HITACHI
Inspire the Next

uCosminexus Service Coordinator Interactive Workflow システム構築・運用ガイド

3020-3-M80-A0

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 03-20 （適用 OS：Windows Server 2019，Windows Server 2022）

P-2955-CG34 uCosminexus Service Coordinator Interactive Workflow 03-20 （適用 OS：Windows Server 2019，Windows Server 2022，Windows 10 x64，Windows 11）

P-9W43-CG31 uCosminexus Service Coordinator Interactive Workflow 03-20 （適用 OS：Red Hat(R) Enterprise Linux(R) Server 7.1（64-bit x86_64）以降，Red Hat(R) Enterprise Linux(R) Server 8.1（64-bit x86_64）以降）

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。

Internet Explorer は，マイクロソフト 企業グループの商標です。

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。

Microsoft Edge は，マイクロソフト 企業グループの商標です。

Oracle および Java は，オラクルおよびその関連会社の登録商標です。

Red Hat is a registered trademark of Red Hat, Inc. in the United States and other countries.

Red Hat 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。

Red Hat Enterprise Linux is a registered trademark of Red Hat, Inc. in the United States and other countries.

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。
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

2023 年 3 月　3020-3-M80-A0

■ 著作権

All Rights Reserved. Copyright (C) 2007, 2023, Hitachi, Ltd.

変更内容

変更内容(3020-3-M80-A0)　uCosminexus Service Coordinator Interactive Workflow 03-20

追加・変更内容	変更個所
適用 OS に Windows Server 2022 を追加した。	－
Cosminexus に同梱されている JAX-RS ライブラリを使用する旨の注意事項を削除した。	4.5.3(1)(a)
Cosminexus に同梱されている JAX-RS ライブラリをコンテナ拡張ライブラリに設定しない場合の説明を追加した。	4.5.3(1)(b)
ciwstartpi コマンドの-n オプション指定時の注意事項を追加した。	8.2.1(1)
SQL スクリプトファイルの編集に「03-11 から 03-20 へ移行する場合」を追加しました。	9.2.3，9.3.5
ワーク管理システムのバージョンアップの手順に，データ移行だけを繰り返し行う手順を追加した。	9.3.8
テーブル<SYSTEMID>_ASYNC_JOB のジョブ ID の INDEX の値を修正した。	付録 B.1(1)(ah)

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 の概要およびシステムの構築・運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の CSCIW の構築・運用方法については，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。

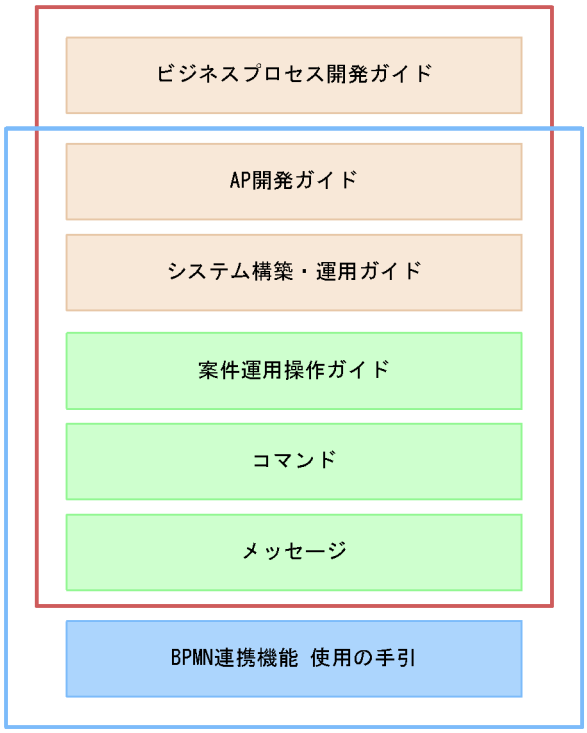
■ 対象読者

このマニュアルは，ワーク管理システムを初めて使用される方，または 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構築・運用す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

■ CSCIW マニュアル全体の読書手順

CSCIW のマニュアルは，BPMN 連携機能を使用するかどうかで読み方が異なります。BPMN 連携機能を使用する場合と使用しない場合に読むマニュアルの違いを，次の図で示します。



（凡例）

：BPMN連携機能を使用しないユーザが読むマニュアル
：BPMN連携機能を使用しないユーザが必ず読むマニュアル
：BPMN連携機能を使用するユーザが読むマニュアル
：BPMN連携機能を使用するユーザが必ず読むマニュアル
：必要に応じて読むマニュアル

BPMN 連携機能を使用する場合

マニュアル「uCosminexus Service Coordinator Interactive Workflow BPMN 連携機能　使用の手引」から読み進めてください。また，次のマニュアルについては，マニュアル「uCosminexus Service Coordinator Interactive Workflow BPMN 連携機能　使用の手引」から参照している個所だけお読みください。

- uCosminexus Service Coordinator Interactive Workflow　システム構築・運用ガイド
- uCosminexus Service Coordinator Interactive Workflow　AP 開発ガイド

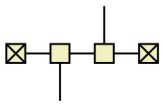
■ このマニュアルの読書手順

このマニュアルは，利用目的に合わせて直接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。

■ 図中で使用する記号

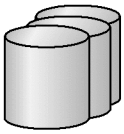
このマニュアルの図中で使用する記号を次のように定義します。

●バス型のＬＡＮ



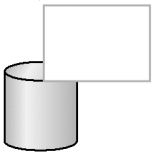
●データベース

●ファイル（複数）



●ファイルまたは
データベース

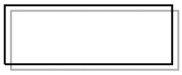
●ファイルまたは
データベースの内容



●工程，作業項目の
流れ

●定義ファイル

●プロセス



●ソースノード

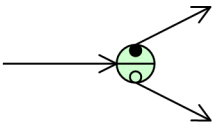
●シンクノード

●アロー

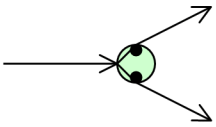
●業務ステップ

●作業

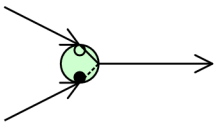
●制御ノード（分岐）



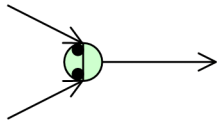
●制御ノード（分業）



●制御ノード（先着）



●制御ノード（待合）



■ 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。

記号	意味
{ }	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。 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。 • ボタン • キーボードのキー • 画面またはダイアログの名称 • 画面またはダイアログに表示される項目
[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　概要編

1　ワーク管理システムの概要　19
1.1　ワーク管理システムとは　20
1.1.1　ワーク管理システムの目的　20
1.1.2　従来のワークフローシステムとワーク管理システムの違い　20
1.1.3　ワーク管理システムの特長　21
1.1.4　ワーク管理システムの適用業務と適用方法　23
1.2　ワーク管理システムの機能　25
1.2.1　ワーク管理システムの全体像　25
1.2.2　各機能の概要　26
1.3　ワーク管理システムによる業務システム開発の流れ　27

2　ワーク管理システムの基礎知識　29
2.1　ワーク管理システムによる業務の流れ　30
2.2　ビジネスプロセス定義　32
2.2.1　ビジネスプロセス定義と案件の流れ　32
2.2.2　ビジネスプロセス定義の構成　34
2.2.3　ビジネスプロセス定義の作成・登録と削除　39
2.2.4　CSCIW 内でのビジネスプロセス定義の管理　40
2.3　振り分けルール定義　41
2.3.1　振り分けルール定義による作業者の特定　41
2.3.2　振り分けルール定義の構成　42
2.3.3　振り分けルール定義の作成・登録と削除　43
2.3.4　CSCIW 内での振り分けルール定義の管理　43
2.3.5　ユーザとユーザグループ　44
2.4　データ条件と業務データ　46
2.4.1　データ条件による案件遷移の制御　46
2.4.2　データ条件の構成　48
2.5　案件の基本的な進み方と動的制御　49
2.5.1　案件の基本的な進み方　49

2.5.2 案件の動的な制御 59
2.5.3 組み込み作業での案件の進み方 68
2.6 作業アプリケーションの呼び出し 71
2.6.1 機能および作業アプリケーション呼び出し時の動作 71
2.6.2 作業アプリケーションの同期呼び出し 72
2.6.3 作業アプリケーションの非同期呼び出し 74
2.7 RDB 検索 76
2.7.1 振り分けルール記述の形式 76
2.7.2 データ条件記述の形式 76
2.7.3 生成ルール記述の形式 76
2.8 Java オブジェクト呼び出し 78

第 2 編 構築・管理編

3 システム構成に関する前提知識 80
3.1 ワーク管理システムのプログラム構成 81
3.1.1 ワーク管理システムの前提プログラム 81
3.1.2 ワーク管理システムの関連プログラム 82
3.2 ワーク管理システムのシステム構成モデル 84
3.2.1 基本システム構成 84
3.2.2 マルチインスタンス構成 85
3.2.3 マルチマシン構成 86
3.2.4 マルチプロセス構成 86
3.2.5 クラスタ構成 87

4 ワーク管理システムの設定 89
4.1 システム設定の流れ 90
4.2 プログラムのインストール 91
4.2.1 前提プログラムのインストール 91
4.2.2 CSCIW のインストール 95
4.3 ワーク管理データベースの構築 98
4.3.1 事前準備 98
4.3.2 ワーク管理データベースの環境設定（HiRDB の場合） 99
4.3.3 ワーク管理データベースの環境設定（ORACLE の場合） 101
4.3.4 ワーク管理データベースの環境設定（SQL Server の場合） 104
4.4 開発環境での環境設定 107
4.5 実行環境での環境設定 108
4.5.1 事前準備 108
4.5.2 実行環境の構築 110
4.5.3 アプリケーションサーバの設定 113

4.5.4 Java アプリケーションの設定 131
4.5.5 Java オブジェクト呼び出しを利用する場合の設定 132
4.6 注意事項 136
4.6.1 マルチインスタンス構成を設定する場合の注意事項 136
4.6.2 マルチマシン構成を設定する場合の注意事項 136
4.6.3 マルチプロセス構成を設定する場合の注意事項 137
4.6.4 クラスタ構成を設定する場合の注意事項 138
4.6.5 業務プログラムの実行権限を設定する場合の注意事項 138

5 ワーク管理システムの削除 140
5.1 システム削除の流れ 141
5.2 実行環境での環境削除 142
5.2.1 事前準備 142
5.2.2 実行環境の削除 146
5.3 開発環境での環境削除 148
5.4 ワーク管理データベースの削除 149
5.4.1 ワーク管理データベースの環境削除（HiRDB の場合） 149
5.4.2 ワーク管理データベースの環境削除（ORACLE の場合） 151
5.4.3 ワーク管理データベースの環境削除（SQL Server の場合） 153
5.5 CSCIW のアンインストール 156
5.5.1 事前準備 156
5.5.2 OS からのアンインストール 156
5.6 注意事項 158
5.6.1 マルチインスタンス構成の環境を削除する場合の注意事項 158
5.6.2 マルチマシン構成の環境を削除する場合の注意事項 158

6 ワーク管理システムの変更 159
6.1 システムの構成変更 160
6.1.1 実行環境マシンの IP アドレスの変更 160
6.1.2 接続先 DBMS の変更 160
6.1.3 DBMS の格納領域の変更（HiRDB の場合） 161
6.1.4 DBMS の格納領域の変更（ORACLE の場合） 163
6.1.5 DBMS の格納領域の変更（SQL Server の場合） 165
6.2 実行環境の設定情報の変更 168
6.2.1 CSCIWManagementServer および CSCIW を使用している Java アプリケーションの再起動が必要なコマンド 168
6.2.2 環境変数の変更 169
6.2.3 システム設定プロパティファイルおよびセットアッププロパティファイルの変更 169
6.2.4 CSCIWManagementServer が使用するデータソース表示名の変更 170
6.2.5 案件運用操作が使用するデータソース表示名の変更 171

6.2.6 アプリケーション呼び出しサービスが使用するデータソース表示名の変更 172
6.2.7 Java オブジェクト呼び出しで呼び出す Java オブジェクトの変更 173
6.2.8 アプリケーション呼び出しサービスの設定の変更 174
6.3 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 176
6.3.1 変更後のビジネスプロセス定義および振り分けルール定義の情報反映 176
6.3.2 ビジネスプロセス定義削除時の注意事項 177

第 3 編 運用編

7 ワーク管理システムの運用 178
7.1 業務の開始と停止 179
7.1.1 業務の開始 179
7.1.2 業務の停止 180
7.2 ビジネスプロセス定義に関する運用 182
7.2.1 ビジネスプロセス定義のバージョン 182
7.2.2 ビジネスプロセス定義の状態と案件投入可能期間 182
7.2.3 ビジネスプロセス定義の登録 183
7.2.4 ビジネスプロセス定義の削除 184
7.3 振り分けルール定義に関する運用 185
7.3.1 振り分けルール定義の管理 185
7.3.2 振り分けルール定義の状態と有効期間 185
7.3.3 振り分けルール定義の登録 186
7.3.4 振り分けルール定義の削除 187
7.4 案件に関する運用 188
7.4.1 案件に対する運用 188
7.4.2 業務ステップに対する運用 190
7.4.3 作業に対する運用 192
7.5 監査ログの取得によるシステム監視 195
7.5.1 ワーク管理システムの監査対象 195
7.5.2 監査ログの出力ポイントと出力項目 197
7.6 バックアップとリストア 200
7.6.1 バックアップする内容 200
7.6.2 バックアップとリストアの流れ 201
7.7 データベースの再編成について 203
7.8 業務システムの変更に伴う運用 204
7.8.1 作業者を変更する場合 204
7.8.2 データ条件を変更する場合 205
7.8.3 ビジネスプロセスを変更する場合 207
7.8.4 データ条件評価や振り分けルール定義評価の RDB 検索に関連するデータを変更する場合 207

7.9 運用上の注意事項　208
7.9.1 CSCIWManagementServer の開始時期について　208
7.9.2 日付・時刻の整合性について　208
7.9.3 業務アプリケーションが異常終了した場合や強制終了した（終了処理をしなかった）場合の影響について　209
7.9.4 データベース使用時のセキュリティについて　210
7.9.5 Java オブジェクト呼び出し機能を使用した，Java オブジェクトの J2EE アプリケーションへの組み込みについて　211

8 障害対策　212
8.1 障害対策の流れ　213
8.2 障害情報の取得　214
8.2.1 CSCIW の出力情報の取得　214
8.2.2 Cosminexus 性能解析トレースの取得　221
8.2.3 snapshot ログの収集　228
8.2.4 DBMS での障害情報の取得　231
8.3 障害要因の特定　232
8.3.1 エラーメッセージの確認　232
8.3.2 ユーザの操作内容の確認　233
8.3.3 ワーク管理システムの運用状況の確認　233
8.3.4 案件の運用状況の確認　233
8.4 障害への対処　235
8.5 障害の回復　236
8.5.1 CSCIW の障害時の回復方法　236
8.5.2 ワーク管理データベースの障害時の回復方法　236
8.6 トラブルシューティング　237
8.6.1 特定の案件が推進されない　237
8.6.2 すべての案件が推進されない　237
8.6.3 CSCIWManagementServer または業務アプリケーションが起動できない　238
8.7 JP1 によるログファイルの監視　239

9 ワーク管理システムのバージョンアップ　240
9.1 バージョンアップの流れ　241
9.2 ワーク管理システムのバージョンアップ（既存のワーク管理システムを引き継ぐ場合）　244
9.2.1 CSCIW の停止および削除　244
9.2.2 CSCIW のインストール（上書きインストール）　246
9.2.3 ワーク管理データベースのバージョンアップ　246
9.2.4 セットアップコマンドによるバージョンアップ　253
9.2.5 Cosminexus の設定　254
9.3 ワーク管理システムのバージョンアップ（ワーク管理システムを新規に構築する場合）　255

9.3.1 移行元環境の CSCIW の停止 256
9.3.2 CSCIW のインストール（新規インストール） 256
9.3.3 ワーク管理データベースの構築 257
9.3.4 ワーク管理データベースのデータ移行 257
9.3.5 ワーク管理データベースのバージョンアップ 257
9.3.6 セットアップコマンドによるバージョンアップ 259
9.3.7 Cosminexus の設定 259
9.3.8 データ移行だけを繰り返し行う手順 260

付録 262
付録 A 環境設定ファイル 263
付録 A.1 システム設定プロパティファイル 263
付録 A.2 セットアッププロパティファイル 264
付録 A.3 コマンド用環境設定ファイル 278
付録 B テーブル容量の見積もり 281
付録 B.1 テーブルおよびインデクス定義 281
付録 B.2 レコード数の概算式 313
付録 C トレース容量の見積もり 318
付録 D キャッシュする ID 範囲の見積もり 321
付録 E ciwdelpi コマンド実行時の排他要求数の見積もり（HiRDB 限定） 322
付録 F 監査ログの詳細 325
付録 F.1 CSCIWManagementServer に関係する監査ログ 325
付録 F.2 コマンドに関係する監査ログ 331
付録 F.3 案件運用操作に関係する監査ログ 345
付録 G ビジネスプロセス定義および振り分けルール定義ファイル 357
付録 G.1 CSCIW で登録できるビジネスプロセス定義の機能 357
付録 G.2 CSCIW で登録できる振り分けルール定義の機能 359
付録 H BPMN 連携機能を使用しない場合のアプリケーション呼び出しサービス 360
付録 H.1 アプリケーション呼び出しまでの処理の流れ 360
付録 H.2 実行間隔とポーリング間隔 362
付録 H.3 呼び出しの流量制御 364
付録 H.4 障害時の動作 365
付録 H.5 呼び出し処理のタイムアウト 368
付録 H.6 呼び出しの一時抑止 369
付録 H.7 使用上の注意事項 371
付録 H.8 ビジネスプロセスの開発 372
付録 H.9 REST アプリケーションの開発 375
付録 H.10 アプリケーション呼び出しサービスの動作を制御する設定項目 384
付録 H.11 共通設定ファイル 385

付録 H.12　アプリケーション呼び出し情報ファイル　392
付録 H.13　リトライの対象から外れた作業に関する運用　402
付録 H.14　チューニング　403
付録 I　REST API による運用管理の操作　404
付録 I.1　運用管理 REST API を使用してできる操作　404
付録 I.2　ジョブを使用した非同期処理の仕組み　405
付録 I.3　運用管理 REST サービス使用時の事前準備　409
付録 I.4　運用管理 REST サービスの設定　410
付録 I.5　運用管理 REST サービスの設定の変更　411
付録 I.6　運用管理 REST サービスの停止および削除　413
付録 I.7　運用管理 REST サービスの開始　413
付録 I.8　運用管理 REST API の構成要素　413
付録 I.9　運用管理 REST API 一覧　417
付録 I.10　運用管理 REST API の記述形式　418
付録 I.11　リクエストパラメタにファイルを指定する方法　418
付録 I.12　XML スキーマファイル　420
付録 I.13　各運用管理 REST API の詳細　421
付録 I.14　投入するジョブの機能ごとの指定値と戻り値　441
付録 J　J2EE サーバを V9 互換モードで使用する場合の設定　444
付録 J.1　互換モード用のファイルへの入れ替え　444
付録 J.2　コンテナ拡張ライブラリに取り込むライブラリの変更　444
付録 K　インストールディレクトリ（フォルダ）　446
付録 L　各バージョンの変更内容　447
付録 L.1　03-11 の変更内容　447
付録 L.2　03-10 の変更内容　447
付録 L.3　03-00 の変更内容　448
付録 L.4　02-30 の変更内容　450
付録 L.5　02-20 の変更内容　451
付録 L.6　02-10 の変更内容　452
付録 L.7　02-00 の変更内容　452
付録 L.8　01-95 の変更内容　453
付録 L.9　01-90 の変更内容　453
付録 L.10　01-70 の変更内容　454
付録 L.11　01-60 の変更内容　454
付録 M　このマニュアルの参考情報　456
付録 M.1　関連マニュアル　456
付録 M.2　このマニュアルでの表記　457
付録 M.3　マイクロソフト製品の表記について　458
付録 M.4　このマニュアルで使用している略語　459

付録 M.5　KB（キロバイト）などの単位表記について　459
付録 N　用語解説　460

索引　468

1 ワーク管理システムの概要

この章では，ワーク管理システムの概要，機能，および CSCIW を使用して構築できるワーク管理システムの全体像について説明します。

1.1　ワーク管理システムとは

ワーク管理システム（CSCIW）とは，ワークフロー型汎用アプリケーションの構築基盤です。

この節では，ワーク管理システムの目的，特長および適用業務について説明します。

1.1.1　ワーク管理システムの目的

一般にワークフローシステムは，「業務システム内の流れの制御を処理本体から切り離し，業務システムの柔軟性や生産性を高める」という利点を持っています。この利点に加え，ワーク管理システムでは，「企業内の多様なシステムをわたって進行するような業務」や「手順をあらかじめ規定しにくいような業務」をワークフローのように管理することを主な目的としています。これによって，企業内の業務システム全体を，変化に即応できる柔軟な構造にしていくことができます。

1.1.2　従来のワークフローシステムとワーク管理システムの違い

企業内の業務をシステムとしてとらえた場合，その仕様の変化はさまざまな形で発生します。業務仕様の変化を大別すると，「流れの変化」と「処理の変化」とに分けられます。このうち「流れの変化」は，組織変更のたびに伝票の回付ルートが変わるなど，頻繁に発生します。また，商品の種類ごとに処理の流れが異なるといった多様性を持っています。

したがって，アプリケーションプログラム内に「流れの制御」と「処理本体」のコーディングが混在している場合，頻繁に発生する業務仕様の変化に伴って，多くのプログラムを見直す必要があります。その結果，業務仕様の変化に対するシステムの対応が遅れることになります。混在の量が多いほど，また混在のしかたが無秩序であるほど，システムは硬直化し，小さな変化に対してもその見直しに多大なコストが掛かります。

こうした問題点を解決するための手段としてワークフローシステムがあります。一般にワークフローシステムは，業務の流れを**ビジネスプロセス**として定義し，それに従って個々の業務処理を進行させるシステムです。ワークフローシステムを利用することで，業務の流れ（ビジネスプロセス）と業務の処理本体（アプリケーションプログラム）とが明確に区別され，アプリケーションプログラムの処理から「流れの制御」に関するコーディングを排除できます。その結果，システムの硬直化を防止したり，改善したりできます。

また，ワークフローシステムは，「システムの簡易開発ツール」という側面も持っています。ワークフローシステムの「流れの制御を切り離す」という特長と，簡易言語，フォームなどの「プログラムの簡易開発ツール」とを組み合わせることで，効率の良いシステム構築を実現できます。

従来のワークフローシステムは，このような「システムの簡易開発ツール」として発展してきたため，適所に応用することで高い生産性を発揮します。反面，生産性を向上するために，ツールとして特定の業務タイプに特化する必要があり，結果的に応用範囲が局所化されてしまいます。そのため，「多様な業務タイプを広く扱うような業務」や「手順が不確定な業務」への適用は不得意でした。

ワーク管理システムは，「個々の業務タイプごとではなく，企業の業務システム全体にワークフロー的な仕組みを適用したい」，「手順をあらかじめ規定できないような業務にもワークフロー的な仕組みを適用したい」といった要求にこたえることを目的としています。ワーク管理システムを利用することで，企業内のさまざまなアプリケーション，業務システム，業務データなどを，柔軟で自由度の高いビジネスプロセス定義によって連携できます。これによって，企業内の多様なシステムをわたって進行するような業務をワークフロー的に管理できます。その結果，企業内の業務システム全体が，変化に即応できる柔軟な構造となります。

このように，ワーク管理システムは，従来のワークフロー製品と異なる「新しいタイプの汎用アプリケーション構築基盤」として開発されました。

1.1.3　ワーク管理システムの特長

(1)　高いオープン性による異種システムの統合

ワーク管理システムは，企業内の多様なシステムをわたって進行するような業務をワークフロー的に管理するために，高いオープン性を備えています。

一般的に企業内では，OA 系システム，C/S 系システム，ホスト系システムなど，さまざまな種類のシステムが運用されています。そのため，1 つの業務を複数のシステムにわたって処理することがあります。例えば，「発注」という業務では，OA 系システムでバイヤーが需要を予測し，C/S 系システムで発注スケジュールの策定や発注処理をし，ホスト系システムに仕入・買掛登録をするといった流れが考えられます。

このような「1 つの業務を複数のシステムにわたって処理する」という傾向は，業務変化に応じてシステムの部分更新を重ねるごとに強まっていきます。その結果，業務の全体状況が迅速に把握できないといった問題が発生します。

ワーク管理システムでは，多様なシステムの連携によって実現するような業務をワークフロー的に管理するため，次の標準的なインタフェースを採用しています。

●アプリケーション構築基盤として JavaEE の採用

アプリケーション構築基盤として，標準仕様である JavaEE に従う JavaEE アプリケーションサーバを採用しています。

●データアクセスインタフェースとしての RDB の採用

案件推進に必要な各種データは，ワーク管理システム外部の RDB を直接参照して取得します。具体的には，案件の制御条件を判定するためのデータや，案件の作業担当者を選択するデータとして，ワーク管理システム外部の業務データ（RDB）を利用します。

上記の標準的なインタフェースを備えることで，既存のシステムを最大限に生かしたまま，ワークフロー型業務システムを構築できます。また，システムの拡張や分散化にも柔軟かつ迅速に対応できます。

このように，ワーク管理システムは，企業内で運用されているさまざまなシステムをつなぎ合わせ，1 つのワークフロー型業務システムとして統合する役割を果たします。

(2) 高度なビジネスプロセス表現能力による適用業務の拡大

ワーク管理システムは，さまざまなタイプの業務をワークフロー的に管理するために，高度なビジネスプロセス表現能力を備えています。

従来のワークフローシステムの多くは，手続き的なビジネスプロセスの表現方式を採用しています。これは，作業の順序や分岐条件を明確に事前定義し，1 つの業務を規格化された作業手続きとして表現するものです。

一方，実際の業務には，手続き的に表現できない要素を含むものがあります。典型的な例としては，プロジェクト管理のような非定型業務が挙げられます。また，定型業務でも，現場が臨機応変に作業手順を変更し，運用面での効率向上を図る場合があります。

ワーク管理システムは，多様な業務タイプに適用するため，より表現力の高い**状態遷移型モデル**をビジネスプロセスの表現方式として採用しています。ワーク管理システムの状態遷移型モデルでは，案件の進行を業務状態（業務ステップ）の進行ととらえ，各状態に対して複数の作業を割り当てます。これは，「業務の状態の進行に応じて，実施できる作業群が入れ替わっていく」プロセスの表現だと言えます。

この表現方式では，業務タイプに合わせて次のような使い分けができます。

●あらかじめ作業手順を規定できる業務の場合

業務状態（業務ステップ）を細かく設定し，業務ステップと作業を 1 対 1 にすることで，作業手順を規定した手続き的なプロセス表現になります。

●作業手順を規定しにくい業務の場合

業務状態（業務ステップ）を大きく設定し，業務ステップと作業を 1 対複数にすることで，「ある業務状態での作業の順番は運用に任せる」という柔軟なプロセス表現になります。

これに加えて，ワーク管理システムでは，従来のワークフロー製品にない，次のような実行時制御機能を提供しています。

●各種組み込み作業

ビジネスプロセスに組み込み作業を定義することで，案件の実行時に業務データを参照し，作業を動的に生成したり，その状態を制御したりできます。

上記の機能を利用することで，「状況に応じて細かい調整が必要な処理」など，事前定義が困難な業務タイプにも対応できます。

このように，ワーク管理システムは，従来のワークフローシステムでは適用が困難だった業務を含め，企業内のさまざまなタイプの業務に適用できるような機能を備えています。

(3) 変化に強い柔軟なシステム構造と運用機能

ワーク管理システムは，企業内のシステム全体を変化に強い構造にすることを目的としています。そのためには，ワーク管理システム自体の構造や機能が，変化に強いものであることが必要です。

ワーク管理システムは，状態遷移の表現方法や実行時のプロセス制御機能によって，事前定義しておくビジネスプロセスの細かさを調整できます。そのため，さまざまな業務タイプや業務上の例外を吸収しやすくなっています。つまり，ワーク管理システムのビジネスプロセス表現は，変化に対する強さを持っていると言えます。

これに加えて，ワーク管理システムは，次のような変化対応の仕組みを備えています。

●業務データの外付け構造

ワーク管理システムは，案件の流れを制御したり，作業担当者を選択したりするデータを内部には持たないで，外部の業務データを参照する構造となっています。そのため，組織変更や人事異動などが発生した場合でも，組織や社員を管理しているデータベースを修正するだけで，ワーク管理システム側に変更が反映されます。ビジネスプロセス定義などを変更する必要はありません。

●ビジネスプロセス定義の部分変更機能

ワーク管理システムでは，ビジネスプロセス定義の運用段階に次のような変更を加えることができます。

ワーク管理システムでは，ビジネスプロセス定義の運用段階で分岐条件や案件の制御条件を変更できます。

このような「運用段階の変更」は，従来のワークフローシステムの多くで制限となっていた処理です。ワーク管理システムでは，コマンドを使用して上記の処理を実行できます。

このように，ワーク管理システムは，業務内容の変更，組織の変更などに対して，柔軟に対応できる構造や機能を備えています。したがって，ワーク管理システムを利用して構築された業務システムは，変化に強い構造を備えることができます。

1.1.4　ワーク管理システムの適用業務と適用方法

(1)　ワーク管理システムの適用業務

ワーク管理システムは，「さまざまなタイプの業務に対して，広くワークフロー的な仕組みを適用する」ことを目的としています。そのため，業務タイプには特化していません。しかし，従来のワークフローシステムと比較した場合，次のような業務への適用が有効です。

- 複数の部門にわたって進行する業務
- 状況に応じて，作業内容や手順を相互調整する業務
- 作業内容や手順が一定でない業務
- 案件の運用期間が長い業務

具体的には，販売，仕入，生産管理など，基幹業務の多くが該当します。

(2)　ワーク管理システムの適用方法

ワーク管理システムの適用方法には，次の 2 種類があります。

●業務状態の管理モニタとしての適用方法

「案件の状態を監視する」目的で，ワーク管理システムを適用する方法です。ワーク管理システムの最も基礎的な適用方法です。

この方法を適用する場合，業務状態（業務ステップ）を大きく設定し，全体の業務状態をさまざまな観点から監視できるようにします。監視できる情報には，「個別案件の進捗度」，「状態別の案件数（滞留数）」，「特定の案件群を対象とした，進捗度および状態別の案件数（滞留数）」などがあり，これらの情報はワーク管理システムから取得します。

この方法は，「複数のシステムにわたって広範囲に進行するような業務」などに適用します。

●ワークフローシステムとしての適用方法

「業務システムを通常のワークフローシステムとして構築する」目的で，ワーク管理システムを適用する方法です。この方法を適用する場合，必要に応じて細かな業務状態（業務ステップ）を設定し，各業務ステップの作業者に対応したフロー制御をします。

この方法は，「基幹業務など，既存のワークフローシステムを適用しにくい業務をワークフロー化する場合」などに適用します。

実際に，ワーク管理システムを利用して業務システムを構築する場合は，2 種類の適用方法を次のように組み合わせます。

1. 業務システムの開発範囲を次のように切り分けます。
   - ワーク管理システムでワークフロー化する部分
   - ほかのツールでワークフロー化する部分
   - ワークフロー以外のシステムで開発する部分

   ワーク管理システムでワークフロー化する部分は，「ワークフローシステムとしての適用方法」によって開発します。
2. ビジネスプロセスとして制御しないで，業務状態だけを管理する場合は，監視対象となる範囲を切り出します。

   業務状態だけを管理する部分は，「業務状態の管理モニタとしての適用方法」によって開発します。

このように，さまざまな業務レベルに対してワーク管理システムを適用することで，企業内の業務システムを統一的な考え方で設計できます。その結果，企業内の業務システム全体を，変化に即応できる柔軟な構造にしていくことができます。

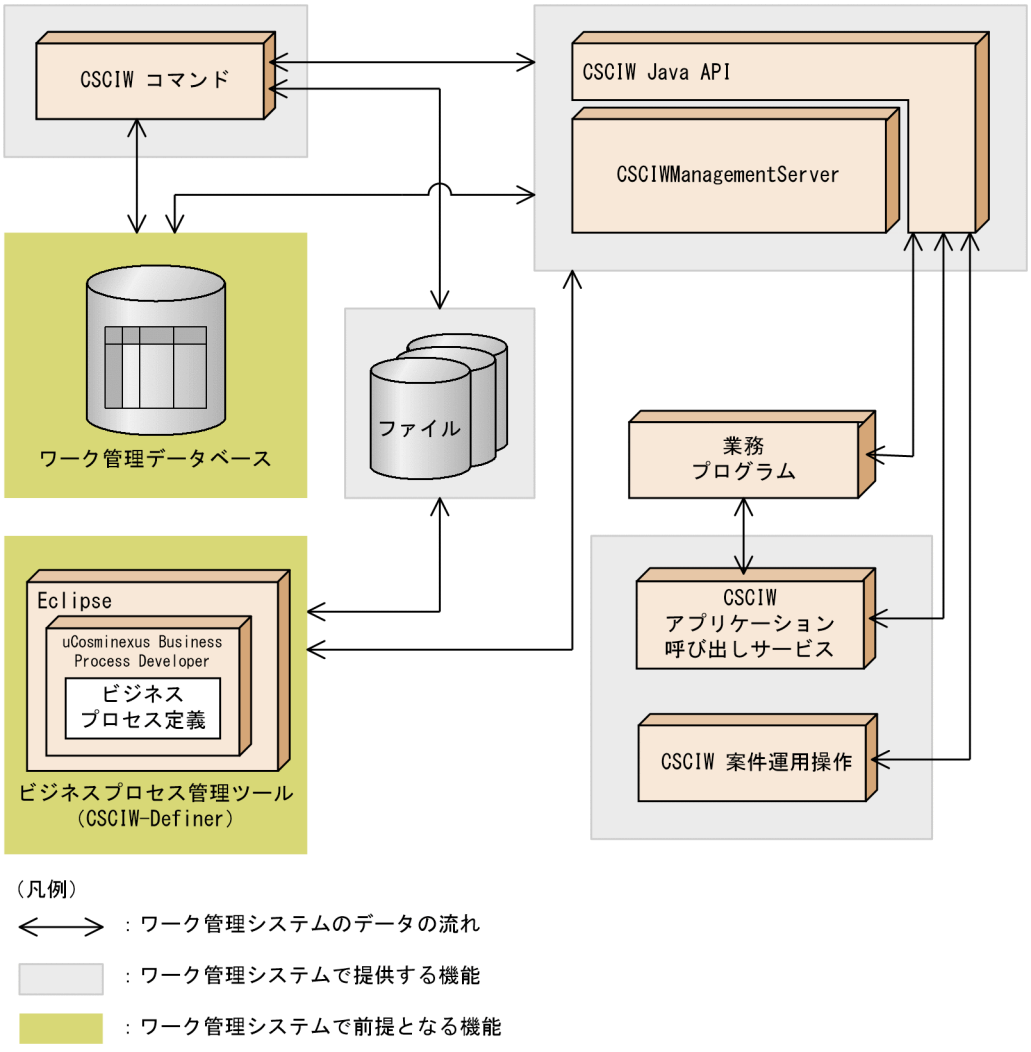
1.2　ワーク管理システムの機能

この節では，ワーク管理システムが提供する機能について説明します。

1.2.1　ワーク管理システムの全体像

ワーク管理システムの全体像を次の図に示します。

図 1-1　ワーク管理システムの全体像



ヒント

BPMN 連携機能を使用する場合の CSCIW のワーク管理システムの全体像については，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。

1.2.2　各機能の概要

ここでは，「1.2.1　ワーク管理システムの全体像」で示したワーク管理システム全体像の各機能の概要について説明します。

(1)　業務プログラム開発のための開発環境（CSCIW Java API）

ワーク管理システムで使用する業務プログラムを開発するために API を提供しています。ワーク管理システムで提供されている API を利用することで，業務プログラムを Java 言語で開発できます。

(2)　CSCIW の運用を支援する機能（CSCIWManagementServer）

アプリケーションサーバ（Cosminexus）での CSCIW の初期化，終了処理などの機能を J2EE アプリケーションとして提供しています。CSCIW を使用した J2EE アプリケーション，アプリケーション呼び出しサービス，または案件運用操作は，CSCIWManagementServer を開始してから実行します。

(3)　作業アプリケーションの非同期呼び出し機能（CSCIW アプリケーション呼び出しサービス）

API と非同期に作業アプリケーションの呼び出しを行う機能を J2EE アプリケーションとして提供しています。

(4)　案件の運用・管理を支援する機能（CSCIW 案件運用操作）

案件の状態を監視し，案件の運用や管理を支援する機能です。案件の運用状況の確認，案件の一時停止，強制終了，削除などを実行できます。

(5)　ワーク管理システムの導入と運用を支援するコマンド（CSCIW コマンド）

ワーク管理システムの環境設定，変更，削除などの機能は，コマンドとして提供しています。また，ビジネスプロセス管理ツール（CSCIW-Definer）で定義した**ビジネスプロセス**を登録・管理する機能，ビジネスプロセスで使用する**振り分けルール**（作業者を決定するためのルール）を登録・管理する機能，案件を削除する機能も，コマンドとして提供しています。

(6)　ビジネスプロセスの定義・運用を支援するツール（ビジネスプロセス管理ツール（CSCIW-Definer））

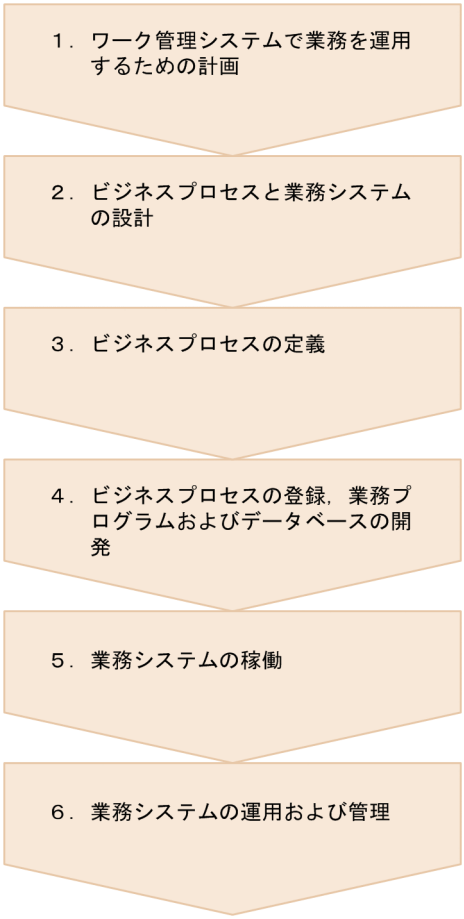
ワーク管理システムで使用するビジネスプロセスの定義・運用を支援するツールです。ビジネスプロセス定義および振り分けルール定義を定義し，CSCIW に登録できます。また，CSSCIW に登録したビジネスプロセス定義および振り分けルール定義に対する運用操作（属性変更，削除など）も実行できます。

1.3　ワーク管理システムによる業務システム開発の流れ

この節では，ワーク管理システムを利用した業務システム開発の流れについて，ワーク管理システムに関連する部分を中心に説明します。

一般的に，ワーク管理システムを利用した業務システムは，次のような流れに沿って開発します。

図 1-2　ワーク管理システムによる業務システム開発の流れ



1. ワーク管理システムで業務を運用するための計画

主に次のような作業をします。

●対象となる業務範囲と適用方法の明確化

現行の業務の問題点や改善点について分析します。そして，ワーク管理システムの適用対象として検討する業務について，その目的と範囲，対象となる組織構造を確認します。また，ワーク管理システムの適用方法を決定します。

●対象業務内容の共通認識の確認

業務フロー図などを作成し，現行業務の基本的な内容を理解します。各業務で実際に使用されているドキュメント類の調査も，業務内容を理解するために有効です。

また，システム化の具体的な目標を明確にするため，業務上発生している問題を収集します。収集した情報を基に，各問題点がどのように関連しているかを検討し，問題解決のための業務課題を設定します。

2. ビジネスプロセスと業務システムの設計

●案件の定義およびビジネスプロセスの切り出し

システム化する業務が対象とするものを案件として定義します。具体例としては，販売業務での「商談」，生産業務での「製造ロット」などがそれぞれ案件となります。案件に対する一連の業務処理群が，1 つのビジネスプロセスの単位となります。

●ビジネスプロセスを構成する業務状態の定義

各ビジネスプロセスをどのような業務状態（業務ステップ）に分解できるか， どのように遷移させるか，どのような条件で開始・終了をするかを明確にします。

●業務状態ごとに実行する作業の定義

業務状態（業務ステップ）ごとに実行する作業を明確にします。ここでは，作業間の順序性など，作業を実行する上で必要となる条件も明確にします。

●作業者を特定するための定義

作業を処理する担当者（作業者）を割り当てるルール（振り分けルール）を明確にします。

●業務プログラムおよび使用する業務データモデルの設計

ワーク管理システムと連携するユーザプログラムの単位を定義します。また，ワーク管理システムの制御に必要なデータ項目を明確にします。これらの結果を基に，個々のユーザプログラムとデータベースを設計します。

3. ビジネスプロセスの定義

2.で明確にした業務ステップ，作業，および振り分けルールを基に，ビジネスプロセスを定義します。定義には，ビジネスプロセス管理ツール（CSCIW-Definer）を使用します。ビジネスプロセスの定義方法については，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。

4. ビジネスプロセスの登録，業務プログラムおよびデータベースの開発

3.で定義したビジネスプロセスをワーク管理システムに登録します。登録には，コマンド，またはビジネスプロセス管理ツール（CSCIW-Definer）を使用します。また，システムに必要な業務プログラムやデータベースを開発し，テストします。

5. 業務システムの稼働

4.で作成したワーク管理システムの環境下で，業務を稼働させます。実際に案件が投入されたり，推進されたりします。

6. 業務システムの運用および管理

CSCIW の案件運用操作またはコマンドで，業務システムを運用および管理します。

2 ワーク管理システムの基礎知識

この章では，ワーク管理システムを使用するための基礎知識として，ビジネスプロセス定義の概要や仕組みを中心に説明します。また，ビジネスプロセス定義に沿って，案件がどのように推進されていくかについても説明します。

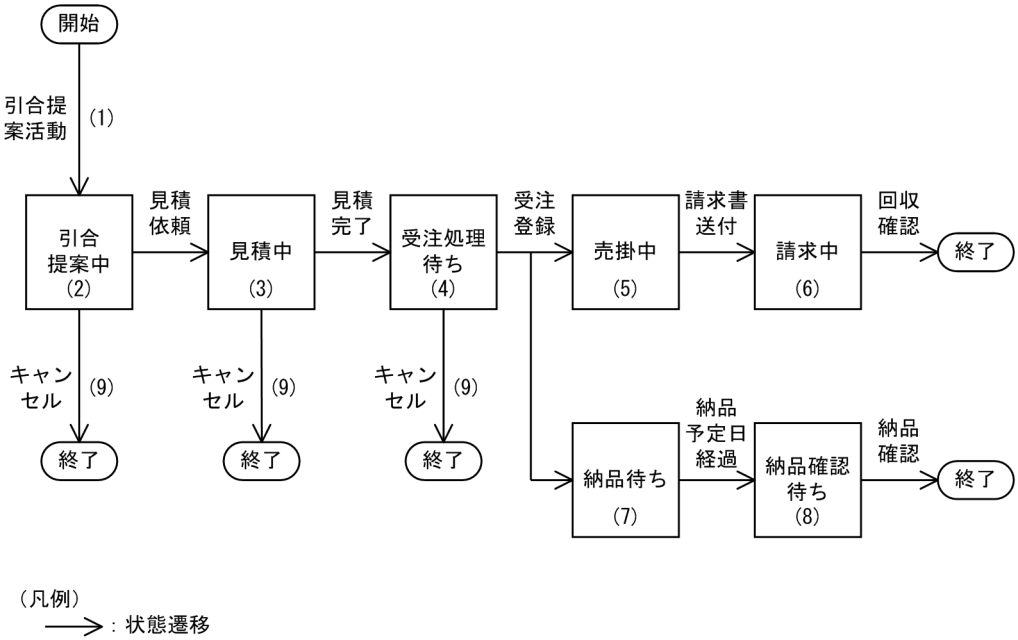
2.1　ワーク管理システムによる業務の流れ

この節では，販売業務での引合～納品・回収という一連の業務を管理する場合を例に，ワーク管理システムによる業務の流れについて説明します。

ワーク管理システムで業務を処理する場合，業務の流れをあらかじめ**ビジネスプロセス**として定義しておきます。定義されたビジネスプロセスに従って業務を進めていきます。

販売業務での引合～納品・回収の流れを例に，ビジネスプロセスと実際の業務がどのように流れているかを次の図に示します。

図 2-1　ビジネスプロセスと業務の流れ



上記のビジネスプロセスでは，次のような業務の流れを示しています。図中の括弧で示した番号が，各手順の番号に対応しています。

1. 引合の提案者は，引合があった商談を，引合～納品・回収ビジネスプロセスに投入します。ここからビジネスプロセスが開始されます。
2. 引合提案が開始されると「引合提案中」の状態になります。ここでは，顧客からの引合に対応して，ニーズを把握し，その内容の提案書を作成します。そして，提案書を顧客へ提出するとともに商談を進めます。
3. 提案書を見た顧客から正式な見積もりの依頼を受けると，「見積中」になります。ここでは，依頼に対して，納期や価格を見積もります。見積もりに必要な書類の作成は，見積書を作成する業務プログラムで実行します。
4. 見積書を提出し，見積交渉が完了すると，「受注処理待ち」となり，顧客から正式な注文連絡を待ちます。顧客から注文書を受け取った時点で，正式に受注を受け付けます。また，注文書に基づいて，受注票を起票します。さらに，受注票に基づいて，受注処理をします。これらの処理は，受注登録の業務プログラムで実行します。

5. 受注登録されたものは，「売掛中」となり，経理担当が業務プログラムを使って請求書を作成します。請求番号を入力した時点で，担当営業が請求内容を確認の上，請求書を顧客へ送付します。
6. 請求書が顧客へ送付されると，「請求中」となります。ここでは，顧客の入金状況を確認し，入金のあったものについて，完了処理をします。これらの処理を業務プログラムで実行します。
7. 受注登録と並行して，担当営業が納品のために作業指示し，その指示内容に従って，出荷・設置の各作業を実施します。ここでは，「納品待ち」になります。
8. 納品予定日の経過後，「納品確認待ち」となります。ここでは，顧客受領書を確認します。
9. 商談によっては，「引合提案中」，「見積中」または「受注処理待ち」の時点で，キャンセルされ終了します。

なお，実際の作業では，「見積もりの完了後，顧客交渉の結果によって見積もりを分割する」，「商品納入後，受注数量を変更する」などの例外的な処理が発生する場合があります。こうした例外的な処理については，一般的にビジネスプロセスで定義しません。ワーク管理システムの API を利用した，業務プログラムで実行します。

例外的な処理をどこまでビジネスプロセスで定義するのか，また，どの部分を業務プログラムで制御するのかは，業務システムの開発時に検討する必要があります。

2.2　ビジネスプロセス定義

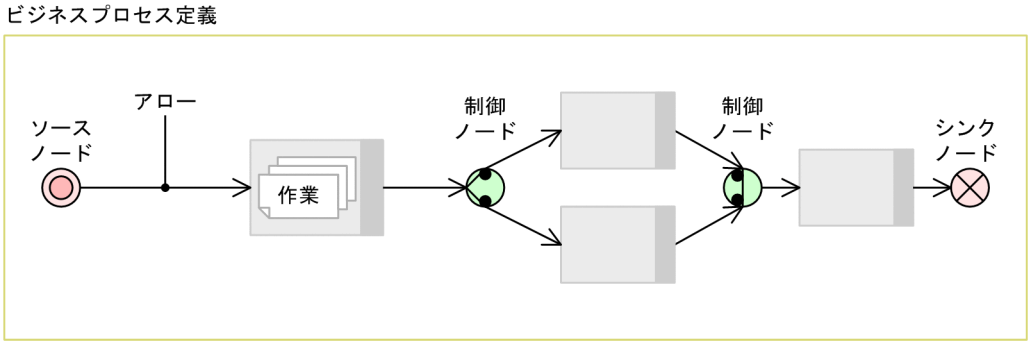
この節では，ビジネスプロセス定義の概要について説明します。

2.2.1　ビジネスプロセス定義と案件の流れ

ワーク管理システムを使用した業務システムでは，定義された流れに従って業務を進めていきます。業務の流れを定義したものを**ビジネスプロセス定義**といいます。

ワーク管理システムを使用したビジネスプロセスの定義例を次の図に示します。

図 2-2　ワーク管理システムを使用したビジネスプロセスの定義例



実際の業務では，ビジネスプロセス定義に従って案件を生成します。生成された**案件**が，実際に処理される業務の実体となります。

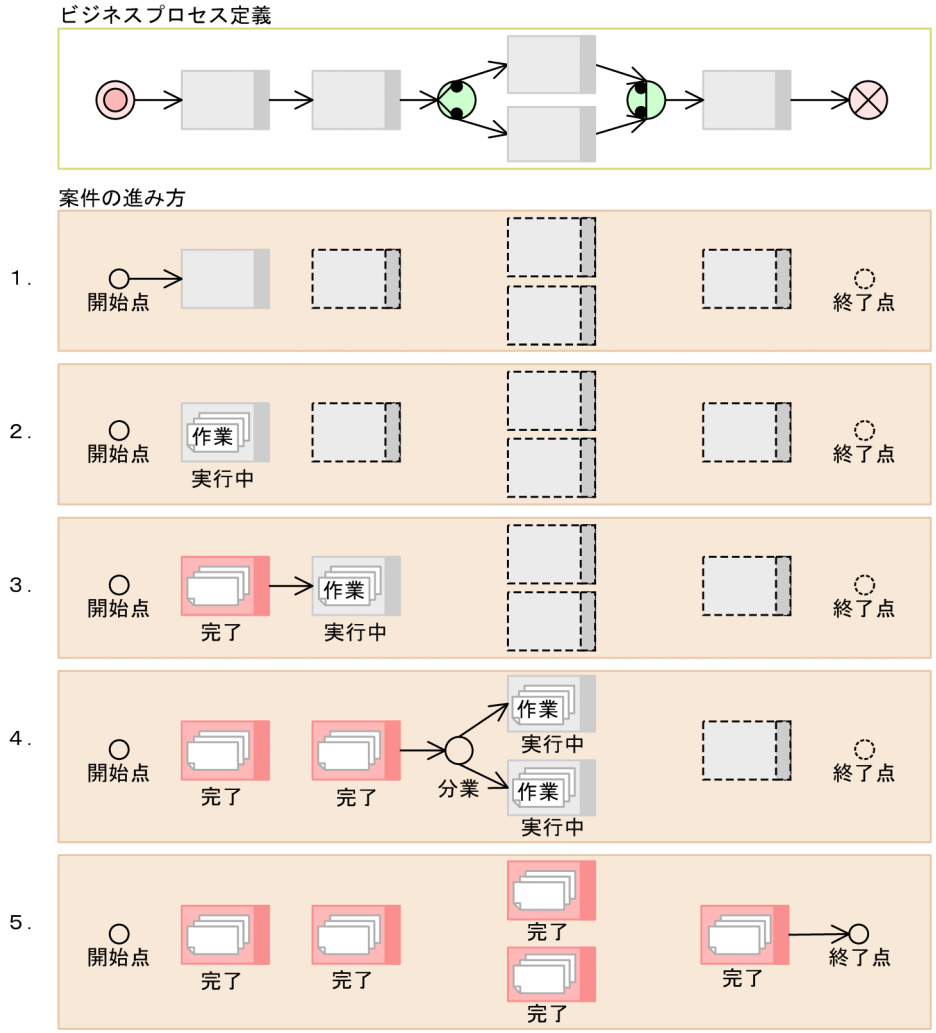
生成された案件には，ワーク管理システムによって識別子（案件 ID）が付けられます。案件 ID によって，各案件が一意に識別されます。

また，各案件には，ユーザが案件を識別するための情報として案件名を設定できます。一般的に案件名には，業務データベース内でその案件を特定するためのキーを設定します。例えば，業務データベース内の「商談番号」や「受注番号」など，案件と 1 対 1 で対応するキーを案件名にするといった運用が考えられます。

案件は，ビジネスプロセス定義に記述された手順で処理されていきます。

案件の基本的な進み方を次の図に示します。

図 2-3　案件の基本的な進み方



＜説明＞

1. ユーザまたは業務プログラムで生成された案件は，ソースノードから開始されます。このとき，ソースノードとアローで結び付けられている最初の業務ステップが実行可能になります。
2. 業務ステップが実行されると，業務ステップ内の作業が実行可能になります。ユーザまたは業務プログラムは，定義内容に従って作業を処理します。
3. 作業が完了すると業務ステップが完了します。これで，次の業務ステップが実行可能になります。定義内容に従って 2.と 3.の手順が繰り返され，案件の処理が進んでいきます。
4. 制御ノードでは，その種類や条件設定によって案件の流れが制御されます。
5. シンクノードに遷移した時点で案件が終了します。

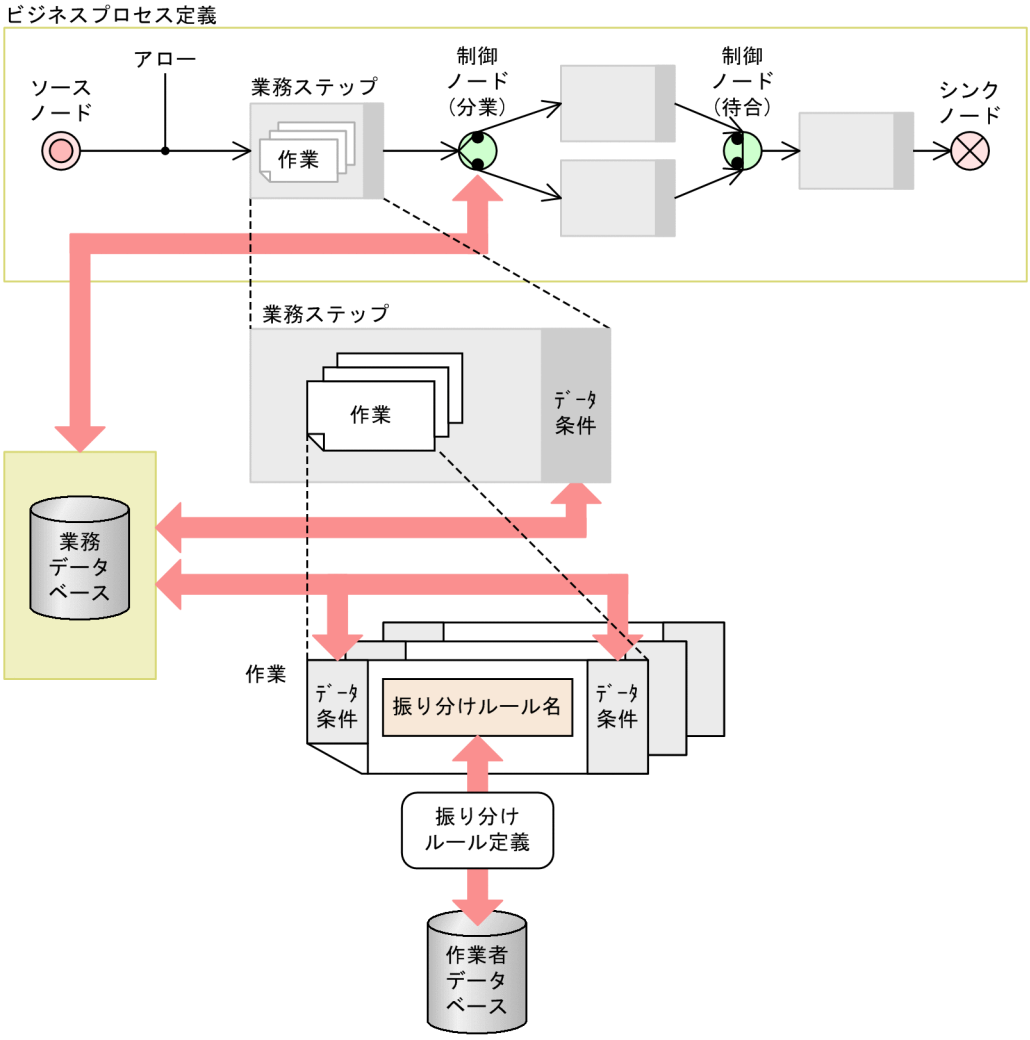
上記に示したように，案件の処理は，業務ステップや作業の状態が変化していくことで推進されます。つまり，「案件の推進」とは，「業務ステップや作業の状態を順次，変化させていくこと」と考えることができます。

2.2.2　ビジネスプロセス定義の構成

ビジネスプロセスは，業務の流れに合わせて幾つかの要素を組み合わせた構成になります。1 つのビジネスプロセスは，ソースノードで開始し，シンクノードで終了するように定義します。ソースノードとシンクノードの間には，実際の業務にあわせて業務ステップ，作業，制御ノード，アロー，振り分けルール名，データ条件といった要素を定義します。

ビジネスプロセス定義の構成を次の図に示します。

図 2-4　ビジネスプロセス定義の構成



ビジネスプロセス定義の主な構成要素について次に説明します。

(1)　ソースノード

ビジネスプロセスの開始点を表すものです。ビジネスプロセスに 1 つ設定します。

(2)　シンクノード

ビジネスプロセスの終了点を表すものです。ビジネスプロセスに 1 つ設定します。

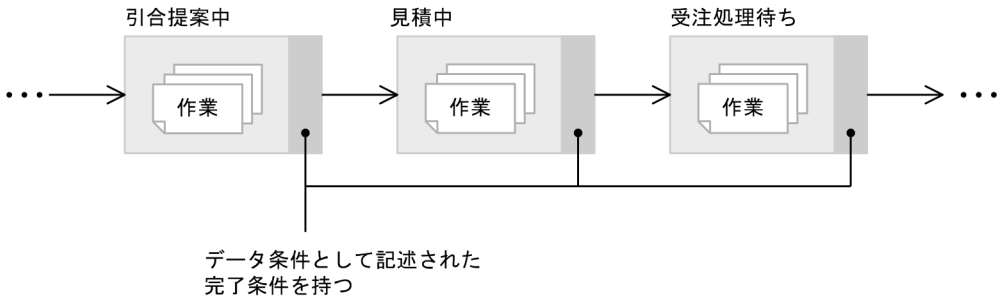
(3) 業務ステップ（業務状態）

業務の状態を示し，ある状態で実行可能な作業の集合として定義されるものです（ただし，運用上，作業を 1 つも含まない業務ステップもあります）。業務ステップは，通常これらの作業群がすべて完了することで「完了」となりますが，データ条件として記述された完了条件を別に持つことができます。

「2.1　ワーク管理システムによる業務の流れ」の例では，「引合提案中」，「見積中」，「受注処理待ち」，「売掛中」，「請求中」，「納品待ち」および「納品確認待ち」がそれぞれ業務ステップとなります。

業務ステップの概念を次の図に示します。

図 2-5　業務ステップの概念



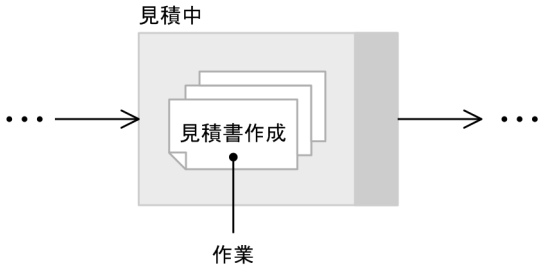
(4) 作業

ある業務ステップで実行する具体的な処理です。作業では，処理を実行する作業者の情報を振り分けルール名として持ちます。振り分けルール名を基に，別途定義されている振り分けルール定義を呼び出し，作業者データベースなどを検索することで作業者を特定します。

「2.1　ワーク管理システムによる業務の流れ」の例では，「見積中」の業務ステップで実行する「見積書作成」という処理などが作業となります。

作業の概念を次の図に示します。

図 2-6　作業の概念



なお，ワーク管理システムでは，組み込み作業を提供しています。組み込み作業とは，ワーク管理システムの基本的な機能を組み合わせることで実現するような複雑な処理を 1 つの作業としてまとめたものです。また，組み込み作業は，ワーク管理システムが作業者となって実行される以外は，通常の作業と同様の動作になります。

ワーク管理システムで提供している組み込み作業は，**並列作業**です。

並列作業は，案件内容に応じて，複数担当者に作業を振り分ける場合などに使用します。並列作業をビジネスプロセスに定義しておくことで，案件の実行時に複数の作業が自動的に生成されます。生成される作業数は，案件の実行時に決まります。

(5)　制御ノード

案件の流れを制御するノードです。制御ノードには，分岐ノード，分業ノード，先着ノードおよび待合ノードの 4 種類があります。

●分岐ノード

次の業務ステップとしてあらかじめ定義された複数の業務ステップから，条件に従って 1 つの業務ステップを選択し，開始します。

●分業ノード

次の業務ステップとしてあらかじめ定義された複数の業務ステップをすべて開始します。

●先着ノード

直前の業務ステップとしてあらかじめ定義された複数の業務ステップのうち，どれか 1 つが完了した時点で次の業務ステップを開始します。

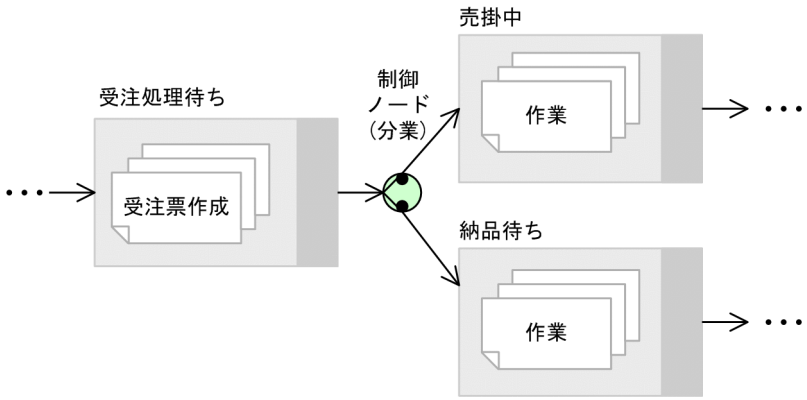
●待合ノード

直前の業務ステップとしてあらかじめ定義された複数の業務ステップのうち，すべてが完了した時点で次の業務ステップを開始します。

「2.1　ワーク管理システムによる業務の流れ」の例では，受注登録後，「売掛中」と「納品待ち」の 2 つの業務ステップに処理が分かれます。この制御は，分業ノードで実行しています。

制御ノードの概念を次の図に示します。

図 2-7　制御ノードの概念



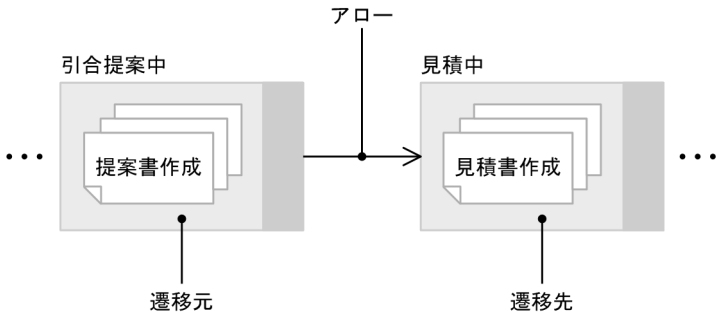
(6)　アロー

業務ステップおよびノード間の遷移を規定するためのものです。アローは，遷移元や遷移先となる業務ステップの情報を持ちます。

「2.1　ワーク管理システムによる業務の流れ」の「引合提案中」から「見積中」へのアローでは，遷移元が「引合提案中」，遷移先は「見積中」に当たります。

アローの概念を次の図に示します。

図 2-8　アローの概念



(7)　振り分けルール名

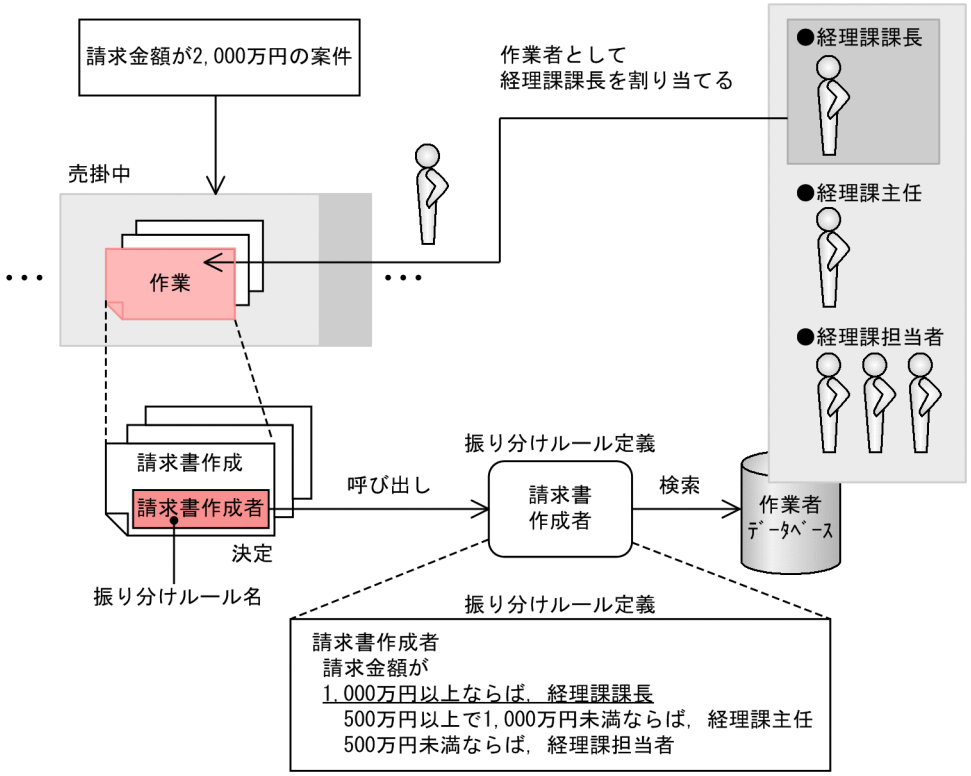
作業者を決定するためのルールの名称です。作業の実行時には，この名称で示された振り分けルール定義が呼び出されます。振り分けルール定義によって作業者データベースが検索され，作業者が決定されます。

「2.1　ワーク管理システムによる業務の流れ」の例では，「売掛中」の業務ステップで「請求書作成」という作業を実行します。この作業の作業者は，例えば次のように決定されます。

1. 「請求書作成」作業には，「請求書作成者」という振り分けルール名が記述されているとします。
2. 作業時には，振り分けルール名を基に「請求書作成者」という振り分けルール定義が呼び出されます。振り分けルール定義「請求書作成者」には，「請求金額が 1,000 万円以上ならば経理課課長，500 万円以上 1,000 万円未満ならば経理課主任，500 万円未満ならば経理課担当者」というルールが定義されているとします。
3. 振り分けルール定義「請求書作成者」に基づいて作業者データベースを検索し，「請求書作成」作業の作業者が決定されます。

振り分けルール名の概念を次の図に示します。

図 2-9　振り分けルール名の概念



振り分けルール定義については，「2.3　振り分けルール定義」を参照してください。

(8)　データ条件

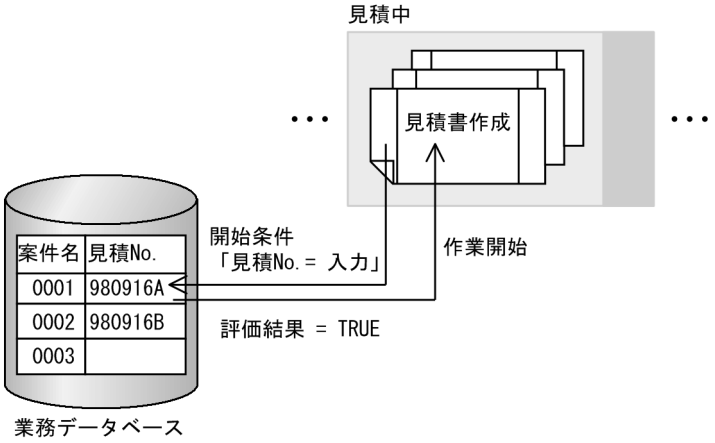
分岐ノードでの案件の流れを規定したり，業務ステップや作業の発生・完了を制御したりする条件です。データ条件は，業務データベースなどを参照することで評価されます。

「2.1　ワーク管理システムによる業務の流れ」の例では，「見積中」の業務ステップで「見積書作成」という作業を実行します。この作業の発生は，例えば次のように制御されます。

1.「見積書作成」作業の発生条件として「見積 No.=入力」が記述されているとします。
2. 業務ステップの開始時などに，発生条件に基づいて業務データベースを参照し，該当案件の見積 No.が入力されていれば，「見積書作成」作業が発生します。

データ条件の概念を次の図に示します。

図 2-10　データ条件



2.2.3　ビジネスプロセス定義の作成・登録と削除

(1)　ビジネスプロセス定義の作成

ビジネスプロセス定義は，ビジネスプロセス管理ツール（CSCIW-Definer）で作成します。ビジネスプロセス定義の作成方法については，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。

(2)　ビジネスプロセス定義の登録

作成したビジネスプロセス定義を実際に運用するためには，CSCIW へ登録する必要があります。ビジネスプロセス定義の登録は，ビジネスプロセス管理ツール（CSCIW-Definer）または CSCIW のコマンド（ciwmngbp）を使用します。登録されたビジネスプロセス定義は，CSCIW で実行できる形式に変換され，ワーク管理データベース内で管理されます。

ビジネスプロセス管理ツール（CSCIW-Definer）によるビジネスプロセス定義の登録については，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。ビジネスプロセス定義を登録するコマンド（ciwmngbp）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。

(3)　ビジネスプロセス定義の削除

不要になったビジネスプロセス定義は，ビジネスプロセス管理ツール（CSCIW-Definer）または CSCIW のコマンド（ciwmngbp）を使ってワーク管理データベースから削除します。

ビジネスプロセス管理ツール（CSCIW-Definer）によるビジネスプロセス定義の削除については，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。ビジネスプロセス定義を削除するコマンド（ciwmngbp）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。

2.2.4　CSCIW 内でのビジネスプロセス定義の管理

(1)　ビジネスプロセス定義のバージョン

ビジネスプロセス定義は，同一名称で複数のバージョンを持つことができます。CSCIW 内では，「ビジネスプロセス定義名+バージョン番号」としてビジネスプロセス定義が管理されます。つまり，同じ名称のビジネスプロセス定義であっても，バージョンが異なれば別のビジネスプロセス定義として扱われます。

ビジネスプロセス定義のバージョンは，ビジネスプロセス定義の作成時にユーザが任意の番号で指定します。

(2)　ビジネスプロセス定義の状態と有効期間

CSCIW 内の各ビジネスプロセス定義には，次の項目を設定できます。

●状態

ビジネスプロセス定義の状態です。状態には「活性」と「非活性」があり，案件を投入できるのは，「活性」状態のビジネスプロセス定義だけです。

●有効期間（案件投入可能期間）

ビジネスプロセス定義に案件を投入できる期間です。

つまり，案件を投入できるのは，「活性」状態で，かつ「有効期間内」のビジネスプロセス定義だけということになります。

ビジネスプロセス定義の「状態」および「有効期間」は，ビジネスプロセス定義の定義時に指定します。また，CSCIW への登録後，ビジネスプロセス管理ツール（CSCIW-Definer），または CSCIW のコマンド（ciwmngbp）で変更できます。

ビジネスプロセス管理ツール（CSCIW-Definer）によるビジネスプロセス定義の変更については，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。ビジネスプロセス定義を変更するコマンド（ciwmngbp）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。

2.3　振り分けルール定義

この節では，振り分けルール定義の概要および作業者決定の仕組みについて説明します。

2.3.1　振り分けルール定義による作業者の特定

(1)　振り分けルールとは

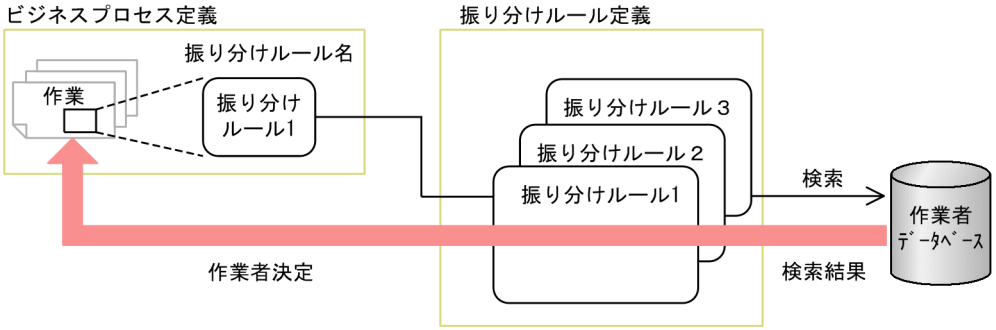
ワーク管理システムでは，ビジネスプロセス定義時に，各作業の作業者を直接指定しません。作業者を決定するためのルールだけを指定しておき，作業の実行時にこのルールの内容を適用することで作業者を決定します。作業者を決定するためのルールを**振り分けルール**といいます。

振り分けルールは，ビジネスプロセスとは別にあらかじめ定義しておきます。振り分けルールを定義したものを**振り分けルール定義**といいます。

ビジネスプロセス定義時には，各作業の作業者情報として振り分けルール定義の名称を指定しておきます。作業の実行時には，この名称で示された振り分けルール定義が呼び出されます。呼び出された振り分けルール定義の内容に基づいて，作業者の情報が格納されているデータベース（**作業者データベース**）などが検索され，実際の作業者が決定します。

振り分けルール定義の概念を次の図に示します。

図 2-11　振り分けルール定義の概念

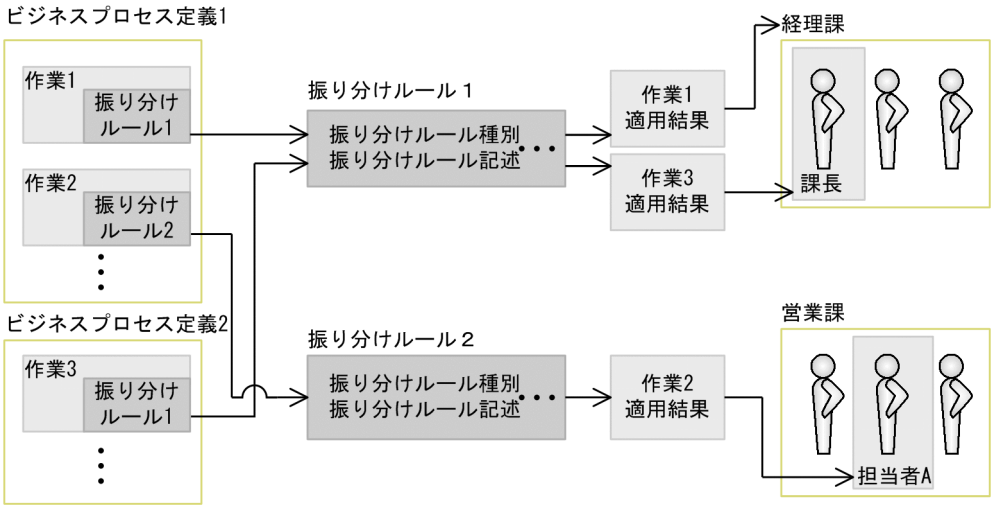


なお，ワーク管理システムでは，振り分けルール定義とビジネスプロセス定義とがそれぞれ独立して管理されています。そのため，1 つの振り分けルール定義を複数のビジネスプロセス定義間で共通に使用できます。

(2)　作業者を決定する仕組み

振り分けルール定義を使用して作業者を決定する仕組みについて，次の図に示します。

図 2-12　作業者決定の仕組み



＜説明＞

1. 作業 1 および作業 3 の作業者情報として「振り分けルール 1」が，作業 2 の作業者情報として「振り分けルール 2」がそれぞれ設定されています。
2. 作業 1 および作業 3 の実行時には，設定されている作業者情報に基づいて「振り分けルール 1 」という名称の振り分けルール定義が呼び出されます。

   また，作業 2 の実行時には，設定されている作業者情報に基づいて「振り分けルール 2」という名称の振り分けルール定義が呼び出されます。
3. 呼び出された振り分けルール定義の記述内容に基づいて業務データベースや作業者データベースなどが検索され，作業を実行する作業者が決定します。

   振り分けルールの適用によって，作業 1 の作業者は「経理課」，作業 2 の作業者は「営業課の担当者 A」，作業 3 の作業者は「経理課長」となります。

2.3.2　振り分けルール定義の構成

振り分けルール定義は，**振り分けルール種別**および**振り分けルール記述**で構成されます。

振り分けルール種別は，「RDB 検索」または「Java オブジェクト呼び出し」のどちらかを選択します。「RDB 検索」を選択した場合，振り分けルール記述には，作業者を特定するための検索条件を記述します。「RDB 検索」については，「2.7　RDB 検索」を参照してください。「Java オブジェクト呼び出し」については，「2.8　Java オブジェクト呼び出し」を参照してください。

2.3.3　振り分けルール定義の作成・登録と削除

(1)　振り分けルール定義の作成

振り分けルール定義は，ビジネスプロセス管理ツール（CSCIW-Definer）で作成します。振り分けルール定義の作成方法については，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。

(2)　振り分けルール定義の登録

作成した振り分けルール定義を実際に運用するためには，CSCIW へ登録する必要があります。振り分けルール定義の登録は，ビジネスプロセス管理ツール（CSCIW-Definer）または CSCIW のコマンド（ciwmngcr）を使用します。登録された振り分けルール定義は，CSCIW で実行できる形式に変換され，ワーク管理データベース内で管理されます。

ビジネスプロセス管理ツール（CSCIW-Definer）による振り分けルール定義の登録については，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。振り分けルール定義を登録するコマンド（ciwmngcr）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。

(3)　振り分けルール定義の削除

不要になった振り分けルール定義は，ビジネスプロセス管理ツール（CSCIW-Definer）または CSCIW のコマンドを使って削除します。ビジネスプロセス管理ツール（CSCIW-Definer）または CSCIW のコマンド（ciwmngcr）を実行することで，ワーク管理データベースの振り分けルール定義のデータが削除されます。

ビジネスプロセス管理ツール（CSCIW-Definer）による振り分けルール定義の削除については，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。振り分けルール定義を削除するコマンド（ciwmngcr）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。

2.3.4　CSCIW 内での振り分けルール定義の管理

(1)　振り分けルール定義のバージョン

振り分けルール定義は，同一名称で複数のバージョンを持つことができます。CSCIW 内では，「振り分けルール定義名+バージョン番号」として振り分けルール定義が管理されます。つまり，同じ名称の振り分けルール定義であっても，バージョンが異なれば別の振り分けルール定義として扱われます。

振り分けルール定義のバージョンは，振り分けルール定義の作成時にユーザが任意の番号で指定します。

(2) 振り分けルール定義の状態と有効期間

CSCIW 内の各振り分けルール定義には，次の項目を設定できます。

●状態

振り分けルール定義の状態です。状態には，「活性」と「非活性」があり，案件実行時に適用されるのは，「活性」状態の振り分けルール定義だけです。

●有効期間（適用可能期間）

振り分けルール定義を適用できる期間です。

つまり，案件実行時に適用されるのは，「活性」状態で，かつ「有効期間内」の振り分けルール定義だけということになります。

振り分けルール定義の「状態」および「有効期間」は，振り分けルールの定義時に指定します。また，CSCIW への登録後，ビジネスプロセス管理ツール（CSCIW-Definer），または CSCIW のコマンド（ciwmngcr）で変更することもできます。

ビジネスプロセス管理ツール（CSCIW-Definer）による振り分けルール定義の変更については，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。振り分けルール定義を変更するコマンド（ciwmngcr）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。

2.3.5 ユーザとユーザグループ

(1) ユーザとユーザグループの関係

ワーク管理システムでは，作業の実行時に振り分けルール定義を適用することで作業者を割り当てます。作業者として割り当てられるのは，作業者データベースなどで管理されているユーザまたはユーザグループとなります。

●ユーザ

作業者データベースなどで管理されている個人のことです。具体例としては，「（営業 1 課の）A さん」，「営業 1 課長」などが挙げられます。

作業者データベースなどの内部では，ユーザはユーザ識別子によって管理されています。振り分けルール定義を適用した結果，作業者としてユーザが割り当てられた場合は，作業者情報にユーザ識別子が設定されます。

●ユーザグループ

ユーザの集合です。また，ユーザグループの集合も，より大きなまとまりのユーザグループとなります。具体例としては，「営業 1 課」，「営業部」などが挙げられます。

作業者データベースなどの内部では，ユーザグループは**ユーザグループ識別子**によって管理されています。振り分けルール定義を適用した結果，作業者としてユーザグループが割り当てられた場合は，作業者情報にユーザグループ識別子が設定されます。

