい Type 列挙型の Type のコード値を指定します。

## 戻り値

Type のコード値に対応する種類を定数を持つ Type 列挙型を返します。

## 例外

fromTypeCode で発生する例外を次の表に示します。

表 7-72 fromTypeCode で発生する例外 (CIWorkItem.Type の場合)

発生する例外	説明
<a href="#">CIWFatalException</a>	処理を続行できない障害が発生した場合

## 注意事項

- Type のコード値に「z」または存在しないコードを指定した場合は、例外 (CIWFatalException) が発生します。
- Type のコード値については、「[付録 B 参照系 API で指定できる属性](#)」を参照してください。

## toString

列挙型：CIWorkItem.Type

## 構文

```
public final java.lang.String toString()
```

## 機能

作業の種類の文字列表現を取得します。

表示形式は、「種類の定数（種類の WorkTypeCode の値）」となります。

## 引数

ありません。

## オーバーライド

java.lang.Enum<CIWWorItem.Type>クラス内の toString

## 戻り値

作業の種類の文字列表現を返します。

## 例外

ありません。

## 注意事項

作業の種類の文字列表現については、「付録 B 参照系 API で指定できる属性」を参照してください。

## toType

列挙型：CIWWorItem.Type

## 構文

```
public final java.lang.String toType()
```

## 機能

作業の種類を示す Type の完全表記形式を取得します。

## 引数

ありません。

## 戻り値

Type の完全表記形式を返します。

## 例外

ありません。

## 注意事項

Type の完全表記形式については、「[付録 B 参照系 API で指定できる属性](#)」を参照してください。

## toTypeCode

列挙型：CIWorkItem.Type

## 構文

```
public final java.lang.String toTypeCode()
```

## 機能

作業の種類を示す Type のコード値を取得します。

## 引数

ありません。

## 戻り値

Type のコード値を返します。

## 例外

ありません。

## 注意事項

Type のコード値については、「[付録 B 参照系 API で指定できる属性](#)」を参照してください。

## valueOf

列挙型：CIWorkItem.Type

## 構文

```
public static CIWorkItem.Type valueOf(  
    java.lang.String name  
)
```

## 機能

指定した名前を持つこの型の列挙型定数を返します。

文字列は、この型の列挙型定数を宣言するのに使用した識別子と完全に一致している必要があります。余分な空白文字を含めることはできません。

## 引数

valueOf の引数を次の表に示します。

表 7-73 valueOf の引数 (CIWorkItem.Type の場合)

仮引数名	名称	in/out	説明
name	列挙型定数の名前	in	返される列挙型定数の名前を指定します。

## 戻り値

指定された名前を持つ列挙型定数を返します。

## 例外

valueOf で発生する例外を次の表に示します。

表 7-74 valueOf で発生する例外 (CIWorkItem.Type の場合)

発生する例外	説明
java.lang.IllegalArgumentException	指定された名前を持つ定数をこの列挙型が持っていない場合

## values

列挙型 : CIWorkItem.Type

## 構文

```
public static CIWorkItem.Type[]  
    values()
```

## 機能

この列挙型の定数が含まれている配列を宣言されている順序で返します。

このメソッドは次のようにして定数を繰り返すために使用できます。

```
for(CIWorkItem.Type c :
    CIWorkItem.Type.values())
    System.out.println(c);
```

## 引数

ありません。

## 戻り値

この列挙型の定数が含まれている配列を宣言されている順序で返します。

## 例外

ありません。

# 8

## 例外クラス

この章では、パッケージ「`jp.co.Hitachi.soft.csciw`」に所属する API の、クラスおよびインタフェース使用時にスローされる例外クラスについて説明します。なお、例外クラスの階層については、「[3.3 パッケージ「`jp.co.Hitachi.soft.csciw`」の階層](#)」を参照してください。

## 8.1 記述形式

---

### 8.1.1 例外クラスの記述形式

例外クラスは、次に示す形式を基本として説明しています。

#### クラス定義

API のクラス定義を示します。

#### すべての実装されたインタフェース

API のすべての実装されたインタフェースを示します。

#### 直系の既知のサブクラス

API 直系の既知のサブクラスを示します。

#### メソッド

API のメソッドを説明します。

### 8.1.2 例外クラスが提供するメソッドの記述形式

例外クラスが提供するメソッドは、次に示す形式を基本として説明しています。

#### メソッド名

メソッド名を示します。

#### クラス名

メソッドが所属するクラス名を示します。

#### 構文

メソッドの構文を示します。

#### 機能

メソッドの機能を説明します。

#### オーバーライド

メソッドのオーバーライドを説明します。

#### 引数

メソッドの引数を説明します。

#### 戻り値

メソッドが返す戻り値を示します。

## 8.2 CIWEntityNotExistException (処理するオブジェクトが存在しない場合)

---

処理しようとしたオブジェクトが存在しない場合にスローされる例外クラスです。

オブジェクトを取得したあとに、ほかから操作されてワーク管理データベース上に存在しなくなったり、ID を指定するメソッドで不正な ID が指定されたりした場合に発生します。

### クラス定義

```
public final class CIWEntityNotExistException
    extends CIWException
```

### すべての実装されたインタフェース

```
java.io.Serializable
```

### メソッド

java.lang.Throwable クラスから継承されたメソッド

fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

java.lang.Object クラスから継承されたメソッド

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait



## 8.3 CIWException (例外の基底の抽象クラス)

CSCIW で発生する例外の基底となる抽象クラスです。

### クラス定義

```
public abstract class CIWException
    extends java.lang.Exception
```

### すべての実装されたインタフェース

```
java.io.Serializable
```

### 直系の既知のサブクラス

CIWException 直系の既知のサブクラスを次の表に示します。

表 8-1 CIWException 直系の既知のサブクラス

サブクラス名	説明
<a href="#">CIWEntityNotExistException</a>	処理しようとしたオブジェクトが存在しない場合にスローされる例外クラスです。
<a href="#">CIWFatalException</a>	処理を続行できないような障害が発生した場合にスローされる例外クラスです。
<a href="#">CIWStateException</a>	不正な状態に関する例外クラスです。
<a href="#">CIWTransientException</a>	一時的な要因で処理を実行できない場合にスローされる例外クラスです。
<a href="#">CIWTransitionFailedException</a>	案件処理中にエラーが発生した場合にスローされる例外クラスです。

### メソッド

#### java.lang.Throwable クラスから継承されたメソッド

fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

#### java.lang.Object クラスから継承されたメソッド

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

## 8.4 CIWFatalException (障害発生の場合)

処理を続行できないような障害が発生した場合にスローされる例外クラスです。

### クラス定義

```
public class CIWFatalException
    extends CIWException
```

### すべての実装されたインタフェース

```
java.io.Serializable
```

### 直系の既知のサブクラス

CIWFatalException 直系の既知のサブクラスを次の表に示します。

表 8-2 CIWFatalException 直系の既知のサブクラス

サブクラス名	説明
<a href="#">CIWSQLException</a>	DBMS で処理を続行できないような障害が発生した場合にスローされる例外クラスです。

### メソッド

java.lang.Throwable クラスから継承されたメソッド

fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

java.lang.Object クラスから継承されたメソッド

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

## 8.5 CIWSQLException (DBMS の障害発生の場合)

DBMS で処理を続行できないような障害が発生した場合にスローされる例外クラスです。

### クラス定義

```
public final class CIWSQLException
    extends CIWFatalException
```

### すべての実装されたインタフェース

```
java.io.Serializable
```

### メソッド

CIWSQLException のメソッドを次の表に示します。

表 8-3 CIWSQLException のメソッド

メソッド名	説明
<code>getCause</code>	原因の SQLException を取得します。
<code>getErrorCode</code>	ベンダー固有の例外コードを取得します。
<code>getSQLState</code>	SQLState 値を取得します。

#### java.lang.Throwable クラスから継承されたメソッド

`fillInStackTrace`, `getLocalizedMessage`, `getMessage`, `getStackTrace`, `initCause`, `printStackTrace`, `printStackTrace`, `printStackTrace`, `setStackTrace`, `toString`

#### java.lang.Object クラスから継承されたメソッド

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

## getCause

クラス名: CIWSQLException

### 構文

```
public java.sql.SQLException getCause()
```

### 機能

原因の SQLException を取得します。

### オーバーライド

java.lang.Throwable クラス内の getCause

## 引数

ありません。

## 戻り値

原因となる SQLException を返します。

## getErrorCode

クラス名：CIWSQLException

## 構文

```
public int getErrorCode()
```

## 機能

ベンダー固有の例外コードを取得します。

## 引数

ありません。

## 戻り値

ベンダー固有の例外コードを返します。

## getSQLState

クラス名：CIWSQLException

## 構文

```
public java.lang.String getSQLState()
```

## 機能

SQLState 値を取得します。

## 引数

ありません。

## 戻り値

SQLState 値を返します。

## 8.6 CIWSQLTransientException (DBMS の一時的な障害発生の場合)

DBMS の一時的な要因で処理を実行できない場合にスローされる例外クラスです。

セットアッププロパティファイルの設定項目の、次のどれかに SQL 例外となる SQLCODE または SQLSTATE の値が設定された場合に、この例外がスローされます。

- HiRDBTransientSQLCode
- HiRDBTransientSQLState
- OracleTransientSQLCode
- OracleTransientSQLState
- SQLServerTransientSQLCode
- SQLServerTransientSQLState
- PostgreSQLTransientSQLState

セットアッププロパティファイルについては、マニュアル「uCosminexus Service Coordinator Interactive Workflow システム構築・運用ガイド」の「環境設定ファイル」を参照してください。

### クラス定義

```
public final class CIWSQLTransientException
    extends CIWTransientException
```

### すべての実装されたインタフェース

```
java.io.Serializable
```

### メソッド

CIWSQLTransientException のメソッドを次の表に示します。

表 8-4 CIWSQLTransientException のメソッド

メソッド	機能概要
<a href="#">getCause</a>	原因の SQLException を取得します。
<a href="#">getErrorCode</a>	ベンダー固有の例外コードを取得します。
<a href="#">getSQLState</a>	SQLState 値を取得します。

### java.lang.Throwable クラスから継承されたメソッド

fillInStackTrace, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

### java.lang.Object クラスから継承されたメソッド

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

## getCause

クラス名：CIWSQLTransientException

### 構文

```
public java.sql.SQLException getCause()
```

### 機能

原因の SQLException を取得します。

### オーバーライド

java.lang.Throwable クラス内の getCause

### 引数

ありません。

### 戻り値

原因となる SQLException を返します。

## getErrorCode

クラス名：CIWSQLTransientException

### 構文

```
public int getErrorCode()
```

### 機能

ベンダー固有の例外コードを取得します。

### 引数

ありません。

### 戻り値

ベンダー固有の例外コードを返します。

# getSQLState

クラス名：CIWSQLTransientException

## 構文

```
public java.lang.String getSQLState()
```

## 機能

SQLState 値を取得します。

## 引数

ありません。

## 戻り値

SQLState 値を返します。



## 8.7 CIWStateException (不正な状態の場合)

---

不正な状態に関する例外クラスです。

### クラス定義

```
public final class CIWStateException
    extends CIWException
```

### すべての実装されたインタフェース

```
java.io.Serializable
```

### メソッド

java.lang.Throwable クラスから継承されたメソッド

fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

java.lang.Object クラスから継承されたメソッド

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

## 8.8 CIWTransientException (一時的な障害発生の場合)

一時的な要因で処理を実行できない場合にスローされる例外クラスです。

### クラス定義

```
public class CIWTransientException
    extends CIWException
```

### すべての実装されたインタフェース

```
java.io.Serializable
```

### 直系の既知のサブクラス

CIWTransientException 直系の既知のサブクラスを次の表に示します。

表 8-5 CIWTransientException 直系の既知のサブクラス

サブクラス名	説明
<a href="#">CIWSQLTransientException</a>	DBMS の一時的な要因で処理を実行できない場合にスローされる例外クラスです。

### メソッド

java.lang.Throwable クラスから継承されたメソッド

fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

java.lang.Object クラスから継承されたメソッド

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

## 8.9 CIWTransitionFailedException (案件処理中のエラーの場合)

---

案件処理中にエラーが発生した場合にスローされる例外クラスです。

### クラス定義

```
public final class CIWTransitionFailedException
    extends CIWException
```

### すべての実装されたインタフェース

```
java.io.Serializable
```

### メソッド

java.lang.Throwable クラスから継承されたメソッド

fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

java.lang.Object クラスから継承されたメソッド

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

# 9

## Java オブジェクト呼び出し用のインタフェース

この章では、Java オブジェクト呼び出し用のインタフェースについて説明します。

このマニュアルでは、条件、振り分けルール、生成ルール、および作業アプリケーションの Java オブジェクト呼び出し機能について説明します。

アプリケーション呼び出しサービスの Java オブジェクト呼び出し機能については、マニュアル「uCosminexus Service Coordinator Interactive Workflow BPMN 連携機能 使用の手引」を参照してください。

## 9.1 記述形式

---

Java オブジェクト呼び出し用のインタフェースの記述形式について説明します。

### 9.1.1 条件またはルールの評価・作業アプリケーション用クラスおよびインタフェースの記述形式

#### 入れ子のクラス

インタフェースの入れ子のクラスを説明します。

#### メソッド

メソッドを説明します。

### 9.1.2 条件またはルールの評価・作業アプリケーション用クラスおよびインタフェースが提供するメソッドの記述形式

条件またはルールの評価・作業アプリケーション用クラスおよびインタフェースが提供するメソッドは、次に示す形式を基本として説明しています。

#### メソッド名

メソッド名を示します。

#### クラス名またはインタフェース名

メソッドが所属するクラス名またはインタフェース名を示します。

#### 構文

メソッドの構文を示します。

#### 機能

メソッドの機能を説明します。

#### 引数

メソッドの引数を説明します。

#### 戻り値

メソッドが返す戻り値を示します。

#### 例外

メソッドに発生する例外を説明します。

#### 注意事項

メソッドの注意事項を説明します。

## 9.2 CIWConditionEvaluator (条件 Java オブジェクトのインタフェース)

条件 Java オブジェクトのインタフェースです。

### 入れ子のクラス

CIWConditionEvaluator の入れ子のクラスを次の表に示します。

表 9-1 CIWConditionEvaluator の入れ子のクラス

クラス名	入れ子のクラスの説明およびクラスの記述形式
CIWConditionEvaluator.Parameter	条件に渡されるパラメタの列挙型です。
	static class CIWConditionEvaluator.Parameter

### メソッド

CIWConditionEvaluator インタフェースのメソッドを次の表に示します。

表 9-2 CIWConditionEvaluator インタフェースのメソッド

メソッド名	説明
evaluateCondition	条件を評価し、評価結果を取得します。

## evaluateCondition

インタフェース名：CIWConditionEvaluator

### 構文

```
boolean evaluateCondition(java.util.Map<CIWConditionEvaluator.Parameter, java.lang.String> aParameters,  
                           java.sql.Connection aDBConnection)  
throws CIWUserException
```

### 機能

条件を評価し、評価結果を取得します。

### 引数

evaluateCondition の引数を次の表に示します。

表 9-3 evaluateCondition の引数

仮引数名	名称	IN/OUT	説明
aParameters	Java オブジェクトに渡されるパラメータ一覧	IN	Parameter 列挙型をキーに値を取得します。値が設定されていない場合、null が取得されます。 Java オブジェクトに渡されるパラメータ一覧については、「 <a href="#">2.6.4 業務プログラムに渡される情報</a> 」を参照してください。
aDBConnection	CSCIW に関連づけられたコネクション	IN	このコネクションを使用することで、CSCIW のシステムと同一のトランザクションでデータベースにアクセスできます。

## 戻り値

- 条件を満たしている場合は true を返してください。
- 条件を満たしていない場合は false を返してください。

## 例外

evaluateCondition でエラーを検出した場合は、CIWUserException をスローしてください。

## 注意事項

- このメソッド内では CSCIW の API を実行できません。
- 例外をスローした場合は、業務アプリケーションで例外 (CIWTransitionFailedException) が発生します。
- aDBConnection のコネクションを切断したり、トランザクションを終了したりしないでください。

## 9.3 CIWCreationRuleEvaluator (生成ルール Java オブジェクトのインタフェース)

生成ルール Java オブジェクトのインタフェースです。

### 入れ子のクラス

CIWCreationRuleEvaluator の入れ子のクラスを次の表に示します。

表 9-4 CIWCreationRuleEvaluator の入れ子のクラス

クラス名	入れ子のクラスの説明およびクラスの記述形式
CIWCreationRuleEvaluator.Parameter	生成ルールに渡されるパラメタの列挙型です。
	static class CIWCreationRuleEvaluator.Parameter

### メソッド

CIWCreationRuleEvaluator インタフェースのメソッドを次の表に示します。

表 9-5 CIWCreationRuleEvaluator インタフェースのメソッド

メソッド名	説明
evaluateCreationRule	生成ルールを評価し、子作業名を取得します。

## evaluateCreationRule

インタフェース名：CIWCreationRuleEvaluator

### 構文

```
java.util.Set<java.lang.String> evaluateCreationRule(java.util.Map<CIWCreationRuleEvaluator.  
Parameter, java.lang.String> aParameters,  
java.sql.Connection aDBConnection)  
throws CIWUserException
```

### 機能

生成ルールを評価し、子作業名を取得します。

### 引数

evaluateCreationRule の引数を次の表に示します。



表 9-6 evaluateCreationRule の引数

仮引数名	名称	IN/OUT	説明
aParameters	Java オブジェクトに渡されるパラメータ一覧	IN	Parameter 列挙型をキーに値を取得します。値が設定されていない場合、null が取得されます。 Java オブジェクトに渡されるパラメータ一覧については、「 <a href="#">2.6.4 業務プログラムに渡される情報</a> 」を参照してください。
aDBConnection	CSCIW に関連づけられたコネクション	IN	このコネクションを使用することで、CSCIW のシステムと同一のトランザクションでデータベースにアクセスできます。

## 戻り値

子作業の作業名 Set を返してください。

## 例外

evaluateCreationRule でエラーを検出した場合は、CIWUserException をスローしてください。

## 注意事項

- このメソッド内では CSCIW の API を実行できません。
- 戻り値に要素数 0 の Set を指定すると、子作業は生成されません。
- 次の場合は業務アプリケーションで例外 (CIWTransitionFailedException) が発生します。
  - 戻り値に null を指定した場合
  - 戻り値に 1 バイト以上、<SYSTEMID>\_WORK\_ITEM テーブルの Name カラムのバイト数未満の作業名を指定しなかった場合
  - 例外をスローした場合
- aDBConnection のコネクションを切断したり、トランザクションを終了したりしないでください。

## 9.4 CIWResourceSelector (振り分けルール Java オブジェクトのインタフェース)

振り分けルール Java オブジェクトのインタフェースです。

### 入れ子のクラス

CIWResourceSelector の入れ子のクラスを次の表に示します。

表 9-7 CIWResourceSelector の入れ子のクラス

クラス名	入れ子のクラスの説明およびクラスの記述形式
CIWResourceSelector.Parameter	振り分けルールに渡されるパラメタの列挙型です。
	static class CIWResourceSelector.Parameter

### メソッド

CIWResourceSelector インタフェースのメソッドを次の表に示します。

表 9-8 CIWResourceSelector インタフェースのメソッド

メソッド名	説明
getAvailableParticipant	振り分けルールを評価し、作業者 ID を取得します。

## getAvailableParticipant

インタフェース名：CIWResourceSelector

### 構文

```
java.lang.String getAvailableParticipant(java.util.Map<CIWResourceSelector.Parameter, java.lang.String> aParameters,  
                                         java.sql.Connection aDBConnection)  
throws CIWUserException
```

### 機能

振り分けルールを評価し、作業者 ID を取得します。

### 引数

getAvailableParticipant の引数を次の表に示します。

表 9-9 getAvailableParticipant の引数

仮引数名	名称	IN/OUT	説明
aParameters	Java オブジェクトに渡されるパラメータ一覧	IN	Parameter 列挙型をキーに値を取得します。値が設定されていない場合、null が取得されます。 Java オブジェクトに渡されるパラメータ一覧については、「 <a href="#">2.6.4 業務プログラムに渡される情報</a> 」を参照してください。
aDBConnection	CSCIW に関連づけられたコネクション	IN	このコネクションを使用することで、CSCIW のシステムと同一のトランザクションでデータベースにアクセスできます。

## 戻り値

作業者 ID を返してください。

## 例外

getAvailableParticipant でエラーを検出した場合は、CIWUserException をスローしてください。

## 注意事項

- このメソッド内では CSCIW の API を実行できません。
- 次の場合は業務アプリケーションで例外 (CIWTransitionFailedException) が発生します。
  - 戻り値に null を指定した場合
  - 戻り値に 1 バイト以上、<SYSTEMID>\_WORK\_ITEM テーブルの Participant カラムのバイト数未満の作業者 ID を指定しなかった場合
  - 例外をスローした場合
- aDBConnection のコネクションを切断したり、トランザクションを終了したりしないでください。

## 9.5 CIWorkApplication (作業アプリケーション Java オブジェクトのインタフェース)

作業アプリケーション Java オブジェクトのインタフェースです。

### 入れ子のクラス

CIWorkApplication の入れ子のクラスを次の表に示します。

表 9-10 CIWorkApplication の入れ子のクラス

クラス名	入れ子のクラスの説明およびクラスの記述形式
CIWorkApplication.Parameter	作業アプリケーションに渡されるパラメタの列挙型です。 static class CIWorkApplication.Parameter
CIWorkApplication.Result	作業アプリケーションの業務処理結果の列挙型です。 static class CIWorkApplication.Result

### メソッド

CIWorkApplication インタフェースのメソッドを次の表に示します。

表 9-11 CIWorkApplication インタフェースのメソッド

メソッド名	説明
startSynchronously	作業アプリケーションを同期処理で開始します。

## startSynchronously

インタフェース名：CIWorkApplication

### 構文

```
CIWorkApplication.Result startSynchronously(java.util.Map<CIWorkApplication.Parameter, java.lang.String> aParameters,
                                             java.sql.Connection aDBConnection)
    throws CIWUserException
```

### 機能

作業アプリケーションを同期処理で開始します。

### 引数

startSynchronously の引数を次の表に示します。

表 9-12 startSynchronously の引数

仮引数名	名称	IN/OUT	説明
aParameters	Java オブジェクトに渡されるパラメータ一覧	IN	Parameter 列挙型をキーに値を取得します。値が設定されていない場合、null が取得されます。 Java オブジェクトに渡されるパラメータ一覧については、「 <a href="#">2.6.4 業務プログラムに渡される情報</a> 」を参照してください。
aDBConnection	CSCIW に関連づけられたコネクション	IN	このコネクションを使用することで、CSCIW のシステムと同一のトランザクションでデータベースにアクセスできます。

## 戻り値

- 作業の状態を変更しない場合は Result.NO\_CHANGE\_STATE を返してください。
- 作業を着手する場合は Result.PERFORM を返してください。
- 作業を完了する場合は Result.PERFORM\_AND\_COMPLETE を返してください。

## 例外

startSynchronously でエラーを検出した場合は、CIWUserException をスローしてください。

## 注意事項

- このメソッド内では CSCIW の API を実行できません。
- 次の場合は業務アプリケーションで例外 (CIWTransitionFailedException) が発生します。
  - 戻り値に null を指定した場合
  - 例外をスローした場合
- aDBConnection のコネクションを切断したり、トランザクションを終了したりしないでください。

# 10

## 列挙型

この章では、パッケージ「jp.co.Hitachi.soft.csciw.callback」が提供する列挙型について説明します。

## 10.1 記述形式

---

### 10.1.1 列挙型の記述形式

列挙型は、次に示す形式を基本として説明しています。

#### 概要

列挙型の概要を説明します。

#### 定義

列挙型の定義を示します。

#### すべての実装されたインタフェース

列挙型のすべての実装されたインタフェースを示します。

#### 含まれているインタフェース

列挙型の含まれているインタフェースを示します。

#### 定数

列挙型の定数を説明します。

#### メソッド

列挙型のメソッドを説明します。

### 10.1.2 列挙型が提供するメソッドの記述形式

列挙型が提供するメソッドは、次に示す形式を基本として説明しています。

#### メソッド名

メソッド名を示します。

#### 列挙型

メソッドが所属する列挙型を示します。

#### 構文

メソッドの構文を示します。

#### 機能

メソッドの機能を説明します。

#### 引数

メソッドの引数を説明します。

#### 戻り値

メソッドが返す戻り値を示します。

## 例外

メソッドに発生する例外を説明します。

## 注意事項

メソッドの注意事項を説明します。



## 10.2 CIWConditionEvaluator.Parameter (条件に渡されるパラメタの列挙型)

### 概要

条件に渡されるパラメタの列挙型です。

### 定義

```
public static enum CIWConditionEvaluator.Parameter extends java.lang.Enum<CIWConditionEvaluator.Parameter>
```

### すべての実装されたインタフェース

```
java.io.Serializable, java.lang.Comparable<CIWConditionEvaluator.Parameter>
```

### 含まれているインタフェース

```
CIWConditionEvaluator
```

### 定数

CIWConditionEvaluator.Parameter の列挙型定数を次の表に示します。

表 10-1 CIWConditionEvaluator.Parameter の列挙型定数

定数名	説明および記述形式
CSCIW_ADEFNAME	業務ステップ定義名
	public static final CIWConditionEvaluator.Parameter CSCIW_ADEFNAME
CSCIW_AIDEADLINE	業務ステップ処理期限
	public static final CIWConditionEvaluator.Parameter CSCIW_AIDEADLINE
CSCIW_AINAME	業務ステップ名
	public static final CIWConditionEvaluator.Parameter CSCIW_AINAME
CSCIW_AIPRIORITY	業務ステップ優先度
	public static final CIWConditionEvaluator.Parameter CSCIW_AIPRIORITY
CSCIW_AISTARTDATE	業務ステップ開始日時
	public static final CIWConditionEvaluator.Parameter CSCIW_AISTARTDATE
CSCIW_CNDEFNAME	制御ノード定義名
	public static final CIWConditionEvaluator.Parameter CSCIW_CNDEFNAME
CSCIW_EXPARAMETER	補助定数パラメタ

定数名	説明および記述形式
	public static final CIWConditionEvaluator.Parameter CSCIW_EXPARAMETER
CSCIW_PDEFNAME	ビジネスプロセス定義名 public static final CIWConditionEvaluator.Parameter CSCIW_PDEFNAME
CSCIW_PDEFVERSION	ビジネスプロセス定義のバージョン public static final CIWConditionEvaluator.Parameter CSCIW_PDEFVERSION
CSCIW_PICREATOR	案件投入者 public static final CIWConditionEvaluator.Parameter CSCIW_PICREATOR
CSCIW_PIDEADLINE	案件処理期限 public static final CIWConditionEvaluator.Parameter CSCIW_PIDEADLINE
CSCIW_PIID	案件 ID public static final CIWConditionEvaluator.Parameter CSCIW_PIID
CSCIW_PINAME	案件名 public static final CIWConditionEvaluator.Parameter CSCIW_PINAME
CSCIW_PIPRIORITY	案件優先度 public static final CIWConditionEvaluator.Parameter CSCIW_PIPRIORITY
CSCIW_PISTARTDATE	案件開始日時 public static final CIWConditionEvaluator.Parameter CSCIW_PISTARTDATE
CSCIW_USERID	ユーザ識別子 public static final CIWConditionEvaluator.Parameter CSCIW_USERID
CSCIW_WDEFNAME	作業定義名 public static final CIWConditionEvaluator.Parameter CSCIW_WDEFNAME
CSCIW_WICREATIONDATE	作業生成日時 public static final CIWConditionEvaluator.Parameter CSCIW_WICREATIONDATE
CSCIW_WIDEADLINE	作業期限 public static final CIWConditionEvaluator.Parameter CSCIW_WIDEADLINE

定数名	説明および記述形式
CSCIW_WINAME	作業名
	public static final CIWConditionEvaluator.Parameter CSCIW_WINAME
CSCIW_WIPARTICIPANT	作業者
	public static final CIWConditionEvaluator.Parameter CSCIW_WIPARTICIPANT
CSCIW_WIPRIORITY	作業優先度
	public static final CIWConditionEvaluator.Parameter CSCIW_WIPRIORITY
CSCIW_WISTARTDATE	作業開始日時
	public static final CIWConditionEvaluator.Parameter CSCIW_WISTARTDATE

## メソッド

CIWConditionEvaluator.Parameter のメソッドを次の表に示します。

表 10-2 CIWConditionEvaluator.Parameter のメソッド

メソッド	説明
<a href="#">toParameterName</a>	パラメタ名を取得します。
<a href="#">valueOf</a>	指定した名前を持つこの型の列挙型定数を返します。
<a href="#">values</a>	この列挙型の定数を含む配列を宣言されている順序で返します。

java.lang.Enum クラスから継承されたメソッド

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

java.lang.Object クラスから継承されたメソッド

finalize, getClass, notify, notifyAll, wait, wait, wait

## toParameterName

列挙型：CIWConditionEvaluator.Parameter

### 構文

```
public final java.lang.String toParameterName()
```

### 機能

パラメタ名を取得します。

## 引数

ありません。

## 戻り値

パラメタ名を返します。

## 例外

ありません。

## valueOf

列挙型：CIWConditionEvaluator.Parameter

## 構文

```
public static CIWConditionEvaluator.Parameter valueOf(java.lang.String name)
```

## 機能

指定した名前を持つこの型の列挙型定数を返します。

文字列は、この型の列挙型定数を宣言するのに使用した識別子と完全に一致している必要があります。余分な空白文字を含めることはできません。

## 引数

valueOf の引数を次の表に示します。

表 10-3 valueOf の引数 (CIWConditionEvaluator.Parameter の場合)

仮引数名	名称	in/out	説明
name	列挙型定数の名前	in	返される列挙型定数の名前を指定します。

## 戻り値

指定された名前を持つ列挙型定数を返します。

## 例外

valueOf で発生する例外を次の表に示します。

表 10-4 valueOf で発生する例外 (CIWConditionEvaluator.Parameter の場合)

発生する例外	説明
java.lang.IllegalArgumentException	指定された名前を持つ定数をこの列挙型が持っていない場合

## values

列挙型 : CIWConditionEvaluator.Parameter

### 構文

```
public static final CIWConditionEvaluator.Parameter[] values()
```

### 機能

この列挙型の定数が含まれている配列を宣言されている順序で返します。

このメソッドは次のようにして定数を繰り返すために使用できます。

```
for(CIWConditionEvaluator.Parameter c : CIWConditionEvaluator.Parameter.values())  
    System.out.println(c);
```

### 引数

ありません。

### 戻り値

この列挙型の定数が含まれている配列を宣言されている順序で返します。

### 例外

ありません。

## 10.3 CIWCreationRuleEvaluator.Parameter (生成ルールに渡されるパラメタの列挙型)

### 概要

生成ルールに渡されるパラメタの列挙型です。

### 定義

```
public static enum CIWCreationRuleEvaluator.Parameter extends java.lang.Enum<CIWCreationRuleEvaluator.Parameter>
```

### すべての実装されたインタフェース

```
java.io.Serializable, java.lang.Comparable<CIWCreationRuleEvaluator.Parameter>
```

### 含まれているインタフェース

```
CIWCreationRuleEvaluator
```

### 定数

CIWCreationRuleEvaluator.Parameter の列挙型定数を次の表に示します。

表 10-5 CIWCreationRuleEvaluator.Parameter の列挙型定数

定数名	説明および記述形式
CSCIW_ADEFNAME	業務ステップ定義名
	public static final CIWCreationRuleEvaluator.Parameter CSCIW_ADEFNAME
CSCIW_AIDEADLINE	業務ステップ処理期限
	public static final CIWCreationRuleEvaluator.Parameter CSCIW_AIDEADLINE
CSCIW_AINAME	業務ステップ名
	public static final CIWCreationRuleEvaluator.Parameter CSCIW_AINAME
CSCIW_AIPRIORITY	業務ステップ優先度
	public static final CIWCreationRuleEvaluator.Parameter CSCIW_AIPRIORITY
CSCIW_AISTARTDATE	業務ステップ開始日時
	public static final CIWCreationRuleEvaluator.Parameter CSCIW_AISTARTDATE
CSCIW_EXPARAMETER	補助定数パラメタ
	public static final CIWCreationRuleEvaluator.Parameter CSCIW_EXPARAMETER

定数名	説明および記述形式
CSCIW_PDEFNAME	ビジネスプロセス定義名
	public static final CIWCreationRuleEvaluator.Parameter CSCIW_PDEFNAME
CSCIW_PDEFVERSION	ビジネスプロセス定義のバージョン
	public static final CIWCreationRuleEvaluator.Parameter CSCIW_PDEFVERSION
CSCIW_PICREATOR	案件投入者
	public static final CIWCreationRuleEvaluator.Parameter CSCIW_PICREATOR
CSCIW_PIDEADLINE	案件処理期限
	public static final CIWCreationRuleEvaluator.Parameter CSCIW_PIDEADLINE
CSCIW_PIID	案件 ID
	public static final CIWCreationRuleEvaluator.Parameter CSCIW_PIID
CSCIW_PINAME	案件名
	public static final CIWCreationRuleEvaluator.Parameter CSCIW_PINAME
CSCIW_PIPRIORITY	案件優先度
	public static final CIWCreationRuleEvaluator.Parameter CSCIW_PIPRIORITY
CSCIW_PISTARTDATE	案件開始日時
	public static final CIWCreationRuleEvaluator.Parameter CSCIW_PISTARTDATE
CSCIW_USERID	ユーザ識別子
	public static final CIWCreationRuleEvaluator.Parameter CSCIW_USERID
CSCIW_WDEFNAME	作業定義名
	public static final CIWCreationRuleEvaluator.Parameter CSCIW_WDEFNAME
CSCIW_WICREATIONDATE	作業生成日時
	public static final CIWCreationRuleEvaluator.Parameter CSCIW_WICREATIONDATE
CSCIW_WIDEADLINE	作業期限
	public static final CIWCreationRuleEvaluator.Parameter CSCIW_WIDEADLINE

定数名	説明および記述形式
CSCIW_WINAME	作業名
	public static final CIWCreationRuleEvaluator.Parameter CSCIW_WINAME
CSCIW_WIPARTICIPANT	作業者
	public static final CIWCreationRuleEvaluator.Parameter CSCIW_WIPARTICIPANT
CSCIW_WIPRIORITY	作業優先度
	public static final CIWCreationRuleEvaluator.Parameter CSCIW_WIPRIORITY
CSCIW_WISTARTDATE	作業開始日時
	public static final CIWCreationRuleEvaluator.Parameter CSCIW_WISTARTDATE

## メソッド

CIWCreationRuleEvaluator.Parameter のメソッドを次の表に示します。

表 10-6 CIWCreationRuleEvaluator.Parameter のメソッド

メソッド	説明
toParameterName	パラメタ名を取得します。
valueOf	指定した名前を持つこの型の列挙型定数を返します。
values	この列挙型の定数を含む配列を宣言されている順序で返します。

java.lang.Enum クラスから継承されたメソッド

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

java.lang.Object クラスから継承されたメソッド

finalize, getClass, notify, notifyAll, wait, wait, wait

## toParameterName

列挙型：CIWCreationRuleEvaluator.Parameter

## 構文

```
public final java.lang.String toParameterName()
```

## 機能

パラメタ名を取得します。



## 引数

ありません。

## 戻り値

パラメタ名を返します。

## 例外

ありません。

## valueOf

列挙型：CIWCreationRuleEvaluator.Parameter

## 構文

```
public static CIWCreationRuleEvaluator.Parameter valueOf(java.lang.String name)
```

## 機能

指定した名前を持つこの型の列挙型定数を返します。

文字列は、この型の列挙型定数を宣言するのに使用した識別子と完全に一致している必要があります。余分な空白文字を含めることはできません。

## 引数

valueOf の引数を次の表に示します。

表 10-7 valueOf の引数 (CIWCreationRuleEvaluator.Parameter の場合)

仮引数名	名称	in/out	説明
name	列挙型定数の名前	in	返される列挙型定数の名前を指定します。

## 戻り値

指定された名前を持つ列挙型定数を返します。

## 例外

valueOf で発生する例外を次の表に示します。

表 10-8 valueOf で発生する例外 (CIWCreationRuleEvaluator.Parameter の場合)

発生する例外	説明
java.lang.IllegalArgumentException	指定された名前を持つ定数をこの列挙型が持っていない場合

## values

列挙型 : CIWCreationRuleEvaluator.Parameter

## 構文

```
public static final CIWCreationRuleEvaluator.Parameter[] values()
```

## 機能

この列挙型の定数が含まれている配列を宣言されている順序で返します。

このメソッドは次のようにして定数を繰り返すために使用できます。

```
for(CIWCreationRuleEvaluator.Parameter c : CIWCreationRuleEvaluator.Parameter.values())  
    System.out.println(c);
```

## 引数

ありません。

## 戻り値

この列挙型の定数が含まれている配列を宣言されている順序で返します。

## 例外

ありません。

## 10.4 CIWResourceSelector.Parameter (振り分けルールに渡されるパラメタの列挙型)

### 概要

振り分けルールに渡されるパラメタの列挙型です。

### 定義

```
public static enum CIWResourceSelector.Parameter extends java.lang.Enum<CIWResourceSelector.Parameter>
```

### すべての実装されたインタフェース

```
java.io.Serializable, java.lang.Comparable<CIWResourceSelector.Parameter>
```

### 含まれているインタフェース

```
CIWResourceSelector
```

### 定数

CIWResourceSelector.Parameter の列挙型定数を次の表に示します。

表 10-9 CIWResourceSelector.Parameter の列挙型定数

定数	説明および記述形式
CSCIW_ADEFNAME	業務ステップ定義名 public static final CIWResourceSelector.Parameter CSCIW_ADEFNAME
CSCIW_AIDEADLINE	業務ステップ処理期限 public static final CIWResourceSelector.Parameter CSCIW_AIDEADLINE
CSCIW_AINAME	業務ステップ名 public static final CIWResourceSelector.Parameter CSCIW_AINAME
CSCIW_AIPRIORITY	業務ステップ優先度 public static final CIWResourceSelector.Parameter CSCIW_AIPRIORITY
CSCIW_AISTARTDATE	業務ステップ開始日時 public static final CIWResourceSelector.Parameter CSCIW_AISTARTDATE
CSCIW_EXPARAMETER	補助定数パラメタ public static final CIWResourceSelector.Parameter CSCIW_EXPARAMETER
CSCIW_PDEFNAME	ビジネスプロセス定義名

定数	説明および記述形式
	public static final CIWResourceSelector.Parameter CSCIW_PDEFNAME
CSCIW_PDEFVERSION	ビジネスプロセス定義のバージョン public static final CIWResourceSelector.Parameter CSCIW_PDEFVERSION
CSCIW_PICREATOR	案件投入者 public static final CIWResourceSelector.Parameter CSCIW_PICREATOR
CSCIW_PIDEADLINE	案件処理期限 public static final CIWResourceSelector.Parameter CSCIW_PIDEADLINE
CSCIW_PIID	案件 ID public static final CIWResourceSelector.Parameter CSCIW_PIID
CSCIW_PINAME	案件名 public static final CIWResourceSelector.Parameter CSCIW_PINAME
CSCIW_PIPRIORITY	案件優先度 public static final CIWResourceSelector.Parameter CSCIW_PIPRIORITY
CSCIW_PISTARTDATE	案件開始日時 public static final CIWResourceSelector.Parameter CSCIW_PISTARTDATE
CSCIW_USERID	ユーザ識別子 public static final CIWResourceSelector.Parameter CSCIW_USERID
CSCIW_WDEFNAME	作業定義名 public static final CIWResourceSelector.Parameter CSCIW_WDEFNAME
CSCIW_WICREATIONDATE	作業生成日時 public static final CIWResourceSelector.Parameter CSCIW_WICREATIONDATE
CSCIW_WIDEADLINE	作業期限 public static final CIWResourceSelector.Parameter CSCIW_WIDEADLINE
CSCIW_WINAME	作業名 public static final CIWResourceSelector.Parameter CSCIW_WINAME
CSCIW_WIPARTICIPANT	作業者

定数	説明および記述形式
	public static final CIWResourceSelector.Parameter CSCIW_WIPARTICIPANT
CSCIW_WISTARTDATE	作業開始日時
	public static final CIWResourceSelector.Parameter CSCIW_WISTARTDATE

## メソッド

CIWResourceSelector.Parameter のメソッドを次の表に示します。

表 10-10 CIWResourceSelector.Parameter のメソッド

メソッド	説明
toParameterName	パラメタ名を取得します。
valueOf	指定した名前を持つこの型の列挙型定数を返します。
values	この列挙型の定数を含む配列を宣言されている順序で返します。

### java.lang.Enum クラスから継承されたメソッド

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

### java.lang.Object クラスから継承されたメソッド

finalize, getClass, notify, notifyAll, wait, wait, wait

## toParameterName

列挙型：CIWResourceSelector.Parameter

### 構文

```
public final java.lang.String toParameterName()
```

### 機能

パラメタ名を取得します。

### 引数

ありません。

### 戻り値

パラメタ名を返します。

## 例外

ありません。

## valueOf

列挙型：CIWResourceSelector.Parameter

## 構文

```
public static CIWResourceSelector.Parameter valueOf(java.lang.String name)
```

## 機能

指定した名前を持つこの型の列挙型定数を返します。

文字列は、この型の列挙型定数を宣言するのに使用した識別子と完全に一致している必要があります。余分な空白文字を含めることはできません。

## 引数

valueOf の引数を次の表に示します。

表 10-11 valueOf の引数 (CIWResourceSelector.Parameter の場合)