(2) ワーク管理システムでのユーザとユーザグループの扱い

ワーク管理システムでは，振り分けルール定義の適用によって設定された作業者情報が，ユーザ識別子なのか，ユーザグループ識別子なのかを区別しません。どちらも，単に作業者情報（**作業者 ID**）として同様に扱われます。そのため，作業者データベースなどでは，ユーザ識別子とユーザグループ識別子が重複しないようにする必要があります。

また，ワーク管理システムでは，ユーザとユーザグループの関係を考慮しません。そのため，ユーザグループを作業者として割り当てる場合，実際の作業をどのユーザが実行するかについては業務プログラム側で制御する必要があります。

作業者 ID には，振り分けルール定義によって取得された ID が設定されます。作業者 ID が設定されるのは，作業が実行開始可能（Ready）になったときです。状態遷移については，マニュアル「uCosminexus Service Coordinator Interactive Workflow AP 開発ガイド」を参照してください。

2.4　データ条件と業務データ

この節では，データ条件と業務データについて説明します。

2.4.1　データ条件による案件遷移の制御

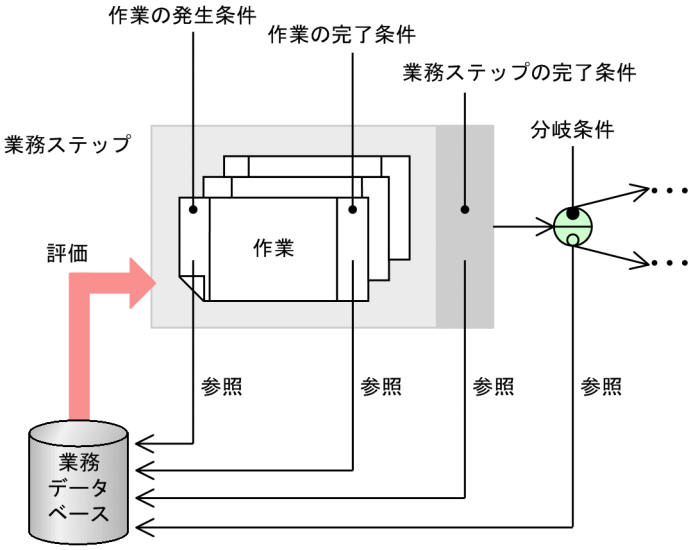
(1)　データ条件とは

データ条件とは，分岐ノードでの案件の流れを規定したり，業務ステップや作業の発生・完了を制御したりするための条件のことです。案件の実行時には，データ条件の内容に基づいて，業務データが格納されているデータベース（**業務データベース**）などが参照され，データ条件が評価されます。この評価結果によって，案件の流れおよび業務ステップや作業の状態が制御されます。

データ条件は，「分岐ノードの分岐条件」，「業務ステップの完了条件」および「作業の発生条件と完了条件」として設定できます。

データ条件による案件制御の概念を次の図に示します。

図 2-13　データ条件による案件制御の概念



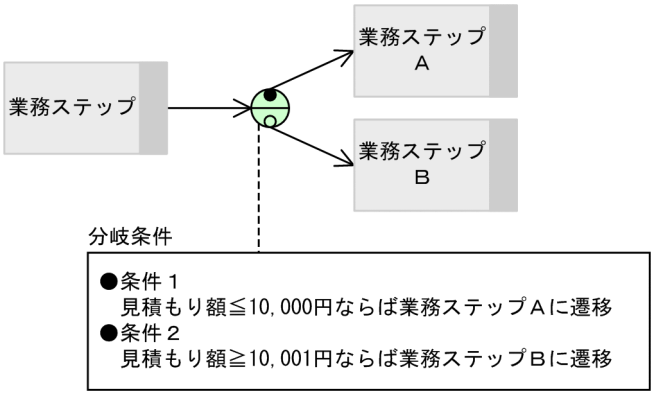
(2)　データ条件による案件制御の例

データ条件による案件制御の例を次に説明します。

●分岐

分岐ノードの分岐条件としてデータ条件を設定できます。案件が分岐ノードに到達した時点でデータ条件が評価され，案件の分岐先を決定します。

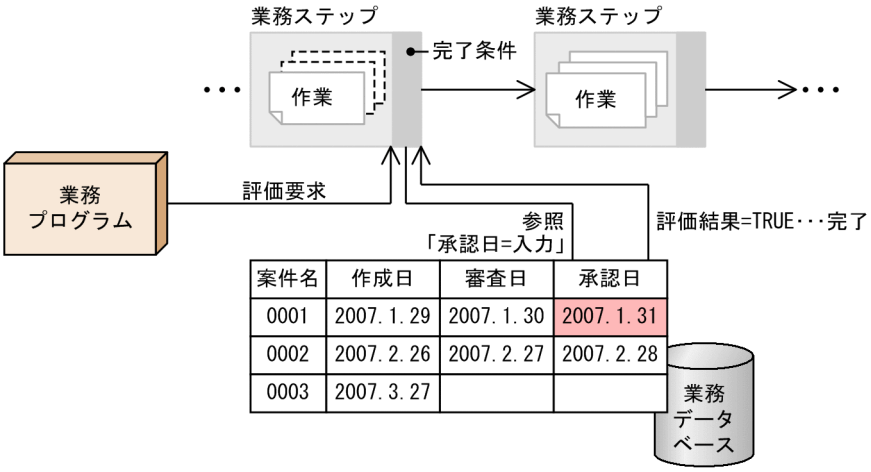
例として，「見積もり金額が 10,000 円以下なら業務ステップ A に，10,001 円以上なら業務ステップ B に遷移する」という運用が考えられます。



●業務ステップの完了

業務ステップの完了条件としてデータ条件を設定できます。業務ステップ内の各作業の完了時や業務プログラムからの評価要求時にデータ条件が評価され，条件が満たされていれば業務ステップが完了します。

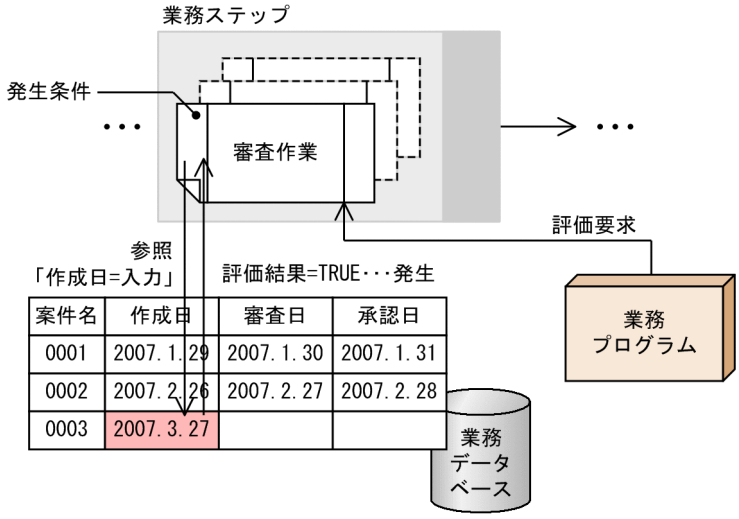
例として，「承認日に日付が入力されていれば，該当する業務ステップ内の作業がすべて完了していなくても業務ステップを完了する」という運用が考えられます。



●作業の発生

作業の発生条件としてデータ条件を設定できます。業務ステップの開始時，業務ステップ内にある，ほかの作業の完了時または業務プログラムからの評価要求時にデータ条件が評価され，条件が満たされていれば作業が発生します。

例として，「作成日に日付が入力されていたら，審査作業を発生させる」という運用が考えられます。

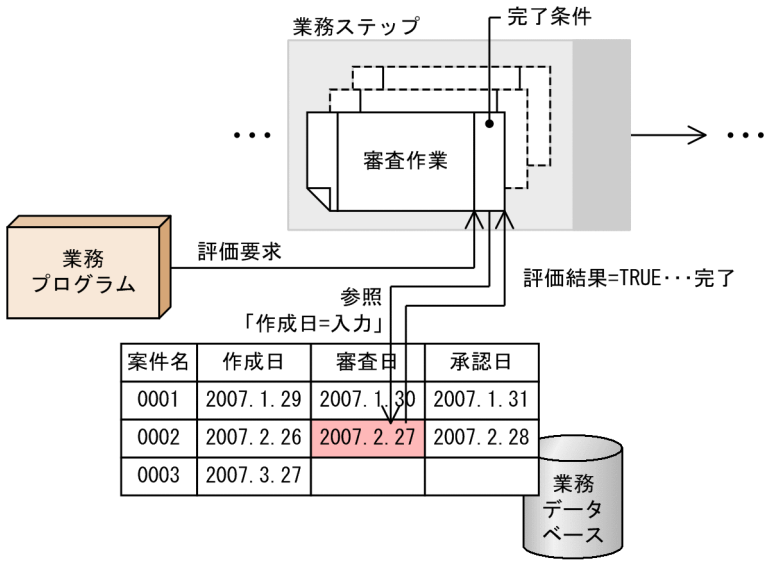


案件名	作成日	審査日	承認日
0001	2007. 1. 29	2007. 1. 30	2007. 1. 31
0002	2007. 2. 26	2007. 2. 27	2007. 2. 28
0003	2007. 3. 27		

●作業の完了

作業の完了条件としてデータ条件を設定できます。作業の発生時や業務プログラムからの評価要求時にデータ条件が評価され，条件が満たされていれば作業が完了します。

例として，「見積書の審査日に日付が入力されていたら，審査作業を完了する」という運用が考えられます。



案件名	作成日	審査日	承認日
0001	2007. 1. 29	2007. 1. 30	2007. 1. 31
0002	2007. 2. 26	2007. 2. 27	2007. 2. 28
0003	2007. 3. 27		

2.4.2　データ条件の構成

データ条件は，**データ条件種別**および**データ条件記述**で構成されます。

データ条件種別は，「RDB 検索」または「Java オブジェクト呼び出し」のどちらかを選択します。「RDB 検索」を選択した場合，データ条件記述には，データ条件を評価するための検索条件を記述します。「RDB 検索」については，「2.7　RDB 検索」を参照してください。「Java オブジェクト呼び出し」については，「2.8　Java オブジェクト呼び出し」を参照してください。

2.5　案件の基本的な進み方と動的制御

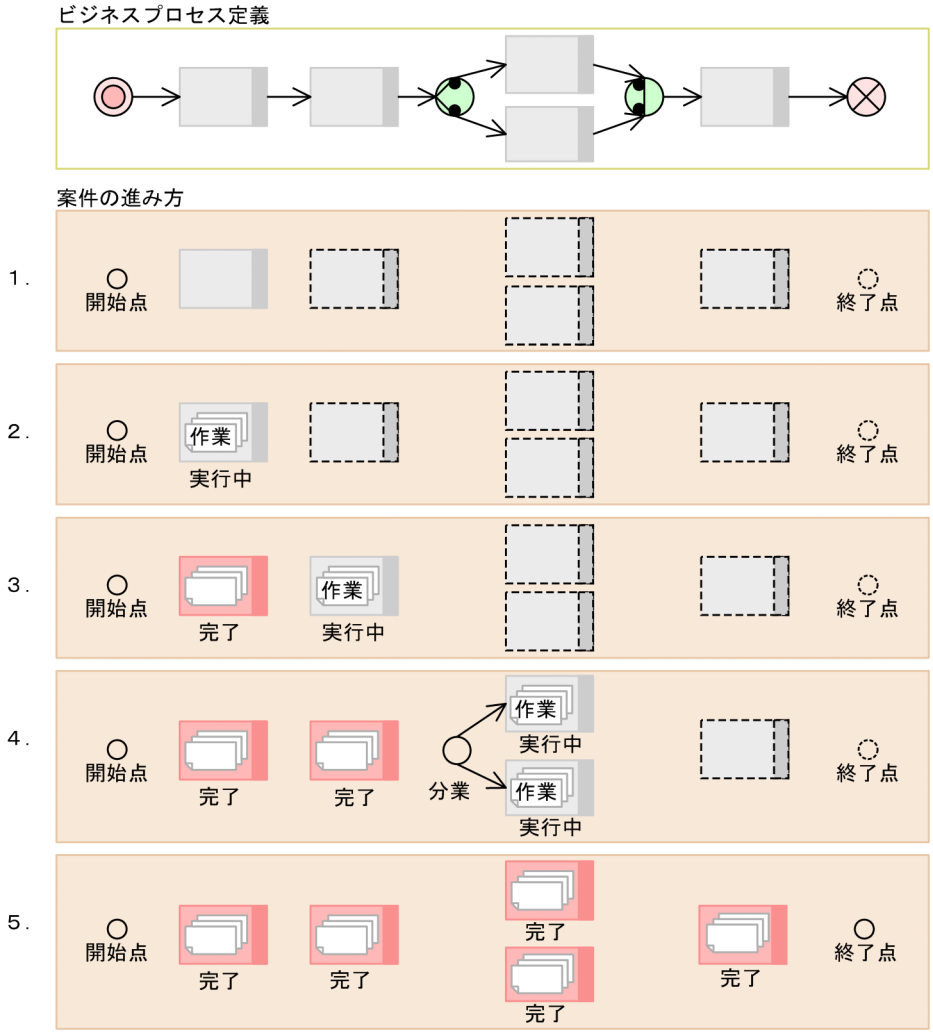
この節では，基本的な案件の進み方と案件の動的な制御について説明します。また，組み込み作業での案件の進み方についてもあわせて説明します。

2.5.1　案件の基本的な進み方

ビジネスプロセス定義に従って生成された案件は，定義に記述された手順で処理されていきます。

案件の基本的な進み方を次の図に示します。

図 2-14　案件の基本的な進み方



＜説明＞

1. ユーザまたは業務プログラムによって生成された案件は，ソースノードから開始されます。このとき，ソースノードとアローで結び付けられている最初の業務ステップが実行可能になります。

2. 業務ステップが実行されると，業務ステップ内の作業が実行可能になります。ユーザまたは業務プログラムは，定義内容に従って作業を処理します。
3. 作業が完了した際に，属する業務ステップ内のすべての作業が完了していると，属する業務ステップも完了します。これで，次の業務ステップが実行可能になります。

   定義内容に従って 2.と 3.の手順が繰り返され，案件の処理が進んでいきます。
4. 制御ノードでは，その種類や条件設定によって案件の流れが制御されます。
5. シンクノードに遷移した時点で案件が終了します。

上記に示したように，案件の処理は，業務ステップや作業の状態が変化していくことで推進されます。つまり，「案件の推進」とは，「業務ステップや作業の状態を順次，変化させていくこと」と考えることができます。

また，通常，業務ステップは，案件が推進することで，新たに生成されていきますが，ビジネスプロセス定義で業務ステップの**事前生成**を指定した場合は，案件が開始された時点で，指定した業務ステップが初期状態で生成されます。事前生成された業務ステップは，案件が推進するときには，実行可能になります。業務ステップの事前生成は，業務ステップの属性（処理期限や優先度など）をあらかじめ決定しておきたい場合に使用します。

案件の開始から終了までの，業務ステップおよび作業の状態変化について次に説明します。また，制御ノードでの案件の遷移についてもあわせて説明します。

(1) 案件の開始

案件は，ビジネスプロセスの入り口であるソースノードから開始されます。**ソースノード**とは，案件の開始を意味するノード（案件の流れを制御する情報）のことです。

ユーザまたは業務プログラムが案件を投入すると，ビジネスプロセス定義上のソースノードと結び付いている最初の業務ステップが開始されます。

(2) 業務ステップの状態遷移

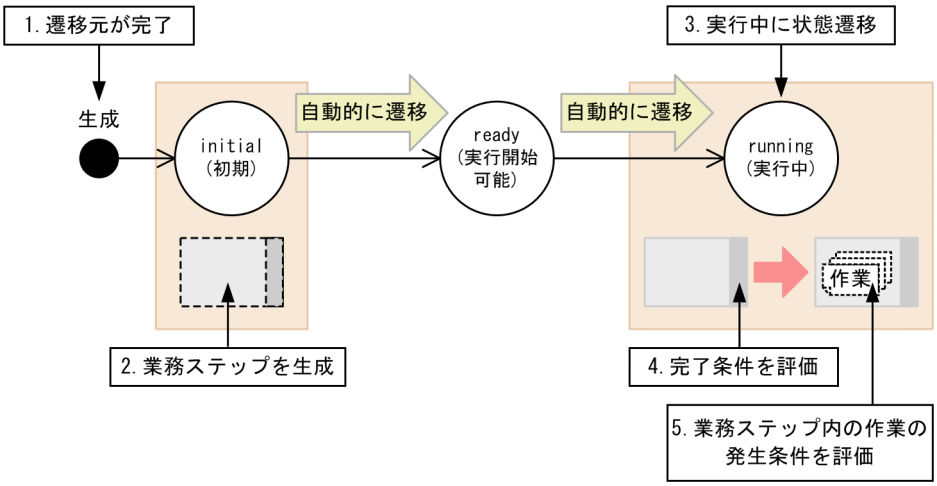
業務ステップは，現在の案件がビジネスプロセス定義のどこまで進んでいるかを表現し，また業務ステップに含まれる作業を管理しています。

作業が生成されるまでの業務ステップの処理の流れ，および業務ステップが完了するまでの処理の流れを次に説明します。

(a) 作業が生成されるまでの業務ステップの処理の流れ

業務ステップが生成され業務ステップ内の作業が生成されるまでの基本的な処理の流れを次の図に示します。

図 2-15　作業が生成されるまでの業務ステップの処理の流れ



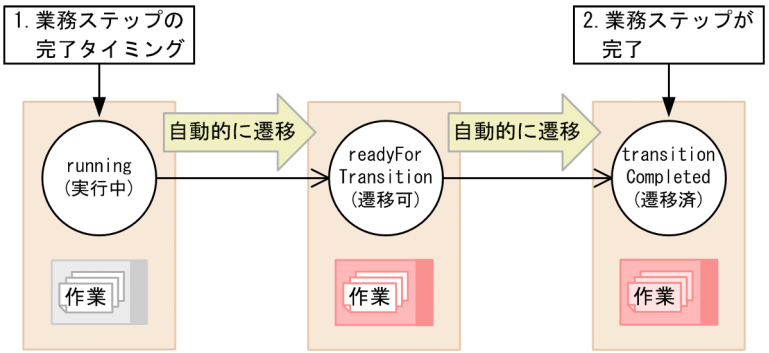
＜説明＞

1. 遷移元の完了によって業務ステップに遷移してきます。
2. 業務ステップを生成し，その状態は「初期」となります。
3. 「初期」から自動的に状態が遷移して，「実行中」となります。
4. 「実行中」になると業務ステップの完了条件を評価します。
   完了条件が成立した場合は，業務ステップが完了し，案件は次に遷移します。
   完了条件が成立しない場合は，5.の処理に進みます。
5. 業務ステップ内の作業の発生条件を評価します。
   発生条件が成立した場合は，作業の生成処理を実行します。
   発生条件が成立しない場合は，何もしません。

(b)　業務ステップが完了するまでの処理の流れ

業務ステップが完了するまでの基本的な処理の流れを次の図に示します。

図 2-16　業務ステップが完了するまでの処理の流れ



＜説明＞

1. 業務ステップの完了タイミングとなります。

2.「実行中」から自動的に状態が遷移して，「遷移済」となり，業務ステップが完了します。

業務ステップが完了するタイミング，および業務ステップの完了条件が評価されるタイミングを次に示します。

●業務ステップが完了するタイミング

- 設定されている完了条件が真になった時点
  ただし，完了条件の設定を省略した場合，完了条件は常に偽（FALSE）として評価されます。
- 業務ステップに含まれるすべての作業が完了した時点
  ただし，業務ステップ内に生成された作業が存在しないときは，該当しません。
- API による完了要求があった時点
- 案件が完了した時点

●業務ステップの完了条件が評価されるタイミング

- 業務ステップを開始した時点
- 業務ステップに含まれる作業が完了した時点
- 業務ステップの条件を評価する API が発行された時点
- 「中断」状態の業務ステップに対する再開要求があった時点

(3) 作業の状態遷移

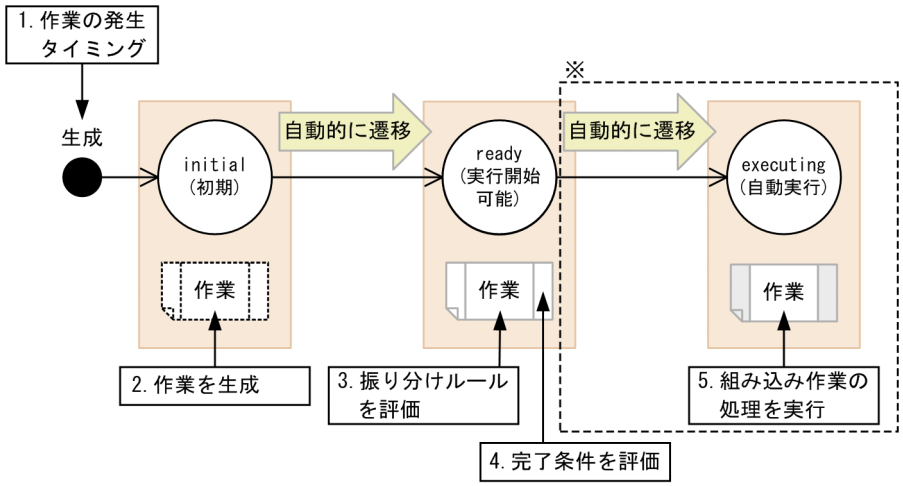
作業は，ある業務ステップで実行する具体的な作業内容を表現し，その作業を実行する作業者などの情報を管理しています。

「実行開始可能」状態に遷移するまでの処理の流れ，および作業が完了するまでの処理の流れを次に説明します。

(a) 「実行開始可能」状態に遷移するまでの処理の流れ

作業が生成され「実行開始可能」状態に遷移するまでの基本的な処理の流れを次の図に示します。

図 2-17 「実行開始可能」状態に遷移するまでの基本的な処理の流れ



注※ 組み込み作業の場合だけ遷移します。

＜説明＞

1. 作業の発生のタイミングとなります。
2. 作業を生成し，その状態は「初期」となります。
3. 「初期」から「実行開始可能」に自動的に遷移する際に，振り分けルールを評価し，作業者を決定します。作業者が決定すると「実行開始可能」となります。
4. 「実行開始可能」になると作業の完了条件を評価します。

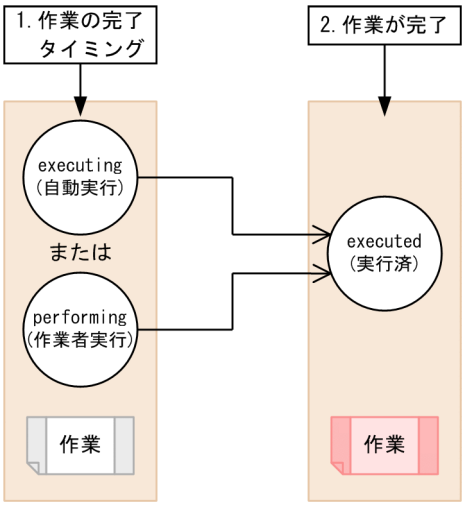
   完了条件が成立した場合，作業は「実行省略」状態となり完了します。そして，作業が属する業務ステップの評価要求処理を実行します。業務ステップの評価要求については，「2.5.2(2) 業務ステップおよび作業の評価要求」を参照してください。

   完了条件が成立しない場合，一般作業のときは，作業アプリケーションが定義されていれば作業アプリケーションを呼び出し，作業アプリケーションが定義されていなければ何もしません。組み込み作業のときは 5.の処理に進みます。
5. 組み込み作業の場合は「自動実行」状態となります。そのあと，組み込み作業の処理を実行します。組み込み作業の処理については，「2.5.3 組み込み作業での案件の進み方」を参照してください。

(b) 作業が完了するまでの処理の流れ

作業が完了するまでの基本的な処理の流れを次の図に示します。

図 2-18　作業が完了するまでの基本的な処理の流れ



＜説明＞

1. 作業の完了タイミングとなります。
2.「自動実行」または「作業者実行」から「実行済」となり，作業が完了します。作業が完了すると，作業が属する業務ステップの評価要求処理を実行します。業務ステップの評価要求については，「2.5.2(2)　業務ステップおよび作業の評価要求」を参照してください。

作業が完了するタイミング，作業の発生条件および完了条件が評価されるタイミングを次に示します。

●作業が完了するタイミング

- 設定されている完了条件が真になった時点
  ただし，完了条件の設定を省略した場合，完了条件は常に偽（FALSE）として評価されます。
- API による完了要求があった時点
- 作業に含まれる業務ステップが完了した時点

●作業の発生条件が評価されるタイミング

- 業務ステップを開始した時点
- 同一業務ステップ内のほかの作業が完了した時点
- 業務ステップの条件を評価する API が発行された時点
- 「中断」状態の業務ステップに対する再開要求があった時点

●作業の完了条件が評価されるタイミング

- 作業を生成した時点
- 作業の条件を評価する API が発行された時点
- 「中断」状態の作業に対する再開要求があった時点

(4) 制御ノードでの案件遷移

案件は，制御ノードによってそのあとの遷移を制御されます。**制御ノード**には，分岐ノード，分業ノード，先着ノードおよび待合ノードの４種類があります。

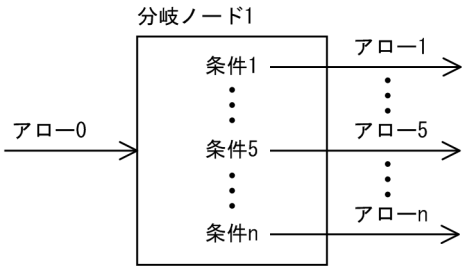
それぞれの制御ノードで，案件の遷移がどのように制御されるかを次に説明します。

(a) 分岐ノードでの案件遷移

分岐ノードは，次の遷移先としてあらかじめ定義された複数のアローから，条件に従って１つのアローを選択し，遷移します。

分岐ノードでの案件遷移を次の図に示します。

図 2-19 分岐ノードでの案件遷移



＜説明＞

1. アロー０の遷移元が完了すると，分岐ノード１が実行されます。
2. 分岐ノード１が実行されると，あらかじめ定義されている分岐条件（1～n）が案件の内容で順次，評価されます。
3. 例えば，分岐条件５が成立したことで，それに対応するアロー５に遷移します。遷移先の業務ステップが開始された時点で分岐ノード１は終了します。

分岐ノードは，定義されている条件を順次評価していき，最初に成立した条件に対応したアローに遷移します。それ以降の条件は評価しません。

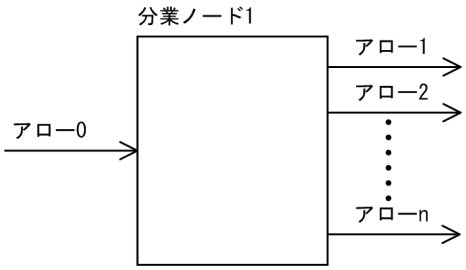
記述されている条件をすべて評価し，成立した条件がなかった場合は，デフォルトの分岐先として設定されているアローに遷移します。

(b) 分業ノードでの案件遷移

分業ノードは，次の遷移先としてあらかじめ定義された複数のアローすべてに遷移します。

分業ノードでの案件遷移を次の図に示します。

図 2-20　分業ノードでの案件遷移



＜説明＞

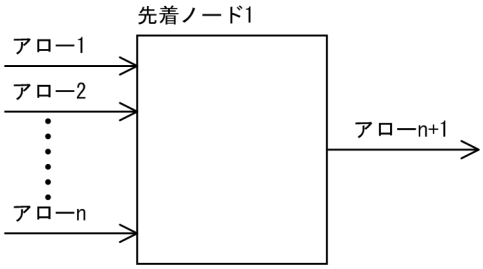
1. アロー 0 の遷移元が完了すると，分業ノード 1 が実行されます。
2. 分業ノード 1 が実行されると，それに続くすべてのアロー（アロー 1～n）へ同時に遷移します。遷移先のすべての業務ステップが開始された時点で分業ノード 1 は終了します。

(c)　先着ノードでの案件遷移

先着ノードは，直前の遷移元としてあらかじめ定義された複数のアローのうち，どれか 1 つが遷移してきた時点で次のアローに遷移します。

先着ノードでの案件遷移を次の図に示します。

図 2-21　先着ノードでの案件遷移



＜説明＞

1. 直前のアロー（アロー 1～n）のうち，1 つのアロー（例えば，アロー 2）が完了すると，先着ノード 1 が実行されます。
2. 先着ノード 1 が実行されると，それに続くアロー（アロー n+1）に遷移します。遷移先の業務ステップが開始された時点で先着ノード 1 は終了します。

なお，先着ノードでは，**後続停止**の設定ができます。

後続停止を設定した場合は，一度後続のアローに遷移したあとは，2 回目以降が実行されません。これによって，後続のアローに同一の案件が再び遷移することを抑止できます。

重要

ソースノードの直後には後続停止を定義しないでください。ソースノードの直後に後続停止を定義した場合，ソースノードからのアローが遷移したあとにほかのアローからの遷移が抑止されます。

後続停止を設定しない場合は，一度後続のアローに遷移しても，直前の 2 つ目以降のアローからの遷移があったときに，再度，後続のアローに遷移します。

後続停止を設定した場合と設定しない場合での，先着ノードでの案件遷移の違いについて次の図に示します。

図 2-22　後続停止を設定した場合の案件遷移

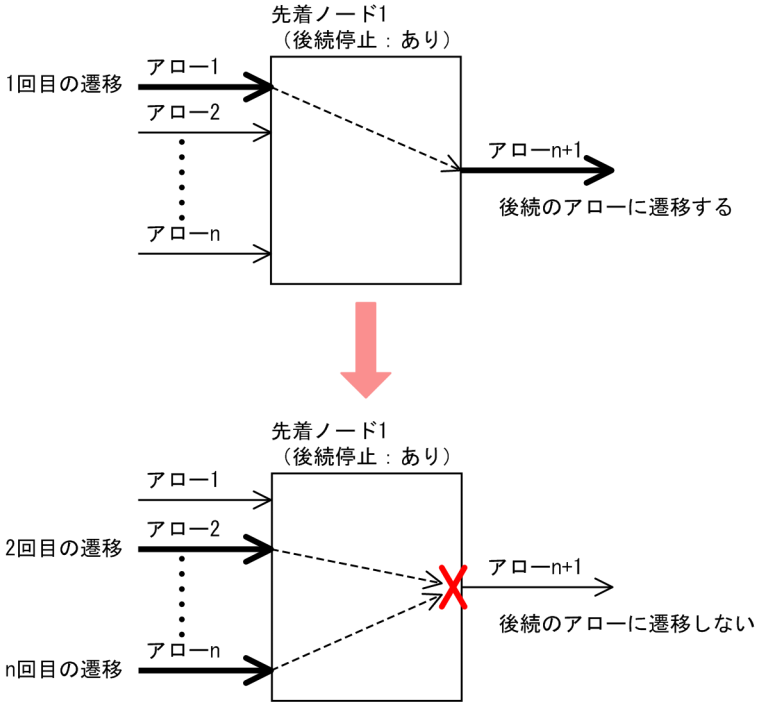
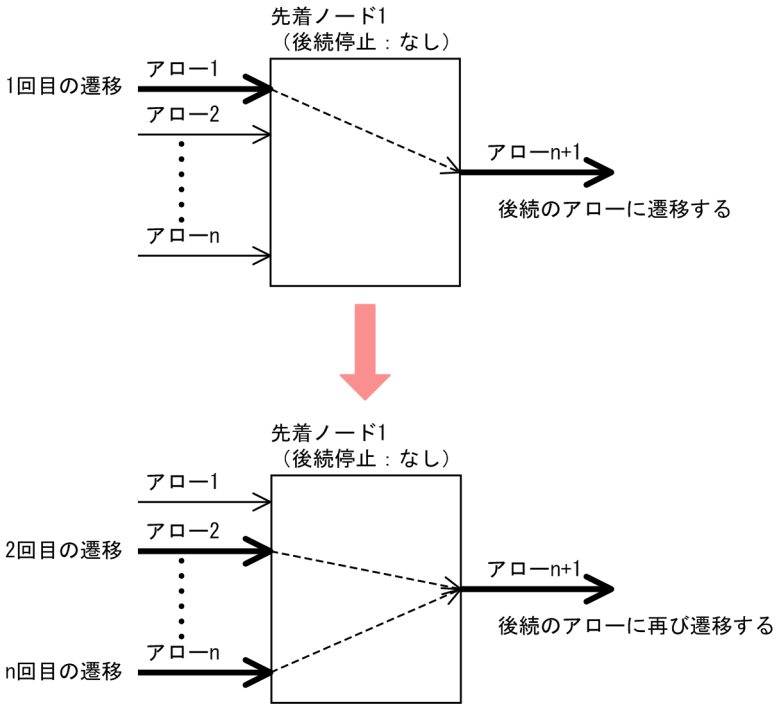


図 2-23　後続停止を設定しない場合の案件遷移

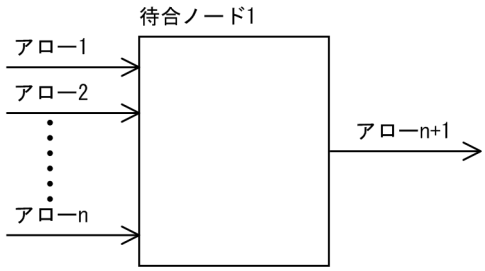


(d)　待合ノードでの案件遷移

待合ノードは，直前の遷移元としてあらかじめ定義された複数のアローのうち，すべてのアローから遷移してきた時点で次のアローに遷移します。

待合ノードでの案件遷移を次の図に示します。

図 2-24　待合ノードでの案件遷移



＜説明＞

1. 直前のアロー（アロー 1～n）のすべてから遷移してくると，待合ノード 1 が実行されます。
2. 待合ノード 1 が実行されると，それに続くアロー（アロー n+1）に遷移します。遷移先の業務ステップが開始された時点で待合ノード 1 は終了します。

重要

ソースノードの直後には待合ノードを定義しないでください。ソースノードの直後に待合ノードを定義した場合，ソースノードから遷移した直後に，別のアローからの遷移を待つ状態となり，次のアローに遷移できなくなります。

(5) 案件の終了

案件は，ビジネスプロセスの出口であるシンクノードに到達すると終了します。**シンクノード**とは，案件の終了を意味するノードのことです。

シンクノードに到達した案件は「完了」の状態となります。また，案件がシンクノードへ到達すると，処理されていない業務ステップおよび作業があっても，それらはすべて強制終了されます。

2.5.2 案件の動的な制御

案件は，基本的にビジネスプロセス定義に従って処理されていきます。一方，ビジネスプロセス定義に記述されている内容とは別に，案件の処理を**動的**に制御することもできます。案件処理に対する動的な制御は，ワーク管理システムで提供されている API を利用した業務プログラムから実行します。ワーク管理システムの API については，マニュアル「uCosminexus Service Coordinator Interactive Workflow AP 開発ガイド」を参照してください。

(1) 引き戻しと差し戻し

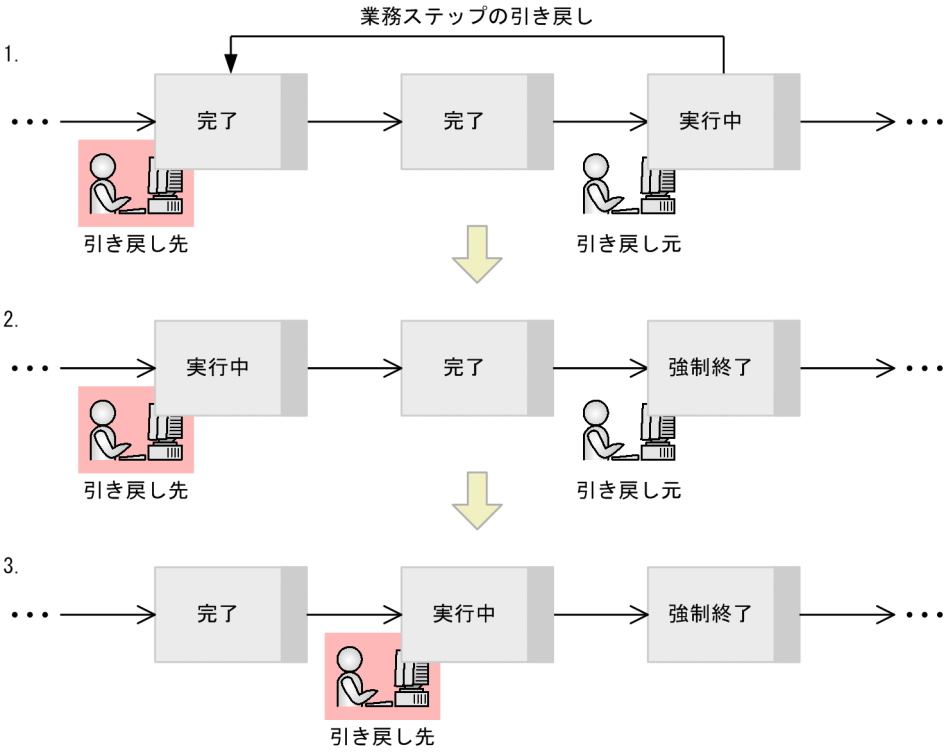
現在実行中の業務ステップから，すでに完了した業務ステップに案件処理を戻し，その時点から案件処理をし直せます。これによって，案件の「引き戻し」や「差し戻し」といった処理を実現できます。

(a) 引き戻し

引き戻し処理では，現在実行中の業務ステップを検索し，引き戻し先の業務ステップに案件処理を戻します。

引き戻し処理の流れを次の図に示します。

図 2-25　引き戻し処理の流れ



＜説明＞

1. ユーザ（業務プログラム）は，引き戻し対象となる現在実行中の業務ステップを検索します。その情報を基に，引き戻し元の業務ステップ（現在実行中の業務ステップ）と引き戻し先の業務ステップを指定し，CSCIW に引き戻し処理を要求します。
2. CSCIW は，ユーザ（業務プログラム）からの要求によって，引き戻し元の業務ステップを「強制終了」状態にし，引き戻し先の業務ステップを新しく生成して「実行中」状態にします。

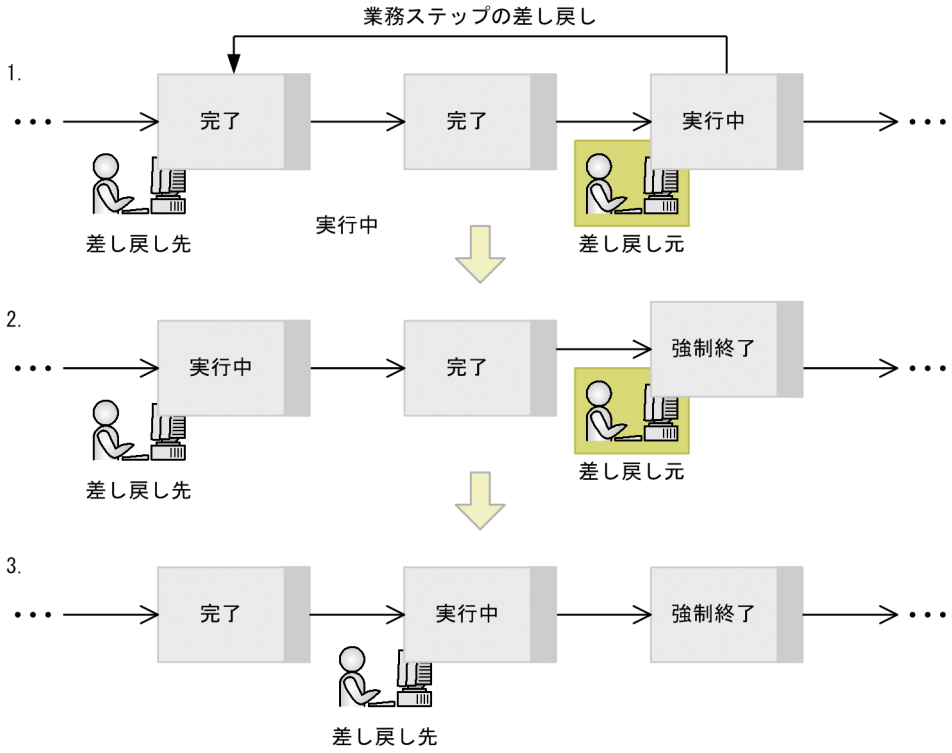
   このとき CSCIW は，引き戻し先と引き戻し元の間にある業務ステップの状態を変更しません。また，引き戻し以前の処理によって更新された業務データの値も変更しません。これらの変更が必要な場合は，業務プログラム側で処理します。
3. 引き戻し先の業務ステップでの処理およびそのあとの処理は，ビジネスプロセス定義に従って実行されます。

(b)　差し戻し

差し戻し処理では，すでに完了した業務ステップを検索して差し戻し先を決定し，現在実行中の業務ステップから差し戻し先の業務ステップに案件処理を戻します。

差し戻し処理の流れを次の図に示します。

図 2-26　差し戻し処理の流れ



＜説明＞

1. ユーザ（業務プログラム）は，すでに完了した業務ステップを検索し，差し戻し先となる業務ステップを選択します。その情報を基に，差し戻し元の業務ステップ（現在実行中の業務ステップ）と差し戻し先の業務ステップを指定し，CSCIW に差し戻し処理を要求します。
2. CSCIW は，ユーザ（業務プログラム）からの要求によって，差し戻し元の業務ステップを「強制終了」状態にし，差し戻し先の業務ステップを新しく生成して「実行中」状態にします。

   このとき CSCIW は，差し戻し先と差し戻し元の間にある業務ステップの状態を変更しません。また，差し戻し以前の処理によって更新された業務データの値も変更しません。これらの変更が必要な場合は，業務プログラム側で処理してください。
3. 差し戻し先の業務ステップでの処理およびそのあとの処理は，ビジネスプロセス定義に従って実行されます。

(c)　引き戻しおよび差し戻しに関する注意事項

引き戻し元と引き戻し先との間，または差し戻し元と差し戻し先との間に制御ノードが含まれる場合，次に示すどちらかの形態に合致している必要があります。

●形態 1

引き戻し元と引き戻し先との間，または差し戻し元と差し戻し先との間に含まれる部分（以降，**巡回部分**と呼ぶ）が，次に示す条件をすべて満たしている形態。

- 巡回部分での遷移は，必ず，引き戻し先または差し戻し先の業務ステップから始まり，引き戻し元または差し戻し元の業務ステップで終了する。

- 引き戻し元または差し戻し元への遷移時は，巡回部分内のすべての待合ノードが「待合中」状態でなく，かつ巡回部分内のすべての業務ステップが「実行中」状態でない。
- 巡回部分内のすべての先着ノードには，後続停止が設定されていない。

●形態 2

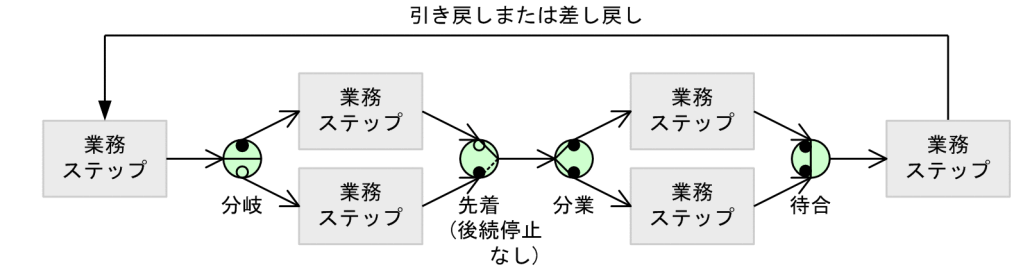
巡回部分に，「業務ステップ」または「後続停止が設定されていない先着ノード」だけが含まれる形態。

上記の形態でない場合，引き戻しまたは差し戻しが正しく動作しないおそれがあります。

引き戻しまたは差し戻しが正しく動作する例，および正しく動作しない例を，それぞれ次の図に示します。

図 2-27　引き戻しまたは差し戻しが正しく動作する例

●形態1の条件を満たす場合



●形態2の条件を満たす場合

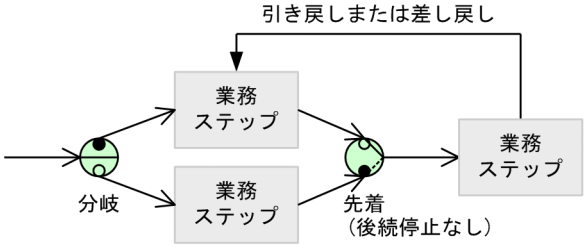
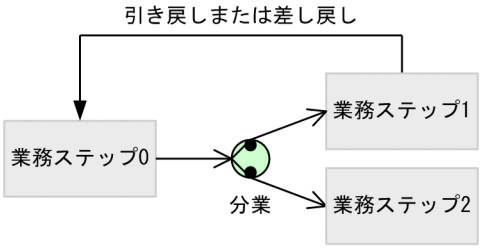


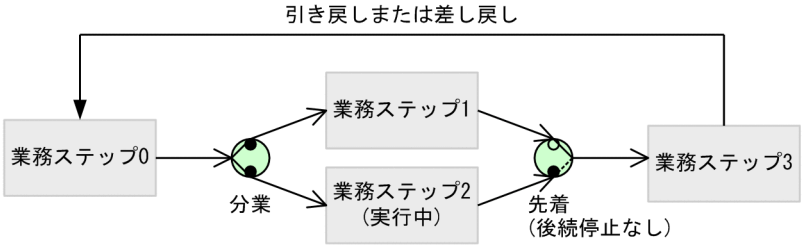
図 2-28　引き戻しまたは差し戻しが正しく動作しない例

●巡回部分内の案件遷移経路が閉じた構造になっていない場合



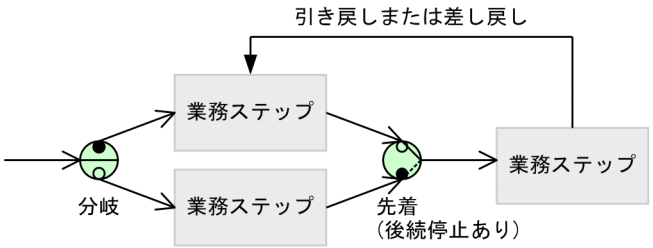
業務ステップ2に同一の案件が複数回遷移してしまう。

●巡回部分内の業務ステップの一部が実行中の場合



実行中の業務ステップ2に同一案件が遷移してしまう可能性がある。

●巡回部分に後続停止が設定されている先着ノードがある場合



案件が先着ノードから先に遷移しなくなってしまう。

(2)　業務ステップおよび作業の評価要求

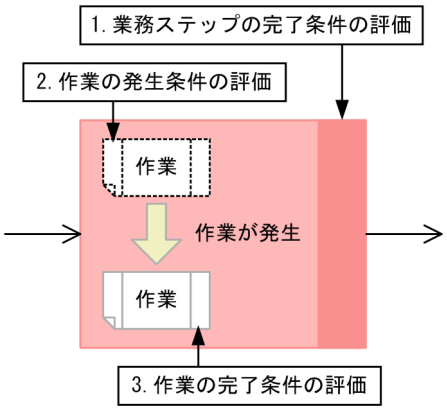
実行中の業務ステップおよびその業務ステップに含まれる各作業のデータ条件を，任意のタイミングで評価できます。

(a)　業務ステップの評価要求

業務ステップの評価要求のタイミングは，業務ステップの条件を評価する API が発行された時点です。

業務ステップの評価要求を実行したときの処理の順序を次の図に示します。

図 2-29　業務ステップの評価要求を実行したときの処理の順序



＜説明＞

1. 業務ステップの完了条件を評価します。

   完了条件が成立した場合，業務ステップが完了し，案件は次に遷移します。

   完了条件が成立しない場合，2.の処理に進みます。

2. 業務ステップに属する作業のうち，未発生の作業または条件再評価フラグが設定されていて，かつ完了状態にある作業の発生条件を評価します。

   一つも発生条件が成立しない場合，終了します。

   1 つ以上の発生条件が成立した場合，3.の処理に進みます。

3. 発生した作業の完了条件を評価します。

   一つも発生した作業の完了条件が成立しない場合，発生した作業は未実行の状態で終了します。

   1 つ以上の発生した作業の完了条件が成立し，かつ業務ステップ内に完了していない作業がある場合，完了条件が成立した作業は終了状態となり，終了します。

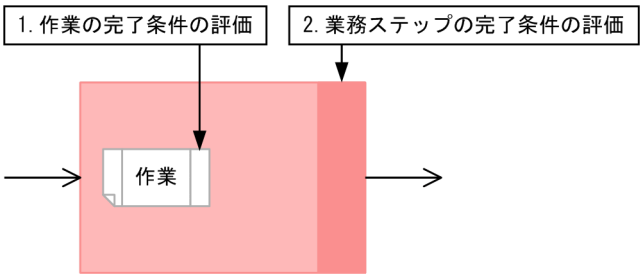
   1 つ以上の発生した作業の完了条件が成立し，かつ業務ステップ内のすべての作業が完了した場合，業務ステップが完了し，案件は次に遷移します。

(b)　作業の評価要求

作業の評価要求のタイミングは，作業の条件を評価する API が発行された時点です。

作業の評価要求を実行したときの処理の順序を次の図に示します。

図 2-30　作業の評価要求を実行したときの処理の順序



＜説明＞

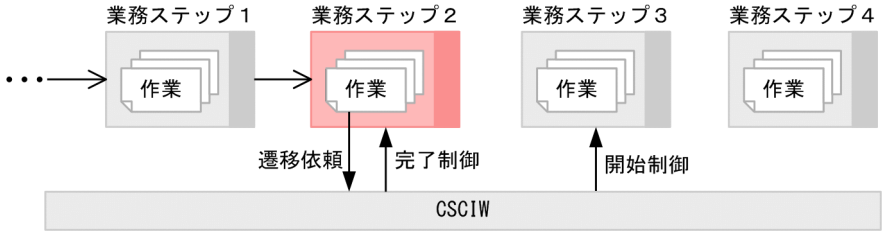
1. 作業の完了条件を評価します。
   完了条件が成立した場合，作業は完了状態となり，2.の処理に進みます。
   完了条件が成立しなかった場合，終了します。
2. 業務ステップの完了条件を評価します。
   処理内容は，「2.5.2(2)(a)　業務ステップの評価要求」の図 2-29 の＜説明＞ 1.の処理になります。

(3)　案件の遷移依頼

案件の遷移依頼処理では，現在実行中の業務ステップを完了し，次の業務ステップを開始します。

案件の遷移依頼の流れを次の図に示します。

図 2-31　案件の遷移依頼の流れ



＜説明＞

1. ユーザ（業務プログラム）は，現在実行中の業務ステップを終了し，次の業務ステップを開始するよう CSCIW に要求します。
2. ユーザ（業務プログラム）からの要求によって，CSCIW は，現在実行中の業務ステップを「完了」の状態にし，次の業務ステップを「実行中」の状態にします。

(4)　アドホック API による業務ステップの生成依頼と遷移依頼

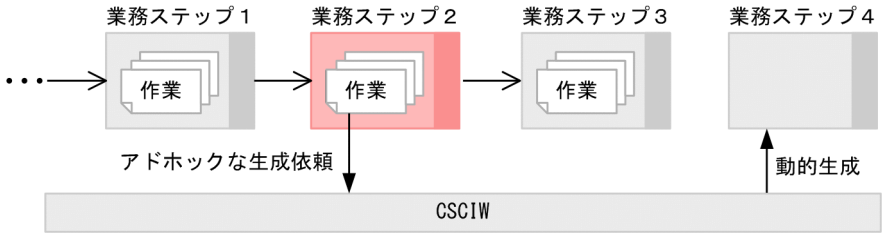
アドホック API を使用して，業務ステップの生成と遷移を動的に制御できます。

(a)　生成依頼

アドホック API による業務ステップの生成依頼処理では，ビジネスプロセスで定義されている任意の業務ステップを動的に生成できます。業務ステップは，「初期」状態で生成されます。

アドホック API による業務ステップの生成依頼を次の図に示します。

図 2-32　アドホック API による業務ステップの生成依頼



＜説明＞

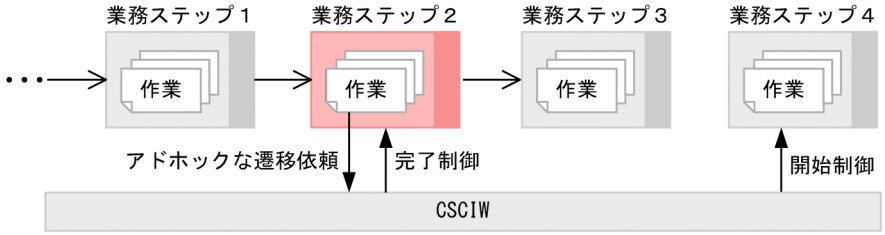
1. ユーザ（業務プログラム）は，動的に業務ステップを生成するよう CSCIW に要求します。
2. ユーザ（業務プログラム）からの要求によって，CSCIW は，指定された業務ステップを生成します。生成した業務ステップの状態は「初期」です。

(b)　遷移依頼

アドホック API による遷移依頼処理では，現在実行中の業務ステップを完了し，遷移定義とは関係なく任意の「初期」状態の業務ステップを開始します。

アドホック API による業務ステップの遷移依頼を次の図に示します。

図 2-33　アドホック API による業務ステップの遷移依頼



＜説明＞

1. ユーザ（業務プログラム）は，現在実行中の業務ステップを終了し，「初期」状態の指定の業務ステップを開始するよう CSCIW に要求します。
2. ユーザ（業務プログラム）からの要求によって，CSCIW は，現在実行中の業務ステップを「完了」の状態にし，指定された業務ステップを「実行中」の状態にします。通常の業務ステップの完了と異なり，次の業務ステップ（業務ステップ 3）には遷移しないで，指定した業務ステップ（業務ステップ 4）に遷移します。

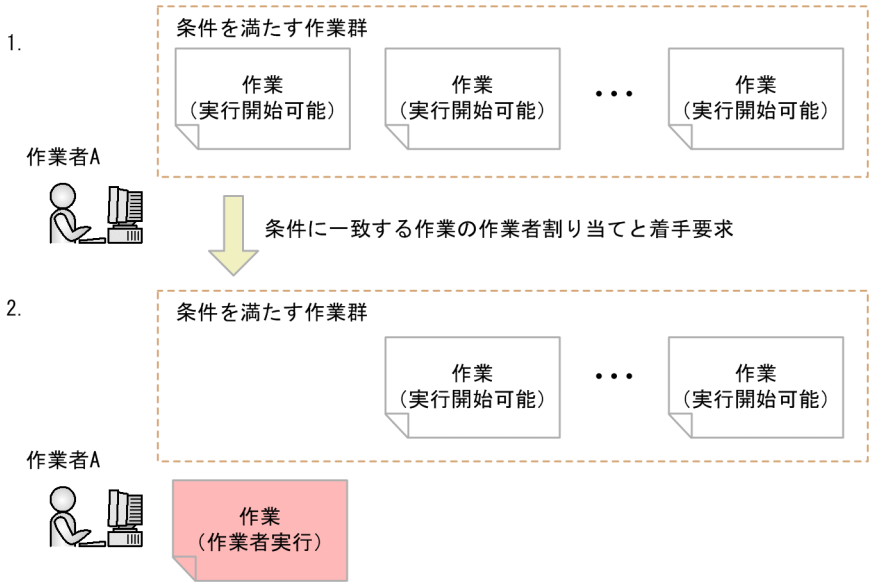
(5)　条件に一致する作業の作業者割り当てと着手

指定された条件に一致する作業群から，ある 1 つの作業を取得し，作業者を割り当てて着手できます。また，割り当てた作業の返却と振り分けルールの再評価もできます。

(a)　条件に一致する作業の作業者割り当てと着手の処理の流れ

条件に一致する作業の作業者割り当てと着手の処理の流れを次の図に示します。

図 2-34　条件に一致する作業の作業者割り当てと着手の処理の流れ



＜説明＞

1. ユーザ（業務プログラム）は，割り当てる作業者名と条件（割り当て元の作業者名や開始日）を指定して，CSCIW に「条件に一致する作業の作業者割り当て」を要求します。
2. ユーザ（業務プログラム）からの要求によって，CSCIW は対象となる作業群（指定された条件と「実行開始可能」状態の両方を満たす作業）を検索します。条件に一致する作業の 1 つを「作業者実行」状態に変更し，作業者を指定された作業者名に変更します。

   変更に失敗した場合は，検索した作業群から別の作業を取得して変更を試みます。変更に成功するか，または対象の作業が存在しなくなるまで，作業の取得と変更を続けます。

   変更が完了した作業を，ユーザ（業務プログラム）に返します。

(b)　作業の返却と振り分けルールの再評価の処理の流れ

作業の返却と振り分けルールの再評価の処理は次の流れで実行します。

＜説明＞

1. ユーザ（業務プログラム）は，条件に一致する作業の作業者割り当てと着手の処理によって作業を取得します。
2. ユーザ（業務プログラム）は，CSCIW に「作業の返却と振り分けルールの再評価」を要求します。
3. ユーザ（業務プログラム）からの要求によって，CSCIW は対象の作業を「作業者実行」状態から「実行開始可能」状態に変更し，振り分けルールを再評価します。

(c)　条件に一致する作業の作業者割り当てに関する注意事項

条件に一致する作業の作業者割り当てに関する注意事項を次に示します。

- 「条件に一致する作業の作業者割り当て」を実行した直後に，トランザクションをコミットしてください。また，トランザクションをコミットする前に，業務テーブルの更新を実行しないでください。

- 「作業の返却と振り分けルールの再評価」を実行する作業には，「条件に一致する作業の作業者割り当て」で取得した作業を指定してください。

2.5.3　組み込み作業での案件の進み方

組み込み作業とは，CSCIW の API を組み合わせることで実現するような複雑な処理を 1 つの作業としてまとめたものです。組み込み作業は，通常の作業と同様にビジネスプロセスで定義できます。

CSCIW が提供している組み込み作業は，**並列作業**です。

組み込み作業の機能概要および案件実行時の動作について次に説明します。

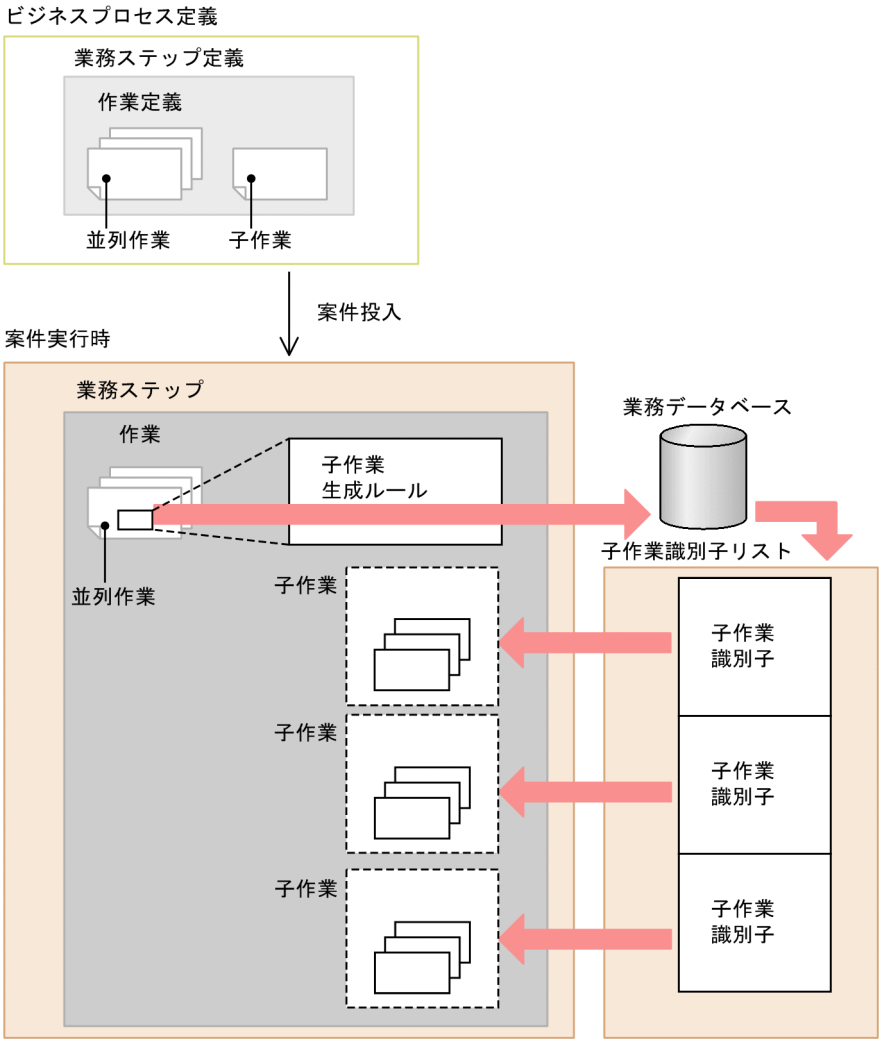
(1)　機能および案件実行時の動作

並列作業は，案件の実行時に複数の作業を自動的に生成する作業です。並列作業をビジネスプロセスで定義する場合は，**子作業**（案件の実行時に生成される作業）と，**子作業生成ルール**（子作業を生成するためのルール）をあわせて定義します。

案件の実行時には，子作業生成ルールを評価して，子作業識別子のリストを取得します。リスト内の識別子を子作業名として，子作業が生成されます。

並列作業の概念（RDB 検索の例）を次の図に示します。

図 2-35　並列作業の概念（RDB 検索の例）



＜説明＞

1. 並列作業の業務ステップが開始された場合，または設定されている発生条件が満たされた場合，並列作業が実行され，子作業生成ルールを評価します。
2. 子作業生成ルールの記述内容に基づいて業務データベースが検索され，子作業識別子リストを取得します。
3. 並列作業は，子作業識別子リストに対応させて子作業を生成します。
4. 生成された子作業の動作は，通常の作業と同じです。また，並列作業の完了条件の評価タイミングも通常の作業と同じです。評価時に，完了条件が満たされた場合は，並列作業の状態が「実行済み」に遷移します。完了条件が満たされていない場合は，項番 2.から項番 4.の処理が繰り返されます。

●子作業生成ルールが評価されるタイミング

- 並列作業が実行された時点
- 業務ステップの完了条件が偽に評価された時点
- 並列作業の完了条件が偽に評価された時点

(2) 補足および注意事項

(a) 子作業生成ルールの種別

子作業生成ルールの種別に指定できるのは，「RDB 検索」または「Java オブジェクト呼び出し」のどちらかです。

(b) 識別子リストから子作業を生成するときの規則

- リスト内の識別子が，生成済みの子作業の識別子と一致しない場合は，その識別子を持つ子作業を生成します。
- リスト内の識別子が，生成済みの子作業の識別子と一致し，その子作業が実行中の場合は，子作業を生成しません。
- リスト内の識別子が，生成済みの子作業の識別子と一致し，その子作業が完了している場合は，再度その識別子を持つ子作業を生成します。
- リスト中に同一識別子が複数ある場合，その識別子を持つ子作業を 1 つだけ生成します。
- リスト内の識別子が 0 件の場合は，子作業を生成しません。

(c) 子作業生成ルールを記述するときの注意事項

- 子作業生成ルールは，その子作業生成ルールの適用時に，必要十分な識別子リストが作成されるように記述してください。子作業と識別子の対応が取れていない場合，不要に子作業が作成されるなどといったことが発生するためご注意ください。
- 生成する子作業に完了条件を持たせている場合は，ユーザの判断なしに子作業が完了してしまうことがありますので，子作業生成ルールを記述する際は，子作業の動作を考慮して定義してください。
- 生成する子作業の完了条件は，子作業生成ルールによる作業の発生と同時に成立しないように考慮して定義してください。同時に成立した場合，子作業の発生と完了が繰り返され，その状態が続くと CIWFatalException 例外が発生します。

(d) 子作業名の長さについて

子作業生成ルールの評価によって返される子作業名の長さは，1 バイト以上で，<SYSTEMID>_WORK_ITEM テーブルの Name カラムで定義したバイト数以内にしてください。範囲外の長さの子作業名が返された場合は，CIWTransitionFailedException 例外が発生します。

2.6　作業アプリケーションの呼び出し

この節では，作業アプリケーション定義に従い，自動で業務プログラムを呼び出す機能について説明します。

2.6.1　機能および作業アプリケーション呼び出し時の動作

作業アプリケーション呼び出しの機能には，API（案件投入 API や作業完了 API など）と同期を取って呼び出される「Java オブジェクト呼び出し」機能と，API と非同期に呼び出される「REST アプリケーション呼び出し」機能があります。

「Java オブジェクト呼び出し」機能については，「2.8　Java オブジェクト呼び出し」を参照してください。「REST アプリケーション呼び出し」機能については，「付録 H　BPMN 連携機能を使用しない場合のアプリケーション呼び出しサービス」を参照してください。

> **ヒント**
>
> このマニュアルでは，BPMN 連携機能を使用しない場合のアプリケーション呼び出しサービスについて説明します。BPMN 連携機能を使用する場合のアプリケーション呼び出しサービスについては，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。

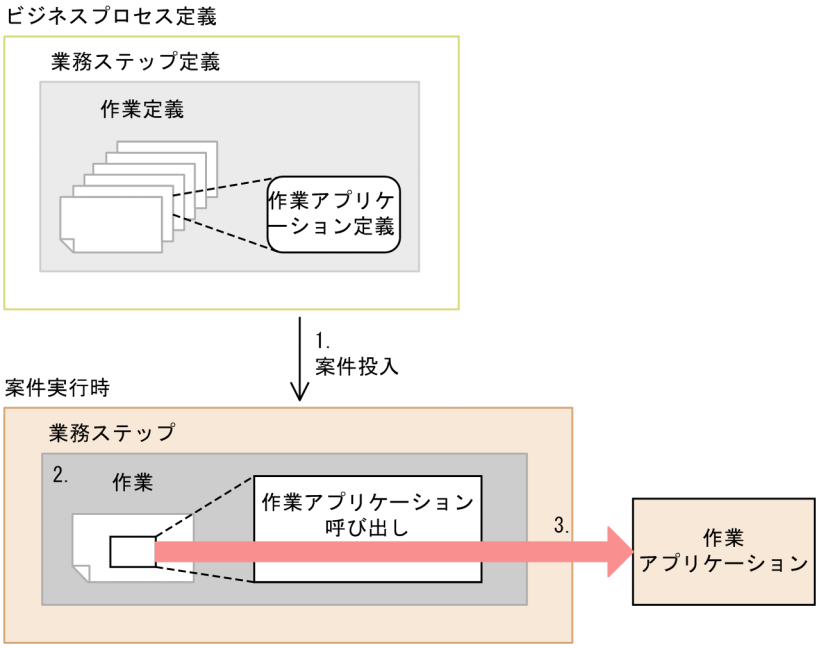
作業アプリケーションが呼び出されるのは，作業の状態が実行開始可能（Ready）になったときです。作業の状態の変化と作業アプリケーション呼び出しの動作について次の表に示します。

表 2-1　作業の状態の変化と作業アプリケーション呼び出しの動作

作業の状態		動作
変化前	変化後	
初期（initial）	実行開始可能（Ready）	作業アプリケーションを新たに起動する要求として，作業アプリケーションを呼び出します。
作業者実行（performing）	実行開始可能（Ready）	すでに起動している作業アプリケーションを再起動する要求として，作業アプリケーションの状態に関係なく必ず呼び出します。 作業アプリケーションの再起動と同じです。
実行開始不可（disabled）	実行開始可能（Ready）	起動を抑止されていた作業アプリケーションを回復した要求として，作業アプリケーションの状態に関係なく必ず呼び出します。

作業アプリケーション呼び出しの概要を次の図に示します。

図 2-36　作業アプリケーション呼び出しの概要



図中の番号は，次の説明の番号に対応しています。

＜説明＞

1. 案件を投入します。
2. 作業のインスタンスが生成されます。
3. 案件が作業アプリケーション呼び出しが定義されている作業に遷移したとき，CSCIW は作業アプリケーションを呼び出します。

2.6.2　作業アプリケーションの同期呼び出し

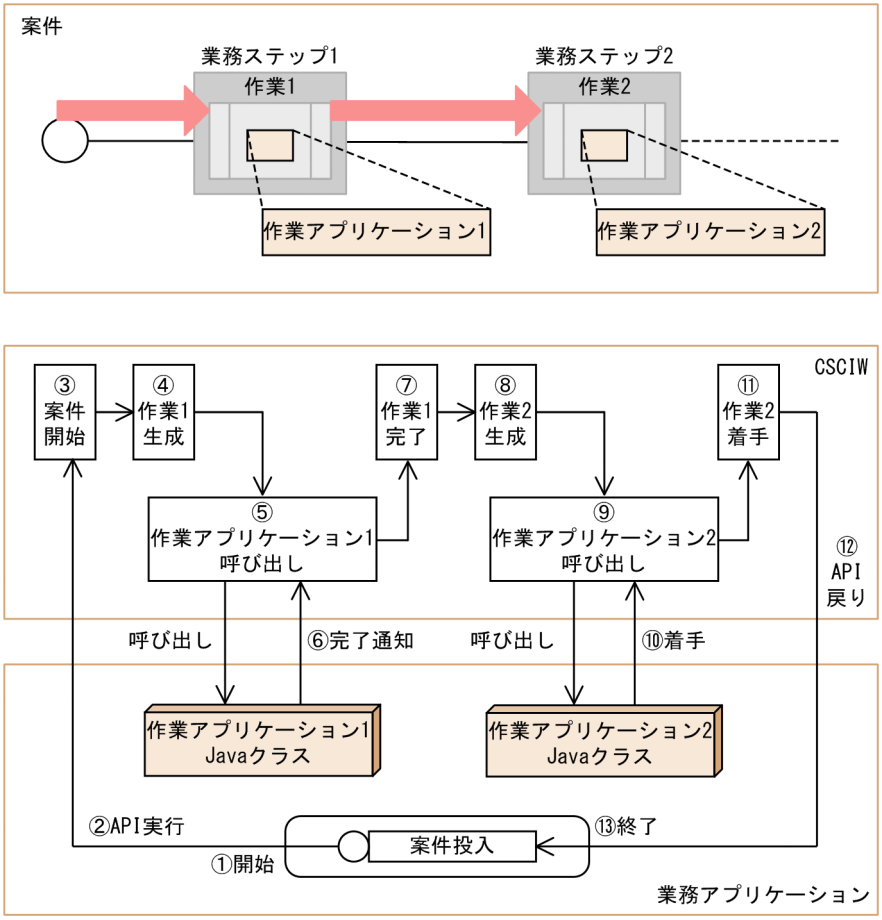
作業アプリケーションの同期呼び出しをする機能として「Java オブジェクト呼び出し」機能があります。

作業アプリケーションの同期呼び出しは，API（案件投入 API や作業完了 API など）と同期を取り，作業アプリケーションを起動します。CSCIW と関連づけたコネクションを渡すため，API と同じトランザクションで処理を実行できます。API と同じトランザクションで処理を実行したい場合，または比較的処理時間の短い業務を実行する場合に使用します。

なお，処理中にエラーが発生した場合，作業アプリケーションの処理もロールバックされます。再実行するときは API を再実行する必要があります。

作業アプリケーション同期呼び出しの概要を次の図に示します。

図 2-37　作業アプリケーション同期呼び出しの概要



図中の番号は，次の説明の番号に対応しています。

＜説明＞

1. 業務アプリケーションがトランザクションを開始します。
2. 業務アプリケーションが案件投入 API を実行します。
3. CSCIW が案件を生成，開始します。
4. ビジネスプロセス定義に従い，「作業 1」を生成します。
5. 「作業 1」に作業アプリケーションの呼び出しが定義されている場合，作業アプリケーション（「作業アプリケーション 1」）の呼び出しを実行します。
6. 作業アプリケーション（「作業アプリケーション 1」）は業務処理を行い，戻り値で作業の完了通知を行います。
7. 作業アプリケーション（「作業アプリケーション 1」）から作業の完了通知を受けた CSCIW は「作業 1」を完了します。
8. ビジネスプロセス定義に従い，業務ステップを遷移し，新たに「作業 2」を生成します。
9. 「作業 2」に作業アプリケーションの呼び出しが定義されている場合，作業アプリケーション（「作業アプリケーション 2」）の呼び出しを実行します。

10. 作業アプリケーション（「作業アプリケーション 2」）は，業務処理を行い，戻り値で作業の着手を行います。
11. 作業アプリケーション（「作業アプリケーション 2」）から作業の着手の指示を受けた CSCIW は「作業 2」を着手します。
12. 作業が完了しないため，遷移を終了して，業務アプリケーションへ戻ります。
13. 業務アプリケーションはトランザクションをコミットします。

同期呼び出しの場合，作業アプリケーションも含め，すべて同一トランザクションで処理を行います。上の説明では，1.～12.が 1 つのトランザクションとなります。処理の途中（例えば，10.の作業着手処理）でエラーが発生した場合，案件投入がエラーとなります。すべての処理がロールバックされるため，案件投入前の状態に戻ります。

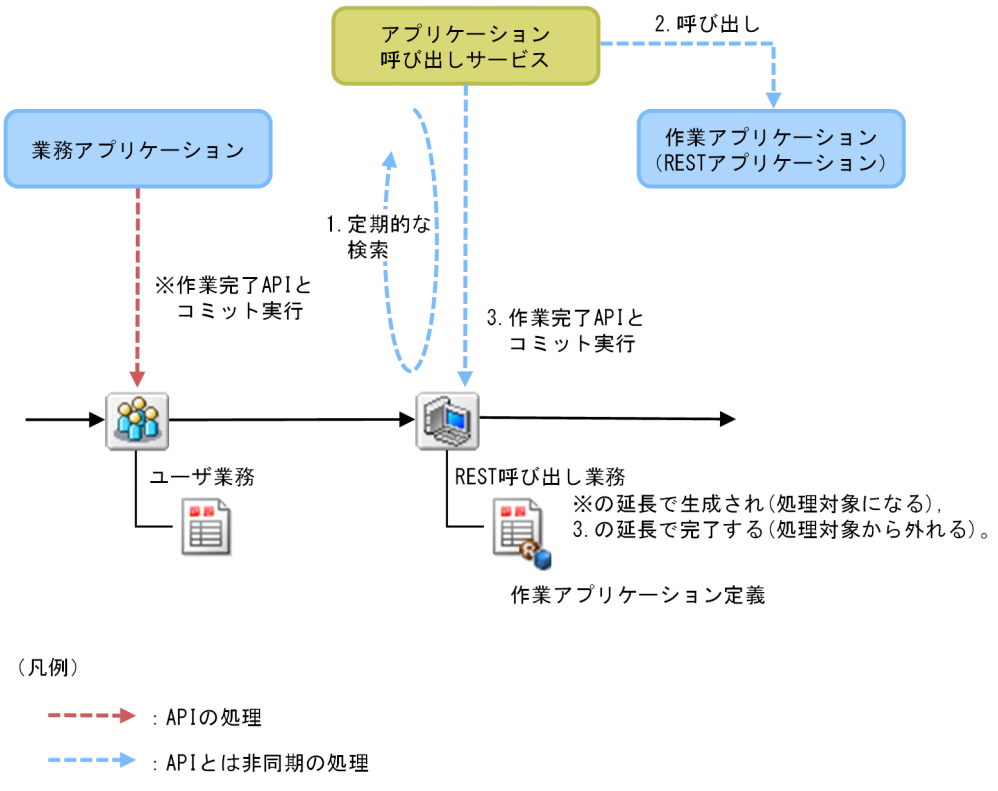
2.6.3 作業アプリケーションの非同期呼び出し

作業アプリケーションの非同期呼び出しをする機能として「REST アプリケーション呼び出し」機能があります。

作業アプリケーションの非同期呼び出しは，API（案件投入 API や作業完了 API など）の処理と非同期に，アプリケーション呼び出しサービスが作業アプリケーションを呼び出します。非同期のため，API とアプリケーション呼び出しサービスのトランザクションは別になります。

作業アプリケーションの非同期呼び出しの概要を次の図に示します。

図 2-38 作業アプリケーションの非同期呼び出しの概要



図中の番号は，次の説明の番号に対応しています。

＜説明＞

1. アプリケーション呼び出しサービスは，処理対象の作業「REST 呼び出し業務」が生成されているか定期的に検索します。作業「REST 呼び出し業務」は，アプリケーション呼び出しサービスと非同期に動作する業務アプリケーションが，手前の作業「ユーザ業務」に対して作業完了 API とトランザクションのコミットを実行した（図中の※）延長で生成されます。
2. アプリケーション呼び出しサービスは，処理対象の作業「REST 呼び出し業務」が生成されていることを検知すると，作業アプリケーション（REST アプリケーション）を呼び出します。
3. 作業アプリケーション（REST アプリケーション）の呼び出しが成功すると，アプリケーション呼び出しサービスは，処理対象の作業「REST 呼び出し業務」に対して作業完了 API とトランザクションのコミットを実行します。これによって作業「REST 呼び出し業務」が処理対象から外れます。

2.7 RDB 検索

ここでは，RDB 検索による振り分けルール定義の振り分けルール記述形式，およびビジネスプロセス定義のデータ条件記述形式と生成ルール記述形式について説明します。

2.7.1 振り分けルール記述の形式

RDB 検索による振り分けルール定義の振り分けルール記述形式および注意事項を次に示します。

振り分けルール記述形式

作業者を特定するための検索条件を SQL 文で記述します。ワーク管理システムでは，作業者を決定するときに記述された SQL 文を作業者データベースに対して実行することで作業者情報を取得します。

SQL 文の記述の一部として，ワーク管理システムが提供する組み込み変数を使用できます。記述に使用された組み込み変数は，SQL 文を実行する前にワーク管理システムによって値に置き換えられ，置き換えられた結果の SQL 文が実行されます。

ワーク管理システムが提供する組み込み変数については，マニュアル「uCosminexus Service Coordinator Interactive Workflow AP 開発ガイド」を参照してください。

2.7.2 データ条件記述の形式

RDB 検索によるデータ条件の評価のデータ条件記述形式および注意事項を次に示します。

データ条件記述形式

データ条件を評価するための検索条件を SQL 文で記述します。ワーク管理システムでは，データ条件を評価する際に，記述された SQL 文を業務データベースに対して実行することでデータ条件の真偽を決定します。SQL の実行結果が存在する場合に真となり，SQL の実行結果が存在しない場合は偽となります。また，SQL の実行結果が存在しない場合に真となり，SQL の実行結果が存在する場合に偽となるように，データ条件の真偽の決定を逆転させる指定もできます。

SQL 文の記述の一部として，ワーク管理システムが提供する組み込み変数を使用できます。記述に使用された組み込み変数は，SQL 文を実行する前にワーク管理システムによって値に置き換えられ，置き換えられた結果の SQL 文が実行されます。

ワーク管理システムが提供する組み込み変数については，マニュアル「uCosminexus Service Coordinator Interactive Workflow AP 開発ガイド」を参照してください。

2.7.3 生成ルール記述の形式

RDB 検索による生成ルール定義の生成ルール記述形式および注意事項を次に示します。

生成ルール記述形式

識別リストを取得する SQL 文を記述します。ワーク管理システムでは，生成ルールを評価する際に，記述された SQL 文を業務データベースに対して実行することで，識別子リストを取得します。

問い合わせ結果表の先頭列のすべての行に格納された文字列を，識別子リストとします。

SQL 文の記述の一部として，ワーク管理システムが提供する組み込み変数を使用できます。記述に使用された組み込み変数は，SQL 文を実行する前にワーク管理システムによって値に置き換えられ，置き換えられた結果の SQL 文が実行されます。

ワーク管理システムが提供する組み込み変数については，マニュアル「uCosminexus Service Coordinator Interactive Workflow AP 開発ガイド」を参照してください。

2.8　Java オブジェクト呼び出し

この節では，Java オブジェクト呼び出し機能について説明します。

Java オブジェクト呼び出し機能は，CSCIW が提供するインタフェースを実装した Java クラスのオブジェクトを同期で呼び出す機能です。

Java オブジェクト呼び出しは，条件，生成ルール，振り分けルールおよび作業アプリケーションの定義で定義できます。

> **ヒント**
>
> このマニュアルでは，条件，生成ルール，振り分けルール，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。

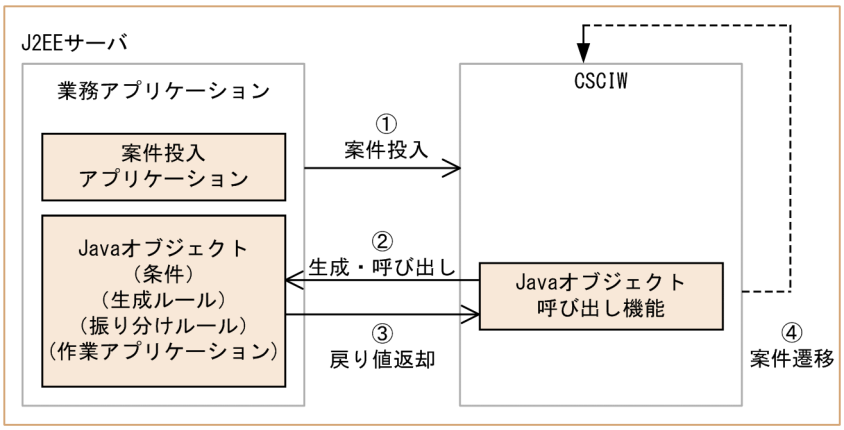
条件，生成ルールおよび振り分けルールでは，Java オブジェクトの戻り値を使用して，データ条件の判定，子作業生成および作業者の決定をします。