仮引数名	名称	in/out	説明
name	列挙型定数の名前	in	返される列挙型定数の名前を指定します。

## 戻り値

指定された名前を持つ列挙型定数を返します。

## 例外

valueOf で発生する例外を次の表に示します。

表 10-12 valueOf で発生する例外 (CIWResourceSelector.Parameter の場合)

発生する例外	説明
java.lang.IllegalArgumentException	指定された名前を持つ定数をこの列挙型が持っていない場合

## values

列挙型：CIWResourceSelector.Parameter

## 構文

```
public static final CIWResourceSelector.Parameter[] values()
```

## 機能

この列挙型の定数が含まれている配列を宣言されている順序で返します。

このメソッドは次のようにして定数を繰り返すために使用できます。

```
for(CIWResourceSelector.Parameter c : CIWResourceSelector.Parameter.values())  
    System.out.println(c);
```

## 引数

ありません。

## 戻り値

この列挙型の定数が含まれている配列を宣言されている順序で返します。

## 例外

ありません。

## 10.5 CIWorkApplication.Parameter (作業アプリケーションに渡されるパラメタの列挙型)

### 概要

作業アプリケーションに渡されるパラメタの列挙型です。

### 定義

```
public static enum CIWorkApplication.Parameter extends java.lang.Enum<CIWorkApplication.Parameter>
```

### すべての実装されたインタフェース

```
java.io.Serializable, java.lang.Comparable<CIWorkApplication.Parameter>
```

### 含まれているインタフェース

```
CIWorkApplication
```

### 定数

CIWorkApplication.Parameter の列挙型定数を次の表に示します。

表 10-13 CIWorkApplication.Parameter の列挙型定数

定数名	説明および記述形式
CSCIW_ADEFNAME	業務ステップ定義名
	public static final CIWorkApplication.Parameter CSCIW_ADEFNAME
CSCIW_AIDEADLINE	業務ステップ処理期限
	public static final CIWorkApplication.Parameter CSCIW_AIDEADLINE
CSCIW_AINAME	業務ステップ名
	public static final CIWorkApplication.Parameter CSCIW_AINAME
CSCIW_AIPRIORITY	業務ステップ優先度
	public static final CIWorkApplication.Parameter CSCIW_AIPRIORITY
CSCIW_AISTARTDATE	業務ステップ開始日時
	public static final CIWorkApplication.Parameter CSCIW_AISTARTDATE
CSCIW_EXPARAMETER	補助定数パラメタ
	public static final CIWorkApplication.Parameter CSCIW_EXPARAMETER
CSCIW_PDEFNAME	ビジネスプロセス定義名



定数名	説明および記述形式
	public static final CIWorkApplication.Parameter CSCIW_PDEFNAME
CSCIW_PDEFVERSION	ビジネスプロセス定義のバージョン public static final CIWorkApplication.Parameter CSCIW_PDEFVERSION
CSCIW_PICREATOR	案件投入者 public static final CIWorkApplication.Parameter CSCIW_PICREATOR
CSCIW_PIDEADLINE	案件処理期限 public static final CIWorkApplication.Parameter CSCIW_PIDEADLINE
CSCIW_PIID	案件 ID public static final CIWorkApplication.Parameter CSCIW_PIID
CSCIW_PINAME	案件名 public static final CIWorkApplication.Parameter CSCIW_PINAME
CSCIW_PIPRIORITY	案件優先度 public static final CIWorkApplication.Parameter CSCIW_PIPRIORITY
CSCIW_PISTARTDATE	案件開始日時 public static final CIWorkApplication.Parameter CSCIW_PISTARTDATE
CSCIW_SYSTEMID	システム ID public static final CIWorkApplication.Parameter CSCIW_SYSTEMID
CSCIW_USERID	ユーザ識別子 public static final CIWorkApplication.Parameter CSCIW_USERID
CSCIW_WDEFNAME	作業定義名 public static final CIWorkApplication.Parameter CSCIW_WDEFNAME
CSCIW_WICREATIONDATE	作業生成日時 public static final CIWorkApplication.Parameter CSCIW_WICREATIONDATE
CSCIW_WIDEADLINE	作業処理期限 public static final CIWorkApplication.Parameter CSCIW_WIDEADLINE
CSCIW_WINAME	作業名

定数名	説明および記述形式
	public static final CIWorkApplication.Parameter CSCIW_WINAME
CSCIW_WIPARTICIPANT	作業者
	public static final CIWorkApplication.Parameter CSCIW_WIPARTICIPANT
CSCIW_WIPRIORITY	作業優先度
	public static final CIWorkApplication.Parameter CSCIW_WIPRIORITY
CSCIW_WISTARTDATE	作業開始日時
	public static final CIWorkApplication.Parameter CSCIW_WISTARTDATE

## メソッド

CIWorkApplication.Parameter のメソッドを次の表に示します。

表 10-14 CIWorkApplication.Parameter のメソッド

メソッド	説明
<a href="#">toParameterName</a>	パラメタ名を取得します。
<a href="#">valueOf</a>	指定した名前を持つこの型の列挙型定数を返します。
<a href="#">values</a>	この列挙型の定数を含む配列を宣言されている順序で返します。

### java.lang.Enum クラスから継承されたメソッド

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

### java.lang.Object クラスから継承されたメソッド

finalize, getClass, notify, notifyAll, wait, wait, wait

## toParameterName

列挙型：CIWorkApplication.Parameter

## 構文

```
public final java.lang.String toParameterName()
```

## 機能

パラメタ名を取得します。

## 引数

ありません。

## 戻り値

パラメタ名を返します。

## 例外

ありません。

## valueOf

列挙型：CIWWWorkApplication.Parameter

## 構文

```
public static CIWWWorkApplication.Parameter valueOf(java.lang.String name)
```

## 機能

指定した名前を持つこの型の列挙型定数を返します。

文字列は、この型の列挙型定数を宣言するのに使用した識別子と完全に一致している必要があります。余分な空白文字を含めることはできません。

## 引数

valueOf の引数を次の表に示します。

表 10-15 valueOf の引数 (CIWWWorkApplication.Parameter の場合)

仮引数名	名称	in/out	説明
name	列挙型定数の名前	in	返される列挙型定数の名前を指定します。

## 戻り値

指定された名前を持つ列挙型定数を返します。

## 例外

valueOf で発生する例外を次の表に示します。

表 10-16 valueOf で発生する例外 (CIWorkApplication.Parameter の場合)

発生する例外	説明
java.lang.IllegalArgumentException	指定された名前を持つ定数をこの列挙型が持っていない場合

## values

列挙型 : CIWorkApplication.Parameter

### 構文

```
public static final CIWorkApplication.Parameter[] values()
```

### 機能

この列挙型の定数が含まれている配列を宣言されている順序で返します。

このメソッドは次のようにして定数を繰り返すために使用できます。

```
for(CIWorkApplication.Parameter c : CIWorkApplication.Parameter.values())  
    System.out.println(c);
```

### 引数

ありません。

### 戻り値

この列挙型の定数が含まれている配列を宣言されている順序で返します。

### 例外

ありません。

## 10.6 CIWorkApplication.Result (作業アプリケーションの業務処理結果の列挙型)

### 概要

作業アプリケーションの業務処理結果の列挙型です。

### 定義

```
public static enum CIWorkApplication.Result extends java.lang.Enum<CIWorkApplication.Result>
```

### すべての実装されたインタフェース

```
java.io.Serializable, java.lang.Comparable<CIWorkApplication.Result>
```

### 含まれているインタフェース

```
CIWorkApplication
```

### 定数

CIWorkApplication.Result の列挙型定数を次の表に示します。

表 10-17 CIWorkApplication.Result の列挙型定数

定数名	説明および記述形式
NO_CHANGE_STATE	作業の状態を変更しない。 <pre>public static final CIWorkApplication.Result NO_CHANGE_STATE</pre>
PERFORM	作業を着手する。 <pre>public static final CIWorkApplication.Result PERFORM</pre>
PERFORM_AND_COMPLETE	作業を完了する。 <pre>public static final CIWorkApplication.Result PERFORM_AND_COMPLETE</pre>

### メソッド

CIWorkApplication.Result のメソッドを次の表に示します。

表 10-18 CIWorkApplication.Result のメソッド

メソッド	説明
<code>valueOf</code>	指定した名前を持つこの型の列挙型定数を返します。
<code>values</code>	この列挙型の定数を含む配列を宣言されている順序で返します。

### java.lang.Enum クラスから継承されたメソッド

`clone`, `compareTo`, `equals`, `getDeclaringClass`, `hashCode`, `name`, `ordinal`, `toString`, `valueOf`

java.lang.Object クラスから継承されたメソッド  
finalize, getClass, notify, notifyAll, wait, wait, wait

## valueOf

列挙型 : CIWorkApplication.Result

### 構文

```
public static CIWorkApplication.Result valueOf(java.lang.String name)
```

### 機能

指定した名前を持つこの型の列挙型定数を返します。

文字列は、この型の列挙型定数を宣言するのに使用した識別子と完全に一致している必要があります。余分な空白文字を含めることはできません。

### 引数

valueOf の引数を次の表に示します。

表 10-19 valueOf の引数 (CIWorkApplication.Result の場合)

仮引数名	名称	in/out	説明
name	列挙型定数の名前	in	返される列挙型定数の名前を指定します。

### 戻り値

指定された名前を持つ列挙型定数を返します。

### 例外

valueOf で発生する例外を次の表に示します。

表 10-20 valueOf で発生する例外 (CIWorkApplication.Result の場合)

発生する例外	説明
java.lang.IllegalArgumentException	指定された名前を持つ定数をこの列挙型が持っていない場合

## values

列挙型 : CIWorkApplication.Result

## 構文

```
public static final CIWorkApplication.Result[] values()
```

## 機能

この列挙型の定数が含まれている配列を宣言されている順序で返します。

このメソッドは次のようにして定数を繰り返すために使用できます。

```
for(CIWorkApplication.Result c :  
    CIWorkApplication.Result.values())  
    System.out.println(c);
```

## 引数

ありません。

## 戻り値

この列挙型の定数が含まれている配列を宣言されている順序で返します。

## 例外

ありません。

# 11

## 例外クラス

この章では、パッケージ「`jp.co.Hitachi.soft.csciw.callback`」に所属する API の、クラスおよびインタフェース使用時にスローされる例外クラスについて説明します。なお、例外クラスの階層については、「[3.4 パッケージ「`jp.co.Hitachi.soft.csciw.callback`」の階層](#)」を参照してください。



## 11.1 記述形式

---

### 11.1.1 例外クラスの記述形式

例外クラスは、次に示す形式を基本として説明しています。

#### クラス定義

API のクラス定義を示します。

#### すべての実装されたインタフェース

API のすべての実装されたインタフェースを示します。

#### メソッド

API のメソッドを説明します。

### 11.1.2 例外クラスが提供するメソッドの記述形式

例外クラスが提供するメソッドは、次に示す形式を基本として説明しています。

#### メソッド名

メソッド名を示します。

#### クラス名

メソッドが所属するクラス名を示します。

#### 構文

メソッドの構文を示します。

#### 機能

メソッドの機能を説明します。

#### 引数

メソッドの引数を説明します。

#### 戻り値

メソッドが返す戻り値を示します。

### 11.1.3 例外クラスが提供するコンストラクタの記述形式

例外クラスが提供するコンストラクタは、次に示す形式を基本として説明しています。

#### コンストラクタ名

コンストラクタ名を示します。

## 構文

コンストラクタの構文を示します。

## 機能

コンストラクタの機能を説明します。

## 引数

コンストラクタの引数を説明します。

## 11.2 CIWUserException (Java オブジェクト呼び出し時にエラーが発生した場合)

---

Java オブジェクト呼び出し時の例外クラスです。

### クラス定義

```
public final class CIWUserException extends java.lang.Exception
```

すべての実装されたインタフェース

```
java.io.Serializable
```

### メソッド

java.lang.Throwable クラスから継承されたメソッド

fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

java.lang.Object クラスから継承されたメソッド

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

### コンストラクタ

指定されたユーザエラーメッセージを使用して、例外を作成するコンストラクタ

```
CIWUserException(java.lang.String aUserErrorMessage)
```

指定されたユーザエラーメッセージおよびエラー原因を使用して、例外を作成するコンストラクタ

```
CIWUserException(java.lang.String aUserErrorMessage, java.lang.Throwable initCause)
```

## CIWUserException (ユーザエラーメッセージを使用する場合)

### 構文

```
public CIWUserException(java.lang.String aUserErrorMessage)
```

### 機能

指定されたユーザエラーメッセージを使用して、Java オブジェクト呼び出し時の例外を作成します。

### 引数

CIWUserException の引数を次の表に示します。

表 11-1 CIWUserException の引数 (ユーザエラーメッセージを使用する場合)

仮引数名	名称	in/out	説明
aUserErrorMessage	ユーザエラーメッセージ	in	<p>ユーザエラーメッセージを指定します。 指定したエラーメッセージは次の CSCIW のメッセージに出力されます。</p> <ul style="list-style-type: none"> <li>条件, 生成ルール, および振り分けルールの場合 KDIW02502-E</li> <li>作業アプリケーションの場合 KDIW05301-E</li> </ul> <p>CSCIW で出力されるメッセージについては, マニュアル「uCosminexus Service Coordinator Interactive Workflow メッセージ」を参照してください。</p>

## CIWUserException (ユーザエラーメッセージおよびエラー原因を使用する場合)

### 構文

```
public CIWUserException(java.lang.String aUserErrorMessage,
                        java.lang.Throwable initCause)
```

### 機能

指定されたユーザエラーメッセージおよびエラー原因を使用して, Java オブジェクト呼び出し時の例外を作成します。

### 引数

CIWUserException の引数を次の表に示します。

表 11-2 CIWUserException の引数 (ユーザエラーメッセージおよびエラー原因を使用する場合)

仮引数名	名称	in/out	説明
aUserErrorMessage	ユーザエラーメッセージ	in	<p>ユーザエラーメッセージを指定します。 指定したエラーメッセージは次の CSCIW のメッセージに出力されます。</p> <ul style="list-style-type: none"> <li>条件, 生成ルール, および振り分けルールの場合 KDIW02502-E</li> <li>作業アプリケーションの場合 KDIW05301-E</li> </ul>

仮引数名	名称	in/out	説明
			CSCIW で出力されるメッセージについては、マニュアル「uCosminexus Service Coordinator Interactive Workflow メッセージ」を参照してください。
initCause	エラー原因	in	エラー原因を指定します。

# 付録

## 付録 A 更新系 API と参照系 API

この節では、CSCIW で使用する更新系 API と参照系 API について説明します。

### 付録 A.1 更新系 API

更新系 API の一覧を次の表に示します。

表 A-1 更新系 API 一覧

クラス名	API 名	説明
CIWServer	createProcessInstance	案件を生成する
	createAndStartProcessInstance	案件を生成して開始する
	allocateWorkItemEx	条件に一致する作業に作業者を割り当てて着手する
CIWProcessInstance	start	案件を開始する
	changeState	案件の状態を変更する
	terminate	案件を強制終了する
	delete	案件を削除する
	setDeadline	案件の処理期限を設定する
	setPriority	案件の優先度を設定する
	adhocInstantiateActivityInstance	案件の業務ステップを動的に生成する
	startAndAdhocInstantiateActivityInstanceByADName	案件の業務ステップ（業務ステップ定義名を指定）を動的に生成し、案件を開始する
CIWActivityInstance	changeState	業務ステップの状態を変更する
	evaluate	業務ステップの条件を評価する

クラス名	API 名	説明
	makeBackwardTransition	業務ステップを差し戻す (引き戻す)
	setDeadline	業務ステップの処理期限を設定する
	setPriority	業務ステップの優先度を設定する
	adhocMakeTransition	業務ステップ間のフロー遷移を実行する
CIWWWorkItem	changeState	作業の状態を変更する
	perform	作業を開始する
	complete	作業を完了する
	performAndComplete	作業を開始して完了する
	reassign	作業の作業者を変更する
	reassignAndPerform	作業の作業者を変更して開始する
	evaluate	作業の条件を評価する
	reevaluateCastingRule	作業の振り分けルールを再評価する
	setDeadline	作業の処理期限を設定する
	setPriority	作業の優先度を設定する
	free	作業を返却して振り分けルールを再評価する

## 付録 A.2 参照系 API

参照系 API の一覧を次の表に示します。



表 A-2 参照系 API 一覧

クラス名	API 名	説明
CIWServer	getProcessInstancesList	指定したフィルター条件を満たした案件リストを取得する
	getProcessInstance	指定した ID の案件を取得する
	getActivityInstancesList	指定したフィルター条件を満たした業務ステップリストを取得する
	getActivityInstance	指定した ID の業務ステップを取得する
	getWorkItemsList	指定したフィルター条件を満たした作業リストを取得する
	getWorkItem	指定した ID の作業を取得する
	getProcessDefinitionsList	指定したフィルター条件を満たしたビジネスプロセス定義リストを取得する
	getProcessDefinition	指定した ID のビジネスプロセス定義を取得する
	getActivityDefinitionsList	指定したフィルター条件を満たす業務ステップ定義リストを取得する
	getActivityDefinition	指定した ID の業務ステップ定義を取得する
	getWorkDefinitionsList	指定したフィルター条件を満たす作業定義リストを取得する
	getWorkDefinition	指定した ID の作業定義を取得する
	getListCount	指定したフィルター条件を満たす件数を取得する
CIWProcessInstance	getProcessDefinition	対象案件のビジネスプロセス定義を取得する
	getWorkItemsList	対象案件に含まれる作業の中で指定したフィルター条件を満たす作業リストを取得する
CIWorkItem	getParentProcessInstance	対象作業が所属する案件を取得する
	getWorkDefinition	対象作業の作業定義を取得する
CIWProcessDefinition	getProcessInstancesList	対象ビジネスプロセス定義を使用している指定したフィルター条件を満たした案件リストを取得する

## 付録 B 参照系 API で指定できる属性

この節では、参照系 API で指定できる属性の範囲について説明します。

### 付録 B.1 指定できる属性一覧

参照系 API の属性リスト、フィルター条件、およびソート条件に指定できる属性の一覧を次に示します。

#### (1) ビジネスプロセス定義属性

CIWProcessDefinition インタフェースの AttributeName 列挙型で定義されています。

ビジネスプロセス定義属性を次の表に示します。

表 B-1 ビジネスプロセス定義属性一覧

属性名	列挙値	説明	参照 ※1	条件 ※2	インデクス※3
ID	ID	ビジネスプロセス定義 ID	○	○	A (U)
Name	NAME	ビジネスプロセス定義名	○	○	B-1 (U)
Version	VERSION	ビジネスプロセス定義のバージョン	○	○	B-2 (U)
StateCode	STATE	ビジネスプロセス定義の状態	○	○	—
Description	DESCRIPTION	ビジネスプロセス定義の説明	○	○	—
Author	AUTHOR	ビジネスプロセス定義の作成者（上書き時は更新者）	○	○	—
CreationDate	CREATION_DATE	ビジネスプロセス定義の作成日（上書き時は更新日）	○	○	—
Responsible	RESPONSIBLE	ビジネスプロセス定義の管理者 ID	○	○	—
ValidFromDate	VALID_FROM_DATE	ビジネスプロセス定義が有効となる日（案件投入可能開始日）	○	○	—
ValidToDate	VALID_TO_DATE	ビジネスプロセス定義が無効となる日（案件投入可能終了日）	○	○	—

(凡例)

○：指定できます。

—：定義されているインデクスがありません。

注※1

属性名セットに指定できるかどうかを示します。

注※2

フィルター条件およびソート条件に指定できるかどうかを示します。

注※3

定義されているインデクスを示します。

英字が各インデクスを示し、「-」に続く数字は、インデクスの構成順序を示します。(U) はユニークインデクスを示します。

## (2) 業務ステップ定義属性

CIWActivityDefinition インタフェースの AttributeName 列挙型で定義されています。

業務ステップ定義属性を次の表に示します。

表 B-2 業務ステップ定義属性一覧

属性名	列挙値	説明	参照※ 1	条件 ※2	インデクス※3
ID	ID	業務ステップ定義 ID	○	○	A-2 (U), C (U)
Name	NAME	業務ステップ定義名	○	○	B
ProcessDefinitionID	PROCESS_DEFINITION_ID	ビジネスプロセス定義 ID	○	○	A-1 (U)
Description	DESCRIPTION	説明	○	○	-

(凡例)

○：指定できます。

-：定義されているインデクスがありません。

注※1

属性名セットに指定できるかどうかを示します。

注※2

フィルター条件およびソート条件に指定できるかどうかを示します。

注※3

定義されているインデクスを示します。

英字が各インデクスを示し、「-」に続く数字は、インデクスの構成順序を示します。(U) はユニークインデクスを示します。

## (3) 作業定義属性

CIWorkDefinition インタフェースの AttributeName 列挙型で定義されています。

作業定義属性を次の表に示します。

表 B-3 作業定義属性一覧

属性名	列挙値	説明	参照 ※1	条件 ※2	インデクス※3
ID	ID	作業定義の ID	○	○	A-2 (U), C (U)

属性名	列挙値	説明	参照 ※1	条件 ※2	インデクス※3
Name	NAME	作業定義の名称	○	○	B
ActivityDefinitionID	ACTIVITY_DEFINITION_ID	作業定義が所属する業務ステップ定義の ID	○	○	—
ProcessDefinitionID	PROCESS_DEFINITION_ID	作業定義が所属するビジネスプロセス定義の ID	○	○	A-1 (U)
WorkTypeCode	TYPE	作業の種類	○	○	—
CastingRuleName	CASTING_RULE_NAME	振り分けルール名	○	○	—
Description	DESCRIPTION	作業の説明	○	○	—

(凡例)

○：指定できます。

—：定義されているインデクスがありません。

注※1

属性名セットに指定できるかどうかを示します。

注※2

フィルター条件およびソート条件に指定できるかどうかを示します。

注※3

定義されているインデクスを示します。

英字が各インデクスを示し、「-」に続く数字は、インデクスの構成順序を示します。(U) はユニークインデクスを示します。

## (4) 案件属性

CIWProcessInstance インタフェースの AttributeName 列挙型で定義されています。

案件属性を次の表に示します。

表 B-4 案件属性一覧

属性名	列挙値	説明	参照 ※1	条件 ※2	インデクス※3
ID	ID	案件の ID	○	○	A (U), C-2, D-2
Name	NAME	案件の名称 (案件キー)	○	○	B
ProcessDefinitionName	PROCESS_DEFINITION_NAME	案件が所属するビジネスプロセス定義の名称	○	○	—
ProcessDefinitionID	PROCESS_DEFINITION_ID	案件が所属するビジネスプロセス定義の ID	○	○	C-1
StateCode	STATE	案件の状態	○	○	D-1

属性名	列挙値	説明	参照 ※1	条件 ※2	インデクス※3
Creator	CREATOR	案件の投入者	○	○	—
StartDate	START_DATE	案件の開始日時	○	○	—
ClosedDate	CLOSED_DATE	案件の終了日時	○	○	—
Deadline	DEADLINE	案件の処理期限	○	○	—
Priority	PRIORITY	案件の優先度	○	○	—
MovedDate	MOVED_DATE	案件の乗せ替え日時	○	○	—

(凡例)

○：指定できます。

—：定義されているインデクスがありません。

注※1

属性名セットに指定できるかどうかを示します。

注※2

フィルター条件およびソート条件に指定できるかどうかを示します。

注※3

定義されているインデクスを示します。

英字が各インデクスを示し、「-」に続く数字は、インデクスの構成順序を示します。(U) はユニークインデクスを示します。

## (5) 業務ステップ属性

CIWActivityInstance インタフェースの AttributeName 列挙型で定義されています。

業務ステップ属性を次の表に示します。

表 B-5 業務ステップ属性一覧

属性名	列挙値	説明	参照 ※1	条件 ※2	インデクス※3
ID	ID	業務ステップの ID	○	○	A-2 (U), B (U)
Name	NAME	業務ステップの名称 (業務ステップキー)	○	○	—
ActivityDefinitionName	ACTIVITY_DEFINITION_NAME	業務ステップ定義の名称	○	○	—
ProcessInstanceID	PROCESS_INSTANCE_ID	業務ステップが所属する案件の ID	○	○	A-1 (U)
ActivityDefinitionID	ACTIVITY_DEFINITION_ID	業務ステップ定義の ID	○	○	—

属性名	列挙値	説明	参照 ※1	条件 ※2	インデクス※3
ProcessDefinitionID	PROCESS_DEFINITION_ID	業務ステップが所属する ビジネスプロセス定義の ID	○	○	—
ActivityTypeCode	TYPE	業務ステップの種類	○	○	—
StateCode	STATE	業務ステップの状態	○	○	—
StartDate	START_DATE	業務ステップの開始日時	○	○	—
ClosedDate	CLOSED_DATE	業務ステップの終了日時	○	○	—
Deadline	DEADLINE	業務ステップの処理期限	○	○	—
ProcessInstanceName	PROCESS_INSTANCE_NAME	案件名（案件キー）	○	○	—
Priority	PRIORITY	業務ステップの優先度	○	○	—

(凡例)

○：指定できます。

—：定義されているインデクスがありません。

注※1

属性名セットに指定できるかどうかを示します。

注※2

フィルター条件およびソート条件に指定できるかどうかを示します。

注※3

定義されているインデクスを示します。

英字が各インデクスを示し、「-」に続く数字は、インデクスの構成順序を示します。(U) はユニークインデクスを示します。

## (6) 作業属性

CIWorkItem インタフェースの AttributeName 列挙型で定義されています。

作業属性を次の表に示します。

表 B-6 作業属性一覧

属性名	列挙値	説明	参照※ 1	条件 ※2	インデクス※3
ID	ID	作業の ID	○	○	A-2 (U), B-2, C (U)
Name	NAME	作業の名称（作業 キー）	○	○	—
WorkDefinitionName	WORK_DEFINITION_NAME	作業定義の名称	○	○	—

属性名	列挙値	説明	参照※ 1	条件 ※2	インデクス※3
ActivityInstanceID	ACTIVITY_INSTANCE_ID	作業が所属する業務ステップの ID	○	○	—
ProcessInstanceID	PROCESS_INSTANCE_ID	作業が所属する案件の ID	○	○	A-1 (U)
WorkDefinitionID	WORK_DEFINITION_ID	作業定義の ID	○	○	—
ProcessDefinitionID	PROCESS_DEFINITION_ID	作業が所属するビジネスプロセス定義の ID	○	○	—
WorkTypeCode	TYPE	作業の種類	○	○	—
StateCode	STATE	作業の状態	○	○	—
Priority	PRIORITY	作業の優先度	○	○	—
Participant	PARTICIPANT	作業者の ID	○	○	B-1
CreationDate	CREATION_DATE	作業の発生日時	○	○	—
StartDate	START_DATE	作業の開始日時	○	○	—
ClosedDate	CLOSED_DATE	作業の終了日時	○	○	—
Deadline	DEADLINE	作業の処理期限の絶対日時	○	○	—
ProcessInstanceName	PROCESS_INSTANCE_NAME	案件名 (案件キー)	○	○	—

(凡例)

○：指定できます。

—：定義されているインデクスがありません。

注※1

属性名セットに指定できるかどうかを示します。

注※2

フィルター条件およびソート条件に指定できるかどうかを示します。

注※3

定義されているインデクスを示します。

英字が各インデクスを示し、「-」に続く数字は、インデクスの構成順序を示します。(U) はユニークインデクスを示します。

## 付録 B.2 指定できる属性値

State 属性および Type 属性で指定できる値を次に示します。

### (1) State 属性で指定できる値

クラスごとの State 列挙型で定義されています。

指定できる値は、State 列挙型ごとに定義された定数です。

State 属性で指定できる値（参照時には属性値として設定される値）を次の表に示します。

表 B-7 State 属性で指定できる値

コード値	意味	State 列挙型での完全表記形式	PD	PI	AI	WI	CR
a	非活性	inactive	○	—	—	—	○
b	活性	active	○	—	—	—	○
c	未終了	open	—	—	—	—	—
d	実行中	open.running	—	○	○	—	—
e	自動実行	open.running.executing	—	—	—	○	—
f	作業実行	open.running.performing	—	—	—	○	—
g	停止中	open.notRunning	—	—	—	—	—
h	未実行	open.notRunning.notStarted	—	○	—	—	—
i	初期	open.notRunning.notStarted.initial	—	—	○	○	—
j	実行開始可能	open.notRunning.notStarted.ready	—	—	○	○	—
k	中断	open.notRunning.suspended	—	—	—	—	—
l	実行開始不可	open.notRunning.suspended.disabled	—	—	○	○	—
m	実行停止	open.notRunning.suspended.intermitted	—	○	○	○	—
n	終了	closed	—	—	—	—	—
o	完了	closed.completed	—	○	—	—	—
p	実行省略	closed.completed.notExecuted	—	—	○	○	—
q	実行取消	closed.completed.canceled	—	—	—	○	—
r	実行済	closed.completed.executed	—	—	—	○	—
s	遷移可	closed.completed.executed.readyForTransition	—	—	○	—	—
t	遷移済	closed.completed.executed.transitionCompleted	—	—	○	—	—
u	強制終了	closed.terminated	—	○	○	○	—
v*	異常終了	closed.aborted	—	○	○	○	—
z	未定義	undefined	○	○	○	○	○

(凡例)

PD：ビジネスプロセス定義属性の StateCode 属性での指定の可否を示します。

PI：案件属性の StateCode 属性での指定の可否を示します。

AI：業務ステップ属性の StateCode 属性での指定の可否を示します。



- WI：作業属性の StateCode 属性での指定の可否を示します。
- CR：振り分けルール定義属性の StateCode 属性での指定の可否を示します。
- ：指定できます。
- －：指定できません。

注※

CSCIW ではサポートしていません。

## (2) Type 属性で指定できる値

クラスごとの Type 列挙型で定義されています。

指定できる値は、Type 列挙型ごとに定義された定数です。

Type 列挙型ごとの ActivityType または WorkType 値、および ActivityTypeCode または WorkTypeCode 値（コード表記形式）を次に示します。

表 B-8 Type 属性で指定できる値

コード値	意味	Type 列挙型での完全表記形式	AI	WD	WI
0	一般作業 または 業務ステップ	normal	○	○	○
g*	エラー回復作業 または エラー回復業務ステップ	errorNotice	○	－	○
a*	並列業務ステップ作業	builtin.concurrentActivity	－	○	○
b	並列作業	builtin.concurrentWork	－	○	○
c*	計時作業	builtin.timer	－	○	○
d*	先手通知作業	builtin.followingActivity	－	○	○
z	未定義	undefined	○	○	○

(凡例)

- AI：業務ステップ属性の ActivityTypeCode 属性での指定の可否を示します。
- WD：作業定義属性の WorkTypeCode 属性での指定の可否を示します。
- WI：作業属性の WorkTypeCode 属性での指定の可否を示します。
- ：指定できます。
- －：指定できません。

注※

CSCIW ではサポートしていません。

## 付録 C 状態遷移モデル

---

ワーク管理システムのオブジェクトごとに、状態遷移の動作モデルについて次に示す構成で説明します。

### 状態遷移図

ワーク管理システムのオブジェクトの状態と遷移の概観を図で示します。

### 状態

状態ごとの説明とオブジェクト間の状態の制約事項について説明します。

### 遷移

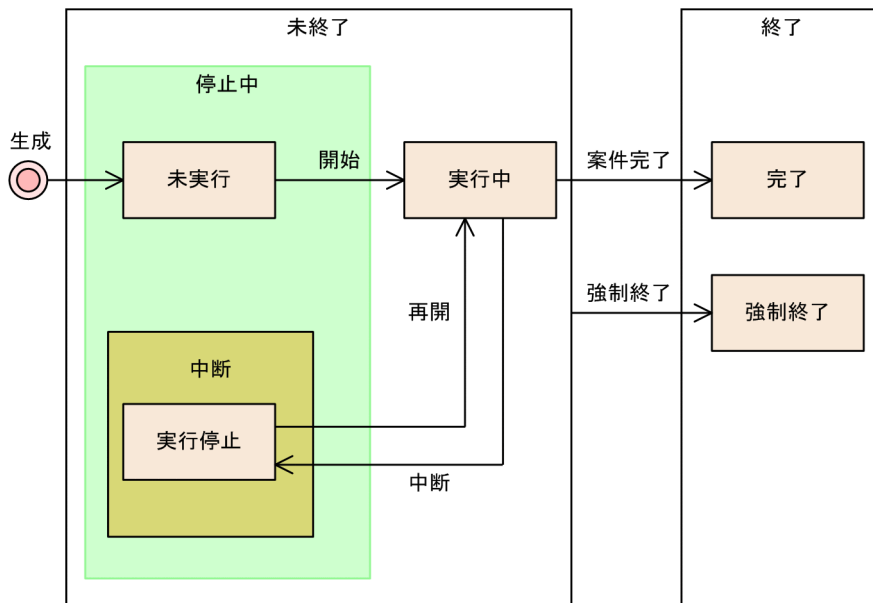
遷移ごとに、遷移を引き起こす要因と遷移時または遷移後に実行されるワーク管理システムの動作について説明します。

## 付録 C.1 案件の状態遷移

### (1) 案件の状態遷移

案件の状態遷移を次の図に示します。

図 C-1 案件の状態遷移



(凡例)

: 案件の状態を示します。  
 —→ : 案件の遷移を示します。

注

大きい枠で示す状態は、小さい枠で示す状態を含んでいます。例えば、「中断」状態は「実行停止」状態を含んでおり、また、「中断」状態は「停止中」状態に含まれています。「未終了」状態は、「未実行」状態、「実行中」状態、「実行停止」状態、「停止中」状態および「中断」状態を含んでいます。

## (2) 案件の状態

案件の状態について次の表に示します。

表 C-1 案件の状態一覧

上段：状態 中段：列挙型定数（コード値） 下段：状態の短縮表記形式	状態の完全表記形式	説明
未実行 NOT_STARTED(h) notStarted	open.notRunning.notStarted	案件が生成されたときの初期の状態を示します。
実行中 RUNNING(d) running	open.running	案件が実行されている状態を示します。「実行中」状態の場合だけ、案件に含まれる業務ステップを「実行中」状態にできます。
実行停止 INTERMITTED(m) intermitted	open.notRunning.suspended.intermitted	案件の実行が中断された一時停止の状態を示します。

上段：状態 中段：列挙型定数（コード値） 下段：状態の短縮表記形式	状態の完全表記形式	説明
		<p>「実行停止」状態の案件に含まれる業務ステップは開始できません。</p> <p>案件が「実行停止」状態になると、業務ステップは「実行開始不可」状態へ遷移します。ただし、すでに実行されている業務ステップはそのまま実行されます。このとき、業務ステップが完了しても、案件は「完了」状態には遷移しないで、「実行停止」状態のままです。</p>
完了 COMPLETED(o) completed	closed.completed	<p>案件が完了した状態を示します。</p> <p>「完了」状態の案件に含まれているすべての業務ステップは、「完了」状態または「強制終了」状態です。</p>
強制終了 TERMINATED(u) terminated	closed.terminated	<p>案件が強制的に終了された状態を示します。</p> <p>「強制終了」状態の案件に含まれているすべての業務ステップは、「完了」状態または「強制終了」状態です。</p>

### (3) 案件の遷移

案件の遷移について次の表に示します。

表 C-2 案件の遷移一覧

遷移種別	要因 および 動作	説明
生成	要因	API 呼び出しによる生成要求
	動作	案件を生成し、そのあと「未実行」状態へ遷移させます。
開始 (start)	要因	API 呼び出しによる遷移要求
	動作	<p>「実行中」状態へ遷移し、そのあとフロー定義でソースノードから到達できる業務ステップを活性化します。この場合、対象となる業務ステップのインスタンスが存在しないときは、新規に業務ステップを生成します。</p> <p>また、事前生成を指定している業務ステップも、このタイミングで生成します。</p>
中断 (suspend)	要因	API 呼び出しによる遷移要求
	動作	案件を「実行停止」状態へ遷移させます。
再開 (resume)	要因	API 呼び出しによる遷移要求

遷移種別	要因 および 動作	説明
	動作	案件を「実行中」状態へ遷移させます。 案件を「実行中」状態へ遷移させると同時に、「実行開始不可」状態のすべての業務ステップも「実行中」状態へ遷移させます。
案件完了 (process completed)	要因	フロー遷移のシンクノードへの到達 (シンクノードへの到達は、ワーク管理システムによるフロー制御で実行)
	動作	案件を「完了」状態へ遷移させます。 案件を「完了」状態へ遷移させると同時に次に示す処理を行います。 <ul style="list-style-type: none"> <li>「初期」状態のすべての業務ステップを「実行省略」状態に遷移。</li> <li>「初期」状態を除く、すべての「未終了」状態の業務ステップを「強制終了」状態に遷移。</li> </ul>
強制終了 (terminate)	要因	API 呼び出しによる遷移要求
	動作	案件を「強制終了」状態に遷移させます。 案件を「強制終了」状態に遷移させると同時に、「終了」状態以外のすべての業務ステップを「強制終了」状態に遷移させます。

案件の遷移時に設定する属性について次の表に示します。

表 C-3 案件の遷移時に設定する属性一覧

属性名	生成	開始	中断	再開	案件完了	強制終了
ID	○	—	—	—	—	—
Name <sup>※1</sup>	○	—	—	—	—	—
ProcessDefinitionName	○	—	—	—	—	—
ProcessDefinitionID	○	—	—	—	—	—
StateCode <sup>※2</sup>	○	○	○	○	○	○
Creator <sup>※3</sup>	○	—	—	—	—	—
StartDate <sup>※4</sup>	N	○	—	—	—	—
ClosedDate <sup>※4</sup>	N	—	—	—	○	○
Deadline <sup>※5</sup>	○	△	—	—	—	—
Priority <sup>※1</sup>	○	—	—	—	—	—
MovedDate	N	—	—	—	—	—

(凡例)

- ：設定します。
- ：設定しません。
- △：条件によって設定します。

N : null を設定します。

注※1

API で指定された値を設定します。指定されない場合は null を設定します。

注※2

遷移先の状態を設定します。

注※3

案件の生成または投入を行った場合のユーザ記述子を設定します。

注※4

遷移が行われた時刻を設定します。

注※5

案件の生成時は API で指定された値を設定します。指定されない場合は null を設定します。  
また、案件の開始 (start) 時は次に示す条件で設定します。

**生成時に設定した値が null 以外の場合**

開始 (start) 時は値を設定しません。生成時の値のままです。

**生成時に設定した値が null の場合**

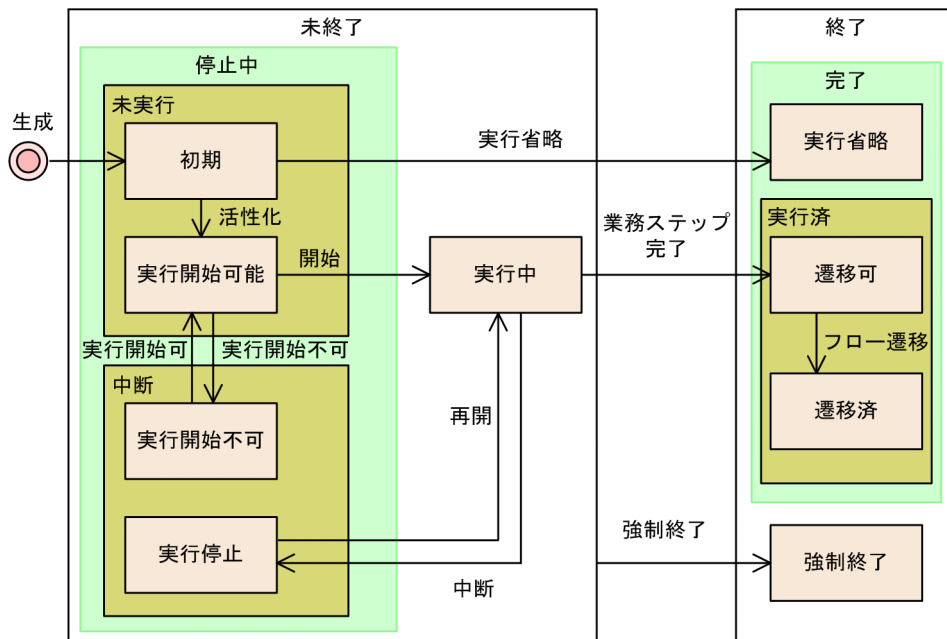
- 処理期限ルールが定義されていない場合  
開始 (start) 時は値を設定しません。生成時の値のままです。
- 処理期限ルールが定義されている場合  
処理期限ルールの評価結果を設定します。

## 付録 C.2 業務ステップの状態遷移

### (1) 業務ステップの状態遷移

業務ステップの状態遷移を次の図に示します。

図 C-2 業務ステップの状態遷移



(凡例)

□ : 業務ステップの状態を示します。

→ : 業務ステップの遷移を示します。

注

大きい枠で示す状態は、小さい枠で示す状態を含んでいます。  
 例えば、「中断」状態は「実行開始不可」状態および「実行停止」状態を含んでおり、また、「中断」状態は「停止中」状態に含まれています。「未終了」状態は、「未実行」状態、「実行中」状態、「実行停止」状態、「停止中」状態、「中断」状態などを含んでいます。

## (2) 業務ステップの状態

業務ステップの状態について次の表に示します。

表 C-4 業務ステップの状態一覧

上段：状態 中段：列挙型定数（コード値） 下段：状態の短縮表記形式	状態の完全表記形式	説明
初期 INITIAL(i) initial	open.notRunning.notStarted.initial	業務ステップが生成されたときの初期の状態を示します。
実行開始可能 READY(j) ready	open.notRunning.notStarted.ready	業務ステップとして実行できる状態ですが、まだ実行開始されていない一時的な状態を示します。
実行中 RUNNING(d) running	open.running	業務ステップが実行されている状態を示します。