作業アプリケーションでは，Java オブジェクトの戻り値に従って，作業の状態を「ready（実行開始可能）」から「performing（作業者実行）」，または「executed（実行済み）」などに変更します。

Java オブジェクト呼び出し機能は J2EE アプリケーションおよび Java アプリケーションのどちらからも使用できます。

Java オブジェクト呼び出しの概要を次の図に示します。

図 2-39　Java オブジェクト呼び出しの概要



図中の番号は，次の説明の番号に対応しています。

＜説明＞

1. 案件を投入します。
2. Java オブジェクト呼び出しが定義されている分岐ノード，業務ステップや作業に遷移したとき，CSCIW は定義された Java クラスのインスタンスを生成し，オブジェクトを呼び出します。
3. Java オブジェクトは，処理を行い，CSCIW に戻り値を返します。
4. CSCIW は Java オブジェクトからの戻り値に従い，案件を遷移します。

3 システム構成に関する前提知識

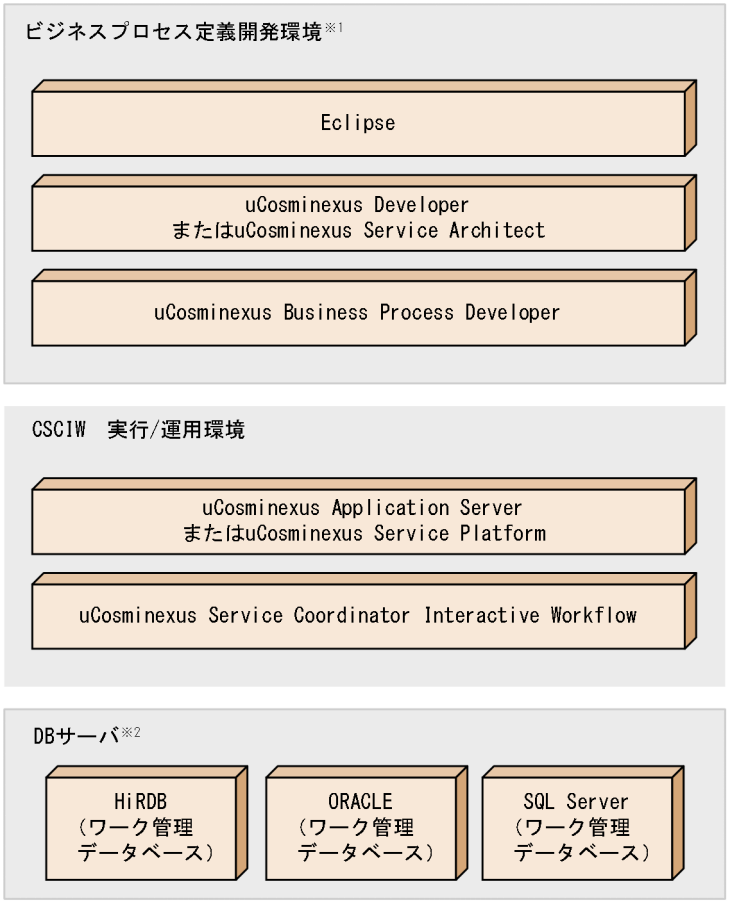
この章では，ワーク管理システムのシステム構成に関する前提知識について説明します。

3.1　ワーク管理システムのプログラム構成

この節では，ワーク管理システムを構成するプログラムと，ワーク管理システムの前提プログラムおよび関連プログラムについて説明します。

ワーク管理システムのプログラム構成を次の図に示します。

図 3-1　ワーク管理システムのプログラム構成



注　OSは，DBMSを含むすべての環境で必要です。
注※1　ビジネスプロセス定義および振り分けルール定義を開発する場合に必要です。
注※2　HiRDB，ORACLE，またはSQL Serverのどれかを使用します。

ヒント

BPMN 連携機能を使用する場合の CSCIW のワーク管理システムのプログラム構成については，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　ワーク管理システムの前提プログラム

ワーク管理システムを運用するための前提プログラムは次のとおりです。

(1) CSCIW 実行/運用環境

●OS

次のどちらかを使用します。

- Windows
- UNIX（AIX または Linux）

●アプリケーションサーバ

CSCIW の動作環境として使用します。必要なプログラムは，次のどちらかです。

- uCosminexus Application Server
- uCosminexus Service Platform

●Web ブラウザ

CSCIW 案件運用操作の動作環境として使用します。使用できる Web ブラウザは，次に示すどれかです。

- Chrome
- Firefox
- Microsoft Edge

(2) DB サーバ

●DBMS

ワーク管理データベースの動作環境として使用します。必要なデータベースは，次に示すどれかです。

- HiRDB
- ORACLE
- SQL Server

HiRDB の場合，SQL を実行するには，HiRDB SQL Executer が必要です。

> **メモ**
>
> CSCIW の各バージョンでサポートしている DBMS については，リリースノートを参照してください。

3.1.2 ワーク管理システムの関連プログラム

ワーク管理システムを運用するときに連携できるプログラムを次に示します。

(1) ビジネスプロセス定義開発環境

●OS

Windows を使用します。

●uCosminexus 開発環境

ビジネスプロセス管理ツール（CSCIW-Definer）の動作環境として使用します。必要なプログラムは，次のどちらかです。

- uCosminexus Developer
- uCosminexus Service Architect

●ビジネスプロセス管理ツール（CSCIW-Definer）

uCosminexus Business Process Developer を Eclipse に組み込んで使用します。

(2) その他

●データベースプログラム

業務データベースのデータを業務データとして利用する場合に使用します。次のどれかを使用してください。

- HiRDB
- ORACLE
- SQL Server

> **メモ**
>
> CSCIW の各バージョンでサポートしている DBMS については，リリースノートを参照してください。

●uCosminexus 開発環境のプログラム

業務プログラムを開発する場合の動作環境として使用します。次のどちらかを使用してください。

- uCosminexus Developer
- uCosminexus Service Architect

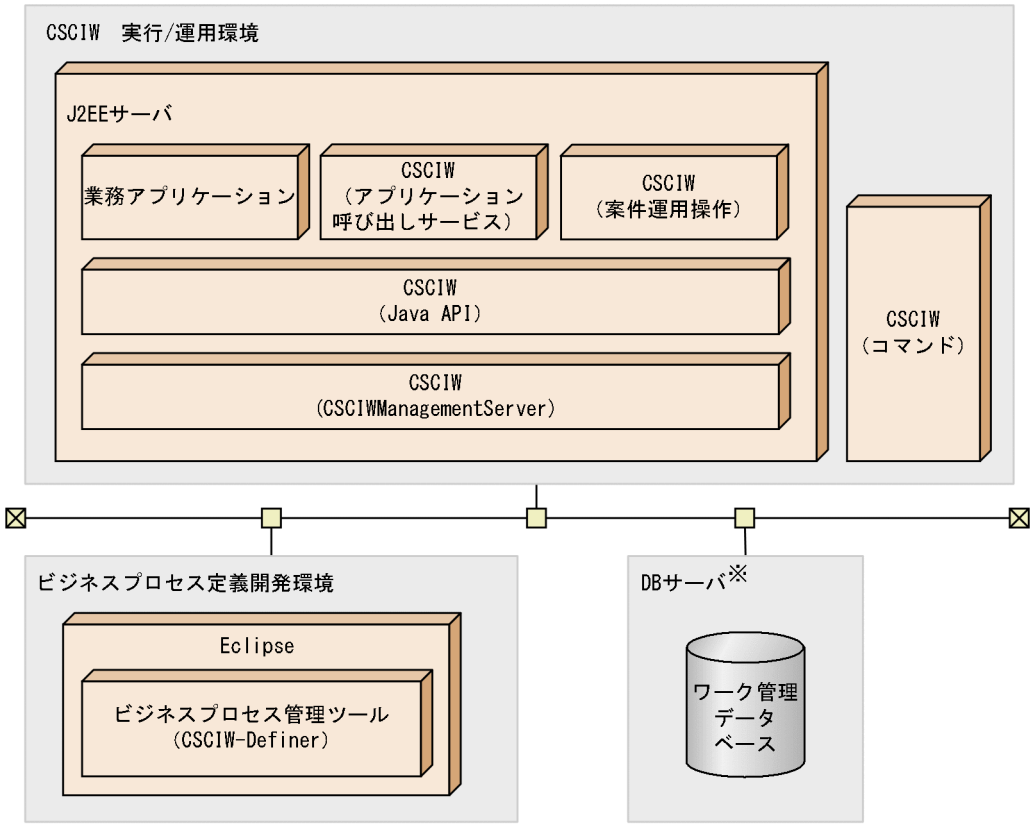
3.2　ワーク管理システムのシステム構成モデル

この節では，ワーク管理システムを運用する上でのシステム構成モデルについて説明します。

3.2.1　基本システム構成

ワーク管理システムを運用する上での基本的なシステム構成モデルを示します。

図 3-2　ワーク管理システムの基本的なシステム構成モデル



注※　HiRDB，ORACLE，またはSQL Serverのどれかを組み合わせて使用します。

システムを構成する上で，次のことに注意してください。

- ワーク管理システムの中で，ワーク管理データベースは業務データベースと同一の DBMS 上で運用することをお勧めします。同一の DBMS 上で運用し，CSCIW の JavaAPI が提供する DB コネクション共有管理機能を使用することで，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。

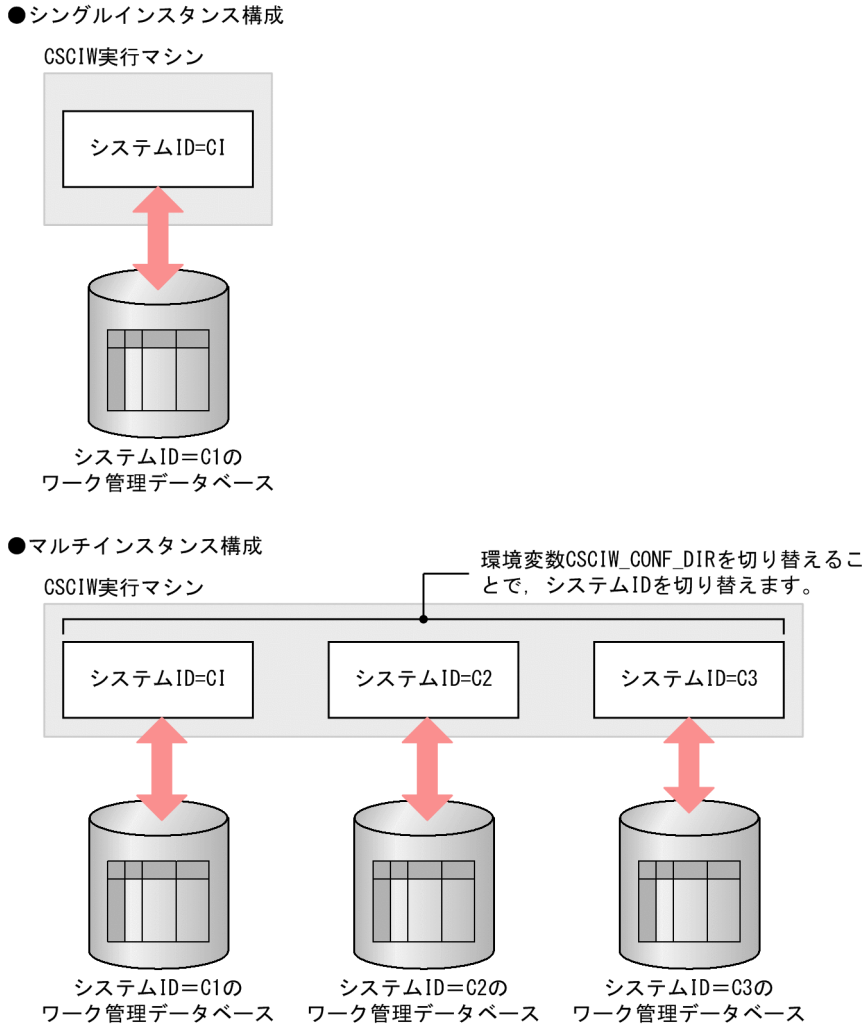
3.2.2　マルチインスタンス構成

マルチインスタンスは，1 台のマシンに，複数の異なるシステム ID を設定して，互いに別々の環境として利用できるシステム構成です。ただし，JavaVM では 1 プロセスで設定できるシステム ID は 1 つです。

CSCIW は，インストールした起動環境に対し，必ず環境設定をして利用します。マルチインスタンス構成の場合，同一マシンに設定された複数のシステム ID は，環境変数 CSCIW_CONF_DIR で切り替えます。

シングルインスタンスとマルチインスタンスの違いを次の図に示します。

図 3-3　シングルインスタンスとマルチインスタンスの違い



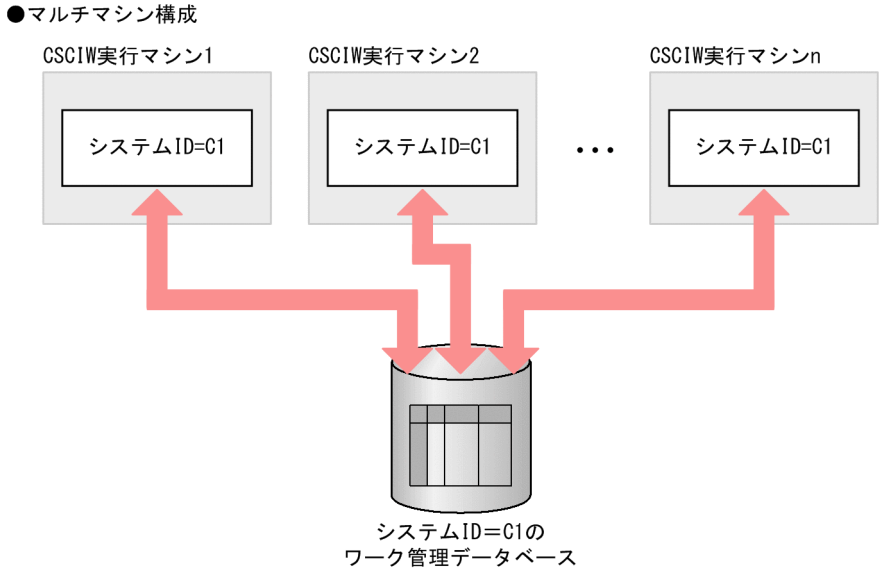
また，CSCIW では，1 台のマシンに設定できるインスタンスの数に制限はありません。CPU，メモリ，ハードディスクなどの資源が許す限りインスタンスを増やせます。

3.2.3　マルチマシン構成

マルチマシンは，複数の異なるマシン上で，同一のシステム ID によって CSCIW を運用するシステム構成です。CSCIW では，マルチマシン構成にする場合，マシンの台数に制限はありません。ただし，複数のマシンで IP アドレスの重複はできません。

マルチマシン構成の例を次の図に示します。

図 3-4　マルチマシン構成



3.2.4　マルチプロセス構成

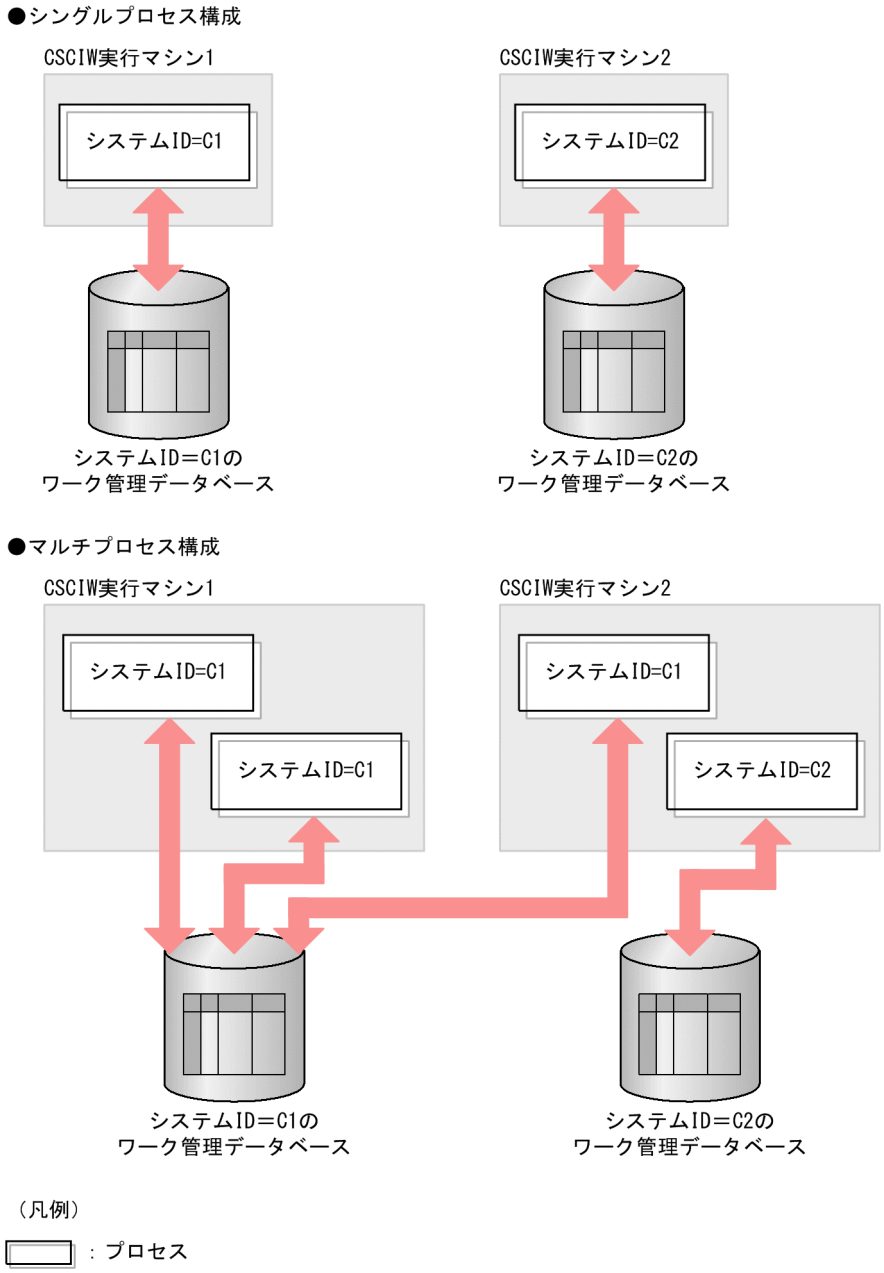
マルチプロセスは，同一のシステム ID に設定されている CSCIW のプロセスを，同時に複数起動できるシステム構成です。

マルチプロセス構成の CSCIW では，1 台のマシンで，同時に複数のプロセスを起動できます。

CSCIW は，プロセスを多重化することで，一方のプロセスがダウンした場合でも，もう一方のプロセスで処理を続行でき，より信頼度の高いシステムとなります。

マルチプロセスとシングルプロセスの違いを次の図に示します。

図 3-5　シングルプロセスとマルチプロセスの違い



3.2.5　クラスタ構成

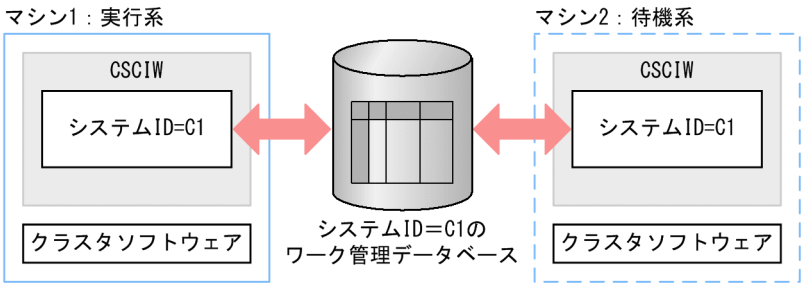
クラスタ構成は，通常運用するマシン環境（実行系）に待機系のマシン環境を加えたシステム構成です。CSCIW では，Cosminexus がサポートするクラスタソフトウェアを使用して，クラスタ構成を構築できます。クラスタ構成にすることで，通常運用しているマシンがダウンしても，待機系のマシンが自動的に稼働し，ワーク管理システムの処理を続行させることができます。

Cosminexus がサポートするクラスタソフトウェアについては，マニュアル「Cosminexus アプリケーションサーバ 機能解説 運用／監視／連携編」を参照してください。

クラスタ構成の例を次の図に示します。

図 3-6　クラスタ構成

●クラスタ構成



4 ワーク管理システムの設定

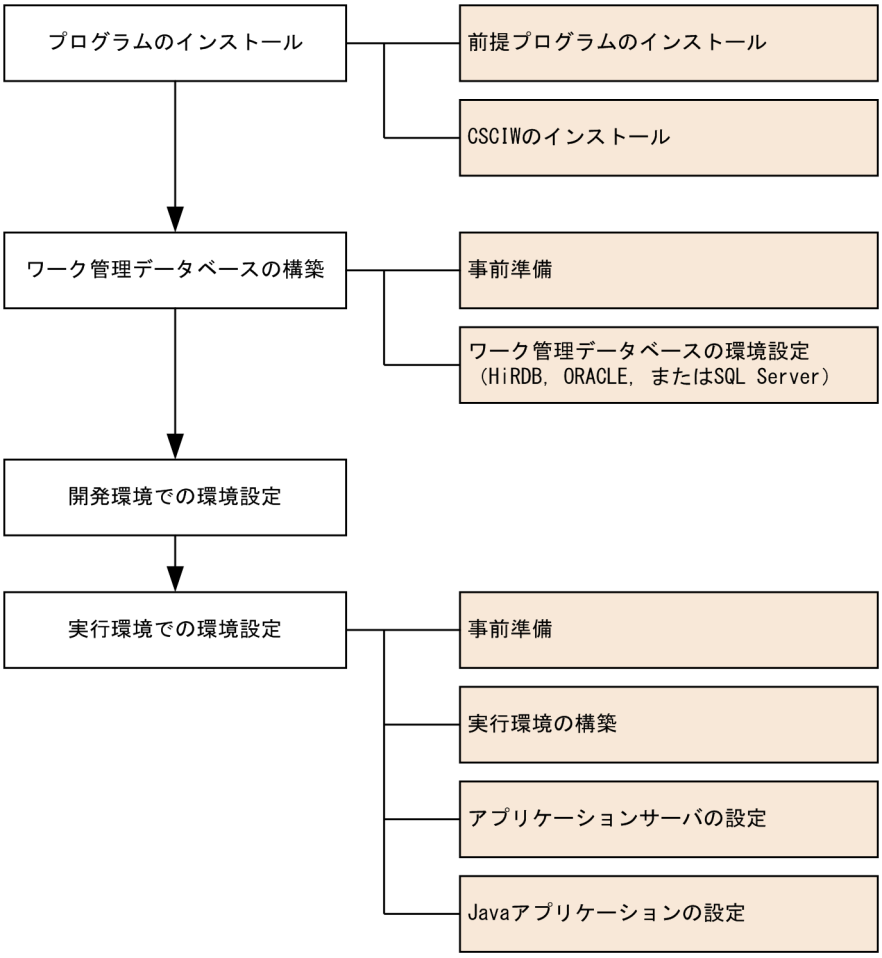
この章では，ワーク管理システムの環境設定について説明します。

4.1　システム設定の流れ

この節では，ワーク管理システムを設定する流れについて説明します。

ワーク管理システムを設定する流れを，次の図に示します。

図 4-1　ワーク管理システムの設定の流れ



上の図に示したワーク管理システムの設定の流れに従って，次の節以降で設定方法を説明します。

ヒント

BPMN 連携機能を使用する場合のシステム設定の流れおよび設定方法については，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。

4.2　プログラムのインストール

この節では，ワーク管理システムを運用するときに利用するプログラムのインストール方法について説明します。

4.2.1　前提プログラムのインストール

ワーク管理システムを構築するときにインストールが必要なプログラムについて説明します。

(1)　CSCIW

CSCIW の実行環境の構築および運用するときに必要な前提プログラムを次の表に示します。

表 4-1　CSCIW の前提プログラム

機能	プログラム	対象 OS	
		Windows	UNIX
OS※1	Windows Server	●	−
	Windows		
	Linux	−	○
	AIX		
アプリケーションサーバ※2	uCosminexus Application Server	●	●
	uCosminexus Service Platform		
Web ブラウザ	Chrome	●	●
	Firefox		
	Microsoft Edge		

（凡例）

●：前提プログラムとして，どれか 1 つまたは 1 組をインストールする必要があります。

○：前提プログラムとして，どちらか 1 つをインストールする必要があります。

−：インストールの必要はありません。

注※1

OS の詳細なバージョンについては，リリースノートを参照してください。

注※2

DBConnector のトランザクションサポートレベルは，LocalTransaction だけを対象としています。

(a)　OS

CSCIW に必要な OS です。

●Windows Server または Windows

Windows Server または Windows のインストールについては，該当するバージョンの各マニュアルのインストールに関する記述を参照してください。

●AIX または Linux

AIX または Linux のインストールについては，それぞれのマニュアルのインストールに関する記述を参照してください。

(b) アプリケーションサーバ

Cosminexus のアプリケーションサーバ機能を利用するために必要なプログラムです。

●uCosminexus Application Server，または uCosminexus Service Platform

uCosminexus Application Server のインストールおよびシステム構築については，マニュアル「Cosminexus アプリケーションサーバ システム構築・運用ガイド」を，また uCosminexus Service Platform のインストールについては，マニュアル「Cosminexus 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」をそれぞれ参照してください。

(c) Web ブラウザ

CSCIW の案件運用操作機能を使用するために必要なプログラムです。

●Chrome

Chrome のインストールについては，Chrome のドキュメントのインストールに関する記述を参照してください。

●Firefox

Firefox のインストールについては，Firefox のドキュメントのインストールに関する記述を参照してください。

●Microsoft Edge

Microsoft Edge のインストールについては，Microsoft Edge のドキュメントのインストールに関する記述を参照してください。

(2) DB サーバ

ワーク管理データベースとして DB サーバを運用するときに必要な前提プログラムを次の表に示します。

表 4-2 DB サーバの前提プログラム

機能	プログラム	対象 OS	
		Windows	UNIX
DBMS	HiRDB Server	●	●
	Oracle		
	SQL Server		

機能	プログラム	対象 OS	
		Windows	UNIX
SQL 実行	HiRDB SQL Executer	△	△

（凡例）

●：前提プログラムとして，どれか 1 つまたは 1 組をインストールする必要があります。

△：DBMS に HiRDB を使用する場合，インストールが必要です。

注

DB サーバの前提プログラムの詳細なバージョンについては，リリースノートを参照してください。

(a) DBMS

ワーク管理システムを運用する場合に使用できる DBMS です。

●HiRDB Server

ワーク管理データベースに必要なプログラムです。

HiRDB のインストールについては，次に示すマニュアルを参照してください。

- HiRDB システム導入・設計ガイド（UNIX(R)用）
- HiRDB システム導入・設計ガイド（Windows(R)用）
- HiRDB UAP 開発ガイド

●Oracle

ワーク管理データベースに必要なプログラムです。

ORACLE のインストールについては，ORACLE のマニュアルのインストールに関する記述を参照してください。

●SQL Server

ワーク管理データベースに必要なプログラムです。

SQL Server のインストールについては，SQL Server のマニュアルのインストールに関する記述を参照してください。

(b) SQL 実行

HiRDB データベースに対して，会話形式で SQL を実行させる場合に必要なプログラムです。

●HiRDB SQL Executer

HiRDB SQL Executer については，マニュアル「HiRDB 解説」を参照してください。

(3) ビジネスプロセス定義開発環境

ビジネスプロセス定義を開発するときに必要な前提プログラムを次に示します。

表 4-3　ビジネスプロセス定義開発環境の前提プログラム

機能	プログラム	必要性
OS	Windows Server	●
	Windows	
Cosminexus 開発環境	uCosminexus Developer	●
	uCosminexus Service Architect	
ビジネスプロセス管理ツール（CSCIW-Definer）	Eclipse	○
	uCosminexus Business Process Developer	○

（凡例）

○：前提プログラムとして，インストールが必要です。

●：前提プログラムとして，どれか 1 つまたは 1 組をインストールする必要があります。

(a)　OS

ビジネスプロセス定義開発に必要な OS です。

●Windows Server または Windows

Windows Server または Windows のインストールについては，該当するバージョンの各マニュアルのインストールに関する記述を参照してください。

(b)　Cosminexus 開発環境

Cosminexus の開発環境を利用するために必要なプログラムです。

●uCosminexus Developer，または uCosminexus Service Architect

uCosminexus Developer，または uCosminexus Service Architect のインストールおよび環境構築については，マニュアル「Cosminexus アプリケーションサーバ アプリケーション開発ガイド」を参照してください。

(c)　ビジネスプロセス管理ツール（CSCIW-Definer）

ビジネスプロセス定義，および振り分けルール定義を開発する場合に必要なプログラムです。

●Eclipse

Eclipse のインストールおよび環境構築については，マニュアル「Cosminexus アプリケーションサーバ アプリケーション開発ガイド」を参照してください。

●uCosminexus Business Process Developer

uCosminexus Business Process Developer のインストールについては，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。

4.2.2　CSCIW のインストール

CSCIW のインストール方法について説明します。

(1)　Windows の場合

インストーラーを使用して，CSCIW をインストールする方法を次に示します。

1. Administrators グループのユーザでログインする

2. Setup.exe を起動する
   ユーザ名，会社名を入力するダイアログが表示されます。

3. ユーザ名，会社名を入力する

4. [次へ] ボタンをクリックする
   インストール先フォルダを指定するダイアログが表示されます。

5. インストール先フォルダを指定する
   指定したフォルダの下にインストールされます。
   なお，インストール先のフォルダには，次のパスは指定できません。
   - ネットワークドライブ
   - 2 バイトコードを含むパス
   - ¥ / : , ; * ? ” < > | $ % & ^ ’ ! ( ) = + { } @ [ ]を含むパス※
   - 半角 129 文字を超えるパス
   - 末尾が¥のパス（C:¥など）

   注※
   Windows の場合は，デフォルト値を除きます。
   また，プログラムフォルダ名にも指定できません。「;」と「/」は，すべての入力情報に指定できません。

6. [次へ] ボタンをクリックする
   インストール先フォルダを確認するためのダイアログが表示されます。インストール先フォルダを確認してください。

7. [次へ] ボタンをクリックする
   インストールが始まります。
   インストールが終了すると，インストールが終了したことを通知するダイアログが表示されます。

8. [完了] ボタンをクリックする
   インストールが終了します。

重要

CSCIW をインストールすると，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もインストールされます。

(2) UNIX の場合

Hitachi PP Installer を使用して，/opt/hitachi/CSCIW ディレクトリに CSCIW をインストールする方法を次に示します。

1. root 権限でログインする

2. mkdir /mnt/cdrom を実行し，マウント用の/mnt/cdrom ディレクトリを作成する

   下線部分は環境によって異なるため，使用する環境に合わせて変更してください。マウント用ディレクトリを作成済みの場合は，この操作は不要です。

3. CSCIW の CD-ROM を CD-ROM ドライブにセットする

4. mount を実行する

   Linux の場合

   mount -r -o mode=0544 /dev/cdrom /mnt/cdrom を実行する

   AIX の場合

   mount -r -v cdrfs /dev/cd0 /mnt/cdrom を実行する

   下線部分は環境によって異なるため，使用する環境に合わせて変更してください。

5. /mnt/cdrom/<OS 名>/setup /mnt/cdrom を実行する

   下線部分は環境によって異なるため，使用する環境に合わせて変更してください。また，<OS 名>の部分は使用する OS によって異なるため，ls コマンドで確認してください。

   なお，/etc/hitachi_setup コマンドがすでにマシン上に展開されている場合には，/etc/hitachi_setup -i /mnt/cdrom を実行しても同様に起動できます。

6. 「I）Install Software」を選択する

7. 「uCosminexus Service Coordinator Interact」にカーソルを移動し，スペースキーで選択する

   正しく選択すると，「uCosminexus Service Coordinator Interact」の左側に<@>が付きます。

   なお，Hitachi PP Installer の仕様により製品名の「uCosminexus Service Coordinator Interact」の一部が欠落して表示される場合があります。

8. 「I）Install」を選択してから，最下行に表示されるメッセージに対して y または Y を入力する

   インストールが始まります。

9. インストールが終了したら，「Q）Quit」を選択して終了する

重要

CSCIW をインストールすると，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もインストールされます。

4.3　ワーク管理データベースの構築

この節では，ワーク管理システムの運用に必要なワーク管理データベースの設定方法について説明します。

4.3.1　事前準備

ワーク管理データベースの構築を開始する前の準備について説明します。

(1)　ワーク管理データベース容量の算出

ワーク管理サーバで管理するビジネスプロセス定義数，案件数，ユーザ数などを基に，ワーク管理データベースの構造および容量を決定します。

ワーク管理データベースの容量を決定するための計算式については，「付録 B　テーブル容量の見積もり」を参照してください。

(2)　ワーク管理データベースへの接続情報の決定

CSCIW が使用するワーク管理データベースを構築する DBMS に接続するための情報を決定しておく必要があります。

必要な情報は次のとおりです。

●HiRDB の場合

- ワーク管理データベースに接続するためのユーザ名
- ワーク管理データベースに接続するためのパスワード
- ワーク管理データベースで使用するテーブルを格納するための RD エリア名
- ワーク管理データベースで使用するインデクスを格納するための RD エリア名
- ワーク管理データベースのポート番号
- ワーク管理データベースのホスト名または IP アドレス

●ORACLE の場合

- ワーク管理データベースに接続するためのユーザ名
- ワーク管理データベースに接続するためのパスワード
- ワーク管理データベースで使用するテーブルを格納するための表領域名
- ワーク管理データベースで使用するインデクスを格納するための表領域名
- ワーク管理データベースのサービス名

●SQL Server の場合

- ワーク管理データベースに接続するためのユーザ名

- ワーク管理データベースに接続するためのパスワード
- ワーク管理データベースで使用するテーブルを格納するためのファイルグループ
- ワーク管理データベースで使用するインデクスを格納するためのファイルグループ

これらの情報は，ワーク管理データベースの作成時に，SQL スクリプトを実行する上で必要となります。

4.3.2 ワーク管理データベースの環境設定（HiRDB の場合）

重要

HiRDB の SQL 最適化について

HiRDB の SQL 最適化オプション，SQL 拡張最適化オプションはデフォルトで指定される最適化方法を無効にしないでください。無効にした場合，性能が劣化するおそれがあります。

該当するクライアント環境定義

PDSQLOPTLVL：SQL 最適化オプション

PDADDITIONALOPTLVL：SQL 拡張最適化オプション

該当するシステム定義のオペランド

pd_optimize_level：SQL 最適化オプション

pd_additional_optimize_level：SQL 拡張最適化オプション

(1) ワーク管理データベースの作成（HiRDB の場合）

ワーク管理データベースのテーブルやインデクスを HiRDB に作成する方法について説明します。

テーブルやインデクスは，SQL スクリプトファイルを使用して作成してください。

(a) SQL スクリプトファイルの編集

テーブルやインデクスは，次に示す SQL スクリプトファイルを編集して作成します。

```
<CSCIWインストールディレクトリ>/sql/createtable_hirdb.sql
<CSCIWインストールディレクトリ>/sql/createtableex_hirdb.sql※
<CSCIWインストールディレクトリ>/sql/insertex_hirdb.sql※
```

注※

アプリケーション呼び出しサービスを使用する場合だけ編集してください。

SQL スクリプトファイル中の文字列を，テキストエディタを使用してワーク管理データベース用に書き換えてください。テーブルやインデクスを作成する際に SQL スクリプトファイルの書き換えが必要な文字列，および書き換える内容を次に示します。

表 4-4　テーブルやインデクスの作成時に書き換えが必要な SQL スクリプトファイル中の文字列（HiRDB の場合）

書き換えが必要な文字列	書き換える内容
<SYSTEMID>	システム ID ワーク管理データベースを一意に識別するためのシステム ID に置換してください。システム ID に指定できる文字は，半角英大文字で始まる半角英大文字および半角数字で，5 文字以内です。
<RDDATA>	テーブルを格納する RD エリア名 ワーク管理データベースのテーブルを格納する RD エリア名に置換してください。
<RDINDEX>	インデクスを格納する RD エリア名 ワーク管理データベースのインデクスを格納する RD エリア名に置換してください。

なお，テーブルごとに格納先の RD エリアを変えて指定するなど，格納する RD エリアをカスタマイズできます。RD エリアの指定方法については，マニュアル「HiRDB SQL リファレンス」を参照してください。

(b)　SQL スクリプトファイルの実行

編集した SQL スクリプトファイルを，HiRDB SQL Executer を使用して実行します。

pdsql コマンドに編集した SQL スクリプトファイルを指定し，実行することで，接続する認可識別子のスキーマにテーブルを作成できます。HiRDB SQL Executer の使用方法については，マニュアル「HiRDB 解説」を参照してください。

pdsql コマンドの指定例を次に示します。

```
pdsql -u <接続認可識別子>/<パスワード>
      -h <HiRDBサーバのホスト名またはIPアドレス>
      -n <HiRDBサーバのポート番号>
       < <編集したSQLスクリプトファイルのパス>
```

注

接続認可識別子には，CONNECT 権限のある認可識別子を指定してください。

(2)　データベースへのアクセス権限の付与（HiRDB の場合）

ワーク管理データベースのテーブルやインデクスを作成した認可識別子と，業務アプリケーションが接続するための認可識別子とが異なっている場合，業務アプリケーションが接続するための認可識別子に，ワーク管理データベースのテーブルやインデクスなどへアクセスするための権限を与える必要があります。

アクセス権限を付与する方法について説明します。

アクセス権限は，SQL スクリプトファイルを使用して付与してください。

(a) SQL スクリプトファイルの編集

アクセス権限は，次に示す SQL スクリプトファイルを編集して付与します。

```
<CSCIWインストールディレクトリ>/sql/granttable_hirdb.sql
<CSCIWインストールディレクトリ>/sql/granttableex_hirdb.sql※
```

注※

アプリケーション呼び出しサービスを使用する場合だけ編集してください。

SQL スクリプトファイル中の文字列を，テキストエディタを使用してワーク管理データベースへアクセスできるように書き換えてください。アクセス権限を付与する際に SQL スクリプトファイルの書き換えが必要な文字列，および書き換える内容を次に示します。

表 4-5 アクセス権限を付与する際に書き換えが必要な SQL スクリプトファイル中の文字列 (HiRDB の場合)

書き換えが必要な文字列	書き換える内容
<SCHEMANAME>	CSCIW の管理用の認可識別子 ワーク管理データベースのテーブルやインデクスを作成した認可識別子に置換してください。
<SYSTEMID>	システム ID ワーク管理データベースを作成した際に設定したシステム ID に置換してください。
<USERNAME>	業務アプリケーション用の認可識別子 業務アプリケーションが接続するための認可識別子に置換してください。

(b) SQL スクリプトファイルの実行

編集した SQL スクリプトファイルを，HiRDB SQL Executer を使用して実行します。

pdsql コマンドに編集した SQL スクリプトファイルを指定し，実行することで，アクセス権限を付与できます。HiRDB SQL Executer の使用方法については，マニュアル「HiRDB 解説」を参照してください。

4.3.3 ワーク管理データベースの環境設定（ORACLE の場合）

(1) ワーク管理データベースの作成（ORACLE の場合）

ワーク管理データベースのテーブルやインデクスを ORACLE に作成する方法について説明します。

テーブルやインデクスは，SQL スクリプトファイルを使用して作成してください。

(a) SQL スクリプトファイルの編集

テーブルやインデクスは，次に示す SQL スクリプトファイルを編集して作成します。

```
<CSCIWインストールディレクトリ>/sql/createtable_oracle.sql
<CSCIWインストールディレクトリ>/sql/createtableex_oracle.sql※
<CSCIWインストールディレクトリ>/sql/insertex_oracle.sql※
```

注※

アプリケーション呼び出しサービスを使用する場合だけ編集してください。

SQL スクリプトファイル中の文字列を，テキストエディタを使用してワーク管理データベース用に書き換えてください。テーブルやインデクスを作成する際に SQL スクリプトファイルの書き換えが必要な文字列，および書き換える内容を次に示します。

表 4-6　テーブルやインデクスの作成時に書き換えが必要な SQL スクリプトファイル中の文字列（ORACLE の場合）

書き換えが必要な文字列	書き換える内容
<SYSTEMID>	システム ID ワーク管理データベースを一意に識別するためのシステム ID に置換してください。システム ID に指定できる文字は，半角英大文字で始まる半角英大文字および半角数字で，5 文字以内です。
<DATASPACE>	テーブルを格納する表領域名 ワーク管理データベースのテーブルを格納する表領域名に置換してください。
<INDEXSPACE>	インデクスを格納する表領域名 ワーク管理データベースのインデクスを格納する表領域名に置換してください。

なお，テーブルごとに格納先の表領域を変えて指定するなど，格納する表領域をカスタマイズできます。表領域の指定方法については，ORACLE のマニュアルを参照してください。

(b)　SQL スクリプトファイルの実行

編集した SQL スクリプトファイルを，SQL*Plus を使用して実行します。

sqlplus コマンドに編集した SQL スクリプトファイルを指定し，実行することで，接続するユーザのスキーマにテーブルを作成できます。SQL*Plus の使用方法については，ORACLE のマニュアルを参照してください。

sqlplus コマンドの指定例を次に示します。

```
sqlplus  <接続ユーザ名>/<パスワード>@<Oracle Net 接続識別子>
         @<編集したSQLスクリプトファイルの絶対パス>
```

注

接続ユーザ名には，次に示す権限がすべて与えられているユーザを指定してください。

- CREATE SESSION システム権限
- CREATE TABLE システム権限

- CREATE VIEW システム権限

(2) データベースへのアクセス権限の付与（ORACLE の場合）

ワーク管理データベースのテーブルやインデクスを作成したユーザと，業務アプリケーションが接続するためのユーザとが異なっている場合，業務アプリケーションが接続するためのユーザに，ワーク管理データベースのテーブルやインデクスなどへアクセスするための権限を与える必要があります。

アクセス権限を付与する方法について説明します。

アクセス権限は，SQL スクリプトファイルを使用して付与してください。

(a) SQL スクリプトファイルの編集

アクセス権限は，次に示す SQL スクリプトファイルを編集して付与します。

```
<CSCIWインストールディレクトリ>/sql/granttable_oracle.sql
<CSCIWインストールディレクトリ>/sql/granttableex_oracle.sql※
```

注※

アプリケーション呼び出しサービスを使用する場合だけ編集してください。

SQL スクリプトファイル中の文字列を，テキストエディタを使用してワーク管理データベースへアクセスできるように書き換えてください。アクセス権限を付与する際に SQL スクリプトファイルの書き換えが必要な文字列，および書き換える内容を次に示します。

表 4-7 アクセス権限を付与する際に書き換えが必要な SQL スクリプトファイル中の文字列（ORACLE の場合）

書き換えが必要な文字列	書き換える内容
<SCHEMANAME>	CSCIW の管理用のユーザ ワーク管理データベースのテーブルやインデクスを作成したユーザに置換してください。
<SYSTEMID>	システム ID ワーク管理データベースを作成した際に設定したシステム ID に置換してください。
<USERNAME>	業務アプリケーション用のユーザ 業務アプリケーションが接続するためのユーザに置換してください。

(b) SQL スクリプトファイルの実行

編集した SQL スクリプトファイルを，SQL*Plus を使用して実行します。

sqlplus コマンドに編集した SQL スクリプトファイルを指定し，実行することで，アクセス権限を付与できます。SQL*Plus の使用方法については，ORACLE のマニュアルを参照してください。

4.3.4　ワーク管理データベースの環境設定（SQL Server の場合）

(1)　ワーク管理データベースの作成（SQL Server の場合）

ワーク管理データベースのテーブルやインデクスを SQL Server に作成する方法について説明します。

テーブルやインデクスは，SQL スクリプトファイルを使用して作成してください。

(a)　SQL スクリプトファイルの編集

テーブルやインデクスは，次に示す SQL スクリプトファイルを編集して作成します。

```
<CSCIWインストールディレクトリ>/sql/createtable_sqlserver.sql
```

SQL スクリプトファイル中の文字列を，テキストエディタを使用してワーク管理データベース用に書き換えてください。テーブルやインデクスを作成する際に SQL スクリプトファイルの書き換えが必要な文字列，および書き換える内容を次に示します。

表 4-8　テーブルやインデクスの作成時に書き換えが必要な SQL スクリプトファイル中の文字列（SQL Server の場合）

書き換えが必要な文字列	書き換える内容
<SYSTEMID>	システム ID ワーク管理データベースを一意に識別するためのシステム ID に置換してください。システム ID に指定できる文字は，半角英大文字で始まる半角英大文字および半角数字で，5 文字以内です。
<FILEGROUP>	テーブルを格納するファイルグループ ワーク管理データベースのテーブルを格納するファイルグループに置換してください。
<INDEXFILEGROUP>	インデクスを格納するファイルグループ ワーク管理データベースのインデクスを格納するファイルグループに置換してください。

なお，テーブルごとに格納先のファイルグループを変えて指定するなど，格納するファイルグループをカスタマイズできます。ファイルグループの指定方法については，SQL Server のマニュアルを参照してください。

重要

CSCIW では，文字コードの照合順序が決まっています。照合順序は，SQL スクリプトファイル中に「<コード部>_CS_AS_KS_WS」の形式で設定されています。「_CS_AS_KS_WS」の部分は変更しないでください。なお，コード部（例えば，「Japanese_CS_AS_KS_WS」の「Japanese」の部分）は，変更しても照合順序に影響しません。必要に応じて書き換えてください。

(b) SQL スクリプトファイルの実行

編集した SQL スクリプトファイルを，sqlcmd ユティリティを使用して実行します。

sqlcmd ユティリティに編集した SQL スクリプトファイルを指定し，実行することで，接続するユーザのスキーマにテーブルを作成できます。sqlcmd ユティリティの使用方法については，SQL Server のマニュアルを参照してください。

sqlcmd ユティリティの指定例を次に示します。

```
sqlcmd -U 〈接続ユーザ名〉
       -P 〈パスワード〉
       -i 〈編集したSQLスクリプトファイルの絶対パス〉
```

注

接続ユーザ名には，次に示す権限がすべて与えられているユーザを指定してください。

- CONNECT 権限
- CREATE TABLE 権限
- CREATE VIEW 権限
- CONTROL 権限

(2) データベースへのアクセス権限の付与（SQL Server の場合）

ワーク管理データベースのテーブルやインデクスを作成したユーザと，業務アプリケーションが接続するためのユーザとが異なっている場合，業務アプリケーションが接続するためのユーザに，ワーク管理データベースのテーブルやインデクスなどへアクセスするための権限を与える必要があります。

アクセス権限を付与する方法について説明します。

アクセス権限は，SQL スクリプトファイルを使用して付与してください。

(a) SQL スクリプトファイルの編集

アクセス権限は，次に示す SQL スクリプトファイルを編集して付与します。

```
<CSCIWインストールディレクトリ>/sql/granttable_sqlserver.sql
```

SQL スクリプトファイル中の文字列を，テキストエディタを使用してワーク管理データベースへアクセスできるように書き換えてください。アクセス権限を付与する際に SQL スクリプトファイルの書き換えが必要な文字列，および書き換える内容を次に示します。

表 4-9　アクセス権限を付与する際に書き換えが必要な SQL スクリプトファイル中の文字列（SQL Server の場合）

書き換えが必要な文字列	書き換える内容
<SCHEMANAME>	CSCIW の管理用のスキーマ ワーク管理データベースのテーブルやインデクスを作成したスキーマに置換してください。
<SYSTEMID>	システム ID ワーク管理データベースを作成した際に設定したシステム ID に置換してください。
<USERNAME>	業務アプリケーション用のユーザ 業務アプリケーションが接続するためのユーザに置換してください。

(b)　SQL スクリプトファイルの実行

編集した SQL スクリプトファイルを，sqlcmd ユティリティを使用して実行します。

sqlcmd ユティリティに編集した SQL スクリプトファイルを指定し，実行することで，アクセス権限を付与できます。sqlcmd ユティリティの使用方法については，SQL Server のマニュアルを参照してください。

4.4　開発環境での環境設定

ビジネスプロセス定義開発環境の設定は，前提プログラムをインストールしたあと，uCosminexus Business Process Developer をインストールし，ビジネスプロセス管理ツール（CSCIW-Definer）の動作環境を設定します。手順については，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。

4.5　実行環境での環境設定

この節では，ワーク管理システムの実行環境の設定方法について説明します。

4.5.1　事前準備

ワーク管理システムの実行環境の設定を開始する前の準備について説明します。

(1)　データベースコネクション数の見積もり

CSCIW で必要とするコネクション数を次の表に示します。この表をベースに，CSCIW を運用するマシン数などを考慮して，使用する DB コネクションを見積もってください。

表 4-10　CSCIW で必要とするコネクション数

分類	必要とするコネクション数
業務アプリケーション	CSCIW を使用する J2EE アプリケーションが必要とするコネクション数は，業務アプリケーションが同時に使用するコネクション数です。
	CSCIW を使用する Java アプリケーションが必要とするコネクション数は，業務アプリケーションが同時に使用するコネクション数＋ 1 です。
CSCIWManagementServer	1 つの CSCIWManagementServer が必要とするコネクション数は 1 です。
コマンド	1 つのコマンドが必要とするコネクション数は 1 です。
案件運用操作	1 つの案件運用操作が必要とするコネクション数は，案件運用操作を同時に実行する数＋案件一括削除を同時に実行する数です。 案件一括削除を同時に実行する数の最大値は，WorkManager の最大スレッド数（デフォルト値は 10）です。
アプリケーション呼び出しサービス※	1 つのアプリケーション呼び出しサービスが必要とするコネクション数は，WorkManager の最大スレッド数です。デフォルト値は 10 です。

注※

アプリケーション呼び出しサービスを使用する場合だけ見積もってください。

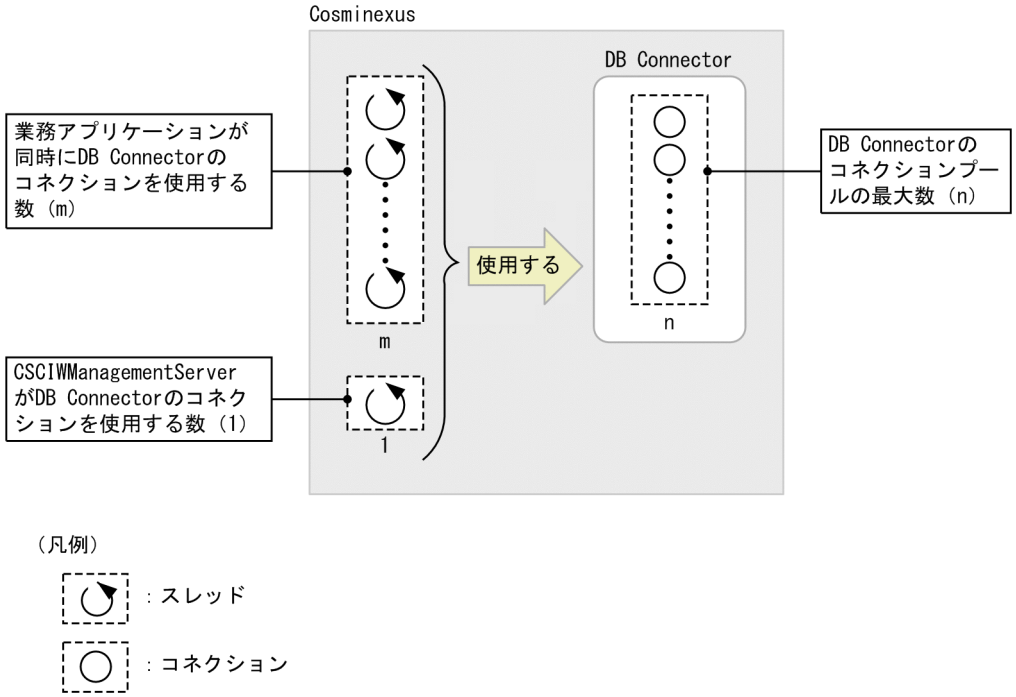
(2)　データベースコネクションプール数の見積もり

CSCIWManagementServer が使用する DB Connector と業務アプリケーションが使用する DB Connector を，同一にするかどうかで，データベースコネクションプール数の見積もりが異なります。DB Connector のデータベースコネクションプール数の考え方を次に示します。

(a) CSCIWManagementServer と業務アプリケーションの DB Connector を同一にする場合

次に示す図のように，CSCIWManagementServer と業務アプリケーションの DB Connector を同一にする場合，「業務アプリケーションが同時に使用するコネクション数＋ 1 ≦DB Connector のコネクションプールの最大数」になるようにしてください。

図 4-2　DB Connector を同一にする場合のコネクション数とコネクションプール数の関係

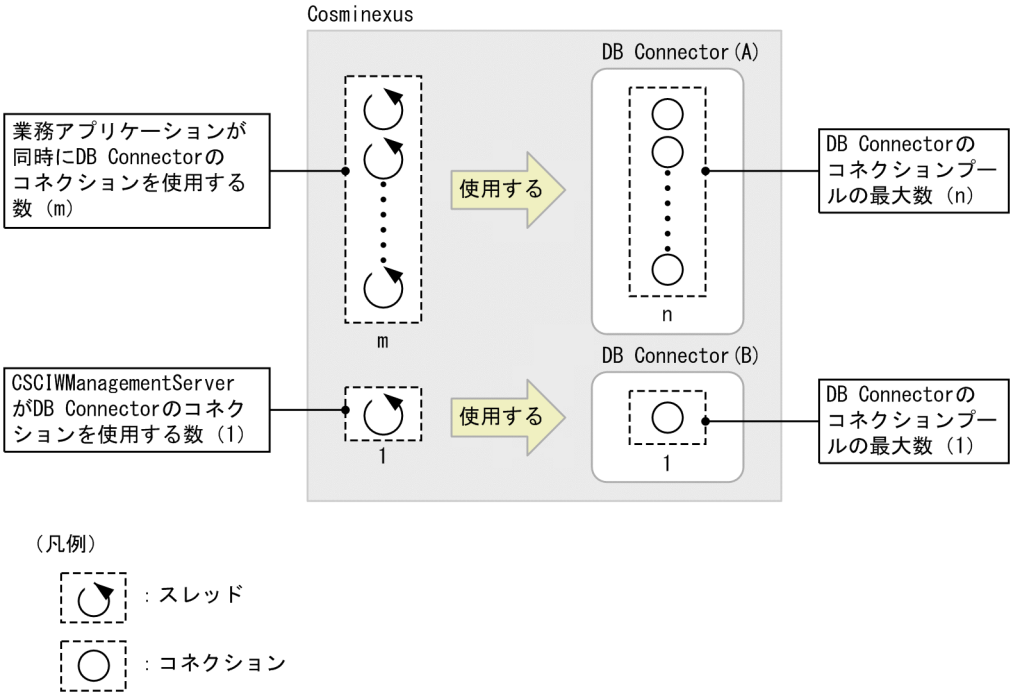


上の図の場合，「m」と「n」は「m+1≦n」という条件を満たす必要があります。

(b) CSCIWManagementServer と業務アプリケーションの DB Connector を別にする場合

次に示す図のように，CSCIWManagementServer と業務アプリケーションの DB Connector を別にする場合，業務アプリケーションが同時に使用するコネクション数と，業務アプリケーションが使用する DB Connector のコネクションプールの最大数は，どちらも任意です。

図 4-3　DB Connector を別にする場合のコネクション数とコネクションプール数の関係



上の図の場合，「m」および「n」は任意ということになります。

4.5.2　実行環境の構築

ワーク管理システムの実行環境を構築する方法について説明します。

(1)　環境変数の設定

(a)　COSMINEXUS_HOME 環境変数の確認

Windows の場合，COSMINEXUS_HOME 環境変数に Cosminexus のインストールフォルダが正しく設定されていることを確認してください。

UNIX の場合，確認は不要です。

(b)　CSCIW_HOME 環境変数の設定

CSCIW_HOME 環境変数は，次のように設定してください。

- Windows の場合

```
CSCIW_HOME=<CSCIWインストールフォルダ>
```

重要

CSCIW のインストールフォルダ以外は設定しないでください。次のように，CSCIW_HOME 環境変数に，誤った設定（ほかの環境変数の参照先フォルダを設定するなど）をした場合，動作は保証できません。

＜誤った設定例＞

```
CSCIW_HOME=%PROGRAMFILES%¥HITACHI¥CSCIW
```

- UNIX の場合

```
CSCIW_HOME=/opt/hitachi/CSCIW
```

(2)　セットアッププロパティファイルの編集

CSCIW のセットアッププロパティファイルを環境に合わせて編集してください。なお，セットアッププロパティファイルの設定内容については，「付録 A.2　セットアッププロパティファイル」を参照してください。

セットアッププロパティファイルの格納パスは，次のとおりです。

- Windows の場合

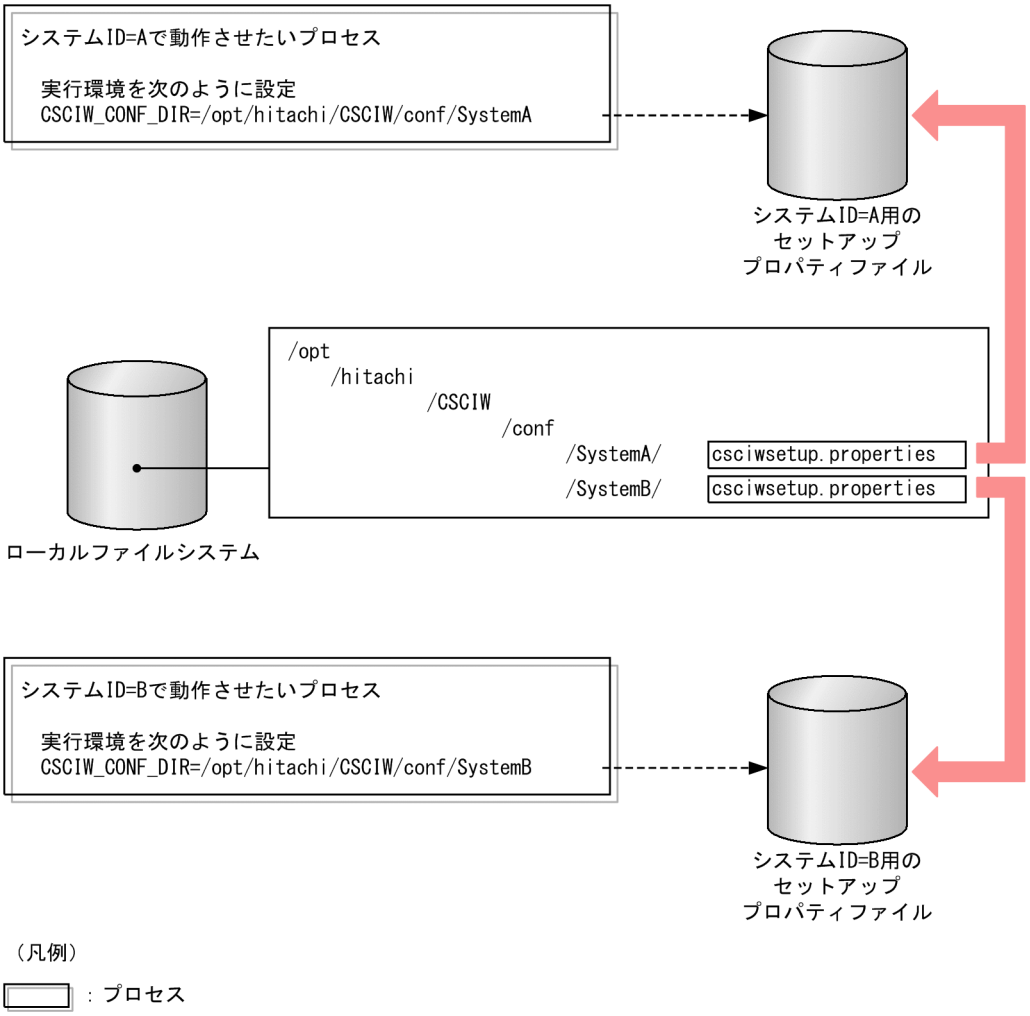
```
%CSCIW_HOME%¥conf¥csciwsetup.properties
```

- UNIX の場合

```
${CSCIW_HOME}/conf/csciwsetup.properties
```

CSCIW をマルチインスタンス構成にする場合は，次のようにセットアッププロパティファイルを設定してください。

図 4-4　CSCIW をマルチインスタンス構成にする場合のセットアッププロパティファイルの設定



1. システム ID ごとにセットアッププロパティファイルを作成する
2. プロセスごとに，動作させたいシステム ID のセットアッププロパティファイルの格納パスを CSCIW_CONF_DIR 環境変数に設定する

アプリケーション呼び出しサービスを使用する場合は，CSCIW のセットアッププロパティファイルに次の設定をしてください。

```
UseApplicationCallService=true
```

(3)　コマンド用環境設定ファイルの編集

CSCIW のコマンド用環境設定ファイルを環境に合わせて編集してください。なお，コマンド用環境設定ファイルの設定内容については，「付録 A.3　コマンド用環境設定ファイル」を参照してください。

コマンド用環境設定ファイルの格納パスは，次のとおりです。

- Windows の場合

```
%CSCIW_HOME%¥conf¥csciwcmdconf.bat
```

- UNIX の場合

```
${CSCIW_HOME}/conf/csciwcmdconf
```

CSCIW をマルチインスタンス構成にする場合は，次のようにコマンド用環境設定ファイルを設定してください。

1. システム ID ごとにコマンド用環境設定ファイルを作成する
2. プロセスごとに，動作させたいシステム ID のコマンド用環境設定ファイルの格納パスを CSCIW_CONF_DIR 環境変数に設定する

(4) コマンドによる実行環境の初期化

CSCIW の実行環境は，ciwsetenv（環境の構築および削除）コマンドを実行して構築します。次に示す形式でコマンドを実行すると，CSCIW の実行環境に必要な設定項目にデフォルト値を設定し，システム ID の設定情報を登録します。

```
ciwsetenv -sid 〈システムID〉 -f 〈環境構築ファイル名〉
```

ciwsetenv（環境の構築および削除）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。

4.5.3 アプリケーションサーバの設定

アプリケーションサーバ（Cosminexus）上で CSCIW を使用するための設定方法を説明します。

次の 2 とおりの手順が記載されている場合は，どちらかの手順を実施してください。

- 運用管理ポータルを使用して設定する場合
- 定義ファイルおよびサーバ管理コマンドを使用して設定する場合

メモ

CSCIW 02-30 以前をお使いの場合

この項で説明する設定方法は，CSCIW 02-30 以前の場合とは異なります。対応するバージョンのマニュアルをご確認ください。

ヒント

Cosminexus J2EE サーバを V9 互換モードで動作させる場合

アプリケーションサーバの設定を開始する前に V9 互換モード用のファイルへの入れ替えが必要です。「付録 J.1　互換モード用のファイルへの入れ替え」を参照して，ファイルを入れ替えてください。

(1)　コンテナ拡張ライブラリへの取り込み

コンテナ拡張ライブラリへの取り込みは，Cosminexus 上で CSCIW を使用するために必要な設定です。Java オブジェクト呼び出しを利用する場合は，次の設定に加えて「4.5.5　Java オブジェクト呼び出しを利用する場合の設定」を参照して，設定してください。

(a)　J2EE サーバ用オプション定義ファイルへの設定

ヒント

Cosminexus J2EE サーバを V9 互換モードで動作させる場合

コンテナ拡張ライブラリに取り込むライブラリが異なります。「付録 J.2　コンテナ拡張ライブラリに取り込むライブラリの変更」を参照して，コンテナ拡張ライブラリを設定してください。

Cosminexus の J2EE サーバ用オプションのadd.class.path に設定する，コンテナ拡張ライブラリ用のファイルは，次のとおりです。

- CSCIW ライブラリ（csciw.jar およびcsciwbpmn.jar）
- JAX-RS ライブラリ（jaxrs-impl.jar およびjaxrs-jackson.jar）
- JDBC ドライバのライブラリ

  JDBC ドライバは，HiRDB の場合は「HiRDB Type4 JDBC Driver」，ORACLE の場合は「Oracle JDBC Thin Driver」，SQL Server の場合は「SQL Server の JDBC ドライバ」を追加してください。
- Windows の場合

```
add.class.path=<CSCIWインストールフォルダ>¥lib¥csciw.jar
add.class.path=<cosminexus.home>¥CC¥javaee¥1100¥lib¥jaxrs-impl.jar
add.class.path=<cosminexus.home>¥CC¥javaee¥1100¥lib¥jaxrs-jackson.jar
add.class.path=<JDBCドライバのライブラリのパス>
```

- UNIX の場合

```
add.class.path=/opt/hitachi/CSCIW/lib/csciw.jar
add.class.path=/opt/Cosminexus/CC/javaee/1100/lib/jaxrs-impl.jar
add.class.path=/opt/Cosminexus/CC/javaee/1100/lib/jaxrs-jackson.jar
add.class.path=<JDBCドライバのライブラリのパス>
```

アプリケーション呼び出しサービスを使用する場合は，add.class.path にcsciwbpmn.jar ファイルを追加してください。

- Windows の場合

```
add.class.path=<CSCIWインストールフォルダ>¥lib¥csciwbpmn.jar
```

- UNIX の場合

```
add.class.path=/opt/hitachi/CSCIW/lib/csciwbpmn.jar
```

csciw.jar およびcsciwbpmn.jar ファイルはコンテナ拡張ライブラリへ取り込むと，アプリケーションサーバのプロセス内で有効になります。このため，各アプリケーションのear にcsciw.jar およびcsciwbpmn.jar ファイルを組み込まないでください。

次のどちらかの手順を実行して，J2EE サーバ用オプション定義を設定してください。

■ 運用管理ポータルを使用して設定する場合

次の手順で操作してください。

1. 運用管理ポータルにログインする
2. 運用管理ポータルで［論理サーバの環境設定］アンカーをクリックする
3. ツリーペインで次の順にクリックする
   ［<操作対象の運用管理ドメイン名>］－［論理 J2EE サーバ］－［J2EE サーバ］－［<操作対象の J2EE サーバ名>］
4. 右ペインの［コンテナ］タブ，［J2EE］タブをクリックし，［J2EE コンテナの設定］画面を表示する
5. ［拡張パラメタ］の欄に，上で示したコンテナ拡張ライブラリを設定する

詳細は，マニュアル「Cosminexus アプリケーションサーバ 運用管理ポータル操作ガイド」の「J2EE コンテナの設定」を参照してください。

■ 定義ファイルおよびサーバ管理コマンドを使用して設定する場合

usrconf.cfg（J2EE サーバ用オプション定義ファイル）を編集してください。

詳細は，マニュアル「Cosminexus 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）」の「usrconf.cfg（J2EE サーバ用オプション定義ファイル）」を参照してください。

(b) サーバ管理コマンド用オプション定義ファイルへの設定

■ アプリケーション呼び出しサービスを使用しない場合

サーバ管理コマンド用オプション定義ファイルに，コンテナ拡張ライブラリ用のcsciw.jar ファイルを設定してください。

- Windows の場合

  usrconf.bat（サーバ管理コマンド用オプション定義ファイル）のUSRCONF_JVM_CLASSPATH キーに，csciw.jar ファイルを設定します。

```
set USRCONF_JVM_CLASSPATH=<CSCIWインストールフォルダ>¥lib¥csciw.jar
```

usrconf.bat（サーバ管理コマンド用オプション定義ファイル）については，マニュアル「Cosminexus 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）」の「usrconf.bat（サーバ管理コマンド用オプション定義ファイル）」の説明を参照してください。

- UNIX の場合

usrconf（サーバ管理コマンド用オプション定義ファイル）のUSRCONF_JVM_CLPATH キーに，csciw.jar ファイルを設定します。

```
set USRCONF_JVM_CLPATH=/opt/hitachi/CSCIW/lib/csciw.jar
```

usrconf（サーバ管理コマンド用オプション定義ファイル）については，マニュアル「Cosminexus 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）」の「usrconf（サーバ管理コマンド用オプション定義ファイル）」の説明を参照してください。

■ アプリケーション呼び出しサービスを使用する場合

サーバ管理コマンド用オプション定義ファイルに，コンテナ拡張ライブラリ用のcsciw.jar および csciwbpmn.jar ファイルを設定してください。

- Windows の場合

usrconf.bat（サーバ管理コマンド用オプション定義ファイル）のUSRCONF_JVM_CLASSPATH キーに，csciw.jar およびcsciwbpmn.jar ファイルを設定します。

```
set USRCONF_JVM_CLASSPATH=<CSCIWインストールフォルダ>¥lib¥csciw.jar;<CSCIWインストールフォルダ>¥lib¥csciwbpmn.jar
```

usrconf.bat（サーバ管理コマンド用オプション定義ファイル）については，マニュアル「Cosminexus 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）」の「usrconf.bat（サーバ管理コマンド用オプション定義ファイル）」の説明を参照してください。

- UNIX の場合

usrconf（サーバ管理コマンド用オプション定義ファイル）のUSRCONF_JVM_CLPATH キーに，csciw.jar およびcsciwbpmn.jar ファイルを設定します。

```
set USRCONF_JVM_CLPATH=/opt/hitachi/CSCIW/lib/csciw.jar:/opt/hitachi/CSCIW/lib/csciwbpmn.jar
```

usrconf（サーバ管理コマンド用オプション定義ファイル）については，マニュアル「Cosminexus 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）」の「usrconf（サーバ管理コマンド用オプション定義ファイル）」の説明を参照してください。

重要

アプリケーション呼び出しサービスを使用する場合，次に示す J2EE サーバ単位の REST（JAX-RS）の通信タイムアウト値のデフォルトは0（タイムアウトなし）のため，REST アプリケー

ションが無応答になった場合などに，アプリケーション呼び出しサービスは，応答を待ち続けます。

- ejbserver.javaee.jaxrs.config.client.connectTimeout（クライアントソケットの接続タイムアウト）
- ejbserver.javaee.jaxrs.config.client.readTimeout（クライアントソケットの読み込みタイムアウト）

そのため，どちらの通信タイムアウト値も 0 以外の値を設定してください。また，クライアントソケットの読み込みタイムアウトは REST アプリケーションの処理時間より大きな値を設定してください。

J2EE サーバ単位の JAX-RS の通信タイムアウトの設定については，マニュアル「Cosminexus 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)」の「J2EE サーバ単位の通信タイムアウトの設定」を参照してください。

次の場合は，アプリケーション呼び出し情報ファイルを使用して通信タイムアウト値を設定します。

- ref 識別子ごとに通信タイムアウト値を設定したい場合
- Cosminexus に同梱されている JAX-RS ライブラリをコンテナ拡張ライブラリに設定しない場合
  この場合，J2EE サーバ単位の REST（JAX-RS）の通信タイムアウト値は無効になります。

詳細については，「付録 H.12(9)　アプリケーション呼び出し情報ファイルに指定する内容」の「(f)　rest.request.read.timeout」と「(g)　rest.request.connect.timeout」を参照してください。

(2)　環境変数の取り込み

設定する環境変数は，次のとおりです。

- Windows の場合

```
CSCIW_HOME=<CSCIWインストールフォルダ>
```

- UNIX の場合

```
CSCIW_HOME=/opt/hitachi/CSCIW
```

なお，CSCIW_CONF_DIR 環境変数でシステム ID を切り替えたい場合は，CSCIW_CONF_DIR 環境変数も設定してください。

(a)または(b)の手順を実行して，環境変数を設定してください。

(a) 運用管理ポータルを使用して設定する場合

次の手順で操作してください。

1. 運用管理ポータルにログインする
2. 運用管理ポータルで［論理サーバの環境設定］アンカーをクリックする
3. ツリーペインで次の順にクリックする
   ［<操作対象の運用管理ドメイン名>］－［論理 J2EE サーバ］－［J2EE サーバ］－［<操作対象の J2EE サーバ名>］
4. 右ペインの［環境変数］タブをクリックし，［環境変数の設定］画面を表示する
5. ［環境変数］の欄に，環境変数を設定する

詳細は，マニュアル「Cosminexus アプリケーションサーバ 運用管理ポータル操作ガイド」の「J2EE サーバの環境変数の設定」を参照してください。

(b) 定義ファイルおよびサーバ管理コマンドを使用して設定する場合

cjstartsv コマンドを実行するシェルに環境変数を設定します。

cjstartsv コマンドの詳細は，マニュアル「Cosminexus アプリケーションサーバ リファレンス コマンド編」を参照してください。

(3) Cosminexus の起動

(a)または(b)の手順を実行して，Cosminexus を起動してください。

(a) 運用管理ポータルを使用して設定する場合

運用管理ポータルで［論理サーバの起動／停止］アンカーをクリックし，対象の J2EE サーバを起動します。

詳細は，マニュアル「Cosminexus アプリケーションサーバ 運用管理ポータル操作ガイド」の「論理サーバの起動／停止の設定」を参照してください。

(b) 定義ファイルおよびサーバ管理コマンドを使用して設定する場合

cjstartsv コマンドで J2EE サーバを起動します。

cjstartsv コマンドの詳細は，マニュアル「Cosminexus アプリケーションサーバ リファレンス コマンド編」の「J2EE サーバを操作するコマンド」を参照してください。

cjstartsv コマンドの実行例を次に示します。

```
cjstartsv 〈サーバ名称〉
```

(4) DB Connector の設定

DB Connector のリソースアダプタの表示名は「DB_Connector_for_CSCIW」（CSCIW のデフォルト）を設定してください。なお，リソースアダプタの表示名は任意の表示名も設定できます。その場合，データソース表示名をリソースアダプタと同じ表示名に変更する必要があります。

なお，Connector 属性のトランザクションサポートレベルでサポートしているのは，LocalTransaction だけです。

(a)または(b)の手順を実行して，データベースと接続するための DB Connector を設定してください。

(a) 運用管理ポータルを使用して設定する場合

運用管理ポータルを使用して DB Connector を設定します。

詳細は，マニュアル「Cosminexus アプリケーションサーバ 運用管理ポータル操作ガイド」の「リソースアダプタの管理」を参照してください。

■ DB Connector をインポートする

次の手順で操作してください。

1. 運用管理ポータルにログインする
2. 運用管理ポータルで［論理サーバのアプリケーション管理］アンカーをクリックする
3. ツリーペインで次の順にクリックする
   ［<操作対象の運用管理ドメイン名>］－［論理 J2EE サーバ］－［J2EE サーバ］－［<操作対象の J2EE サーバ名>］－［リソース］－［リソースアダプタ］
4. 右ペインの［インポート］タブをクリックし，［リソースアダプタのインポート］画面を表示する
5. インポートするリソースアダプタの指定方法として，［DB Connector］を選択する
6. メニューに表示される DB Connector（RAR ファイル）から，接続するデータベースに対応したリソースアダプタを指定する
7. ［リソースアダプタ名称］に「DB_Connector_for_CSCIW」を入力する
8. ［実行］ボタンをクリックする

■ プロパティを設定する

1. 右ペインの［プロパティ設定］タブをクリックし，［リソースアダプタのプロパティ設定］画面を表示する
2. ［リソース名］が「DB_Connector_for_CSCIW」になっていることを確認する
3. プロパティを設定するリソースアダプタの［基本設定］アンカーをクリックする

4. [リソースアダプタの基本設定] 画面で，プロパティを設定する

5. [設定] ボタンをクリックする

■ 接続を確認する

1. 右ペインの［開始／停止］タブをクリックし，［リソースアダプタの開始／停止］画面を表示する

2. 接続テストを実行するリソースアダプタの［接続テスト］アンカーをクリックする

3. 内容を確認して，［はい］ボタンをクリックする

■ DB Connector を開始する

1. 右ペインの［開始／停止］タブをクリックし，［リソースアダプタの開始／停止］画面を表示する

2. 開始するリソースアダプタの［開始］アンカーをクリックする

3. 内容を確認して，［はい］ボタンをクリックする

(b)　定義ファイルおよびサーバ管理コマンドを使用して設定する場合

サーバ管理コマンドを使用して DB Connector を設定します。

■ DB Connector をインポートする

cjimportres コマンドを使用して，DB Connector をインポートします。

インポート方法については，マニュアル「Cosminexus 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」の「DB Connector のインポート」または「リソースアダプタのインポート」の説明を参照してください。

cjimportres コマンドの実行例を次に示します。

・HiRDB の場合

```
cjimportres <サーバ名称> -type rar -f <Cosminexusインストールディレクトリ>/CC/DBConnector
/DBConnector_HiRDB_Type4_CP.rar
```

・Oracle の場合

```
cjimportres <サーバ名称> -type rar -f <Cosminexusインストールディレクトリ>/CC/DBConnector
/DBConnector_Oracle_CP.rar
```

■ プロパティを設定する

DB Connector がワーク管理データベースおよび業務データベースに接続できるように，cjgetresprop コマンドおよびcjsetresprop コマンドを使用して，DB Connector のプロパティを設定します。

cjgetresprop コマンドの実行例を次に示します。

・HiRDB の場合

```
cjgetresprop <サーバ名称> -type rar -resname DB_Connector_for_HiRDB_Type4 -c <属性ファイルパス>
```

・Oracle の場合

```
cjgetresprop <サーバ名称> -type rar -resname DB_Connector_for_Oracle -c <属性ファイルパス>
```

cjsetresprop コマンドの実行例を次に示します。

・HiRDB の場合

```
cjsetresprop <サーバ名称> -type rar -resname DB_Connector_for_HiRDB_Type4 -c <属性ファイルパス>
```

・Oracle の場合

```
cjsetresprop <サーバ名称> -type rar -resname DB_Connector_for_Oracle -c <属性ファイルパス>
```

cjgetresprop コマンドで取得した Connector 属性ファイルの，DB Connector のリソースアダプタの表示名に「DB_Connector_for_CSCIW」（CSCIW のデフォルト）を設定してください。cjsetresprop コマンドで属性を設定します。

```
  <display-name xml:lang="en">DB_Connector_for_CSCIW</display-name>
```

項目の詳細については，マニュアル「Cosminexus 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）」の「Connector 属性ファイル」の説明を参照してください。

■ DB Connector をデプロイする

デプロイとは，DB Connector を J2EE サーバに共有スタンドアロンモジュール（J2EE リソースアダプタ）として配備することです。

cjdeployrar コマンドを使用して，DB Connector をデプロイします。

設定方法については，マニュアル「Cosminexus 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」の「DB Connector のデプロイ」または「リソースアダプタのデプロイ」の説明を参照してください。

cjdeployrar コマンドの実行例を次に示します。

```
cjdeployrar <サーバ名称> -resname DB_Connector_for_CSCIW
```

■ 接続を確認する

cjtestres コマンドを使用して，接続テストを実施します。

接続テストの実施方法については，マニュアル「Cosminexus 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」の「DB Connector の接続テスト」または「J2EE リソースアダプタの接続テスト」の説明を参照してください。

cjtestres コマンドの実行例を次に示します。

```
cjtestres <サーバ名称> -type rar -resname DB_Connector_for_CSCIW
```

■ DB Connector を開始する

cjstartrar コマンドを使用して，DB Connector を開始します。

開始方法については，マニュアル「Cosminexus 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」の「DB Connector の開始」または「J2EE リソースアダプタの開始」の説明を参照してください。

cjstartrar コマンドの実行例を次に示します。

```
cjstartrar <サーバ名称> -resname DB_Connector_for_CSCIW
```

(5) CSCIWManagementServer の設定

CSCIWManagementServer を J2EE アプリケーションとして設定します。

CSCIWManagementServer の設定方法を次に示します。

1. セキュリティロールを設定する
   - セキュリティロールの追加
   - ユーザの追加
   - セキュリティロールとユーザの関連づけ
2. CSCIWManagementServer をインポートする
3. CSCIWManagementServer のプロパティを設定する

   DB Connector のリソースアダプタの表示名に「DB_Connector_for_CSCIW」以外を設定した場合だけ，データソース表示名を変更します。
4. CSCIWManagementServer を開始する
5. コンテキストルートを設定する

詳細な設定方法を次に示します。

(a) セキュリティロールの設定

CSCIWManagementServer にセキュリティロールを設定する方法を示します。

●セキュリティロールの追加

CSCIWManagementServer には，CSCIW Definer の認証をするために，Cosminexus のセキュリティロールとして，csciwdef ロールが必要です。次に示すコマンドを実行すると，セキュリティロールを追加できます。

```
cjaddsec <サーバ名称> -type role -name csciwdef
```

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

●ユーザの追加

セキュリティロールと関連づけるためのユーザを作成してください。ユーザ名およびパスワードには，「:」（半角コロン）は使用できません。

次に示すコマンドを実行すると，ユーザを追加できます。なお，すでに登録しているユーザを関連づける場合は，ユーザの追加は不要です。

```
cjaddsec <サーバ名称> -type user -name <ユーザ名> -password <パスワード>
```

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

●セキュリティロールとユーザの関連づけ

追加したセキュリティロールとユーザの関連づけが必要です。次に示すコマンドを実行すると，セキュリティロールとユーザの関連づけができます。

```
cjmapsec <サーバ名称> -role csciwdef -user <ユーザ名>
```

注

ユーザ名には，関連づけるユーザ名を 32 バイト以内の文字列で指定してください。

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

(b) CSCIWManagementServer のインポート

CSCIWManagementServer のファイルの格納パスは，次のとおりです。

- Windows の場合

  <CSCIW インストールフォルダ>¥lib¥csciw.ear

- UNIX の場合

  /opt/hitachi/CSCIW/lib/csciw.ear

次のどちらかの手順を実行して，CSCIWManagementServer をインポートしてください。

■ 運用管理ポータルを使用して設定する場合

次の手順で操作してください。

1. 運用管理ポータルにログインする

2. 運用管理ポータルで［論理サーバのアプリケーション管理］アンカーをクリックする

3. ツリーペインで次の順にクリックする

［<操作対象の運用管理ドメイン名>］－［論理J2EEサーバ］－［J2EEサーバ］－［<操作対象のJ2EEサーバ名>］－［アプリケーション］

4. 右ペインの［インポート］タブをクリックし，［J2EEアプリケーションのインポート］画面を表示する

5. ［インポートディレクトリ］にCSCIWManagementServerのファイルが格納されているパスを指定して［適用］ボタンをクリックする

6. ［J2EEアプリケーションファイル］で「csciw.ear」を選択して［実行］ボタンをクリックする

詳細は，マニュアル「Cosminexusアプリケーションサーバ 運用管理ポータル操作ガイド」の「J2EEアプリケーションの設定」を参照してください。

■ 定義ファイルおよびサーバ管理コマンドを使用して設定する場合

cjimportappコマンドを使用して，インポートします。

インポート方法については，マニュアル「Cosminexus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」の「J2EEアプリケーションのインポート」の説明を参照してください。

cjimportappコマンドの実行例を次に示します。

```
cjimportapp <サーバ名称> -f <csciw.earファイルのパス>
```

(c) CSCIWManagementServerのプロパティ設定

データベースと接続するためのDB Connectorのリソースアダプタの表示名に「DB_Connector_for_CSCIW」を設定した場合，この手順は不要です。

「DB_Connector_for_CSCIW」以外を設定した場合は，データソース表示名をリソースアダプタと同じ表示名に変更してください。変更方法については「6.2.4 CSCIWManagementServerが使用するデータソース表示名の変更」を参照してください。

(d) CSCIWManagementServerの開始

次のどちらかの手順を実行して，CSCIWManagementServerを開始してください。

■ 運用管理ポータルを使用して設定する場合

次の手順で操作してください。

1. 運用管理ポータルにログインする

2. 運用管理ポータルで［論理サーバのアプリケーション管理］アンカーをクリックする

3. ツリーペインで次の順にクリックする

［<操作対象の運用管理ドメイン名>］－［論理 J2EE サーバ］－［J2EE サーバ］－［<操作対象の J2EE サーバ名>］－［アプリケーション］

4. 右ペインの［開始／停止］タブをクリックし，［J2EE アプリケーションの開始／停止］画面を表示する

5. アプリケーション名「CSCIWManagementServer」の［開始］アンカーをクリックする

6. 内容を確認して，［はい］ボタンをクリックする

詳細は，マニュアル「Cosminexus アプリケーションサーバ 運用管理ポータル操作ガイド」の「J2EE アプリケーションの設定」を参照してください。

■ 定義ファイルおよびサーバ管理コマンドを使用して設定する場合

cjstartapp コマンドを使用して，CSCIWManagementServer を開始します。

開始方法については，マニュアル「Cosminexus 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」の「J2EE アプリケーションの開始」の説明を参照してください。

cjstartapp コマンドの実行例を次に示します。

```
cjstartapp <サーバ名称> -name CSCIWManagementServer
```

(e) コンテキストルートの設定

CSCIWManagementServer のコンテキストルートのデフォルトは「csciw」です。

J2EE サーバのフロントエンドにリバースプロキシを配置する場合，コンテキストルート「csciw」のリクエストを転送する設定をしてください。

(6) 案件運用操作の設定

案件運用操作はすべての Cosminexus に対して組み込む必要はありません。案件運用操作を実行する Cosminexus に組み込んでください。Java オブジェクト呼び出しを利用する場合は，次の設定に加えて「4.5.5 Java オブジェクト呼び出しを利用する場合の設定」を参照して，設定してください。

案件運用操作の設定方法を次に示します。

1. セキュリティロールを設定する
   - セキュリティロールの追加
   - ユーザの追加
   - セキュリティロールとユーザの関連づけ
2. 案件運用操作をインポートする
3. 案件運用操作のプロパティを設定する

DB Connector のリソースアダプタの表示名に「DB_Connector_for_CSCIW」以外を設定した場合だけ，データソース表示名を変更します。

4. 案件運用操作を開始する

5. コンテキストルートを設定する

詳細な設定方法を次に示します。

(a) セキュリティロールの設定

案件運用操作にセキュリティロールを設定する方法を示します。

●セキュリティロールの追加

案件運用操作には Cosminexus のセキュリティロールとして，csciwadmin ロールが必要です。次に示すコマンドを実行すると，セキュリティロールを追加できます。

```
cjaddsec <サーバ名称> -type role -name csciwadmin
```

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

●ユーザの追加

セキュリティロールと関連づけるためのユーザを作成してください。次に示すコマンドを実行すると，ユーザを追加できます。なお，すでに登録しているユーザを関連づける場合は，ユーザの追加は不要です。

```
cjaddsec <サーバ名称> -type user -name <ユーザ名※> -password <パスワード※>
```

注※

ユーザ名およびパスワードには，マルチバイト文字は使用できません。

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

●セキュリティロールとユーザの関連づけ

追加したセキュリティロールとユーザの関連づけが必要です。次に示すコマンドを実行すると，セキュリティロールとユーザの関連づけができます。

```
cjmapsec <サーバ名称> -role csciwadmin -user <ユーザ名※>
```

注※

ユーザ名には，関連づけるユーザ名を 32 バイト以内の文字列で指定してください。なお，ユーザ名には，マルチバイト文字は使用できません。

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

(b) 案件運用操作のインポート

案件運用操作のファイルの格納パスは，次のとおりです。

- Windows の場合

```
<CSCIWインストールフォルダ>¥lib¥csciwadmin.ear
```

- UNIX の場合

```
/opt/hitachi/CSCIW/lib/csciwadmin.ear
```

次のどちらかの手順を実行して，案件運用操作をインポートしてください。

■ 運用管理ポータルを使用して設定する場合

次の手順で操作してください。

1. 運用管理ポータルにログインする
2. 運用管理ポータルで［論理サーバのアプリケーション管理］アンカーをクリックする
3. ツリーペインで次の順にクリックする
   ［<操作対象の運用管理ドメイン名>］－［論理 J2EE サーバ］－［J2EE サーバ］－［<操作対象の J2EE サーバ名>］－［アプリケーション］
4. 右ペインの［インポート］タブをクリックし，［J2EE アプリケーションのインポート］画面を表示する
5. ［インポートディレクトリ］に案件運用操作のファイルが格納されているパスを指定して［適用］ボタンをクリックする
6. ［J2EE アプリケーションファイル］で「csciwadmin.ear」を選択して［実行］ボタンをクリックする

詳細は，マニュアル「Cosminexus アプリケーションサーバ 運用管理ポータル操作ガイド」の「J2EE アプリケーションの設定」を参照してください。

■ 定義ファイルおよびサーバ管理コマンドを使用して設定する場合

cjimportapp コマンドを使用して，インポートします。

インポート方法については，マニュアル「Cosminexus 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」の「J2EE アプリケーションのインポート」の説明を参照してください。

cjimportapp コマンドの実行例を次に示します。

```
cjimportapp <サーバ名称> -f <csciwadmin.earファイルのパス>
```

(c)　案件運用操作のプロパティ設定

データベースと接続するための DB Connector のリソースアダプタの表示名に「DB_Connector_for_CSCIW」を設定した場合，この手順は不要です。

「DB_Connector_for_CSCIW」以外を設定した場合は，データソース表示名をリソースアダプタと同じ表示名に変更してください。変更方法については「6.2.5　案件運用操作が使用するデータソース表示名の変更」を参照してください。

(d)　案件運用操作の開始

次のどちらかの手順を実行して，案件運用操作を開始してください。

■ 運用管理ポータルを使用して設定する場合

次の手順で操作してください。

1. 運用管理ポータルにログインする
2. 運用管理ポータルで［論理サーバのアプリケーション管理］アンカーをクリックする
3. ツリーペインで次の順にクリックする
   ［<操作対象の運用管理ドメイン名>］－［論理 J2EE サーバ］－［J2EE サーバ］－［<操作対象の J2EE サーバ名>］－［アプリケーション］
4. 右ペインの［開始／停止］タブをクリックし，［J2EE アプリケーションの開始／停止］画面を表示する
5. アプリケーション名「CSCIWAdminServlet」の［開始］アンカーをクリックする
6. 内容を確認して，［はい］ボタンをクリックする

詳細は，マニュアル「Cosminexus アプリケーションサーバ 運用管理ポータル操作ガイド」の「J2EE アプリケーションの設定」を参照してください。

■ 定義ファイルおよびサーバ管理コマンドを使用して設定する場合

cjstartapp コマンドを使用して，案件運用操作を開始します。

注

cjstartapp コマンドを使用する場合，-jspc オプションを指定しないで実行してください。

開始方法については，マニュアル「Cosminexus 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」の「J2EE アプリケーションの実行」の説明を参照してください。

cjstartapp コマンドの実行例を次に示します。

```
cjstartapp <サーバ名称> -name CSCIWAdminServlet
```

(e) コンテキストルートの設定

案件運用操作のコンテキストルートのデフォルトは「CSCIWAdminServlet」です。

J2EE サーバのフロントエンドにリバースプロキシを配置する場合，コンテキストルート「CSCIWAdminServlet」のリクエストを転送する設定をしてください。

(7) アプリケーション呼び出しサービスの設定

アプリケーション呼び出しサービスを使用する場合は，アプリケーション呼び出しサービスの設定をします。

アプリケーション呼び出しサービスは，アプリケーション呼び出しサービスを実行する Cosminexus に組み込んでください。すべての Cosminexus に対して組み込む必要はありません。

アプリケーション呼び出しサービスの設定方法を次に示します。

1. アプリケーション呼び出しサービスをインポートする
2. アプリケーション呼び出しサービスのプロパティを設定する
   DB Connector のリソースアダプタの表示名に「DB_Connector_for_CSCIW」以外を設定した場合だけ，データソース表示名を変更します。
3. アプリケーション呼び出しサービスを開始する

詳細な設定方法を次に示します。

(a) アプリケーション呼び出しサービスのインポート

アプリケーション呼び出しサービスのファイルの格納パスは，次のとおりです。

- Windows の場合

```
<CSCIWインストールフォルダ>¥lib¥csciwapsrv.ear
```

- UNIX の場合

```
/opt/hitachi/CSCIW/lib/csciwapsrv.ear
```

次のどちらかの手順を実行して，アプリケーション呼び出しサービスをインポートしてください。

■ 運用管理ポータルを使用して設定する場合

次の手順で操作してください。

1. 運用管理ポータルにログインする
2. 運用管理ポータルで［論理サーバのアプリケーション管理］アンカーをクリックする
3. ツリーペインで次の順にクリックする

［<操作対象の運用管理ドメイン名>］－［論理 J2EE サーバ］－［J2EE サーバ］－［<操作対象の J2EE サーバ名>］－［アプリケーション］

4. 右ペインの［インポート］タブをクリックし，［J2EE アプリケーションのインポート］画面を表示する

5. ［インポートディレクトリ］にアプリケーション呼び出しサービスのファイルが格納されているパスを指定して［適用］ボタンをクリックする

6. ［J2EE アプリケーションファイル］で「csciwapsrv.ear」を選択して［実行］ボタンをクリックする

詳細は，マニュアル「Cosminexus アプリケーションサーバ 運用管理ポータル操作ガイド」の「J2EE アプリケーションの設定」を参照してください。

■ 定義ファイルおよびサーバ管理コマンドを使用して設定する場合

cjimportapp コマンドを使用して，インポートします。

インポート方法については，マニュアル「Cosminexus 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」の「J2EE アプリケーションのインポート」の説明を参照してください。

cjimportapp コマンドの実行例を次に示します。

```
cjimportapp <サーバ名称> -f <csciwapsrv.earファイルのパス>
```

(b)　アプリケーション呼び出しサービスのプロパティ設定

データベースと接続するための DB Connector のリソースアダプタの表示名に「DB_Connector_for_CSCIW」を設定した場合，この手順は不要です。

「DB_Connector_for_CSCIW」以外を設定した場合は，データソース表示名をリソースアダプタと同じ表示名に変更してください。変更方法については「6.2.6　アプリケーション呼び出しサービスが使用するデータソース表示名の変更」を参照してください。

(c)　アプリケーション呼び出しサービスの開始

次のどちらかの手順を実行して，アプリケーション呼び出しサービスを開始してください。

■ 運用管理ポータルを使用して設定する場合

次の手順で操作してください。

1. 運用管理ポータルにログインする

2. 運用管理ポータルで［論理サーバのアプリケーション管理］アンカーをクリックする

3. ツリーペインで次の順にクリックする
［<操作対象の運用管理ドメイン名>］－［論理 J2EE サーバ］－［J2EE サーバ］－［<操作対象の J2EE サーバ名>］－［アプリケーション］

4. 右ペインの［開始／停止］タブをクリックし，［J2EE アプリケーションの開始／停止］画面を表示する

5. アプリケーション名「CSCIWBpmnAPService01」の［開始］アンカーをクリックする

6. 内容を確認して，［はい］ボタンをクリックする

詳細は，マニュアル「Cosminexus アプリケーションサーバ 運用管理ポータル操作ガイド」の「J2EE アプリケーションの設定」を参照してください。

■ 定義ファイルおよびサーバ管理コマンドを使用して設定する場合

cjstartapp コマンドを使用して，アプリケーション呼び出しサービスを開始します。

開始方法については，マニュアル「Cosminexus 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」の「J2EE アプリケーションの開始」の説明を参照してください。

cjstartapp コマンドの実行例を次に示します。

```
cjstartapp <サーバ名称> -name CSCIWBpmnAPService01
```

メモ

同一の J2EE サーバ上にアプリケーション呼び出しサービスと REST アプリケーションを配備している場合は，REST アプリケーションの開始時の順番をアプリケーション呼び出しサービスの開始時の順番（10）より小さくしてください。

開始時の順番は，アプリケーション属性ファイルのhitachi-application-property/start-orderで設定します。

4.5.4 Java アプリケーションの設定

Java アプリケーションで CSCIW を使用する場合は，環境変数と Java 実行時の-classpath オプションの設定が必要です。Java オブジェクト呼び出しを利用する場合は，次の設定に加えて「4.5.5(3) Java アプリケーションの場合」も参照して，設定してください。また，Java アプリケーションを起動する場合は，使用する OS が Windows のときには管理者特権モードで，UNIX のときには root 権限で起動してください。

(1) 環境変数の設定

Java アプリケーションを実行するシェルに，次に示す環境変数を設定してください。

- Windows の場合

```
CSCIW_HOME=<CSCIWインストールフォルダ>
```

- UNIX の場合

```
CSCIW_HOME=/opt/hitachi/CSCIW
```

なお，CSCIW_CONF_DIR 環境変数でシステム ID を切り替えたい場合は，CSCIW_CONF_DIR 環境変数も同様に設定してください。

(2) JAR ファイルの追加

Java アプリケーションを実行する際の-classpath オプションに，次に示すファイルを追加してください。

- Windows の場合

```
<プログラムファイルフォルダ>¥Hitachi¥HNTRLib2¥classes¥hntrlib2j.jar
<CSCIWインストールフォルダ>¥lib¥csciw.jar
```

> **重要**
>
> Windows 環境で，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のインストール先を変更している場合は，変更したインストール先ディレクトリにあるhntrlib2j.jarを追加する必要があります。

- UNIX の場合

```
/opt/hitachi/HNTRLib2/classes/hntrlib2j.jar
/opt/hitachi/CSCIW/lib/csciw.jar
```

また，使用する JDBC ドライバ（HiRDB の場合は「HiRDB Type4 JDBC Driver」，ORACLE の場合は「Oracle JDBC Thin Driver」，SQL Server の場合は「SQL Server の JDBC ドライバ」）を-classpath オプションに追加してください。

なお，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の 64bit 版を必要とする場合は，hntrlib2j.jar の代わりに hntrlib2j64.jar を指定してください。

4.5.5 Java オブジェクト呼び出しを利用する場合の設定

Java オブジェクト呼び出しを利用する場合は，Java オブジェクトの組み込みが必要です。

Java オブジェクト呼び出しを利用するには，Java オブジェクトの組み込みのほかに，アプリケーションサーバおよび Java アプリケーションの設定が必要です。アプリケーションサーバの設定については，「4.5.3 アプリケーションサーバの設定」を，Java アプリケーションの設定については，「4.5.4 Java アプリケーションの設定」を参照してください。

アプリケーションサーバ上で Java オブジェクト呼び出しを利用する場合，Java オブジェクトは組み込む場所によって，設定方法が異なります。

- J2EE アプリケーションに Java オブジェクトを組み込むとき

- コンテナ拡張ライブラリに Java オブジェクトを組み込むとき

それぞれの設定方法の特徴を次の表に示します。特徴を参考に設定方法を選択してください。

表 4-11　Java オブジェクト呼び出しを利用する場合の設定方法の特徴

Java オブジェクトを組み込む場所	特徴
J2EE アプリケーション	Java オブジェクトの入れ替えは，J2EE アプリケーションの Java オブジェクトのライブラリ JAR 入れ替え，および J2EE アプリケーションの再起動で完了します。 ただし，CSCIW のすべての J2EE アプリケーションに Java オブジェクトのライブラリ JAR と csciwj2ee.jar の組み込みが必要になります。
コンテナ拡張ライブラリ	既存の J2EE アプリケーションの設定の変更は不要です。 ただし，Java オブジェクトを入れ替える場合，J2EE サーバの再起動が必要になります。

(1)　J2EE アプリケーションに Java オブジェクトを組み込む場合

J2EE アプリケーションに Java オブジェクトを組み込む場合の設定方法を次に示します。

(a)　コンテナ拡張ライブラリの設定

「4.5.3(1)　コンテナ拡張ライブラリへの取り込み」で組み込んだcsciw.jar ファイルをcsciwcmn.jar ファイルに変更します。

●J2EE サーバ用オプション定義ファイルへの設定

Cosminexus の J2EE サーバ用オプションのadd.class.path に，コンテナ拡張ライブラリ用のcsciwcmn.jar ファイルを設定してください。

- Windows の場合

  add.class.path=<CSCIW インストールフォルダ>¥lib¥csciwcmn.jar

- UNIX の場合

  add.class.path=/opt/hitachi/CSCIW/lib/csciwcmn.jar

add.class.path については，マニュアル「Cosminexus 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）」の「usrconf.cfg（J2EE サーバ用オプション定義ファイル）」の説明を参照してください。

●サーバ管理コマンド用オプション定義ファイルへの設定

サーバ管理コマンド用オプション定義ファイルに，コンテナ拡張ライブラリ用のcsciwcmn.jar ファイルを設定してください。

- Windows の場合

  usrconf.bat（サーバ管理コマンド用オプション定義ファイル）の USRCONF_JVM_CLASSPATH キーに，csciwcmn.jar ファイルを設定します。

  set USRCONF_JVM_CLASSPATH=<CSCIW インストールフォルダ>¥lib¥csciwcmn.jar

usrconf.bat（サーバ管理コマンド用オプション定義ファイル）については，マニュアル「Cosminexus 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)」の「usrconf.bat（サーバ管理コマンド用オプション定義ファイル)」の説明を参照してください。

- UNIX の場合

  usrconf（サーバ管理コマンド用オプション定義ファイル）の USRCONF_JVM_CLPATH キーに，csciwcmn.jar ファイルを設定します。

  ```
  set USRCONF_JVM_CLPATH=/opt/hitachi/CSCIW/lib/csciwcmn.jar
  ```

  usrconf（サーバ管理コマンド用オプション定義ファイル）については，マニュアル「Cosminexus 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)」の「usrconf（サーバ管理コマンド用オプション定義ファイル)」の説明を参照してください。

(b) 案件運用操作の設定

案件運用操作に Java オブジェクトのライブラリ JAR を組み込んでください。次に示すコマンドを実行すると組み込むことができます。

```
cjimportlibjar <サーバ名称> -name CSCIWAdminServlet -f <Javaオブジェクトのライブラリ JARファイルパス>
```

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

(c) J2EE アプリケーションの設定

J2EE アプリケーションに Java オブジェクトのライブラリ JAR，およびcsciwj2ee.jar ファイルを組み込んでください。次に示すコマンドを実行すると組み込むことができます。

- Java オブジェクトのライブラリ JAR の場合

  ```
  cjimportlibjar <サーバ名称> -name <業務アプリケーション名> -f <Javaオブジェクトのライブラリ JARファイルパス>
  ```

- csciwj2ee.jar ファイルの場合

  ```
  cjimportlibjar <サーバ名称> -name <業務アプリケーション名> -f <csciwj2ee.jarファイルパス>
  ```

csciwj2ee.jar のファイルパスは，次のとおりです。

- Windows の場合

  ```
  <CSCIWインストールフォルダ>¥lib¥csciwj2ee.jar
  ```

- UNIX の場合

  ```
  /opt/hitachi/CSCIW/lib/csciwj2ee.jar
  ```

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

(2) コンテナ拡張ライブラリに Java オブジェクトを組み込む場合

コンテナ拡張ライブラリに Java オブジェクトを組み込む場合の設定方法を次に示します。

(a) コンテナ拡張ライブラリの設定

J2EE サーバ用オプション定義ファイルのコンテナ拡張ライブラリに，Java オブジェクトのライブラリ JAR を設定してください。

```
add.class.path=<Javaオブジェクトのライブラリ JARファイルパス>
```

add.class.path については，マニュアル「Cosminexus 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)」の「usrconf.cfg（J2EE サーバ用オプション定義ファイル)」の説明を参照してください。

(3) Java アプリケーションの場合

Java アプリケーションを実行する場合の-classpath オプションに，Java オブジェクトのライブラリ JAR へのファイルパスを追加してください。

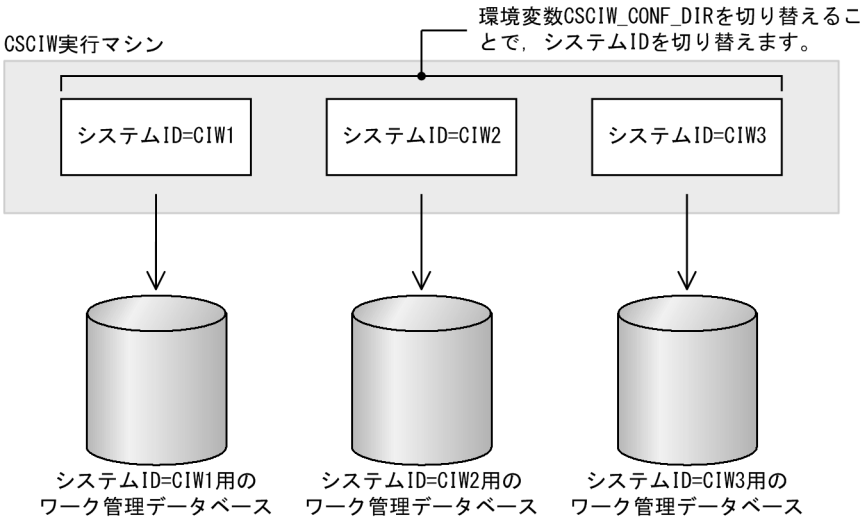
4.6　注意事項

この節では，利用環境とするマシンや，実行する機能（プロセス）などに合わせて，システム構成を設定する場合の注意事項，および業務プログラムの実行権限を設定する場合の注意事項について説明します。

4.6.1　マルチインスタンス構成を設定する場合の注意事項

マルチインスタンスの構成例と設定する場合の注意事項を次に示します。

図 4-5　マルチインスタンスの構成例

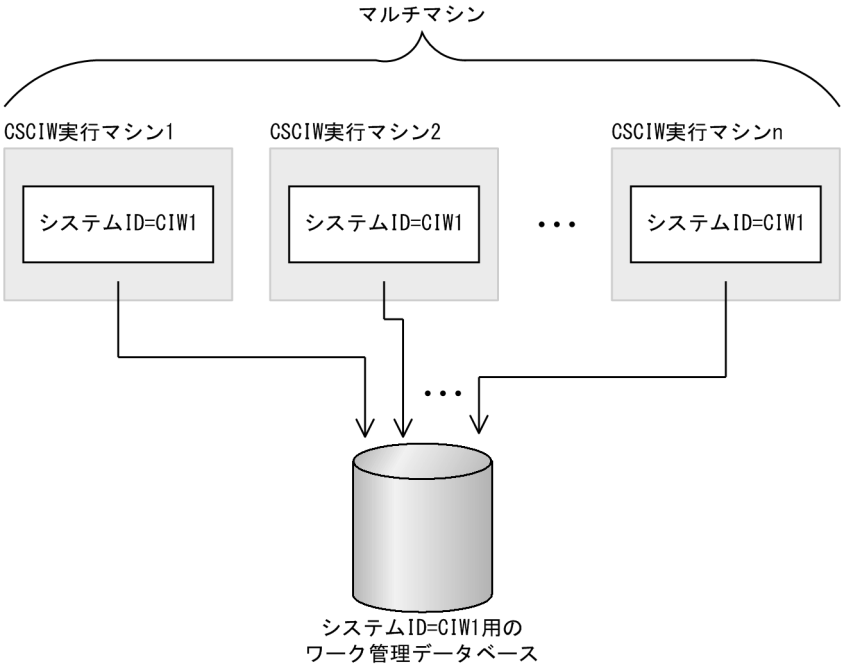


- マルチインスタンス構成は，CSCIW_CONF_DIR 環境変数を切り替えることで，複数のシステム ID の CSCIW を同一マシン内で運用できます。ただし，JavaVM では，1 プロセスで設定できるシステム ID は 1 つです。
- マルチインスタンス構成にするには，システム ID ごとにワーク管理データベースを構築し，CSCIW の実行環境を構築する必要があります。ワーク管理データベースの構築については「4.3　ワーク管理データベースの構築」を，CSCIW の実行環境の構築については，「4.5　実行環境での環境設定」をそれぞれ参照してください。

4.6.2　マルチマシン構成を設定する場合の注意事項

マルチマシンの構成例と設定する場合の注意事項を次に示します。

図 4-6　マルチマシンの構成例

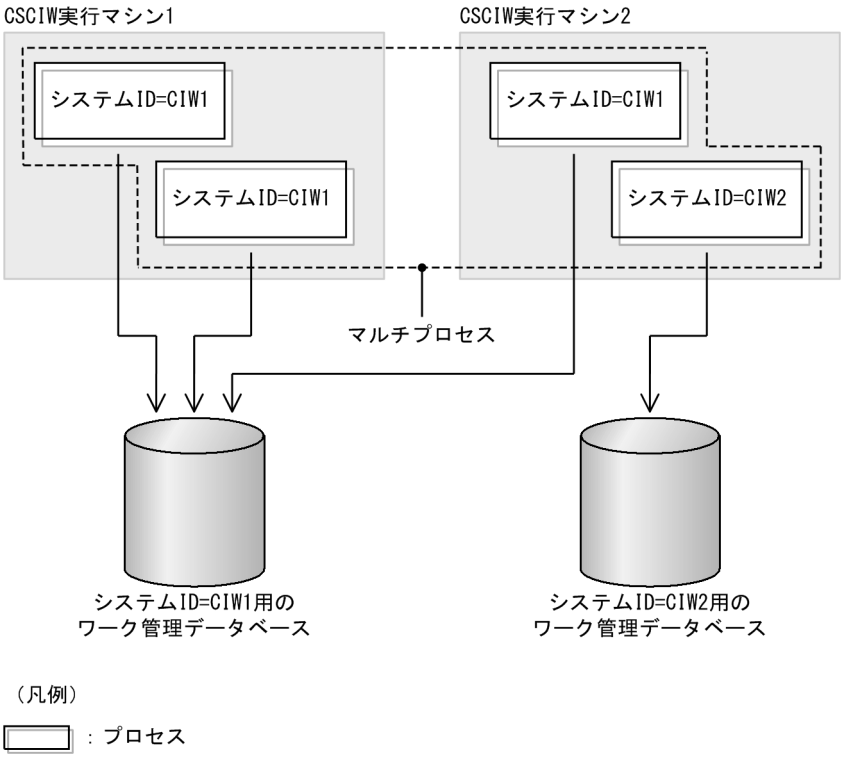


- マルチマシン構成は，複数の異なるマシン上で，同一のシステム ID によって CSCIW を運用できます。複数のマシンで同一のシステム ID の CSCIW を動作させるには，マシンを増設するごとに CSCIW をインストールし，CSCIW の実行環境の構築と環境変数の設定が必要です。CSCIW の実行環境の構築については，「4.5　実行環境での環境設定」を，環境変数の設定については，「4.5.2(1)　環境変数の設定」を参照してください。
- マルチマシンを構成しているすべてのマシンの IP アドレスが一意になるように設定してください。

4.6.3　マルチプロセス構成を設定する場合の注意事項

マルチプロセスの構成例と設定する場合の注意事項を次に示します。

図 4-7　マルチプロセスの構成例



マルチプロセス構成は，マルチインスタンス構成とマルチマシン構成の両方の機能を兼ね備えた形で，CSCIW の複数のプロセスを同時に運用できます。マルチプロセス構成を設定する場合の注意事項については，「4.6.1　マルチインスタンス構成を設定する場合の注意事項」および「4.6.2　マルチマシン構成を設定する場合の注意事項」を参照してください。

4.6.4　クラスタ構成を設定する場合の注意事項

CSCIW でクラスタ構成を構築するには，Cosminexus がサポートするクラスタソフトウェアを使用します。通常運用するマシン環境（実行系）と待機系のマシン環境は，すべて同じ設定をしてください。クラスタ構成の構築方法，およびクラスタ構成の注意事項については，マニュアル「Cosminexus アプリケーションサーバ 機能解説 運用／監視／連携編」の「1:1 系切り替えシステム（クラスタソフトウェアとの連携)」の説明を参照してください。

4.6.5　業務プログラムの実行権限を設定する場合の注意事項

業務プログラムの実行権限を設定する場合の注意事項を次に示します。

(1)　UNIX の場合の注意事項

- 業務プログラムの実行ユーザには root 権限が必要です。

- 業務プログラムの実行権限を変更したい場合，OS のコマンドや機能を使用して，実行権限のグループを変更できます。なお，実行権限のグループを変更する場合は，CSCIW インストールディレクトリ下のディレクトリおよびファイルを変更する必要があります。
- 製品のデフォルト（CSCIW をインストールしただけ）の状態では，CSCIW の構築や設定は，root ユーザで実行する必要がありますが，製品提供ファイルの所有者とグループを次のように変更することで，root 以外のユーザが構築や設定を実行できるようになります。

```
chown <owner>:<group> /opt/hitachi/CSCIW
chown -R <owner>:<group> /opt/hitachi/CSCIW/bin
chown -R <owner>:<group> /opt/hitachi/CSCIW/bpmn
chown -R <owner>:<group> /opt/hitachi/CSCIW/conf
chown -R <owner>:<group> /opt/hitachi/CSCIW/lib
chown -R <owner>:<group> /opt/hitachi/CSCIW/lock
chown -R <owner>:<group> /opt/hitachi/CSCIW/log
```

  所有者には，Component Container 管理者を指定し，グループには，Component Container 管理者グループを指定してください。

(2) Windows の場合の注意事項

業務プログラムの実行ユーザには，管理者特権が必要です。

5 ワーク管理システムの削除

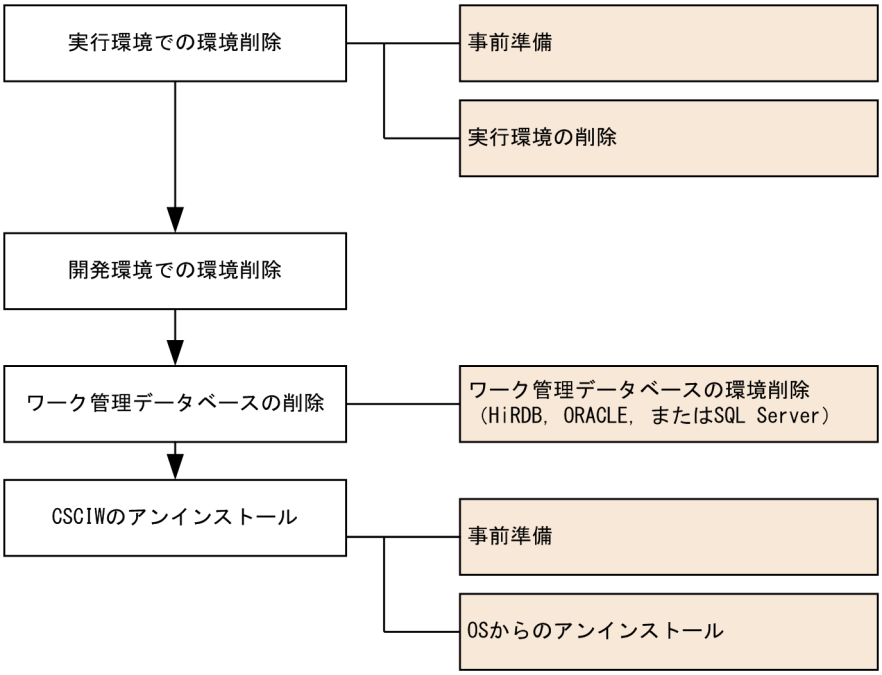
この章では，ワーク管理システムの削除方法について説明します。

5.1　システム削除の流れ

この節では，ワーク管理システムを削除する流れについて説明します。

ワーク管理システムを削除する流れを，次の図に示します。

図 5-1　ワーク管理システムの削除の流れ



上の図に示したワーク管理システムの削除の流れに従って，次の節以降で削除方法を説明します。

ヒント

BPMN 連携機能を使用する場合のシステム削除の流れおよび削除方法については，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。

5.2　実行環境での環境削除

この節では，CSCIW の実行環境での削除方法について説明します。

5.2.1　事前準備

ワーク管理システムの実行環境を削除する前の準備について説明します。

(1)　業務アプリケーションの停止および削除

(a)または(b)の手順を実行して，業務アプリケーションを停止および削除してください。

(a)　運用管理ポータルを使用する場合

次の手順で操作してください。

1. 運用管理ポータルにログインする
2. 運用管理ポータルで［論理サーバのアプリケーション管理］アンカーをクリックする
3. ツリーペインで次の順にクリックする
   ［<操作対象の運用管理ドメイン名>］－［論理 J2EE サーバ］－［J2EE サーバ］－［<操作対象の J2EE サーバ名>］－［アプリケーション］
4. 右ペインの［開始／停止］タブをクリックし，［J2EE アプリケーションの開始／停止］画面を表示する
5. 停止する業務アプリケーションの［停止］アンカーをクリックする
6. 内容を確認して，［はい］ボタンをクリックする
7. 右ペインの［削除］タブをクリックし，［J2EE アプリケーションの削除］画面を表示する
8. 削除する業務アプリケーションの［削除］アンカーをクリックする
9. 内容を確認して，［はい］ボタンをクリックする

(b)　サーバ管理コマンドを使用する場合

次の手順で操作してください。

1. cjstopapp コマンドを使用して，CSCIW を使用している業務アプリケーションを停止する
   J2EE アプリケーションの停止方法については，マニュアル「Cosminexus 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」の「J2EE アプリケーションの停止」の説明を参照してください。
   cjstopapp コマンドの実行例を次に示します。

```
cjstopapp 〈サーバ名称〉 -name 〈業務アプリケーション名〉
```

2. cjdeleteapp コマンドを使用して，業務アプリケーションを削除する

J2EE アプリケーションの削除方法については，マニュアル「Cosminexus 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」の「J2EE アプリケーションの削除」の説明を参照してください。

cjdeleteapp コマンドの実行例を次に示します。

```
cjdeleteapp <サーバ名称> -name <業務アプリケーション名>
```

(2) 案件運用操作の停止および削除

(a)または(b)の手順を実行して，案件運用操作を停止および削除してください。

(a) 運用管理ポータルを使用する場合

次の手順で操作してください。

1. 運用管理ポータルにログインする
2. 運用管理ポータルで［論理サーバのアプリケーション管理］アンカーをクリックする
3. ツリーペインで次の順にクリックする

［<操作対象の運用管理ドメイン名>］－［論理 J2EE サーバ］－［J2EE サーバ］－［<操作対象の J2EE サーバ名>］－［アプリケーション］

4. 右ペインの［開始／停止］タブをクリックし，［J2EE アプリケーションの開始／停止］画面を表示する
5. アプリケーション名「CSCIWAdminServlet」の［停止］アンカーをクリックする
6. 内容を確認して，［はい］ボタンをクリックする
7. 右ペインの［削除］タブをクリックし，［J2EE アプリケーションの削除］画面を表示する
8. アプリケーション名「CSCIWAdminServlet」の［削除］アンカーをクリックする
9. 内容を確認して，［はい］ボタンをクリックする

(b) サーバ管理コマンドを使用する場合

次の手順で操作してください。

1. cjstopapp コマンドを使用して，案件運用操作を停止する

停止方法については，マニュアル「Cosminexus 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」の「J2EE アプリケーションの停止」の説明を参照してください。

cjstopapp コマンドの実行例を次に示します。

```
cjstopapp <サーバ名称> -name CSCIWAdminServlet
```

2. cjdeleteapp コマンドを使用して，案件運用操作を削除する

削除方法については，マニュアル「Cosminexus 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」の「J2EE アプリケーションの削除」の説明を参照してください。

cjdeleteapp コマンドの実行例を次に示します。

```
cjdeleteapp <サーバ名称> -name CSCIWAdminServlet
```

(3) アプリケーション呼び出しサービスの停止および削除

(a)または(b)の手順を実行して，アプリケーション呼び出しサービスを停止および削除してください。

(a) 運用管理ポータルを使用する場合

次の手順で操作してください。

1. 運用管理ポータルにログインする
2. 運用管理ポータルで［論理サーバのアプリケーション管理］アンカーをクリックする
3. ツリーペインで次の順にクリックする
   ［<操作対象の運用管理ドメイン名>］－［論理 J2EE サーバ］－［J2EE サーバ］－［<操作対象の J2EE サーバ名>］－［アプリケーション］
4. 右ペインの［開始／停止］タブをクリックし，［J2EE アプリケーションの開始／停止］画面を表示する
5. アプリケーション名「CSCIWBpmnAPService01」の［停止］アンカーをクリックする
6. 内容を確認して，［はい］ボタンをクリックする
7. 右ペインの［削除］タブをクリックし，［J2EE アプリケーションの削除］画面を表示する
8. アプリケーション名「CSCIWBpmnAPService01」の［削除］アンカーをクリックする
9. 内容を確認して，［はい］ボタンをクリックする

(b) サーバ管理コマンドを使用する場合

次の手順で操作してください。

1. cjstopapp コマンドを使用して，アプリケーション呼び出しサービスを停止する
   停止方法については，マニュアル「Cosminexus 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」の「J2EE アプリケーションの停止」の説明を参照してください。
   cjstopapp コマンドの実行例を次に示します。

```
cjstopapp <サーバ名称> -name CSCIWBpmnAPService01
```

2. cjdeleteapp コマンドを使用して，アプリケーション呼び出しサービスを削除する

削除方法については，マニュアル「Cosminexus 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」の「J2EE アプリケーションの削除」の説明を参照してください。

cjdeleteapp コマンドの実行例を次に示します。

```
cjdeleteapp <サーバ名称> -name CSCIWBpmnAPService01
```

(4) CSCIWManagementServer の停止および削除

(a)または(b)の手順を実行して，CSCIWManagementServer を停止および削除してください。

(a) 運用管理ポータルを使用する場合

次の手順で操作してください。

1. 運用管理ポータルにログインする
2. 運用管理ポータルで［論理サーバのアプリケーション管理］アンカーをクリックする
3. ツリーペインで次の順にクリックする
   ［<操作対象の運用管理ドメイン名>］－［論理 J2EE サーバ］－［J2EE サーバ］－［<操作対象の J2EE サーバ名>］－［アプリケーション］
4. 右ペインの［開始／停止］タブをクリックし，［J2EE アプリケーションの開始／停止］画面を表示する
5. アプリケーション名「CSCIWManagementServer」の［停止］アンカーをクリックする
6. 内容を確認して，［はい］ボタンをクリックする
7. 右ペインの［削除］タブをクリックし，［J2EE アプリケーションの削除］画面を表示する
8. アプリケーション名「CSCIWManagementServer」の［削除］アンカーをクリックする
9. 内容を確認して，［はい］ボタンをクリックする

(b) サーバ管理コマンドを使用する場合

次の手順で操作してください。

1. cjstopapp コマンドを使用して，CSCIWManagementServer を停止する
   停止方法については，マニュアル「Cosminexus 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」の「J2EE アプリケーションの停止」の説明を参照してください。
   cjstopapp コマンドの実行例を次に示します。

```
cjstopapp <サーバ名称> -name CSCIWManagementServer
```

2. cjdeleteapp コマンドを使用して，CSCIWManagementServer を削除する

削除方法については，マニュアル「Cosminexus 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」の「J2EE アプリケーションの削除」の説明を参照してください。

cjdeleteapp コマンドの実行例を次に示します。

```
cjdeleteapp <サーバ名称> -name CSCIWManagementServer
```

(5) J2EE サーバの停止

(a)または(b)の手順を実行して，CSCIW を使用している J2EE サーバを停止してください。

(a) 運用管理ポータルを使用する場合

運用管理ポータルで［論理サーバの起動／停止］アンカーをクリックし，対象の J2EE サーバを停止します。

詳細は，マニュアル「Cosminexus アプリケーションサーバ 運用管理ポータル操作ガイド」の「論理サーバの起動／停止の設定」を参照してください。

(b) サーバ管理コマンドを使用する場合

cjstopsv コマンドを使用して，J2EE サーバを停止します。

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

cjstopsv コマンドの実行例を次に示します。

```
cjstopsv <サーバ名称>
```

(6) 環境変数の削除

「4.5.3(2) 環境変数の取り込み」で設定した環境変数を削除してください。なお，運用管理ポータルで環境変数を設定した場合は，運用管理ポータルで削除します。

(7) コンテナ拡張ライブラリからの削除

コンテナ拡張ライブラリの設定をしている場合は，コンテナ拡張ライブラリからコンテナ拡張ライブラリ用の JAR ファイルを削除します。

削除するのは，「4.5.3(1) コンテナ拡張ライブラリへの取り込み」で組み込んだファイルです。なお，運用管理ポータルでコンテナ拡張ライブラリを設定した場合は，運用管理ポータルで削除します。

5.2.2 実行環境の削除

実行環境を削除する方法を説明します。

実行環境の削除は次の手順で実施します。

1. コマンドを実行して実行環境を削除します。
2. 環境変数を削除します。

(1) コマンドによる実行環境の削除

CSCIW の実行環境は，ciwsetenv（環境の構築または削除）コマンドを実行して削除します。

次に示す形式でコマンドを実行すると，CSCIW の実行環境を削除できます。

```
ciwsetenv -sid 〈システムID〉 -del
```

ciwsetenv（環境の構築または削除）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) 環境変数の削除

CSCIW の実行環境を削除したあと，環境変数 CSCIW_HOME を削除します。

なお，マルチインスタンス構成のシステムを構築している場合は，環境変数 CSCIW_CONF_DIR も削除してください。

5.3　開発環境での環境削除

ビジネスプロセス開発環境の削除は，Eclipse に組み込んだビジネスプロセス管理ツール（CSCIW-Definer）を削除したあとに，uCosminexus Business Process Developer をアンインストールすることで削除できます。手順については，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。

5.4　ワーク管理データベースの削除

この節では，ワーク管理データベースの削除方法について説明します。

なお，データベースの削除は，削除したいシステム ID が設定されているすべてのマシンの実行環境を削除した状態で実施してください。

5.4.1　ワーク管理データベースの環境削除（HiRDB の場合）

(1)　ワーク管理データベースの削除（HiRDB の場合）

ワーク管理データベースのテーブルやインデクスを HiRDB から削除する方法を説明します。

テーブルやインデクスは，SQL スクリプトファイルを使用して削除してください。

(a)　SQL スクリプトファイルの編集

テーブルやインデクスは，次に示す SQL スクリプトファイルを編集して削除します。

```
<CSCIWインストールディレクトリ>/sql/droptable_hirdb.sql
<CSCIWインストールディレクトリ>/sql/droptableex_hirdb.sql※
```

注※

アプリケーション呼び出しサービスを使用する場合だけ編集してください。

SQL スクリプトファイル中の文字列を，テキストエディタを使用してワーク管理データベース用に書き換えてください。テーブルやインデクスを削除する際に SQL スクリプトファイルの書き換えが必要な文字列，および書き換える内容を次に示します。

表 5-1　テーブルやインデクスの削除時に書き換えが必要な SQL スクリプトファイル中の文字列（HiRDB の場合）

書き換えが必要な文字列	書き換える内容
<SYSTEMID>	システム ID ワーク管理データベースを作成した際に設定したシステム ID に置換してください。

(b)　SQL スクリプトファイルを実行

編集した SQL スクリプトファイルを，HiRDB SQL Executer を使用して実行します。

pdsql コマンドに編集した SQL スクリプトファイルを指定し，実行することで，テーブルやインデクスを削除できます。HiRDB SQL Executer の使用方法については，マニュアル「HiRDB 解説」を参照してください。

pdsql コマンドの指定例を次に示します。

```
pdsql -u <接続認可識別子>/<パスワード>
      -h <HiRDBサーバのホスト名またはIPアドレス>
      -n <HiRDBサーバのポート番号>
       < <編集したSQLスクリプトファイルのパス>
```

注

各オプションの指定値は，ワーク管理データベースを作成した際に設定した値と同じにしてください。

(2) データベースへのアクセス権限の削除（HiRDB の場合）

ワーク管理データベースのテーブルやインデクスへ付与したアクセス権限を HiRDB から削除する方法を示します。

「5.4.1(1) ワーク管理データベースの削除（HiRDB の場合）」の操作を実施してワーク管理データベースのテーブルを削除済みの場合は，アクセス権限の削除は不要です。

アクセス権限は，SQL スクリプトファイルを使用して削除してください。

(a) SQL スクリプトファイルの編集

テーブルやインデクスへ付与したアクセス権限は，次に示す SQL スクリプトファイルを編集して削除します。

```
<CSCIWインストールディレクトリ>/sql/revoketable_hirdb.sql
<CSCIWインストールディレクトリ>/sql/revoketableex_hirdb.sql※
```

注※

アプリケーション呼び出しサービスを使用する場合だけ編集してください。

SQL スクリプトファイル中の文字列を，テキストエディタを使用してワーク管理データベースへアクセスできるように書き換えてください。アクセス権限を削除する際に SQL スクリプトファイルの書き換えが必要な文字列，および書き換える内容を次に示します。

表 5-2 アクセス権限を削除する際に書き換えが必要な SQL スクリプトファイル中の文字列（HiRDB の場合）

書き換えが必要な文字列	書き換える内容
<SCHEMANAME>	CSCIW の管理用の認可識別子 ワーク管理データベースのテーブルやインデクスを作成した認可識別子に置換してください。
<SYSTEMID>	システム ID ワーク管理データベースを作成した際に設定したシステム ID に置換してください。
<USERNAME>	業務アプリケーション用の認可識別子

書き換えが必要な文字列	書き換える内容
<USERNAME>	業務アプリケーションが接続するための認可識別子に置換してください。

(b) SQL スクリプトファイルを実行

編集した SQL スクリプトファイルを，HiRDB SQL Executer を使用して実行します。

pdsql コマンドに編集した SQL スクリプトファイルを指定し，実行することで，アクセス権限を削除できます。HiRDB SQL Executer の使用方法については，マニュアル「HiRDB 解説」を参照してください。

5.4.2 ワーク管理データベースの環境削除（ORACLE の場合）

(1) ワーク管理データベースの削除（ORACLE の場合）

ワーク管理データベースのテーブルやインデクスを ORACLE から削除する方法を説明します。

テーブルやインデクスは，SQL スクリプトファイルを使用して削除してください。

(a) SQL スクリプトファイルの編集

テーブルやインデクスは，次に示す SQL スクリプトファイルを編集して削除します。

```
<CSCIWインストールディレクトリ>/sql/droptable_oracle.sql
<CSCIWインストールディレクトリ>/sql/droptableex_oracle.sql※
```

注※

アプリケーション呼び出しサービスを使用する場合だけ編集してください。

SQL スクリプトファイル中の文字列を，テキストエディタを使用してワーク管理データベース用に書き換えてください。テーブルやインデクスを削除する際に SQL スクリプトファイルの書き換えが必要な文字列，および書き換える内容を次に示します。

表 5-3 テーブルやインデクスの削除時に書き換えが必要な SQL スクリプトファイル中の文字列（ORACLE の場合）

書き換えが必要な文字列	書き換える内容
<SYSTEMID>	システム ID ワーク管理データベースを作成した際に設定したシステム ID に置換してください。

重要

ORACLE の場合，SQL スクリプトファイル中に，DROP TABLE 文に PURGE オプションを使用しています。PURGE オプションが不要な場合は削除してください。DROP TABLE 文の PURGE オプションについては，ORACLE のマニュアルを参照してください。

(b) SQL スクリプトファイルを実行

編集した SQL スクリプトファイルを，SQL*Plus を使用して実行します。

sqlplus コマンドに編集した SQL スクリプトファイルを指定し，実行することで，テーブルやインデクスを削除できます。SQL*Plus の使用方法については，ORACLE のマニュアルを参照してください。

sqlplus コマンドの指定例を次に示します。

```
sqlplus  <接続ユーザ名>/<パスワード>@<Oracle Net 接続識別子>
         @<編集したSQLスクリプトファイルの絶対パス>
```

注

接続ユーザ名，パスワード，および Oracle Net 接続識別子は，ワーク管理データベースを作成した際に設定した値と同じにしてください。

(2) データベースへのアクセス権限の削除（ORACLE の場合）

ワーク管理データベースのテーブルやインデクスへ付与したアクセス権限を ORACLE から削除する方法を示します。

「5.4.2(1) ワーク管理データベースの削除（ORACLE の場合）」の操作を実施してワーク管理データベースのテーブルを削除済みの場合は，アクセス権限の削除は不要です。

アクセス権限は，SQL スクリプトファイルを使用して削除してください。

(a) SQL スクリプトファイルの編集

テーブルやインデクスへ付与したアクセス権限は，次に示す SQL スクリプトファイルを編集して削除します。

```
<CSCIWインストールディレクトリ>/sql/revoketable_oracle.sql
<CSCIWインストールディレクトリ>/sql/revoketableex_oracle.sql※
```

注※

アプリケーション呼び出しサービスを使用する場合だけ編集してください。

SQL スクリプトファイル中の文字列を，テキストエディタを使用してワーク管理データベースへアクセスできるように書き換えてください。アクセス権限を削除する際に SQL スクリプトファイルの書き換えが必要な文字列，および書き換える内容を次に示します。

表 5-4　アクセス権限を削除する際に書き換えが必要な SQL スクリプトファイル中の文字列（ORACLE の場合）

書き換えが必要な文字列	書き換える内容
<SCHEMANAME>	CSCIW の管理用のユーザ ワーク管理データベースのテーブルやインデクスを作成したユーザに置換してください。
<SYSTEMID>	システム ID ワーク管理データベースを作成した際に設定したシステム ID に置換してください。
<USERNAME>	業務アプリケーション用のユーザ 業務アプリケーションが接続するためのユーザに置換してください。

(b)　SQL スクリプトファイルを実行

編集した SQL スクリプトファイルを，SQL*Plus を使用して実行します。

sqlplus コマンドに編集した SQL スクリプトファイルを指定し，実行することで，アクセス権限を削除できます。SQL*Plus の使用方法については，ORACLE のマニュアルを参照してください。

5.4.3　ワーク管理データベースの環境削除（SQL Server の場合）

(1)　ワーク管理データベースの削除（SQL Server の場合）

ワーク管理データベースのテーブルやインデクスを SQL Server から削除する方法を説明します。

テーブルやインデクスは，SQL スクリプトファイルを使用して削除してください。

(a)　SQL スクリプトファイルの編集

テーブルやインデクスは，次に示す SQL スクリプトファイルを編集して削除します。

```
<CSCIWインストールディレクトリ>/sql/droptable_sqlserver.sql
```

SQL スクリプトファイル中の文字列を，テキストエディタを使用してワーク管理データベース用に書き換えてください。テーブルやインデクスを削除する際に SQL スクリプトファイルの書き換えが必要な文字列，および書き換える内容を次に示します。

表 5-5　テーブルやインデクスの削除時に書き換えが必要な SQL スクリプトファイル中の文字列（SQL Server の場合）

書き換えが必要な文字列	書き換える内容
<SYSTEMID>	システム ID ワーク管理データベースを作成した際に設定したシステム ID に置換してください。

(b) SQL スクリプトファイルを実行

編集した SQL スクリプトファイルを，sqlcmd ユティリティを使用して実行します。

sqlcmd ユティリティに編集した SQL スクリプトファイルを指定し，実行することで，テーブルやインデクスを削除できます。sqlcmd ユティリティの使用方法については，SQL Server のマニュアルを参照してください。

sqlcmd ユティリティの指定例を次に示します。

```
sqlcmd -U 〈接続ユーザ名〉
       -P 〈パスワード〉
       -i 〈編集したSQLスクリプトファイルの絶対パス〉
```

注

接続ユーザ名およびパスワードの指定値は，ワーク管理データベースを作成した際に設定した値と同じにしてください。

(2) データベースへのアクセス権限の削除（SQL Server の場合）

ワーク管理データベースのテーブルやインデクスへ付与したアクセス権限を SQL Server から削除する方法を示します。

「5.4.3(1) ワーク管理データベースの削除（SQL Server の場合）」の操作を実施してワーク管理データベースのテーブルを削除済みの場合は，アクセス権限の削除は不要です。

アクセス権限は SQL スクリプトファイルを使用して削除してください。

(a) SQL スクリプトファイルの編集

テーブルやインデクスへ付与したアクセス権限は，次に示す SQL スクリプトファイルを編集して削除します。

```
〈CSCIWインストールディレクトリ〉/sql/revoketable_sqlserver.sql
```

SQL スクリプトファイル中の文字列を，テキストエディタを使用してワーク管理データベースへアクセスできるように書き換えてください。アクセス権限を削除する際に SQL スクリプトファイルの書き換えが必要な文字列，および書き換える内容を次に示します。

表 5-6 アクセス権限を削除する際に書き換えが必要な SQL スクリプトファイル中の文字列（SQL Server の場合）

書き換えが必要な文字列	書き換える内容
〈SCHEMANAME〉	CSCIW の管理用のスキーマ ワーク管理データベースのテーブルやインデクスを作成したスキーマに置換してください。

書き換えが必要な文字列	書き換える内容
<SYSTEMID>	システム ID ワーク管理データベースを作成した際に設定したシステム ID に置換してください。
<USERNAME>	業務アプリケーション用のユーザ 業務アプリケーションが接続するためのユーザに置換してください。

(b) SQL スクリプトファイルを実行

編集した SQL スクリプトファイルを，sqlcmd ユティリティを使用して実行します。

sqlcmd ユティリティに編集した SQL スクリプトファイルを指定し，実行することで，アクセス権限を削除できます。sqlcmd ユティリティの使用方法については，SQL Server のマニュアルを参照してください。

5.5　CSCIW のアンインストール

この節では，CSCIW の削除方法について説明します。

5.5.1　事前準備

CSCIW を削除する前の準備について説明します。

(1)　実行環境の削除およびワーク管理データベースの削除

CSCIW をアンインストールする前に実行環境の削除を必ず実施してください。また，必要に応じてデータベース内に生成されたテーブルの削除を実施してください。実行環境の削除については，「5.2.2　実行環境の削除」を参照してください。データベース内に生成されたテーブルの削除については，「5.4　ワーク管理データベースの削除」を参照してください。

(2)　必要ファイルの退避

アンインストール時には環境設定ファイルも削除されます。環境設定ファイルが必要な場合は，アンインストール前に退避してください。

また，出力されたメッセージおよびトレースファイルの情報が必要な場合は，アンインストール前に退避してください。

5.5.2　OS からのアンインストール

CSCIW を OS からアンインストール方法について説明します。

(1)　Windows からのアンインストール

コントロールパネルの［アプリケーションの追加と削除］（または［プログラムの追加と削除］）を使用して，CSCIW をアンインストールする方法を示します。

1. Administrators グループのユーザでログインする
2. コントロールパネルの［アプリケーションの追加と削除］（または［プログラムの追加と削除］）をクリックする
   ［アプリケーションの追加と削除］（または［プログラムの追加と削除］）ダイアログボックスが表示されます。
3. 「uCosminexus Service Coordinator Interactive Workflow」を選択する
   製品名の後ろにインストールした際のバージョンも表示されるので，確認してください。

4. [変更と削除] ボタンをクリックする

削除を確認するダイアログが表示されます。

5. [はい] ボタンをクリックする

アンインストールが開始されます。アンインストールの中断はできません。

> **重要**
>
> このアンインストール操作では，インストール後に作成されたファイル，またはフォルダは削除されません。必要に応じて削除してください。

(2) UNIX からのアンインストール

Hitachi PP Installer を使用して，/opt/hitachi/CSCIW ディレクトリへインストールされている，CSCIW をアンインストールする方法を示します。

1. /etc/hitachi_setup を実行する
2. 「D) Delete」を選択する
3. 「uCosminexus Service Coordinator Interact」にカーソルを移動し，スペースキーで選択する

正しく選択すると，「uCosminexus Service Coordinator Interact」の左側に<@>が付きます。

なお，Hitachi PP Installer の仕様により製品名の「uCosminexus Service Coordinator Interact」の一部が欠落して表示される場合があります。

4. 「D) Delete」を選択後，最下行に表示されるメッセージに対して「y」または「Y」を入力する

アンインストールが開始されます。アンインストールの中断はできません。

5. アンインストールが終了したら，「Q) Quit」を選択して終了する

5.6　注意事項

この節では，マルチインスタンス構成の環境，およびマルチマシン構成の環境を削除する場合の注意事項について説明します。

5.6.1　マルチインスタンス構成の環境を削除する場合の注意事項

マルチインスタンス構成は，CSCIW_CONF_DIR 環境変数を切り替えることで，複数のシステム ID の CSCIW を同一マシン内で運用しています。環境を削除する場合は，次の注意が必要です。

(1)　実行環境の削除について

実行環境の削除は，システム ID ごとに実施する必要があります。

削除手順については，「5.2.2　実行環境の削除」を参照してください。

(2)　ワーク管理データベースの削除について

ワーク管理データベースの削除は，システム ID ごとに実施する必要があります。

削除手順については，「5.4　ワーク管理データベースの削除」を参照してください。

5.6.2　マルチマシン構成の環境を削除する場合の注意事項

マルチマシン構成は，複数の異なるマシン上で，同一のシステム ID によって CSCIW を運用しています。環境を削除する場合は，次の注意が必要です。

(1)　実行環境の削除について

実行環境の削除は，同一システム ID が稼働しているすべての実行環境で実施する必要があります。

削除手順については，「5.2.2　実行環境の削除」を参照してください。

(2)　ワーク管理データベースの削除について

ワーク管理データベースの削除は，同一システム ID が稼働しているすべて実行環境を削除したあと，ワーク管理データベースを構築したマシンに対して実施する必要があります。

削除手順については，「5.4　ワーク管理データベースの削除」を参照してください。

6 ワーク管理システムの変更

この章では，ワーク管理システムの変更方法について説明します。

6.1　システムの構成変更

この節では，ワーク管理システムの構成を変更する場合の手順について説明します。

6.1.1　実行環境マシンの IP アドレスの変更

CSCIW の実行環境マシンの IP アドレスの変更は，不要なプロセス起動情報が残っていないことを確認したあとに実施します。

不要なプロセス起動情報が残っているかどうかは，ciwlistsid（システム ID 情報の表示）で確認します。不要なプロセス起動情報が残っている場合は，ciwcleanup（プロセス情報の正常化）コマンドで不要な起動情報を削除してください。

不要なプロセス起動情報の確認および削除については，「7.9.3　業務アプリケーションが異常終了した場合や強制終了した（終了処理をしなかった）場合の影響について」を参照してください。

6.1.2　接続先 DBMS の変更

接続先 DBMS の変更について説明します。

接続先 DBMS の変更は，変更前の接続先情報を削除したあと，変更後の接続先情報を登録し直してください。

(1)　変更前の接続先情報の削除

接続先 DBMS の変更前の接続先情報は，ciwsetenv（環境の構築および削除）コマンドを実行して削除します。

ciwsetenv（環境の構築および削除）コマンドの指定例を次に示します。

```
ciwsetenv -sid <接続先を変更したいシステムID> -del
```

ciwsetenv（環境の構築および削除）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)　変更後の接続先情報の登録

接続先 DBMS の変更後の接続先情報は，ciwsetenv（環境の構築および削除）コマンドを実行して登録します。

ciwsetenv（環境の構築および削除）コマンドの指定例を次に示します。

```
ciwsetenv -sid 〈接続先を変更したいシステムID〉
          -f 〈変更後の情報を記述した環境構築ファイル名〉
```

ciwsetenv（環境の構築および削除）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。

6.1.3 DBMS の格納領域の変更（HiRDB の場合）

DBMS の機能を利用して RD エリアやスキーマを変更する方法を示します。

1. すべてのデータをアンロードする
2. すべてのテーブル・インデクス・ビューを削除する
3. すべてのテーブル・インデクス・ビューを再作成する
4. すべてのデータをリロードする
5. DBMS への接続先情報を変更する

詳細な変更方法を次に示します。

(1) すべてのデータのアンロード

すべてのデータをアンロードするには，テーブルごとにpdrorg コマンドを実行します。pdrorg コマンドについては，マニュアル「HiRDB コマンドリファレンス（UNIX(R)用）」または「HiRDB コマンドリファレンス（Windows(R)用）」を参照してください。

pdrorg コマンドの指定例を次に示します。

```
pdrorg -k unld
       -t 〈スキーマ名.テーブル名〉
       -W bin
       -u 〈移行元の認可識別子〉
       〈制御情報ファイル名〉
```

(2) すべてのテーブル・インデクス・ビューの削除

テーブル・インデクス・ビューを削除するには，次に示す SQL スクリプトファイルを編集します。

```
<CSCIWインストールディレクトリ>/sql/droptable_hirdb.sql
<CSCIWインストールディレクトリ>/sql/droptableex_hirdb.sql※
```

注※

アプリケーション呼び出しサービスを使用する場合だけ編集してください。

pdsql コマンドに編集した SQL スクリプトファイルを指定し，実行することで，すべてのテーブル・インデクス・ビューを削除できます。HiRDB SQL Executer の使用方法については，マニュアル「HiRDB 解説」を参照してください。

pdsql コマンドの指定例を次に示します。

```
pdsql -u <移行元の認可識別子>/<パスワード>
      -h <HiRDBサーバのホスト名またはIPアドレス>
      -n <HiRDBサーバのポート番号>
       < <削除用のSQLスクリプトファイルのパス>
```

(3) すべてのテーブル・インデクス・ビューの再作成

テーブル・インデクス・ビューを再作成するには，次に示す SQL スクリプトファイルを編集します。

```
<CSCIWインストールディレクトリ>/sql/createtable_hirdb.sql
<CSCIWインストールディレクトリ>/sql/createtableex_hirdb.sql※
```

注※

アプリケーション呼び出しサービスを使用する場合だけ編集してください。

pdsql コマンドに編集した SQL スクリプトファイルを指定し，実行することで，すべてのテーブル・インデクス・ビューを再作成できます。HiRDB SQL Executer の使用方法については，マニュアル「HiRDB 解説」を参照してください。

pdsql コマンドの指定例を次に示します。

```
pdsql -u <移行先の認可識別子>/<パスワード>
      -h <HiRDBサーバのホスト名またはIPアドレス>
      -n <HiRDBサーバのポート番号>
       < <再作成用のSQLスクリプトファイルのパス>
```

(4) すべてのデータのリロード

すべてのデータをリロードするには，テーブルごとにpdload コマンドを実行します。pdload コマンドについては，マニュアル「HiRDB コマンドリファレンス（UNIX(R)用)」または「HiRDB コマンドリファレンス（Windows(R)用)」を参照してください。

pdload コマンドの指定例を次に示します。

```
pdload -b
       -W
       -u <移行先の認可識別子>
       <スキーマ名.テーブル名>
       <制御情報ファイル名>
```

(5)　DBMS への接続先情報の変更

認可識別子またはスキーマを変更した場合は，変更後の認可識別子またはスキーマで接続できる認可識別子およびパスワードに変更する必要があります。

変更方法については，「6.1.2　接続先 DBMS の変更」を参照してください。

6.1.4　DBMS の格納領域の変更（ORACLE の場合）

DBMS の機能を利用して表領域やスキーマを変更する方法を示します。

1. すべてのデータをエクスポートする
2. すべてのテーブル・インデクス・ビューを削除する
3. すべてのテーブル・インデクス・ビューを再作成する
4. すべてのデータをインポートする
5. DBMS への接続先情報を変更する

詳細な変更方法を次に示します。

(1)　すべてのデータのエクスポート

すべてのデータをエクスポートするには，expdp コマンドを実行します。expdp コマンドについては，ORACLE のマニュアルを参照してください。

expdp コマンドの指定例を次に示します。

```
expdp <DBアクセスユーザ名1>/<DBアクセスパスワード>
      PARFILE=<パラメタファイル名>
```

パラメタファイルには，CSCIW のすべてのテーブル名を記述します。パラメタファイルの記述例を次に示します。

```
TABLES=<SYSTEMID>_PROCESS_DEF,<SYSTEMID>_ACTIVITY_DEF,…
CONTENT=DATA_ONLY
```

注

SYSTEMID には，ワーク管理データベースを一意に識別するためのシステム ID を指定してください。

重要

expdp コマンドを実行すると，デフォルトでディレクトリ・オブジェクト DATA_PUMP_DIR にダンプファイル・ログファイルを出力します。このため，DATA_PUMP_DIR に読み書

き権限のあるユーザで実行するか，DIRECTORY オプションなどを使用して読み書き権限のあるディレクトリ・オブジェクトを出力先に指定してください。また，ほかのスキーマからエクスポートする場合は，EXP_FULL_DATABASE 権限が必要です。expdp コマンドについては，ORACLE のマニュアルを参照してください。

(2) すべてのテーブル・インデクス・ビューの削除

テーブル・インデクス・ビューを削除するには，次に示す SQL スクリプトファイルを編集します。

```
<CSCIWインストールディレクトリ>/sql/droptable_oracle.sql
<CSCIWインストールディレクトリ>/sql/droptableex_oracle.sql※
```

注※

アプリケーション呼び出しサービスを使用する場合だけ編集してください。

削除用に編集した SQL スクリプトファイルは，SQL*Plus を使用して実行します。

sqlplus コマンドに編集した SQL スクリプトファイルを指定し，実行することで，すべてのテーブル・インデクス・ビューを削除できます。SQL*Plus の使用方法については，ORACLE のマニュアルを参照してください。

sqlplus コマンドの指定例を次に示します。

```
sqlplus  <DBアクセスユーザ名1>/<DBアクセスパスワード>
         @<Oracle Net 接続識別子>
         @<削除用のSQLスクリプトファイルの絶対パス>
```

(3) すべてのテーブル・インデクス・ビューの再作成

テーブル・インデクス・ビューを再作成するには，次に示す SQL スクリプトファイルを編集します。

```
<CSCIWインストールディレクトリ>/sql/createtable_oracle.sql
<CSCIWインストールディレクトリ>/sql/createtableex_oracle.sql※
```

注※

アプリケーション呼び出しサービスを使用する場合だけ編集してください。

再作成用に編集した SQL スクリプトファイルは，SQL*Plus を使用して実行します。sqlplus コマンドに編集した SQL スクリプトファイルを指定し，実行することで，すべてのテーブル・インデクス・ビューを再作成できます。SQL*Plus の使用方法については，ORACLE のマニュアルを参照してください。

sqlplus コマンドの指定例を次に示します。

```
sqlplus  <DBアクセスユーザ名2>/<DBアクセスパスワード>
         @<Oracle Net 接続識別子>
         @<再作成用のSQLスクリプトファイルの絶対パス>
```

(4)　すべてのデータのインポート

すべてのデータをインポートするには，テーブルごとにimpdp コマンドを実行します。impdp コマンドについては，ORACLE のマニュアルを参照してください。

impdp コマンドの指定例を次に示します。

```
impdp <DBアクセスユーザ名2>/<DBアクセスパスワード>
      [REMAP_SCHEMA=<移行元スキーマ>:<移行先スキーマ>]
```

重要

impdp コマンドを実行すると，デフォルトでディレクトリ・オブジェクト DATA_PUMP_DIR にダンプファイル・ログファイルを出力します。このため，DATA_PUMP_DIR に読み書き権限のあるユーザで実行するか，DIRECTORY オプションなどを使用して読み書き権限のあるディレクトリ・オブジェクトを出力先に指定してください。また，ほかのスキーマからインポートする場合は，IMP_FULL_DATABASE 権限が必要です。impdp コマンドについては，ORACLE のマニュアルを参照してください。

(5)　DBMS への接続先情報の変更

ユーザまたはスキーマを変更した場合は，変更後のユーザまたはスキーマで接続できるユーザおよびパスワードに変更する必要があります。

変更方法については，「6.1.2　接続先 DBMS の変更」を参照してください。

6.1.5　DBMS の格納領域の変更（SQL Server の場合）

DBMS の機能を利用してファイルグループやスキーマを変更する方法を示します。

1. すべてのデータをエクスポートする
2. すべてのテーブル・インデクス・ビューを削除する
3. すべてのテーブル・インデクス・ビューを再作成する
4. すべてのデータをインポートする
5. DBMS への接続先情報を変更する

詳細な変更方法を次に示します。

(1) すべてのデータのエクスポート

すべてのデータをエクスポートするには，bcp コマンドを実行します。bcp コマンドについては，SQL Server のマニュアルを参照してください。

bcp コマンドの指定例を次に示します。

```
bcp <テーブル名> out <データファイルの絶対パス>
       -U <ユーザ名>
       -P <パスワード>
       -c
```

(2) すべてのテーブル・インデクス・ビューの削除

テーブル・インデクス・ビューを削除するには，次に示す SQL スクリプトファイルを編集します。

```
<CSCIWインストールディレクトリ>/sql/droptable_sqlserver.sql
```

削除用に編集した SQL スクリプトファイルは，sqlcmd ユティリティを使用して実行します。

sqlcmd ユティリティに編集した SQL スクリプトファイルを指定し，実行することで，すべてのテーブル・インデクス・ビューを削除できます。sqlcmd ユティリティの使用方法については，SQL Server のマニュアルを参照してください。

sqlcmd ユティリティの指定例を次に示します。

```
sqlcmd -U <接続ユーザ名1>
       -P <パスワード>
       -i <削除用のSQLスクリプトファイルの絶対パス>
```

(3) すべてのテーブル・インデクス・ビューの再作成

テーブル・インデクス・ビューを再作成するには，次に示す SQL スクリプトファイルを編集します。

```
<CSCIWインストールディレクトリ>/sql/createtable_sqlserver.sql
```

再作成用に編集した SQL スクリプトファイルは，sqlcmd ユティリティを使用して実行します。sqlcmd ユティリティに編集した SQL スクリプトファイルを指定し，実行することで，すべてのテーブル・インデクス・ビューを再作成できます。sqlcmd ユティリティの使用方法については，SQL Server のマニュアルを参照してください。

sqlcmd ユティリティの指定例を次に示します。

```
sqlcmd -U <接続ユーザ名2>
       -P <パスワード>
       -i <再作成用のSQLスクリプトファイルの絶対パス>
```

(4)　すべてのデータのインポート

すべてのデータをインポートするには，テーブルごとにbcp コマンドを実行します。bcp コマンドについては，SQL Server のマニュアルを参照してください。

bcp コマンドの指定例を次に示します。

```
bcp 〈テーブル名〉 in 〈データファイルの絶対パス〉
       -U 〈ユーザ名〉
       -P 〈パスワード〉
       -c
```

(5)　DBMS への接続先情報の変更

ユーザまたはスキーマを変更した場合は，変更後のユーザまたはスキーマで接続できるユーザおよびパスワードに変更する必要があります。

変更方法については，「6.1.2　接続先 DBMS の変更」を参照してください。

6.2　実行環境の設定情報の変更

この節では，CSCIW が動作している実行環境で，設定情報を変更する場合の手順について説明します。

6.2.1　CSCIWManagementServer および CSCIW を使用している Java アプリケーションの再起動が必要なコマンド

CSCIW が動作している実行環境で，次のコマンドを実行して設定情報を変更した場合，CSCIWManagementServer および CSCIW を使用している Java アプリケーションの再起動が必要です。再起動しないと，変更後の設定情報は反映されません。

表 6-1　CSCIWManagementServer の再起動が必要なコマンド

コマンド名	CSCIWManagementServer の再起動後に反される内容
ciwchgenv（システム共通環境情報の変更）	ciwchgenv コマンドで変更したシステム共通環境情報の内容
ciweditbp（ビジネスプロセス定義の変更）	ciweditbp コマンドで変更したビジネスプロセス定義の内容
ciwmngbp（ビジネスプロセス定義の管理）※	ciwmngbp コマンドで状態変更，または削除したビジネスプロセス定義の内容
ciwmngcr（振り分けルール定義の管理）	ciwmngcr コマンドで追加，状態変更，または削除した振り分けルール定義の内容
ciwsetenv（環境の構築または削除）	ciwsetenv コマンドで追加，削除，または変更したシステム ID の内容
ciwchgdef（条件・ルール・作業アプリケーションの変更）	ciwchgdef コマンドで変更した条件・ルール・作業アプリケーションの定義の内容

注※

ciwmngbp コマンドでビジネスプロセス定義を追加する場合は，バージョンアップによる追加となるため，案件生成時にバージョンを指定しなければ，追加された最新のバージョンが自動的に反映されます。そのため，CSCIWManagementServer の再起動は必須ではありません。

CSCIWManagementServer を再起動する手順を次に示します。

1. CSCIWManagementServer を停止する

   cjstopapp コマンドを使用して，CSCIWManagementServer を停止します。

   停止方法については，マニュアル「Cosminexus 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」の「J2EE アプリケーションの停止」の説明を参照してください。

2. Java アプリケーションを停止する

   CSCIW を使用している Java アプリケーションを停止します。

3. CSCIWManagementServer を再起動する

   cjstartapp コマンドを使用して，1.で停止した CSCIWManagementServer を開始します。

   開始方法については，マニュアル「Cosminexus 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」の「J2EE アプリケーションの開始」の説明を参照してください。

4. Java アプリケーションを開始する

CSCIW を使用している Java アプリケーションを開始します。

6.2.2 環境変数の変更

環境変数の変更について説明します。

CSCIW が動作している実行環境で，次の環境変数を変更する場合，Java アプリケーションおよび J2EE サーバの再起動が必要です。再起動しないと，変更後の設定情報は反映されません。

- CSCIW_HOME 環境変数
- CSCIW_CONF_DIR 環境変数（マルチインスタンス構成の場合）

設定情報を変更する手順を次に示します。

1. 設定情報を変更する

   CSCIW_HOME 環境変数，または CSCIW_CONF_DIR 環境変数の設定情報を変更します。

2. Java アプリケーションおよび J2EE サーバを停止する

   CSCIW を使用している Java アプリケーションを停止します。

   また，`cjstopsv` コマンドを使用して，J2EE サーバを停止します。

   J2EE サーバの停止方法については，マニュアル「Cosminexus アプリケーションサーバ リファレンス コマンド編」を参照してください。

3. Java アプリケーションおよび J2EE サーバを再起動する

   2.で停止させた Java アプリケーションを開始します。

   また，`cjstartsv` コマンドを使用して，J2EE サーバを開始します。

   J2EE サーバの開始方法については，マニュアル「Cosminexus アプリケーションサーバ リファレンス コマンド編」を参照してください。

6.2.3 システム設定プロパティファイルおよびセットアッププロパティファイルの変更

システム設定プロパティファイルおよびセットアッププロパティファイルの変更について説明します。

CSCIW が動作している実行環境で，次の操作をした場合，CSCIWManagementServer および CSCIW を使用している Java アプリケーションの再起動が必要になります。再起動しないと，変更後の設定情報は反映されません。

- システム設定プロパティファイルへの情報の追加または削除した場合
- セットアッププロパティファイルの設定情報を変更した場合

設定情報を変更する手順を次に示します。

1. 設定情報を変更する

   システム設定プロパティファイル，またはセットアッププロパティファイルの設定情報を変更します。

2. CSCIWManagementServer を停止する

   cjstopapp コマンドを使用して，CSCIWManagementServer を停止します。

   停止方法については，マニュアル「Cosminexus 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」の「J2EE アプリケーションの停止」の説明を参照してください。

3. Java アプリケーションを停止する

   CSCIW を使用している Java アプリケーションを停止します。

4. CSCIWManagementServer を再起動する

   cjstartapp コマンドを使用して，2.で停止した CSCIWManagementServer を開始します。

   開始方法については，マニュアル「Cosminexus 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」の「J2EE アプリケーションの開始」の説明を参照してください。

5. Java アプリケーションを開始する

   CSCIW を使用している Java アプリケーションを開始します。

6.2.4 CSCIWManagementServer が使用するデータソース表示名の変更

CSCIWManagementServer が使用するデータソース表示名には，データベースと接続するための DB Connector のプロパティに設定した DB Connector のリソースアダプタの表示名と同じ名前を指定します。

1. cjimportapp コマンドを使用して，CSCIWManagementServer をインポートする

   CSCIWManagementServer を組み込みます。インポート済みの場合は，この手順は不要です。

2. cjgetappprop コマンドを使用して，XML ファイル（属性ファイル）を取得する

   cjgetappprop コマンドの実行例

   ```
   cjgetappprop <サーバ名称> -name CSCIWManagementServer -type all -c <属性ファイルパス>
   ```

   cjgetappprop コマンドの詳細については，マニュアル「Cosminexus アプリケーションサーバ　リファレンス コマンド編」を参照してください。

3. 取得した XML ファイル（属性ファイル）を編集する

   取得した XML ファイル（属性ファイル）をテキストエディタで開き，resource-ref 要素の linked-to 要素を「DB_Connector_for_CSCIW」から「4.5.3(4)　DB Connector の設定」で設定したリソースアダプタの表示名に変更します。

   ```
   <resource-ref>
     <description xml:lang="en"></description>
     <res-ref-name>jdbc/CSCIWDataSource</res-ref-name>
   ```

```
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
  <res-sharing-scope>Shareable</res-sharing-scope>
  <linked-to>DB_Connector_for_CSCIW</linked-to>
 </resource-ref>
```

4. cjsetappprop コマンドを使用して，編集した XML ファイル（属性ファイル）の情報をプロパティに設定する

   cjsetappprop コマンドの実行例

```
cjsetappprop <サーバ名称> -name CSCIWManagementServer -type all -c <属性ファイルパス>
```

   cjsetappprop コマンドの詳細については，マニュアル「Cosminexus アプリケーションサーバ　リファレンス コマンド編」を参照してください。

5. cjstartapp コマンドを使用して，CSCIWManagementServer を開始する

6.2.5　案件運用操作が使用するデータソース表示名の変更

案件運用操作が使用するデータソース表示名には，データベースと接続するための DB Connector のプロパティに設定した DB Connector のリソースアダプタの表示名と同じ名前を指定します。

1. cjimportapp コマンドを使用して，案件運用操作をインポートする

   案件運用操作を組み込みます。インポート済みの場合は，この手順は不要です。

2. cjgetappprop コマンドを使用して，XML ファイル（属性ファイル）を取得する

   cjgetappprop コマンドの実行例

```
cjgetappprop <サーバ名称> -name CSCIWAdminServlet -type all -c <属性ファイルパス>
```

   cjgetappropコマンドの詳細については，マニュアル「Cosminexus アプリケーションサーバ　リファレンス コマンド編」を参照してください。

3. 取得した XML ファイル（属性ファイル）を編集する

   取得した XML ファイル（属性ファイル）をテキストエディタで開き，2 か所のresource-ref 要素の linked-to 要素を「DB_Connector_for_CSCIW」から「4.5.3(4)　DB Connector の設定」で設定したリソースアダプタの表示名に変更します。

```
 <resource-ref>
   <description xml:lang="en"></description>
   <res-ref-name>jdbc/CSCAdminIWDataSource</res-ref-name>
   <res-type>javax.sql.DataSource</res-type>
   <res-auth>Container</res-auth>
   <res-sharing-scope>Shareable</res-sharing-scope>
   <linked-to>DB_Connector_for_CSCIW</linked-to>
 </resource-ref>
…中略…
 <resource-ref>
   <description xml:lang="en"></description>
```

```
  <res-ref-name>jdbc/CSCAdminIWDataSource</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
  <res-sharing-scope>Shareable</res-sharing-scope>
  <linked-to>DB_Connector_for_CSCIW</linked-to>
</resource-ref>
```