上段：状態 中段：列挙型定数（コード値） 下段：状態の短縮表記形式	状態の完全表記形式	説明
		<p>「実行中」状態の場合だけ、業務ステップに含まれる作業を「実行中」状態にできます。</p> <p>また、「実行中」状態の場合だけ、業務ステップの評価が行われます。</p>
実行開始不可 DISABLED(l) disabled	open.notRunning.suspended.disabled	<p>案件が「実行停止」状態のため、業務ステップを実行できない状態を示します。</p> <p>案件が「実行停止」状態から再開されたときに、業務ステップも「実行開始可能」状態になります。</p>
実行停止 INTERMITTED(m) intermitted	open.notRunning.suspended.intermitted	<p>業務ステップの実行が中断された一時停止の状態を示します。</p> <p>「実行停止」状態の場合、業務ステップは完了できません。</p> <p>また、「実行停止」状態の業務ステップに含まれる作業は開始できません。ただし、すでに実行されている作業はそのまま実行されます。</p> <p>なお、「実行停止」状態の場合、業務ステップの評価は行われません。</p>
遷移可 READY_FOR_TRANSITION(s) readyForTransition	closed.completed.executed.readyForTransition	<p>業務ステップが完了し、次の業務ステップへ遷移できる状態を示します。</p> <p>業務ステップが「遷移可」状態へ遷移した場合は、対象となる業務ステップに含まれているすべての作業は「終了」状態です。</p>
遷移済 TRANSITION_COMPLETED(t) transitionCompleted	closed.completed.executed.transitionCompleted	<p>業務ステップが次の業務ステップへフロー遷移した状態を示します。</p>
実行省略 NOT_EXECUTED(p) notExecuted	closed.completed.notExecuted	<p>業務ステップが何も実行されないで完了した状態を示します。</p> <p>案件が遷移していく過程で、業務ステップを通らないで案件が完了した場合に、この業務ステップは「実行省略」状態になります。</p>
強制終了 TERMINATED(u) terminated	closed.terminated	<p>業務ステップが強制的に終了された状態を示します。</p> <p>業務ステップが「強制終了」状態へ遷移した場合は、対象となる業務ステップに含まれているすべての作業は「完了」状態または「強制終了」状態です。</p>



上段：状態 中段：列挙型定数（コード値） 下段：状態の短縮表記形式	状態の完全表記形式	説明
		<p>なお、強制終了された業務ステップは、次の業務ステップへ遷移されません。</p> <p>このとき、ほかに「実行中」状態の業務ステップが存在しない場合は、案件を強制終了する以外の操作ができなくなるため、注意してください。</p> <p>また、実行中の業務ステップが存在している場合でも、待合ノードに続く業務ステップを強制終了すると、案件は待合ノードから遷移されなくなるため、注意してください。</p>

### (3) 業務ステップの遷移

業務ステップの遷移について次の表に示します。

表 C-5 業務ステップの遷移一覧

遷移種別	要因 および 動作	説明
生成	要因	<ul style="list-style-type: none"> <li>フロー制御（未生成業務ステップへのフロー遷移）</li> <li>事前生成が設定された業務ステップの場合、案件の「実行中」状態への遷移時</li> <li>API 呼び出しによる生成要求</li> </ul>
	動作	業務ステップを「初期」状態で生成します。
活性化 (activate)	要因	<ul style="list-style-type: none"> <li>API 呼び出しによる活性化要求</li> <li>フロー制御（業務ステップへのフロー遷移）</li> </ul>
	動作	<p>業務ステップを「実行開始可能」状態へ遷移させます。そのあと、続けて次に示す状態遷移を行います。</p> <ul style="list-style-type: none"> <li>案件が「実行中」状態の場合は、「実行中」状態へ遷移。</li> <li>案件が「実行中」状態以外の場合は、「実行開始不可」状態へ遷移。</li> </ul>
開始 (start)	要因	業務ステップが「実行開始可能」状態へ遷移したときに、案件が「実行中」状態の場合
	動作	業務ステップを「実行中」状態へ遷移させ、そのあと業務ステップの評価処理を行います。
中断 (suspend)	要因	API 呼び出しによる遷移要求
	動作	<p>業務ステップを「実行停止」状態へ遷移させます。</p> <p>業務ステップを「実行停止」状態へ遷移させると同時に、対象となる業務ステップに含まれるすべての「未実行」状態の作業を「実行開始不可」状態へ遷移させます。</p>
再開 (resume)	要因	API 呼び出しによる遷移要求
	動作	業務ステップを「実行中」状態へ遷移させ、そのあと業務ステップの評価処理を行います。

遷移種別	要因 および 動作	説明
		業務ステップを「実行中」状態へ遷移させると同時に、対象となる業務ステップに含まれるすべての「実行開始不可」状態の作業を「未実行」状態へ遷移させます。
業務ステップ完了 (activity completed)	要因	<ul style="list-style-type: none"> <li>API 呼び出しによる「実行済」状態への遷移要求</li> <li>業務ステップの評価処理で、完了要件が満たされた場合</li> </ul> 例えば、業務ステップに含まれるすべての作業が「完了」状態へ遷移した場合や業務ステップの完了条件が存在する場合に条件が満たされたときなどです。
	動作	業務ステップを「遷移可」状態へ遷移させます。 業務ステップを「遷移可」状態へ遷移させると同時に、対象となる業務ステップに含まれる作業に対して次に示す処理を行います。 <ul style="list-style-type: none"> <li>すべての「未実行」状態の作業を「実行省略」状態へ遷移。</li> <li>すべての「実行中」状態の作業を「実行取消」状態へ遷移。</li> <li>すべての「実行停止」状態の作業を「強制終了」状態へ遷移。</li> </ul>
実行開始可 (enable)	要因	案件の「再開」状態への遷移
	動作	業務ステップを「実行開始可能」状態へ遷移させ、そのあと続けて「実行中」状態へ遷移させます。
実行開始不可 (disable)	要因	業務ステップが「実行開始可能」状態へ遷移したときに、案件が「実行停止」状態の場合
	動作	業務ステップを「実行開始不可」状態へ遷移させます。
フロー遷移 (transit)	要因	次の業務ステップへのフロー遷移処理時。 ただし、次のどちらかの場合はフロー遷移しません。 <ul style="list-style-type: none"> <li>遷移先がない</li> <li>遷移先に「未終了」状態、かつ「初期」状態でない業務ステップが存在する（遷移先の業務ステップ間に制御ノードがある場合を除く）</li> </ul>
	動作	業務ステップを「遷移済」状態へ遷移させます。
実行省略 (skip)	要因	案件の「完了」状態への遷移時（「案件完了」）、および案件の完了時（シンクノードへの到達など）に、業務ステップが「初期」状態の場合
	動作	業務ステップを「実行省略」状態へ遷移させます。
強制終了 (terminate)	要因	<ul style="list-style-type: none"> <li>API 呼び出しによる遷移要求</li> <li>案件の状態遷移に伴う制約</li> </ul> 案件の状態遷移に伴う制約については、「表 C-2 案件の遷移一覧」の「案件完了」状態遷移および「強制終了」状態遷移を参照してください。
	動作	業務ステップを「強制終了」状態に遷移させます。 業務ステップを「強制終了」状態に遷移させると同時に、対象となる業務ステップに含まれる「終了」状態以外のすべての作業を強制終了させます。

業務ステップの遷移時に設定する属性について次の表に示します。

表 C-6 業務ステップの遷移時に設定する属性一覧

属性名	生成	活性化	開始	中断	再開	業務ステップ完了	実行開始可	実行開始不可	フロー遷移	実行省略	強制終了
ID	○	—	—	—	—	—	—	—	—	—	—
Name※1	○	—	—	—	—	—	—	—	—	—	—
ActivityDefinitionName	○	—	—	—	—	—	—	—	—	—	—
ProcessInstanceID	○	—	—	—	—	—	—	—	—	—	—
ActivityDefinitionID	○	—	—	—	—	—	—	—	—	—	—
ProcessDefinitionID	○	—	—	—	—	—	—	—	—	—	—
ActivityTypeCode	○	—	—	—	—	—	—	—	—	—	—
StateCode※2	○	○	○	○	○	○	○	○	○	○	○
StartDate※3	N	—	○	—	—	—	—	—	—	—	—
ClosedDate※3	N	—	—	—	—	○	—	—	—	○	○
Deadline※4	N	—	○	—	—	—	—	—	—	—	—
ProcessInstanceName	○	—	—	—	—	—	—	—	—	—	—
Priority※5	○	—	—	—	—	—	—	—	—	—	—

(凡例)

- ：設定します。
- ：設定しません。
- N：null を設定します。

注※1

子業務ステップの場合は、生成ルールが返した値を設定します。  
子業務ステップ以外の場合は、null を設定します。

注※2

遷移先の状態を設定します。

注※3

遷移が行われた時刻を設定します。

注※4

開始 (start) 時は次の条件で設定します。

**処理期限ルールが定義されていない場合**

値を設定しません。生成時の値のままです。

## 処理期限ルールが定義されている場合

処理期限ルールの評価結果を設定します。

注※5

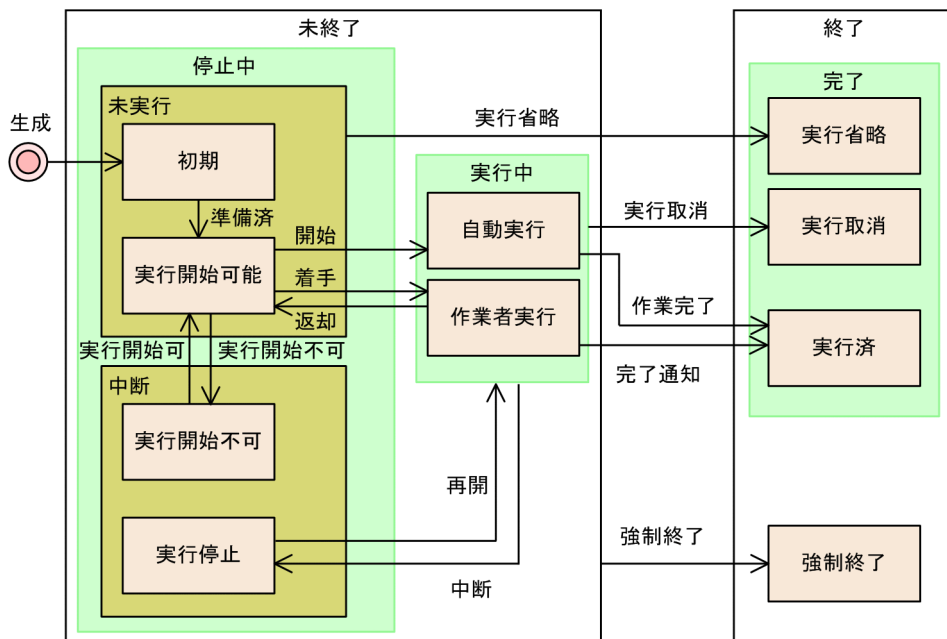
Priority には案件の Priority の値を設定します。

## 付録 C.3 作業の状態遷移

### (1) 作業の状態遷移

作業の状態遷移を次の図に示します。

図 C-3 作業の状態遷移



(凡例)

□ : 作業の状態を示します。

→ : 作業の遷移を示します。

注

大きい枠で示す状態は、小さい枠で示す状態を含んでいます。  
例えば、「中断」状態は「実行開始不可」状態および「実行停止」状態を含んでおり、また、「中断」状態は「停止中」状態に含まれています。「未終了」状態は、「未実行」状態、「実行中」状態、「実行停止」状態、「停止中」状態、「中断」状態などを含んでいます。

### (2) 作業の状態

作業の状態について次の表に示します。

表 C-7 作業の状態一覧

上段：状態 中段：列挙型定数（コード値） 下段：状態の短縮表記形式	状態の完全表記形式	説明
初期 INITIAL(i) initial	open.notRunning.notStarted.in initial	作業が生成されたときの初期の状態を示します。
実行開始可能 READY(j) ready	open.notRunning.notStarted.re ady	<p>作業が実行できる状態を示します。</p> <p>作業が「実行開始可能」状態へ遷移した時点で、作業への割り当て（リソースセレクション）は実行済みです。このため、作業からは実行開始可能な作業として見えます。</p> <p>なお、「実行開始可能」状態の場合、作業の評価および作業振り分けルールの再評価ができません。</p> <p>作業が「作業実行」状態となるタイミングは次のとおりです。</p> <ul style="list-style-type: none"> <li>• 作業が生成されたあとに、引き続いて遷移されたとき（ワーク管理システムによって作業が生成されたとき）</li> <li>• 作業（外部リソース）によって、「作業の返却と振り分けルールの再評価」の処理が実行されたとき</li> </ul>
自動実行 EXECUTING(e) executing	open.running.executing	<p>作業が自動的に実行されている状態を示します。</p> <p>組み込み作業など、ワーク管理システムが制御する作業が実行されていることを示します。</p> <p>なお、「自動実行」状態の場合、作業の評価ができません。</p>
作業実行 PERFORMING(f) performing	open.running.performing	<p>作業が作業を実行している状態を示します。</p> <p>人的リソースによる作業の実行など、ワーク管理システムの外部リソースによって作業が実行されていることを示します。</p> <p>なお、「作業実行」状態の場合、作業の評価および作業振り分けルールの再評価ができません。</p> <p>作業が「作業実行」状態となるタイミングは次のとおりです。</p> <ul style="list-style-type: none"> <li>• 作業（外部リソース）によって作業が実行されたとき</li> <li>• 作業アプリケーション（同期）によって作業が実行されたとき</li> <li>• 作業（外部リソース）によって、「条件に一致する作業の作業割り当てと着手」の処理が実行されたとき</li> </ul>

上段：状態 中段：列挙型定数（コード値） 下段：状態の短縮表記形式	状態の完全表記形式	説明
実行省略 NOT_EXECUTED(p) notExecuted	closed.completed.notExecuted	作業が何も実行されないで完了した状態を示します。 「未実行」状態の作業を含む業務ステップが「完了」状態になった場合、この作業は「実行省略」状態になります。 また、発生と同時に作業の完了条件が満たされた場合、作業は「実行省略」状態になります。
実行取消 CANCELED(q) canceled	closed.completed.canceled	作業の実行が取り消された状態を示します。 「実行中」状態の作業を含む業務ステップが「完了」状態になった場合に、この作業は「実行取消」状態になります。
実行済 EXECUTED(r) executed	closed.completed.executed	ワーク管理システムまたは作業によって、作業が実行され、完了した状態を示します。 「自動実行」状態の作業の場合、作業の完了条件が満たされると「実行済」状態になります。 また、「作業者実行」状態の作業の場合、「実行済」状態となるタイミングは次のとおりです。 <ul style="list-style-type: none"> <li>作業者（外部リソース）によって作業が完了されたときに</li> <li>作業アプリケーション（同期）によって作業が完了されたとき</li> </ul>
実行開始不可 DISABLED(l) disabled	open.notRunning.suspended.disabled	業務ステップが「実行停止」状態のため、作業を開始または着手できない状態を示します。 業務ステップが「実行停止」状態へ遷移した場合に、業務ステップに含まれるすべての「未実行」状態の作業は「実行開始不可」状態へ遷移します。 なお、作業が「実行開始不可」状態の場合は、作業振り分けルールの再評価ができます。
実行停止 INTERMITTED(m) intermitted	open.notRunning.suspended.intermitted	作業の実行が中断された一時停止の状態を示します。 作業実行の一時停止を、その作業の作業者へ要求する場合に、APIによって、「実行停止」状態への遷移要求を出します。 なお、作業が「実行停止」状態の場合は、作業振り分けルールの再評価ができます。
強制終了 TERMINATED(u) terminated	closed.terminated	作業が強制的に終了された状態を示します。 APIによって強制終了が要求された場合、作業は「強制終了」状態になります。 また、作業を含む業務ステップが「強制終了」状態になった場合、この作業は「強制終了」状態になります。

上段：状態 中段：列挙型定数（コード値） 下段：状態の短縮表記形式	状態の完全表記形式	説明
		なお、強制終了された作業が含まれる業務ステップは、完了条件が評価されません。この場合は、業務ステップを直接完了させるか、または <code>CIWActivityInstance#evaluate()</code> を使用して業務ステップの完了条件を評価し、業務ステップの完了または作業の再作成をしてください。作業を再作成する場合は、作業の発生条件および完了条件を適切に定義してください。

### (3) 作業の遷移

作業の遷移について次の表に示します。

表 C-8 作業の遷移一覧

遷移種別	要因 および 動作	説明
生成	要因	作業の発生条件が満たされた場合（業務ステップの評価処理時）
	動作	作業を「初期」状態で生成します。
準備済 (ready)	要因	作業の生成時に引続いて遷移する場合（ワーク管理システムによって生成された場合）
	動作	作業の完了条件を評価し、条件が満たされた場合は、続けて「実行省略」状態へ遷移させます。条件が満たされなかった場合は、振分ルール評価（リソースセレクション）を実行し、作業者を割り当てたあとで「実行開始可能」状態へ遷移します。
開始 (start)	要因	組み込み作業の「実行開始可能」状態への遷移
	動作	作業を「自動実行」状態へ遷移させます。
着手 (get)	要因	API 呼び出しによる遷移要求
	動作	作業を「作業中実行」状態へ遷移させます。
返却 (back)	要因	API 呼び出しによる遷移要求
	動作	作業を「実行開始可能」状態へ遷移させます。
作業完了 (workitem completed)	要因	<ul style="list-style-type: none"> <li>API 呼び出しによる遷移要求</li> <li>作業の評価処理で完了条件が満たされた場合（作業の完了条件あり）</li> <li>組み込み作業の完了時</li> </ul>
	動作	作業を「実行済」状態へ遷移させます。
完了通知 (complete)	要因	<ul style="list-style-type: none"> <li>API 呼び出しによる遷移要求</li> <li>作業の評価処理で完了条件が満たされた場合。（作業の完了条件あり）</li> </ul>
	動作	作業を「実行済」状態へ遷移させます。

遷移種別	要因 および 動作	説明
実行省略 (skip)	要因	<ul style="list-style-type: none"> <li>業務ステップの「遷移可」状態への遷移時</li> <li>作業評価時に完了条件が満たされた場合</li> </ul>
	動作	作業を「実行省略」状態へ遷移させます。
実行取消 (cancel)	要因	業務ステップの「遷移可」状態への遷移時
	動作	作業を「実行取消」状態へ遷移させます。 ワーク管理システムによって実行制御が行われている作業の場合は、「実行取消」状態へ遷移することで、実際の作業実行の強制終了制御を行います。
中断 (suspend)	要因	API 呼び出しによる遷移要求
	動作	作業を「実行停止」状態へ遷移させます。 ワーク管理システムによって実行制御が行われている作業の場合は、「中断」状態へ遷移することで、実際の作業実行の停止制御を行います。
再開 (resume)	要因	API 呼び出しによる遷移要求
	動作	作業を「自動実行」状態、または「作業実行」状態へ遷移させます。 ワーク管理システムによって実行制御が行われている作業の場合は、「再開」状態へ遷移することで、実際の作業実行の再開制御を行います。
実行開始不可 (disable)	要因	<ul style="list-style-type: none"> <li>業務ステップの「実行停止」状態への遷移時</li> <li>作業の「実行開始可能」状態への遷移時に、作業を含む業務ステップが「実行停止」状態の場合</li> </ul>
	動作	作業を「実行開始不可」状態へ遷移させます。
実行開始可 (enable)	要因	業務ステップの「実行中」状態への遷移時
	動作	作業を「実行開始可能」状態へ遷移させます。
強制終了 (terminate)	要因	<ul style="list-style-type: none"> <li>API 呼び出しによる遷移要求</li> <li>業務ステップの状態遷移に伴う制約</li> </ul> 業務ステップの状態遷移に伴う制約については、「表 C-5 業務ステップの遷移一覧」の「業務ステップ完了」状態遷移および「強制終了」状態遷移を参照してください。
	動作	作業の状態を「強制終了」状態へ遷移させます。 ワーク管理システムによって実行制御が行われている作業の場合は、「強制終了」状態へ遷移することで、実際の作業実行の中断制御を行います。

作業の遷移時に設定する属性について次の表に示します。



表 C-9 作業の遷移時に設定する属性一覧

属性名	生成	準備済	開始	着手	返却	作業完了	完了通知	実行省略	実行取消	中断	再開	実行開始不可	実行開始可	強制終了
ID	○	-	-	-	-	-	-	-	-	-	-	-	-	-
Name <sup>※1</sup>	○	-	-	-	-	-	-	-	-	-	-	-	-	-
WorkDefinitionName	○	-	-	-	-	-	-	-	-	-	-	-	-	-
ActivityInstanceID	○	-	-	-	-	-	-	-	-	-	-	-	-	-
ProcessInstanceID	○	-	-	-	-	-	-	-	-	-	-	-	-	-
WorkDefinitionID	○	-	-	-	-	-	-	-	-	-	-	-	-	-
ProcessDefinitionID	○	-	-	-	-	-	-	-	-	-	-	-	-	-
WorkTypeCode	○	-	-	-	-	-	-	-	-	-	-	-	-	-
StateCode <sup>※2</sup>	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Participant	N	○	-	-	-	-	-	-	-	-	-	-	-	-
CreationDate <sup>※3</sup>	N	○	-	-	-	-	-	-	-	-	-	-	-	-
StartDate <sup>※4</sup>	N	-	○	○	-	-	-	-	-	-	-	-	-	-
ClosedDate <sup>※3</sup>	N	-	-	-	-	○	○	○	○	-	-	-	-	○
Deadline <sup>※5</sup>	○	△	-	-	-	-	-	-	-	-	-	-	-	-
ProcessInstanceName	○	-	-	-	-	-	-	-	-	-	-	-	-	-
Priority <sup>※6</sup>	○	-	-	-	-	-	-	-	-	-	-	-	-	-