4. cjsetappprop コマンドを使用して，編集した XML ファイル（属性ファイル）の情報をプロパティに設定する

   cjsetappprop コマンドの実行例

```
cjsetappprop <サーバ名称> -name CSCIWAdminServlet -type all -c <属性ファイルパス>
```

   cjsetappprop コマンドの詳細については，マニュアル「Cosminexus アプリケーションサーバ　リファレンス コマンド編」を参照してください。

5. cjstartapp コマンドを使用して，案件運用操作を開始する

6.2.6　アプリケーション呼び出しサービスが使用するデータソース表示名の変更

アプリケーション呼び出しサービスが使用するデータソース表示名には，データベースと接続するための DB Connector のプロパティに設定した DB Connector のリソースアダプタの表示名と同じ名前を指定します。

1. cjimportappコマンドを使用して，アプリケーション呼び出しサービスをインポートする

   アプリケーション呼び出しサービスを組み込みます。インポート済みの場合は，この手順は不要です。

2. cjgetappprop コマンドを使用して，XML ファイル（属性ファイル）を取得する

   cjgetappprop コマンドの実行例

```
cjgetappprop <サーバ名称> -name CSCIWBpmnAPService01 -type all -c <属性ファイルパス>
```

   cjgetappropコマンドの詳細については，マニュアル「Cosminexus アプリケーションサーバ　リファレンス コマンド編」を参照してください。

3. 取得した XML ファイル（属性ファイル）を編集する

   取得した XML ファイル（属性ファイル）をテキストエディタで開き，resource-ref 要素のlinked-to 要素を「DB_Connector_for_CSCIW」から「4.5.3(4)　DB Connector の設定」で設定したリソースアダプタの表示名に変更します。

```
<resource-ref>
   <description xml:lang="en"></description>
   <res-ref-name>jdbc/CSCIWApServiceDataSource</res-ref-name>
   <res-type>javax.sql.DataSource</res-type>
   <res-auth>Container</res-auth>
   <res-sharing-scope>Shareable</res-sharing-scope>
   <injection-target>
```

```
        <injection-target-class>jp.co.Hitachi.soft.csciw.bpmn.apservice.APServiceTimer</i
njection-target-class>
        <injection-target-name>mDataSource</injection-target-name>
    </injection-target>
    <linked-to>DB_Connector_for_CSCIW</linked-to>
 </resource-ref>
```

4. cjsetappprop コマンドを使用して，編集した XML ファイル（属性ファイル）の情報をプロパティに設定する

   cjsetappprop コマンドの実行例

```
cjsetappprop <サーバ名称> -name CSCIWBpmnAPService01 -type all -c <属性ファイルパス>
```

   cjsetappprop コマンドの詳細については，マニュアル「Cosminexus アプリケーションサーバ　リファレンス コマンド編」を参照してください。

5. cjstartapp コマンドを使用して，アプリケーション呼び出しサービスを開始する

6.2.7　Java オブジェクト呼び出しで呼び出す Java オブジェクトの変更

Java オブジェクト呼び出しで呼び出す Java オブジェクトの変更について説明します。

Java オブジェクト呼び出しで呼び出す Java オブジェクトの変更は，Java オブジェクトを J2EE アプリケーションに組み込んでいる場合と，コンテナ拡張ライブラリに組み込んでいる場合で手順が異なります。それぞれの手順を次に示します。

(1)　Java オブジェクトを J2EE アプリケーションに組み込みこんでいる場合

1. J2EE アプリケーションを停止する

   CSCIW を使用している J2EE アプリケーションを停止します。

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

2. Java オブジェクトを入れ替える

   J2EE アプリケーションに組み込まれている Java オブジェクトのライブラリ JAR を削除したあと，J2EE アプリケーションに新しい Java オブジェクトのライブラリ JAR を追加します。

   ライブラリ JAR の削除および追加方法については，マニュアル「Cosminexus 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」を参照してください。

3. J2EE アプリケーションを開始する

   1.で停止させた J2EE アプリケーションを開始します。

   J2EE アプリケーションの開始方法については，マニュアル「Cosminexus 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」の「J2EE アプリケーションの開始」の説明を参照してください。

これらの作業を CSCIW のすべての J2EE アプリケーション（案件運用操作を含む）で行ってください。

(2) Java オブジェクトをコンテナ拡張ライブラリに組み込んでいる場合

1. J2EE サーバを停止する

   CSCIW を使用している J2EE サーバを停止します。

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

2. Java オブジェクトを入れ替える

   コンテナ拡張ライブラリに指定されている Java オブジェクトのライブラリ JAR を新しい Java オブジェクトのライブラリ JAR に変更します。

3. J2EE サーバを開始する

   1.で停止させた J2EE サーバを開始します。

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

6.2.8 アプリケーション呼び出しサービスの設定の変更

アプリケーション呼び出しサービスに関する設定の変更手順について説明します。

(1) ファイルに格納した設定情報を変更する

ファイルに格納した設定情報を次の手順で変更します。

1. ファイルの内容を変更する。

   目的に応じて，必要なファイルだけ内容を変更してください。

   - 共通設定ファイル
   - アプリケーション呼び出し情報ファイル
   - REST アプリケーション呼び出し用ヘッダファイル
   - REST アプリケーション呼び出しスキーマ変換用スタイルシート

2. 変更したファイルに応じて，必要な作業を実施する。

   - 共通設定ファイルの内容を変更した場合

     変更方法はパラメタのカテゴリごとに異なります。各パラメタがどのカテゴリに属するかについては，「付録 H.11 共通設定ファイル」を参照してください。

     カテゴリごとに必要な作業を次に示します。

     パラメタのカテゴリが「共通」のとき

     J2EE サーバを再起動してください。再起動後にアプリケーション呼び出しサービスを開始した際に，変更したファイルの内容が適用されます。

パラメタのカテゴリが「アプリケーション呼び出しサービス」のとき

アプリケーション呼び出しサービスを再開始してください。

また，同一の J2EE サーバに複数のアプリケーション呼び出しサービスを配備している場合，すべてのアプリケーション呼び出しサービスを再開始してください。変更したファイルの内容が適用されます。

- アプリケーション呼び出し情報ファイル，REST アプリケーション呼び出し用ヘッダファイル，または REST アプリケーション呼び出しスキーマ変換用スタイルシートの内容を変更した場合

  アプリケーション呼び出しサービスを再開始してください。

  また，同一の J2EE サーバに複数のアプリケーション呼び出しサービスを配備している場合，すべてのアプリケーション呼び出しサービスを再開始してください。変更したファイルの内容が適用されます。

(2) アプリケーション呼び出しに関する設定情報を変更する

ref 識別子ごとのアプリケーション呼び出し制御情報は，ciwmngap コマンドで変更できます。また，アプリケーション呼び出しグループは，ciwmngapgrp コマンドで変更できます。

各コマンドの詳細については，マニュアル「uCosminexus Service Coordinator Interactive Workflow コマンド」を参照してください。

(3) WorkManager の最大スレッド数を変更する

WorkManager の最大スレッド数を次の手順で変更します。

1. J2EE サーバ用ユーザプロパティファイル（usrconf.properties）に，WorkManager の最大スレッド数を指定する。

   WorkManager の最大スレッド数の指定例を次に示します。

```
ejbserver.commonj.WorkManager.non_daemon_work_threads=20
```

2. 変更対象の J2EE サーバを再起動する。

   WorkManager の最大スレッド数に，設定した値が適用されます。

重要

WorkManager の最大スレッド数は，スレッド数の見積もりおよびデータベースコネクション数の見積もりに影響があります。そのため，設定値を変更する場合は，スレッド数やデータベースコネクション数の再見積もりが必要です。

6.3　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

この節では，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について説明します。

6.3.1　変更後のビジネスプロセス定義および振り分けルール定義の情報反映

CSCIW が動作している環境で次の操作をした場合，CSCIWManagementServer および CSCIW を使用している Java アプリケーションの再起動が必要になります。再起動しないと，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後の情報は反映されません。

- ビジネスプロセス定義の追加
- ビジネスプロセス定義の削除
- ビジネスプロセス定義の属性の変更
- 振り分けルール定義の追加
- 振り分けルール定義の削除
- 振り分けルール定義の属性の変更

ビジネスプロセス定義または振り分けルール定義を変更した場合は，CSCIWManagementServer の再起動をすべての J2EE サーバで実施してください。また，CSCIW を使用している Java アプリケーションについては，変更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使用している Java アプリケーションをすべて再起動する必要があります。

CSCIWManagementServer を再起動する手順を次に示します。

1. CSCIWManagementServer を停止する

   `cjstopapp` コマンドを使用して，CSCIWManagementServer を停止します。

   停止方法については，マニュアル「Cosminexus 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」の「J2EE アプリケーションの停止」の説明を参照してください。

2. Java アプリケーションを停止する

   CSCIW を使用している Java アプリケーションを停止します。

3. CSCIWManagementServer を再起動する

   `cjstartapp` コマンドを使用して，1.で停止した CSCIWManagementServer を開始します。

   開始方法については，マニュアル「Cosminexus 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」の「J2EE アプリケーションの開始」の説明を参照してください。

4. Java アプリケーションを開始する

   CSCIW を使用している Java アプリケーションを開始します。

6.3.2　ビジネスプロセス定義削除時の注意事項

ビジネスプロセス定義を削除する際に，CSCIWManagementServer を再起動しないと，CSCIWManagementServer には削除対象の古いビジネスプロセス定義がキャッシュされた状態になっていることがあります。この状態で，該当のビジネスプロセス定義に対して，バージョンを指定して案件を投入すると，DBMS からは不正な案件（削除されているビジネスプロセス定義に対する案件）として投入されてしまいます。

また，ビジネスプロセス定義を削除する際に，API やciwdelpi コマンドで対象となるビジネスプロセス定義の案件を削除している途中で，そのあとからロールバックされて案件の削除が取り消された場合も，同様に不正な案件として残ってしまいます。

このように，不正になってしまった案件については，SQL で検索できるので，不正な案件 ID の一覧を作成したあと，ciwdelpi コマンドの-pidfile オプションを指定して削除してください。

SQL 文の指定例を次に示します。

```
select ID from <SYSTEMID>_PROCESS_INSTANCE
          where ProcessDefinitionID
             not in (select ID from <SYSTEMID>_PROCESS_DEF);
```

注

SYSTEMID には，ワーク管理データベースを一意に識別するためのシステム ID に置き換えてください。

7 ワーク管理システムの運用

この章では，ワーク管理システムの運用方法について説明します。

7.1　業務の開始と停止

この節では，業務の開始と停止について説明します。

> **ヒント**
>
> BPMN 連携機能を使用する場合の業務の開始と停止の詳細については，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。

7.1.1　業務の開始

業務を開始する方法を説明します。

業務の開始は次の手順で実施します。

1. CSCIWManagementServer を開始します。
   アプリケーション名：CSCIWManagementServer
2. アプリケーション呼び出しサービスから呼び出される REST アプリケーションを開始します。
   アプリケーション呼び出しサービスを使用する場合だけ実施してください。
3. アプリケーション呼び出しサービスを開始します。
   アプリケーション呼び出しサービスを使用する場合だけ実施してください。
   アプリケーション名：CSCIWBpmnAPService01
4. REST API または Java API を実行する業務アプリケーションを開始します。

各アプリケーションを(1)または(2)の手順で開始します。

(1)　運用管理ポータルを使用する場合

次の手順で操作してください。

1. 運用管理ポータルにログインする
2. 運用管理ポータルで［論理サーバのアプリケーション管理］アンカーをクリックする
3. ツリーペインで次の順にクリックする
   ［論理 J2EE サーバ］－［J2EE サーバ］－［<サーバ名称>］－［アプリケーション］
4. 右ペインの［開始／停止］タブをクリックし，［J2EE アプリケーションの開始／停止］画面を表示する
5. 対象アプリケーションの［開始］アンカーをクリックする

6. 内容を確認して，［はい］ボタンをクリックする

(2) サーバ管理コマンドを使用する場合

cjstartapp コマンドを使用して，各アプリケーションを開始します。

開始方法については，マニュアル「Cosminexus 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」の「J2EE アプリケーションの開始」の説明を参照してください。

cjstartapp コマンドの実行例を次に示します。

```
cjstartapp 〈サーバ名称〉 -name 〈アプリケーション名〉
```

7.1.2 業務の停止

業務を停止する方法を説明します。

業務の停止は，次の手順で実施します。

1. REST API または Java API を実行する業務アプリケーションを停止します。
2. アプリケーション呼び出しサービスを停止します。
   アプリケーション呼び出しサービスを使用する場合だけ実施してください。
   対象アプリケーション：CSCIWBpmnAPService01
3. アプリケーション呼び出しサービスから呼び出される REST アプリケーションを停止します。
   アプリケーション呼び出しサービスを使用する場合だけ実施してください。
4. CSCIWManagementServer を停止します。
   対象アプリケーション：CSCIWManagementServer

各アプリケーションを(1)または(2)の手順で停止します。

(1) 運用管理ポータルを使用する場合

次の手順で操作してください。

1. 運用管理ポータルにログインする
2. 運用管理ポータルで［論理サーバのアプリケーション管理］アンカーをクリックする
3. ツリーペインで次の順にクリックする
   ［論理 J2EE サーバ］－［J2EE サーバ］－［<サーバ名称>］－［アプリケーション］
4. 右ペインの［開始／停止］タブをクリックし，［J2EE アプリケーションの開始／停止］画面を表示する

5. 対象アプリケーションの［停止］アンカーをクリックする

6. 内容を確認して，［はい］ボタンをクリックする

(2) サーバ管理コマンドを使用する場合

cjstopapp コマンドを使用して，各アプリケーションを停止します。

停止方法については，マニュアル「Cosminexus 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」の「J2EE アプリケーションの停止」の説明を参照してください。

cjstopapp コマンドの実行例を次に示します。

```
cjstopapp 〈サーバ名称〉 -name 〈アプリケーション名〉
```

7.2　ビジネスプロセス定義に関する運用

この節では，CSCIW の管理者またはビジネスプロセスの管理者が日常的に実施するビジネスプロセス定義の運用操作について説明します。また，運用操作の際の前提知識として，CSCIW 内でビジネスプロセス定義がどのように管理されるかについても説明します。

ビジネスプロセス定義を実際に運用するためには，CSCIW へ登録する必要があります。CSCIW へ登録することで，ビジネスプロセス定義をワーク管理システムで一元的に管理できるようになります。

> **ヒント**
>
> BPMN 連携機能を使用する場合のビジネスプロセス定義に関する運用の詳細については，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。

7.2.1　ビジネスプロセス定義のバージョン

ビジネスプロセス定義は，同一名称で複数の**バージョン**を持つことができます。CSCIW 内では，「ビジネスプロセス定義名+バージョン番号」としてビジネスプロセス定義が管理されます。つまり，同じ名称のビジネスプロセス定義であっても，バージョンが異なれば別のビジネスプロセス定義として扱われます。

ビジネスプロセス定義のバージョンは，ビジネスプロセス定義の作成時にユーザが任意の番号で指定します。

7.2.2　ビジネスプロセス定義の状態と案件投入可能期間

CSCIW 内の各ビジネスプロセス定義には，次の項目を設定できます。

●状態

ビジネスプロセス定義の状態です。状態には「活性」と「非活性」があり，案件を投入できるのは，「活性」状態のビジネスプロセス定義だけです。

なお，ビジネスプロセス定義の状態変更は，運用中の案件に対しては影響しません。ビジネスプロセス定義の状態を「活性」から「非活性」に変更しても，そのビジネスプロセス定義ですでに運用されている案件は，そのまま実行できます。

●案件投入可能期間

ビジネスプロセス定義に案件を投入できる期間です。

なお，ビジネスプロセス定義の案件投入可能期間は，運用中の案件に対しては影響しません。ビジネスプロセス定義の案件投入可能期間が過ぎた場合でも，そのビジネスプロセス定義ですでに運用されている案件は，そのまま実行できます。

ビジネスプロセス定義の状態および案件投入可能期間と，案件投入との関係を次の表に示します。

表 7-1　ビジネスプロセス定義の状態および案件投入可能期間と，案件投入との関係

状態	案件投入可能期間	案件投入の可否
活性	案件投入可能期間内	可
	案件投入可能期間外	不可
非活性	案件投入可能期間内	不可
	案件投入可能期間外	不可

7.2.3　ビジネスプロセス定義の登録

ビジネスプロセス定義の登録は，ビジネスプロセス管理ツール（CSCIW-Definer），またはコマンドで実行できます。

ビジネスプロセス管理ツール（CSCIW-Definer）によるビジネスプロセス定義の登録については，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。

ここでは，ciwmngbp（ビジネスプロセス定義の管理）コマンドを使用したビジネスプロセス定義の登録について説明します。

次に示す形式でコマンドを実行すると，ビジネスプロセス定義は，CSCIW で実行できる形式に変換され，ワーク管理データベースに登録されます。

```
ciwmngbp -sid 〈システムID〉 -reg
         -bpf 〈ビジネスプロセス定義ファイル名※〉
```

注※

ビジネスプロセス管理ツール（CSCIW-Definer）で作成したビジネスプロセス定義ファイル名を指定してください。

ビジネスプロセス定義をワーク管理データベースに登録する際，「ビジネスプロセス定義名+バージョン」ごとに一意となる識別子が付けられます。登録したビジネスプロセス定義は，この識別子で管理されます。

ワーク管理データベースに登録されているビジネスプロセス定義の一覧表示，登録後のビジネスプロセス定義の属性変更，および取得についても，ciwmngbp（ビジネスプロセス定義の管理）コマンドで実施できます。

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。

7.2.4　ビジネスプロセス定義の削除

ビジネスプロセス定義の削除は，ビジネスプロセス管理ツール（CSCIW-Definer），またはコマンドで実行できます。

ビジネスプロセス管理ツール（CSCIW-Definer）によるビジネスプロセス定義の削除については，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。

ここでは，ciwmngbp（ビジネスプロセス定義の管理）コマンドを使用したビジネスプロセス定義の削除について説明します。

次に示す形式でコマンドを実行すると，削除要求のあったビジネスプロセス定義が，ワーク管理データベースから削除されます。

```
ciwmngbp -sid 〈システムID〉 -del
         -bpn 〈ビジネスプロセス定義名〉
         -bpv 〈ビジネスプロセス定義バージョン〉
```

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。

重要

ビジネスプロセス定義を削除するときに，次の条件を満たしていなければ，コマンドを実行してもビジネスプロセス定義は削除できません。

- 対象となるビジネスプロセス定義が「非活性」状態であること
- 対象となるビジネスプロセス定義内に案件が存在しないこと

7.3　振り分けルール定義に関する運用

この節では，CSCIW の管理者またはビジネスプロセスの管理者が日常的に実施する振り分けルール定義の運用操作について説明します。また，運用操作の際の前提知識として，CSCIW 内で振り分けルール定義がどのように管理されるかについても説明します。

振り分けルール定義を実際に運用するためには，CSCIW へ登録する必要があります。CSCIW へ登録することで，振り分けルール定義をワーク管理システムで一元的に管理できるようになります。

7.3.1　振り分けルール定義の管理

(1)　振り分けルール定義のバージョン指定

振り分けルール定義のバージョンは，振り分けルール定義の作成時にユーザが任意の番号で指定します。

振り分けルール定義は，同一名称で複数の**バージョン**を持つことができます。CSCIW 内では，「振り分けルール定義名+バージョン番号」として振り分けルール定義が管理されます。つまり，同じ名称の振り分けルール定義であっても，バージョンが異なれば別の振り分けルール定義として扱われます。

(2)　複数バージョンを持つ振り分けルール定義の特定

案件実行時に呼び出される振り分けルール定義に複数のバージョンがある場合，CSCIW は次のように振り分けルール定義を特定します。

1. 指定された名称の振り分けルール定義について，バージョンごとに状態をチェックし，「活性」状態のものを選択します。
2. 1.で選択した結果から，有効期限内のものを選択します。
3. 2.で選択した結果が複数ある場合は，バージョン番号が最大（最新）のものを選択します。

7.3.2　振り分けルール定義の状態と有効期間

CSCIW 内の各振り分けルール定義には，次の項目を設定できます。

●状態

振り分けルール定義の状態です。状態には，「活性」と「非活性」があり，案件実行時に適用されるのは，「活性」状態の振り分けルール定義だけです。

案件実行時に「非活性」状態の振り分けルール定義を適用した場合は，エラーとなります。

●有効期間（適用可能期間）

振り分けルール定義を適用できる期間です。

案件実行時に有効期間外の振り分けルール定義を適用した場合は，エラーとなります。

振り分けルール定義の状態および有効期間と，案件に対する適用の関係を次の表に示します。

表 7-2　振り分けルール定義の状態および有効期間と，案件に対する適用の関係

状態	有効期間	案件に対する適用の可否
活性	有効期間内	可
	有効期間外	不可
非活性	有効期間内	不可
	有効期間外	不可

7.3.3　振り分けルール定義の登録

振り分けルール定義の登録は，ビジネスプロセス管理ツール（CSCIW-Definer），またはコマンドで実行できます。

ビジネスプロセス管理ツール（CSCIW-Definer）による振り分けルール定義の登録については，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。

ここでは，ciwmngcr（振り分けルール定義の管理）コマンドを使用した振り分けルール定義の登録について説明します。

次に示す形式でコマンドを実行すると，振り分けルール定義は，CSCIW で実行できる形式に変換され，ワーク管理データベースに登録されます。

```
ciwmngcr -sid 〈システムID〉 -reg
         -crf 〈振り分けルール定義ファイル名※〉
```

注※

ビジネスプロセス管理ツール（CSCIW-Definer）で作成した振り分けルール定義ファイル名を指定してください。

振り分けルール定義をワーク管理データベースに登録する際，「振り分けルール定義名+バージョン」ごとに一意となる識別子が付けられます。登録した振り分けルール定義は，この識別子で管理されます。

ワーク管理データベースに登録されている振り分けルール定義の一覧表示，登録後の振り分けルール定義の属性変更，および取得についても，ciwmngcr（振り分けルール定義の管理）コマンドで実施できます。

ciwmngcr（振り分けルール定義の管理）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。

7.3.4　振り分けルール定義の削除

振り分けルール定義の削除は，ビジネスプロセス管理ツール（CSCIW-Definer），またはコマンドで実行できます。

ビジネスプロセス管理ツール（CSCIW-Definer）による振り分けルール定義の削除については，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。

ここでは，ciwmngcr（振り分けルール定義の管理）コマンドを使用した振り分けルール定義の削除について説明します。

次に示す形式でコマンドを実行すると，削除要求のあった振り分けルール定義が，ワーク管理データベースから削除されます。

```
ciwmngcr -sid 〈システムID〉 -del
         -crn 〈振り分けルール定義名〉
         -crv 〈振り分けルール定義バージョン〉
```

ciwmngcr（振り分けルール定義の管理）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。

重要

振り分けルール定義を削除するときに，対象となる振り分けルール定義が非活性状態でなければ，コマンドを実行しても振り分けルール定義は削除できません。

7.4　案件に関する運用

この節では，CSCIW の管理者が日常的に実施する案件の運用について説明します。

CSCIW の管理者は，案件（業務ステップおよび作業を含む）に対して，中断，再開などの運用ができます。案件に対する主な運用は，CSCIW の案件運用操作，API，およびコマンドで実行できます。

7.4.1　案件に対する運用

案件に対する運用の種類と CSCIW が提供する機能でできる操作を次の表に示します。

表 7-3　案件に対する運用の種類と CSCIW が提供する機能でできる操作

案件に対する運用の種類	操作		
	案件運用操作	API	コマンド
開始	○	○	－
中断	○	○	－
再開	○	○	－
強制終了	○	○	－
削除	○	○	○
優先度の設定と更新	－	○	－
処理期限の設定	－	○	－

（凡例）

○：操作できます

－：操作できません

案件に対する運用について次に説明します。

(1)　案件の開始

ビジネスプロセス定義のインスタンスとして「未実行」状態で生成された案件を開始します。案件を開始すると，その案件は「実行中」の状態になります。案件が「実行中」の状態に遷移すると，ソースノードから到達できる「初期」状態の業務ステップが自動的に活性化されます。ソースノードから直接到達できる「初期」状態の業務ステップがない場合は，新規に業務ステップが生成されます。

また，ビジネスプロセス定義の中で，事前生成が指定された業務ステップがある場合は，この時点でその業務ステップを生成します。

(2) 案件の中断と再開

何らかの理由で案件を一時的に停止したい場合は，案件を中断します。

案件を中断すると，その案件は「中断（実行停止)」の状態になります。案件の状態が「中断（実行停止)」の場合，その案件の業務ステップは「実行中」状態への遷移が抑止されます。また，「中断（実行停止)」状態の案件がシンクノードに到達した場合，「完了」状態への遷移が抑止されます。

案件を中断した場合の，業務ステップの状態遷移を次の表に示します。

表 7-4 案件を中断した場合の業務ステップの状態遷移

案件中断時の業務ステップの状態	業務ステップの状態遷移
未実行（初期）	状態は変わりません。ただし，案件の状態が「中断（実行停止)」のときに活性化した業務ステップは，「中断（実行開始不可)」の状態に遷移します。
実行中	状態は変わりませんが，業務ステップは続行されます。
中断（実行停止)，完了，強制終了	状態は変わりません。

案件を再開すると，その案件の状態は「中断（実行停止)」から「実行中」になります。案件の中断によって，「中断（実行開始不可)」の状態に遷移した業務ステップは，「未実行（実行開始可能)」の状態になります。また，「中断（実行停止)」の状態でシンクノードに到達した案件は，「完了」状態へ遷移できるようになります。

(3) 案件の強制終了

何らかの理由で案件を強制的に終了したい場合は，案件を強制終了します。

案件を強制終了すると，その案件は「強制終了」の状態になります。また，その案件が含まれるビジネスプロセス定義内の業務ステップおよび作業のうち，終了していないものはすべて「強制終了」の状態に遷移します。

強制終了された案件は，再実行できません。

(4) 案件の削除

完了した案件（シンクノードに到達した案件)，強制終了した案件のうち，不要なものがある場合は，CSCIW の管理者が削除します。

案件を削除すると，その案件が保持する業務ステップおよび作業も削除されます。

また，完了した案件および強制終了した案件の削除は，`ciwdelpi`（案件の削除）コマンドでも実行できます。`ciwdelpi`（案件の削除）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。

(5) 案件の優先度の設定と更新

CSCIW では，案件，業務ステップまたは作業に対して，それぞれ 1 つの優先度を整数値で設定できます。優先度を設定するには，CSCIW の API を使用します。設定した優先度は，案件，業務ステップおよび作業の属性として，検索条件に使用できます。ただし，優先度の設定によって CSCIW が案件の推進を制御することはありません。

(a) 優先度の設定

優先度を利用する場合は，案件投入時に案件の属性として優先度を設定します。案件の優先度として設定された値は，その案件内の業務ステップの優先度として設定が引き継がれます。さらに，業務ステップの優先度として設定された値は，その業務ステップ内の作業の優先度として設定が引き継がれます。

(b) 優先度の更新

対象となる案件，業務ステップまたは作業が未終了状態の場合には，CSCIW の API を利用して，優先度の設定を個別に更新できます。ただし，この API を使用して案件の優先度を更新しても，その案件内にすでに生成された業務ステップの優先度は更新されません。同様に，この API を使用して業務ステップの優先度を更新しても，その業務ステップ内のすでに生成された作業の優先度は更新されません。CSCIW が提供する API については，マニュアル「uCosminexus Service Coordinator Interactive Workflow AP 開発ガイド」を参照してください。

(6) 案件の処理期限の設定

CSCIW では案件の開始時に，案件の開始日時を起算日として，ビジネスプロセス定義で指定された処理期限を絶対日時で設定します。また，API によってユーザが設定することもできます。設定した処理期限は，案件の検索条件として使用できます。なお，処理期限は単なる属性値の一つであり，この値によって，CSCIW が案件の推進を制御することはありません。

7.4.2 業務ステップに対する運用

業務ステップに対する運用の種類と CSCIW が提供する機能でできる操作を次の表に示します。

表 7-5 業務ステップに対する運用の種類と CSCIW が提供する機能でできる操作

業務ステップに対する運用の種類	操作		
	案件運用操作	API	コマンド
活性化	○	○	−
中断	○	○	−
再開	○	○	−
完了	○	○	−
強制終了	○	○	−

業務ステップに対する運用の種類	操作		
	案件運用操作	API	コマンド
優先度の設定と更新	－	○	－
処理期限の設定	－	○	－

（凡例）

○：操作できます

－：操作できません

業務ステップに対する運用について次に説明します。

(1) 業務ステップの活性化

「初期」状態の業務ステップを「実行可能状態」にするには，業務ステップを活性化します。

活性化した業務ステップを含む案件の状態が「実行中」の場合，業務ステップの状態は，「実行中」になります。それ以外の場合は，業務ステップの状態は「実行開始不可」になります。

(2) 業務ステップの中断と再開

何らかの理由で業務ステップを一時的に停止したい場合は，業務ステップを中断します。

業務ステップを中断すると，その業務ステップは「中断（実行停止）」の状態になり，「完了」状態への遷移が抑止されます。また，その業務ステップ内の作業の発生が抑止されます。

業務ステップを中断した場合の，作業の状態遷移を次の表に示します。

表 7-6 業務ステップを中断した場合の作業の状態遷移

業務ステップ中断時の作業の状態	作業の状態遷移
未実行（初期）	状態は変わりません。
未実行（実行開始可能）	「中断（実行開始不可）」の状態に遷移します。
実行中	状態は変わりませんが，作業は続行されます。
中断（実行停止），完了，強制終了	状態は変わりません。

業務ステップを再開すると，その業務ステップの状態は「中断（実行停止）」から「実行中」になります。業務ステップの中断によって，「中断（実行開始不可）」の状態に遷移した作業は，「未実行（実行開始可能）」の状態になります。また，業務ステップの再開時には，業務ステップの完了条件および作業の発生条件が再度評価されます。

(3) 業務ステップの完了

「実行中」状態の業務ステップを完了して，次の業務ステップに遷移できるようにするには，業務ステップを完了します。

案件を完了すると，その業務ステップは「遷移可」状態になり，業務ステップに含まれるすべての「未実行」状態の作業が，「実行省略」状態になります。また，業務ステップに含まれるすべての「実行中」状態の作業が「実行取消」状態になります。さらに，業務ステップに含まれるすべての「実行停止」状態の作業が「強制終了」状態になります。

(4) 業務ステップの強制終了

何らかの理由で，業務ステップを強制的に終了したい場合は，業務ステップを強制終了します。

業務ステップを強制終了すると，その業務ステップは「強制終了」の状態になります。また，業務ステップが終了していないすべての作業が「強制終了」状態になります。

(5) 業務ステップの優先度の更新

対象となる業務ステップが未終了状態の場合には，CSCIW の API を利用して，優先度の設定を個別に更新できます。ただし，この API を使用して業務ステップの優先度を更新しても，その業務ステップ内のすでに生成された作業の優先度は更新されません。CSCIW が提供する API については，マニュアル「uCosminexus Service Coordinator Interactive Workflow AP 開発ガイド」を参照してください。

(6) 業務ステップの処理期限の設定

CSCIW では業務ステップの開始時に，案件の開始日時を起算日として，ビジネスプロセス定義で指定された処理期限を絶対日時で設定します。処理期限を設定するには，CSCIW の API を使用します。CSCIW が提供する API については，マニュアル「uCosminexus Service Coordinator Interactive Workflow AP 開発ガイド」を参照してください。

設定した処理期限は，業務ステップの検索条件として使用できます。なお，処理期限は単なる属性値の一つです。処理期限の設定によって，CSCIW が業務ステップの推進を制御することはありません。

7.4.3 作業に対する運用

作業に対する運用の種類と CSCIW が提供する機能でできる操作を次の表に示します。

表 7-7 作業に対する運用の種類と CSCIW が提供する機能でできる操作

作業に対する運用の種類	操作		
	案件運用操作	API	コマンド
準備済み	○	○	−
着手	○	○	−
中断	○	○	−
再開	○	○	−

作業に対する運用の種類	操作		
	案件運用操作	API	コマンド
完了通知	○	○	－
強制終了	○	○	－
優先度の設定と更新	－	○	－
処理期限の設定	－	○	－

（凡例）

○：操作できます

－：操作できません

作業に対する運用について次に説明します。

(1) 作業の準備と着手

「初期」状態の作業を開始するには，最初に作業を準備済みにします。作業を準備済みにすると，その作業は「実行開始可能」状態になります。「実行開始可能」状態の作業に対しては，評価が実施されます。作業が実行されないまま，作業の完了条件が満たされた場合は，「実行省略」の状態になります。

また，「実行開始可能」状態の作業に対して，ユーザ側で何らかの実行処理をする場合は，作業を着手します。作業を着手すると，その作業は，「作業者実行」状態になります。

(2) 作業の中断と再開

何らかの理由で作業を一時的に停止したい場合は，作業を中断します。

作業を中断すると，その作業は「中断（実行停止)」の状態になります。この状態では，作業の完了条件および組み込み作業実行時の評価が抑止されます。

作業を再開すると，その作業の状態は「中断（実行停止)」から「実行中」になります。また，作業の再開時には，作業の完了条件および組み込み作業が再度評価されます。

(3) 作業の完了通知

実行中の作業を任意の時点で完了したい場合は，作業の完了通知をします。作業の完了通知をすると，その作業は「完了」の状態になります。

(4) 作業の強制終了

何らかの理由で作業を強制的に終了したい場合は，作業を強制終了します。

作業を強制終了すると，その作業は「強制終了」の状態になります。

(5) 作業の優先度の更新

対象となる作業が未終了状態の場合には，CSCIW の API を利用して，優先度の設定を個別に更新できます。CSCIW が提供する API については，マニュアル「uCosminexus Service Coordinator Interactive Workflow AP 開発ガイド」を参照してください。

(6) 作業の処理期限の設定

CSCIW では作業の開始時に，その作業が属する業務ステップの処理期限をそのまま設定します。処理期限を設定するには，CSCIW の API を使用します。CSCIW が提供する API については，マニュアル「uCosminexus Service Coordinator Interactive Workflow AP 開発ガイド」を参照してください。

設定した処理期限は作業の検索条件として使用できます。なお，処理期限は単なる属性値の一つです。処理期限の設定によって，CSCIW が作業の推進を制御することはありません。

7.5　監査ログの取得によるシステム監視

この節では，監査ログの取得によるシステムの監視について説明します。

なお，CSCIW の監査ログの取得は，Cosminexus の監査ログを出力する機能を利用しています。Cosminexus の監査ログを出力する機能については，マニュアル「Cosminexus アプリケーションサーバ 機能解説 運用／監視／連携編」の「監査ログ出力機能」の説明を参照してください。

また，監査ログの出力の設定については，マニュアル「Cosminexus アプリケーションサーバ 機能解説 運用／監視／連携編」の「監査ログ出力の設定」の説明を，監査ログの出力先については，マニュアル「Cosminexus アプリケーションサーバ 機能解説 運用／監視／連携編」の「監査ログの出力先」の説明をそれぞれ参照してください。

7.5.1　ワーク管理システムの監査対象

監査ログが出力される契機になるのは，CSCIW のプログラムに対する操作です。

監査ログは，CSCIW の各プロセスで監査事象が発生したタイミングで出力されます。監査事象とは，CSCIWManagementServer へのログインおよびログアウト，コマンドの実行，CSCIW のプログラムに対して実行した操作，およびその操作に伴うプログラムの処理のうち，システムの構築や運用の正当性を証明するために記録する必要がある事象のことです。この操作は，システム管理者，システム運用者など，その作業に応じたユーザが実行します。

CSCIW で監査の対象になるのは，CSCIWManagementServer，コマンドおよび案件運用操作の 3 つです。それぞれの監査事象を次に示します。

(1)　監査対象が CSCIWManagementServer の場合の監査事象

CSCIWManagementServer の監査事象を次の表に示します。

表 7-8　CSCIWManagementServer の監査事象

監査事象	説明
StartStop	起動と停止※
Authentication	次に示す操作 • CSCIW-Definer からのログイン • CSCIW-Definer からのログアウト
ContentAccess	CSCIW-Definer からの次に示す操作 • ビジネスプロセス定義の登録 • ビジネスプロセス定義の情報取得 • ビジネスプロセス定義の属性変更 • ビジネスプロセス定義の削除

監査事象	説明
ContentAccess	• 振り分けルール定義の登録 • 振り分けルール定義の情報取得 • 振り分けルール定義の属性変更 • 振り分けルール定義の削除

注※

J2EE サーバが，J2EE アプリケーションの開始または停止の際に出力する監査ログで代用します。

(2) 監査対象がコマンドの場合の監査事象

コマンドの監査事象を次の表に示します。

表 7-9 コマンドの監査事象

コマンド名	監査事象	説明
ciwchgenv	ConfigurationAccess	コマンドの実行によるシステム共通環境情報の変更
ciwcleanup	ConfigurationAccess	コマンドの実行によるプロセス情報の正常化
ciwdelpi	ContentAccess	コマンドの実行による案件の削除
ciweditbp	ContentAccess	コマンドの実行によるビジネスプロセス定義の変更
ciwlistsid	ConfigurationAccess	コマンドの実行によるシステム ID 情報の表示
ciwmngbp	ContentAccess	コマンドの実行による次の操作 • ビジネスプロセス定義の登録 • ビジネスプロセス定義の情報取得 • ビジネスプロセス定義の属性変更 • ビジネスプロセス定義の削除
ciwmngcr	ContentAccess	コマンドの実行による次の操作 • 振り分けルール定義の登録 • 振り分けルール定義の情報取得 • 振り分けルール定義の属性変更 • 振り分けルール定義の削除
ciwreuseid	ConfigurationAccess	コマンドの実行によるシステム ID の再利用
ciwsetenv	ConfigurationAccess	コマンドの実行による次の操作 • 環境の構築 • 環境の削除
	Maintenance	コマンドの実行によるバージョンアップ
ciwchgdef	ContentAccess	コマンドの実行によるビジネスプロセス定義の変更
ciwmngap	−	−
ciwmngapgrp	−	−

コマンド名	監査事象	説明
ciwtransbpmn	−	−

（凡例）

−：監査事象はありません（監査ログを出力しません）。

(3)　監査対象が案件運用操作の場合の監査事象

案件運用操作の監査事象を次の表に示します。

表 7-10　案件運用操作の監査事象

監査事象	説明
StartStop	起動と停止※
Authentication	次に示す操作 • ログイン • ログアウト
ContentAccess	次に示す操作 • 案件の開始 • 案件の中断 • 案件の再開 • 案件の強制終了 • 案件の削除 • 業務ステップの活性化 • 業務ステップの中断 • 業務ステップの再開 • 業務ステップの完了 • 業務ステップの強制終了 • 作業の着手 • 作業の中断 • 作業の再開 • 作業の完了 • 作業の返却または準備済み • 作業の強制終了 • 作業の作業者再割り当て

注※

J2EE サーバが，J2EE アプリケーションの開始または停止の際に出力する監査ログで代用します。

7.5.2　監査ログの出力ポイントと出力項目

CSCIW に関係する監査ログの出力ポイントと出力項目について説明します。

(1) 出力ポイント

監査事象が発生した場合に，監査ログを出力するポイントを次に示します。

- 事象の発生時（result=Occurrence）
- 事象の終了時（result=Success または Failure）

(2) 出力項目

CSCIW に関係する監査ログの出力項目名と出力項目の意味を次に示します。

表 7-11 監査ログの出力項目名と出力項目の意味

出力項目名	出力項目の意味	説明
msgid	メッセージ ID	メッセージが出力されます。
compid	発生構成ソフトウェア名	固定文字列の「UAP_CSCIW」が出力されます。
ctgry	監査事象の種別	監査事象の種別が出力されます。 • StartStop サーバ，プロセス，サービスなどの起動・終了を示す事象です。 • Authentication ログイン，ログアウトなど，管理者 ID と管理者パスワードによる認証が実行されたことを示す事象です。 • ConfigurationAccess 設定および構成の変更，構成情報の参照を示す事象です。 • ContentAccess ユーザの重要なデータへのアクセスを試みたことを示す事象です。 • Maintenance 保守作業を実行して，操作が正常終了または失敗したことを示す事象です。
result	監査事象の結果	監査事象の結果（成功・失敗・発生）が出力されます。 • Success 監査事象の成功を示します。 • Failure 監査事象の失敗を示します。 • Occurrence 成功および失敗の区別がない事象の発生を示します。
subj:uid	サブジェクト識別情報	ユーザ ID が出力されます。 次に示す事象の場合に出力されます。 • CSCIWManagementServer の Authentication または ContentAccess • 案件運用操作の Authentication または ContentAccess
obj	オブジェクト情報	システム ID が，「"」で囲まれて出力されます。
op	動作情報	監査事象となった操作の種別が，「"」で囲まれて出力されます。 • Start

出力項目名	出力項目の意味	説明
op	動作情報	起動を示します。 • Stop 停止を示します。 • Login ログインを示します。 • Logout ログアウトを示します。 • Refer 設定情報または重要情報の参照を示します。 • Add 設定情報または重要情報の追加を示します。 • Update 設定情報または重要情報の更新を示します。 • Delete 設定情報または重要情報の削除を示します。
msg	自由記述	監査事情の内容を示す文章が出力されます。

7.6　バックアップとリストア

障害が発生し，環境を再構築する場合などに備えて，定期的にバックアップを取得することをお勧めします。

この節では，ワーク管理システムのバックアップおよびリストアの方法について説明します。

7.6.1　バックアップする内容

CSCIW の運用で，バックアップが必要な内容を次に示します。

(1)　環境設定ファイル

環境設定ファイルは，稼働中の CSCIW システムで使用するため，バックアップを取得してください。

(2)　アプリケーション呼び出しサービスの設定ファイル

アプリケーション呼び出しサービスの設定ファイルは，稼働中の CSCIW システムで使用するため，バックアップを取得してください。

アプリケーション呼び出しサービスの設定ファイルを次に示します。

- 共通設定ファイル
- アプリケーション呼び出し情報ファイル
- REST アプリケーション呼び出し用ヘッダファイル
- REST アプリケーション呼び出しスキーマ変換用スタイルシート

各ファイルの詳細については，「付録 H　BPMN 連携機能を使用しない場合のアプリケーション呼び出しサービス」を参照してください。

(3)　ワーク管理データベース，業務データベースおよび作業者データベース

ワーク管理データベース，業務データベースおよび作業者データベースのバックアップは，過去に運用していた（すでに削除されている）案件の情報を回復するなどの場合に利用します。

ワーク管理データベースには，CSCIW で運用されている次の情報が格納されています。

- ビジネスプロセス定義情報
- 振り分けルール定義情報
- 案件の情報
- 定義情報および案件情報に関連する情報
- CSCIW のシステム管理情報

ワーク管理データベースのバックアップは，業務データベースのバックアップと同期を取って実施する必要があります。

7.6.2　バックアップとリストアの流れ

バックアップとリストアの流れについて説明します。

(1)　バックアップの流れ

1. CSCIW を使用している Java アプリケーションおよび J2EE サーバを停止させてください。また，コマンドが実行されていないことを確認してください。
2. OS の機能を使用して，CSCIW の環境設定ファイルのバックアップを取得します。

   以前にバックアップを作成した時点から，環境設定ファイルの内容を変更している場合は，バックアップを必ず取得してください。

   また，環境を変更する前の状態に戻す場合は，変更前のバックアップが必要になります。必要に応じて環境を変更する前にもバックアップを取得してください。
3. アプリケーション呼び出しサービスを使用する場合，OS の機能を使用して，アプリケーション呼び出しサービスの設定ファイルのバックアップを取得します。

   以前にバックアップを作成した時点から，アプリケーション呼び出しサービスの設定ファイルの内容を変更している場合は，バックアップを必ず取得してください。

   また，環境を変更する前の状態に戻す場合は，変更前のバックアップが必要になります。必要に応じて環境を変更する前にもバックアップを取得してください。
4. DBMS の機能を使用して，ワーク管理データベースのバックアップを取得します。また，業務データベースのバックアップを取得します。

   このとき，データの整合性を保つために，ワーク管理データベース，および業務データベースのバックアップは同期を取る必要があります。
5. バックアップの取得完了後に，CSCIW を使用している Java アプリケーションおよび J2EE サーバを起動します。

(2)　リストアの流れ

1. CSCIW を使用している Java アプリケーションおよび J2EE サーバを停止させてください。また，コマンドが実行されていないことを確認してください。
2. バックアップから回復する前に，現在運用中の環境のバックアップを必要に応じて，取得しておきます。

   バックアップとして取得する内容は，現在運用中の環境設定ファイル，ワーク管理データベース，業務データベース，および作業者データベースです。
3. DBMS の機能を使用して，ワーク管理データベースのバックアップから回復します。また，業務データベースのバックアップから回復します。
4. OS の機能を使用して，取得済みの環境設定ファイルのバックアップから回復します。

以前にバックアップを作成した時点から，環境設定ファイルの内容を変更している場合，取得済みの環境設定ファイルのバックアップから必ず回復してください。

5. アプリケーション呼び出しサービスを使用する場合，OS の機能を使用して，取得済みのアプリケーション呼び出しサービスの設定ファイルのバックアップから回復します。

   以前にバックアップを作成した時点から，アプリケーション呼び出しサービスの設定ファイルの内容を変更している場合，取得済みのアプリケーション呼び出しサービスの設定ファイルのバックアップから必ず回復してください。

6. バックアップからの回復完了後に，CSCIW を使用している Java アプリケーションおよび J2EE サーバを起動します。

重要

CSCIW の運用中にワーク管理データベースのリストアを実行すると，ワーク管理データベースの内容に，回復できない不整合が発生します。データベースのリストアは，CSCIW を使用している Java アプリケーション，J2EE サーバ，およびコマンドが動作していないことを確認してから実行してください。

なお，CSCIW の運用中にワーク管理データベースのリストアを実行して不整合が発生した場合は，コマンドが動作していないことを確認したあと，CSCIW を使用している Java アプリケーションおよび J2EE サーバを停止させてから，リストアをし直してください。

(3) 運用中のバックアップについて

次に示す条件をすべて満たす場合は，CSCIW の運用中にバックアップを取得できます（ただし，バックアップを取得する際は，コマンドが実行されていないことを確認してください）。

- CSCIW の運用中にバックアップを取得できる機能が，ワーク管理データベースの DBMS に備わっている
- 業務プログラムの運用中にバックアップを取得できる機能が，業務データベースを運用しているシステムの機能に備わっている
- 業務データを運用しているシステムおよびワーク管理データベースに，双方のバックアップを一括して（同期を取って）取得できる機能が備わっている

上記の条件を満たす例として，業務データベース，作業者データベースおよびワーク管理データベースを同一の（運用中のバックアップが可能な）DBMS 上で運用し，これらのバックアップファイルを一括して取得するといった運用が考えられます。

運用中に取得したバックアップをリストアした場合，プロセスの起動情報が残るため，CSCIW を使用している Java アプリケーションおよび J2EE サーバを起動する前に`ciwcleanup`（プロセス情報の正常化）コマンドの実行が必要となる場合があります。`ciwcleanup`（プロセス情報の正常化）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。

7.7 データベースの再編成について

ワーク管理データベースとして使用している DBMS は，長期間の運用によって性能が劣化していきます。バックアップの取得とあわせて，定期的にデータベースの再編成を実施してください。

HiRDB を使用している場合，データベースの再編成方法については，マニュアル「HiRDB システム運用ガイド（UNIX(R)用）」または「HiRDB システム運用ガイド（Windows(R)用）」を参照してください。

ORACLE を使用している場合，データベースの再編成方法については，ORACLE のマニュアルを参照してください。

SQL Server を使用している場合，データベースの再編成方法については，SQL Server のマニュアルを参照してください。

重要

ORACLE のインデクスを再編成する際に，`REBUILD ONLINE` オプションを指定すると SELECT 文の結果が正しく得られないことがあります。そのため，CSCIW の API を使用している業務アプリケーションまたは CSCIW のコマンドが実行中に，`REBUILD ONLINE` オプションを指定したインデクスのオンライン再編成をしないでください。

7.8　業務システムの変更に伴う運用

業務システムの運用中には，組織改編によって業務を担当する部署が変更になる，業務プログラムの内容が変更になる，業務の手順が変更になるといったさまざまな変更が発生します。

ワーク管理システムはこのような変更が発生した場合でも，業務システム全体を停止して対処する必要はありません。変更内容に応じて，部分的な変更を加えることで対処できます。

この節では，業務システムの運用中に変更が発生した場合の，対処方法について説明します。

7.8.1　作業者を変更する場合

業務システムの運用中に人事異動や組織改編が発生した場合は，必要に応じて次の変更処理をします。

- ユーザおよびユーザグループ情報の変更
- 振り分けルール定義の変更
- 作業者の再割り当て

それぞれの変更処理について次に説明します。

(1)　ユーザおよびユーザグループに関する情報の変更

業務システムの運用中に人事異動や組織改編が発生した場合は，作業者データベースなどで管理されているユーザおよびユーザグループに関する情報を変更します。ユーザおよびユーザグループに関する情報は，作業者データベースなどの機能で変更します。

変更後の情報は，各作業に対して作業者が割り当てられる時点で自動的に反映されます。

(2)　振り分けルール定義の変更

人事異動や組織改編の程度によっては，ユーザおよびユーザグループに関する情報の変更に加え，振り分けルール定義の内容を変更する必要があります。具体例としては，組織変更によって，ある作業の担当部署が変更された場合などが考えられます。

振り分けルール定義の変更方法を次に示します。

1. 振り分けルール定義の変更

   ビジネスプロセス管理ツール（CSCIW-Definer）で，振り分けルール定義の内容を変更します。バージョン番号は変更前より大きな番号に変更してください。必要に応じて有効期間（開始日時）を指定します。

2. 変更後の振り分けルール定義の登録

変更後の振り分けルール定義は，ビジネスプロセス管理ツール（CSCIW-Definer），またはciwmngcr（振り分けルール定義の管理）コマンドを実行して，ワーク管理データベースに登録します。ビジネスプロセス管理ツール（CSCIW-Definer）による振り分けルール定義の変更および登録については，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。ciwmngcr（振り分けルール定義の管理）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。

3. 変更後の振り分けルール定義の運用開始

指定した有効期間（開始日時）になると，変更後の振り分けルール定義が適用できる状態になります。これ以降，変更後の振り分けルール定義によって作業者が割り当てられます。

> **メモ**
>
> ビジネスプロセス定義を変更しないで振り分けルール定義だけを変更する場合，案件の削除は不要です。

振り分けルール定義を変更して新しいバージョンで登録した場合，登録した直後から新しいバージョンの振り分けルール定義が使用されます。すでに開始している案件も新しい振り分けルール定義が使用されます。

新しい振り分けルール定義が有効になるのは，次の条件を満たしている場合です。

- 振り分けルール定義の運用状態が「活性」になっている。
- 振り分けルール定義が有効期間内である。
- ビジネスプロセス定義，または振り分けルール定義を登録したあと，CSCIWManagementServer を再起動している。

(3) 作業者の再割り当て

すでに作業者が割り当てられている作業に，ほかの作業者を割り当てたい場合は，次のどちらかの方法で変更します。

- 振り分けルールを再度適用し，（結果として）ほかの作業者に再割り当てする
- 指定した作業者に再割り当てする

作業者の再割り当てには，CSCIW が提供する API を使用します。CSCIW が提供する API については，マニュアル「uCosminexus Service Coordinator Interactive Workflow AP 開発ガイド」を参照してください。

7.8.2 データ条件を変更する場合

業務システムの運用中に，データ条件の変更が必要になる場合があります。例として，参照する業務データのデータ項目名が変更された場合，条件式の比較値を変更したい場合などが考えられます。

データ条件は，変更を反映する案件の範囲に応じて次の方法で変更します。

データ条件の変更は，運用中の案件にも変更した内容を反映するかどうかで方法が異なります。

●運用中の案件にも変更を反映したい場合（ciwchgdef コマンドを使用するとき）

1. 登録されているビジネスプロセス定義の変更

ciwchgdef（条件・ルール・作業アプリケーションの変更）コマンドを実行して，ワーク管理データベースに登録されているビジネスプロセス定義内のデータ条件を直接変更します。

ciwchgdef（条件・ルール・作業アプリケーションの変更）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。

なお，ciwchgdef（条件・ルール・作業アプリケーションの変更）コマンドでデータ条件を変更した場合，ワーク管理データベースに登録されているビジネスプロセス定義と，ビジネスプロセス管理ツール（CSCIW-Definer）で保存しているビジネスプロセス定義との整合性が取れなくなります。整合性を取るためには，ワーク管理データベースに登録されているビジネスプロセス定義をビジネスプロセス管理ツール（CSCIW-Definer），またはciwmngbp（ビジネスプロセス定義の管理）コマンドで取得して保存してください。

●運用中の案件にも変更を反映したい場合（ciweditdp コマンドを使用するとき）

1. ビジネスプロセス定義の変更

ビジネスプロセス管理ツール（CSCIW-Definer）で，データ条件を変更します。このとき，ビジネスプロセス定義のバージョンは変更しません。

2. 変更後のビジネスプロセス定義の反映

変更後のビジネスプロセス定義は，ciweditdp（ビジネスプロセス定義の変更）コマンドを実行して，ワーク管理データベースに登録されているビジネスプロセス定義内のデータ条件を直接変更します。

ビジネスプロセス管理ツール（CSCIW-Definer）によるビジネスプロセス定義の変更については，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。ciweditdp（ビジネスプロセス定義の変更）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。

データ条件の変更には，より手順が簡単なciwchgdef（条件・ルール・作業アプリケーションの変更）コマンドの使用をお勧めします。

●運用中の案件には変更を反映しない場合

一般的な変更手順を次に示します。

1. ビジネスプロセス定義の変更

ビジネスプロセス管理ツール（CSCIW-Definer）で，データ条件を変更します。バージョン番号は変更前より大きな番号に変更してください。必要に応じて案件投入可能期間（開始日時）を指定します。

2. 変更後のビジネスプロセス定義の登録

変更後のビジネスプロセス定義は，ビジネスプロセス管理ツール（CSCIW-Definer），またはciwmngbp（ビジネスプロセス定義の管理）コマンドを実行して，ワーク管理データベースに登録します。ビジネスプロセス管理ツール（CSCIW-Definer）によるビジネスプロセス定義の変更および登録については，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。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。

3. 変更後のビジネスプロセス定義の運用開始

指定した有効期間（開始日時）になると，変更後のビジネスプロセス定義が案件投入可能な状態になります。これ以降，変更後のビジネスプロセス定義に案件が投入されます。

7.8.3 ビジネスプロセスを変更する場合

業務システムの運用中に，業務の流れが変更され，ビジネスプロセス自体の変更が必要になる場合があります。

変更方法は，「7.8.2 データ条件を変更する場合」で示した手順と同様です。ただし，データ条件を変更する場合と比較して，ビジネスプロセス自体を変更すると，既存の案件に与える影響が大きいため，「**●運用中の案件には変更を反映しない場合**」で示した方法で変更することを推奨します。

7.8.4 データ条件評価や振り分けルール定義評価の RDB 検索に関連するデータを変更する場合

データ条件や振り分けルールの種別として RDB 検索を選択した場合，評価するごとに，定義されている SQL 文を実行し，評価結果を取得します。

RDB 検索に関連する業務データベースのデータを変更する場合，CSCIW の運用状態に関係なく，対象とする RDB 検索に関連したデータを変更してください。

データ変更後に，CSCIW が関連する RDB 検索を実行した時点で，変更したデータが有効になります。

7.9　運用上の注意事項

この節では，ワーク管理システムを運用する上での注意事項について説明します。

7.9.1　CSCIWManagementServer の開始時期について

CSCIWManagementServer の開始時期については，次の点を注意してください。

- CSCIW を使用した J2EE アプリケーション，案件運用操作，またはアプリケーション呼び出しサービスを実行する場合，CSCIWManagementServer を最初に開始するように設定してください。
- ワーク管理システムを運用するすべての J2EE サーバで，CSCIWManagementServer を開始してください。

7.9.2　日付・時刻の整合性について

CSCIW では，ワーク管理データベース上に案件の時間情報を保存しています。したがって，サーバマシンの日付や時刻を変更しても，ワーク管理データベースに保存されている案件の時間情報は変更されません。サーバマシンの日付・時刻を変更する場合は，ワーク管理データベースに保存されている案件の時間情報を考慮した上で運用してください。

サーバマシンの日付・時刻を変更した場合，次に示すような影響があります。

(1)　ビジネスプロセス定義・振り分けルール定義

ビジネスプロセス定義は，サーバマシンの日付を有効期間外の日時に設定すると，案件として投入できなくなります。同様に，振り分けルール定義も，サーバマシンの日付を有効期間外の日時に設定すると，案件に対する適用ができなくなります。

(例)

ビジネスプロセス定義に「2007 年 1 月～2007 年 12 月」の有効期間を設定していた場合，サーバマシンの日付を 2006 年に戻すと，そのビジネスプロセス定義は案件として投入できなくなります。

(2)　日時を指定できるコマンド

ciwdelpi（案件の削除）コマンドなど，日時を指定するコマンドを使用する場合は，コマンドの実行対象となる案件の日付情報が，どの時点で格納された情報かを意識する必要があります。

(例)

2007 年 2 月 1 日にシンクノードに案件 A が到達したあとに，サーバマシンの日付を 2008 年 2 月 1 日に進めた場合，ciwdelpi（案件の削除）コマンドで案件 A を含む 2007 年 2 月 1 日以降の案件を削除するには，2007 年 2 月 1 日から 2008 年 2 月 1 日までの日数を指定する必要があります。

なお，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。

7.9.3 業務アプリケーションが異常終了した場合や強制終了した（終了処理をしなかった）場合の影響について

業務アプリケーションが異常終了した場合の影響について説明します。

CSCIW を使用した業務アプリケーションが異常終了した場合や，［Ctrl］＋［C］で強制終了などして終了処理を実行しなかった場合は，データベースに CSCIW のプロセス起動情報が残ったままになっていないかを確認する必要があります。起動情報が残っていると，再起動できないことがあります。

不要な情報が残っているかどうかについては，ciwlistsid（システム ID 情報の表示）コマンドで確認します。不要な情報が残っている場合は，ciwcleanup（プロセス情報の正常化）コマンドで削除してください。

不要な情報を削除する手順を次に示します。

1. Java アプリケーション，およびコマンドを停止する

   不要な情報を確認したいマシンで，CSCIW を使用している Java アプリケーション，およびコマンドを停止します。

2. J2EE サーバを停止する

   不要な情報を確認したいマシンで，cjstopsv コマンドを使用して，CSCIW を使用している J2EE サーバを停止します。

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

3. 起動情報を確認する

   不要な情報を確認したいマシンで，ciwlistsid（システム ID 情報の表示）を実行して，起動情報を確認します。ciwlistsid（システム ID 情報の表示）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。

   次に示す形式でコマンドを実行すると，CSCIW の起動情報を確認できます。

   ```
   ciwlistsid -sid <システムID>
   ```

   システム ID が「TEST」の場合のコマンド実行例を次に示します。

   ```
   >ciwlistsid -sid TEST
   SID     NAME      STATUS
   TEST   Library     *
   TEST   Library     *
   ```

   STATUS に「*」がある場合，CSCIW を使用している Java アプリケーションおよび J2EE サーバ，およびコマンドが異常終了した情報が残っていることを示しています。

4. 起動情報を削除する

ciwcleanup（プロセス情報の正常化）コマンドを実行し，不要な起動情報を削除します。ciwcleanup（プロセス情報の正常化）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。

次に示す形式でコマンドを実行すると，不要な起動情報を削除できます。

```
ciwcleanup -sid <システムID> -del
```

重要

CSCIW を使用している Java アプリケーションおよび J2EE サーバが起動中の場合は，コマンドの実行は失敗します。CSCIW を使用している Java アプリケーションおよび J2EE サーバが動作していないことを確認してからコマンドを実行してください。

7.9.4 データベース使用時のセキュリティについて

データベース使用時のセキュリティについて説明します。

(1) データベースの管理

CSCIW で使用するデータベースは，業務データベースと同様にシステム管理者が管理する必要があります。そのため，ユーザ ID やパスワードなどのセキュリティ管理についても業務データベースと同様にシステム管理者が実施する必要があります。

(2) SQL インジェクション

CSCIW では DBMS の機能を使用してワークフロー機能を実現しています。DB の操作性や使用方法については JDBC に依存しています。

SQL 文の発行時のチェック機構および制限については，使用している JDBC および接続しているデータベースに依存しています。しかし，CSCIW では，DBMS のように厳密な SQL 文の構文チェックをしていません。振り分けルールやデータ条件の SQL についても構文のチェックをしていないため，定義系 SQL（DELETE や DROP TABLE など）を定義した場合も実行されてしまいます。また，API 発行時に引き渡されるデータについても SQL 文としての構文チェックをしていないため，一般的な SQL セキュリティの対策は，ワーク管理システムを使用する側で実施してください。

次に示すパラメタについては，CSCIW で SQL 発行時に無条件に SQL 文を付与しますので，運用の際には注意が必要です。

- 案件名
- 業務ステップ名
- 作業名

- 作業者
- 検索フィルター条件
- 検索ソート条件
- 振り分けルール実行時パラメタ
- 子作業名リスト
- ビジネスプロセス定義名
- ビジネスプロセス定義のバージョン
- 業務ステップ定義名
- 作業定義名
- 制御ノード定義名
- 案件投入者
- 案件開始日
- 案件処理期限
- 案件優先度
- 業務ステップ開始日
- 業務ステップ処理期限
- 業務ステップ優先度
- 作業生成日
- 作業開始日
- 作業処理期限
- 作業優先度
- ユーザ ID

なお，SQL のセキュリティ問題に関しては，情報処理振興事業協会（IPA）セキュリティセンターで公開している情報を参照してください。

7.9.5 Java オブジェクト呼び出し機能を使用した，Java オブジェクトの J2EE アプリケーションへの組み込みについて

Java オブジェクト呼び出し機能を使用し，Java オブジェクトを J2EE アプリケーションに組み込む場合は，CSCIW のすべての J2EE アプリケーションに Java オブジェクトおよび csciwj2ee.jar を組み込む必要があります。組み込み方法については，「4.5.5(1) J2EE アプリケーションに Java オブジェクトを組み込む場合」を参照してください。

8 障害対策

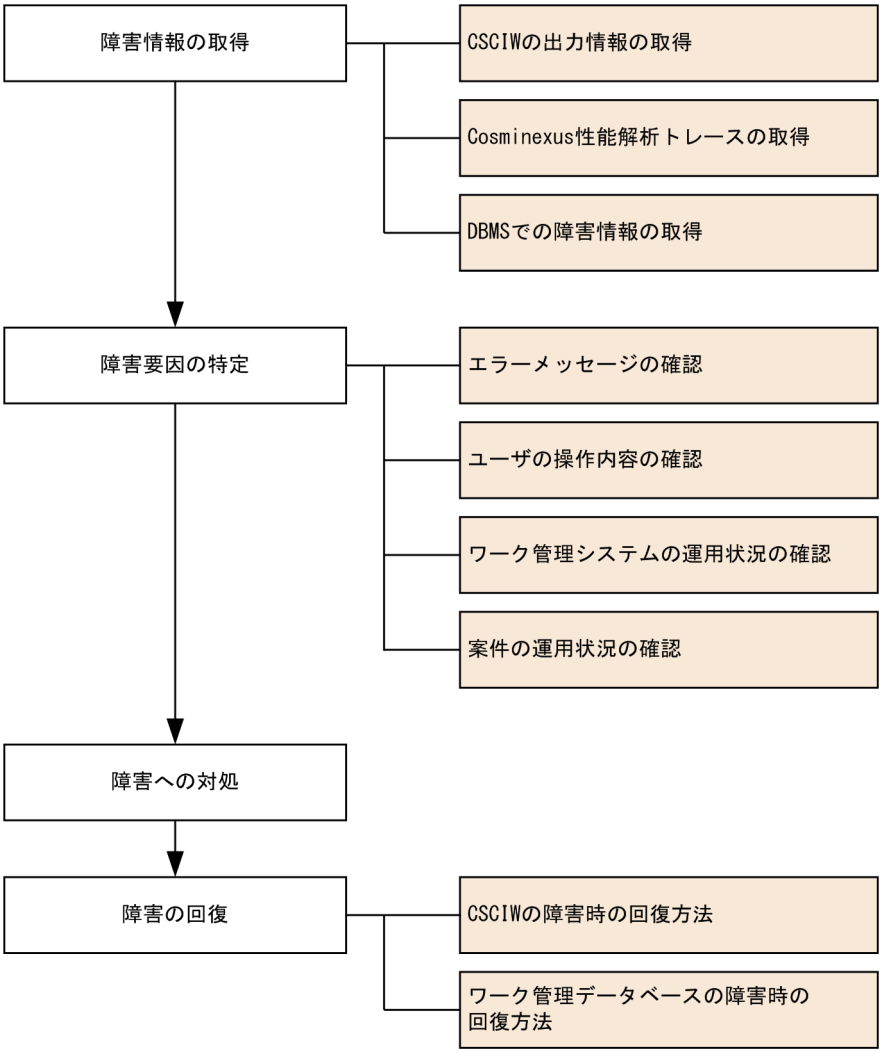
この章では，ワーク管理システムに障害が発生した場合の対処方法について説明します。

8.1　障害対策の流れ

この節では，ワーク管理システムに障害が発生したときの対処の流れについて説明します。

ワーク管理システムで障害が発生してから，対処が完了するまでの作業の流れを次の図に示します。

図 8-1　障害発生時の対処の流れ



注意事項

ワーク管理システムを複数のマシンで運用している場合に，障害が発生したときは，図 8-1 に示す対処を，該当するシステム ID で環境設定した各マシンに対して実施する必要があります。複数のマシンによる CSCIW の構成については，「3.2.3　マルチマシン構成」および「4.6.2　マルチマシン構成を設定する場合の注意事項」を参照してください。

8.2　障害情報の取得

障害が発生した場合，障害要因を特定するために必要な情報（トレース，メッセージなど）を取得します。

この節では，障害情報を取得する方法について説明します。

> **ヒント**
>
> BPMN 連携機能を使用する場合の障害情報の取得の詳細については，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。

8.2.1　CSCIW の出力情報の取得

CSCIW を運用しているマシン上で，障害情報を取得してください。

(1)　出力する情報およびファイル名

CSCIW が出力する情報，および出力ファイル名を次に示します。

表 8-1　CSCIW が出力する情報

種別	出力する情報
エラートレースファイル	初期化トレースを作成するより前にエラーが発生した場合に出力します。エラーが発生しなかった場合，このファイルは作成されません。
初期化時トレースファイル	初期化時に作成し，初期化処理中のトレース情報を出力すします。なお，初期化時の情報が上書きされて消えてしまうのを防ぐため，通常のトレースファイルとは分けてあります。
トレースファイル	初期化処理が終了したあと，通常の運用時にトレース情報を出力します。
初期化時メッセージファイル	初期化時に作成し，エラーや警告のメッセージを出力します。
メッセージファイル	エラーや警告のメッセージを出力します。
コマンドトレースファイル	コマンドを実行した際のトレース情報を出力します。
コマンドメッセージファイル	コマンドを実行した際のエラーや警告のメッセージを出力します。
案件運用操作トレースファイル	案件運用操作を実行した際のトレース情報を出力します。
案件運用操作メッセージファイル	案件運用操作を実行した際のエラーや警告のメッセージを出力します。
アプリケーション呼び出しサービストレースファイル	アプリケーション呼び出しサービスを実行した際のトレース情報を出力します。
アプリケーション呼び出しサービスメッセージファイル	アプリケーション呼び出しサービスを実行した際のエラーや警告のメッセージを出力します。

種別	出力する情報
運用管理 REST サービストレースファイル	運用管理 REST サービスを実行した際のトレース情報を出力します。
運用管理 REST サービスメッセージファイル	運用管理 REST サービスを実行した際のエラーや警告のメッセージを出力します。

表 8-2　出力ファイル名

種別	ローテーション種別 ※1	出力ファイル名
エラートレースファイル	ラップアラウンド	CIW_ERROR_<ハッシュ値※2>_TRC_<通番>.log
初期化時トレースファイル	ラップアラウンド	INIT_CIW<システム ID>_<ロック ID※3>_TRC_<通番>.log
	シフト	INIT_CIW<システム ID>_<ロック ID※3>_TRC<通番>.log
トレースファイル	ラップアラウンド	CIW<システム ID>_<ロック ID※3>_TRC_<通番>.log
	シフト	CIW<システム ID>_<ロック ID※3>_TRC<通番>.log
初期化時メッセージファイル	ラップアラウンド	INIT_CIW<システム ID>_<ロック ID※3>_MSG_<通番>.log
	シフト	INIT_CIW<システム ID>_<ロック ID※3>_MSG<通番>.log
メッセージファイル	ラップアラウンド	CIW<システム ID>_<ロック ID※3>_MSG_<通番>.log
	シフト	CIW<システム ID>_<ロック ID※3>_MSG<通番>.log
コマンドトレースファイル	ラップアラウンド	CIWCMD_<システム ID>_<コマンド名※4>_TRC_<通番>.log
	シフト	CIWCMD_<システム ID>_<コマンド名※4>_TRC<通番>.log
コマンドメッセージファイル	ラップアラウンド	CIWCMD_<システム ID>_<コマンド名※4>_MSG_<通番>.log
	シフト	CIWCMD_<システム ID>_<コマンド名※4>_MSG<通番>.log
案件運用操作トレースファイル	ラップアラウンド	CIWADM_<システム ID>_<ロック ID※3>_TRC_<通番>.log
	シフト	CIWADM_<システム ID>_<ロック ID※3>_TRC<通番>.log
案件運用操作メッセージファイル	ラップアラウンド	CIWADM_<システム ID>_<ロック ID※3>_MSG_<通番>.log
	シフト	CIWADM_<システム ID>_<ロック ID※3>_MSG<通番>.log
アプリケーション呼び出しサービストレースファイル	ラップアラウンド	APPCALLSVC_<システム ID>_<アプリケーション番号※5>_<ロック ID※3>_TRC_<通番>.log
	シフト	APPCALLSVC_<システム ID>_<アプリケーション番号※5>_<ロック ID※3>_TRC<通番>.log
アプリケーション呼び出しサービスメッセージファイル	ラップアラウンド	APPCALLSVC_<システム ID>_<アプリケーション番号※5>_<ロック ID※3>_MSG_<通番>.log
	シフト	APPCALLSVC_<システム ID>_<アプリケーション番号※5>_<ロック ID※3>_MSG<通番>.log

種別	ローテーション種別 ※1	出力ファイル名
運用管理 REST サービストレースファイル	ラップアラウンド	MNGRESTSVC_<システム ID>_<ロック ID※3>_TRC_<通番>.log
	シフト	MNGRESTSVC_<システム ID>_<ロック ID※3>_TRC<通番>.log
運用管理 REST サービスメッセージファイル	ラップアラウンド	MNGRESTSVC_<システム ID>_<ロック ID※3>_MSG_<通番>.log
	シフト	MNGRESTSVC_<システム ID>_<ロック ID※3>_MSG<通番>.log

注※1

ローテーション種別については，「(2)　出力ファイルのローテーションの設定」を参照してください。

注※2

JavaVM が java.lang.Runtime のインスタンスに付与したハッシュ値が入ります。

注※3

OS のプロセス間で排他を取るためのロック ID が入ります。

注※4

先頭の「ciw」を省略したコマンド名が入ります。

ciwstartpi コマンドで-n オプションを指定した場合，コマンド名は「startpiXX」形式になります。

（凡例）XX：-n オプションで指定した値

注※5

J2EE サーバにインポートしたアプリケーション呼び出しサービスの J2EE アプリケーション名の，末尾の番号が入ります。

(2)　出力ファイルのローテーションの設定

CSCIW が出力するメッセージ・トレースのローテーション種別は，次の 2 種類のモードがあります。メッセージ・トレースのローテーション種別は，セットアッププロパティファイルのLogRotationMode キーで設定します。セットアッププロパティファイルの設定内容については，「付録 A.2　セットアッププロパティファイル」を参照してください。

(a)　ラップアラウンドモード

- ローテーション条件

  指定ファイル容量を超えた場合

- ローテーション時のファイル名

  ファイル名が xxx_TRC の場合，xxx_TRC_1.log，xxx_TRC_2.log…のように通番が付与されたファイルが作成されます。通番の範囲は，1～（設定したファイル面数）です。ファイル面数分のファイルが作成されたあとでローテーションが発生すると，出力先が通番 1 のファイルにローテーションします。

  ラップアラウンドモードのファイル名の付与例を次に示します。

```
xxx_TRC_1.log
xxx_TRC_2.log
xxx_TRC_3.log
```

(b) シフトモード

- ローテーション条件

  指定ファイル容量を超えた場合，または指定時刻になった時

- ローテーション時のファイル名

  ファイル名が xxx_TRC の場合，トレースを出力するファイルは常に通番無しの xxx_TRC.log（現用系ファイル）になります。設定したローテーション条件を満たし，ローテーションが発生すると，バックアップファイルに対して，xxx_TRC1.log，xxx_TRC2.log…のように通番が付与されたファイル名に変更（リネーム）されます。このとき，通番が小さいファイルから順に，新しいファイルとなります。通番の範囲は，1～（設定したバックアップ面数）です。また，シフトモードの場合には，トレースファイル以外に条件やローテーション状態を管理するためのコントロールファイルが生成されます。このため，ファイルの合計数は，指定したバックアップ面数＋ 2（現用系ファイル，コントロールファイル）となります。

  シフトモードのファイル名の付与例を次に示します。

```
xxx_TRC.conf      ←  コントロールファイル
xxx_TRC.log       ←  現用系ファイル
xxx_TRC1.log      ←  バックアップファイル
xxx_TRC2.log      ←  バックアップファイル
xxx_TRC3.log      ←  バックアップファイル
```

(3) 出力情報の設定

(a) 設定方法

エラートレース以外のトレースファイルとメッセージファイルに関してはセットアッププロパティファイルで設定します。セットアッププロパティファイルについては，「付録 A.2 セットアッププロパティファイル」を参照してください。

エラートレースの出力先ディレクトリは環境変数で設定します。

エラートレースの設定値を次の表に示します。

表 8-3　エラートレースの設定値

設定項目	出力する情報
出力先ディレクトリ	環境変数で設定（${CSCIW_HOME}/log）※
トレースレベル	固定（20）
面数	固定（16）
サイズ	固定（2MB）
削除までの日数	固定（2 日）
ローテーション種別	固定(ラップアラウンド)

注※

環境変数で設定していない場合は，カレントディレクトリに出力されます。また，環境変数で設定している場合でも，設定したディレクトリが使用できないときは，カレントディレクトリに出力されます。

(b) 出力レベルの設定

エラートレース以外の出力ファイルは，運用および障害発生時の対応時間を考慮し，出力レベルを設定できます。

設定できる出力レベル，および各レベルで出力される内容を次の表に示します。

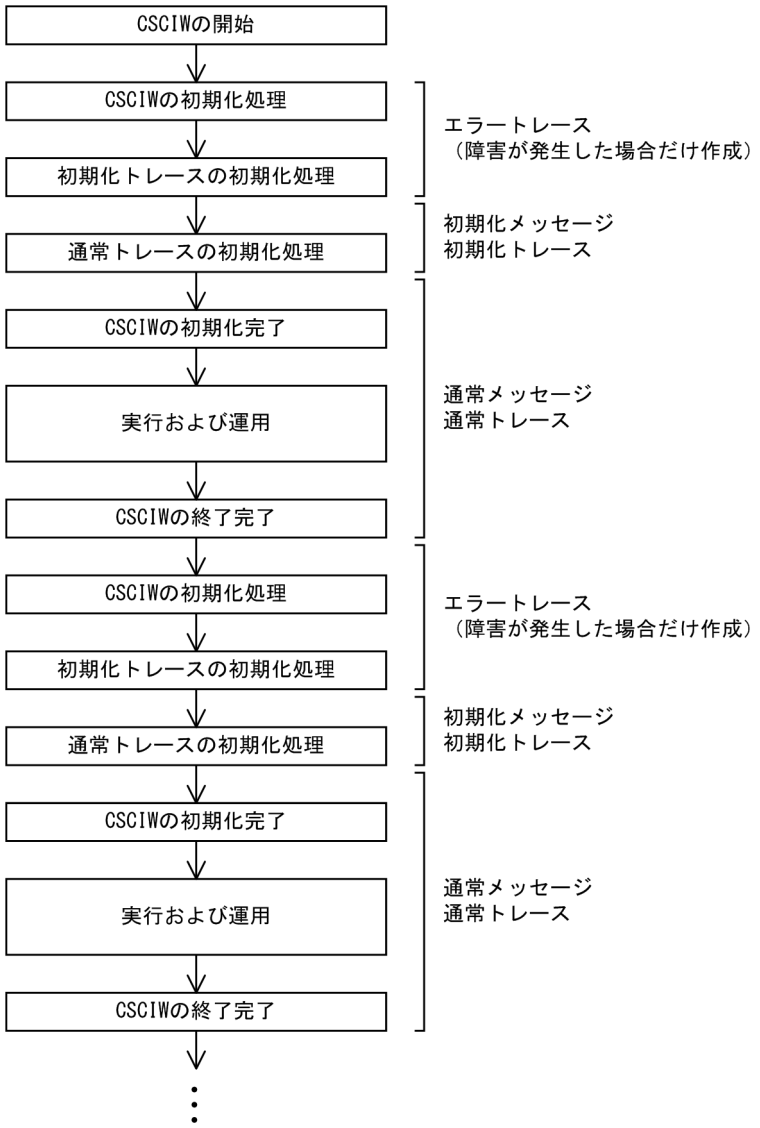
表 8-4 出力レベルおよび出力内容

出力レベル	設定基準	出力内容	
		トレースファイル	メッセージファイル
-1	トレース情報の出力を抑止	−	−
0	エラー発生時の情報だけが出力されるレベル	• 例外発生 • 例外発生時のスタックトレース	• エラーメッセージ • 初期化・終了メッセージ
10	障害発生時の調査より性能およびトレース出力量を優先したレベル	• 製品公開メソッドの開始・終了（属性取得メソッドは除く） • 他プログラムの呼び出し・戻り（一部頻度の高いものは除く）	警告メッセージ
20	通常運用で利用されることを想定したレベル	内部の主要メソッドの開始・終了	インフォメーション
30	障害発生時の調査を優先したレベル	• 一部他プログラムの呼び出し・戻り（一部頻度の高いもの） • 一部の内部メソッドの開始・終了	
40	障害発生時の調査を最優先したレベル	内部メソッドの開始・終了	

(4) 各種トレースファイルの切り替えの流れ

各種トレースファイルが切り替わるのは，各種トレースの初期化処理が完了した時点です。各種トレースファイルの切り替えの流れを次の図に示します。

図 8-2　各種トレースファイルの切り替えの流れ



説明

上の図に示すように，終了処理が行われた時点で，通常トレースファイルと通常メッセージファイルはクローズするため，トレースおよびメッセージの出力は停止します。トレースおよびメッセージの再出力が開始されるのは初期化トレースの初期化完了後になります。ただし，例外発生などのエラーが発生した場合には，エラートレースファイルを作成します。また，初期化メッセージおよび通常メッセージの作成に失敗した場合もエラートレースを作成します。

(5)　トレースファイル取得時の注意事項

(a)　トレースファイルの自動削除機能について

CSCIW が出力するトレースファイルは，最終更新日より設定した日数を経過すると最初の出力処理時に，対象トレースファイルを自動で削除します。

自動で削除されるまでの日数は，エラートレースについては固定（2 日），エラートレース以外のトレースファイルについては，セットアッププロパティファイルで指定できます。

なお，削除されるエラートレースは，環境変数 CSCIW_HOME 下の log ファイル（${CSCIW_HOME}/log）が対象です。環境変数 CSCIW_HOME が未設定時の場合は，エラートレースファイルは削除されません。また，カレントディレクトリに出力されたエラートレースは，不要になった時点で削除してください。

重要

Windows の場合，1970 年 1 月 1 日より以前のファイルを削除するような日数を指定した場合，すべてのトレースファイルが削除されます。

メモ

UNIX の場合，1970 年 1 月 1 日より以前のファイルを削除するような日数を指定した場合，すべてのトレースファイルが削除されるようなことはありません。指定に従ってファイルが自動的に削除されます。

(b) 改行文字の扱いについて

各種トレースファイル（Cosminexus 性能解析トレースを除く）に出力されるメッセージ本文（ファイル中の message の列）に改行文字が含まれている場合は，改行文字の個所でメッセージを区切り，改行文字以降のメッセージを次の行へ出力します。複数行にわたるメッセージについては，メッセージ ID およびメッセージタイプを 1 行目に出力し，2 行目以降にはメッセージ ID およびメッセージタイプを出力しません。