(凡例)

- ：設定します。
- ：設定しません。
- △：条件によって設定します。
- N：null を設定します。

注※1

子作業の場合は、生成ルールが返した値を設定します。  
子作業以外の場合は、null を設定します。

注※2

遷移先の状態を設定します。

注※3

遷移が行われた時刻を設定します。

注※4

遷移が行われた時刻を設定します。返却 (back) への遷移時は未設定に戻しません。2 回目の開始 (start) の遷移が行われた場合は、遷移が行われた時刻を設定します。

注※5

生成時は業務ステップの Deadline の値を設定します。開始 (start) 時に null が設定されている場合は、業務ステップの Deadline の値を設定します。

注※6

業務ステップの Priority の値を設定します。

## 付録 D WorkCoordinator Definer で作成された定義ファイルの利用

CSCIW では、WorkCoordinator Definer で作成したビジネスプロセス定義ファイルおよび振り分けルール定義ファイルを利用できます。WorkCoordinator Definer で作成した定義ファイルは、CSCIW-Definer を使用して読み込みます。なお、CSCIW でサポートしていない項目が定義されている場合は、読み込み時に CSCIW-Definer が対象の項目を削除します。削除された項目については、メッセージに出力されま

す。

WorkCoordinator Definer でサポートしているビジネスプロセス定義および振り分けルール定義の機能のうち、CSCIW で利用できる機能を次の表に示します。

表 D-1 CSCIW で利用できる WorkCoordinator Definer の機能（ビジネスプロセス定義の場合）

機能		利用可否
ソースノード		○
シンクノード		○
業務ステップ	事前生成	○
	完了条件	○
	処理期限	○
	説明	○
作業	作業アプリケーション	×
	振り分けルール	○
	発生条件*	○
	完了条件	○
	条件再評価	○
	説明	○
組み込み作業	並列業務ステップ	×
	並列作業	○
	計時作業	×
	先手通知作業	×
制御ノード	分岐	○
	分業	○
	先着（後続停止あり）	○
	先着（後続停止なし）	○

機能			利用可否	
	待合		○	
アロー			○	
条件	論理データモデル	論理データ項目	×	
	物理データモデル	RDB 検索	○	
		分散オブジェクト	×	
	プロセスデータ	×		
ルール	生成ルール	論理データモデル	論理データ項目	×
		物理データモデル	RDB 検索	○
			分散オブジェクト	×
			プロセスデータ	×
	時間取得ルール	論理データモデル	論理データ項目	×
		物理データモデル	RDB 検索	×
			分散オブジェクト	×
			プロセスデータ	×
		相対時刻	○	
作業アプリケーション情報			×	
論理データ項目			×	
プロセスデータ			×	
階層			○	
ビジネスプロセスの ロパティ	バージョン		○	
	処理期限ルール		○	
	アドホック要求を使用する		○	
	状態		○	
	管理者		○	
	案件投入可能期間		○	
	説明		○	

(凡例)

○：利用できます。

×：利用できません。

注※ 並列作業の子作業以外は、条件に「@False」を設定できません。

表 D-2 CSCIW で利用できる WorkCoordinator Definer の機能 (振り分けルール定義の場合)

機能		利用可否
振り分けルール	RDB 検索	○
	分散オブジェクト	×
振り分けルールのプロパティ	バージョン	○
	管理者	○
	有効期間	○
	説明	○

(凡例)

- ：利用できます。
- ×：利用できません。

## 付録 E 各バージョンの変更内容

---

### 付録 E.1 03-11 の変更内容

- Windows 11 に対応しました。
- 対応する製品から Oracle Database 18c を削除しました。
- CIWActivityDefinition (業務ステップ定義のインタフェース) を追加しました。
- CIWActivityDefinition.AttributeName (業務ステップ定義属性の属性名の列挙型) を追加しました。
- CIWProcessInstance インタフェースの startAndAdhocInstantiateActivityInstanceByADName メソッドの機能説明を修正しました。
- CIWServer インタフェースのメソッドに次を追加しました。
  - getActivityDefinition
  - getActivityDefinitionsList
- CIWServer.Targetkind の列挙型定数に ACTIVITY\_DEFINITION を追加しました。
- 更新系 API 一覧に startAndAdhocInstantiateActivityInstanceByADName を追加しました。
- 参照系 API 一覧に次を追加しました。
  - getActivityDefinitionsList
  - getActivityDefinition
- 参照系 API で指定できる属性に業務ステップ定義属性を追加しました。

### 付録 E.2 03-10 の変更内容

- JDK のバージョンの記述を変更しました。
- 「注意事項」に、Java アプリケーションを実行する際に必要なクラスパスについての参照マニュアルを追加しました。
- CIWProcessInstance インタフェースに startAndAdhocInstantiateActivityInstanceByADName メソッドを追加しました。

### 付録 E.3 03-00 の変更内容

- Cosminexus V11 に対応しました。
- 業務アプリケーションをコンパイルする場合のクラスパスの設定方法を変更しました。

また、Java API 利用時に、uCosminexus Business Process Developer をインストールした環境で業務アプリケーションをコンパイルするときにクラスパスに指定する JAR ファイルの説明を追加しました。

- 条件に一致する作業の作業割当てと着手をする場合の実装例を 2.6 から 2.3.5 に移動しました。

## 付録 E.4 02-30 の変更内容

- Firefox および Chrome に対応しました。

## 付録 E.5 02-20 の変更内容

- 業務プログラムのコンパイルで使用する JDK のバージョンを JDK8.0 に変更しました。
- 条件に一致する作業の作業割当てと着手をする場合の実装例を追加しました。
- CIWServer インタフェースに allocateWorkItemEx メソッドを追加しました。
- CIWorkItem インタフェースに free メソッドを追加しました。
- 業務ステップ遷移一覧の遷移種別「フロー遷移」の要因の説明に、フロー遷移しない場合の説明を追加しました。
- 作業の状態一覧の状態「実行開始可能」および「作業実行」の「説明」に、「作業実行」状態に遷移するタイミングについての説明を追加しました。

## 付録 E.6 02-10 の変更内容

- Java オブジェクト作成時の注意事項を追加しました。

## 付録 E.7 02-00 の変更内容

- BPMN 連携機能を追加しました。
- CIWActivityInstance インタフェースに adhocMakeTransition メソッドを追加しました。
- CIWProcessInstance インタフェースに adhocInstantiateActivityInstance メソッドを追加しました。
- 業務ステップ遷移一覧に、遷移種別「生成」の要因の説明として、API 呼び出しによる生成要求を追加しました。

## 付録 E.8 01-95 の変更内容

なし。

## 付録 E.9 01-90 の変更内容

- 例外がスローされる設定項目に次の SQLSTATE を追加しました。

HiRDBTransientSQLState

OracleTransientSQLState

SQLServerTransientSQLState

## 付録 E.10 01-70 の変更内容

- Cosminexus V8 に対応しました。
- 次の組み込み変数を追加しました。  
@PICreator, @PIStartDate, @PIDecline, @PIPRIORITY, @AIDecline, @AIDecline,  
@AIPRIORITY, @WICreationDate, @WISStartDate, @WIDecline, @WIPRIORITY,  
@WIParticipant, @UserID
- 組み込み変数に関する注意事項を追加しました。
- クラスパスに設定する JAR ファイルに「csciwcmn.jar」を追加しました。
- Java オブジェクト呼び出しへの対応に伴い、説明を追加しました。
- 作業アプリケーション呼び出しへの対応に伴い、説明を追加しました。
- CSCIW で管理するトランザクションに関する説明を追加しました。
- パッケージ「jp.co.Hitachi.soft.csciw.callback」の追加に伴い、API の一覧をパッケージごとに分けました。
- パッケージ「jp.co.Hitachi.soft.csciw.callback」の追加に伴い、CSCIW のクラスの構成をパッケージごとに分けました。
- パッケージ「jp.co.Hitachi.soft.csciw.callback」の階層を追加しました。
- initializeCIWFactory (CIWAdmin クラスのメソッド) に SQL Server の場合を追加しました。
- 次のインタフェースを追加しました。  
CIWConditionEvaluator (条件 Java オブジェクトのインタフェース), CIWCreationRuleEvaluator (生成ルール Java オブジェクトのインタフェース), CIWResourceSelector (振り分けルール Java オブジェクトのインタフェース), CIWWorkApplication (作業アプリケーション Java オブジェクトのインタフェース)
- パッケージ「jp.co.Hitachi.soft.csciw.callback」が提供する列挙型を追加しました。



- パッケージ「jp.co.Hitachi.soft.csciw.callback」の例外クラスを追加しました。
- CSCIW で使用する API に更新系 API および参照系 API を追加しました。
- 次の作業の状態の説明を変更しました。  
作業者実行, 実行済

## 付録 E.11 01-60 の変更内容

- ビジネスプロセスを管理するツールを, WorkCoordinator Definer から CSCIW-Definer へ変更しました。
- ビジネスプロセスの変更に, 定義のバージョンを更新する場合の手順を追加しました。
- CSCIW がワーク管理データベースを検索するときの SQL 文の評価規則, および注意事項を追加しました。
- CIWActivityInstance (業務ステップのインタフェース) に, 引き戻しまたは差し戻しができる makeBackwardTransition メソッドを追加しました。
- 例外クラス CIWTransientException が発生する要因の説明に, SQL Server 使用時に SQL 例外となる SQLCODE の値を設定する, セットアッププロパティファイルの設定項目「SQLServerTransientSQLCode」を追加しました。

## 付録 F 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 F.1 関連マニュアル

このマニュアルの関連マニュアルを次に示します。必要に応じてお読みください。

#### (1) CSCIW 関連

- uCosminexus Service Coordinator Interactive Workflow システム構築・運用ガイド (3020-3-M80)
- uCosminexus Service Coordinator Interactive Workflow 案件運用操作ガイド (3020-3-M82)
- uCosminexus Service Coordinator Interactive Workflow メッセージ (3020-3-M83)
- uCosminexus Service Coordinator Interactive Workflow コマンド (3020-3-M84)
- uCosminexus Service Coordinator Interactive Workflow ビジネスプロセス開発ガイド (3020-3-M85)
- uCosminexus Service Coordinator Interactive Workflow BPMN 連携機能 使用の手引 (3020-3-M86)

#### (2) Cosminexus 関連

- Cosminexus V11 アプリケーションサーバ システム構築・運用ガイド (3021-3-J02)
- Cosminexus V11 アプリケーションサーバ 機能解説 基本・開発編 (Web コンテナ) (3021-3-J05)
- Cosminexus V11 アプリケーションサーバ 機能解説 基本・開発編 (EJB コンテナ) (3021-3-J06)
- Cosminexus V11 アプリケーションサーバ 機能解説 基本・開発編 (コンテナ共通機能) (3021-3-J07)
- Cosminexus V11 アプリケーションサーバ 機能解説 拡張編 (3021-3-J08)
- Cosminexus V11 アプリケーションサーバ 機能解説 運用／監視／連携編 (3021-3-J10)
- Cosminexus V11 アプリケーションサーバ 機能解説 保守／移行編 (3021-3-J11)
- Cosminexus V11 アプリケーションサーバ 機能解説 互換編 (3021-3-J12)
- Cosminexus V11 アプリケーションサーバ アプリケーション設定操作ガイド (3021-3-J13)
- Cosminexus V11 アプリケーションサーバ リファレンス コマンド編 (3021-3-J15)
- Cosminexus V11 アプリケーションサーバ リファレンス 定義編 (サーバ定義) (3021-3-J16)
- Cosminexus V11 アプリケーションサーバ リファレンス 定義編 (アプリケーション／リソース定義) (3021-3-J17)
- Cosminexus V11 アプリケーションサーバ アプリケーション開発ガイド (3021-3-J20)

- Cosminexus V11 BPM/ESB 基盤 サービスプラットフォーム システム構築・運用ガイド (3021-3-J45)

なお、このマニュアルでは、Cosminexus 関連のマニュアルについて、バージョン番号を省略して表記しています。

### (3) HiRDB 関連

- HiRDB Version 10 解説 (3020-6-551)
- HiRDB Version 10 システム導入・設計ガイド (UNIX(R)用) (3020-6-552)
- HiRDB Version 10 システム定義 (UNIX(R)用) (3020-6-554)
- HiRDB Version 10 システム運用ガイド (UNIX(R)用) (3020-6-556)
- HiRDB Version 10 コマンドリファレンス (UNIX(R)用) (3020-6-558)
- HiRDB Version 10 システム導入・設計ガイド (Windows(R)用) (3020-6-553)
- HiRDB Version 10 システム定義 (Windows(R)用) (3020-6-555)
- HiRDB Version 10 システム運用ガイド (Windows(R)用) (3020-6-557)
- HiRDB Version 10 コマンドリファレンス (Windows(R)用) (3020-6-559)
- HiRDB Version 10 UAP 開発ガイド (3020-6-560)
- HiRDB Version 10 SQL リファレンス (3020-6-561)
- HiRDB Version 10 メッセージ (3020-6-562)

なお、このマニュアルでは、HiRDB 関連のマニュアルについて、バージョン番号を省略して表記しています。

## 付録 F.2 このマニュアルでの表記

このマニュアルで使用している表記と、対応する製品名を次に示します。

表記	製品名
AIX*	IBM AIX V7.1
Chrome	Google Chrome
Cosminexus	uCosminexus Application Server
CSCIW	uCosminexus Service Coordinator Interactive Workflow
Firefox	Mozilla Firefox
HiRDB	HiRDB Server Version 10
Linux*	Red Hat Enterprise Linux Server 7 (64-bit x86_64)
	Red Hat Enterprise Linux Server 8 (64-bit x86_64)

表記	製品名
	Red Hat Enterprise Linux Server 9 (64-bit x86_64)
ORACLE	Oracle(R) Database 19c
WorkCoordinator Definer	WorkCoordinator Definer Version 6

注※

AIX および Linux の製品名称を特に区別する必要がない場合、UNIX と表記しています。

## 付録 F.3 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
SQL Server	SQL Server 2005	Microsoft SQL Server 2005
	SQL Server 2008	Microsoft SQL Server 2008
		Microsoft SQL Server 2008 R2
SQL Server の JDBC ドライバ		Microsoft SQL Server 2005 JDBC Driver 1.1
		Microsoft SQL Server 2005 JDBC Driver 1.2
		Microsoft SQL Server JDBC Driver 2.0
		Microsoft SQL Server JDBC Driver 3.0
Windows Server	Windows Server 2016	Microsoft Windows Server 2016 Standard 日本語版
		Microsoft Windows Server 2016 Datacenter 日本語版
	Windows Server 2019	Microsoft Windows Server 2019 Standard 日本語版
		Microsoft Windows Server 2019 Datacenter 日本語版
	Windows Server 2022	Microsoft Windows Server 2022 Standard 日本語版
		Microsoft Windows Server 2022 Datacenter 日本語版
Windows	Windows 10	Windows 10 Pro 日本語版(64 ビット版)
		Windows 10 Enterprise 日本語版(64 ビット版)
	Windows 11	Windows 11 Pro 日本語版
		Windows 11 Enterprise 日本語版

Windows Server 2016, Windows Server 2019, Windows Server 2022, Windows 10, および Windows 11 を特に区別する必要がない場合、**Windows** と表記しています。

## 付録 F.4 このマニュアルで使用している略語

このマニュアルで使用している英略語を次に示します。

英略語	英字での表記
EJB	Enterprise JavaBeans
GMT	Greenwich Mean Time
J2EE	Java 2 Platform, Enterprise Edition, および J2EE
JAR	Java ARchive
JDBC	Java Database Connectivity, および JDBC
JDK	Java Development Kit, および JDK

## 付録 F.5 KB (キロバイト) などの単位表記について

1KB (キロバイト), 1MB (メガバイト), 1GB (ギガバイト), 1TB (テラバイト) はそれぞれ  $1,024$  バイト,  $1,024^2$  バイト,  $1,024^3$  バイト,  $1,024^4$  バイトです。

# 索引

## A

- adhocInstantiateActivityInstance [CIWProcessInstance] 134
- adhocInstantiateActivityInstance で発生する例外 [CIWProcessInstance] 135
- adhocInstantiateActivityInstance の引数 [CIWProcessInstance] 134
- adhocMakeTransition [CIWActivityInstance] 99
- adhocMakeTransition で発生する例外 [CIWActivityInstance] 100
- adhocMakeTransition の引数 [CIWActivityInstance] 100
- allocateWorkItemEx [CIWServer] 157
- allocateWorkItemEx で発生する例外 [CIWServer] 158
- allocateWorkItemEx の引数 [CIWServer] 158
- aNewState で指定できる案件の状態 [CIWProcessInstance] 136
- aNewState で指定できる業務ステップの状態 [CIWActivityInstance] 101
- aNewState で指定できる作業の状態 [CIWorkItem] 192
- API の特長 40
- API 利用時の注意事項 74
- attachDatabaseConnection [CIWConnectionCoordinator] 222
- attachDatabaseConnection で発生する例外 [CIWConnectionCoordinator] 223
- attachDatabaseConnection の引数 [CIWConnectionCoordinator] 223

## C

- changeState [CIWActivityInstance] 100
- changeState [CIWProcessInstance] 136
- changeState [CIWorkItem] 191
- changeState で発生する例外 [CIWActivityInstance] 102

- changeState で発生する例外 [CIWProcessInstance] 137
- changeState で発生する例外 [CIWorkItem] 193
- changeState の引数 [CIWActivityInstance] 101
- changeState の引数 [CIWProcessInstance] 136
- changeState の引数 [CIWorkItem] 192
- CIWActivityDefinition 94
- CIWActivityDefinition.AttributeName 231
- CIWActivityDefinition.AttributeName のメソッド 231
- CIWActivityDefinition.AttributeName の列挙型定数 231
- CIWActivityDefinition インタフェースの入れ子のクラス 94
- CIWActivityDefinition インタフェースのメソッド 94
- CIWActivityInstance 98
- CIWActivityInstance.AttributeName 235
- CIWActivityInstance.AttributeName のメソッド 236
- CIWActivityInstance.AttributeName の列挙型定数 235
- CIWActivityInstance.State 240
- CIWActivityInstance.State のメソッド 242
- CIWActivityInstance.State の列挙型定数 240
- CIWActivityInstance.Type 252
- CIWActivityInstance.Type のメソッド 253
- CIWActivityInstance.Type の列挙型定数 252
- CIWActivityInstance インタフェースの入れ子のクラス 98
- CIWActivityInstance インタフェースのメソッド 98
- CIWAdmin 88
- CIWAdmin クラスのメソッド 88
- CIWConditionEvaluator 342
- CIWConditionEvaluator.Parameter 353
- CIWConditionEvaluator.Parameter のメソッド 355
- CIWConditionEvaluator.Parameter の列挙型定数 353

CIWConditionEvaluator インタフェースのメソッド  
342

CIWConditionEvaluator の入れ子のクラス 342

CIWConnectionCoordinator 222

CIWConnectionCoordinator クラスのメソッド  
222

CIWConnectionCoordinator を使用した  
Connection オブジェクトの関連づけ [業務プログラ  
ムで管理するトランザクション (CIWServer とコネ  
クションの関連づけ)] 70