Cosminexus 性能解析トレースについては，メッセージ本文に改行文字が含まれていると，OPT の列に「0a」，ASCII の列に「.」が出力されます。

(c) 1 行に出力されるメッセージのバイト数について

各種トレースファイル（Cosminexus 性能解析トレースを除く）のメッセージ本文（ファイル中の message の列）に出力される出力バイト数は，メッセージ 1 行につき最大 4,095 バイトです。4,095 バイトを超えるメッセージ本文については 4,096 バイト目以降が削除され，4,095 バイト目までを出力します。

(d) Unicode の補助文字について

Unicode の補助文字とは，基本多言語面（BMP）以外の文字，Unicode のコードポイント U+10000～U+10FFFF の範囲の文字，UTF-16 エンコーディングでのサロゲート・ペアで表される文字のことです。この Unicode の補助文字は，Windows Vista でサポートされた JIS X0213:2004 の第三水準および第四水準の文字の一部に含まれます。メッセージ本文（ファイル中の message の列）に Unicode の補助文字がある場合は，各種トレースファイルには該当する補助文字の部分が正しく出力されませんが，同一行および同一ファイル内のそのほかの文字出力には影響ありません。

8.2.2　Cosminexus 性能解析トレースの取得

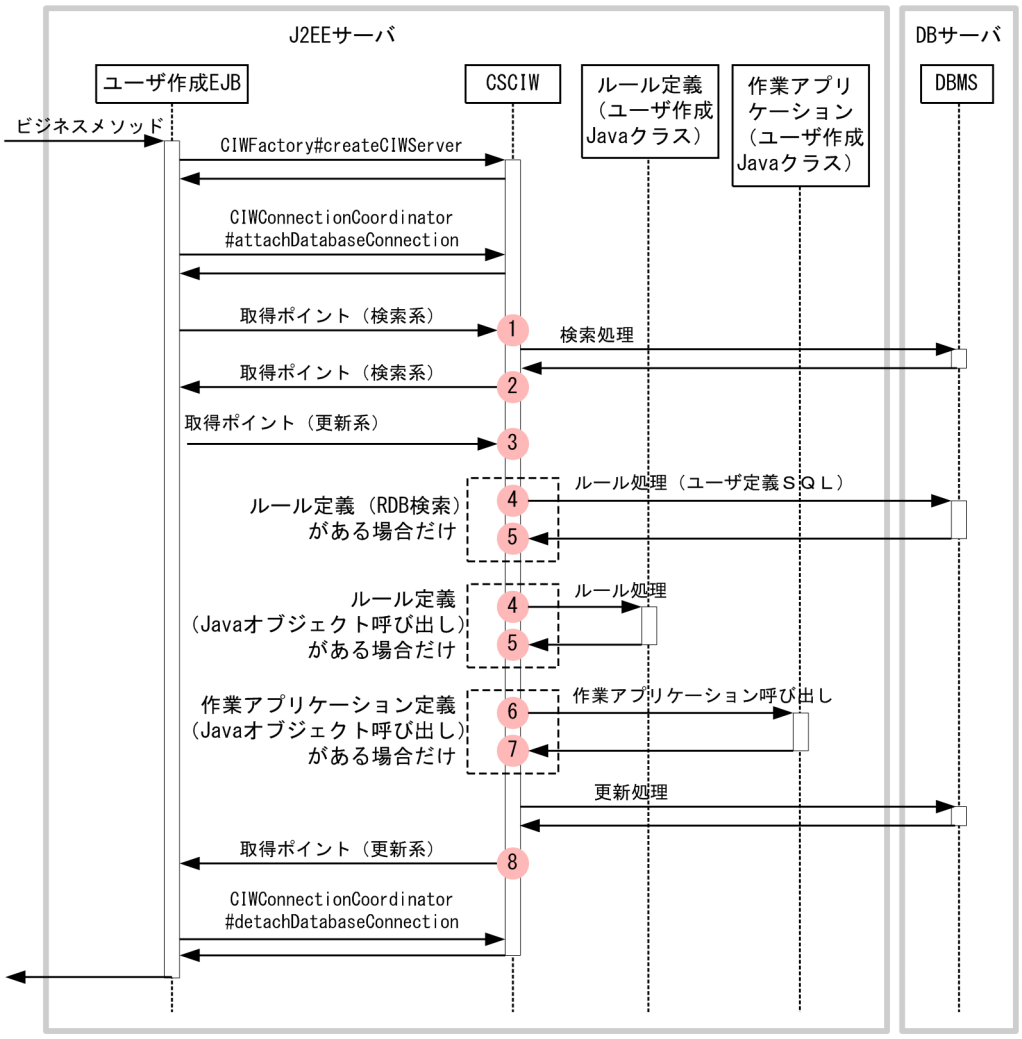
CSCIW は，Cosminexus の共通機能である性能解析トレースに対応しています。

CSCIW の性能解析トレースの取得ポイント，および CSCIW が性能解析トレースとして出力する情報を説明します。

(1)　性能解析トレースの取得ポイント

CSCIW の性能解析トレースの取得ポイントを次の図に示します。

図 8-3　CSCIW の性能解析トレースの取得ポイント



（凡例）

EJB ：Enterprise JavaBeans

●：トレース取得ポイントを示します。トレース取得レベルは「標準」です。

(2)　性能解析トレースとして出力する情報

CSCIW の性能解析トレースとして，インタフェース名にクラス名，オペレーション名にメソッド名，追加情報に案件 ID などの情報を出力します。

性能解析トレースで出力するイベント ID およびメソッドの概要を次の表に示します。なお，CSCIW のイベント ID の範囲は 0x9E00～0x9EFF です。また，取得個所の数字は，図 8-3 のトレース取得ポイントの数字と対応しています。

表 8-5　CSCIW が性能解析トレースで出力するイベント ID およびメソッドの概要

クラス名	メソッド名※	メソッドの概要	種類	イベント ID	取得個所
CIWServer	createAndStartProcessInstance	案件を生成し開始する。	call	0x9E04	3
			out	0x9E05	8
	createProcessInstance	案件を生成する。	call	0x9E06	3
			out	0x9E07	8
	getActivityInstance	業務ステップを取得する。	call	0x9E08	1
			out	0x9E09	2
	getActivityInstancesList	業務ステップのリストを取得する。	call	0x9E0A	1
			out	0x9E0B	2
	getCastingRulesList	振り分けルールのリストを取得する。	call	0x9E56	1
			out	0x9E57	2
	getListCount	各種オブジェクトのリスト件数を取得する。	call	0x9E0C	1
			out	0x9E0D	2
	getProcessDefinition	ビジネスプロセス定義を取得する。	call	0x9E0E	1
			out	0x9E0F	2
	getProcessDefinitionsList	ビジネスプロセス定義のリストを取得する。	call	0x9E10	1
			out	0x9E11	2
	getProcessInstance	案件を取得する。	call	0x9E12	1

クラス名	メソッド名※	メソッドの概要	種類	イベントID	取得個所
CIWServer	getProcessInstance	案件を取得する。	out	0x9E13	2
	getProcessInstancesList	案件のリストを取得する。	call	0x9E14	1
			out	0x9E15	2
	getWorkDefinition	作業定義を取得する。	call	0x9E16	1
			out	0x9E17	2
	getWorkDefinitionsList	作業定義のリストを取得する。	call	0x9E18	1
			out	0x9E19	2
	getWorkItem	作業を取得する。	call	0x9E1A	1
			out	0x9E1B	2
	getWorkItemsList	作業のリストを取得する。	call	0x9E1C	1
			out	0x9E1D	2
	allocateWorkItemEx	条件に一致する作業に，作業者を割り当てる。	call	0x9E64	3
			out	0x9E65	8
	getActivityDefinition	業務ステップ定義を取得する。	call	0x9E6C	1
			out	0x9E6D	2
	getActivityDefinitionsList	業務ステップ定義のリストを取得する。	call	0x9E6E	1
			out	0x9E6F	2
CIWActivityInstance	changeState	業務ステップの状態を変更する。	call	0x9E22	3
			out	0x9E23	8
	evaluate	業務ステップの条件を評価する。	call	0x9E52	3
			out	0x9E53	8

クラス名	メソッド名※	メソッドの概要	種類	イベントID	取得個所
CIWActivityInstance	makeBackwardTransition	業務ステップを差し戻す(引き戻す)。	call	0x9E58	3
			out	0x9E59	8
	setDeadline	絶対日時として業務ステップの処理期限を設定する。	call	0x9E24	3
			out	0x9E25	8
	setPriority	業務ステップの優先度を変更する。	call	0x9E26	3
			out	0x9E27	8
	adhocMakeTransition	業務ステップ間のフロー遷移を実行する。	call	0x9E62	3
			out	0x9E63	8
CIWProcessDefinition	getProcessInstancesList	案件のリストを取得する。	call	0x9E28	1
			out	0x9E29	2
CIWProcessInstance	changeState	案件の状態を変更する。	call	0x9E2A	3
			out	0x9E2B	8
	delete	案件を削除する。	call	0x9E2C	3
			out	0x9E2D	8
	getProcessDefinition	ビジネスプロセス定義を取得する。	call	0x9E2E	1
			out	0x9E2F	2
	getWorkItemsList	作業のリストを取得する。	call	0x9E30	1
			out	0x9E31	2
	setDeadline	絶対日時として案件の処理	call	0x9E32	3
			out	0x9E33	8

クラス名	メソッド名※	メソッドの概要	種類	イベントID	取得個所
CIWProcessInstance	setDeadline	期限を設定する。	out	0x9E33	8
	setPriority	案件の優先度を変更する。	call	0x9E34	3
			out	0x9E35	8
	start	案件を開始する。	call	0x9E36	3
			out	0x9E37	8
	terminate	案件を強制終了する。	call	0x9E38	3
			out	0x9E39	8
	adhocInstantiateActivityInstance	案件の業務ステップを動的に生成する。	call	0x9E60	3
			out	0x9E61	8
	startAndAdhocInstantiateActivityInstanceByADID	案件を開始し，業務ステップ定義IDで指定した業務ステップを生成して実行状態にする。	call	0x9E68	3
			out	0x9E69	8
	startAndAdhocInstantiateActivityInstanceByADName	案件を開始し，業務ステップ定義名で指定した業務ステップを生成して実行状態にする。	call	0x9E6A	3
			out	0x9E6B	8
CIWWorkItem	changeState	作業の状態を変更する。	call	0x9E3A	3
			out	0x9E3B	8
	complete	作業を「実行中」	call	0x9E3C	3

クラス名	メソッド名※	メソッドの概要	種類	イベントID	取得個所
CIWWorkItem	complete	状態から「完了」状態にする。	out	0x9E3D	8
	getParentProcessInstance	作業が所属する案件を取得する。	call	0x9E3E	1
			out	0x9E3F	2
	getWorkDefinition	作業定義を取得する。	call	0x9E40	1
			out	0x9E41	2
	perform	作業を「実行開始可能」状態から「作業者実行」状態にする。	call	0x9E42	3
			out	0x9E43	8
	performAndComplete	作業を「実行開始可能」状態から「作業者実行」状態にし「完了」状態にする。	call	0x9E44	3
			out	0x9E45	8
	reassign	作業者を，指定した別の作業者に割り当てる。	call	0x9E46	3
			out	0x9E47	8
	reassignAndPerform	作業者を，指定した別の作業者に割り当てし，「未実行」状態から	call	0x9E48	3
			out	0x9E49	8

クラス名	メソッド名※	メソッドの概要	種類	イベントID	取得個所
CIWWorkItem	reassignAndPerform	「実行中」状態にする。	out	0x9E49	8
	evaluate	作業の条件を評価する。	call	0x9E54	3
			out	0x9E55	8
	reevaluateCastingRule	振り分けルールを再評価し，作業者を再割り当てする。	call	0x9E4A	3
			out	0x9E4B	8
	setDeadline	絶対日時として作業の処理期限を設定する。	call	0x9E4C	3
			out	0x9E4D	8
	setPriority	指定した優先度を設定する。	call	0x9E4E	3
			out	0x9E4F	8
	free	作業を返却して，振り分けルールを再評価する。	call	0x9E66	3
			out	0x9E67	8
SQLEvaluatorUtil	evaluate	ルール評価を実行する(RDB検索)。	call	0x9E50	4
			out	0x9E51	5
JavaEvaluatorUtil	evaluate	ルール評価を実行する(Javaオブジェクト呼び出し)。	call	0x9E5C	4
			out	0x9E5D	5

クラス名	メソッド名※	メソッドの概要	種類	イベントID	取得個所
JavaInvoker	startSynchronously	作業アプリケーションを呼び出す(Java オブジェクト呼び出し)。	call	0x9E5E	6
			out	0x9E5F	7

注※

各メソッドについては，マニュアル「uCosminexus Service Coordinator Interactive Workflow AP 開発ガイド」を参照してください。

8.2.3 snapshot ログの収集

Cosminexus には，アプリケーションサーバの障害発生時に，スレッドダンプや性能解析トレース，各種構成ソフトウェアのログを自動的に収集し出力する snapshot ログがあります。snapshot ログ情報については，マニュアル「Cosminexus アプリケーションサーバ 機能解説 保守／移行編」の「snapshot ログの収集」の説明を参照してください。

CSCIW の出力するメッセージ・トレースファイル，設定情報を snapshot ログの対象とすることで，障害発生時に CSCIW の稼動情報が自動的に収集されるようになります。

(1) snapshot 収集定義ファイル

CSCIW では，メッセージ・トレースファイルの容量が大きくなるおそれがあるため，一次収集資料は取得せず，二次収集資料として CSCIW のメッセージ・トレースファイル，設定情報を収集することを推奨します。

snapshot ログ収集対象を記載する収集対象定義ファイルを次に示します。

表 8-6 収集対象定義ファイル一覧

ファイル名	ファイルパス	内容
snapshotlog.conf	• Windows の場合 <Cosminexus のインストールディレクトリ>¥manager¥config¥snapshotlog.conf • UNIX の場合 /opt/Cosminexus/manager/config/snapshotlog.conf	一次送付資料として取得する内容
snapshotlog2.conf	• Windows の場合	二次送付資料として取得する内容

ファイル名	ファイルパス	内容
snapshotlog2.conf	<Cosminexus のインストールディレクトリ>¥manager¥config¥snapshotlog.2.conf • UNIX の場合 /opt/Cosminexus/manager/config/snapshotlog2.conf	二次送付資料として取得する内容

(2) 収集対象

実行環境と開発環境の snapshot ログ収集対象を次に示します。

表 8-7 実行環境と開発環境の収集対象一覧

情報	実行環境	開発環境
CSCIW 環境設定ファイル	○	○
CSCIW メッセージ・トレースファイル	○	○
CSCIW エラートレースファイル	○	○
CSCIW-Definer 環境設定ファイル	×	○
CSCIW-Definer メッセージ・トレースファイル	×	○
eclipse 稼動情報ログ	×	○

重要

収集対象一覧に示した資料以外は，収集対象外となるため，障害発生時に別途収集する必要があります。

(3) 記述例

snapshot ログ収集対象定義ファイルに，実行環境と開発環境の収集対象を追加する例を次に示します。

表 8-8 インストール先ディレクトリ・環境変数の設定例

項目	インストール先ディレクトリ・設定値
CSCIW インストール先	• Windows の場合 C:¥Program Files¥HITACHI¥CSCIW • UNIX の場合 /opt/hitachi/CSCIW
CSCIW_HOME	• Windows の場合 C:¥Program Files¥HITACHI¥CSCIW • UNIX の場合 /opt/hitachi/CSCIW
CSCIW_CONF_DIR	• Windows の場合

項目	インストール先ディレクトリ・設定値
CSCIW_CONF_DIR	記載なし（デフォルト値：%CSCIW_HOME%¥conf） • UNIX の場合 記載なし（デフォルト値：${CSCIW_HOME}/conf）
CSCIW ログ出力先	• Windows の場合 記載なし（デフォルト値：C:¥Program Files¥ HITACHI¥CSCIW¥log） • UNIX の場合 記載なし（デフォルト値：/opt/hitachi/CSCIW/log）
CSCIW-Definer インストール先	• Windows の場合 C:¥Program Files¥HITACHI¥CSCIWBP
CSCIW-Definer ログ出力先	• Windows の場合 記載なし（デフォルト値：C:¥Program Files¥HITACHI¥CSCIWBP¥log）
eclipse ワークスペース	• Windows の場合 C:¥eclipse¥workspace

二次収集対象定義ファイル（snapshotlog2.conf）への記述例を次に示します。

- Windows の場合

  CSCIW 環境設定ファイルを収集する場合

  ```
  C:/Program Files/HITACHI/CSCIW/conf/.+
  ```

  CSCIW メッセージ・トレースファイル，CSCIW エラートレースファイルを収集する場合

  ```
  C:/Program Files/HITACHI/CSCIW/log/.+
  ```

  CSCIW-Definer 環境設定ファイルを収集する場合

  ```
  C:/Program Files/HITACHI/CSCIWBP/conf/.+
  ```

  CSCIW-Definer メッセージ・トレースファイルを収集する場合

  ```
  C:/Program Files/HITACHI/CSCIWBP/log/.+
  ```

  eclipse 稼動情報ログを収集する場合

  ```
  C:/eclipse/workspace/¥.metadata/.+
  ```

- UNIX の場合

  CSCIW 環境設定ファイルを収集する場合

  ```
  /opt/hitachi/CSCIW/conf/.+
  ```

  CSCIW メッセージ・トレースファイル，CSCIW エラートレースファイルを収集する場合

  ```
  /opt/hitachi/CSCIW/log/.+
  ```

重要

- ディレクトリのセパレータは"/"で記載してください。(OS 共通)
- ".+"（ピリオド，プラス）は，ディレクトリ直下の全ファイルが収集対象となることを表します。
- "¥"はエスケープ文字です。
- 収集対象のファイル名に日本語が含まれている場合，収集後のファイル名が文字化けしますが，ファイルの内容に影響はありません。

8.2.4 DBMS での障害情報の取得

出力されたエラーメッセージが DBMS に関するメッセージの場合，ワーク管理データベース，および業務データベースを運用しているマシン上で，障害情報を取得してください。障害情報の取得については，使用している DBMS のマニュアルを参照してください。

8.3　障害要因の特定

この節では，障害要因の特定方法について説明します。

障害が発生した場合，その対処方法を決めるためには，その障害の要因を特定する必要があります。障害の要因を特定するには，次の項目を確認してください。

- エラーメッセージ
- ユーザの操作内容
- ワーク管理システムの運用状況
- 案件の運用状況

8.3.1　エラーメッセージの確認

障害の要因を特定するために，まず，CSCIW が作成するメッセージファイルにエラーメッセージが出力されているかどうかを確認してください。

エラーメッセージの格納パスは，次のとおりです。

- Windows の場合

```
%CSCIW_HOME%¥log
```

- UNIX の場合

```
${CSCIW_HOME}/log
```

> **メモ**
>
> エラーメッセージの出力先を変更できます。出力先の変更については，「付録 A.2 セットアッププロパティファイル」を参照してください。

エラーメッセージが出力されている場合，エラーメッセージのメッセージ ID からエラーが発生しているプログラム，障害の要因が特定できます。メッセージ ID の概要は次のとおりです。

メッセージ ID	障害の検出元
KDIW00000～KDIW00199	製品内共通
KDIW00200～KDIW19999	実行基盤※
KDIW20000～KDIW29999	コマンド
KDIW30000～KDIW39999	案件運用操作
KDIW40000～KDIW49999	ビジネスプロセス管理（CSCIW-Definer）

メッセージ ID	障害の検出元
KDIW60000～KDIW69999	BPMN 連携機能

注※

実行基盤とは，API や各種処理の管理など，CSCIW の実行環境および運用環境（コマンドや案件運用操作を除く）のことです。

エラー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。

8.3.2 ユーザの操作内容の確認

障害が発生したときに実行していた操作の詳細な手順を確認してください。正しい操作手順をしていないことが要因で，障害が発生していることがあります。操作手順を確認することで，障害の要因がある程度特定できます。

8.3.3 ワーク管理システムの運用状況の確認

障害が発生した場合，ワーク管理システムの運用状況を確認してください。ワーク管理システムが正常に運用していないことが要因で，障害が発生していることがあります。運用状況を確認することで，障害の要因がある程度特定できます。確認する内容としては次のものが挙げられます。

●CSCIW を使用している Java アプリケーションおよび J2EE サーバ

CSCIW を使用している Java アプリケーションおよび J2EE サーバが正しく運用されているかどうかを確認してください。何らかの理由で CSCIW を使用している Java アプリケーションおよび J2EE サーバが終了していることがあります。

●作業者データベース

ワーク管理システムとは別に運用されている作業者データベースが，正しく動作しているかどうかを確認してください。案件の作業の作業者を決定するために参照した作業者データベースが正しく運用されていないことがあります。

●業務データベース

業務データベースが正しく運用されているかどうかを確認してください。案件の作業などに指定されたデータ条件を評価するために参照した業務データベースが正しく運用されていないことがあります。

8.3.4 案件の運用状況の確認

障害が発生した場合，案件の運用状況を確認してください。案件が正常に運用されていないことが要因で，障害が発生していることがあります。状況を確認することで，障害の要因がある程度特定できます。

案件の運用状況は，案件運用操作で確認できます。また，ビジネスプロセス定義および振り分けルール定義の状態は，ビジネスプロセス管理ツール（CSCIW-Definer），または次のコマンドで確認できます。

- ciwmngbp（ビジネスプロセス定義の管理）コマンド
  ビジネスプロセス定義の状態を確認できます。
- ciwmngcr（振り分けルール定義の管理）コマンド
  振り分けルール定義の状態を確認できます。

確認する内容としては次のものが挙げられます。

●案件の状態

管理者によって案件が停止されていないかどうかを確認してください。管理者によって，その案件が一時停止または強制終了されていることがあります。

●ビジネスプロセス定義，振り分けルール定義の定義内容

ビジネスプロセス定義，振り分けルール定義の定義内容が正しいかどうかを確認してください。定義内容を新規作成，または変更したあとに障害が発生している場合，定義が正しく作成されていないことがあります。

●ビジネスプロセス定義，振り分けルール定義の運用状況

ビジネスプロセス定義，振り分けルール定義が，正しく運用されているかどうかを確認してください。定義を新規作成または変更した直後に障害が発生している場合，定義が正しく運用されていないことがあります。

案件運用操作については，マニュアル「uCosminexus Service Coordinator Interactive Workflow 案件運用操作ガイド」を，ビジネスプロセス管理ツール（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，それぞれ参照してください。

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。

8.4　障害への対処

この節では，障害への対処方法について説明します。

障害要因が判明したら，適切な対処方法で障害の要因を取り除いてください。エラーメッセージが出力されている場合，対処方法をマニュアル「uCosminexus Service Coordinator Interactive Workflow メッセージ」で確認し，それに従って対処してください。エラーメッセージを確認する方法については，「8.3.1 エラーメッセージの確認」を参照してください。

サポートサービスへの連絡が必要になる障害の場合，または対処方法が不明の障害の場合，必要な障害情報を取得して，サポートサービスに連絡し，その指示に従って対処してください。

障害情報の取得の方法については，「8.2　障害情報の取得」を参照してください。

8.5　障害の回復

この節では，ワーク管理システムで発生した障害の回復方法について説明します。

8.5.1　CSCIW の障害時の回復方法

CSCIW のプロセスが異常終了した場合，障害要因を取り除いたあとに，異常終了したプロセスを再起動してください。

8.5.2　ワーク管理データベースの障害時の回復方法

ワーク管理データベースに障害が発生した場合，使用している DBMS の運用手順に従って障害要因を取り除いて，必要な回復処理をしてください。なお，CSCIW のプロセスが起動されている場合，これらのプロセスを停止したあとに障害回復処理をする必要があります。

CSCIW の起動は，ワーク管理データベースの DBMS の障害回復処理が完了して，ワーク管理データベースを再起動したあとに，実施してください。

8.6　トラブルシューティング

この節では，ワーク管理システムの運用時に発生するトラブルの対処方法の一例について説明します。

8.6.1　特定の案件が推進されない

現象

特定の案件が推進されません。

要因

次の要因が考えられます。

- ビジネスプロセス定義に不備がある
  ビジネスプロセス定義と運用されている環境との間に不整合があることがあります。
- 振り分けルール定義を適用した結果として作業者を決定できない
  振り分けルールで定義した，条件を満たすユーザが存在しないことがあります。
- 適用する振り分けルール定義が運用されていない
  振り分けルール定義が非活性状態または有効期間外になっていることがあります。
- ワーク管理システムの作業者データベースの参照時にエラーが発生した
  作業者データベースが運用されていないことがあります。
- 業務データベースの参照時にエラーが発生した
  業務データベースが正しく運用されていないことがあります。

対処

次に示す手順で対処してください。

1. 例外の発生した API を特定します。
2. メッセージファイルから，エラーが発生した要因を取得します。
3. 業務データベースを運用する，振り分けルール定義を適用できるようにするなどの対処をしたあとに，再度 API を発行します。

以上の手順で，その案件に対する処理は続行されます。

8.6.2　すべての案件が推進されない

現象

すべての案件が推進されません。

要因

次の要因が考えられます。

- CSCIWManagementServer が停止している
- ワーク管理データベースが終了している

対処

次に示す手順で対処してください。

1. 出力されるエラーメッセージで，エラーの要因を確認します。
2. CSCIWManagementServer およびワーク管理データベースを正しく起動します。

以上の手順で，案件に対する処理は続行されます。

8.6.3 CSCIWManagementServer または業務アプリケーションが起動できない

現象

CSCIWManagementServer または業務アプリケーションが起動できません。

要因

次の要因が考えられます。

- 環境変数が設定されていない，または環境変数の値が正しくない
- 環境設定ファイルの内容に不備がある
- ワーク管理データベースが運用されていない

対処

次に示す手順で対処してください。

1. 初期化時メッセージファイルまたはメッセージファイルに出力されているエラーメッセージので，エラーの要因を確認します。
2. 要因に合わせて，次の対処をします。
   ・環境変数を確認します。
   ・環境設定ファイルの内容を確認します。
   ・ワーク管理データベースを正しく起動します。

8.7　JP1 によるログファイルの監視

JP1 のログファイルトラップ機能を使用して，CSCIW が出力するメッセージファイルを監視する場合，CSCIW のメッセージファイルのローテーション種別の設定※1 に応じて，JP1 のログファイルトラップ動作定義ファイルのFILETYPE パラメーター※2 を次のとおりに設定してください。

ローテーション種別	FILETYPE パラメーターに設定する値
ラップアラウンドモード	ラップアラウンドファイル（WRAP2）
シフトモード	シーケンシャルファイル（SEQ2）

注※1

ローテーション種別の設定については，「付録 A.2　セットアッププロパティファイル」を参照してください。

注※2

FILETYPE パラメーターについては，マニュアル「JP1 JP1/Base 運用ガイド」を参照してください。

重要

CSCIW のメッセージファイルのローテーション種別がラップアラウンドモードの場合，JP1 によるログファイル監視中に対象のメッセージファイルが削除されると，正常に監視ができない状態になります。そのため，メッセージファイル削除までの日数を「-1」（削除しない）にしてください。

詳細については，「付録 A.2　セットアッププロパティファイル」および，マニュアル「uCosminexus Service Coordinator Interactive Workflow BPMN 連携機能　使用の手引」の「共通設定ファイルに指定する内容」を参照してください。

9 ワーク管理システムのバージョンアップ

この章では，ワーク管理システムのバージョンアップ時の移行方法について説明します。

移行方法として，既存のワーク管理システムを引き継ぐ方法と，ワーク管理システムを新規に構築する方法があります。どちらの方法もワーク管理データベースに保持するデータ（案件，ビジネスプロセス定義，環境設定）を引き継ぎます。

9.1　バージョンアップの流れ

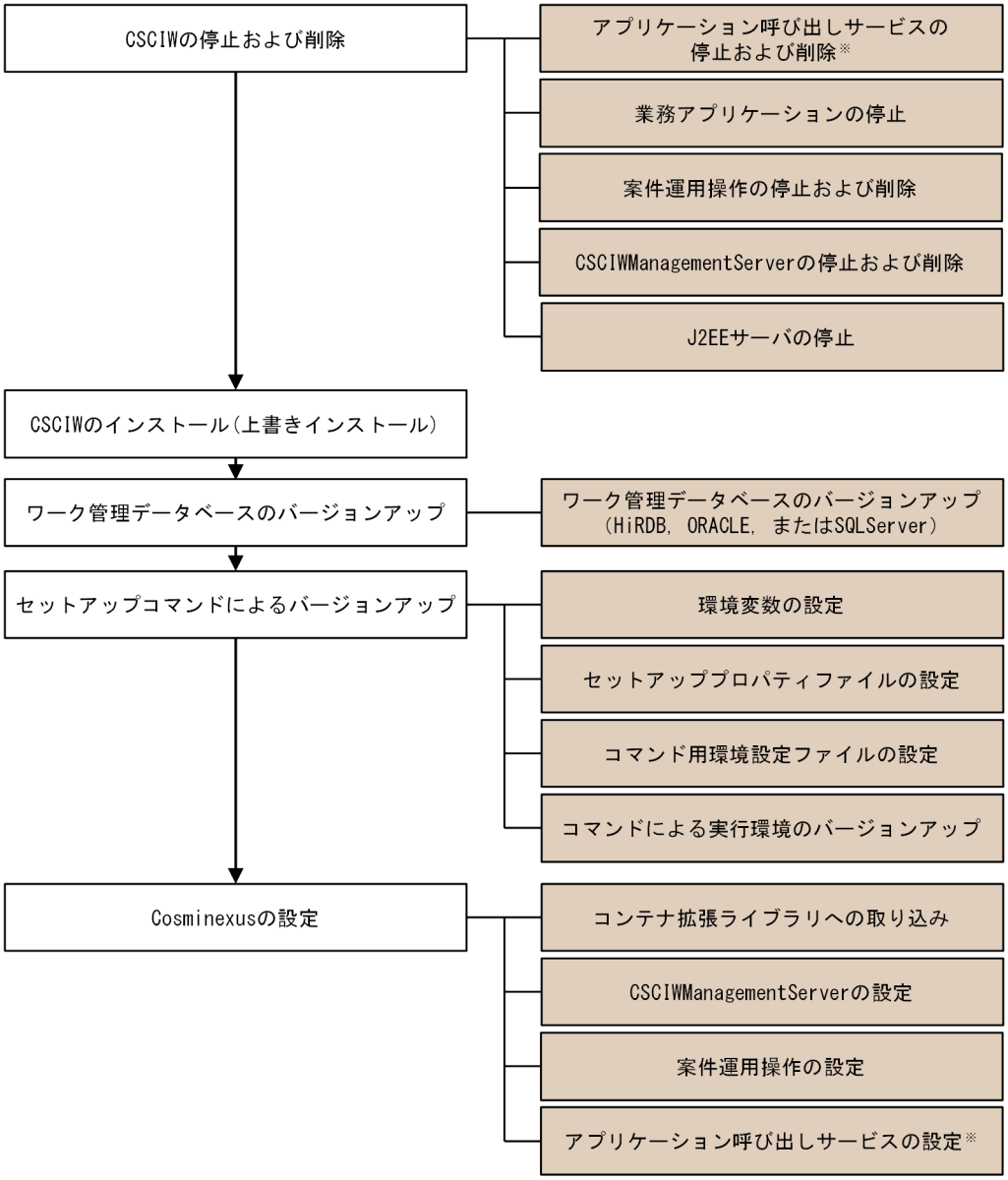
ここでは，ワーク管理システムをバージョンアップする流れについて説明します。

なお，ワーク管理システムをバージョンアップする前に，必要に応じてファイルやデータベースのバックアップを取得してください。バックアップの方法については，「7.6　バックアップとリストア」を参照してください。

ワーク管理システムをバージョンアップするには，既存のワーク管理システムを引き継ぐ場合とワーク管理システムを新規に構築する場合があります。

既存のワーク管理システムを引き継ぐ場合のワーク管理システムのバージョンの流れを次の図に示します。

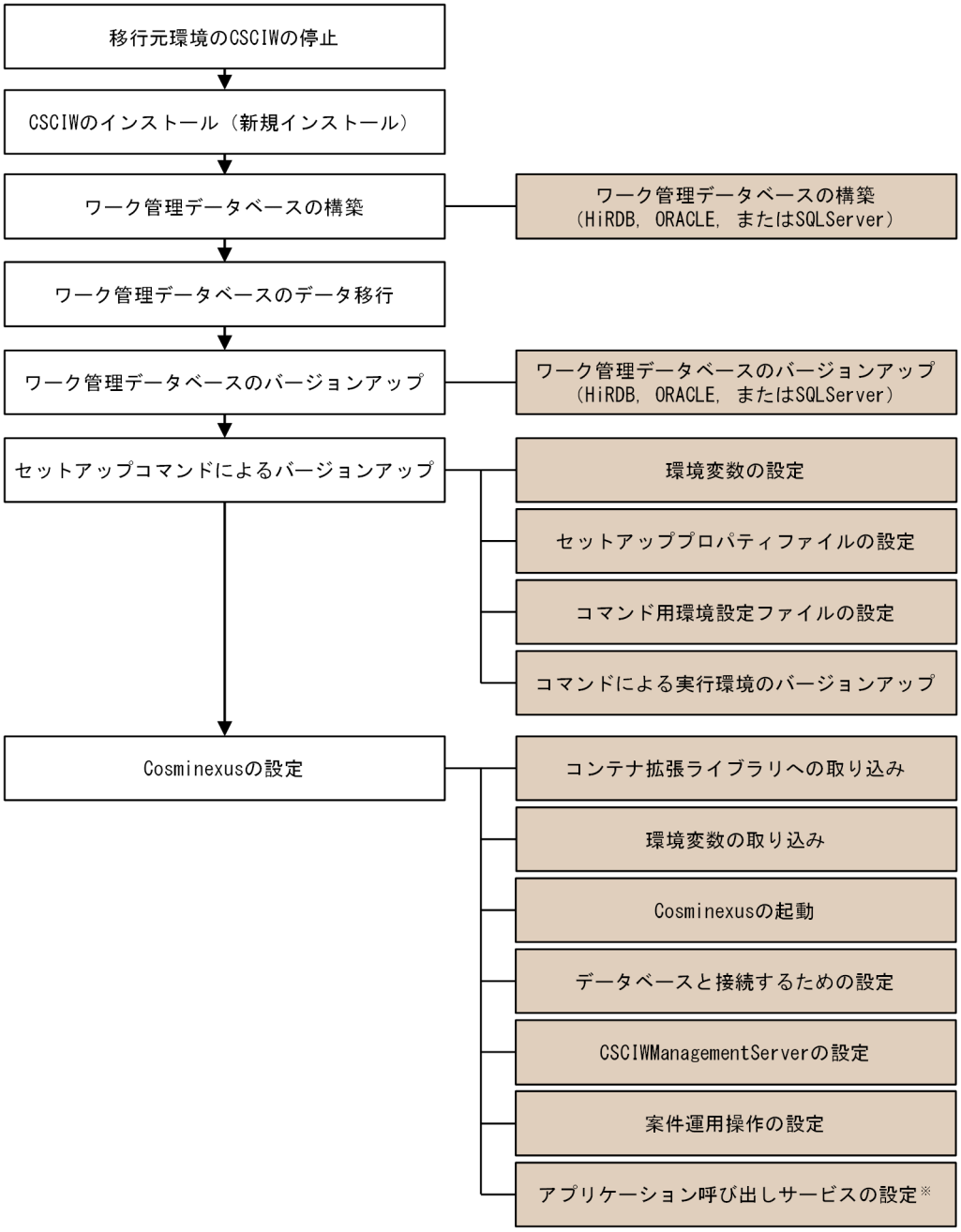
図 9-1　ワーク管理システムのバージョンアップの流れ（既存のワーク管理システムを引き継ぐ場合）



注※　アプリケーション呼び出しサービスを使用している場合だけ再設定します。

ワーク管理システムを新規に構築する場合のワーク管理システムのバージョンの流れを次の図に示します。

図 9-2　ワーク管理システムのバージョンアップの流れ（ワーク管理システムを新規に構築する場合）



注※　アプリケーション呼び出しサービスを使用している場合だけ再設定します。

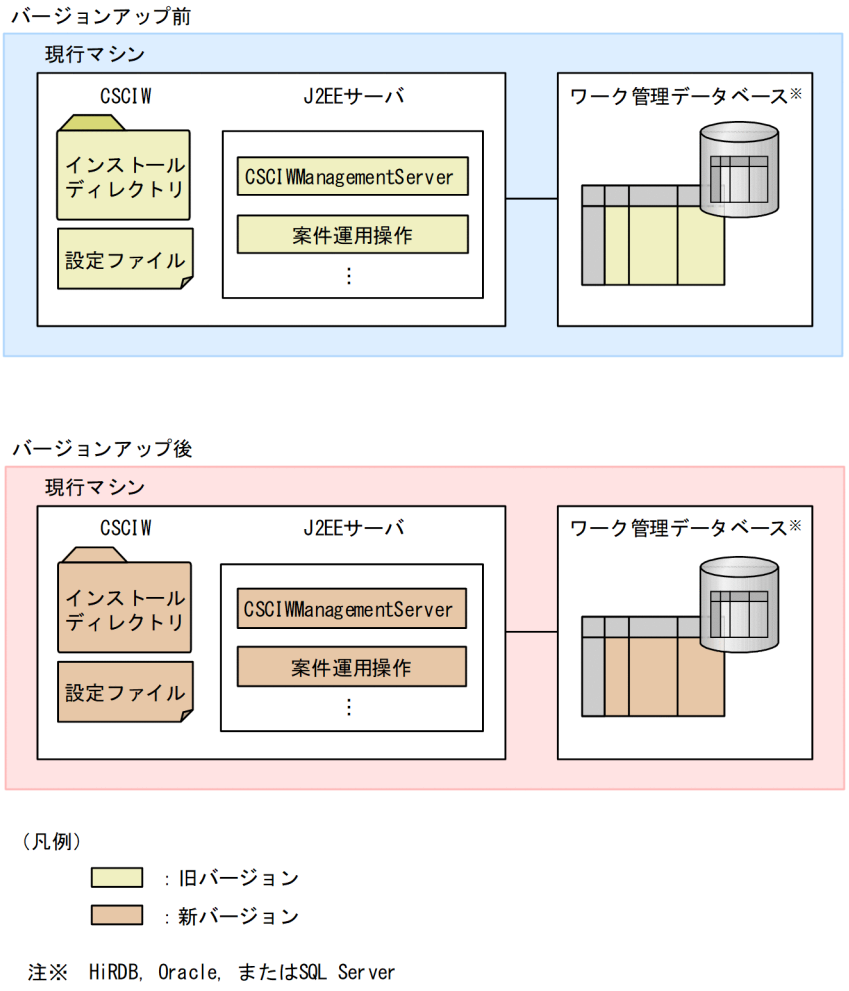
既存のワーク管理システムを引き継ぐ場合は「9.2　ワーク管理システムのバージョンアップ（既存のワーク管理システムを引き継ぐ場合）」を，ワーク管理システムを新規に構築する場合は「9.3　ワーク管理システムのバージョンアップ（ワーク管理システムを新規に構築する場合）」を参照してください。

9.2　ワーク管理システムのバージョンアップ（既存のワーク管理システムを引き継ぐ場合）

ここでは，既存のワーク管理システムを引き継ぐ場合のバージョンアップ方法について説明します。

既存のワーク管理システムを引き継ぐ場合の概念図を次に示します。

図 9-3　ワーク管理システムのバージョンアップ（既存のワーク管理システムを引き継ぐ場合）



9.2.1　CSCIW の停止および削除

CSCIW の停止および削除について説明します。

(1)　アプリケーション呼び出しサービスの停止および削除

アプリケーション呼び出しサービスを使用している場合は，アプリケーション呼び出しサービスを停止および削除します。詳細については，「5.2.1(3)　アプリケーション呼び出しサービスの停止および削除」を参照してください。

(2)　業務アプリケーションの停止

CSCIW を使用している Java アプリケーションを停止します。

また，J2EE サーバ上で実行している J2EE アプリケーションも停止します。J2EE アプリケーションは(a)または(b)の手順を実行して停止します。

(a)　運用管理ポータルを使用する場合

次の手順で操作してください。

1. 運用管理ポータルにログインする
2. 運用管理ポータルで［論理サーバの環境設定］アンカーをクリックする
3. ツリーペインで次の順にクリックする
   ［<操作対象の運用管理ドメイン名>］－［論理 J2EE サーバ］－［J2EE サーバ］－［<操作対象の J2EE サーバ名>］－［アプリケーション］
4. 右ペインの［開始／停止］タブをクリックし，［J2EE アプリケーションの開始／停止］画面を表示する
5. 業務アプリケーションの［停止］アンカーをクリックする
6. 内容を確認して，［はい］ボタンをクリックする

(b)　サーバ管理コマンドを使用する場合

cjstopapp コマンドを使用して，J2EE サーバ上で実行している J2EE アプリケーションを停止します。

J2EE アプリケーションの停止方法については，マニュアル「Cosminexus 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」の「J2EE アプリケーションの停止」の説明を参照してください。

cjstopapp コマンドの実行例を次に示します。

```
cjstopapp 〈サーバ名称〉 -name 〈業務アプリケーション名〉
```

(3)　案件運用操作の停止および削除

案件運用操作を停止および削除します。詳細については，「5.2.1(2)　案件運用操作の停止および削除」を参照してください。

(4)　CSCIWManagementServer の停止および削除

CSCIWManagementServer を停止および削除します。詳細については，「5.2.1(4) CSCIWManagementServer の停止および削除」を参照してください。

(5)　J2EE サーバの停止

CSCIW を使用している J2EE サーバを停止します。詳細については，「5.2.1(5)　J2EE サーバの停止」を参照してください。

9.2.2　CSCIW のインストール（上書きインストール）

最新の CSCIW をインストール（上書きインストール）します。

CSCIW インストールについては，「4.2.2　CSCIW のインストール」を参照してください。

> **メモ**
>
> バージョンアップ後，Cosminexus J2EE サーバを V9 互換モードで使用する場合，CSCIW を上書きインストールしたあとに「付録 J.1　互換モード用のファイルへの入れ替え」を参照して，ファイルを入れ替えてください。

9.2.3　ワーク管理データベースのバージョンアップ

ここでは，ワーク管理データベースをバージョンアップする方法について説明します。

なお，マルチマシン構成の場合，同一システム ID によって CSCIW を運用しているため，すべてのマシンでワーク管理データベースの移行を実施する必要はありません。どれか 1 台のマシンで実施してください。

また，ワーク管理データベースを移行する前にアクセス権限を付与していた場合，アクセス権限を付与し直す必要があります。アクセス権限の付与については，「4.3　ワーク管理データベースの構築」を参照してください。

SQL スクリプトファイルはバージョンごとに段階的に実行する必要があります。例えば，CSCIW の 01-50 から 03-20 へバージョンアップする場合，次の順序で実行します。

1. 01-50 から 01-60 へ移行する SQL スクリプトファイルを編集および実行
2. 01-60 から 01-70 へ移行する SQL スクリプトファイルを編集および実行
3. 01-70 から 01-90 へ移行する SQL スクリプトファイルを編集および実行
4. 01-90 から 01-95 へ移行する SQL スクリプトファイルを編集および実行
5. 01-95 から 02-00 へ移行する SQL スクリプトファイルを編集および実行
6. 02-00 から 02-10 へ移行する SQL スクリプトファイルを編集および実行
7. 02-10 から 02-20 へ移行する SQL スクリプトファイルを編集および実行
8. 02-20 から 02-30 へ移行する SQL スクリプトファイルを編集および実行

9. 02-30 から 03-00 へ移行する SQL スクリプトファイルを編集および実行
10. 03-00 から 03-10 へ移行する SQL スクリプトファイルを編集および実行
11. 03-10 から 03-11 へ移行する SQL スクリプトファイルを編集および実行
12. 03-11 から 03-20 へ移行する SQL スクリプトファイルを編集および実行

(1) ワーク管理データベースのバージョンアップ（HiRDB の場合）

ワーク管理データベースをバージョンアップする場合，テーブルやインデクスを追加および更新する必要があります。テーブルやインデクスは，SQL スクリプトファイルを使用して追加および更新してください。

(a) SQL スクリプトファイルの編集

使用しているワーク管理データベースが HiRDB の場合，次に示す SQL スクリプトファイルを編集します。

●01-50 から 01-60 へ移行する場合

```
<CSCIWインストールディレクトリ>/sql/vup_0150to0160_hirdb.sql
```

●01-60 から 01-70 へ移行する場合

```
<CSCIWインストールディレクトリ>/sql/vup_0160to0170_hirdb.sql
```

●01-70 から 01-90 へ移行する場合

```
<CSCIWインストールディレクトリ>/sql/vup_0180to0190_hirdb.sql
```

●01-90 から 01-95 へ移行する場合

```
<CSCIWインストールディレクトリ>/sql/vup_0190to0195_hirdb.sql
```

●01-95 から 02-00 へ移行する場合

```
<CSCIWインストールディレクトリ>/sql/vup_0195to0200_hirdb.sql
```

●02-00 から 02-10 へ移行する場合

```
<CSCIWインストールディレクトリ>/sql/vup_0200to0210_hirdb.sql
```

●02-10 から 02-20 へ移行する場合

```
<CSCIWインストールディレクトリ>/sql/vup_0210to0220_hirdb.sql
```

●02-20 から 02-30 へ移行する場合

```
<CSCIWインストールディレクトリ>/sql/vup_0220to0230_hirdb.sql
```

●02-30 から 03-00 へ移行する場合

```
<CSCIWインストールディレクトリ>/sql/vup_0230to0300_hirdb.sql
```

●03-00 から 03-10 へ移行する場合

```
<CSCIWインストールディレクトリ>/sql/vup_0300to0310_hirdb.sql
<CSCIWインストールディレクトリ>/sql/vupex_0300to0310_hirdb.sql※
```

●03-10 から 03-11 へ移行する場合

```
<CSCIWインストールディレクトリ>/sql/vup_0310to0311_hirdb.sql
```

●03-11 から 03-20 へ移行する場合

```
<CSCIWインストールディレクトリ>/sql/vup_0311to0320_hirdb.sql
```

注※

アプリケーション呼び出しサービスを使用する場合だけ編集してください。

SQL スクリプトファイル中の文字列を，テキストエディタを使用してワーク管理データベースへアクセスできるように書き換えてください。テーブルやインデクスの追加および更新をする際に SQL スクリプトファイルの書き換えが必要な文字列，および書き換える内容を次に示します。

表 9-1　テーブルやインデクスの追加および更新時に書き換えが必要な SQL スクリプトファイル中の文字列（HiRDB の場合）

書き換えが必要な文字列	書き換える内容
<SYSTEMID>	システム ID ワーク管理データベースを一意に識別するためのシステム ID に置換してください。システム ID に指定できる文字は，半角英大文字で始まる半角英大文字または半角数字で，5 文字以内です。
<RDDATA>	テーブルを格納する RD エリア名 ワーク管理データベースのテーブルを格納する RD エリア名に置換してください。
<RDINDEX>	インデクスを格納する RD エリア名 ワーク管理データベースのインデクスを格納する RD エリア名に置換してください。

なお，テーブルごとに格納先の RD エリアを変えて指定するなど，格納する RD エリアをカスタマイズできます。RD エリアの指定方法については，マニュアル「HiRDB SQL リファレンス」を参照してください。

(b)　SQL スクリプトファイルの実行

編集した SQL スクリプトファイルを，HiRDB SQL Executer を使用して実行します。

pdsql コマンドに編集した SQL スクリプトファイルを指定し，実行することで，テーブルやインデクスを追加および更新できます。HiRDB SQL Executer の使用方法については，マニュアル「HiRDB 解説」を参照してください。

pdsql コマンドの指定例を次に示します。

```
pdsql -u <接続認可識別子>/<パスワード>
      -h <HiRDBサーバのホスト名またはIPアドレス>
```

```
-n <HiRDBサーバのポート番号>
 < <編集したSQLスクリプトファイルのパス>
```

注

接続認可識別子には，ワーク管理データベースが作成されている認可識別子を指定してください。

(2) ワーク管理データベースのバージョンアップ（ORACLE の場合）

ワーク管理データベースをバージョンアップする場合，テーブルやインデクスを追加および更新する必要があります。テーブルやインデクスは，SQL スクリプトファイルを使用して追加および更新してください。

(a) SQL スクリプトファイルの編集

使用しているワーク管理データベースが ORACLE の場合，次に示す SQL スクリプトファイルを編集します。

●01-50 から 01-60 へ移行する場合

```
<CSCIWインストールディレクトリ>/sql/vup_0150to0160_oracle.sql
```

●01-60 から 01-70 へ移行する場合

```
<CSCIWインストールディレクトリ>/sql/vup_0160to0170_oracle.sql
```

●01-70 から 01-90 へ移行する場合

```
<CSCIWインストールディレクトリ>/sql/vup_0180to0190_oracle.sql
```

●01-90 から 01-95 へ移行する場合

```
<CSCIWインストールディレクトリ>/sql/vup_0190to0195_oracle.sql
```

●01-95 から 02-00 へ移行する場合

```
<CSCIWインストールディレクトリ>/sql/vup_0195to0200_oracle.sql
```

●02-00 から 02-10 へ移行する場合

```
<CSCIWインストールディレクトリ>/sql/vup_0200to0210_oracle.sql
```

●02-10 から 02-20 へ移行する場合

```
<CSCIWインストールディレクトリ>/sql/vup_0210to0220_oracle.sql
```

●02-20 から 02-30 へ移行する場合

```
<CSCIWインストールディレクトリ>/sql/vup_0220to0230_oracle.sql
```

●02-30 から 03-00 へ移行する場合

```
<CSCIWインストールディレクトリ>/sql/vup_0230to0300_oracle.sql
```

●03-00 から 03-10 へ移行する場合

```
<CSCIWインストールディレクトリ>/sql/vup_0300to0310_oracle.sql
<CSCIWインストールディレクトリ>/sql/vupex_0300to0310_oracle.sql※
```

●03-10 から 03-11 へ移行する場合

```
<CSCIWインストールディレクトリ>/sql/vup_0310to0311_oracle.sql
```

●03-11 から 03-20 へ移行する場合

```
<CSCIWインストールディレクトリ>/sql/vup_0311to0320_oracle.sql
```

注※

アプリケーション呼び出しサービスを使用する場合だけ編集してください。

SQL スクリプトファイル中の文字列を，テキストエディタを使用してワーク管理データベースへアクセスできるように書き換えてください。テーブルやインデクスの追加および更新をする際に SQL スクリプトファイルの書き換えが必要な文字列，および書き換える内容を次に示します。

表 9–2　テーブルやインデクスの追加および更新時に書き換えが必要な SQL スクリプトファイル中の文字列（ORACLE の場合）

書き換えが必要な文字列	書き換える内容
<SYSTEMID>	システム ID ワーク管理データベースを一意に識別するためのシステム ID に置換してください。システム ID に指定できる文字は，半角英大文字で始まる半角英大文字または半角数字で，5 文字以内です。
<DATASPACE>	テーブルを格納する表領域名 ワーク管理データベースのテーブルを格納する表領域名に置換してください。
<INDEXSPACE>	インデクスを格納する表領域名 ワーク管理データベースのインデクスを格納する表領域名に置換してください。

なお，テーブルごとに格納先の表領域を変えて指定するなど，格納する表領域をカスタマイズできます。表領域の指定方法については，ORACLE のマニュアルを参照してください。

(b)　SQL スクリプトファイルの実行

編集した SQL スクリプトファイルを，SQL*Plus を使用して実行します。

sqlplus コマンドに編集した SQL スクリプトファイルを指定し，実行することで，テーブルやインデクスを追加および更新できます。SQL*Plus の使用方法については，ORACLE のマニュアルを参照してください。

sqlplus コマンドの指定例を次に示します。

```
sqlplus  <接続ユーザ名>/<パスワード>@<Oracle Net 接続識別子>
         @<編集したSQLスクリプトファイルの絶対パス>
```

注

接続ユーザ名には，ワーク管理データベースが作成されているユーザを指定してください。

(3) ワーク管理データベースのバージョンアップ（SQL Server の場合）

ワーク管理データベースをバージョンアップする場合，テーブルやインデクスを追加および更新する必要があります。テーブルやインデクスは，SQL スクリプトファイルを使用して追加および更新してください。

(a) SQL スクリプトファイルの編集

使用しているワーク管理データベースが SQL Server の場合，次に示す SQL スクリプトファイルを編集します。

●01-60 から 01-70 へ移行する場合

```
<CSCIWインストールディレクトリ>/sql/vup_0160to0170_sqlserver.sql
```

●01-70 から 01-90 へ移行する場合

```
<CSCIWインストールディレクトリ>/sql/vup_0180to0190_sqlserver.sql
```

●01-90 から 01-95 へ移行する場合

```
<CSCIWインストールディレクトリ>/sql/vup_0190to0195_sqlserver.sql
```

●01-95 から 02-00 へ移行する場合

```
<CSCIWインストールディレクトリ>/sql/vup_0195to0200_sqlserver.sql
```

●02-00 から 02-10 へ移行する場合

```
<CSCIWインストールディレクトリ>/sql/vup_0200to0210_sqlserver.sql
```

●02-10 から 02-20 へ移行する場合

```
<CSCIWインストールディレクトリ>/sql/vup_0210to0220_sqlserver.sql
```

●02-20 から 02-30 へ移行する場合

```
<CSCIWインストールディレクトリ>/sql/vup_0220to0230_sqlserver.sql
```

●02-30 から 03-00 へ移行する場合

```
<CSCIWインストールディレクトリ>/sql/vup_0230to0300_sqlserver.sql
```

●03-00 から 03-10 へ移行する場合

```
<CSCIWインストールディレクトリ>/sql/vup_0300to0310_sqlserver.sql
```

●03-10 から 03-11 へ移行する場合

```
<CSCIWインストールディレクトリ>/sql/vup_0310to0311_sqlserver.sql
```

●03-11 から 03-20 へ移行する場合

```
<CSCIWインストールディレクトリ>/sql/vup_0311to0320_sqlserver.sql
```

SQL スクリプトファイル中の文字列を，テキストエディタを使用してワーク管理データベースへアクセスできるように書き換えてください。テーブルやインデクスの追加および更新をする際に SQL スクリプトファイルの書き換えが必要な文字列，および書き換える内容を次に示します。

表 9-3　テーブルやインデクスの追加および更新時に書き換えが必要な SQL スクリプトファイル中の文字列（SQL Server の場合）

書き換えが必要な文字列	書き換える内容
<SYSTEMID>	システム ID ワーク管理データベースを一意に識別するためのシステム ID に置換してください。システム ID に指定できる文字は，半角英大文字で始まる半角英大文字または半角数字で，5 文字以内です。
<FILEGROUP>	テーブルを格納するファイルグループ ワーク管理データベースのテーブルを格納するファイルグループに置換してください。
<INDEXFILEGROUP>	インデクスを格納するファイルグループ ワーク管理データベースのインデクスを格納するファイルグループに置換してください。

なお，テーブルごとに格納先のファイルグループを変えて指定するなど，格納するファイルグループをカスタマイズできます。ファイルグループの指定方法については，SQL Server のマニュアルを参照してください。

(b)　SQL スクリプトファイルの実行

編集した SQL スクリプトファイルを，sqlcmd ユティリティを使用して実行します。

sqlcmd ユティリティに編集した SQL スクリプトファイルを指定し，実行することで，接続するユーザのスキーマにテーブルを作成できます。sqlcmd ユティリティの使用方法については，SQL Server のマニュアルを参照してください。

sqlcmd ユティリティの指定例を次に示します。

```
sqlcmd -U <接続ユーザ名>
       -P <パスワード>
       -i <編集したSQLスクリプトファイルの絶対パス>
```

注

接続ユーザ名には，次に示す権限がすべて与えられているユーザを指定してください。

- CONNECT 権限
- CREATE TABLE 権限
- CREATE VIEW 権限

- CONTROL 権限

9.2.4　セットアップコマンドによるバージョンアップ

(1)　環境変数の設定

環境変数は，バージョンアップ前と同じ値を使用します。

環境変数の設定については，「4.5.2(1)　環境変数の設定」を参照してください。

(2)　セットアッププロパティファイルの設定

セットアッププロパティファイルは，バージョンアップ前と同じ値を使用します。

セットアッププロパティファイルの設定内容については，「付録 A.2　セットアッププロパティファイル」を参照してください。

(3)　コマンド用環境設定ファイルの設定

コマンド用環境設定ファイルは，バージョンアップ前と同じ値を使用します。

コマンド用環境設定ファイルの設定内容については，「付録 A.3　コマンド用環境設定ファイル」を参照してください。

(4)　コマンドによる実行環境のバージョンアップ

CSCIW の実行環境は，ciwsetenv（環境の構築または削除）コマンドを実行してバージョンアップします。なお，ciwsetenv（環境の構築または削除）コマンドは，すべてのマシンで実行してください。

次に示す形式でコマンドを実行すると，システム設定プロパティファイルの内容は，指定した環境構築ファイルの内容に書き換えられます。なお，環境構築ファイルに設定する項目「SystemDBPassword」および「SystemDBURL」については，同じワーク管理データベースに接続できる範囲で，値を変更できます。

```
ciwsetenv -sid 〈システムID〉-vup 〈環境構築ファイル名〉
```

ciwsetenv（環境の構築または削除）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。

9.2.5　Cosminexus の設定

(1)　コンテナ拡張ライブラリへの取り込み

コンテナ拡張ライブラリに jar を追加します。詳細については，「4.5.3(1)　コンテナ拡張ライブラリへの取り込み」を参照してください。

> **メモ**
>
> バージョンアップ後，Cosminexus J2EE サーバを V9 互換モードで使用する場合，「付録 J.2 コンテナ拡張ライブラリに取り込むライブラリの変更」を参照して，コンテナ拡張ライブラリを設定してください。

(2)　Cosminexus の起動

Cosminexus を起動します。詳細については，「4.5.3(3)　Cosminexus の起動」を参照してください。

(3)　CSCIWManagementServer の再設定

CSCIWManagementServer を再設定します。CSCIWManagementServer の設定手順については，「4.5.3(5)　CSCIWManagementServer の設定」を参照してください。

なお，「4.5.3(5)(e)　コンテキストルートの設定」を参照して，CSCIWManagementServer へのリダイレクトを必ず設定してください。CSCIWManagementServer へのリダイレクトができないと，ビジネスプロセス管理ツール（CSCIW-Definer）から接続できません。

(4)　案件運用操作の再設定

案件運用操作を再設定します。案件運用操作の設定手順については，「4.5.3(6)　案件運用操作の設定」を参照してください。

(5)　アプリケーション呼び出しサービスの再設定

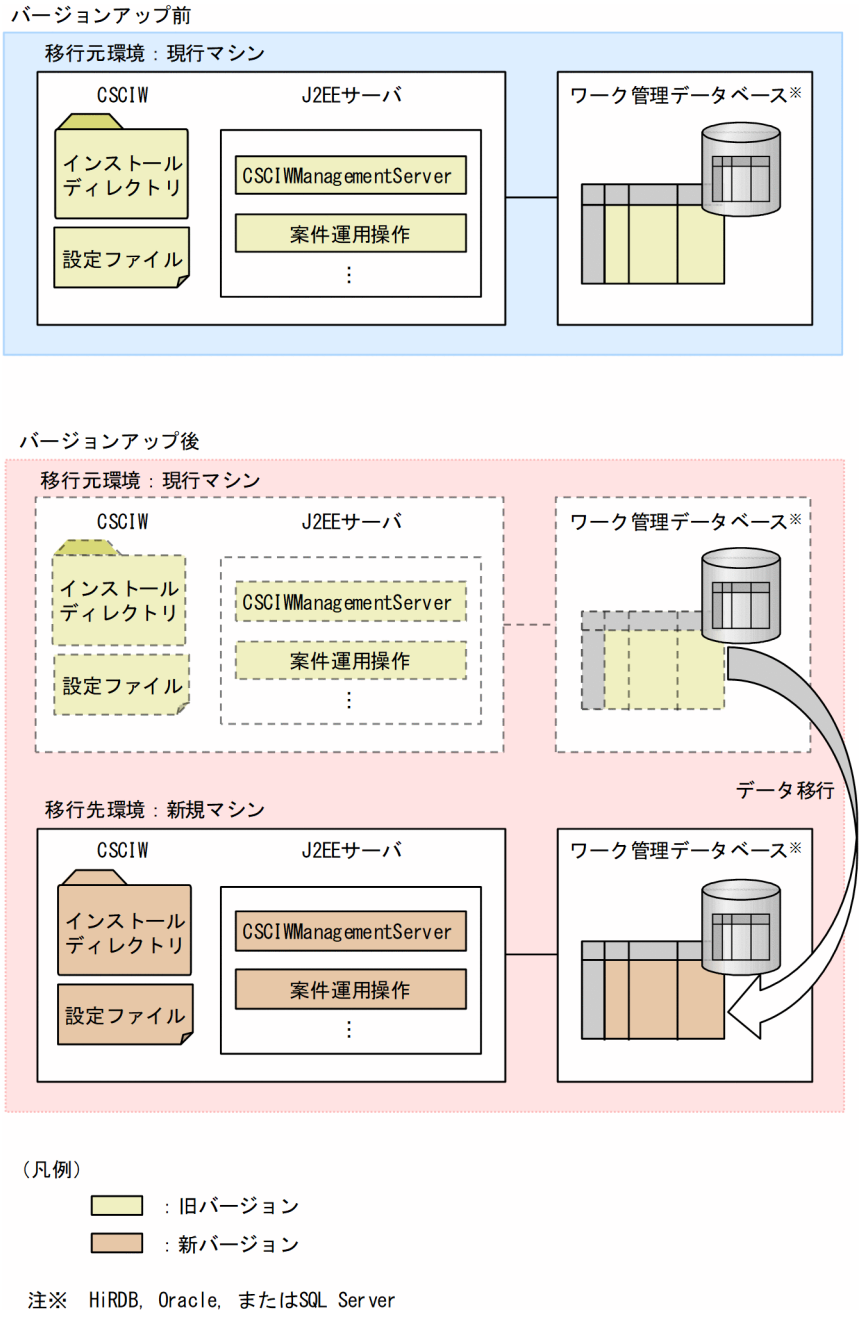
アプリケーション呼び出しサービスを使用している場合，アプリケーション呼び出しサービスを再設定します。アプリケーション呼び出しサービスの設定手順については，「4.5.3(7)　アプリケーション呼び出しサービスの設定」を参照してください。

9.3　ワーク管理システムのバージョンアップ（ワーク管理システムを新規に構築する場合）

ここでは，ワーク管理システムを新規に構築する場合のバージョンアップ方法について説明します。

移行先環境（新規マシン）に新しいバージョンの CSCIW をインストールして，ワーク管理システムを新規に構築する場合の概念図を次に示します。

図 9-4　ワーク管理システムのバージョンアップ（ワーク管理システムを新規に構築する場合）



9.3.1　移行元環境の CSCIW の停止

移行元環境の CSCIW を停止します。詳細については，「7.1.2　業務の停止」を参照してください。

CSCIW の停止後にciwcleanup コマンドを実行してください。そのあと，ciwlistsid コマンドで，各プロセスの状態がStopped であることを確認してください。

9.3.2　CSCIW のインストール（新規インストール）

最新の CSCIW をインストール（新規インストール）します。

CSCIW のインストールについては，「4.2.2　CSCIW のインストール」を参照してください。

メモ

- バージョンアップ後，Cosminexus J2EE サーバを V9 互換モードで使用する場合，CSCIW を新規インストールしたあとに「付録 J.1　互換モード用のファイルへの入れ替え」を参照して，ファイルを入れ替えてください。
- Windows の場合，移行元環境と同じフォルダにインストールしてください。

(1)　移行元環境から移行先環境へファイルを移行

次のファイルを移行元環境から取得し，移行先環境に格納してください。格納先は移行元環境と同じディレクトリにしてください。また，必要に応じて，ファイルの内容を変更してください。

1. セットアッププロパティファイル
2. コマンド用環境設定ファイル
   JDBC のクラスパスが変更になっている場合は，コマンド用環境設定ファイルのCSCIWCMD_JVM_CLPATH の設定を変更してください。
3. 共通設定ファイル※
4. アプリケーション呼び出し情報ファイル※
   アプリケーション呼び出しサービスが呼び出す REST アプリケーションの URL に変更がある場合は，rest.request.url プロパティに指定した URL を変更してください。
5. REST アプリケーション呼び出し用ヘッダファイル※
6. REST アプリケーション呼び出しスキーマ変換用スタイルシート※

注※
アプリケーション呼び出しサービスを使用する場合だけ移行してください。

9.3.3　ワーク管理データベースの構築

移行先環境へ移行元環境と同じバージョンのワーク管理データベースを作成します。

(1)　ワーク管理データベースの環境設定（HiRDB の場合）

ワーク管理データベースのテーブルやインデクスを HiRDB に作成します。移行元環境と同じバージョンのものを作成してください。詳細については，「4.3.2　ワーク管理データベースの環境設定（HiRDB の場合）」を参照してください。

なお，ワーク管理データベースのテーブル定義，およびインデクスの移行には，DBMS の機能も使用できます。また，テーブル定義，インデクスと同時にテーブルのデータも移行する場合は，「9.3.4　ワーク管理データベースのデータ移行」の手順を実施する必要はありません。

(2)　ワーク管理データベースの環境設定（ORACLE の場合）

ワーク管理データベースのテーブルやインデクスを ORACLE に作成します。移行元環境と同じバージョンのものを作成してください。詳細については，「4.3.3　ワーク管理データベースの環境設定（ORACLE の場合）」を参照してください。

なお，ワーク管理データベースのテーブル定義，およびインデクスの移行には，DBMS の機能も使用できます。また，テーブル定義，インデクスと同時にテーブルのデータも移行する場合は，「9.3.4　ワーク管理データベースのデータ移行」の手順を実施する必要はありません。

(3)　ワーク管理データベースの環境設定（SQL Server の場合）

ワーク管理データベースのテーブルやインデクスを SQL Server に作成します。移行元環境と同じバージョンのものを作成してください。詳細については，「4.3.4　ワーク管理データベースの環境設定（SQL Server の場合）」を参照してください。

なお，ワーク管理データベースのテーブル定義，およびインデクスの移行には，DBMS の機能も使用できます。また，テーブル定義，インデクスと同時にテーブルのデータも移行する場合は，「9.3.4　ワーク管理データベースのデータ移行」の手順を実施する必要はありません。

9.3.4　ワーク管理データベースのデータ移行

移行元環境のワーク管理データベースのすべてのテーブルのデータを，移行先環境のワーク管理データベースへ移行します。データの移行については，各 DBMS のマニュアルを参照してください。

9.3.5　ワーク管理データベースのバージョンアップ

ここでは，ワーク管理データベースをバージョンアップする方法について説明します。

なお，マルチマシン構成の場合，同一システム ID によって CSCIW を運用しているため，すべてのマシンでワーク管理データベースの移行を実施する必要はありません。どれか 1 台のマシンで実施してください。

SQL スクリプトファイルはバージョンごとに段階的に実行する必要があります。例えば，CSCIW の 01-50 から 03-20 へバージョンアップする場合，次の順序で実行します。

1. 01-50 から 01-60 へ移行する SQL スクリプトファイルを編集および実行
2. 01-60 から 01-70 へ移行する SQL スクリプトファイルを編集および実行
3. 01-70 から 01-90 へ移行する SQL スクリプトファイルを編集および実行
4. 01-90 から 01-95 へ移行する SQL スクリプトファイルを編集および実行
5. 01-95 から 02-00 へ移行する SQL スクリプトファイルを編集および実行
6. 02-00 から 02-10 へ移行する SQL スクリプトファイルを編集および実行
7. 02-10 から 02-20 へ移行する SQL スクリプトファイルを編集および実行
8. 02-20 から 02-30 へ移行する SQL スクリプトファイルを編集および実行
9. 02-30 から 03-00 へ移行する SQL スクリプトファイルを編集および実行
10. 03-00 から 03-10 へ移行する SQL スクリプトファイルを編集および実行
11. 03-10 から 03-11 へ移行する SQL スクリプトファイルを編集および実行
12. 03-11 から 03-20 へ移行する SQL スクリプトファイルを編集および実行

(1)　ワーク管理データベースのバージョンアップ（HiRDB の場合）

ワーク管理データベースをバージョンアップする場合，テーブルやインデクスを追加および更新する必要があります。詳細については，「9.2.3(1)　ワーク管理データベースのバージョンアップ（HiRDB の場合）」を参照してください。

(2)　ワーク管理データベースのバージョンアップ（ORACLE の場合）

ワーク管理データベースをバージョンアップする場合，テーブルやインデクスを追加および更新する必要があります。詳細については，「9.2.3(2)　ワーク管理データベースのバージョンアップ（ORACLE の場合）」を参照してください。

(3)　ワーク管理データベースのバージョンアップ（SQL Server の場合）

ワーク管理データベースをバージョンアップする場合，テーブルやインデクスを追加および更新する必要があります。詳細については，「9.2.3(3)　ワーク管理データベースのバージョンアップ（SQL Server の場合）」を参照してください。

9.3.6　セットアップコマンドによるバージョンアップ

(1)　環境変数の設定

環境変数は，バージョンアップ前と同じ値を使用します。

環境変数の設定については，「4.5.2(1)　環境変数の設定」を参照してください。

(2)　セットアッププロパティファイルの設定

セットアッププロパティファイルは，バージョンアップ前と同じ値を使用します。

セットアッププロパティファイルの設定内容については，「付録 A.2　セットアッププロパティファイル」を参照してください。

(3)　コマンド用環境設定ファイルの設定

コマンド用環境設定ファイルは，バージョンアップ前と同じ値を使用します。

コマンド用環境設定ファイルの設定内容については，「付録 A.3　コマンド用環境設定ファイル」を参照してください。

(4)　コマンドによる実行環境のバージョンアップ

CSCIW の実行環境は，ciwsetenv（環境の構築または削除）コマンドを実行してバージョンアップします。なお，ciwsetenv（環境の構築または削除）コマンドは，すべてのマシンで実行してください。

次に示す形式でコマンドを実行すると，システム設定プロパティファイルの内容は，指定した環境構築ファイルの内容に書き換えられます。なお，環境構築ファイルに設定する項目「SystemDBPassword」および「SystemDBURL」については，同じワーク管理データベースに接続できる範囲で，値を変更できます。

```
ciwsetenv -sid 〈システムID〉-vup 〈環境構築ファイル名〉
```

ciwsetenv（環境の構築または削除）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。

9.3.7　Cosminexus の設定

(1)　コンテナ拡張ライブラリへの取り込み

コンテナ拡張ライブラリに JAR ファイルを取り込みます。詳細については，「4.5.3(1)　コンテナ拡張ライブラリへの取り込み」を参照してください。

> **メモ**
>
> Cosminexus J2EE サーバを V9 互換モードで使用する場合，「付録 J.2　コンテナ拡張ライブラリに取り込むライブラリの変更」を参照して，コンテナ拡張ライブラリを設定してください。

(2)　環境変数の取り込み

環境変数を取り込みます。詳細については，「4.5.3(2)　環境変数の取り込み」を参照してください。

(3)　Cosminexus の起動

Cosminexus を起動します。詳細については，「4.5.3(3)　Cosminexus の起動」を参照してください。

(4)　DB Connector の設定

データベースと接続するための DB Connector を設定します。詳細については，「4.5.3(4)　DB Connector の設定」を参照してください。

(5)　CSCIWManagementServer の設定

CSCIWManagementServer を J2EE アプリケーションとして設定します。詳細については，「4.5.3(5) CSCIWManagementServer の設定」を参照してください。

なお，「4.5.3(5)(e)　コンテキストルートの設定」を参照して，CSCIWManagementServer へのリダイレクトを必ず設定してください。CSCIWManagementServer へのリダイレクトができないと，ビジネスプロセス管理ツール（CSCIW-Definer）から接続できません。

(6)　案件運用操作の設定

案件運用操作を設定します。詳細については，「4.5.3(6)　案件運用操作の設定」を参照してください。

(7)　アプリケーション呼び出しサービスの設定

アプリケーション呼び出しサービスを設定します。詳細については，「4.5.3(7)　アプリケーション呼び出しサービスの設定」を参照してください。

以上で移行先環境のバージョンアップは完了です。

9.3.8　データ移行だけを繰り返し行う手順

移行先環境のバージョンアップが完了したあと，データ移行だけを繰り返し行うことができます。例えば，次のような場面で利用できます。

- 本番環境のバージョンアップ時に，あらかじめ移行先環境を構築しておき，本番直前はデータ移行だけ行う
- テスト環境のバージョンアップ時に，移行元環境のデータを使用してテストを繰り返し行う

> **メモ**
>
> データ移行だけを繰り返し行うには，移行元環境のバージョンが 01-60 以降である必要があります。

データ移行だけを繰り返し行う手順を次に示します。

1. CSCIW を停止する

   移行元環境および移行先環境の CSCIW を停止します。

2. 移行先環境のテーブルのレコードを削除する

   このとき，移行先環境の<SYSTEMID>_USER_SETTING_INFO テーブルのデータは削除しないでください。

3. テーブルのデータを移行する

   <SYSTEMID>_USER_SETTING_INFO 以外のテーブルのデータを移行します。

> **ヒント**
>
> <SYSTEMID>_USER_SETTING_INFO テーブルを対象外にすることで，システム共通環境情報（ciwchgenvコマンドで設定した内容）は，移行元環境から移行先環境に反映されません。
>
> それ以外のすべてのデータ（ビジネスプロセス定義や案件）は反映されます。

付録

付録 A　環境設定ファイル

ここでは，環境設定ファイルを示します。

CSCIW の環境設定ファイルは次の 3 種類です。

- **システム設定プロパティファイル**
  ライブラリおよびコマンドを実行するために必要な DB 情報や共通情報を管理するファイルです。
- **セットアッププロパティファイル**
  ライブラリおよびコマンドを実行するために必要な情報をシステム ID 単位で管理するファイルです。
- **コマンド用環境設定ファイル**
  コマンドを実行するために必要な環境変数をシステム ID 単位で管理するファイルです。

環境設定ファイルの内容を次に示します。

付録 A.1　システム設定プロパティファイル

システム設定プロパティファイルの内容は，ワーク管理データベースへ接続するための認証情報として使用します。また，上書きインストール時に，システム設定プロパティファイルは上書きされません。

ファイルに情報を追加したり，内容を削除したりするには，ciwsetenv（環境の構築または削除）コマンドを使用します。ただし，ファイルの内容を直接更新しないでください。ciwsetenv（環境の構築または削除）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。

システム設定プロパティファイルの格納パスは，次のとおりです。

- Windows の場合

```
%CSCIW_HOME%¥conf¥csciwsystem.properties
```

- UNIX の場合

```
${CSCIW_HOME}/conf/csciwsystem.properties
```

> **重要**
>
> システム設定プロパティファイルは，アンインストール時に削除されるため，アンインストール前に必要に応じてファイルのバックアップを取得してください。

システム設定プロパティファイルの設定内容を次に示します。なお，設定内容の登録順序は不定のため，参照したい項目については，必要に応じて検索をしてください。

表 A-1　システム設定プロパティファイルの設定内容

キー	内容
<SYSTEMID>.SystemDBType	ワーク管理データベースの種別 ワーク管理データベースの種別が設定されます。 • 種別が HiRDB の場合 HiRDB を設定 • 種別が ORACLE の場合 Oracle を設定 • 種別が SQL Server の場合 SQL Server を設定
<SYSTEMID>.SystemDBURL	DB 接続情報 URL ワーク管理データベースへ接続するための URL が設定されます。
<SYSTEMID>.SystemDBDriverName	JDBC のドライバ名 JDBC のドライバ名が設定されます。
<SYSTEMID>.SystemDBUser	DB ユーザ名 ワーク管理データベースへ接続するための認可識別子またはユーザ名が設定されます。
<SYSTEMID>.SystemDBPasswd	DB パスワード ワーク管理データベースへ接続するためのパスワードが設定されます。
<SYSTEMID>.SetupVersion	バージョン セットアップ時の CSCIW のバージョンが設定されます。
<SYSTEMID>.SystemDBSchemaName	DB のスキーマ名 ワーク管理データベースのスキーマ名が設定されます。

注

SYSTEMID は，ワーク管理データベースを一意に識別するためのシステム ID に置き換えてください。

付録 A.2　セットアッププロパティファイル

セットアッププロパティファイルは，設定内容をユーザの環境に合わせて直接変更できます。また，上書きインストール時に，セットアッププロパティファイルは上書きされません。

セットアッププロパティファイルの格納パスは，次のとおりです。

(1)　環境変数 CSCIW_CONF_DIR が設定されていない場合

- Windows の場合

```
%CSCIW_HOME%¥conf¥csciwsetup.properties
```

- UNIX の場合

```
${CSCIW_HOME}/conf/csciwsetup.properties
```

(2) 環境変数 CSCIW_CONF_DIR が設定されている場合

- Windows の場合

```
%CSCIW_CONF_DIR%¥csciwsetup.properties
```

- UNIX の場合

```
${CSCIW_CONF_DIR}/csciwsetup.properties
```

> **重要**
>
> アンインストール時に<CSCIW インストールフォルダ>/conf の下のセットアッププロパティファイルは削除されるため，アンインストール前に必要に応じてファイルのバックアップを取得してください。
>
> 環境変数 CSCIW_CONF_DIR 下に作成されたセットアッププロパティファイルはアンインストール時には削除されません。アンインストール後に残ったセットアッププロパティファイルの処理（バックアップの取得や削除）については，必要に応じて実行してください。

セットアッププロパティファイルの設定内容を次に示します。

なお，セットアッププロパティファイルはプロパティファイル形式のため，Java のプロパティファイルの仕様に従います。値に「¥」を記述する場合，「¥」の前に「¥」をエスケープ文字として記述してください。また，ファイルのエンコーディングは UTF-8 とし，BOM を付けないで保存してください。

CSCIW 02-20 以前で使用していたセットアッププロパティファイル（Unicode 形式のファイルを含む）を，そのまま使用することもできます。

表 A-2 セットアッププロパティファイルの設定内容

キー※1	内容	設定
SystemID	システム ID ワーク管理データベースを一意に識別する値で，SQL スクリプトファイルの<SYSTEMID>に指定した値を指定します。 指定できる値は，半角英大文字で始まる半角英大文字および半角数字で，5 文字以内です。	必須
DebugSQLTrace	SQL トレースファイルへの出力条件 SQL のトレースファイルへの出力条件を 0～2 の整数で指定します。 デフォルト値は 0 です。 • 出力しない場合	任意

キー※1	内容	設定
DebugSQLTrace	0 を設定 • SQL 文だけ出力する場合 1 を指定 • SQL 文と？パラメタ値を出力する場合 2 を指定	任意
HiRDBTransientSQLCode	SQL 例外となる SQLCODE の値 HiRDB 使用時の一時的な SQL 例外にしたい SQLCODE の値を指定します。複数の場合，値をコンマで区切って指定します。 デフォルト値は空文字列です。	任意
OracleTransientSQLCode	SQL 例外となる SQLCODE の値 Oracle 使用時の一時的な SQL 例外にしたい SQLCODE の値を指定します。複数の場合，値をコンマで区切って指定します。 デフォルト値は空文字列です。	任意
SQLServerTransientSQLCode	SQL 例外となる SQLCODE の値 SQL Server 使用時の一時的な SQL 例外にしたい SQLCODE の値を指定します。複数の場合，値をコンマで区切って指定します。 デフォルト値は空文字列です。	任意
HiRDBTransientSQLState	SQL 例外となる SQLSTATE の値 HiRDB 使用時の一時的な SQL 例外にしたい SQLSTATE の値を指定します。複数の場合，値をコンマで区切って指定します。 デフォルト値は空文字列です。	任意
OracleTransientSQLState	SQL 例外となる SQLSTATE の値 Oracle 使用時の一時的な SQL 例外にしたい SQLSTATE の値を指定します。複数の場合，値をコンマで区切って指定します。 デフォルト値は空文字列です。	任意
SQLServerTransientSQLState	SQL 例外となる SQLSTATE の値 SQL Server 使用時の一時的な SQL 例外にしたい SQLSTATE の値を指定します。複数の場合，値をコンマで区切って指定します。 デフォルト値は空文字列です。	任意
MsgFileDir	メッセージファイル出力先ディレクトリ メッセージファイルの出力先ディレクトリを指定します。 初期化メッセージファイルもこの設定に従います。 デフォルトの出力先ディレクトリは次のとおりです。 • Windows の場合 %CSCIW_HOME%¥log	任意