CIWConnectionCoordinator を使用してコネクショ  
ンと関連づける場合 [CIWServer オブジェクトの生  
成] 44

CIWCreationRuleEvaluator 344

CIWCreationRuleEvaluator.Parameter 358

CIWCreationRuleEvaluator.Parameter のメソッド  
360

CIWCreationRuleEvaluator.Parameter の列挙型  
定数 358

CIWEntityNotExistException 328

CIWException 329

CIWException 直系の既知のサブクラス 329

CIWFactory 118

CIWFactory オブジェクトの取得 43

CIWFactory クラスのメソッド 118

CIWFactory と CIWServer オブジェクトの生成ク  
ラス 118

CIWFatalException 330

CIWFatalException 直系の既知のサブクラス 330

CIWProcessDefinition 122

CIWProcessDefinition.AttributeName 258

CIWProcessDefinition.AttributeName のメソッド  
259

CIWProcessDefinition.AttributeName の列挙型  
定数 258

CIWProcessDefinition.State 262

CIWProcessDefinition.State のメソッド 263

CIWProcessDefinition.State の列挙型定数 262

CIWProcessDefinition インタフェースの入れ子のク  
ラス 122

CIWProcessDefinition インタフェースのメソッド  
122

CIWProcessInstance 133

CIWProcessInstance.AttributeName 269

CIWProcessInstance.AttributeName のメソッド  
270

CIWProcessInstance.AttributeName の列挙型  
定数 269

CIWProcessInstance.State 273

CIWProcessInstance.State のメソッド 274

CIWProcessInstance.State の列挙型定数 273

CIWProcessInstance インタフェースの入れ子のク  
ラス 133

CIWProcessInstance インタフェースのメソッド  
133

CIWResourceSelector 346

CIWResourceSelector.Parameter 363

CIWResourceSelector.Parameter のメソッド 365

CIWResourceSelector.Parameter の列挙型定数  
363

CIWResourceSelector インタフェースのメソッド  
346

CIWResourceSelector の入れ子のクラス 346

CIWServer 156

CIWServer.Targetkind 283

CIWServer.Targetkind のメソッド 284

CIWServer.Targetkind の列挙型定数 283

CIWServer インタフェースの入れ子のクラス 156

CIWServer インタフェースのメソッド 156

CIWServer オブジェクトに対する Connection オブ  
ジェクトの関連づけ 45

CIWServer オブジェクトに対する Connection オブ  
ジェクトの関連づけの解除 48

CIWServer オブジェクトの生成 44

CIWServer と JDBC コネクションの関連づけを管理  
するクラス 222

CIWServer とコネクションの関連づけ [業務プログ  
ラムで管理するトランザクション] 70

CIWSQLException 331

CIWSQLException のメソッド 331

CIWSQLTransientException 334

CIWSQLTransientException のメソッド 334  
CIWStateException 337  
CIWTransientException 338  
CIWTransientException 直系の既知のサブクラス 338  
CIWTransitionFailedException 339  
CIWUserException 379  
CIWUserException (ユーザエラーメッセージ) 379  
CIWUserException (ユーザエラーメッセージおよびエラー原因) 380  
CIWUserException の引数 (ユーザエラーメッセージ) 380  
CIWUserException の引数 (ユーザエラーメッセージおよびエラー原因) 380  
CIWUtil 225  
CIWUtil.InfinityDate 287  
CIWUtil.InfinityDate のメソッド 287  
CIWUtil.InfinityDate の列挙型定数 287  
CIWUtil クラスのメソッド 225  
CIWUtil の入れ子のクラス 225  
CIWorkApplication 348  
CIWorkApplication.Parameter 368  
CIWorkApplication.Parameter のメソッド 370  
CIWorkApplication.Parameter の列挙型定数 368  
CIWorkApplication.Result 373  
CIWorkApplication.Result のメソッド 373  
CIWorkApplication.Result の列挙型定数 373  
CIWorkApplication インタフェースのメソッド 348  
CIWorkApplication の入れ子のクラス 348  
CIWorkDefinition 184  
CIWorkDefinition.AttributeName 291  
CIWorkDefinition.AttributeName のメソッド 292  
CIWorkDefinition.AttributeName の列挙型定数 291  
CIWorkDefinition.Type 295  
CIWorkDefinition.Type のメソッド 296  
CIWorkDefinition.Type の列挙型定数 295  
CIWorkDefinition インタフェースの入れ子のクラス 184  
CIWorkDefinition インタフェースのメソッド 184  
CIWorkItem 190  
CIWorkItem.AttributeName 302  
CIWorkItem.AttributeName のメソッド 303  
CIWorkItem.AttributeName の列挙型定数 302  
CIWorkItem.State 307  
CIWorkItem.State のメソッド 309  
CIWorkItem.State の列挙型定数 307  
CIWorkItem.Type 319  
CIWorkItem.Type のメソッド 320  
CIWorkItem.Type の列挙型定数 319  
CIWorkItem インタフェースの入れ子のクラス 190  
CIWorkItem インタフェースのメソッド 190  
complete [CIWorkItem] 194  
complete で発生する例外 [CIWorkItem] 194  
convertDateToString [CIWUtil] 225, 226  
convertDateToString の引数 [CIWUtil] 226  
createAndStartProcessInstance [CIWServer] 159  
createAndStartProcessInstance で発生する例外 [CIWServer] 160  
createAndStartProcessInstance の引数 [CIWServer] 160  
createCIWServer [CIWFactory] 118  
createCIWServer [CIWFactory (javax.sql.DataSource)] 119  
createCIWServer で発生する例外 [CIWFactory] 119  
createCIWServer で発生する例外 [CIWFactory (javax.sql.DataSource)] 120  
createCIWServer の引数 [CIWFactory] 119  
createCIWServer の引数 [CIWFactory (javax.sql.DataSource)] 120  
createProcessInstance [CIWServer] 161  
createProcessInstance で発生する例外 [CIWServer] 162  
createProcessInstance の引数 [CIWServer] 162  
CSCIWManagementServer を停止した状態での API 実行について [API 利用時の注意事項] 74



CSCIW で管理するトランザクション 69  
CSCIW で使用するビジネスプロセス定義と振り分け  
ルール定義の開発 22  
CSCIW の終了 48  
CSCIW の初期化 43  
CSCIW の初期化, 終了処理をするクラス 88  
CSCIW を利用した業務処理の実装 45

## D

DBMS の一時的な障害発生の場合 334  
DBMS の障害発生の場合 331  
delete [CIWProcessInstance] 137  
delete で発生する例外 [CIWProcessInstance]  
138  
detachDatabaseConnection  
[CIWConnectionCoordinator] 223

## E

evaluate [CIWActivityInstance] 102  
evaluate [CIWWorkItem] 195  
evaluateCondition [CIWConditionEvaluator]  
342  
evaluateCondition の引数 343  
evaluateCreationRule  
[CIWCreationRuleEvaluator] 344  
evaluateCreationRule の引数 345  
evaluate で発生する例外 [CIWActivityInstance]  
103  
evaluate で発生する例外 [CIWWorkItem] 195

## F

finalize()メソッド内での使用について [API 利用時の  
注意事項] 74  
finalizeCIWFactory [CIWAdmin] 88  
free [CIWWorkItem] 196  
free で発生する例外 [CIWWorkItem] 196  
fromStateCode [CIWActivityInstance.State]  
242  
fromStateCode [CIWProcessDefinition.State]  
263

fromStateCode [CIWProcessInstance.State]  
275  
fromStateCode [CIWWorkItem.State] 310  
fromStateCode で発生する例外  
[CIWWorkItem.State] 310  
fromStateCode の引数  
[CIWActivityInstance.State] 243  
fromStateCode の引数  
[CIWProcessDefinition.State] 263  
fromStateCode の引数  
[CIWProcessInstance.State] 275  
fromStateCode の引数 [CIWWorkItem.State]  
310  
fromStateCode の例外  
[CIWActivityInstance.State] 243  
fromStateCode の例外  
[CIWProcessDefinition.State] 264  
fromStateCode の例外  
[CIWProcessInstance.State] 275  
fromTypeCode [CIWActivityInstance.Type]  
253  
fromTypeCode [CIWWorkDefinition.Type]  
296  
fromTypeCode [CIWWorkItem.Type] 320  
fromTypeCode で発生する例外  
[CIWWorkDefinition.Type] 297  
fromTypeCode で発生する例外  
[CIWWorkItem.Type] 321  
fromTypeCode の引数  
[CIWActivityInstance.Type] 253  
fromTypeCode の引数  
[CIWWorkDefinition.Type] 297  
fromTypeCode の引数 [CIWWorkItem.Type]  
321  
fromTypeCode の例外  
[CIWActivityInstance.Type] 254

## G

getActivityDefinition [CIWServer] 163  
getActivityDefinitionID [CIWActivityInstance]  
103

[getActivityDefinitionID \[CIWorkDefinition\]](#) [184](#)  
[getActivityDefinitionID で発生する例外 \[CIWActivityInstance\]](#) [104](#)  
[getActivityDefinitionID で発生する例外 \[CIWorkDefinition\]](#) [185](#)  
[getActivityDefinitionName \[CIWActivityInstance\]](#) [104](#)  
[getActivityDefinitionName で発生する例外 \[CIWActivityInstance\]](#) [105, 113](#)  
[getActivityDefinitionsList \[CIWServer\]](#) [164](#)  
[getActivityDefinitionsList で発生する例外 \[CIWServer\]](#) [166](#)  
[getActivityDefinitionsList の引数 \[CIWServer\]](#) [165](#)  
[getActivityDefinition で発生する例外 \[CIWServer\]](#) [164](#)  
[getActivityDefinition の引数 \[CIWServer\]](#) [163](#)  
[getActivityInstance \[CIWServer\]](#) [166](#)  
[getActivityInstanceID \[CIWorkItem\]](#) [197](#)  
[getActivityInstanceID で発生する例外 \[CIWorkItem\]](#) [197](#)  
[getActivityInstancesList \[CIWServer\]](#) [168](#)  
[getActivityInstancesList で発生する例外 \[CIWServer\]](#) [169](#)  
[getActivityInstancesList の引数 \[CIWServer\]](#) [168](#)  
[getActivityInstance で発生する例外 \[CIWServer\]](#) [167](#)  
[getActivityInstance の引数 \[CIWServer\]](#) [167](#)  
[getAuthor \[CIWProcessDefinition\]](#) [123](#)  
[getAuthor で発生する例外 \[CIWProcessDefinition\]](#) [123](#)  
[getAvailableParticipant \[CIWResourceSelector\]](#) [346](#)  
[getAvailableParticipant の引数](#) [347](#)  
[getCastingRuleName \[CIWorkDefinition\]](#) [185](#)  
[getCastingRuleName で発生する例外 \[CIWorkDefinition\]](#) [186](#)  
[getCause \[CIWSQLException\]](#) [331](#)  
[getCause \[CIWSQLTransientException\]](#) [335](#)  
[getClosedDate \[CIWActivityInstance\]](#) [105](#)  
[getClosedDate \[CIWProcessInstance\]](#) [138](#)  
[getClosedDate \[CIWorkItem\]](#) [197](#)  
[getClosedDate で発生する例外 \[CIWActivityInstance\]](#) [105](#)  
[getClosedDate で発生する例外 \[CIWProcessInstance\]](#) [139](#)  
[getClosedDate で発生する例外 \[CIWorkItem\]](#) [198](#)  
[getConnectionCoordinator \[CIWServer\]](#) [169](#)  
[getConnectionCoordinator で発生する例外 \[CIWServer\]](#) [170](#)  
[getCreationDate \[CIWProcessDefinition\]](#) [123](#)  
[getCreationDate \[CIWorkItem\]](#) [198](#)  
[getCreationDate で発生する例外 \[CIWProcessDefinition\]](#) [124](#)  
[getCreationDate で発生する例外 \[CIWorkItem\]](#) [199](#)  
[getCreator \[CIWProcessInstance\]](#) [139](#)  
[getCreator で発生する例外 \[CIWProcessInstance\]](#) [139](#)  
[getDeadline \[CIWActivityInstance\]](#) [106](#)  
[getDeadline \[CIWProcessInstance\]](#) [140](#)  
[getDeadline \[CIWorkItem\]](#) [199](#)  
[getDeadline で発生する例外 \[CIWActivityInstance\]](#) [106](#)  
[getDeadline で発生する例外 \[CIWProcessInstance\]](#) [140](#)  
[getDeadline で発生する例外 \[CIWorkItem\]](#) [200](#)  
[getDescription \[CIWActivityDefinition\]](#) [94](#)  
[getDescription \[CIWProcessDefinition\]](#) [124](#)  
[getDescription \[CIWorkDefinition\]](#) [186](#)  
[getDescription で発生する例外 \[CIWActivityDefinition\]](#) [95](#)  
[getDescription で発生する例外 \[CIWProcessDefinition\]](#) [125](#)  
[getDescription で発生する例外 \[CIWorkDefinition\]](#) [187](#)  
[getErrorCode \[CIWSQLException\]](#) [332](#)

[getErrorCode \[CIWSQLTransientException\]](#)  
[335](#)

[getID \[CIWActivityDefinition\]](#) [95](#)  
[getID \[CIWActivityInstance\]](#) [106](#)  
[getID \[CIWProcessDefinition\]](#) [125](#)  
[getID \[CIWProcessInstance\]](#) [140](#)  
[getID \[CIWorkDefinition\]](#) [187](#)  
[getID \[CIWorkItem\]](#) [200](#)  
[getID で発生する例外 \[CIWActivityDefinition\]](#) [96](#)  
[getID で発生する例外 \[CIWActivityInstance\]](#) [107](#)  
[getID で発生する例外 \[CIWProcessDefinition\]](#)  
[125](#)  
[getID で発生する例外 \[CIWProcessInstance\]](#) [141](#)  
[getID で発生する例外 \[CIWorkDefinition\]](#) [187](#)  
[getID で発生する例外 \[CIWorkItem\]](#) [200](#)  
[getInstance \[CIWFactory\]](#) [121](#)  
[getInstance で発生する例外 \[CIWFactory\]](#) [121](#)  
[getListCount \[CIWServer\]](#) [170](#)  
[getListCount で発生する例外 \[CIWServer\]](#) [171](#)  
[getListCount の引数 \[CIWServer\]](#) [170](#)  
[getMovedDate \[CIWProcessInstance\]](#) [141](#)  
[getMovedDate で発生する例外 \[CIWProcessInstance\]](#) [142](#)  
[getName \[CIWActivityDefinition\]](#) [96](#)  
[getName \[CIWActivityInstance\]](#) [107](#)  
[getName \[CIWProcessDefinition\]](#) [125](#)  
[getName \[CIWProcessInstance\]](#) [142](#)  
[getName \[CIWorkDefinition\]](#) [187](#)  
[getName \[CIWorkItem\]](#) [200](#)  
[getName で発生する例外 \[CIWActivityDefinition\]](#)  
[96](#)  
[getName で発生する例外 \[CIWActivityInstance\]](#)  
[108](#)  
[getName で発生する例外 \[CIWProcessDefinition\]](#)  
[126](#)  
[getName で発生する例外 \[CIWProcessInstance\]](#)  
[142](#)  
[getName で発生する例外 \[CIWorkDefinition\]](#)  
[188](#)  
[getName で発生する例外 \[CIWorkItem\]](#) [201](#)

[getParentProcessInstance \[CIWorkItem\]](#)  
[201](#)  
[getParentProcessInstance で発生する例外 \[CIWorkItem\]](#) [202](#)  
[getParentProcessInstance の引数 \[CIWorkItem\]](#) [202](#)  
[getParticipant \[CIWorkItem\]](#) [202](#)  
[getParticipant で発生する例外 \[CIWorkItem\]](#)  
[203](#)  
[getPriority \[CIWActivityInstance\]](#) [108](#)  
[getPriority \[CIWProcessInstance\]](#) [143](#)  
[getPriority \[CIWorkItem\]](#) [203](#)  
[getPriority で発生する例外 \[CIWActivityInstance\]](#)  
[108](#)  
[getPriority で発生する例外 \[CIWProcessInstance\]](#)  
[143](#)  
[getPriority で発生する例外 \[CIWorkItem\]](#) [204](#)  
[getProcessDefinition \[CIWProcessInstance\]](#)  
[144](#)  
[getProcessDefinition \[CIWServer\]](#) [171](#)  
[getProcessDefinitionID \[CIWActivityDefinition\]](#)  
[96](#)  
[getProcessDefinitionID \[CIWActivityInstance\]](#)  
[109](#)  
[getProcessDefinitionID \[CIWProcessInstance\]](#)  
[145](#)  
[getProcessDefinitionID \[CIWorkDefinition\]](#)  
[188](#)  
[getProcessDefinitionID \[CIWorkItem\]](#) [204](#)  
[getProcessDefinitionID で発生する例外 \[CIWActivityDefinition\]](#) [97](#)  
[getProcessDefinitionID で発生する例外 \[CIWActivityInstance\]](#) [109](#)  
[getProcessDefinitionID で発生する例外 \[CIWProcessInstance\]](#) [145](#)  
[getProcessDefinitionID で発生する例外 \[CIWorkDefinition\]](#) [189](#)  
[getProcessDefinitionID で発生する例外 \[CIWorkItem\]](#) [204](#)  
[getProcessDefinitionName \[CIWProcessInstance\]](#) [145](#)

getProcessDefinitionName で発生する例外  
 [CIWProcessInstance] 146

getProcessDefinitionsList [CIWServer] 172

getProcessDefinitionsList で発生する例外  
 [CIWServer] 174

getProcessDefinitionsList の引数 [CIWServer]  
 173

getProcessDefinition で発生する例外  
 [CIWProcessInstance] 144

getProcessDefinition で発生する例外  
 [CIWServer] 172

getProcessDefinition の引数  
 [CIWProcessInstance] 144

getProcessDefinition の引数 [CIWServer] 172

getProcessInstance [CIWServer] 174

getProcessInstanceId [CIWActivityInstance]  
 109

getProcessInstanceId [CIWWorkItem] 205

getProcessInstanceId で発生する例外  
 [CIWActivityInstance] 110

getProcessInstanceId で発生する例外  
 [CIWWorkItem] 205

getProcessInstanceName  
 [CIWActivityInstance] 110

getProcessInstanceName [CIWWorkItem] 205

getProcessInstanceName で発生する例外  
 [CIWActivityInstance] 111

getProcessInstanceName で発生する例外  
 [CIWWorkItem] 206

getProcessInstancesList  
 [CIWProcessDefinition] 126

getProcessInstancesList [CIWServer] 176

getProcessInstancesList で発生する例外  
 [CIWProcessDefinition] 128

getProcessInstancesList で発生する例外  
 [CIWServer] 177

getProcessInstancesList の引数  
 [CIWProcessDefinition] 127

getProcessInstancesList の引数 [CIWServer]  
 176

getProcessInstance で発生する例外 [CIWServer]  
 175

getProcessInstance の引数 [CIWServer] 175

getResponsible [CIWProcessDefinition] 128

getResponsible で発生する例外  
 [CIWProcessDefinition] 128

getSQLState [CIWSQLException] 332

getSQLState [CIWSQLTransientException] 336

getStartDate [CIWActivityInstance] 111

getStartDate [CIWProcessInstance] 146

getStartDate [CIWWorkItem] 206

getStartDate で発生する例外  
 [CIWActivityInstance] 111

getStartDate で発生する例外  
 [CIWProcessInstance] 147

getStartDate で発生する例外 [CIWWorkItem]  
 207

getState [CIWActivityInstance] 112

getState [CIWProcessDefinition] 129

getState [CIWProcessInstance] 147

getState [CIWWorkItem] 207

getState で発生する例外 [CIWActivityInstance]  
 112

getState で発生する例外 [CIWProcessDefinition]  
 129

getState で発生する例外 [CIWProcessInstance]  
 148

getState で発生する例外 [CIWWorkItem] 207

getType [CIWActivityInstance] 113

getType [CIWWorkDefinition] 189

getType [CIWWorkItem] 208

getType で発生する例外 [CIWWorkDefinition]  
 189

getType で発生する例外 [CIWWorkItem] 208

getValidFromDate [CIWProcessDefinition]  
 129

getValidFromDate で発生する例外  
 [CIWProcessDefinition] 130

getValidToDate [CIWProcessDefinition] 130

getValidToDate で発生する例外  
 [CIWProcessDefinition] 131

getVersion [CIWProcessDefinition] 131

getVersion で発生する例外  
[CIWProcessDefinition] 132

getWorkDefinition [CIWServer] 177

getWorkDefinition [CIWWorkItem] 208

getWorkDefinitionID [CIWWorkItem] 210

getWorkDefinitionID で発生する例外  
[CIWWorkItem] 210

getWorkDefinitionName [CIWWorkItem] 210

getWorkDefinitionName で発生する例外  
[CIWWorkItem] 211

getWorkDefinitionsList [CIWServer] 179

getWorkDefinitionsList で発生する例外  
[CIWServer] 180

getWorkDefinitionsList の引数 [CIWServer] 179

getWorkDefinition で発生する例外 [CIWServer]  
178

getWorkDefinition で発生する例外  
[CIWWorkItem] 209

getWorkDefinition の引数 [CIWServer] 178

getWorkDefinition の引数 [CIWWorkItem] 209

getWorkItem [CIWServer] 180

getWorkItemsList [CIWProcessInstance] 148

getWorkItemsList [CIWServer] 182

getWorkItemsList で発生する例外  
[CIWProcessInstance] 149

getWorkItemsList で発生する例外 [CIWServer]  
183

getWorkItemsList の引数 [CIWProcessInstance]  
148

getWorkItemsList の引数 [CIWServer] 182

getWorkItem で発生する例外 [CIWServer] 181

getWorkItem の引数 [CIWServer] 181

## I

initializeCIWFactory [CIWAdmin] 89

initializeCIWFactory で発生する例外 90

initializeCIWFactory の引数 89

isClosed [CIWActivityInstance.State] 243

isClosed [CIWProcessInstance.State] 276

isClosed [CIWWorkItem.State] 310

isCompleted [CIWActivityInstance.State] 244

isCompleted [CIWWorkItem.State] 311

isExecuted [CIWActivityInstance.State] 244

isNotRunning [CIWActivityInstance.State] 245

isNotRunning [CIWProcessInstance.State] 276

isNotRunning [CIWWorkItem.State] 312

isNotStarted [CIWActivityInstance.State] 246

isNotStarted [CIWWorkItem.State] 312

isOpen [CIWActivityInstance.State] 246

isOpen [CIWProcessInstance.State] 277

isOpen [CIWWorkItem.State] 313

isRunning [CIWWorkItem.State] 313

isSuspended [CIWActivityInstance.State] 247

isSuspended [CIWProcessInstance.State] 277

isSuspended [CIWWorkItem.State] 314

## J

J2EE サーバで使用する場合 [CSCIW の終了] 48

Java アプリケーションで使用する場合 [CSCIW の終了] 48

Java アプリケーションで使用する場合 [CSCIW の初期化] 43

Java オブジェクトに渡される情報 (作業アプリケーションの場合) 58

Java オブジェクトに渡される情報 (条件、生成ルールおよび振り分けルールの場合) 57

Java オブジェクトの作成 52

Java オブジェクトの実装例 (作業アプリケーション) 55

Java オブジェクトの実装例 (条件、生成ルール、および振り分けルール) 52

Java オブジェクト呼び出し時にエラーが発生した場合 379

Java オブジェクト呼び出しとコネクションの関連づけ 71

Java オブジェクト呼び出し用のインタフェース 340

Java オブジェクト呼び出し利用時の設定 50

Java 業務プログラムの作成 41

Java コンパイラのオプション 41

## M

- makeBackwardTransition [CIWActivityInstance] 113
- makeBackwardTransition で発生する例外 [CIWActivityInstance] 114
- makeBackwardTransition の引数 [CIWActivityInstance] 114

## P

- performAndComplete [CIWWorkItem] 212
- performAndComplete で発生する例外 [CIWWorkItem] 213
- perform [CIWWorkItem] 211
- perform で発生する例外 [CIWWorkItem] 212

## R

- reassignAndPerform [CIWWorkItem] 214
- reassignAndPerform で発生する例外 [CIWWorkItem] 215
- reassignAndPerform の引数 [CIWWorkItem] 215
- reassign [CIWWorkItem] 213
- reassign で発生する例外 [CIWWorkItem] 214
- reassign の引数 [CIWWorkItem] 214
- reevaluateCastingRule [CIWWorkItem] 216
- reevaluateCastingRule で発生する例外 [CIWWorkItem] 217

## S

- setDeadline [CIWActivityInstance] 115
- setDeadline [CIWProcessInstance] 149
- setDeadline [CIWWorkItem] 217
- setDeadline で発生する例外 [CIWActivityInstance] 116
- setDeadline で発生する例外 [CIWProcessInstance] 150
- setDeadline で発生する例外 [CIWWorkItem] 218
- setDeadline の引数 [CIWActivityInstance] 115
- setDeadline の引数 [CIWProcessInstance] 150
- setDeadline の引数 [CIWWorkItem] 218

- setPriority [CIWActivityInstance] 116
- setPriority [CIWProcessInstance] 151
- setPriority [CIWWorkItem] 218
- setPriority で発生する例外 [CIWActivityInstance] 117
- setPriority で発生する例外 [CIWProcessInstance] 151
- setPriority で発生する例外 [CIWWorkItem] 219
- setPriority の引数 [CIWActivityInstance] 116
- setPriority の引数 [CIWProcessInstance] 151
- setPriority の引数 [CIWWorkItem] 219
- startAndAdhocInstantiateActivityInstanceByAD Name [CIWProcessInstance] 153
- startAndAdhocInstantiateActivityInstanceByAD Name で発生する例外 [CIWProcessInstance] 154
- startAndAdhocInstantiateActivityInstanceByAD Name の引数 [CIWProcessInstance] 153
- start [CIWProcessInstance] 152
- startSynchronously [CIWWorkApplication] 348
- startSynchronously の引数 349
- start で発生する例外 [CIWProcessInstance] 152
- State 属性で指定できる値 391, 392

## T

- terminate [CIWProcessInstance] 154
- terminate で発生する例外 [CIWProcessInstance] 155
- toAttributeName [CIWActivityDefinition.AttributeName] 232
- toAttributeName [CIWActivityInstance.AttributeName] 237
- toAttributeName [CIWProcessDefinition.AttributeName] 259
- toAttributeName [CIWProcessInstance.AttributeName] 270
- toAttributeName [CIWWorkDefinition.AttributeName] 292
- toAttributeName [CIWWorkItem.AttributeName] 304

toDate [CIWUtil.InfinityDate] 288  
 toParameterName  
 [CIWConditionEvaluator.Parameter] 355  
 toParameterName  
 [CIWCreationRuleEvaluator.Parameter] 360  
 toParameterName  
 [CIWResourceSelector.Parameter] 365  
 toParameterName  
 [CIWWorkApplication.Parameter] 370  
 toShortState [CIWActivityInstance.State] 247  
 toShortState [CIWProcessDefinition.State] 264  
 toShortState [CIWProcessInstance.State] 278  
 toShortState [CIWWorkItem.State] 314  
 toState [CIWActivityInstance.State] 248  
 toState [CIWProcessDefinition.State] 265  
 toState [CIWProcessInstance.State] 278  
 toState [CIWWorkItem.State] 315  
 toStateCode [CIWActivityInstance.State] 248  
 toStateCode [CIWProcessDefinition.State]  
 265  
 toStateCode [CIWProcessInstance.State] 279  
 toStateCode [CIWWorkItem.State] 316  
 toString [CIWActivityInstance.State] 249  
 toString [CIWActivityInstance.Type] 254  
 toString [CIWProcessDefinition.State] 266  
 toString [CIWProcessInstance.State] 280  
 toString [CIWWorkDefinition.Type] 297  
 toString [CIWWorkItem.State] 316  
 toString [CIWWorkItem.Type] 321  
 toTargetkind [CIWServer.Targetkind] 284  
 toType [CIWActivityInstance.Type] 255  
 toType [CIWWorkDefinition.Type] 298  
 toType [CIWWorkItem.Type] 322  
 toTypeCode [CIWActivityInstance.Type] 255  
 toTypeCode [CIWWorkDefinition.Type] 299  
 toTypeCode [CIWWorkItem.Type] 323  
 Type 属性で指定できる値 393

## V

valueOf [CIWActivityDefinition.AttributeName]  
 232  
 valueOf [CIWActivityInstance.AttributeName]  
 237  
 valueOf [CIWActivityInstance.State] 250  
 valueOf [CIWActivityInstance.Type] 256  
 valueOf [CIWConditionEvaluator.Parameter]  
 356  
 valueOf  
 [CIWCreationRuleEvaluator.Parameter] 361  
 valueOf  
 [CIWProcessDefinition.AttributeName] 260  
 valueOf [CIWProcessDefinition.State] 266  
 valueOf [CIWProcessInstance.AttributeName]  
 271  
 valueOf [CIWProcessInstance.State] 280  
 valueOf [CIWResourceSelector.Parameter]  
 366  
 valueOf [CIWServer.Targetkind] 284  
 valueOf [CIWUtil.InfinityDate] 288  
 valueOf [CIWWorkApplication.Parameter]  
 371  
 valueOf [CIWWorkApplication.Result] 374  
 valueOf [CIWWorkDefinition.AttributeName]  
 292  
 valueOf [CIWWorkDefinition.Type] 299  
 valueOf [CIWWorkItem.AttributeName] 304  
 valueOf [CIWWorkItem.State] 317  
 valueOf [CIWWorkItem.Type] 323  
 valueOf で発生する例外  
 [CIWActivityDefinition.AttributeName] 233  
 valueOf で発生する例外  
 [CIWActivityInstance.AttributeName] 238  
 valueOf で発生する例外  
 [CIWActivityInstance.State] 251  
 valueOf で発生する例外  
 [CIWActivityInstance.Type] 257  
 valueOf で発生する例外  
 [CIWConditionEvaluator.Parameter] 357

valueOf で発生する例外  
 [CIWCreationRuleEvaluator.Parameter] 362

valueOf で発生する例外  
 [CIWProcessDefinition.AttributeName] 261

valueOf で発生する例外  
 [CIWProcessDefinition.State] 267

valueOf で発生する例外  
 [CIWProcessInstance.AttributeName] 272

valueOf で発生する例外  
 [CIWProcessInstance.State] 281

valueOf で発生する例外  
 [CIWResourceSelector.Parameter] 366

valueOf で発生する例外 [CIWServer.Targetkind]  
 285

valueOf で発生する例外 [CIWUtil.InfinityDate]  
 289

valueOf で発生する例外  
 [CIWWorkApplication.Parameter] 372

valueOf で発生する例外  
 [CIWWorkApplication.Result] 374

valueOf で発生する例外  
 [CIWWorkDefinition.AttributeName] 293

valueOf で発生する例外  
 [CIWWorkDefinition.Type] 300

valueOf で発生する例外  
 [CIWWorkItem.AttributeName] 305

valueOf で発生する例外 [CIWWorkItem.State]  
 318

valueOf で発生する例外 [CIWWorkItem.Type]  
 324

valueOf の引数  
 [CIWActivityDefinition.AttributeName] 233

valueOf の引数  
 [CIWActivityInstance.AttributeName] 238

valueOf の引数 [CIWActivityInstance.State] 250

valueOf の引数 [CIWActivityInstance.Type] 256

valueOf の引数  
 [CIWConditionEvaluator.Parameter] 356

valueOf の引数  
 [CIWCreationRuleEvaluator.Parameter] 361

valueOf の引数  
 [CIWProcessDefinition.AttributeName] 260

valueOf の引数 [CIWProcessDefinition.State]  
 267

valueOf の引数  
 [CIWProcessInstance.AttributeName] 271

valueOf の引数 [CIWProcessInstance.State]  
 281

valueOf の引数  
 [CIWResourceSelector.Parameter] 366

valueOf の引数 [CIWServer.Targetkind] 285

valueOf の引数 [CIWUtil.InfinityDate] 289

valueOf の引数  
 [CIWWorkApplication.Parameter] 371

valueOf の引数 [CIWWorkApplication.Result]  
 374

valueOf の引数  
 [CIWWorkDefinition.AttributeName] 293

valueOf の引数 [CIWWorkDefinition.Type] 300

valueOf の引数 [CIWWorkItem.AttributeName]  
 305

valueOf の引数 [CIWWorkItem.State] 317

valueOf の引数 [CIWWorkItem.Type] 324

values [CIWActivityDefinition.AttributeName]  
 233

values [CIWActivityInstance.AttributeName]  
 238

values [CIWActivityInstance.State] 251

values [CIWActivityInstance.Type] 257

values [CIWConditionEvaluator.Parameter]  
 357

values [CIWCreationRuleEvaluator.Parameter]  
 362

values [CIWProcessDefinition.AttributeName]  
 261

values [CIWProcessDefinition.State] 267

values [CIWProcessInstance.AttributeName]  
 272

values [CIWProcessInstance.State] 281

values [CIWResourceSelector.Parameter] 366

values [CIWServer.Targetkind] 285

values [CIWUtil.InfinityDate] 289

values [CIWWorkApplication.Parameter] 372



values [CIWorkApplication.Result] 374  
values [CIWorkDefinition.AttributeName] 293  
values [CIWorkDefinition.Type] 300  
values [CIWorkItem.AttributeName] 305  
values [CIWorkItem.State] 318  
values [CIWorkItem.Type] 324

## W

WorkCoordinator Definer で作成された定義ファイルの利用 411

## あ

案件実行・運用クラスおよびインターフェースが提供するメソッド 92  
案件実行・運用クラスおよびインターフェースの一覧 77  
案件実行・運用クラスおよびインターフェースの階層 83  
案件処理中のエラーの場合 339  
案件属性 388  
案件属性一覧 388  
案件属性の属性名の列挙型 269  
案件のインターフェース 133  
案件の状態 395  
案件の状態一覧 395  
案件の状態遷移 394  
案件の状態の列挙型 273  
案件の遷移 396  
案件の遷移一覧 396  
案件の遷移時に設定する属性一覧 397  
案件の投入 [CSCIW を利用した業務処理の実装] 45

## い

一時的な障害発生の場合 338  
一覧情報取得時の CSCIW 内での動作 62  
一覧情報取得時の最大取得件数の指定 62  
一覧情報取得時のソート条件 64  
一覧情報取得時のフィルター条件 63  
一覧情報の取得 61  
一覧情報の取得手順 61  
一覧情報の取得手順と基本動作 61

インターフェースの一覧 (パッケージ [jp.co.Hitachi.soft.csciw.callback]) 79  
インターフェースの一覧 (パッケージ [jp.co.Hitachi.soft.csciw]) 77  
インターフェースの階層 (パッケージ [jp.co.Hitachi.soft.csciw.callback]) 85  
インターフェースの階層 (パッケージ [jp.co.Hitachi.soft.csciw]) 83

## え

エラーが発生した場合のトランザクションの終了について [API 利用時の注意事項] 74

## お

オブジェクトの取得で指定できる値 65  
オブジェクトのリストの取得で指定できる値 65

## か

各種ユーティリティ機能を提供するクラス 225

## き

記述形式 87, 93, 221, 341  
記述形式 (パッケージ [jp.co.Hitachi.soft.csciw.callback]) 351, 377  
記述形式 (パッケージ [jp.co.Hitachi.soft.csciw]) 229, 327  
業務ステップ属性 389  
業務ステップ属性一覧 389  
業務ステップ属性の属性名の列挙型 235  
業務ステップ定義属性 387  
業務ステップ定義属性の属性名の列挙型 231  
業務ステップ定義のインターフェース 94  
業務ステップのインターフェース 98  
業務ステップの種類列挙型 252  
業務ステップの状態 399  
業務ステップの状態一覧 399  
業務ステップの状態遷移 398  
業務ステップの状態列挙型 240  
業務ステップの遷移 401  
業務ステップの遷移一覧 401

業務ステップの遷移時に設定する属性一覧 403  
業務プログラムで管理するトランザクション 70  
業務プログラムとデータベース間での文字コード 73  
業務プログラムに渡される情報 56  
業務プログラムのコーディング例 [CIWFactory オブジェクトの取得] 43  
業務プログラムのコーディング例 [CIWServer オブジェクトに対する Connection オブジェクトの関連づけ] 45  
業務プログラムのコーディング例 [CIWServer オブジェクトに対する Connection オブジェクトの関連づけの解除] 48  
業務プログラムのコーディング例 [CIWServer オブジェクトの生成 (コネクションと関連づける場合)] 44  
業務プログラムのコーディング例 [CIWServer オブジェクトの生成 (データソースと関連づける場合)] 44  
業務プログラムのコーディング例 [CSCIW の終了 (Java アプリケーションで使用する場合)] 48  
業務プログラムのコーディング例 [CSCIW の初期化] 43  
業務プログラムのコーディング例 [CSCIW を利用した業務処理の実装 (案件の投入)] 45  
業務プログラムのコーディング例 [CSCIW を利用した業務処理の実装 (作業一覧の取得)] 46  
業務プログラムのコーディング例 [CSCIW を利用した業務処理の実装 (作業の開始と終了)] 46  
業務プログラムのコンパイル 41  
業務プログラムの作成方法 39  
業務プログラムの処理の流れ 43

## <

組み込み変数 32  
クラスの一覧 77  
クラスの構成 81  
クラスパス 41

## こ

更新系 API 383  
更新系 API 一覧 383  
更新系 API で指定できる値 66

## さ

作業アプリケーション Java オブジェクトのインタフェース 348  
作業アプリケーションに渡される情報 58  
作業アプリケーションに渡されるパラメタの列挙型 368  
作業アプリケーションの業務処理結果の列挙型 373  
作業アプリケーションを作成するときの注意事項 59  
作業一覧の取得 [CSCIW を利用した業務処理の実装] 46  
作業属性 390  
作業属性一覧 390  
作業属性の属性名の列挙型 302  
作業定義属性 387  
作業定義属性一覧 387  
作業定義属性の属性名の列挙型 291  
作業定義のインタフェース 184  
作業定義の種類列挙型 295  
作業のインタフェース 190  
作業の開始と終了 [CSCIW を利用した業務処理の実装] 46  
作業の種類列挙型 319  
作業の状態 404  
作業の状態一覧 405  
作業の状態遷移 404  
作業の状態列挙型 307  
作業の遷移 407  
作業の遷移一覧 407  
作業の遷移時に設定する属性一覧 409  
作成時の注意事項 [Java オブジェクト] 52  
参照系 API 384  
参照系 API 一覧 385  
参照系 API で指定できる属性 386  
参照系 API の使用について 75

## し

指定できる属性一覧 386  
指定できる属性 [ソート条件] 64  
指定できる属性値 391

指定できる属性〔フィルター条件〕 63  
指定できるデータ 65  
指定方法〔ソート条件〕 64  
指定方法〔フィルター条件〕 63  
巡回部分を含むビジネスプロセス定義 34  
障害発生の場合 330  
条件 Java オブジェクトのインタフェース 342  
条件およびルールの評価 49  
条件定義, 生成ルール定義, または振り分けルール定義に指定できる種別 49  
条件に一致する作業の作業割り当てと着手をする場合の実装例 46  
条件に渡されるパラメタの列挙型 353  
条件またはルールの評価に渡される情報 57  
状態遷移モデル 394  
初期化・終了クラスが提供するメソッド 86  
初期化・終了クラスの一覧 77  
初期化・終了クラスの階層 83  
処理するオブジェクトが存在しない場合 328

## す

スレッドからの利用について〔API 利用時の注意事項〕 74

## せ

生成ルール Java オブジェクトのインタフェース 344  
生成ルールに渡されるパラメタの列挙型 358

## そ

属性値の指定方法〔フィルター条件〕 63  
その他の階層 83  
その他のクラス 220  
その他のクラス一覧 78  
その他のクラスで指定できる値 67

## ち

注意事項〔ソート条件〕 64  
注意事項〔フィルター条件〕 64

## て

提供 API 一覧 76  
データソースと関連づける場合〔CIWServer オブジェクトの生成〕 44  
データソースを使用した Connection オブジェクトの関連づけ〔業務プログラムで管理するトランザクション (CIWServer とコネクションの関連づけ)〕 71

## と

トランザクションの開始〔業務プログラムで管理するトランザクション〕 72  
トランザクションの終了〔業務プログラムで管理するトランザクション〕 72

## は

排他およびデッドロックについて〔API 利用時の注意事項〕 74  
パッケージのインポート 41  
パッケージ「jp.co.Hitachi.soft.csciw.callback」に所属するクラス 79  
パッケージ「jp.co.Hitachi.soft.csciw.callback」の階層 85  
パッケージ「jp.co.Hitachi.soft.csciw」に所属するクラス 77  
パッケージ「jp.co.Hitachi.soft.csciw」の階層 83

## ひ

ビジネスプロセス定義属性 386  
ビジネスプロセス定義属性一覧 386  
ビジネスプロセス定義属性の属性名の列挙型 258  
ビジネスプロセス定義のインタフェース 122  
ビジネスプロセス定義の状態の列挙型 262  
ビジネスプロセスの開発方法 21  
ビジネスプロセスの変更 25  
ビジネスプロセスを定義するときの注意事項 32

## ふ

不正な状態の場合 337  
振り分けルール Java オブジェクトのインタフェース 346  
振り分けルールに渡されるパラメタの列挙型 363

## む

無限遠の日付の列挙型 287

## れ

- 例外クラス (パッケージ「jp.co.Hitachi.soft.csciw.callback」) 376
- 例外クラス (パッケージ「jp.co.Hitachi.soft.csciw」) 326
- 例外クラスの一覧 (パッケージ「jp.co.Hitachi.soft.csciw.callback」) 80
- 例外クラスの一覧 (パッケージ「jp.co.Hitachi.soft.csciw」) 79
- 例外クラスの階層 (パッケージ「jp.co.Hitachi.soft.csciw.callback」) 85
- 例外クラスの階層 (パッケージ「jp.co.Hitachi.soft.csciw」) 84
- 例外の基底の抽象クラス 329
- 列挙型 (パッケージ「jp.co.Hitachi.soft.csciw.callback」) 350
- 列挙型 (パッケージ「jp.co.Hitachi.soft.csciw」) 228
- 列挙型の一覧 (パッケージ「jp.co.Hitachi.soft.csciw.callback」) 80
- 列挙型の一覧 (パッケージ「jp.co.Hitachi.soft.csciw」) 78
- 列挙型の階層 (パッケージ「jp.co.Hitachi.soft.csciw.callback」) 85
- 列挙型の階層 (パッケージ「jp.co.Hitachi.soft.csciw」) 83

## わ

- ワーク管理システムが管理するオブジェクトの種別を表す列挙型 283
- ワーク管理システム全般にかかわる処理を行うための機能を提供するインタフェース 156

---

 株式会社 日立製作所

〒100-8280 東京都千代田区丸の内一丁目6番6号

---