キー※1	内容	設定
MsgFileDir	• Unix の場合 ${CSCIW_HOME}/log	任意
TraceFileDir	トレースファイル出力先ディレクトリ トレースファイルの出力先ディレクトリを指定します。 初期化トレースファイルもこの設定に従います。 デフォルトの出力先ディレクトリは次のとおりです。 • Windows の場合 %CSCIW_HOME%¥log • Unix の場合 ${CSCIW_HOME}/log	任意
CmdMsgFileDir	コマンドメッセージファイル出力先ディレクトリ コマンドメッセージファイルの出力先ディレクトリを指定します。 デフォルトの出力先ディレクトリは次のとおりです。 • Windows の場合 %CSCIW_HOME%¥log • Unix の場合 ${CSCIW_HOME}/log	任意
CmdTraceFileDir	コマンドトレースファイル出力先ディレクトリ コマンドトレースファイルの出力先ディレクトリを指定します。 デフォルトの出力先ディレクトリは次のとおりです。 • Windows の場合 %CSCIW_HOME%¥log • Unix の場合 ${CSCIW_HOME}/log	任意
AdmMsgFileDir	案件運用操作メッセージファイル出力先ディレクトリ 案件運用操作メッセージファイルの出力先ディレクトリを指定します。 デフォルトの出力先ディレクトリは次のとおりです。 • Windows の場合 %CSCIW_HOME%¥log • Unix の場合 ${CSCIW_HOME}/log	任意
AdmTraceFileDir	案件運用操作トレースファイル出力先ディレクトリ 案件運用操作トレースファイルの出力先ディレクトリを指定します。 デフォルトの出力先ディレクトリは次のとおりです。 • Windows の場合 %CSCIW_HOME%¥log	任意

キー※1	内容	設定
AdmTraceFileDir	• Unix の場合 ${CSCIW_HOME}/log	任意
MngRestServiceMsgFileDir	運用管理 REST サービスメッセージファイル出力先ディレクトリ 運用管理 REST サービスメッセージファイルの出力先ディレクトリを指定します。 デフォルトの出力先ディレクトリは次のとおりです。 • Windows の場合 %CSCIW_HOME%¥log • Unix の場合 ${CSCIW_HOME}/log	任意
MngRestServiceTraceFileDir	運用管理 REST サービストレースファイル出力先ディレクトリ 運用管理 REST サービスのトレースファイルの出力先ディレクトリを指定します。 デフォルトの出力先ディレクトリは次のとおりです。 • Windows の場合 %CSCIW_HOME%¥log • Unix の場合 ${CSCIW_HOME}/log	任意
LogRotationMode	メッセージ・トレースファイルのローテーション種別 メッセージ・トレースファイルのローテーション種別を指定します。 • WRAP ラップアラウンドモード • SHIFT シフトモード デフォルト値は WRAP です。	任意
RotationTime	メッセージ・トレースファイルのローテーション時刻 メッセージ・トレースファイルのローテーション時刻を指定します。 ローテーション種別がシフトモードの場合だけ有効になります。 HHMMSS 形式の 6 桁固定文字列を 000000～235959 の範囲で指定します。 空文字列の場合には，時刻によるローテーションは無効となり，ローテーション条件にファイルサイズだけを使用します（この設定はシステムで一意であり，すべてのログ種別に有効です）。 デフォルト値は空文字列です。	任意

キー※1	内容	設定
MsgFileNum	メッセージファイル出力面数 メッセージファイルの出力面数を，1～16 の整数で指定します。ただし，ローテーション種別がシフトモードの場合は，バックアップ面数を指定します。 デフォルト値は 2 です。 初期化メッセージファイルもこの設定に従います。	任意
TraceFileNum	トレースファイル出力面数 トレースファイルの出力面数を，1～16 の整数で指定します。ただし，ローテーション種別がシフトモードの場合は，バックアップ面数を指定します。 デフォルト値は 4 です。 初期化トレースファイルもこの設定に従います。	任意
CmdMsgFileNum	コマンドメッセージファイル出力面数 コマンドメッセージファイルの出力面数を，1～16 の整数で指定します。ただし，ローテーション種別がシフトモードの場合は，バックアップ面数を指定します。 デフォルト値は 2 です。	任意
CmdTraceFileNum	コマンドトレースファイル出力面数 コマンドトレースファイルの出力面数を，1～16 の整数で指定します。ただし，ローテーション種別がシフトモードの場合は，バックアップ面数を指定します。 デフォルト値は 4 です。	任意
AdmMsgFileNum	案件運用操作メッセージファイル出力面数 案件運用操作メッセージファイルの出力面数を，1～16 の整数で指定します。ただし，ローテーション種別がシフトモードの場合は，バックアップ面数を指定します。 デフォルト値は 2 です。	任意
AdmTraceFileNum	案件運用操作トレースファイル出力面数 案件運用操作トレースファイルの出力面数を，1～16 の整数で指定します。ただし，ローテーション種別がシフトモードの場合は，バックアップ面数を指定します。 デフォルト値は 4 です。	任意
MngRestServiceMsgFileNum	運用管理 REST サービスメッセージファイル出力面数 運用管理 REST サービスのメッセージファイルの出力面数を 1～16 の整数で指定します。ただし，ローテーション種別がシフトモードの場合は，バックアップ面数を指定します。 デフォルト値は 2 です。	任意
MngRestServiceTraceFileNum	運用管理 REST サービストレースファイルの出力面数 運用管理 REST サービスのトレースファイルの出力面数を 1～16 の整数で指定します。ただし，ローテーショ	任意

キー※1	内容	設定
MngRestServiceTraceFileNum	ン種別がシフトモードの場合は，バックアップ面数を指定します。 デフォルト値は 4 です。	任意
MsgFileSize	メッセージファイル出力サイズ メッセージファイルの出力サイズを，100000～2147483647 の整数で指定します。(単位：バイト) デフォルト値は 2097152 です。 初期化メッセージファイルもこの設定に従います。	任意
TraceFileSize	トレースファイル出力サイズ トレースファイルの出力サイズを，100000～2147483647 の整数で指定します。(単位：バイト) デフォルト値は 16777216 です。 初期化トレースファイルもこの設定に従います。	任意
CmdMsgFileSize	コマンドメッセージファイル出力サイズ コマンドメッセージファイルの出力サイズを，100000～2147483647 の整数で指定します。(単位：バイト) デフォルト値は 2097152 です。	任意
CmdTraceFileSize	コマンドトレースファイル出力サイズ コマンドメッセージファイルの出力サイズを，100000～2147483647 の整数で指定します。(単位：バイト) デフォルト値は 16777216 です。	任意
AdmMsgFileSize	案件運用操作メッセージファイル出力サイズ 案件運用操作メッセージファイルの出力サイズを，100000～2147483647 の整数で指定します。(単位：バイト) デフォルト値は 2097152 です。	任意
AdmTraceFileSize	案件運用操作トレースファイル出力サイズ 案件運用操作トレースファイルの出力サイズを，100000～2147483647 の整数で指定します。(単位：バイト) デフォルト値は 16777216 です。	任意
MngRestServiceMsgFileSize	運用管理 REST サービスメッセージファイル出力サイズ 運用管理 REST サービスメッセージファイルの出力サイズを，100000～2147483647 の整数で指定します。(単位：バイト) デフォルト値は 2097152 です。	任意

キー※1	内容	設定
MngRestServiceTraceFileSize	運用管理 REST サービストレースファイルの出力サイズ 運用管理 REST サービストレースファイルの出力サイズを，100000～2147483647 の整数で指定します。（単位：バイト） デフォルト値は 16777216 です。	任意
MsgOutputThreashold	メッセージ出力レベル メッセージ出力レベルを，-1～1000 の数値で指定します。 デフォルト値は 20 です。 • 出力しない場合 -1 を指定 • 値以下のレベルを出力する場合 0 以上の値を指定 初期化メッセージファイルもこの設定に従います。	任意
TraceOutputThreashold	トレース出力レベル トレース出力レベルを，-1～1000 の数値で指定します。 デフォルト値は 20 です。 • 出力しない場合 -1 を指定 • 値以下のレベルを出力する場合 0 以上の値を指定 初期化トレースファイルもこの設定に従います。	任意
CmdMsgOutputThreashold	コマンドメッセージ出力レベル コマンドメッセージ出力レベルを，-1～1000 の数値で指定します。 デフォルト値は 20 です。 • 出力しない場合 -1 を指定 • 値以下のレベルを出力する場合 0 以上の値を指定	任意
CmdTraceOutputThreashold	コマンドトレース出力レベル コマンドトレース出力レベルを，-1～1000 の数値で指定します。 デフォルト値は 20 です。 • 出力しない場合 -1 を指定 • 値以下のレベルを出力する場合 0 以上の値を指定	任意

キー※1	内容	設定
AdmMsgOutputThreashold	**案件運用操作メッセージ出力レベル** 案件運用操作メッセージ出力レベルを，-1～1000 の数値で指定します。 デフォルト値は 20 です。 • 出力しない場合 -1 を指定 • 値以下のレベルを出力する場合 0 以上の値を指定	任意
AdmTraceOutputThreashold	**案件運用操作トレース出力レベル** 案件運用操作トレース出力レベルを，-1～1000 の数値で指定します。 デフォルト値は 20 です。 • 出力しない場合 -1 を指定 • 値以下のレベルを出力する場合 0 以上の値を指定	任意
MngRestServiceMsgOutputThreashold	**運用管理 REST サービスメッセージの出力レベル** 運用管理 REST サービスメッセージの出力レベルを，-1～1000 の数値で指定します。 デフォルト値は 20 です。 • 出力しない場合 -1 を指定 • 値以下のレベルを出力する場合 0 以上の値を指定	任意
MngRestServiceTraceOutputThreashold	**運用管理 REST サービストレースの出力レベル** 運用管理 REST サービストレースの出力レベルを，-1～1000 の数値で指定します。 デフォルト値は 20 です。 • 出力しない場合 -1 を指定 • 値以下のレベルを出力する場合 0 以上の値を指定	任意
MsgTimeToDelete	**メッセージファイル削除までの日数** メッセージファイルを削除するまでの日数を，-1～24855 の数値で指定します。 ローテーション種別がラップアラウンドモードの場合だけ有効になります。 デフォルト値は-1 です。 • 削除しない場合 -1 を指定 • 常に削除する場合	任意

キー※1	内容	設定
MsgTimeToDelete	0 を指定※2 • 経過日数で削除する場合 1 以上の値を指定 初期化メッセージファイルもこの設定に従います。	任意
TraceTimeToDelete	**トレースファイル削除までの日数** トレースファイルを削除するまでの日数を，-1～24855 の数値で指定します。 ローテーション種別がラップアラウンドモードの場合だけ有効になります。 デフォルト値は-1 です。 • 削除しない場合 -1 を指定 • 常に削除する場合 0 を指定※2 • 経過日数で削除する場合 1 以上の値を指定 初期化トレースファイルもこの設定に従います。	任意
CmdMsgTimeToDelete	**コマンドメッセージファイル削除までの日数** コマンドメッセージファイルを削除するまでの日数を，-1～24855 の数値で指 定します。 ローテーション種別がラップアラウンドモードの場合だけ有効になります。 デフォルト値は-1 です。 • 削除しない場合 -1 を指定 • 常に削除する場合 0 を指定※2 • 経過日数で削除する場合 1 以上の値を指定	任意
CmdTraceTimeToDelete	**コマンドトレースファイル削除までの日数** コマンドトレースファイルを削除するまでの日数 を，-1～24855 の数値で指定します。 ローテーション種別がラップアラウンドモードの場合だけ有効になります。 デフォルト値は-1 です。 • 削除しない場合 -1 を指定 • 常に削除する場合 0 を指定※2 • 指定日数の経過時点で削除する場合	任意

キー※1	内容	設定
CmdTraceTimeToDelete	1 以上の値を指定	任意
AdmMsgTimeToDelete	**案件運用操作メッセージファイル削除までの日数** 案件運用操作メッセージファイルを削除するまでの日数を，-1～24855 の数値で指定します。 ローテーション種別がラップアラウンドモードの場合だけ有効になります。 デフォルト値は-1 です。 • 削除しない場合 -1 を指定 • 常に削除する場合 0 を指定※2 • 経過日数で削除する場合 1 以上の値を指定	任意
AdmTraceTimeToDelete	**案件運用操作トレースファイル削除までの日数** 案件運用操作トレースファイルを削除するまでの日数を，-1～24855 の数値で指定します。 ローテーション種別がラップアラウンドモードの場合だけ有効になります。 デフォルト値は-1 です。 • 削除しない場合 -1 を指定 • 常に削除する場合 0 を指定※2 • 経過日数で削除する場合 1 以上の値を指定	任意
MngRestServiceJobTimeToDelete	**ジョブレコード自動削除までの日数** ジョブレコードを自動削除するまでの日数を 1～24855 の数値で指定します。 デフォルト値は 60 です。	任意
MngRestServiceMsgTimeToDelete	**運用管理 REST サービスメッセージファイル削除までの日数** 運用管理 REST サービスメッセージファイルを削除するまでの日数を，-1～24855 の数値で指定します。 ローテーション種別がラップアラウンドモードの場合だけ有効になります。 デフォルト値は-1 です。 • 削除しない場合 -1 を指定 • 常に削除する場合 0 を指定※2 • 経過日数で削除する場合	任意

キー※1	内容	設定
MngRestServiceMsgTimeToDelete	1 以上の値を指定	任意
MngRestServiceTraceTimeToDelete	**運用管理 REST サービストレースファイル削除までの日数** 運用管理 REST サービストレースファイルを削除するまでの日数を，-1～24855 の数値で指定します。 ローテーション種別がラップアラウンドモードの場合だけ有効になります。 デフォルト値は-1 です。 • 削除しない場合 -1 を指定 • 常に削除する場合 0 を指定※2 • 経過日数で削除する場合 1 以上の値を指定	任意
ProcessDefinitionCacheSize	**ビジネスプロセス定義のキャッシュサイズ** ビジネスプロセス定義をメモリ上にキャッシュするサイズを，0～1000 の数値で指定します。 デフォルト値は 100 です。	任意
ResourceSelectorCacheSize	**振り分けルール定義のキャッシュサイズ** 振り分けルール定義をメモリ上にキャッシュするサイズを，0～1000 の数値で指定します。 デフォルト値は 1000 です。	任意
AdmBpCrDispMaxCount	**ビジネスプロセス定義一覧／振り分けルール定義一覧の表示最大件数** ビジネスプロセス定義一覧および振り分けルール定義一覧で表示する最大件数を，1～2000 の数値で指定します。 デフォルト値は 1000 です。	任意
AdmPIDispMaxCount	**案件一覧の表示最大件数** 案件一覧の一画面に表示する最大件数を，1～1000 の数値で指定します。 デフォルト値は 100 です。	任意
AdmAIDispMaxCount	**業務ステップ一覧の表示最大件数** 業務ステップ一覧の一画面に表示する最大件数を，1～1000 の数値で指定します。 デフォルト値は 100 です。	任意
AdmWIDispMaxCount	**作業一覧の表示最大件数** 作業一覧の一画面に表示する最大件数を，1～1000 の数値で指定します。 デフォルト値は 100 です。	任意

キー※1	内容	設定
AdmDelPIResultsDispMaxCount	**案件一括削除ジョブ一覧表示最大件数** 案件一括削除ジョブ一覧の一画面に表示する最大件数を，1～1000 の数値で指定します。 デフォルト値は 100 です。	任意
AdmAcDispMaxCount	**アプリケーション呼び出し制御情報一覧表示最大件数** アプリケーション呼び出し制御情報一覧で表示する最大件数を，1～2000 の数値で指定します。 デフォルト値は 1000 です。	任意
AdmAgDispMaxCount	**アプリケーション呼び出しグループ定義一覧表示最大件数** アプリケーション呼び出しグループ定義一覧で表示する最大件数を，1～2000 の数値で指定します。 デフォルト値は 1000 です。	任意
AdmDelPIUnit	**案件運用操作の案件一括削除での削除処理単位** 案件運用操作の案件一括削除での削除処理単位を，1～10000 の数値で指定します。ciwdelpi コマンドの-unt オプションに相当します。 デフォルト値は 100 です。	任意
AdmDelPIUseTimeOffset	**案件運用操作の案件一括削除での経過日数の区切りオプション** 案件運用操作の案件一括削除での経過日数の区切りオプションを，true または false で指定します。ciwdelpi コマンドの-ofs オプションに相当します。 • 経過日数の区切りを「00（時）：00（分）：00（秒）」に設定する場合 true を指定 • 経過日数の区切りを設定しない場合 false を指定 デフォルト値は false です。	任意
SQLServerReadMode	**排他オプションの指定** SQL Server 使用時の排他オプションのモードを 0 または 1 で指定します。 • ダーティリードモードを指定する場合 0 を指定 • リードコミッテッドモードを指定する場合 1 を指定 デフォルト値は 0 です。 なお，リードコミッテッドモードを指定する場合，SQL Server のデータベースの設定で READ_COMMITTED_SNAPSHOT を ON にしてください。READ_COMMITTED_SNAPSHOT が OFF の状態でリードコミッテッドモードにするとデッドロックが発生するおそれがあります。	任意

キー※1	内容	設定
RangeSkipCleanup	**異常終了した ID 範囲のスキップ有無の指定** プロセス情報の正常化処理の際に，異常終了した ID 範囲をスキップするかどうかを，true または false で指定します。 • 使用中になっていた ID 範囲を強制的にスキップして，利用できないままにする場合 true を指定 • 使用中になっていた ID 範囲を回復し，再利用できるようにする場合 false を指定 デフォルト値は false です。false を指定した場合の動作は，CSCIW のバージョンが 02-10 以前の場合の動作と同じです。	任意
UseApplicationCallService	**アプリケーション呼び出しサービスの使用有無の指定** BPMN 連携機能を使用しないで，アプリケーション呼び出しサービスを使用するかどうかを，true または false で指定します。 • 使用する場合 true を指定 • 使用しない場合 false を指定 デフォルト値は false です。 BPMN 連携機能を使用する場合，このプロパティを指定する必要はありません。	任意
MngRestServiceResponseMaxCount	**運用管理 REST サービスの一覧取得最大件数** 運用管理 REST サービスの一覧取得 API で取得する一覧の最大件数を，-1～2147483647 の数値で指定します。この値は，一覧取得 API で取得件数を指定しなかった場合に使用されます。 • すべてのデータを取得する場合 -1 を指定 • 最大取得件数を指定する場合 0 以上を指定 デフォルト値は 100 です。	任意
MngRestServiceUserDescription	**運用管理 REST サービスの API でのユーザ記述子** 運用管理 REST サービスの API で使用するユーザ記述子を 1～32 バイトの文字列で指定します。この値は，API でユーザ記述子を指定しなかった場合に使用されます。 デフォルト値は csciwmngws です。	任意

注※1

記載しているキー以外の内容を指定した場合，指定したキーおよび値は無視されます。

注※2

0 を指定した場合，複数のプロセスを実行したときに，ほかのプロセスのファイルも削除するので注意してください。

付録 A.3　コマンド用環境設定ファイル

コマンド用環境設定ファイルは，設定内容をユーザの環境に合わせて直接変更できます。また，上書きインストール時に，コマンド用環境設定ファイルは上書きされません。

コマンド用環境設定ファイルの格納パスは，次のとおりです。

(1)　環境変数 CSCIW_CONF_DIR が設定されていない場合

- Windows の場合

```
%CSCIW_HOME%¥conf¥csciwcmdconf.bat
```

- UNIX の場合

```
${CSCIW_HOME}/conf/csciwcmdconf
```

(2)　環境変数 CSCIW_CONF_DIR が設定されている場合

- Windows の場合

```
%CSCIW_CONF_DIR%¥csciwcmdconf.bat
```

- UNIX の場合

```
${CSCIW_CONF_DIR}/csciwcmdconf
```

重要

アンインストール時に<CSCIW インストールフォルダ>/conf の下のコマンド用環境設定ファイルは削除されるため，アンインストール前に必要に応じてファイルのバックアップを取得してください。

環境変数 CSCIW_CONF_DIR 下に作成されたコマンド用環境設定ファイルはアンインストール時には削除されません。アンインストール後に残ったコマンド用環境設定ファイルの処理（バックアップの取得や削除）については，必要に応じて実行してください。

コマンド用環境設定ファイルの設定内容を次に示します。

表 A-3　コマンド用環境設定ファイルの設定内容

キー	内容
CSCIWCMD_JVM_ARGS	JavaVM の起動オプション JavaVM を起動するオプションを設定します。次の JavaVM のオプションを指定できます。特に値を指定しない場合は，java コマンドのデフォルト値になります。 • -Xms<size> • -Xmx<size> • -Xss<size> • -Xdebug • -Xrun<libraryName>
CSCIWCMD_JVM_CLPATH※1，※2	クラスパスへの設定値の追加 クラスパスに値を追加します。複数の値を指定したい場合は，次のようにしてください。 • Windows の場合 値を「;」（セミコロン）で区切って指定する • UNIX の場合 値を「:」（コロン）で区切って指定する
CSCIWCMD_JVM_LIBPATH	ライブラリ検索パス環境変数への設定値の追加 ライブラリ検索パス環境変数に値を追加します。複数の値を指定したい場合は，次のようにしてください。 • Windows の場合 値を「;」（セミコロン）で区切って指定する • UNIX の場合 値を「:」（コロン）で区切って指定する
CSCIWCMD_JVM_ARCHITECTURE	この指定は，互換用の設定のため，デフォルト値「64」から値を変更しないでください。 使用する OS が UNIX の場合だけ有効になります。

注※1

コマンドを実行するには，CSCIWCMD_JVM_CLPATH に JDBC のクラスパスを設定する必要があります。JDBC のクラスパスの指定例を次に示します。

・HiRDB の場合

```
set CSCIWCMD_JVM_CLPATH=
    <HiRDBインストールディレクトリ>¥CLIENT¥UTL¥pdjdbc2.jar
```

または

```
set CSCIWCMD_JVM_CLPATH=
    <HiRDBインストールディレクトリ>¥CLIENT¥UTL¥pdjdbc4.jar
```

・Oracle の場合

```
set CSCIWCMD_JVM_CLPATH=
    <Oracleインストールディレクトリ>¥jdbc¥lib¥ojdbc8.jar
```

または

```
set CSCIWCMD_JVM_CLPATH=
    <Oracleインストールディレクトリ>¥jdbc¥lib¥ojdbc10.jar
```

注※2

Windows 環境で，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のインストール先を変更している場合は，変更したインストール先ディレクトリにあるhntrlib2j.jar およびhntrlibMj.jar（64 ビット版の場合は，hntrlib2j64.jar およびhntrlibMj64.jar）を，CSCIWCMD_JVM_CLPATH に追加する必要があります。指定例を次に示します。

```
set CSCIWCMD_JVM_CLPATH=
    <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のインストール先>¥classes¥hn
trlib2j.jar;<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のインストール先>¥cl
asses¥hntrlibMj.jar
```

付録 B　テーブル容量の見積もり

次の情報を基に，使用する DBMS のマニュアルを参考にデータベースの容量見積もりをしてください。

- テーブルおよびインデクス定義
- レコード数の概算式

付録 B.1　テーブルおよびインデクス定義

ワーク管理データベースに定義するテーブルおよびインデクスの内容を基に，テーブル容量を見積もってください。

見積もりに必要なテーブルおよびインデクスの情報を次に示します。

(1)　テーブルの一覧

テーブルの一覧を次に示します。

表 B-1　テーブル一覧

分類	テーブル名	インデクス名 プレフィクス	参照 個所
ビジネスプロセス定義	<SYSTEMID>_PROCESS_DEF	<SYSTEMID>_PDEF	(a)
業務ステップ定義	<SYSTEMID>_ACTIVITY_DEF	<SYSTEMID>_ADEF	(b)
作業定義	<SYSTEMID>_WORK_DEF	<SYSTEMID>_WDEF	(c)
制御ノード定義	<SYSTEMID>_CONTROLNODE_DEF	<SYSTEMID>_CNDEF	(d)
フロー遷移定義	<SYSTEMID>_FLOW_DEF	<SYSTEMID>_FDEF	(e)
並列作業定義	<SYSTEMID>_PARALLEL_WORK_DEF	<SYSTEMID>_PWDEF	(f)
条件定義	<SYSTEMID>_CONDITION_DEF	<SYSTEMID>_CDDEF	(g)
条件 SQL 定義	<SYSTEMID>_CONDITION_SQL_DEF	<SYSTEMID>_CDSQLDEF	(h)
条件 Java 定義	<SYSTEMID>_CONDITION_JAVA_DEF	<SYSTEMID>_CDJAVADEF	(i)
生成ルール定義	<SYSTEMID>_CREATIONRULE_DEF	<SYSTEMID>_CRDEF	(j)
生成ルール SQL 定義	<SYSTEMID>_CREATIONRULE_SQL_DEF	<SYSTEMID>_CRSQLDEF	(k)
生成ルール Java 定義	<SYSTEMID>_CREATIONRULE_JAVA_DEF	<SYSTEMID>_CRJAVADEF	(l)
時間取得ルール定義	<SYSTEMID>_TIMERULE_DEF	<SYSTEMID>_TRDEF	(m)
時間取得ルール相対時間定義	<SYSTEMID>_TIMERULE_DATETIME_DEF	<SYSTEMID>_TRDDEF	(n)
作業アプリケーション定義	<SYSTEMID>_APPLICATION_DEF	<SYSTEMID>_APPDEF	(o)

分類	テーブル名	インデクス名 プレフィクス	参照 個所
作業アプリケーション JMS 定義	<SYSTEMID>_APPLICATION_JMS_DEF	<SYSTEMID>_APPJMSDEF	(p)
作業アプリケーション Java 定義	<SYSTEMID>_APPLICATION_JAVA_DEF	<SYSTEMID>_APPJAVADEF	(q)
階層定義	<SYSTEMID>_HIERARCHY_DEF	<SYSTEMID>_HIEDEF	(r)
フロー遷移折れ線情報	<SYSTEMID>_FLOW_POLYGONALLINE_INFO	<SYSTEMID>_FPOLINF	(s)
振り分けルール定義	<SYSTEMID>_CASTINGRULE_DEF	<SYSTEMID>_CASTDEF	(t)
振り分けルール SQL 定義	<SYSTEMID>_CASTINGRULE_SQL_DEF	<SYSTEMID>_CASTSQLDEF	(u)
振り分けルール Java 定義	<SYSTEMID>_CASTINGRULE_JAVA_DEF	<SYSTEMID>_CASTJAVADEF	(v)
案件	<SYSTEMID>_PROCESS_INSTANCE	<SYSTEMID>_PI	(w)
業務ステップ	<SYSTEMID>_ACTIVITY_INSTANCE	<SYSTEMID>_AI	(x)
作業	<SYSTEMID>_WORK_ITEM	<SYSTEMID>_WI	(y)
制御ノード	<SYSTEMID>_CONTROLNODE_INSTANCE	<SYSTEMID>_CI	(z)
制御ノード管理	<SYSTEMID>_CONTROLNODE_INSTANCEMNG	<SYSTEMID>_CIM	(aa)
ユーザ設定情報	<SYSTEMID>_USER_SETTING_INFO	<SYSTEMID>_USM	(ab)
プロセス情報	<SYSTEMID>_SERVER_PROC_INFO	<SYSTEMID>_SPI	(ac)
未使用 ID 範囲格納テーブル	<SYSTEMID>_UNUSED_ID_RANGE	<SYSTEMID>_UID	(ad)
ID 範囲予約ログテーブル	<SYSTEMID>_ID_RANGE_RESERVE_LOG	<SYSTEMID>_IDR	(ae)
アプリケーション呼び出し制御情報テーブル	<SYSTEMID>_APPLICATION_LOCK_INFO	<SYSTEMID>_APPLCK	(af)
アプリケーション呼び出しグループテーブル	<SYSTEMID>_APPLICATION_LOCK_GROUP	<SYSTEMID>_APPGRP	(ag)
ジョブテーブル	<SYSTEMID>_ASYNC_JOB	<SYSTEMID>_ASYNCJOB	(ah)

各テーブルの内容を次に示します。

(a) <SYSTEMID>_PROCESS_DEF の内容

このテーブルでは，ビジネスプロセス定義の情報を管理します。

<SYSTEMID>_PROCESS_DEF の内容を次に示します。

表 B-2　<SYSTEMID>_PROCESS_DEF の内容

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
1	ビジネスプロセス定義 ID	ID	INTEGER	NUMBER(10)	INT	P1	N	不可
2	ビジネスプロセス定義名	Name	MVARCHAR(64)	VARCHAR2(64)	VARCHAR(64)	U2-1	N	不可
3	バージョン	Version	SMALLINT	NUMBER(4)	SMALLINT	U2-2	N	不可
4	状態	StateCode	CHAR(1)	CHAR(1)	CHAR(1)	－	－	不可
5	説明	Description	MVARCHAR(250)	VARCHAR2(250)	VARCHAR(250)	－	－	可 (NULL)
6	作成者	Author	MVARCHAR(32)	VARCHAR2(32)	VARCHAR(32)	－	－	不可
7	作成日時	CreationDate	DECIMAL(19)	NUMBER(19)	BIGINT	－	－	不可
8	管理者	Responsible	MVARCHAR(32)	VARCHAR2(32)	VARCHAR(32)	－	－	不可
9	案件投入可能期間開始日時	ValidFromDate	DECIMAL(19)	NUMBER(19)	BIGINT	－	－	不可
10	案件投入可能期間終了日時	ValidToDate	DECIMAL(19)	NUMBER(19)	BIGINT	－	－	不可
11	処理期限ルール定義 ID	DeadlineRuleID	INTEGER	NUMBER(10)	INT	－	－	可 (NULL)
12	Adhoc 操作対応フラグ	AdhocOperationTypeCode	CHAR(1)	CHAR(1)	CHAR(1)	－	－	不可
13	スクリプトバージョン	ScriptVersion	SMALLINT	NUMBER(4)	SMALLINT	－	－	不可
14	クライアント識別情報	ClientInformation	MVARCHAR(32)	VARCHAR2(32)	VARCHAR(32)	－	－	可 (NULL)

(凡例)

N：NOT NULL 制約を定義する

－：該当しない

注※

インデクスの形式は，<T><通番>[-<構成順序>]です。

<T>部分のアルファベットの意味は次のとおりです。

U：ユニーク

P：主キー（ただし，ユニークとNOT NULL の組み合わせで実現）

(b) <SYSTEMID>_ACTIVITY_DEF の内容

このテーブルでは，業務ステップ定義の情報を管理します。

<SYSTEMID>_ACTIVITY_DEF の内容を次に示します。

表 B-3 <SYSTEMID>_ACTIVITY_DEF の内容

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
1	業務ステップ定義 ID	ID	INTEGER	NUMBER(10)	INT	U2-2 P1	N	不可
2	業務ステップ定義名	Name	MVARCHAR(64)	VARCHAR2(64)	VARCHAR(64)	N3	－	不可
3	ビジネスプロセス定義 ID	ProcessDefinitionID	INTEGER	NUMBER(10)	INT	U2-1	N	不可
4	説明	Description	MVARCHAR(250)	VARCHAR2(250)	VARCHAR(250)	－	－	可 (NULL)
5	処理期限ルール定義 ID	DeadlineRuleID	INTEGER	NUMBER(10)	INT	－	－	可 (NULL)
6	完了条件 ID	CompletionConditionID	INTEGER	NUMBER(10)	INT	－	－	可 (NULL)
7	生成ポリシー	CreationPolicy	CHAR(1)	CHAR(1)	CHAR(1)	－	－	不可
8	階層定義 ID	HierarchyDefinitionID	INTEGER	NUMBER(10)	INT	－	－	可 (NULL)
9	アイコン番号	IconNumber	SMALLINT	NUMBER(4)	SMALLINT	－	－	不可
10	描画順位	DrawPriority	INTEGER	NUMBER(10)	INT	－	－	可 (NULL)
11	x 座標	X	INTEGER	NUMBER(10)	INT	－	－	不可

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
12	y 座標	Y	INTEGER	NUMBER(10)	INT	－	－	不可

(凡例)

N：NOT NULL 制約を定義する

－：該当しない

注※

インデクスの形式は，<T><通番>[-<構成順序>]です。

<T>部分のアルファベットの意味は次のとおりです。

N：通常

U：ユニーク

P：主キー（ただし，ユニークとNOT NULL の組み合わせで実現）

(c) <SYSTEMID>_WORK_DEF の内容

このテーブルでは，作業定義の情報を管理します。

<SYSTEMID>_WORK_DEF の内容を次に示します。

表 B-4 <SYSTEMID>_WORK_DEF の内容

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
1	作業定義 ID	ID	INTEGER	NUMBER (10)	INT	U2-2 P1	N	不可
2	作業定義名	Name	MVARCHAR(64)	VARCHAR2(64)	VARCHAR(64)	N3	－	不可
3	業務ステップ定義 ID	ActivityDefinitionID	INTEGER	NUMBER(10)	INT	－	－	不可
4	ビジネスプロセス定義 ID	ProcessDefinitionID	INTEGER	NUMBER(10)	INT	U2-1	N	不可
5	作業種別	WorkTypeCode	CHAR(1)	CHAR(1)	CHAR(1)	－	－	不可
6	振り分けルール名	CastingRuleName	MVARCHAR(64)	VARCHAR2(64)	VARCHAR(64)	－	－	可 (NULL)
7	説明	Description	MVARCHAR(250)	VARCHAR2(250)	VARCHAR(250)	－	－	可 (NULL)
8	発生条件	CreationConditionID	INTEGER	NUMBER (10)	INT	－	－	可 (NULL)

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
9	完了条件	CompletionConditionID	INTEGER	NUMBER(10)	INT	－	－	可 (NULL)
10	再評価指定	ReEvalFlag	CHAR(1)	CHAR(1)	CHAR(1)	－	－	不可
11	アプリケーション定義 ID	ApplicationID	INTEGER	NUMBER(10)	INT	－	－	可 (NULL)

(凡例)

N：NOT NULL 制約を定義する

－：該当しない

注※

インデクスの形式は，<T><通番>[-<構成順序>]です。

<T>部分のアルファベットの意味は次のとおりです。

N：通常

U：ユニーク

P：主キー（ただし，ユニークとNOT NULL の組み合わせで実現）

(d) <SYSTEMID>_CONTROLNODE_DEF の内容

このテーブルでは，制御ノード定義の情報を管理します。

<SYSTEMID>_CONTROLNODE_DEF の内容を次に示します。

表 B-5 <SYSTEMID>_CONTROLNODE_DEF の内容

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
1	制御ノード定義 ID	ID	INTEGER	NUMBER(10)	INT	U2-2 P1	N	不可
2	ビジネスプロセス定義 ID	ProcessDefinitionID	INTEGER	NUMBER(10)	INT	U2-1	N	不可
3	制御ノード種別	ControlNodeTypeCode	CHAR(1)	CHAR(1)	CHAR(1)	－	－	不可
4	制御ノード名	Name	MVARCHAR(64)	VARCHAR2(64)	VARCHAR(64)	－	－	不可
5	階層定義 ID	HierarchyDefinitionID	INTEGER	NUMBER(10)	INT	－	－	可 (NULL)
6	アイコン番号	IconNumber	SMALLINT	NUMBER(4)	SMALLINT	－	－	不可

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
7	描画順位	DrawPriority	INTEGER	NUMBER(10)	INT	－	－	可 (NULL)
8	x 座標	X	INTEGER	NUMBER(10)	INT	－	－	不可
9	y 座標	Y	INTEGER	NUMBER(10)	INT	－	－	不可

(凡例)

N：NOT NULL 制約を定義する

－：該当しない

注※

インデクスの形式は，<T><通番>[-<構成順序>]です。

<T>部分のアルファベットの意味は次のとおりです。

U：ユニーク

P：主キー（ただし，ユニークとNOT NULL の組み合わせで実現）

(e) <SYSTEMID>_FLOW_DEF の内容

このテーブルでは，フロー遷移定義の情報を管理します。

<SYSTEMID>_FLOW_DEF の内容を次に示します。

表 B-6 <SYSTEMID>_FLOW_DEF の内容

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
1	フロー遷移定義 ID	ID	INTEGER	NUMBER(10)	INT	U2-2 P1	N	不可
2	ビジネスプロセス定義 ID	ProcessDefinitionID	INTEGER	NUMBER(10)	INT	U2-1	N	不可
3	遷移元ノード種別	SourceNodeTypeCode	CHAR(1)	CHAR(1)	CHAR(1)	－	－	不可
4	遷移元ノード ID	SourceNodeID	INTEGER	NUMBER(10)	INT	－	－	不可
5	遷移先ノード種別	DestinationNodeTypeCode	CHAR(1)	CHAR(1)	CHAR(1)	－	－	不可
6	遷移先ノード ID	DestinationNodeID	INTEGER	NUMBER(10)	INT	－	－	不可

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
7	優先順位	Priority	SMALLINT	NUMBER(6)	SMALLINT	－	－	可 (NULL)
8	遷移条件 ID	FlowConditionID	INTEGER	NUMBER(10)	INT	－	－	可 (NULL)
9	階層定義 ID	HierarchyDefinitionID	INTEGER	NUMBER(10)	INT	－	－	可 (NULL)
10	描画順位	DrawPriority	INTEGER	NUMBER(10)	INT	－	－	可 (NULL)
11	線種	LineStyle	SMALLINT	NUMBER(4)	SMALLINT	－	－	不可
12	線色	LineColor	INTEGER	NUMBER(10)	INT	－	－	不可
13	線幅	LineWidth	SMALLINT	NUMBER(4)	SMALLINT	－	－	不可

(凡例)

N：NOT NULL 制約を定義する

－：該当しない

注※

インデクスの形式は，<T><通番>[-<構成順序>]です。

<T>部分のアルファベットの意味は次のとおりです。

U：ユニーク

P：主キー（ただし，ユニークとNOT NULL の組み合わせで実現）

(f) <SYSTEMID>_PARALLEL_WORK_DEF の内容

このテーブルでは，並列作業定義の情報を管理します。

<SYSTEMID>_PARALLEL_WORK_DEF の内容を次に示します。

表 B-7 <SYSTEMID>_PARALLEL_WORK_DEF の内容

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
1	並列作業定義 ID	ID	INTEGER	NUMBER(10)	INT	P1-2	N	不可
2	ビジネスプロセス定義 ID	ProcessDefinitionID	INTEGER	NUMBER(10)	INT	P1-1	N	不可
3	子作業定義 ID	ChildWorkDefID	INTEGER	NUMBER(10)	INT	－	－	不可

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
4	並列作業生成ルール ID	CreationRuleID	INTEGER	NUMBER(10)	INT	－	－	不可

(凡例)

N：NOT NULL 制約を定義する

－：該当しない

注※

インデクスの形式は，<T><通番>[-<構成順序>]です。

<T>部分のアルファベットの意味は次のとおりです。

P：主キー（ただし，ユニークとNOT NULL の組み合わせで実現）

(g) <SYSTEMID>_CONDITION_DEF の内容

このテーブルでは，条件定義の情報を管理します。

<SYSTEMID>_CONDITION_DEF の内容を次に示します。

表 B-8 <SYSTEMID>_CONDITION_DEF の内容

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
1	条件定義 ID	ID	INTEGER	NUMBER(10)	INT	P1 U2-2	N	不可
2	ビジネスプロセス定義 ID	ProcessDefinitionID	INTEGER	NUMBER(10)	INT	U2-1	N	不可
3	条件名	Name	MVARCHAR(64)	VARCHAR2(64)	VARCHAR(64)	－	－	不可
4	条件種別	ConditionTypeCode	CHAR(1)	CHAR(1)	CHAR(1)	－	－	不可
5	説明	Description	MVARCHAR(250)	VARCHAR2(250)	VARCHAR(250)	－	－	可 (NULL)
6	NOT 指定	NotFlagTypeCode	CHAR(1)	CHAR(1)	CHAR(1)	－	－	不可

(凡例)

N：NOT NULL 制約を定義する

－：該当しない

注※

インデクスの形式は，<T><通番>[-<構成順序>]です。

<T>部分のアルファベットの意味は次のとおりです。

U：ユニーク

P：主キー（ただし，ユニークとNOT NULL の組み合わせで実現）

(h) <SYSTEMID>_CONDITION_SQL_DEF の内容

このテーブルでは，条件 SQL 定義の情報を管理します。

<SYSTEMID>_CONDITION_SQL_DEF の内容を次に示します。

表 B-9 <SYSTEMID>_CONDITION_SQL_DEF の内容

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
1	条件定義 ID	ID	INTEGER	NUMBER(10)	INT	P1	N	不可
2	データベース識別子	DatabaseID	MVARCHAR(32)	VARCHAR2(32)	VARCHAR(32)	－	－	不可
3	SQL 文	Content	MVARCHAR(2000)	VARCHAR2(2000)	VARCHAR(2000)	－	－	不可

(凡例)

N：NOT NULL 制約を定義する

－：該当しない

注※

インデクスの形式は，<T><通番>[-<構成順序>]です。

<T>部分のアルファベットの意味は次のとおりです。

P：主キー（ただし，ユニークとNOT NULL の組み合わせで実現）

(i) <SYSTEMID>_CONDITION_JAVA_DEF の内容

このテーブルでは，条件 Java 定義の情報を管理します。

<SYSTEMID>_CONDITION_JAVA_DEF の内容を次に示します。

表 B-10 <SYSTEMID>_CONDITION_JAVA_DEF の内容

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
1	条件定義 ID	ID	INTEGER	NUMBER(10)	INT	P1	N	不可
2	Java クラス名	ClassName	MVARCHAR(255)	VARCHAR2(255)	VARCHAR(255)	－	－	不可
3	補助定数パラメタ	ExParameter	MVARCHAR(2000)	VARCHAR2(2000)	VARCHAR(2000)	－	－	可 (NULL)

(凡例)

N：NOT NULL 制約を定義する

－：該当しない

注※

インデクスの形式は，<T><通番>[-<構成順序>]です。

<T>部分のアルファベットの意味は次のとおりです。

P：主キー（ただし，ユニークとNOT NULL の組み合わせで実現）

(j) <SYSTEMID>_CREATIONRULE_DEF の内容

このテーブルでは，生成ルール定義の情報を管理します。

<SYSTEMID>_CREATIONRULE_DEF の内容を次に示します。

表 B-11 <SYSTEMID>_CREATIONRULE_DEF の内容

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
1	生成ルール定義 ID	ID	INTEGER	NUMBER(10)	INT	P1 U2-2	N	不可
2	ビジネスプロセス定義 ID	ProcessDefinitionID	INTEGER	NUMBER(10)	INT	U2-1	N	不可
3	生成ルール名	Name	MVARCHAR(64)	VARCHAR2(64)	VARCHAR(64)	－	－	不可
4	生成ルール種別	CreationRuleType	CHAR(1)	CHAR(1)	CHAR(1)	－	－	不可
5	説明	Description	MVARCHAR(250)	VARCHAR2(250)	VARCHAR(250)	－	－	可 (NULL)

(凡例)

N：NOT NULL 制約を定義する

－：該当しない

注※

インデクスの形式は，<T><通番>[-<構成順序>]です。

<T>部分のアルファベットの意味は次のとおりです。

U：ユニーク

P：主キー（ただし，ユニークとNOT NULL の組み合わせで実現）

(k) <SYSTEMID>_CREATIONRULE_SQL_DEF の内容

このテーブルでは，生成ルール SQL 定義の情報を管理します。

<SYSTEMID>_CREATIONRULE_SQL_DEF の内容を次に示します。

表 B-12　<SYSTEMID>_CREATIONRULE_SQL_DEF の内容

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
1	生成ルール定義 ID	ID	INTEGER	NUMBER(10)	INT	P1	N	不可
2	データベース識別子	DatabaseID	MVARCHAR(32)	VARCHAR2(32)	VARCHAR(32)	－	－	不可
3	SQL 文	Content	MVARCHAR(2000)	VARCHAR2(2000)	VARCHAR(2000)	－	－	不可

（凡例）

N：NOT NULL 制約を定義する

－：該当しない

注※

インデクスの形式は，<T><通番>[-<構成順序>]です。

<T>部分のアルファベットの意味は次のとおりです。

P：主キー（ただし，ユニークとNOT NULL の組み合わせで実現）

(l)　<SYSTEMID>_CREATIONRULE_JAVA_DEF の内容

このテーブルでは，生成ルール Java 定義の情報を管理します。

<SYSTEMID>_CREATIONRULE_JAVA_DEF の内容を次に示します。

表 B-13　<SYSTEMID>_CREATIONRULE_JAVA_DEF の内容

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
1	生成ルール定義 ID	ID	INTEGER	NUMBER(10)	INT	P1	N	不可
2	Java クラス名	ClassName	MVARCHAR(255)	VARCHAR2(255)	VARCHAR(255)	－	－	不可
3	補助定数パラメタ	ExParameter	MVARCHAR(2000)	VARCHAR2(2000)	VARCHAR(2000)	－	－	可 (NULL)

（凡例）

N：NOT NULL 制約を定義する

－：該当しない

注※

インデクスの形式は，<T><通番>[-<構成順序>]です。

<T>部分のアルファベットの意味は次のとおりです。

P：主キー（ただし，ユニークとNOT NULL の組み合わせで実現）

(m) <SYSTEMID>_TIMERULE_DEF の内容

このテーブルでは，時間取得ルール定義の情報を管理します。

<SYSTEMID>_TIMERULE_DEF の内容を次に示します。

表 B-14 <SYSTEMID>_TIMERULE_DEF の内容

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
1	時間取得ルール定義 ID	ID	INTEGER	NUMBER(10)	INT	P1 U2-2	N	不可
2	ビジネスプロセス定義 ID	ProcessDefinitionID	INTEGER	NUMBER(10)	INT	U2-1	N	不可
3	時間取得ルール名	Name	MVARCHAR(64)	VARCHAR2(64)	VARCHAR(64)	－	－	不可
4	時間取得ルール種別	TimeRuleType	CHAR(1)	CHAR(1)	CHAR(1)	－	－	不可
5	説明	Description	MVARCHAR(250)	VARCHAR2(250)	VARCHAR(250)	－	－	可 (NULL)

(凡例)

N：NOT NULL 制約を定義する

－：該当しない

注※

インデクスの形式は，<T><通番>[-<構成順序>]です。

<T>部分のアルファベットの意味は次のとおりです。

U：ユニーク

P：主キー（ただし，ユニークとNOT NULL の組み合わせで実現）

(n) <SYSTEMID>_TIMERULE_DATETIME_DEF の内容

このテーブルでは，時間取得ルール相対時間定義の情報を管理します。

<SYSTEMID>_TIMERULE_DATETIME_DEF の内容を次に示します。

表 B-15　<SYSTEMID>_TIMERULE_DATETIME_DEF の内容

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
1	時間取得ルール定義 ID	ID	INTEGER	NUMBER(10)	INT	P1	N	不可
2	日数	NumDays	SMALLINT	NUMBER(4)	SMALLINT	－	－	不可

(凡例)

N：NOT NULL 制約を定義する

－：該当しない

注※

インデクスの形式は，<T><通番>[-<構成順序>]です。

<T>部分のアルファベットの意味は次のとおりです。

P：主キー（ただし，ユニークとNOT NULL の組み合わせで実現）

(o)　<SYSTEMID>_APPLICATION_DEF の内容

このテーブルでは，作業アプリケーション定義の情報を管理します。

<SYSTEMID>_APPLICATION_DEF の内容を次に示します。

表 B-16　<SYSTEMID>_APPLICATION_DEF の内容

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
1	作業アプリケーション定義 ID	ID	INTEGER	NUMBER(10)	INT	P1 U2-2	N	不可
2	ビジネスプロセス定義 ID	ProcessDefinitionID	INTEGER	NUMBER(10)	INT	U2-1	N	不可
3	作業アプリケーション名	Name	MVARCHAR(64)	VARCHAR2(64)	VARCHAR(64)	－	－	不可
4	作業アプリケーション種別	ApplicationTypeCode	CHAR(1)	CHAR(1)	CHAR(1)	－	－	不可
5	アプリケーショ	ApplicationParam	MVARCHAR(2000)	VARCHAR2(2000)	VARCHAR(2000)	－	－	可 (NULL)

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
5	ンパラメータ	ApplicationParam	MVARCHAR(2000)	VARCHAR2(2000)	VARCHAR(2000)	－	－	可 (NULL)
6	説明	Description	MVARCHAR(250)	VARCHAR2(250)	VARCHAR(250)	－	－	可 (NULL)

(凡例)

N：NOT NULL 制約を定義する

－：該当しない

注※

インデクスの形式は，<T><通番>[-<構成順序>]です。

<T>部分のアルファベットの意味は次のとおりです。

P：主キー（ただし，ユニークとNOT NULL の組み合わせで実現）

(p) <SYSTEMID>_APPLICATION_JMS_DEF の内容

このテーブルでは，作業アプリケーション JMS 定義の情報を管理します。

<SYSTEMID>_APPLICATION_JMS_DEF の内容を次に示します。

表 B-17 <SYSTEMID>_APPLICATION_JMS_DEF の内容

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
1	作業アプリケーション定義 ID	ID	INTEGER	NUMBER(10)	INT	P1	N	不可
2	キュー名称	QueueName	MVARCHAR(20)	VARCHAR2(20)	VARCHAR(20)	－	－	不可
3	優先度	MessagePriority	INTEGER	NUMBER(10)	INT	－	－	不可
4	グループ識別子	MessageGroupID	MVARCHAR(512)	VARCHAR2(512)	VARCHAR(512)	－	－	可 (NULL)
5	グループシーケンス番号	MessageGroupSeq	INTEGER	NUMBER(10)	INT	－	－	可 (NULL)
6	自動作業完了	AutoWICompleteFlag	CHAR(1)	CHAR(1)	CHAR(1)	－	－	不可

(凡例)

N：NOT NULL 制約を定義する

－：該当しない

注※

インデクスの形式は，<T><通番>[-<構成順序>]です。

<T>部分のアルファベットの意味は次のとおりです。

P：主キー（ただし，ユニークとNOT NULL の組み合わせで実現）

(q) <SYSTEMID>_APPLICATION_JAVA_DEF の内容

このテーブルでは，作業アプリケーション Java 定義の情報を管理します。

<SYSTEMID>_APPLICATION_JAVA_DEF の内容を次に示します。

表 B-18 <SYSTEMID>_APPLICATION_JAVA_DEF の内容

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
1	作業アプリケーション定義 ID	ID	INTEGER	NUMBER(10)	INT	P1	N	不可
2	Java クラス名	ClassName	MVARCHAR(255)	VARCHAR2(255)	VARCHAR(255)	－	－	不可

(凡例)

N：NOT NULL 制約を定義する

－：該当しない

注※

インデクスの形式は，<T><通番>[-<構成順序>]です。

<T>部分のアルファベットの意味は次のとおりです。

P：主キー（ただし，ユニークとNOT NULL の組み合わせで実現）

(r) <SYSTEMID>_HIERARCHY_DEF の内容

このテーブルでは，階層定義の情報を管理します。

<SYSTEMID>_HIERARCHY_DEF の内容を次に示します。

表 B-19 <SYSTEMID>_HIERARCHY_DEF の内容

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
1	階層定義 ID	ID	INTEGER	NUMBER(10)	INT	P1-2	N	不可
2	ビジネスプロセス定義 ID	ProcessDefinitionID	INTEGER	NUMBER(10)	INT	P1-1	N	不可

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
3	階層名	Name	MVARCHAR(64)	VARCHAR2(64)	VARCHAR(64)	－	－	不可
4	親階層定義 ID	ParentHierarchyDefinitionID	INTEGER	NUMBER(10)	INT	－	－	可 (NULL)
5	アイコン番号	IconNumber	SMALLINT	NUMBER(4)	SMALLINT	－	－	不可
6	描画順位	DrawPriority	INTEGER	NUMBER(10)	INT	－	－	可 (NULL)
7	x 座標	X	INTEGER	NUMBER(10)	INT	－	－	不可
8	y 座標	Y	INTEGER	NUMBER(10)	INT	－	－	不可

(凡例)

N：NOT NULL 制約を定義する

－：該当しない

注※

インデクスの形式は，<T><通番>[-<構成順序>]です。

<T>部分のアルファベットの意味は次のとおりです。

P：主キー（ただし，ユニークとNOT NULL の組み合わせで実現）

(s) <SYSTEMID>_FLOW_POLYGONALLINE_INFO の内容

このテーブルでは，フロー遷移折れ線情報の情報を管理します。

<SYSTEMID>_FLOW_POLYGONALLINE_INFO の内容を次に示します。

表 B-20 <SYSTEMID>_FLOW_POLYGONALLINE_INFO の内容

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
1	フロー遷移定義 ID	ID	INTEGER	NUMBER(10)	INT	P1-2	N	不可
2	ビジネスプロセス定義 ID	ProcessDefinitionID	INTEGER	NUMBER(10)	INT	P1-1	N	不可
3	シリアル番号	SerialNo	SMALLINT	NUMBER(4)	SMALLINT	P1-3	N	不可
4	x 座標	X	INTEGER	NUMBER(10)	INT	－	－	不可

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
5	y 座標	Y	INTEGER	NUMBER(10)	INT	－	－	不可

(凡例)

N：NOT NULL 制約を定義する

－：該当しない

注※

インデクスの形式は，<T><通番>[-<構成順序>]です。

<T>部分のアルファベットの意味は次のとおりです。

P：主キー（ただし，ユニークとNOT NULL の組み合わせで実現）

(t) <SYSTEMID>_CASTINGRULE_DEF の内容

このテーブルでは，振り分けルール定義の情報を管理します。

<SYSTEMID>_CASTINGRULE_DEF の内容を次に示します。

表 B-21 <SYSTEMID>_CASTINGRULE_DEF の内容

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
1	振り分けルール定義 ID	ID	INTEGER	NUMBER(10)	INT	P1	N	不可
2	ルール名	Name	MVARCHAR(64)	VARCHAR2(64)	VARCHAR(64)	U2-1	N	不可
3	バージョン	Version	SMALLINT	NUMBER(4)	SMALLINT	U2-2	N	不可
4	管理者	Responsible	MVARCHAR(32)	VARCHAR2(32)	VARCHAR(32)	－	－	不可
5	作成者	Author	MVARCHAR(32)	VARCHAR2(32)	VARCHAR(32)	－	－	不可
6	作成日時	CreationDate	DECIMAL(19)	NUMBER(19)	BIGINT	－	－	不可
7	説明	Description	MVARCHAR(250)	VARCHAR2(250)	VARCHAR(250)	－	－	可 (NULL)
8	有効期間開始日時	ValidFromDate	DECIMAL(19)	NUMBER(19)	BIGINT	－	－	不可
9	有効期間終了日時	ValidToDate	DECIMAL(19)	NUMBER(19)	BIGINT	－	－	不可
10	状態	StateCode	CHAR(1)	CHAR(1)	CHAR(1)	－	－	不可

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
11	振り分けルール種別	CastingRuleType	CHAR(1)	CHAR(1)	CHAR(1)	－	－	不可
12	スクリプトバージョン	ScriptVersion	SMALLINT	NUMBER(4)	SMALLINT	－	－	不可

(凡例)

N：NOT NULL 制約を定義する

－：該当しない

注※

インデクスの形式は，<T><通番>[-<構成順序>]です。

<T>部分のアルファベットの意味は次のとおりです。

U：ユニーク

P：主キー（ただし，ユニークとNOT NULL の組み合わせで実現）

(u) <SYSTEMID>_CASTINGRULE_SQL_DEF の内容

このテーブルでは，振り分けルール SQL 定義の情報を管理します。

<SYSTEMID>_CASTINGRULE_SQL_DEF の内容を次に示します。

表 B-22 <SYSTEMID>_CASTINGRULE_SQL_DEF の内容

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
1	振り分けルール定義 ID	ID	INTEGER	NUMBER(10)	INT	P1	N	不可
2	データベース識別子	DatabaseID	MVARCHAR(32)	VARCHAR2(32)	VARCHAR(32)	－	－	不可
3	SQL 文	Content	MVARCHAR(2000)	VARCHAR2(2000)	VARCHAR(2000)	－	－	不可

(凡例)

N：NOT NULL 制約を定義する

－：該当しない

注※

インデクスの形式は，<T><通番>[-<構成順序>]です。

<T>部分のアルファベットの意味は次のとおりです。

P：主キー（ただし，ユニークとNOT NULL の組み合わせで実現）

(v) <SYSTEMID>_CASTINGRULE_JAVA_DEF の内容

このテーブルでは，振り分けルール Java 定義の情報を管理します。

<SYSTEMID>_CASTINGRULE_JAVA_DEF の内容を次に示します。

表 B-23 <SYSTEMID>_CASTINGRULE_JAVA_DEF の内容

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
1	振り分けルール定義 ID	ID	INTEGER	NUMBER(10)	INT	P1	N	不可
2	Java クラス名	ClassName	MVARCHAR(255)	VARCHAR2(255)	VARCHAR(255)	－	－	不可
3	補助定数パラメタ	ExParameter	MVARCHAR(2000)	VARCHAR2(2000)	VARCHAR(2000)	－	－	可 (NULL)

(凡例)

N：NOT NULL 制約を定義する

－：該当しない

注※

インデクスの形式は，<T><通番>[-<構成順序>]です。

<T>部分のアルファベットの意味は次のとおりです。

P：主キー（ただし，ユニークとNOT NULL の組み合わせで実現）

(w) <SYSTEMID>_PROCESS_INSTANCE の内容

このテーブルでは，案件の情報を管理します。

<SYSTEMID>_PROCESS_INSTANCE の内容を次に示します。

表 B-24 <SYSTEMID>_PROCESS_INSTANCE の内容

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※1	NULLABLE	省略可否 (省略時の値)
1	案件 ID	ID	INTEGER	NUMBER(10)	INT	P1 N3-2 N4-2	N	不可
2	案件名 (案件キー)	Name	MVARCHAR(240)	VARCHAR2(240)	VARCHAR(240)	N2	－	可 (NULL)
3	ビジネスプロセス定義名	ProcessDefinitionName	MVARCHAR(64)	VARCHAR2(64)	VARCHAR(64)	N5-1※2	－	不可

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※1	NULLABLE	省略可否 (省略時の値)
4	ビジネスプロセス定義 ID	ProcessDefinitionID	INTEGER	NUMBER(10)	INT	N3-1	－	不可
5	状態	StateCode	CHAR(1)	CHAR(1)	CHAR(1)	N4-1 N5-2※2		不可
6	投入者	Creator	MVARCHAR(32)	VARCHAR2(32)	VARCHAR(32)	－	－	不可
7	開始日時	StartDate	DECIMAL(19)	NUMBER(19)	BIGINT	－	－	可 (NULL)
8	終了日時	ClosedDate	DECIMAL(19)	NUMBER(19)	BIGINT	－	－	可 (NULL)
9	処理期限	Deadline	DECIMAL(19)	NUMBER(19)	BIGINT	－	－	可 (NULL)
10	優先度	Priority	INTEGER	NUMBER(10)	INT	－	－	可 (NULL)
11	案件乗せ替え日時	MovedDate	DECIMAL(19)	NUMBER(19)	BIGINT	－	－	可 (NULL)

(凡例)

N：NOT NULL 制約を定義する

－：該当しない

注※1

インデクスの形式は，<T><通番>[-<構成順序>]です。

<T>部分のアルファベットの意味は次のとおりです。

N：通常

U：ユニーク

P：主キー（ただし，ユニークとNOT NULL の組み合わせで実現）

注※2

BPMN 連携機能を使用する場合にだけ，追加されるインデクスです。

(x) <SYSTEMID>_ACTIVITY_INSTANCE の内容

このテーブルでは，業務ステップの情報を管理します。

<SYSTEMID>_ACTIVITY_INSTANCE の内容を次に示します。

表 B-25　<SYSTEMID>_ACTIVITY_INSTANCE の内容

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
1	業務ステップ ID	ID	INTEGER	NUMBER(10)	INT	U2-2 P1	N	不可
2	業務ステップ名	Name	MVARCHAR(64)	VARCHAR2(64)	VARCHAR(64)	－	－	可 (NULL)
3	業務ステップ定義名	ActivityDefinitionName	MVARCHAR(64)	VARCHAR2(64)	VARCHAR(64)	－	－	不可
4	案件 ID	ProcessInstanceID	INTEGER	NUMBER(10)	INT	U2-1	N	不可
5	業務ステップ定義 ID	ActivityDefinitionID	INTEGER	NUMBER(10)	INT	－	－	可 (NULL)
6	ビジネスプロセス定義 ID	ProcessDefinitionID	INTEGER	NUMBER(10)	INT	－	－	不可
7	業務ステップ種別	ActivityTypeCode	CHAR(1)	CHAR(1)	CHAR(1)	－	－	不可
8	状態	StateCode	CHAR(1)	CHAR(1)	CHAR(1)	－	－	不可
9	開始日時	StartDate	DECIMAL(19)	NUMBER(19)	BIGINT	－	－	可 (NULL)
10	終了日時	ClosedDate	DECIMAL(19)	NUMBER(19)	BIGINT	－	－	可 (NULL)
11	処理期限	Deadline	DECIMAL(19)	NUMBER(19)	BIGINT	－	－	可 (NULL)
12	案件名 (案件キー)	ProcessInstanceName	MVARCHAR(240)	VARCHAR2(240)	VARCHAR(240)	－	－	可 (NULL)
13	優先度	Priority	INTEGER	NUMBER(10)	INT	－	－	可 (NULL)

(凡例)

N：NOT NULL 制約を定義する

－：該当しない

注※

インデクスの形式は，<T><通番>[-<構成順序>]です。

<T>部分のアルファベットの意味は次のとおりです。

U：ユニーク

P：主キー（ただし，ユニークとNOT NULL の組み合わせで実現）

(y) <SYSTEMID>_WORK_ITEM の内容

このテーブルでは，作業の情報を管理します。

<SYSTEMID>_WORK_ITEM の内容を次に示します。

表 B-26 <SYSTEMID>_WORK_ITEM の内容

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※1	NULLABLE	省略可否 (省略時の値)
1	作業 ID	ID	INTEGER	NUMBER(10)	INT	U2-2 N3-2 P1	N	不可
2	作業名	Name	MVARCHAR(64)	VARCHAR2(64)	VARCHAR(64)	－	－	可 (NULL)
3	作業定義名	WorkDefinitionName	MVARCHAR(64)	VARCHAR2(64)	VARCHAR(64)	－	－	不可
4	業務ステップ ID	ActivityInstanceID	INTEGER	NUMBER(10)	INT	－	－	不可
5	案件 ID	ProcessInstanceID	INTEGER	NUMBER(10)	INT	U2-1 N6-1※2	N	不可
6	作業定義 ID	WorkDefinitionID	INTEGER	NUMBER(10)	INT	N5-1※2 N6-2※2	－	可 (NULL)
7	ビジネスプロセス定義 ID	ProcessDefinitionID	INTEGER	NUMBER(10)	INT	－	－	不可
8	作業種別	WorkTypeCode	CHAR(1)	CHAR(1)	CHAR(1)	－	－	不可
9	状態	StateCode	CHAR(1)	CHAR(1)	CHAR(1)	N4-1※2 N5-2※2	－	不可
10	作業者 ID	Participant	MVARCHAR(32)	VARCHAR2(32)	VARCHAR(32)	N3-1 N4-2※2	－	可 (NULL)
11	発生日時	CreationDate	DECIMAL(19)	NUMBER(19)	BIGINT	－	－	可 (NULL)
12	開始日時	StartDate	DECIMAL(19)	NUMBER(19)	BIGINT	－	－	可 (NULL)
13	終了日時	ClosedDate	DECIMAL(19)	NUMBER(19)	BIGINT	－	－	可 (NULL)

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※1	NULLABLE	省略可否 (省略時の値)
14	処理期限	Deadline	DECIMAL(19)	NUMBER(19)	BIGINT	－	－	可 (NULL)
15	案件名 (案件キー)	ProcessInstanceName	MVARCHAR(240)	VARCHAR2(240)	VARCHAR(240)	－	－	可 (NULL)
16	優先度	Priority	INTEGER	NUMBER(10)	INT	－	－	可 (NULL)

(凡例)

N：NOT NULL 制約を定義する

－：該当しない

注※1

インデクスの形式は，<T><通番>[-<構成順序>]です。

<T>部分のアルファベットの意味は次のとおりです。

N：通常

U：ユニーク

P：主キー（ただし，ユニークとNOT NULL の組み合わせで実現）

注※2

BPMN 連携機能を使用する場合にだけ，追加されるインデクスです。

(z) <SYSTEMID>_CONTROLNODE_INSTANCE の内容

このテーブルでは，制御ノードの情報を管理します。

<SYSTEMID>_CONTROLNODE_INSTANCE の内容を次に示します。

表 B-27 <SYSTEMID>_CONTROLNODE_INSTANCE の内容

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
1	制御ノード ID	ID	INTEGER	NUMBER(10)	INT	U2-2 P1	N	不可
2	案件 ID	ProcessInstanceID	INTEGER	NUMBER(10)	INT	U2-1	N	不可
3	ビジネスプロセス定義 ID	ProcessDefinitionID	INTEGER	NUMBER(10)	INT	－	－	不可
4	制御ノード定義 ID	ControlNodeDefID	INTEGER	NUMBER(10)	INT	－	－	可 (NULL)
5	制御ノード種別	ControlNodeType	CHAR(1)	CHAR(1)	CHAR(1)	－	－	不可

（凡例）

N：NOT NULL 制約を定義する

－：該当しない

注※

インデクスの形式は，<T><通番>[-<構成順序>]です。

<T>部分のアルファベットの意味は次のとおりです。

U：ユニーク

P：主キー（ただし，ユニークとNOT NULL の組み合わせで実現）

(aa)　<SYSTEMID>_CONTROLNODE_INSTANCEMNG の内容

このテーブルでは，制御ノード管理の情報を管理します。

<SYSTEMID>_CONTROLNODE_INSTANCEMNG の内容を次に示します。

表 B-28　<SYSTEMID>_CONTROLNODE_INSTANCEMNG の内容

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX※	NULLABLE	省略可否 (省略時の値)
1	制御ノード ID	ControlNodeID	INTEGER	NUMBER(10)	INT	P1-2	N	不可
2	案件 ID	ProcessInstanceID	INTEGER	NUMBER(10)	INT	P1-1	N	不可
3	詳細情報 1	Data1	INTEGER	NUMBER(10)	INT	P1-3	N	不可
4	詳細情報 2	Data2	CHAR(1)	CHAR(1)	CHAR(1)	01-50，01-60，01-60-01，01-70 でワーク管理データベースのテーブルやインデクスを作成した場合：－ 01-70-01 以降でワーク管理データベースのテーブルやインデクスを作成した場合：P1-4	－	不可
5	詳細情報 3	Data3	CHAR(1)	CHAR(1)	CHAR(1)	－	－	不可

（凡例）

N：NOT NULL 制約を定義する

－：該当しない

注※

インデクスの形式は，<T><通番>[-<構成順序>]です。

<T>部分のアルファベットの意味は次のとおりです。

P：主キー（ただし，ユニークとNOT NULL の組み合わせで実現）

(ab)　<SYSTEMID>_USER_SETTING_INFO の内容

このテーブルでは，ユーザ設定情報の情報を管理します。

<SYSTEMID>_USER_SETTING_INFO の内容を次に示します。

表 B-29　<SYSTEMID>_USER_SETTING_INFO の内容

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
1	設定情報名	UserSettingObject	MVARCHAR(64)	VARCHAR2(64)	VARCHAR(64)	P1	N	不可
2	設定情報値	UserSettingValue	MVARCHAR(4000)	VARCHAR2(4000)	VARCHAR(4000)	－	－	可 (NULL)

(凡例)

N：NOT NULL 制約を定義する

－：該当しない

注※

インデクスの形式は，<T><通番>[-<構成順序>]です。

<T>部分のアルファベットの意味は次のとおりです。

P：主キー（ただし，ユニークとNOT NULL の組み合わせで実現）

(ac)　<SYSTEMID>_SERVER_PROC_INFO の内容

このテーブルでは，プロセス情報の情報を管理します。

<SYSTEMID>_SERVER_PROC_INFO の内容を次に示します。

表 B-30　<SYSTEMID>_SERVER_PROC_INFO の内容

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
1	プロセス種別	ProcessType	CHAR(2)	CHAR(2)	CHAR(2)	－	－	不可
2	ロック ID	LockID	INTEGER	NUMBER(10)	INT	－	－	不可
3	通知名称	NotificationID	VARCHAR(64)	VARCHAR2(64)	VARCHAR(64)	P1	N	不可
4	起動状態	Status	VARCHAR(9)	VARCHAR2(9)	VARCHAR(9)	－	－	不可

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
5	IP アドレス	IPAddress	BINARY(16)	RAW(16)	VARBINARY(16)	－	－	不可

(凡例)

N：NOT NULL 制約を定義する

－：該当しない

注※

インデクスの形式は，<T><通番>[-<構成順序>]です。

<T>部分のアルファベットの意味は次のとおりです。

P：主キー（ただし，ユニークとNOT NULL の組み合わせで実現）

(ad) <SYSTEMID>_UNUSED_ID_RANGE の内容

このテーブルでは，未使用 ID 範囲格納テーブルの情報を管理します。

<SYSTEMID>_UNUSED_ID_RANGE の内容を次に示します。

表 B-31 <SYSTEMID>_UNUSED_ID_RANGE の内容

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
1	採番対象 ID	TargetID	MVARCHAR(30)	VARCHAR2(30)	VARCHAR(30)	P1-1	N	不可
2	未使用 ID 範囲最小値	IDRangeMin	INTEGER	NUMBER(10)	INT	－	－	不可
3	未使用 ID 範囲最大値	IDRangeMax	INTEGER	NUMBER(10)	INT	P1-2	N	不可
4	未使用 ID 数	IDCount	INTEGER	NUMBER(10)	INT	－	－	不可

(凡例)

N：NOT NULL 制約を定義する

－：該当しない

注※

インデクスの形式は，<T><通番>[-<構成順序>]です。

<T>部分のアルファベットの意味は次のとおりです。

P：主キー（ただし，ユニークとNOT NULL の組み合わせで実現）

(ae) <SYSTEMID>_ID_RANGE_RESERVE_LOG の内容

このテーブルでは，ID 範囲予約ログテーブルの情報を管理します。

<SYSTEMID>_ID_RANGE_RESERVE_LOG の内容を次に示します。

表 B-32　<SYSTEMID>_ID_RANGE_RESERVE_LOG の内容

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
1	通知名称	NotificationID	MVARCHAR(64)	VARCHAR2(64)	VARCHAR(64)	－	－	不可
2	採番対象 ID	TargetID	MVARCHAR(30)	VARCHAR2(30)	VARCHAR(30)	P1-1	N	不可
3	予約 ID 範囲最小値	IDRangeMin	INTEGER	NUMBER(10)	INT	－	－	不可
4	予約 ID 範囲最大値	IDRangeMax	INTEGER	NUMBER(10)	INT	P1-2	N	不可
5	使用状態	StateCode	CHAR(1)	CHAR(1)	CHAR(1)	－	－	不可

（凡例）

N：NOT NULL 制約を定義する

－：該当しない

注※

インデクスの形式は，<T><通番>[-<構成順序>]です。

<T>部分のアルファベットの意味は次のとおりです。

P：主キー（ただし，ユニークとNOT NULL の組み合わせで実現）

(af)　<SYSTEMID>_APPLICATION_LOCK_INFO の内容

このテーブルでは，アプリケーション呼び出し制御情報を管理します。

<SYSTEMID>_APPLICATION_LOCK_INFO の内容を次に示します。

表 B-33　<SYSTEMID>_APPLICATION_LOCK_INFO の内容

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
1	ロック識別子	LockName	MVARCHAR(32)	VARCHAR2(32)	P1	N	不可
2	実行間隔	ExecuteInterval	INTEGER	NUMBER(10)	-	N	不可
3	リトライ間隔	RetryInterval	INTEGER	NUMBER(10)	－	N	不可
4	リトライ回数	RetryCount	INTEGER	NUMBER(10)	-	N	不可

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
5	最大作業件数	WorkItemMax	INTEGER	NUMBER(10)	－	N	不可
6	ロック取得時刻	LockDate	DECIMAL(19)	NUMBER(19)	－	－	可 (NULL)
7	ロック状態	LockState	INTEGER	NUMBER(10)	－	N	不可
8	障害復旧間隔	RecoveryTime	INTEGER	NUMBER(10)	－	N	不可

(凡例)

N：NOT NULL 制約を定義する

－：該当しない

注※

インデクスの形式は，<T><通番>[-<構成順序>]です。

<T>部分のアルファベットの意味は次のとおりです。

P：主キー（ただし，ユニークとNOT NULL の組み合わせで実現）

(ag) <SYSTEMID>_APPLICATION_LOCK_GROUP の内容

このテーブルでは，アプリケーション呼び出しグループテーブルの情報を管理します。

<SYSTEMID>_APPLICATION_LOCK_GROUP の内容を次に示します。

表 B-34 <SYSTEMID>_APPLICATION_LOCK_GROUP の内容

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	INDEX ※	NULLABLE	省略可否 (省略時の値)
1	ref 識別子	RefIdentifier	MVARCHAR(32)	VARCHAR2(32)	P1	N	不可
2	グループ名	GroupName	MVARCHAR(32)	VARCHAR2(32)	N2	N	不可

(凡例)

N：NOT NULL 制約を定義する

注※

インデクスの形式は，<T><通番>[-<構成順序>]です。

<T>部分のアルファベットの意味は次のとおりです。

P：主キー（ただし，ユニークとNOT NULL の組み合わせで実現）

N：通常

(ah) <SYSTEMID>_ASYNC_JOB の内容

このテーブルでは，ジョブの情報を管理します。

<SYSTEMID>_ASYNC_JOB の内容を次に示します。

表 B-35　<SYSTEMID>_ASYNC_JOB の内容

項番	名前	列名	HiRDB のデータ型	ORACLE のデータ型	SQL Server のデータ型	INDEX※	NULLABLE	省略可否 (省略時の値)
1	ジョブ ID	ID	DECIMAL(19)	NUMBER(19)	BIGINT	P1	N	不可
2	ファンクション名	FunctionName	MVARCHAR(32)	VARCHAR2(32)	VARCHAR(32)	－	N	不可
3	投入者	Creator	MVARCHAR(32)	VARCHAR2(32)	VARCHAR(32)	－	N	不可
4	引数データ	Argument	MVARCHAR(512)	VARCHAR2(512)	VARCHAR(512)	－	－	可 (NULL)
5	状態	StateCode	CHAR(1)	CHAR(1)	CHAR(1)	－	N	不可
6	リターンコード	ReturnCode	SMALLINT	NUMBER(4)	SMALLINT	－	－	可 (NULL)
7	結果	Result	MVARCHAR(512)	VARCHAR2(512)	VARCHAR(512)	－	－	可 (NULL)
8	開始日時	StartDate	DECIMAL(19)	NUMBER(19)	BIGINT	－	－	可 (NULL)
9	終了日時	ClosedDate	DECIMAL(19)	NUMBER(19)	BIGINT	－	－	可 (NULL)

(凡例)

N：NOT NULL 制約を定義する

－：該当しない

注※

インデクスの形式は，<T><通番>[-<構成順序>]です。

<T>部分のアルファベットの意味は次のとおりです。

P：主キー（ただし，ユニークとNOT NULL の組み合わせで実現）

(2)　インデクスの一覧

インデクスの一覧を次に示します。

表 B-36　インデクス一覧

テーブル名	インデクス名	制約	列
<SYSTEMID>_ACTIVITY_DEF	IDX_<SYSTEMID>_ADEF_P1	U	1
	IDX_<SYSTEMID>_ADEF_U2	U	2
	IDX_<SYSTEMID>_ADEF_N3	NU	1
<SYSTEMID>_ACTIVITY_INSTANCE	IDX_<SYSTEMID>_AI_P1	U	1

テーブル名	インデクス名	制約	列
<SYSTEMID>_ACTIVITY_INSTANCE	IDX_<SYSTEMID>_AI_U2	U	2
<SYSTEM>_APPLICATION_DEF	IDX_<SYSTEMID>_APPDEF_P1	U	1
	IDX_<SYSTEMID>_APPDEF_U2	U	2
<SYSTEM>_APPLICATION_JMS_DEF	IDX_<SYSTEMID>_APPJMSDEF_P1	U	1
<SYSTEM>_APPLICATION_JAVA_DEF	IDX_<SYSTEMID>_APPJAVADEF_P1	U	1
<SYSTEMID>_CASTINGRULE_DEF	IDX_<SYSTEMID>_CASTDEF_P1	U	1
	IDX_<SYSTEMID>_CASTDEF_U2	U	2
<SYSTEMID>_CASTINGRULE_SQL_DEF	IDX_<SYSTEMID>_CASTSQLDEF_P1	U	1
<SYSTEMID>_CASTINGRULE_JAVA_DEF	IDX_<SYSTEMID>_CASTJAVADEF_P1	U	1
<SYSTEMID>_CONDITION_DEF	IDX_<SYSTEMID>_CDDEF_P1	U	1
	IDX_<SYSTEMID>_CDDEF_U2	U	2
<SYSTEMID>_CONDITION_SQL_DEF	IDX_<SYSTEMID>_CDSQLDEF_P1	U	1
<SYSTEMID>_CONDITION_JAVA_DEF	IDX_<SYSTEMID>_CDJAVADEF_P1	U	1
<SYSTEMID>_CONTROLNODE_DEF	IDX_<SYSTEMID>_CNDEF_P1	U	1
	IDX_<SYSTEMID>_CNDEF_U2	U	2
<SYSTEMID>_CONTROLNODE_INSTANCE	IDX_<SYSTEMID>_CI_P1	U	1
	IDX_<SYSTEMID>_CI_U2	U	2
<SYSTEMID>_CONTROLNODE_INSTANCEMNG	IDX_<SYSTEMID>_CIM_P1	U	CSCIW01-50, 01-60, 01-60-01, または 01-70 でワーク管理データベースのテーブルやインデクスを作成した場合：3 CSCIW01-70-01 以降でワーク管理データベースのテーブルやインデクスを作成した場合：4
<SYSTEMID>_CREATIONRULE_DEF	IDX_<SYSTEMID>_CRDEF_P1	U	1
	IDX_<SYSTEMID>_CRDEF_U2	U	2
<SYSTEMID>_CREATIONRULE_SQL_DEF	IDX_<SYSTEMID>_CRSQLDEF_P1	U	1
<SYSTEMID>_CREATIONRULE_JAVA_DEF	IDX_<SYSTEMID>_CRJAVADEF_P1	U	1
<SYSTEMID>_FLOW_DEF	IDX_<SYSTEMID>_FDEF_P1	U	1
	IDX_<SYSTEMID>_FDEF_U2	U	2

テーブル名	インデクス名	制約	列
<SYSTEMID>_FLOW_POLYGONALLINE_INFO	IDX_<SYSTEMID>_FPOLINF_P1	U	3
<SYSTEMID>_HIERARCHY_DEF	IDX_<SYSTEMID>_HIEDEF_P1	U	2
<SYSTEMID>_ID_RANGE_RESERVE_LOG	IDX_<SYSTEMID>_IDR_P1	U	2
<SYSTEMID>_PARALLEL_WORK_DEF	IDX_<SYSTEMID>_PWDEF_P1	U	2
<SYSTEMID>_PROCESS_DEF	IDX_<SYSTEMID>_PDEF_P1	U	1
	IDX_<SYSTEMID>_PDEF_U2	U	2
<SYSTEMID>_PROCESS_INSTANCE	IDX_<SYSTEMID>_PI_P1	U	1
	IDX_<SYSTEMID>_PI_N2	NU	1
	IDX_<SYSTEMID>_PI_N3	NU	2
	IDX_<SYSTEMID>_PI_N4	NU	2
<SYSTEMID>_SERVER_PROC_INFO	IDX_<SYSTEMID>_SPI_P1	U	1
<SYSTEMID>_TIMERULE_DATETIME_DEF	IDX_<SYSTEMID>_TRDDEF_P1	U	1
<SYSTEMID>_TIMERULE_DEF	IDX_<SYSTEMID>_TRDEF_P1	U	1
	IDX_<SYSTEMID>_TRDEF_U2	U	2
<SYSTEMID>_UNUSED_ID_RANGE	IDX_<SYSTEMID>_UID_P1	U	2
<SYSTEMID>_USER_SETTING_INFO	IDX_<SYSTEMID>_USM_P1	U	2
<SYSTEMID>_WORK_DEF	IDX_<SYSTEMID>_WDEF_P1	U	1
	IDX_<SYSTEMID>_WDEF_U2	U	2
	IDX_<SYSTEMID>_WDEF_N3	NU	1
<SYSTEMID>_WORK_ITEM	IDX_<SYSTEMID>_WI_P1	U	1
	IDX_<SYSTEMID>_WI_U2	U	2
	IDX_<SYSTEMID>_WI_N3	NU	2
<SYSTEMID>_APPLICATION_LOCK_INFO	IDX_<SYSTEMID>_APPLCK_P1	U	1
<SYSTEMID>_APPLICATION_LOCK_GROUP	IDX_<SYSTEMID>_APPGRP_P1	U	1
	IDX_<SYSTEMID>_APPGRP_N2	NU	1
<SYSTEMID>_ASYNC_JOB	IDX_<SYSTEMID>_ASYNCJOB_P1	U	1

(凡例)

U：UNIQUE

NU：NONUNIQUE

付録 B.2　レコード数の概算式

データベースの容量は，テーブルのレコード数が影響します。特にインスタンステーブルのレコード数が大きく影響するため，そのほかのテーブルのインデクスを見積もるときよりも細かな見積もりをしてください。

レコード数の見積もりは，それぞれ次に示す式で算出してください。

(1)　インスタンステーブルのレコード数

(a)　案件（テーブル名：<SYSTEMID>_PROCESS_INSTANCE）

$$\text{レコード数} = \sum_{\text{全ビジネスプロセス定義}} (\text{案件数})$$

なお，案件数には終了している案件も含みます。

(b)　業務ステップ（テーブル名：<SYSTEMID>_ACTIVITY_INSTANCE）

$$\text{レコード数} = \sum_{\text{全ビジネスプロセス定義}} \left(\text{案件数} \times \left(\begin{array}{l}\text{ビジネスプロセス定義ごとに1案件で}\\ \text{生成される平均業務ステップ数}\end{array}\right)\right)$$

なお，案件数には終了している案件も含みます。

(c)　作業（テーブル名：<SYSTEMID>_WORK_ITEM）

$$\text{レコード数} = \sum_{\text{全ビジネスプロセス定義}} \left(\text{案件数} \times \left(\begin{array}{l}\text{ビジネスプロセス定義ごとに}\\ \text{1案件で生成される平均作業数}\end{array}\right)\right)$$

なお，案件数には終了している案件も含みます。

また，作業には組み込み作業定義や子作業定義で生成される作業も含みます。

(d)　制御ノード（テーブル名：<SYSTEMID>_CONTROLNODE_INSTANCE）

$$\text{レコード数} = \sum_{\text{全ビジネスプロセス定義}} \left(\text{案件数} \times \left(\left(\begin{array}{l}\text{ビジネスプロセス定義ごとに定義}\\ \text{している制御ノード定義の合計}\end{array}\right)+1\right)\right)$$

なお，案件数には終了している案件も含みます。

また，制御ノード定義は，次に示す制御ノードが対象となります。

- 待合ノード
- 先着ノード（後続停止あり）

(e) 制御ノード管理（テーブル名：<SYSTEMID>_CONTROLNODE_INSTANCEMNG）

$$\text{レコード数} = \sum_{\text{全ビジネスプロセス定義}} \left(\text{案件数} \times \left(\sum_{\text{制御ノード定義}} \left(\begin{array}{l}\text{制御ノード定義ごとの} \\ \text{遷移元のアローの合計}\end{array} \right) \right)\right)$$

なお，案件数には終了している案件も含みます。

また，制御ノード定義は，次に示す制御ノードが対象となります。

- 待合ノード
- 先着ノード（後続停止あり）

(2) 定義テーブルのレコード数

(a) ビジネスプロセス定義（テーブル名：<SYSTEMID>_PROCESS_DEF）

レコード数＝ビジネスプロセス定義数

(b) 業務ステップ定義（テーブル名：<SYSTEMID>_ACTIVITY_DEF）

$$\text{レコード数} = \sum_{\text{全ビジネスプロセス定義}} \left(\text{ビジネスプロセス定義ごとの業務ステップ定義の合計}\right)$$

(c) 作業定義（テーブル名：<SYSTEMID>_WORK_DEF）

$$\text{レコード数} = \sum_{\text{全ビジネスプロセス定義}} \left(\text{ビジネスプロセス定義ごとの作業定義の合計}\right)$$

なお，作業定義には組み込み作業定義や子作業定義も含みます。

(d) 制御ノード定義（テーブル名：<SYSTEMID>_CONTROLNODE_DEF）

$$\text{レコード数} = \sum_{\text{全ビジネスプロセス定義}} \left(\text{ビジネスプロセス定義ごとの制御ノード定義の合計}\right)$$

(e) フロー遷移定義（テーブル名：<SYSTEMID>_FLOW_DEF）

$$\text{レコード数} = \sum_{\text{全ビジネスプロセス定義}} \left(\text{ビジネスプロセス定義ごとのアロー定義の合計}\right)$$

なお，階層内のアロー定義数も含みます。

(f) 並列作業定義（テーブル名：<SYSTEMID>_PARALLEL_WORK_DEF）

$$\text{レコード数} = \sum_{\text{全ビジネスプロセス定義}} \left(\text{ビジネスプロセス定義ごとの並列作業定義の合計}\right)$$

(g) 条件定義（テーブル名：<SYSTEMID>_CONDITION_DEF）

レコード数＝ $\sum_{\text{全ビジネスプロセス定義}}$（ビジネスプロセス定義ごとの条件定義の合計＋2）

(h) 条件 SQL 定義（テーブル名：<SYSTEMID>_CONDITION_SQL_DEF）

レコード数＝ $\sum_{\text{全ビジネスプロセス定義}}$（ビジネスプロセス定義ごとの条件SQL定義の合計）

(i) 条件 Java 定義（テーブル名：<SYSTEMID>_CONDITION_JAVA_DEF）

レコード数＝ $\sum_{\text{全ビジネスプロセス定義}}$（ビジネスプロセス定義ごとの条件Java定義の合計）

(j) 生成ルール定義（テーブル名：<SYSTEMID>_CREATIONRULE_DEF）

レコード数＝ $\sum_{\text{全ビジネスプロセス定義}}$（ビジネスプロセス定義ごとの生成ルール定義の合計）

(k) 生成ルール SQL 定義（テーブル名：<SYSTEMID>_CREATIONRULE_SQL_DEF）

レコード数＝ $\sum_{\text{全ビジネスプロセス定義}}$（ビジネスプロセス定義ごとの生成ルールSQL定義の合計）

(l) 生成ルール Java 定義（テーブル名：<SYSTEMID>_CREATIONRULE_ JAVA _DEF）

レコード数＝ $\sum_{\text{全ビジネスプロセス定義}}$（ビジネスプロセス定義ごとの生成ルールJava定義の合計）

(m) 時間取得ルール定義（テーブル名：<SYSTEMID>_TIMERULE_DEF）

レコード数＝ $\sum_{\text{全ビジネスプロセス定義}}$（ビジネスプロセス定義ごとの時間取得ルール定義の合計）

(n) 時間取得ルール相対時間定義（テーブル名：<SYSTEMID>_TIMERULE_DATETIME_DEF）

レコード数＝ $\sum_{\text{全ビジネスプロセス定義}}$（ビジネスプロセス定義ごとの時間取得ルール相対時間定義の合計）

(o) 作業アプリケーション定義（テーブル名：<SYSTEMID>_APPLICATION_DEF）

レコード数＝ $\sum_{\text{全ビジネスプロセス定義}}$（ビジネスプロセス定義ごとの作業アプリケーション定義の合計）

(p) 作業アプリケーション JMS 定義（テーブル名：<SYSTEMID>_APPLICATION_JMS_DEF）

$$\text{レコード数}=\sum_{\text{全ビジネスプロセス定義}}\left(\begin{array}{l}\text{ビジネスプロセス定義ごとの}\\ \text{作業アプリケーションJMS定義の合計}\end{array}\right)$$

(q) 作業アプリケーション Java 定義（テーブル名：<SYSTEMID>_APPLICATION_JAVA_DEF）

$$\text{レコード数}=\sum_{\text{全ビジネスプロセス定義}}\left(\begin{array}{l}\text{ビジネスプロセス定義ごとの}\\ \text{作業アプリケーションJava定義の合計}\end{array}\right)$$

(r) 階層定義（テーブル名：<SYSTEMID>_HIERARCHY_DEF）

$$\text{レコード数}=\sum_{\text{全ビジネスプロセス定義}}\left(\text{ビジネスプロセス定義ごとの階層定義の合計}\right)$$

(s) フロー遷移折れ線情報（テーブル名：<SYSTEMID>_FLOW_POLYGONALLINE_INFO）

$$\text{レコード数}=\sum_{\text{全ビジネスプロセス定義}}\left(\sum_{\begin{array}{l}\text{ビジネスプロセス}\\ \text{定義ごとのアロー数}\end{array}}\left(\text{アローごとの折れ線の頂点の合計}\right)\right)$$

(t) 振り分けルール定義（テーブル名：<SYSTEMID>_CASTINGRULE_DEF）

レコード数＝振り分けルール定義数

(u) 振り分けルール SQL 定義（テーブル名：<SYSTEMID>_CASTINGRULE_SQL_DEF）

レコード数＝振り分けルール SQL 定義数

(v) 振り分けルール Java 定義（テーブル名：<SYSTEMID>_CASTINGRULE_ JAVA _DEF）

レコード数＝振り分けルール Java 定義数

(3) そのほかのテーブルのレコード数

(a) 未使用 ID 範囲格納テーブル（テーブル名：<SYSTEMID>_UNUSED_ID_RANGE）

レコード数＝ max(8601)

なお，初期状態のレコード数は 15 です。ciwreuseid（ID の再利用）コマンドで未使用 ID を再利用するごとに増減します。未使用 ID の範囲ブロック数に 11 を加えた数がレコード数になります。未使用 ID の範囲ブロック数は，ciwreuseid（ID の再利用）コマンドで確認できます。ciwreuseid（ID の再利用）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。

(b) ID 範囲予約ログテーブル（テーブル名：<SYSTEMID>_ID_RANGE_RESERVE_LOG）

$$\text{レコード数}=\max\left(\sum_{\text{全マシン数}}\left(\sum_{\substack{\text{マシンごとで起動する}\\ \text{CSCIW全プロセス数}}}\left(\begin{array}{l}\text{CSCIWのプロセスが使用する}\\ \text{ID範囲予約レコード数の合計}\end{array}\right)\right)\right)$$

なお，CSCIW のプロセスが使用する ID 範囲予約レコード数は次のとおりです。

- CSCIW ライブラリを使用している J2EE サーバ：15
- CSCIW ライブラリを使用している Java アプリケーション：4
- ciwmngbp（ビジネスプロセス定義の管理）コマンドおよび ciweditbp（ビジネスプロセス定義の変更）コマンド：10
- ciwmngcr（振り分けルール定義の管理）コマンド：1

(c) ユーザ設定情報（テーブル名：<SYSTEMID>_USER_SETTING_INFO）

レコード数＝ 21

(d) サーバプロセス情報（テーブル名：<SYSTEMID>_SERVER_PROC_INFO）

$$\text{レコード数}=\sum_{\text{全マシン数}}\left(\max\left(\begin{array}{l}\text{マシンごとで同時に起動するCSCIWライブラリ}\\ \text{を使用しているプロセス数の合計}\end{array}\right)+\left(\begin{array}{l}\text{マシンごとで実行する}\\ \text{CSCIWコマンド数の合計}\end{array}\right)\right)$$

(e) アプリケーション呼び出し制御情報 （テーブル名：<SYSTEMID>_APPLICATION_LOCK_INFO）

レコード数＝ ref 識別子※の数＋アプリケーション呼び出しグループの数＋ 5

注※

個別に実行間隔やリトライ回数を設定し，アプリケーション呼び出しグループに所属しない ref 識別子

なお，このレコード数はアプリケーション呼び出しサービスを使用する場合だけ見積もってください。

(f) アプリケーション呼び出しグループ情報 （テーブル名：<SYSTEMID>_APPLICATION_LOCK_GROUP）

$$\text{レコード数}=\sum_{\substack{\text{全アプリケーション}\\ \text{呼び出しグループ数}}}(\text{各グループに所属するref識別子の数})$$

なお，このレコード数はアプリケーション呼び出しサービスを使用する場合だけ見積もってください。

付録 C　トレース容量の見積もり

トレースはファイルへ追記する形で出力します。また，設定したファイル容量を超えた場合，設定したローテーション種別で，メッセージ・トレースファイルのローテーションを行います。ローテーション種別については，「8.2.1(2)　出力ファイルのローテーションの設定」を参照してください。

トレースファイルは，設定した面数を最後まで使い切ると，最初に戻って上書き更新します。トレースファイルが満杯になって，障害発生時の情報が上書きされないように，ある一定の期間は情報を保持できるように余裕を持ってトレース容量を見積もる必要があります。

業務アプリケーションでの時間当たりのメソッドの発行回数とトレース量から，業務システム全体のトレース容量を見積もってください。なお，トレース量は指定する引数やビジネスプロセス定義によって増減するため，余裕を持って見積もってください。

見積もりの目安となる，デフォルトトレースレベルで出力されるトレース量の概算を次に示します。

表 C-1　デフォルトトレースレベルで出力されるトレース量の概算

クラス	メソッド	トレース量の概算 (単位：KB)
CIWFactory	getInstance	0.3
	createCIWServer	0.3
CIWConnectionCoordinator	attachDatabaseConnection	0.4
	detachDatabaseConnection	0.3
CIWServer	allocateWorkItemEx	1.9
	createAndStartProcessInstance	3.4
	createProcessInstance	1.0
	getActivityDefinition	0.6
	getActivityDefinitionsList	0.6
	getActivityInstance	0.6
	getActivityInstancesList	0.6
	getListCount	0.6
	getProcessDefinition	0.6
	getProcessDefinitionsList	0.6
	getProcessInstance	0.6
	getProcessInstancesList	0.6
	getWorkDefinition	0.6
	getWorkDefinitionsList	0.6

クラス	メソッド	トレース量の概算 (単位：KB)
CIWServer	getWorkItem	0.6
	getWorkItemsList	0.6
CIWProcessInstance	adhocInstantiateActivityInstance	1.6
	changeState	1.0
	delete	2.0
	getProcessDefinition	0.6
	getWorkItemsList	0.6
	setDeadline	0.9
	setPriority	0.8
	start	2.1
	terminate	2.3
CIWActivityInstance	adhocMakeTransition	4.9
	changeState	2.0
	evaluate	1.5
	setDeadline	1.1
	makeBackwardTransition	4.9
	setPriority	1.1
CIWWorkItem	changeState	1.7
	complete	4.5
	evaluate	1.5
	free	8.0
	perform	1.5
	performAndComplete	4.5
	reassign	1.1
	reassignAndPerform	1.2
	reevaluateCastingRule	2.5
	setDeadline	1.1
	setPriority	1.1
CIWProcessDefinition	getProcessInstancesList	0.6

注

この表の中に存在しないメソッドについては，デフォルトレベルではトレースを出力しません。

付録 D　キャッシュする ID 範囲の見積もり

CSCIW では，ビジネスプロセスの要素（案件，業務ステップ，作業，制御ノードなど）を識別するためにユニークな ID を採番しています。ID を採番するごとに DBMS にアクセスしないようにするため，ID の採番範囲をメモリ中にキャッシュしています。

ID の採番時にキャッシュする ID 範囲を超えた場合，キャッシュする ID 範囲を DBMS から再度取得します。キャッシュする ID 範囲とは，1 プロセスで ID 採番のためにキャッシュする ID 範囲です。

キャッシュする ID 範囲は，ciwchgenv（システム共通環境情報の変更）コマンドで変更できます。ciwchgenv（システム共通環境情報の変更）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。

ID 採番回数がキャッシュする ID 範囲を超えたときに，キャッシュする ID 範囲を DBMS から取得するため，ID 採番回数に応じてキャッシュする ID 範囲を見積もる必要があります。

ID 種別ごとに ID 採番回数を算出する計算式を次に示します。

●案件 ID

$$\text{ID採番回数} = \sum_{\text{全ビジネスプロセス定義}} (\text{生成する案件数})$$

●業務ステップ ID

$$\text{ID採番回数} = \sum_{\text{全ビジネスプロセス定義}} \left(\text{生成する案件数} \times \left(\begin{array}{l}\text{ビジネスプロセス定義ごとに1案件で}\\ \text{生成される平均業務ステップ数}\end{array}\right)\right)$$

●作業 ID

$$\text{ID採番回数} = \sum_{\text{全ビジネスプロセス定義}} \left(\text{生成する案件数} \times \left(\begin{array}{l}\text{ビジネスプロセス定義ごとに}\\ \text{1案件で生成される平均作業数}\end{array}\right)\right)$$

なお，作業には組み込み作業定義や子作業定義で生成される作業も含みます。

●制御ノード ID

$$\text{ID採番回数} = \sum_{\text{全ビジネスプロセス定義}} \left(\text{生成する案件数} \times \left(\left(\begin{array}{l}\text{ビジネスプロセス定義ごとに定義}\\ \text{している制御ノード定義の合計}\end{array}\right) + 1\right)\right)$$

なお，制御ノード定義は，次に示す制御ノードが対象となります。

- 待合ノード
- 先着ノード（後続停止あり）

付録 E　ciwdelpi コマンド実行時の排他要求数の見積もり（HiRDB 限定）

DBMS が HiRDB の場合に，ciwdelpi（案件の削除）コマンド実行時の排他要求数を見積もる必要があります。ciwdelpi（案件の削除）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。

排他要求数の見積もりについては，マニュアル「HiRDB システム定義（UNIX(R)用）」または「HiRDB システム定義（Windows(R)用）」の「排他制御に関するオペランド」の「pd_lck_pool_size」の説明を参考に算出してください。

算出対象の SQL は，操作系 SQL の DELETE（LOCK TABLE なし，およびインデクスキー値無排他方式の場合）です。DELETE（LOCK TABLE なし，およびインデクスキー値無排他方式の場合）については，マニュアル「HiRDB システム定義（UNIX(R)用）」または「HiRDB システム定義（Windows(R)用）」の「排他資源数の見積もり」の説明を参照してください。

排他資源数の見積もりに必要な値については，次に示します。

表 E-1　前処理の排他要求数を算出する場合に必要な値

項目	指定値
使用する表数	5（BPMN 連携機能を使用する場合は 11） 使用する表の内訳を次に示します。 • 案件テーブル • 業務ステップテーブル • 作業テーブル • 制御ノードテーブル • 制御ノード管理テーブル BPMN 連携機能を使用する場合は，次の表も追加します。 • プロセスデータ（文字列用）テーブル • プロセスデータ（数値用）テーブル • マルチインスタンス管理テーブル • ビジネスプロセス連携テーブル • サブプロセス用マルチインスタンス管理テーブル • アドホック・サブプロセス管理テーブル
使用するビュー表数	－
列数	「付録 B.1　テーブルおよびインデクス定義」を参照し，算出してください。
インデクス数	「付録 B.1　テーブルおよびインデクス定義」を参照し，算出してください。
表分割数	_※
インデクス分割数	_※
ユーザ定義型，または関数の使用	－

項目	指定値
最適化情報	－
ビュー表の使用	－
LOB 列の定義	－
ユーザ定義型の列の定義	－
LOB 属性	－
上位型	－
関数の使用	－
プラグインの使用	－
ルーチンの使用	－

（凡例）

－：未定義，または未使用

注※

ユーザが CREATE TABLE，または CREATE INDEX 文をカスタマイズして，表分割およびインデクス分割を行った場合は算出対象となります。

表 E-2　SQL 文の排他要求数を算出する場合に必要な値

項目	指定値
削除対象 RD エリア数	ユーザの指定した値
削除行数	1 トランザクションで削除する案件の数※1 ×1 案件当たりの削除行数※2 1 案件当たりの削除行数 ＝案件テーブルの削除行数 ＋業務ステップテーブルの削除行数 ＋作業テーブルの削除行数 ＋制御ノードテーブルの削除行数 ＋制御ノード管理テーブルの削除行数 BPMN 連携機能を使用する場合は，次の行数も追加します。 ＋プロセスデータ（文字列用）テーブルの削除行数 ＋プロセスデータ（数値用）テーブルの削除行数 ＋マルチインスタンス管理テーブルの削除行数 ＋ビジネスプロセス連携テーブルの削除行数 ＋サブプロセス用マルチインスタンス管理テーブルの削除行数 ＋アドホック・サブプロセス管理テーブルの削除行数
VARCHAR，NVARCHAR，MVARCHAR の 256 バイト以上の列数	－
1 ページに格納できない BINARY 型の列数	－

項目	指定値
削除した LOB データ数	－
削除した LOB データ格納セグメント数	－
データ型プラグインまたはインデクス型プラグイン	－
pd_inner_replica_control オペランド（UNIX 版 HiRDB の場合）	ユーザの指定した値
ルーチン	－

（凡例）

－：未定義，または未使用

注※1

1 トランザクションで削除する案件の数の最大値は，ciwdelpi（案件の削除）コマンドの-unt オプションで指定した値になります。なお，BPMN 連携機能を使用していて，コールアクティビティを使用している場合は，-unt オプションに指定する案件数は，ルート案件の件数になります。ciwdelpi（案件の削除）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。

注※2

排他要求数の算出対象の削除行数は，ルート案件の行数と，ルート案件に子案件がいる場合は，子案件（子案件の子案件も含む）の行数を合計した値になります。

各テーブルの削除行数の算出方法は，「付録 B.2(1)　インスタンステーブルのレコード数」を参照してください。BPMN 連携機能を使用する場合は，マニュアル「uCosminexus Service Coordinator Interactive Workflow BPMN 連携機能　使用の手引」の「レコード数の概算数」の説明も参照してください。

付録 F　監査ログの詳細

ここでは，出力される監査ログの詳細について説明します。

監査ログの取得対象となるのは，CSCIWManagementServer，コマンド，および案件運用操作に関係する操作です。それぞれに関係する監査ログのメッセージの出力個所を次に示します。

付録 F.1　CSCIWManagementServer に関係する監査ログ

(1)　ログインまたはログアウトの場合

表 F-1　監査ログのメッセージの出力個所（CSCIWManagementServer（ログインまたはログアウトの場合））

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
認証（ログイン）	Authentication	Login	Success	KDIW50000-I
				{0} successfully logged in to uCosminexus Service Coordinator Interactive Workflow ManagementServer. • {0}：ユーザ
			Failure	KDIW50001-E
				An attempt by {0} to log in to uCosminexus Service Coordinator Interactive Workflow ManagementServer has failed. • {0}：ユーザ
ログアウト	Authentication	Logout	Success	KDIW50002-I
				{0} successfully logged out of uCosminexus Service Coordinator Interactive Workflow ManagementServer. • {0}：ユーザ

(2) ビジネスプロセス定義の場合

表 F-2 監査ログのメッセージの出力個所（CSCIWManagementServer（ビジネスプロセス定義の場合））

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
登録	ContentAccess	Add	Occurrence	KDIW50003-I
				{0} executed the operation ({1}). • {0}：ユーザ • {1}：実行されたオペレーション
			Success	KDIW50004-I
				The operation ({0}, PDN:{1}, PDV:{2}) executed by {3} was successful. • {0}：実行されたオペレーション • {1}：ビジネスプロセス定義名 • {2}：ビジネスプロセス定義バージョン • {3}：ユーザ
			Failure	KDIW50015-E
				An attempt by {0} to execute the operation ({1}) has failed. • {0}：ユーザ • {1}：実行されたオペレーション
上書き登録	ContentAccess	Update	Occurrence	KDIW50003-I
				{0} executed the operation ({1}). • {0}：ユーザ • {1}：実行されたオペレーション
			Success	KDIW50004-I
				The operation ({0}, PDN:{1}, PDV:{2}) executed by {3} was successful. • {0}：実行されたオペレーション • {1}：ビジネスプロセス定義名 • {2}：ビジネスプロセス定義バージョン • {3}：ユーザ
			Failure	KDIW50015-E
				An attempt by {0} to execute the operation ({1}) has failed. • {0}：ユーザ • {1}：実行されたオペレーション

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
変更	ContentAccess	Update	Occurrence	KDIW50003-I
				{0} executed the operation ({1}). • {0}：ユーザ • {1}：実行されたオペレーション
			Success	KDIW50004-I
				The operation ({0}, PDN:{1}, PDV:{2}) executed by {3} was successful. • {0}：実行されたオペレーション • {1}：ビジネスプロセス定義名 • {2}：ビジネスプロセス定義バージョン • {3}：ユーザ
			Failure	KDIW50005-E
				An attempt by {0} to execute the operation ({1}, PDN:{2}, PDV:{3}) has failed. • {0}：ユーザ • {1}：実行されたオペレーション • {2}：ビジネスプロセス定義名 • {3}：ビジネスプロセス定義バージョン
削除	ContentAccess	Delete	Occurrence	KDIW50003-I
				{0} executed the operation ({1}). • {0}：ユーザ • {1}：実行されたオペレーション
			Success	KDIW50004-I
				The operation ({0}, PDN:{1}, PDV:{2}) executed by {3} was successful. • {0}：実行されたオペレーション • {1}：ビジネスプロセス定義名 • {2}：ビジネスプロセス定義バージョン • {3}：ユーザ
			Failure	KDIW50005-E
				An attempt by {0} to execute the operation ({1}, PDN:{2}, PDV:{3}) has failed. • {0}：ユーザ • {1}：実行されたオペレーション • {2}：ビジネスプロセス定義名 • {3}：ビジネスプロセス定義バージョン

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
取得	ContentAccess	Refer	Occurrence	KDIW50006-I
				{0} executed the operation ({1}). • {0}：ユーザ • {1}：実行されたオペレーション
			Success	KDIW50007-I
				The operation ({0}, PDN:{1}, PDV:{2}) executed by {3} was successful. • {0}：実行されたオペレーション • {1}：ビジネスプロセス定義名 • {2}：ビジネスプロセス定義バージョン • {3}：ユーザ
			Failure	KDIW50008-E
				An attempt by {0} to execute the operation ({1}, PDN:{2}, PDV:{3}) has failed. • {0}：ユーザ • {1}：実行されたオペレーション • {2}：ビジネスプロセス定義名 • {3}：ビジネスプロセス定義バージョン

(3) 振り分けルール定義の場合

表 F-3 監査ログのメッセージの出力個所（ManagementServer（振り分けルール定義の場合））

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
登録	ContentAccess	Add	Occurrence	KDIW50003-I
				{0} executed the operation ({1}). • {0}：ユーザ • {1}：実行されたオペレーション
			Success	KDIW50010-I
				Theoperation ({0}, CRN:{1}, CRV:{2}) executed by {3} was successful. • {0}：実行されたオペレーション • {1}：振り分けルール名 • {2}：振り分けルールバージョン • {3}：ユーザ

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
登録	ContentAccess	Add	Failure	KDIW50016-E
				An attempt by {0} to execute the operation ({1}) has failed. • {0}：ユーザ • {1}：実行されたオペレーション
上書き登録	ContentAccess	Update	Occurrence	KDIW50003-I
				{0} executed the operation ({1}). • {0}：ユーザ • {1}：実行されたオペレーション
			Success	KDIW50010-I
				The operation ({0}, CRN:{1}, CRV:{2}) executed by {3} was successful. • {0}：実行されたオペレーション • {1}：振り分けルール名 • {2}：振り分けルールバージョン • {3}：ユーザ
			Failure	KDIW50016-E
				An attempt by {0} to execute the operation ({1}) has failed. • {0}：ユーザ • {1}：実行されたオペレーション
変更	ContentAccess	Update	Occurrence	KDIW50003-I
				{0} executed the operation ({1}). • {0}：ユーザ • {1}：実行されたオペレーション
			Success	KDIW50010-I
				The operation ({0}, CRN:{1}, CRV:{2}) executed by {3} was successful. • {0}：実行されたオペレーション • {1}：振り分けルール名 • {2}：振り分けルールバージョン • {3}：ユーザ
			Failure	KDIW50011-E
				An attempt by {0} to execute the operation ({1} , CRN:{2}, CRV:{3}) has failed. • {0}：ユーザ

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
変更	ContentAccess	Update	Failure	• {1}：実行されたオペレーション • {2}：振り分けルール名 • {3}：振り分けルールバージョン
削除	ContentAccess	Delete	Occurrence	KDIW50003-I
				{0} executed the operation ({1}). • {0}：ユーザ • {1}：実行されたオペレーション
			Success	KDIW50010-I
				The operation ({0}, CRN:{1}, CRV:{2}) executed by {3} was successful. • {0}：実行されたオペレーション • {1}：振り分けルール名 • {2}：振り分けルールバージョン • {3}：ユーザ
			Failure	KDIW50011-E
				An attempt by {0} to execute the operation ({1} , CRN:{2}, CRV:{3}) has failed. • {0}：ユーザ • {1}：実行されたオペレーション • {2}：振り分けルール名 • {3}：振り分けルールバージョン
取得	ContentAccess	Refer	Occurrence	KDIW50006-I
				{0} executed the operation ({1}). • {0}：ユーザ • {1}：実行されたオペレーション
			Success	KDIW50013-I
				The operation ({0}, CRN:{1}, CRV:{2}) executed by {3} was successful. • {0}：実行されたオペレーション • {1}：振り分けルール名 • {2}：振り分けルールバージョン • {3}：ユーザ
			Failure	KDIW50014-E
				An attempt by {0} to execute the operation ({1}, CRN:{2}, CRV:{3}) has failed. • {0}：ユーザ

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
取得	ContentAccess	Refer	Failure	• {1}：実行されたオペレーション • {2}：振り分けルール名 • {3}：振り分けルールバージョン

付録 F.2　コマンドに関係する監査ログ

(1)　ciwchgdef（条件・ルール・作業アプリケーション変更）コマンドの場合

表 F-4　監査ログのメッセージの出力個所（ciwchgdef コマンドの場合）

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
形式 1 形式 2 形式 3	ContentAccess	Update	Occurrence	KDIW51006-I
				{0} executed the command ({1}). • {0}：ユーザ • {1}：実行されたコマンド
			Success	KDIW51019-I
				The command ({0}) executed by {1} was successful (PDN:{2}, PDV:{3,number,#}, DefType:{4}, DefName:{5}). • {0}：実行されたコマンド • {1}：ユーザ • {2}：ビジネスプロセス定義名 • {3}：ビジネスプロセス定義バージョン • {4}：定義種別※1 • {5}：定義名
			Success	KDIW51020-I
				The command ({0}) executed by {1} was successful (PDN:{2}, DefType:{3}, DefName:{4}). • {0}：実行されたコマンド • {1}：ユーザ • {2}：ビジネスプロセス定義名 • {3}：定義種別※2 • {4}：定義名
			Failure	KDIW51008-E

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
形式 1 形式 2 形式 3	ContentAccess	Update	Failure	An attempt by {0} to execute the command ({1}) has failed. • {0}：ユーザ • {1}：実行されたコマンド

注※1

ビジネスプロセス定義バージョンの指定（-bpv）がある場合，定義種別は次のどれかです。

Condition：-cdn オプションを指定したとき

CreationRule：-crrn オプションを指定したとき

Application：-apn オプションを指定したとき

注※2

ビジネスプロセス定義バージョンの指定（-bpv）がない場合，定義種別は次のどれかです。

Condition：-cdn オプションを指定したとき

CreationRule：-crrn オプションを指定したとき

Application：-apn オプションを指定したとき

(2) ciwchgenv（システム共通環境情報の変更）コマンドの場合

表 F-5 監査ログのメッセージの出力個所（ciwchgenv コマンドの場合）

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
形式 1	ConfigurationAccess	Update	Occurrence	KDIW51000-I
				{0} executed the command ({1}). • {0}：ユーザ • {1}：実行されたコマンド
			Success	KDIW51001-I
				The command ({0}) executed by {1} was successful. • {0}：実行されたコマンド • {1}：ユーザ
			Failure	KDIW51002-E
				An attempt by {0} to execute the command ({1}) has failed. • {0}：ユーザ • {1}：実行されたコマンド
形式 2	ConfigurationAccess	Refer	Occurrence	KDIW51003-I
				{0} executed the command ({1}).

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
形式 2	ConfigurationAccess	Refer	Occurrence	• {0}：ユーザ • {1}：実行されたコマンド
			Success	KDIW51004-I
				The command ({0}) executed by {1} was successful. • {0}：実行されたコマンド • {1}：ユーザ
			Failure	KDIW51005-E
				An attempt by {0} to execute the command ({1}) has failed. • {0}：ユーザ • {1}：実行されたコマンド

(3) ciwcleanup（プロセス情報の正常化）コマンドの場合

表 F-6 監査ログのメッセージの出力個所（ciwcleanup コマンドの場合）

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
形式 1	ConfigurationAccess	Update	Occurrence	KDIW51000-I
				{0} executed the command ({1}). • {0}：ユーザ • {1}：実行されたコマンド
			Success	KDIW51001-I
				The command ({0}) executed by {1} was successful. • {0}：実行されたコマンド • {1}：ユーザ
			Failure	KDIW51002-E
				An attempt by {0} to execute the command ({1}) has failed. • {0}：ユーザ • {1}：実行されたコマンド
形式 1（-del あり）	ConfigurationAccess	Delete	Occurrence	KDIW51000-I
				{0} executed the command ({1}). • {0}：ユーザ

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
形式 1（-del あり）	ConfigurationAccess	Delete	Occurrence	• {1}：実行されたコマンド
			Success	KDIW51001-I
				The command ({0}) executed by {1} was successful. • {0}：実行されたコマンド • {1}：ユーザ
			Failure	KDIW51002-E
				An attempt by {0} to execute the command ({1}) has failed. • {0}：ユーザ • {1}：実行されたコマンド
形式 4	ConfigurationAccess	Delete	Occurrence	KDIW51000-I
				{0} executed the command ({1}). • {0}：ユーザ • {1}：実行されたコマンド
			Success	KDIW51001-I
				The command ({0}) executed by {1} was successful. • {0}：実行されたコマンド • {1}：ユーザ
			Failure	KDIW51002-E
				An attempt by {0} to execute the command ({1}) has failed. • {0}：ユーザ • {1}：実行されたコマンド

(4) ciwdelpi（案件の削除）コマンドの場合

表 F–7 監査ログのメッセージの出力個所（ciwdelpi コマンドの場合）

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
形式 1	ContentAccess	Delete	Occurrence	KDIW51006-I
				{0} executed the command ({1}). • {0}：ユーザ • {1}：実行されたコマンド

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
形式 1	ContentAccess	Delete	Success	KDIW51007-I
				The command ({0}) executed by {1} was successful. • {0}：実行されたコマンド • {1}：ユーザ
			Failure	KDIW51008-E
				An attempt by {0} to execute the command ({1}) has failed. • {0}：ユーザ • {1}：実行されたコマンド
形式 2	ContentAccess	Delete	Occurrence	KDIW51006-I
				{0} executed the command ({1}). • {0}：ユーザ • {1}：実行されたコマンド
			Success	KDIW51007-I
				The command ({0}) executed by {1} was successful. • {0}：実行されたコマンド • {1}：ユーザ
			Failure	KDIW51008-E
				An attempt by {0} to execute the command ({1}) has failed. • {0}：ユーザ • {1}：実行されたコマンド
形式 3	ContentAccess	Delete	Occurrence	KDIW51006-I
				{0} executed the command ({1}). • {0}：ユーザ • {1}：実行されたコマンド
			Success	KDIW51007-I
				The command ({0}) executed by {1} was successful. • {0}：実行されたコマンド • {1}：ユーザ
			Failure	KDIW51008-E
				An attempt by {0} to execute the command ({1}) has failed.

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
形式 3	ContentAccess	Delete	Failure	• {0}：ユーザ • {1}：実行されたコマンド

(5) ciweditbp（ビジネスプロセス定義の変更）コマンドの場合

表 F-8 監査ログのメッセージの出力個所（ciweditbp コマンドの場合）

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
形式 1	ContentAccess	Refer	Occurrence	KDIW51009-I
				{0} executed the command ({1}). • {0}：ユーザ • {1}：実行されたコマンド
			Success	KDIW51016-I
				The command ({0}) executed by {1} was successful (PDN:{2}, PDV:{3,number,#}). • {0}：実行されたコマンド • {1}：ユーザ • {2}：ビジネスプロセス定義名 • {3}：ビジネスプロセス定義バージョン
			Failure	KDIW51011-E
				An attempt by {0} to execute the command ({1}) has failed. • {0}：ユーザ • {1}：実行されたコマンド
形式 2	ContentAccess	Update	Occurrence	KDIW51006-I
				{0} executed the command ({1}). • {0}：ユーザ • {1}：実行されたコマンド
			Success	KDIW51015-I
				The command ({0}) executed by {1} was successful (PDN:{2}, PDV:{3,number,#}). • {0}：実行されたコマンド • {1}：ユーザ • {2}：ビジネスプロセス定義名 • {3}：ビジネスプロセス定義バージョン

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
形式 2	ContentAccess	Update	Failure	KDIW51008-E
				An attempt by {0} to execute the command ({1}) has failed. • {0}：ユーザ • {1}：実行されたコマンド
形式 3	ContentAccess	Update	Occurrence	KDIW51006-I
				{0} executed the command ({1}). • {0}：ユーザ • {1}：実行されたコマンド
			Success	KDIW51015-I
				The command ({0}) executed by {1} was successful (PDN:{2}, PDV:{3,number,#}). • {0}：実行されたコマンド • {1}：ユーザ • {2}：ビジネスプロセス定義名 • {3}：ビジネスプロセス定義バージョン
			Failure	KDIW51008-E
				An attempt by {0} to execute the command ({1}) has failed. • {0}：ユーザ • {1}：実行されたコマンド

(6) ciwlistsid（システム ID 情報の表示）コマンドの場合

表 F-9 監査ログのメッセージの出力個所（ciwlistsid コマンドの場合）

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
形式 1	ConfigurationAccess	Refer	Occurrence	KDIW51003-I
				{0} executed the command ({1}). • {0}：ユーザ • {1}：実行されたコマンド
			Success	KDIW51004-I
				The command ({0}) executed by {1} was successful. • {0}：実行されたコマンド

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
形式 1	ConfigurationAccess	Refer	Success	• {1}：ユーザ
			Failure	KDIW51005-E
				An attempt by {0} to execute the command ({1}) has failed. • {0}：ユーザ • {1}：実行されたコマンド
形式 2	ConfigurationAccess	Refer	Occurrence	KDIW51003-I
				{0} executed the command ({1}). • {0}：ユーザ • {1}：実行されたコマンド
			Success	KDIW51004-I
				The command ({0}) executed by {1} was successful. • {0}：実行されたコマンド • {1}：ユーザ
			Failure	KDIW51005-E
				An attempt by {0} to execute the command ({1}) has failed. • {0}：ユーザ • {1}：実行されたコマンド

(7) ciwmngbp（ビジネスプロセス定義の管理）コマンドの場合

表 F-10　監査ログのメッセージの出力個所（ciwmngbp コマンドの場合）

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
形式 1	ContentAccess	Add	Occurrence	KDIW51006-I
				{0} executed the command ({1}). • {0}：ユーザ • {1}：実行されたコマンド
			Success	KDIW51015-I
				The command ({0}) executed by {1} was successful (PDN:{2}, PDV:{3,number,#}). • {0}：実行されたコマンド • {1}：ユーザ

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
形式 1	ContentAccess	Add	Success	• {2}：ビジネスプロセス定義名 • {3}：ビジネスプロセス定義バージョン
			Failure	KDIW51008-E
				An attempt by {0} to execute the command ({1}) has failed. • {0}：ユーザ • {1}：実行されたコマンド
形式 2	ContentAccess	Refer	Occurrence	KDIW51009-I
				{0} executed the command ({1}). • {0}：ユーザ • {1}：実行されたコマンド
			Success	KDIW51016-I
				The command ({0}) executed by {1} was successful (PDN:{2}, PDV:{3,number,#}). • {0}：実行されたコマンド • {1}：ユーザ • {2}：ビジネスプロセス定義名 • {3}：ビジネスプロセス定義バージョン
			Failure	KDIW51011-E
				An attempt by {0} to execute the command ({1}) has failed. • {0}：ユーザ • {1}：実行されたコマンド
形式 3	ContentAccess	Update	Occurrence	KDIW51006-I
				{0} executed the command ({1}). • {0}：ユーザ • {1}：実行されたコマンド
			Success	KDIW51015-I
				The command ({0}) executed by {1} was successful (PDN:{2}, PDV:{3,number,#}). • {0}：実行されたコマンド • {1}：ユーザ • {2}：ビジネスプロセス定義名 • {3}：ビジネスプロセス定義バージョン
			Failure	KDIW51008-E

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
形式 3	ContentAccess	Update	Failure	An attempt by {0} to execute the command ({1}) has failed. • {0}：ユーザ • {1}：実行されたコマンド
形式 4	ContentAccess	Delete	Occurrence	KDIW51006-I
				{0} executed the command ({1}). • {0}：ユーザ • {1}：実行されたコマンド
			Success	KDIW51015-I
				The command ({0}) executed by {1} was successful (PDN:{2}, PDV:{3,number,#}). • {0}：実行されたコマンド • {1}：ユーザ • {2}：ビジネスプロセス定義名 • {3}：ビジネスプロセス定義バージョン
			Failure	KDIW51008-E
				An attempt by {0} to execute the command ({1}) has failed. • {0}：ユーザ • {1}：実行されたコマンド

(8) ciwmngcr（振り分けルール定義の管理）コマンドの場合

表 F–11　監査ログのメッセージの出力個所（ciwmngcr コマンドの場合）

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
形式 1	ContentAccess	Add	Occurrence	KDIW51006-I
				{0} executed the command ({1}). • {0}：ユーザ • {1}：実行されたコマンド
			Success	KDIW51017-I
				The command ({0}) executed by {1} was successful (CRN:{2}, CRV:{3,number,#}). • {0}：実行されたコマンド • {1}：ユーザ

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
形式 1	ContentAccess	Add	Success	• {2}：振り分けルール名 • {3}：振り分けルールバージョン
			Failure	KDIW51008-E
				An attempt by {0} to execute the command ({1}) has failed. • {0}：ユーザ • {1}：実行されたコマンド
形式 2	ContentAccess	Refer	Occurrence	KDIW51009-I
				{0} executed the command ({1}). • {0}：ユーザ • {1}：実行されたコマンド
			Success	KDIW51018-I
				The command ({0}) executed by {1} was successful (CRN:{2}, CRV:{3,number,#}). • {0}：実行されたコマンド • {1}：ユーザ • {2}：振り分けルール名 • {3}：振り分けルールバージョン
			Failure	KDIW51011-E
				An attempt by {0} to execute the command ({1}) has failed. • {0}：ユーザ • {1}：実行されたコマンド
形式 3	ContentAccess	Update	Occurrence	KDIW51006-I
				{0} executed the command ({1}). • {0}：ユーザ • {1}：実行されたコマンド
			Success	KDIW51017-I
				The command ({0}) executed by {1} was successful (CRN:{2}, CRV:{3,number,#}). • {0}：実行されたコマンド • {1}：ユーザ • {2}：振り分けルール名 • {3}：振り分けルールバージョン
			Failure	KDIW51008-E

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
形式 3	ContentAccess	Update	Failure	An attempt by {0} to execute the command ({1}) has failed. • {0}：ユーザ • {1}：実行されたコマンド
形式 4	ContentAccess	Delete	Occurrence	KDIW51006-I
				{0} executed the command ({1}). • {0}：ユーザ • {1}：実行されたコマンド
			Success	KDIW51017-I
				The command ({0}) executed by {1} was successful (CRN:{2}, CRV:{3,number,#}). • {0}：実行されたコマンド • {1}：ユーザ • {2}：振り分けルール名 • {3}：振り分けルールバージョン
			Failure	KDIW51008-E
				An attempt by {0} to execute the command ({1}) has failed. • {0}：ユーザ • {1}：実行されたコマンド

(9) ciwreuseid（ID の再利用）コマンドの場合

表 F-12 監査ログのメッセージの出力個所（ciwreuseid コマンドの場合）

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
形式 1	ConfigurationAccess	Update	Occurrence	KDIW51000-I
				{0} executed the command ({1}). • {0}：ユーザ • {1}：実行されたコマンド
			Success	KDIW51001-I
				The command ({0}) executed by {1} was successful. • {0}：実行されたコマンド • {1}：ユーザ

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
形式 1	ConfigurationAccess	Update	Failure	KDIW51002-E
				An attempt by {0} to execute the command ({1}) has failed. • {0}：ユーザ • {1}：実行されたコマンド
形式 2	ConfigurationAccess	Refer	Occurrence	KDIW51003-I
				{0} executed the command ({1}). • {0}：ユーザ • {1}：実行されたコマンド
			Success	KDIW51004-I
				The command ({0}) executed by {1} was successful. • {0}：実行されたコマンド • {1}：ユーザ
			Failure	KDIW51005-E
				An attempt by {0} to execute the command ({1}) has failed. • {0}：ユーザ • {1}：実行されたコマンド

(10) ciwsetenv（環境の構築または削除）コマンドの場合

表 F-13 監査ログのメッセージの出力個所（ciwsetenv コマンドの場合）

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
形式 1	ConfigurationAccess	Add	Occurrence	KDIW51000-I
				{0} executed the command ({1}). • {0}：ユーザ • {1}：実行されたコマンド
			Success	KDIW51001-I
				The command ({0}) executed by {1} was successful. • {0}：実行されたコマンド • {1}：ユーザ
			Failure	KDIW51002-E

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
形式 1	ConfigurationAccess	Add	Failure	An attempt by {0} to execute the command ({1}) has failed. • {0}：ユーザ • {1}：実行されたコマンド
形式 2	ConfigurationAccess	Delete	Occurrence	KDIW51000-I
				{0} executed the command ({1}). • {0}：ユーザ • {1}：実行されたコマンド
			Success	KDIW51001-I
				The command ({0}) executed by {1} was successful. • {0}：実行されたコマンド • {1}：ユーザ
			Failure	KDIW51002-E
				An attempt by {0} to execute the command ({1}) has failed. • {0}：ユーザ • {1}：実行されたコマンド
形式 3	Maintenance	Update	Occurrence	KDIW51012-I
				{0} executed the command ({1}). • {0}：ユーザ • {1}：実行されたコマンド
			Success	KDIW51013-I
				The command ({0}) executed by {1} was successful. • {0}：実行されたコマンド • {1}：ユーザ
			Failure	KDIW51014-E
				An attempt by {0} to execute the command ({1}) has failed. • {0}：ユーザ • {1}：実行されたコマンド

付録 F.3　案件運用操作に関係する監査ログ

(1)　ログインまたはログアウト

表 F-14　監査ログのメッセージの出力個所（案件運用操作（ログインまたはログアウトの場合））

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
ログイン	Authentication	Login	Success	KDIW52000-I
				{0} successfully logged in to uCosminexus Service Coordinator Interactive Workflow Process Instance Administrator. • {0}：ユーザ
			Failure	KDIW52001-E
				An attempt by {0} to log in to uCosminexus Service Coordinator Interactive Workflow Process Instance Administrator has failed. • {0}：ユーザ
ログアウト	Authentication	Logout	Success	KDIW52002-I
				{0} successfully logged out of uCosminexus Service Coordinator Interactive Workflow Process Instance Administrator. • {0}：ユーザ

(2)　案件の場合

表 F-15　監査ログのメッセージの出力個所（案件運用操作（案件の場合））

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
投入と開始	ContentAccess	Add	Occurrence	KDIW52003-I
				{0} executed the operation ({1}, PDN:{2}, PDV:{3,number,#}). • {0}：ユーザ • {1}：実行されたオペレーション PI_CRATE：案件の生成 PI_CRATE_AND_START：案件の生成と開始 • {2}：ビジネスプロセス定義名 • {3}：ビジネスプロセス定義バージョン
			Success	KDIW52004-I

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
投入と開始	ContentAccess	Add	Success	The operation ({0}, PDN:{1}, PDV:{2,number,#}, PIID:{3,number,#}) executed by {4} was successful. • {0}：実行されたオペレーション PI_CRATE：案件の生成 PI_CRATE_AND_START：案件の生成と開始 • {1}：ビジネスプロセス定義名 • {2}：ビジネスプロセス定義バージョン • {3}：案件 ID • {4}：ユーザ
			Failure	KDIW52005-E
				An attempt by {0} to execute the operation ({1}, PD N:{2}, PDV:{3,number,#}) has failed. • {0}：ユーザ • {1}：実行されたオペレーション PI_CRATE：案件の生成 PI_CRATE_AND_START：案件の生成と開始 • {2}：ビジネスプロセス定義名 • {3}：ビジネスプロセス定義バージョン
開始	ContentAccess	Update	Occurrence	KDIW52006-I
				{0} executed the operation ({1}, PIID: {2,number,#}). • {0}：ユーザ • {1}：実行されたオペレーション • {2}：案件 ID
			Success	KDIW52007-I
				The operation ({0}, PIID:{1,number,#}) executed by {2} was successful. • {0}：実行されたオペレーション • {1}：案件 ID • {2}：ユーザ
			Failure	KDIW52008-E
				An attempt by {0} to execute the operation ({1}, PIID:{2,number,#}) has failed. • {0}：ユーザ • {1}：実行されたオペレーション • {2}：案件 ID
中断	ContentAccess	Update	Occurrence	KDIW52006-I

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
中断	ContentAccess	Update	Occurrence	{0} executed the operation ({1}, PIID:{2,number,#}). • {0}：ユーザ • {1}：実行されたオペレーション • {2}：案件 ID
			Success	KDIW52007-I
				The operation ({0}, PIID:{1,number,#}) executed by {2} was successful. • {0}：実行されたオペレーション • {1}：案件 ID • {2}：ユーザ
			Failure	KDIW52008-E
				An attempt by {0} to execute the operation ({1}, PIID:{2,number,#}) has failed. • {0}：ユーザ • {1}：実行されたオペレーション • {2}：案件 ID
再開	ContentAccess	Update	Occurrence	KDIW52006-I
				{0} executed the operation ({1}, PIID:{2,number,#}). • {0}：ユーザ • {1}：実行されたオペレーション • {2}：案件 ID
			Success	KDIW52007-I
				The operation ({0}, PIID:{1,number,#}) executed by {2} was successful. • {0}：実行されたオペレーション • {1}：案件 ID • {2}：ユーザ
			Failure	KDIW52008-E
				An attempt by {0} to execute the operation ({1}, PIID:{2,number,#}) has failed. • {0}：ユーザ • {1}：実行されたオペレーション • {2}：案件 ID
強制終了	ContentAccess	Update	Occurrence	KDIW52006-I

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
強制終了	ContentAccess	Update	Occurrence	{0} executed the operation ({1}, PIID:{2,number,#}). • {0}：ユーザ • {1}：実行されたオペレーション • {2}：案件 ID
			Success	KDIW52007-I
				The operation ({0}, PIID:{1,number,#}) executed by {2} was successful. • {0}：実行されたオペレーション • {1}：案件 ID • {2}：ユーザ
			Failure	KDIW52008-E
				An attempt by {0} to execute the operation ({1}, PIID:{2,number,#}) has failed. • {0}：ユーザ • {1}：実行されたオペレーション • {2}：案件 ID
削除	ContentAccess	Delete	Occurrence	KDIW52006-I
				{0} executed the operation ({1}, PIID:{2,number,#}). • {0}：ユーザ • {1}：実行されたオペレーション • {2}：案件 ID
			Success	KDIW52007-I
				The operation ({0}, PIID:{1,number,#}) executed by {2} was successful. • {0}：実行されたオペレーション • {1}：案件 ID • {2}：ユーザ
			Failure	KDIW52008-E
				An attempt by {0} to execute the operation ({1}, PIID:{2,number,#}) has failed. • {0}：ユーザ • {1}：実行されたオペレーション • {2}：案件 ID

(3) 業務ステップの場合

表 F-16 監査ログのメッセージの出力個所（案件運用操作（業務ステップの場合））

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
活性化	ContentAccess	Update	Occurrence	KDIW52009-I
				{0} executed the operation ({1}, PIID:{2,number,#}, AIID:{3,number,#}). • {0}：ユーザ • {1}：実行されたオペレーション • {2}：案件 ID • {3}：業務ステップ ID
			Success	KDIW52010-I
				The operation ({0}, PIID:{1,number,#}, AIID:{2,number,#}) executed by {3} was successful. • {0}：実行されたオペレーション • {1}：案件 ID • {2}：業務ステップ ID • {3}：ユーザ
			Failure	KDIW52011-E
				An attempt by {0} to execute the operation ({1}, PIID:{2,number,#}, AIID:{3,number,#}) has failed. • {0}：ユーザ • {1}：実行されたオペレーション • {2}：案件 ID • {3}：業務ステップ ID
中断	ContentAccess	Update	Occurrence	KDIW52009-I
				{0} executed the operation ({1}, PIID:{2,number,#}, AIID:{3,number,#}). • {0}：ユーザ • {1}：実行されたオペレーション • {2}：案件 ID • {3}：業務ステップ ID
			Success	KDIW52010-I
				The operation ({0}, PIID:{1,number,#}, AIID:{2,number,#}) executed by {3} was successful. • {0}：実行されたオペレーション • {1}：案件 ID • {2}：業務ステップ ID

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
中断	ContentAccess	Update	Success	• {3}：ユーザ
			Failure	KDIW52011-E
				An attempt by {0} to execute the operation ({1}, PIID:{2,number,#}, AIID:{3,number,#}) has failed. • {0}：ユーザ • {1}：実行されたオペレーション • {2}：案件 ID • {3}：業務ステップ ID
再開	ContentAccess	Update	Occurrence	KDIW52009-I
				{0} executed the operation ({1}, PIID:{2,number,#}, AIID:{3,number,#}). • {0}：ユーザ • {1}：実行されたオペレーション • {2}：案件 ID • {3}：業務ステップ ID
			Success	KDIW52010-I
				The operation ({0}, PIID:{1,number,#}, AIID:{2,number,#}) executed by {3} was successful. • {0}：実行されたオペレーション • {1}：案件 ID • {2}：業務ステップ ID • {3}：ユーザ
			Failure	KDIW52011-E
				An attempt by {0} to execute the operation ({1}, PIID:{2,number,#}, AIID:{3,number,#}) has failed. • {0}：ユーザ • {1}：実行されたオペレーション • {2}：案件 ID • {3}：業務ステップ ID
完了	ContentAccess	Update	Occurrence	KDIW52009-I
				{0} executed the operation ({1}, PIID:{2,number,#}, AIID:{3,number,#}). • {0}：ユーザ • {1}：実行されたオペレーション • {2}：案件 ID • {3}：業務ステップ ID

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
完了	ContentAccess	Update	Success	KDIW52010-I
				The operation ({0}, PIID:{1,number,#}, AIID:{2,number,#}) executed by {3} was successful. • {0}：実行されたオペレーション • {1}：案件 ID • {2}：業務ステップ ID • {3}：ユーザ
			Failure	KDIW52011-E
				An attempt by {0} to execute the operation ({1}, PIID:{2,number,#}, AIID:{3,number,#}) has failed. • {0}：ユーザ • {1}：実行されたオペレーション • {2}：案件 ID • {3}：業務ステップ ID
強制終了	ContentAccess	Update	Occurrence	KDIW52009-I
				{0} executed the operation ({1}, PIID:{2,number,#}, AIID:{3,number,#}). • {0}：ユーザ • {1}：実行されたオペレーション • {2}：案件 ID • {3}：業務ステップ ID
			Success	KDIW52010-I
				The operation ({0}, PIID:{1,number,#}, AIID:{2,number,#}) executed by {3} was successful. • {0}：実行されたオペレーション • {1}：案件 ID • {2}：業務ステップ ID • {3}：ユーザ
			Failure	KDIW52011-E
				An attempt by {0} to execute the operation ({1}, PIID:{2,number,#}, AIID:{3,number,#}) has failed. • {0}：ユーザ • {1}：実行されたオペレーション • {2}：案件 ID • {3}：業務ステップ ID

(4) 作業の場合

表 F-17 監査ログのメッセージの出力個所（案件運用操作（作業の場合））

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
着手	ContentAccess	Update	Occurrence	KDIW52012-I
				{0} executed the operation ({1}, PIID:{2,number,#}, WIID:{3,number,#}). • {0}：ユーザ • {1}：実行されたオペレーション • {2}：案件 ID • {3}：作業 ID
			Success	KDIW52013-I
				The operation ({0}, PIID:{1,number,#}, WIID:{2,number,#}) executed by {3} was successful. • {0}：実行されたオペレーション • {1}：案件 ID • {2}：作業 ID • {3}：ユーザ
			Failure	KDIW52014-E
				An attempt by {0} to execute the operation ({1}, PIID:{2,number,#}, WIID:{3,number,#}) has failed. • {0}：ユーザ • {1}：実行されたオペレーション • {2}：案件 ID • {3}：作業 ID
中断	ContentAccess	Update	Occurrence	KDIW52012-I
				{0} executed the operation ({1}, PIID:{2,number,#}, WIID:{3,number,#}). • {0}：ユーザ • {1}：実行されたオペレーション • {2}：案件 ID • {3}：作業 ID
			Success	KDIW52013-I
				The operation ({0}, PIID:{1,number,#}, WIID:{2,number,#}) executed by {3} was successful. • {0}：実行されたオペレーション • {1}：案件 ID • {2}：作業 ID

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
中断	ContentAccess	Update	Success	• {3}：ユーザ
			Failure	KDIW52014-E
				An attempt by {0} to execute the operation ({1}, PIID:{2,number,#}, WIID:{3,number,#}) has failed. • {0}：ユーザ • {1}：実行されたオペレーション • {2}：案件 ID • {3}：作業 ID
再開	ContentAccess	Update	Occurrence	KDIW52012-I
				{0} executed the operation ({1}, PIID:{2,number,#}, WIID:{3,number,#}). • {0}：ユーザ • {1}：実行されたオペレーション • {2}：案件 ID • {3}：作業 ID
			Success	KDIW52013-I
				The operation ({0}, PIID:{1,number,#}, WIID:{2,number,#}) executed by {3} was successful. • {0}：実行されたオペレーション • {1}：案件 ID • {2}：作業 ID • {3}：ユーザ
			Failure	KDIW52014-E
				An attempt by {0} to execute the operation ({1}, PIID:{2,number,#}, WIID:{3,number,#}) has failed. • {0}：ユーザ • {1}：実行されたオペレーション • {2}：案件 ID • {3}：作業 ID
完了	ContentAccess	Update	Occurrence	KDIW52012-I
				{0} executed the operation ({1}, PIID:{2,number,#}, WIID:{3,number,#}). • {0}：ユーザ • {1}：実行されたオペレーション • {2}：案件 ID • {3}：作業 ID

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
完了	ContentAccess	Update	Success	KDIW52013-I
				The operation ({0}, PIID:{1,number,#}, WIID:{2,number,#}) executed by {3} was successful. • {0}：実行されたオペレーション • {1}：案件 ID • {2}：作業 ID • {3}：ユーザ
			Failure	KDIW52014-E
				An attempt by {0} to execute the operation ({1}, PIID:{2,number,#}, WIID:{3,number,#}) has failed. • {0}：ユーザ • {1}：実行されたオペレーション • {2}：案件 ID • {3}：作業 ID
返却または作業準備済み	ContentAccess	Update	Occurrence	KDIW52012-I
				{0} executed the operation ({1}, PIID:{2,number,#}, WIID:{3,number,#}). • {0}：ユーザ • {1}：実行されたオペレーション • {2}：案件 ID • {3}：作業 ID
			Success	KDIW52013-I
				The operation ({0}, PIID:{1,number,#}, WIID:{2,number,#}) executed by {3} was successful. • {0}：実行されたオペレーション • {1}：案件 ID • {2}：作業 ID • {3}：ユーザ
			Failure	KDIW52014-E
				An attempt by {0} to execute the operation ({1}, PIID:{2,number,#}, WIID:{3,number,#}) has failed. • {0}：ユーザ • {1}：実行されたオペレーション • {2}：案件 ID • {3}：作業 ID
強制終了	ContentAccess	Update	Occurrence	KDIW52012-I

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
強制終了	ContentAccess	Update	Occurrence	{0} executed the operation ({1}, PIID:{2,number,#}, WIID:{3,number,#}). • {0}：ユーザ • {1}：実行されたオペレーション • {2}：案件 ID • {3}：作業 ID
			Success	KDIW52013-I
				The operation ({0}, PIID:{1,number,#}, WIID:{2,number,#}) executed by {3} was successful. • {0}：実行されたオペレーション • {1}：案件 ID • {2}：作業 ID • {3}：ユーザ
			Failure	KDIW52014-E
				An attempt by {0} to execute the operation ({1}, PIID:{2,number,#}, WIID:{3,number,#}) has failed. • {0}：ユーザ • {1}：実行されたオペレーション • {2}：案件 ID • {3}：作業 ID
作業者 再割り当て	ContentAccess	Update	Occurrence	KDIW52012-I
				{0} executed the operation ({1}, PIID:{2,number,#}, WIID:{3,number,#}). • {0}：ユーザ • {1}：実行されたオペレーション • {2}：案件 ID • {3}：作業 ID
			Success	KDIW52016-I
				The operation ({0}, PIID:{1,number,#}, WIID:{2,number,#} , PARTICIPANT:{3}) executed by {4} was successful. • {0}：実行されたオペレーション • {1}：案件 ID • {2}：作業 ID • {3}：作業者 • {4}：ユーザ
			Failure	KDIW52017-E

出力個所	監査事象	動作情報	監査事象の結果	メッセージ 上段：メッセージ ID 下段：メッセージ（自由記述）
作業者 再割り当て	ContentAccess	Update	Failure	An attempt by {0} to execute theoperation ({1}, PIID:{2,number,#}, WIID:{3,number,#}, PARTICIPANT:{4}) has failed. • {0}：ユーザ • {1}：実行されたオペレーション • {2}：案件 ID • {3}：作業 ID • {4}：作業者

付録 G　ビジネスプロセス定義および振り分けルール定義ファイル

ビジネスプロセス定義および振り分けルール定義は，ビジネスプロセス管理ツール（CSCIW-Definer）で定義したファイルをそのまま使用します。

また，CSCIW では，WorkCoordinator Definer で作成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利用できます。WorkCoordinator Definer で作成された定義ファイルは，ビジネスプロセス管理ツール（CSCIW-Definer）を使用して読み込みます。ただし，CSCIW でサポートしていない項目が定義されている場合は，読み込み時にビジネスプロセス管理ツール（CSCIW-Definer）がその項目を削除します。また，このとき削除された内容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 で登録できる機能を次に示します。

付録 G.1　CSCIW で登録できるビジネスプロセス定義の機能

WorkCoordinator Definer でサポートしているビジネスプロセス定義の機能のうち，CSCIW で登録できる機能を次の表に示します。

表 G-1　WorkCoordinator Definer で定義したビジネスプロセス定義の機能の登録可否

プロセス定義の機能		登録可否
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※1	○
	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	×

プロセス定義の機能				登録可否
組み込み作業	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)			○
	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プロパティ	バージョン			○
	処理期限ルール			○
	アドホック要求を使用する			○※2
	状態			○
	管理者			○

プロセス定義の機能		登録可否
ビジネスプロセスの プロパティ	案件投入可能期間	○
	説明	○

（凡例）

○：CSCIW のワーク管理データベースに登録できる

×：CSCIW のワーク管理データベースに登録できない

注※1　並列作業の子作業以外は，条件に「@False」を設定できません。

注※2　アドホック API を使用する場合は，CSCIW-Definer で「アドホック API の使用」をチェックする必要があります。

付録 G.2　CSCIW で登録できる振り分けルール定義の機能

WorkCoordinator Definer でサポートしている振り分けルール定義の機能のうち，CSCIW で登録できる機能を次の表に示します。

表 G–2　WorkCoordinator Definer で定義した振り分けルール定義の機能の登録可否

振り分けルール定義の機能		登録 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）

○：CSCIW のワーク管理データベースに登録できる

×：CSCIW のワーク管理データベースに登録できない

付録 H　BPMN 連携機能を使用しない場合のアプリケーション呼び出しサービス

アプリケーション呼び出しサービス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は，作業アプリケーション定義（種別がREST）に定義された業務アプリケーションを呼び出す J2EE アプリケーションです。呼び出す業務アプリケーションは，REST に基づいて作成されたアプリケーション（REST アプリケーション）です。

> **ヒント**
>
> このマニュアルでは，BPMN 連携機能を使用しない場合のアプリケーション呼び出しサービスについて説明します。BPMN 連携機能を使用する場合のアプリケーション呼び出しサービスについては，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。

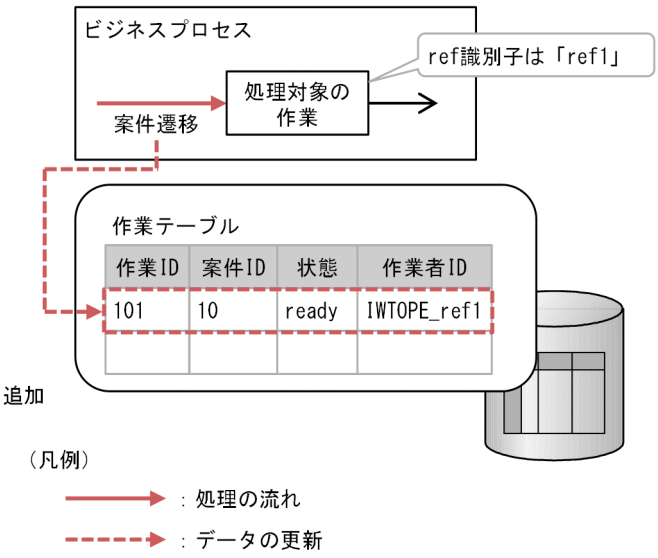
付録 H.1　アプリケーション呼び出しまでの処理の流れ

アプリケーション呼び出しが行われるまでの処理の流れを説明します。

(1)　CSCIW のライブラリの処理の流れ

アプリケーション呼び出しサービスの処理対象の作業が，作業テーブルに追加されるまでの CSCIW のライブラリの処理の流れを，次の図に示します。

図 H-1　CSCIW のライブラリの処理の流れ



＜説明＞

業務アプリケーションまたはアプリケーション呼び出しサービスが，アプリケーション呼び出しサービスの処理対象の作業の 1 つ前の作業に対して遷移要求すると，CSCIW のライブラリは案件を遷移させます（図中の「案件遷移」）。

その際，CSCIW のライブラリは，アプリケーション呼び出しサービスの処理対象の作業（図では作業 ID=101 の作業）を作業テーブルに追加します。また，作業の作業者 ID に「IWTOPE_ref1」※を設定します。

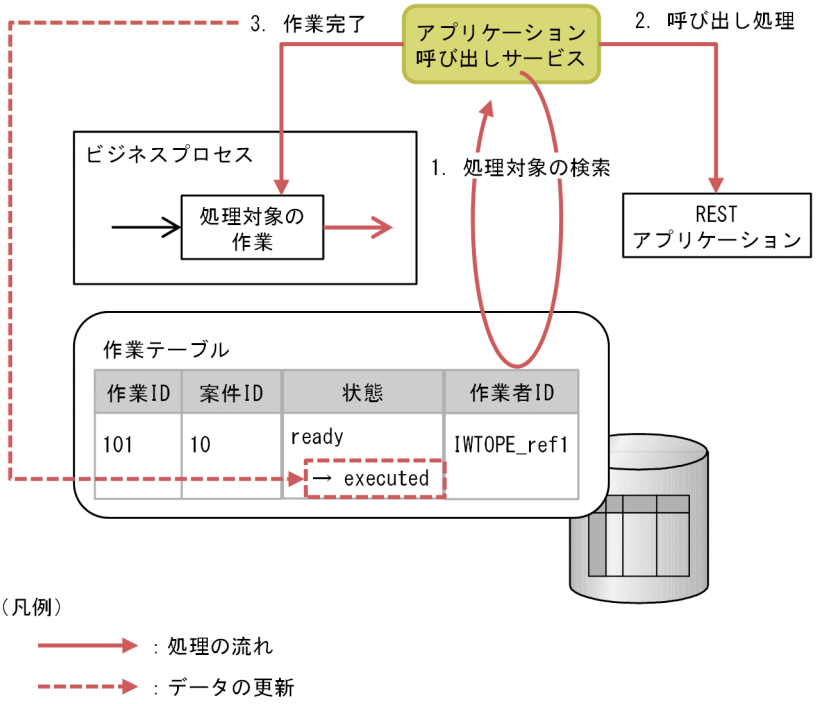
注※

ref 識別子が「ref1」の場合の例です。

(2)　アプリケーション呼び出しサービスの処理の流れ

アプリケーション呼び出しサービスの処理の流れを，次の図に示します。

図 H-2　アプリケーション呼び出しサービスの処理の流れ



＜説明＞

アプリケーション呼び出しサービスは，定期的に次の 1.～3.の処理を行います。

1. 処理対象の ref 識別子を決め，その ref 識別子を持つ作業を作業テーブルから検索します。図の例では，次の条件で検索しています。
   - ref 識別子：ref1
   - 状態：「実行開始可能（ready）」
   - 作業者 ID：IWTOPE_ref1
2. 作業（図の例では作業 ID=101 の作業）が見つかった場合，ref 識別子に対応するアプリケーション呼び出し情報ファイルに従い，REST アプリケーションの呼び出しを行います。

3. REST アプリケーションの呼び出しが成功すると，アプリケーション呼び出しサービスは，作業の状態を「実行開始可能（ready）」から「実行済（executed）」に変更します。

付録 H.2　実行間隔とポーリング間隔

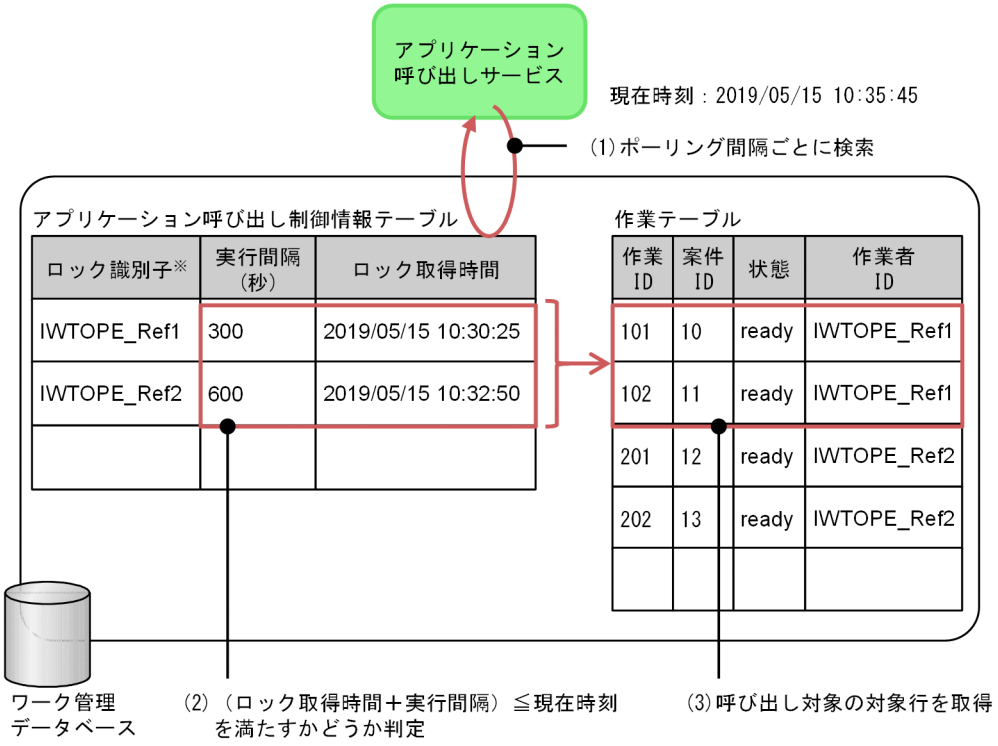
アプリケーション呼び出しサービスが処理を実行する契機は，実行間隔とポーリング間隔によって決まります。

実行間隔は，ref 識別子ごとに設定する間隔で，呼び出す必要がある作業があるかどうかを検索する間隔です。

ポーリング間隔は，アプリケーション呼び出しサービスごとに設定する間隔で，呼び出す必要がある ref 識別子があるかどうかを検索する間隔です。

次の図で，実行間隔とポーリング間隔について説明します。

図 H-3　実行間隔とポーリング間隔



注※

ref識別子に対応する作業者IDの値を指します。

＜説明＞

アプリケーション呼び出しサービスは，呼び出す必要がある ref 識別子を見つけるために，ポーリング間隔ごとにアプリケーション呼び出し制御情報テーブルを検索します（図中の(1)）。アプリケーション呼び出し制御情報テーブルには，前回呼び出し処理を行った時間（ロック取得時間）が格納されていま

す。図の例では，Ref1 は 2019/05/15 10:30:25 に，Ref2 は 2019/05/15 10:32:50 に前回の呼び出し処理を行っています。

さらに，アプリケーション呼び出しサービスは，Ref1 およびRef2 が実行間隔を過ぎているかどうかを判定します（図中の(2)）。図の例では，Ref1 の実行間隔が 300 秒，Ref2 の実行間隔が 600 秒に設定されています。ロック取得時間に対して，設定されている実行間隔を現在時刻が過ぎているRef1 が呼び出し処理の対象となります（図中の(3)）。

アプリケーション呼び出しサービスは，呼び出し処理の対象となったRef1 を作業者 ID に持ち，状態が「実行開始可能（ready）」の作業に対して，呼び出し処理を行います。

実行間隔は，ref 識別子単位で設定できます。アプリケーション呼び出し制御情報の「実行間隔(ExecuteInterval)」に設定してください。アプリケーション呼び出し制御情報の詳細については，マニュアル「uCosminexus Service Coordinator Interactive Workflow コマンド」の「ciwmngap（アプリケーション呼び出し制御情報の管理）」を参照してください。

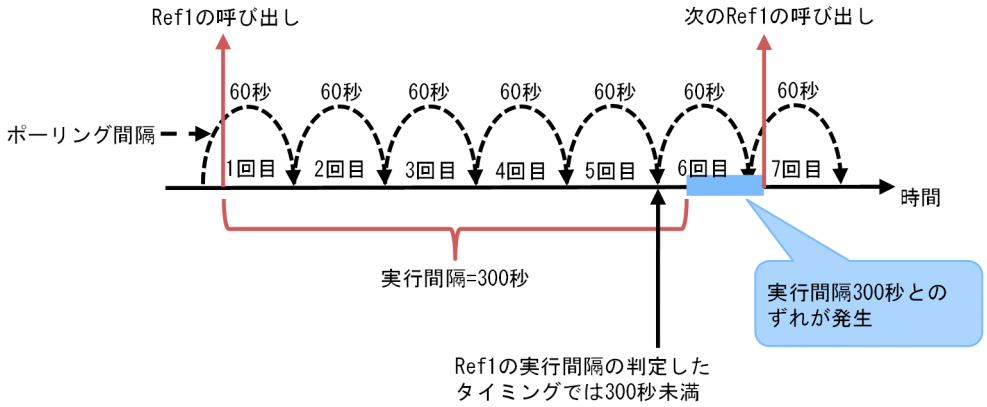
ポーリング間隔は，J2EE サーバ単位で設定できます。共通設定ファイルの「AppCallServicePollingInterval」に設定してください。AppCallServicePollingInterval の詳細については，「付録 H.11(5)　共通設定ファイルに指定するパラメタ一覧および内容」の「(c) AppCallServicePollingInterval」を参照してください。

メモ

アプリケーション呼び出しサービスが実際に作業を呼び出す間隔は，実行間隔の設定値から最大でポーリング間隔分のずれが発生します。ポーリング間隔を小さくするほど，実行間隔の設定値と実際に作業を呼び出す間隔とのずれは小さくなりますが，その分アプリケーション呼び出し制御情報テーブルの検索実行回数が多くなります。実行間隔とポーリング間隔のずれを次の図に示します。

図の例では，ポーリング間隔 60 秒ごとに，アプリケーション呼び出しサービスがアプリケーション呼び出し制御情報テーブルを検索しています。まず，1 回目のポーリング間隔の途中の時刻にRef1 を呼び出します。そのあと，ポーリングを繰り返し，6 回目のポーリング間隔の最初にRef1 の実行間隔経過を判定します。しかし，判定のタイミングでは，1 回目のポーリング間隔でのRef1 の呼び出しから 300 秒が経過していません。よって，次のRef1 の呼び出し処理は 7 回目のポーリング間隔の間に行われるため，実行間隔の設定値である 300 秒とのずれが発生します。

図 H-4　実行間隔とポーリング間隔のずれ



付録 H.3　呼び出しの流量制御

アプリケーション呼び出しサービスでは，呼び出す作業数を制御して，負荷を抑えることができます。呼び出しは，最大作業件数および WorkManager の最大スレッド数の設定によって制御できます。

最大作業件数

最大作業件数は，1 回の実行間隔内で発生した作業のうち，呼び出し処理を行う作業件数の最大値です。業務処理のピーク時に呼び出し対象の作業数が増大しても，呼び出し処理を実行する作業数を制限して，負荷増加を抑えることができます。呼び出し処理の対象から外れた作業が発生した場合，該当する作業は，次回の実行間隔の経過後に呼び出し処理の対象となります。

最大作業件数は，ref 識別子単位で設定できます。アプリケーション呼び出し制御情報の「最大作業件数（`WorkItemMax`）」に設定してください。詳細については，マニュアル「uCosminexus Service Coordinator Interactive Workflow コマンド」の「ciwmngap（アプリケーション呼び出し制御情報の管理）」を参照してください。

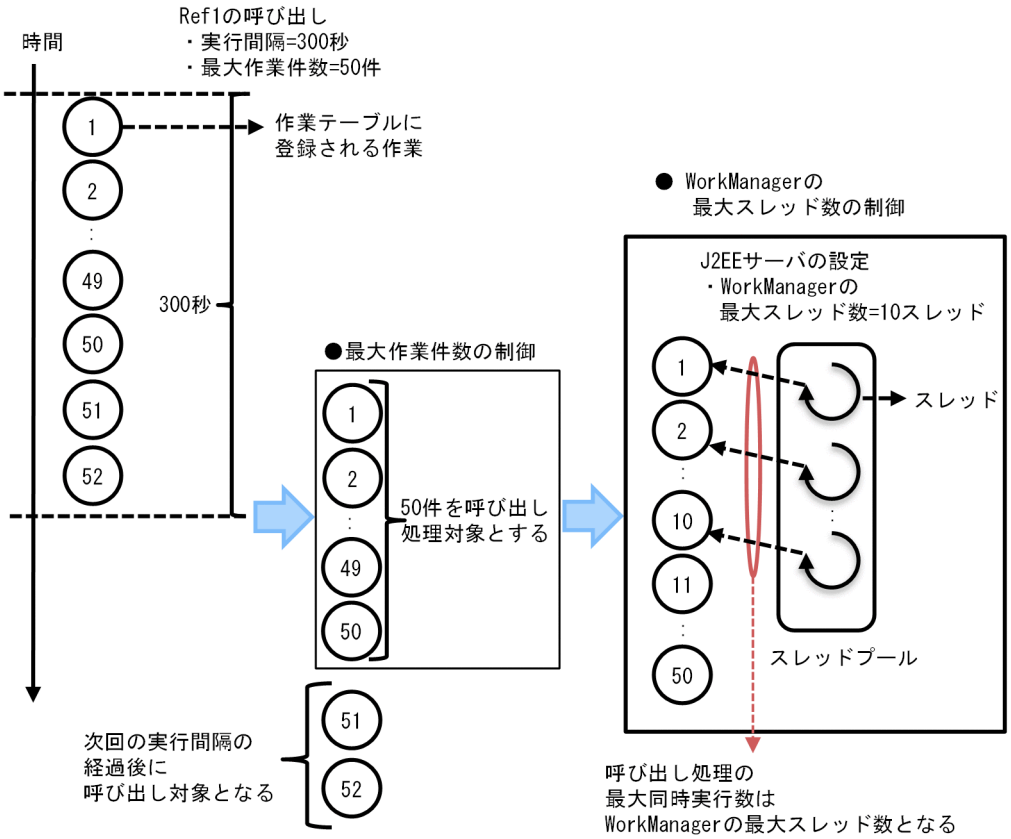
WorkManager の最大スレッド数

WorkManager の最大スレッド数は，呼び出し対象となった作業に対して，並列で同時に処理を実行するためのスレッド数の最大値です。呼び出し対象の作業数が増大しても，呼び出し処理を実行するスレッド数を制限して，負荷増加を抑えることができます。スレッドプール内の最大スレッド数で指定した数のスレッドによって，作業の呼び出し処理が行われます。

WorkManager の最大スレッド数は，J2EE サーバ単位で設定できます。J2EE サーバの設定ファイルに指定してください。詳細については，「6.2.8(3)　WorkManager の最大スレッド数を変更する」を参照してください。

最大作業件数と WorkManager の最大スレッド数による流量制御の概要を次の図に示します。

図 H-5　流量制御の概要



＜説明＞

図の例では，Ref1 の最大作業件数が50，WorkManager の最大スレッド数が10 と設定されています。

このとき，Ref1 の実行間隔内で 52 件の作業が作業テーブルに登録されたとすると，呼び出し対象となる作業の件数は最大作業件数の 50 件となります。50 件を超えた作業 2 件は，次回の実行間隔の経過後の呼び出し対象になります。

呼び出し対象となる 50 件の作業に対しては，WorkManager の最大スレッド数 10 のスレッドで同時に呼び出し処理を実行します。

付録 H.4　障害時の動作

アプリケーション呼び出しの処理中に障害が発生した場合，呼び出しが成功するまでリトライします。アプリケーション呼び出しサービス自身に障害が発生した場合は，アプリケーション呼び出しサービスを多重化していれば，ほかのアプリケーション呼び出しサービスに処理が引き継がれます。

アプリケーション呼び出しサービスでは，大きく分けて次に示す障害の発生が考えられます。

- アプリケーション呼び出し処理での障害
  - アプリケーション呼び出し情報ファイルの設定不備
  - 呼び出し先の REST アプリケーションとの通信障害が発生した場合
  - 呼び出し先の REST アプリケーションが返却したステータスコードの判定結果が失敗であった場合

- 呼び出し元作業の完了の失敗

- アプリケーション呼び出しサービス自身の障害
  - J2EE サーバプロセスのダウン

次に，これらの障害時の動作について説明します。

(1)　アプリケーション呼び出し処理での障害

アプリケーション呼び出し処理で障害が発生すると，呼び出しが成功するまでリトライします。

リトライは呼び出しが失敗したあと，リトライ間隔後に行われます。

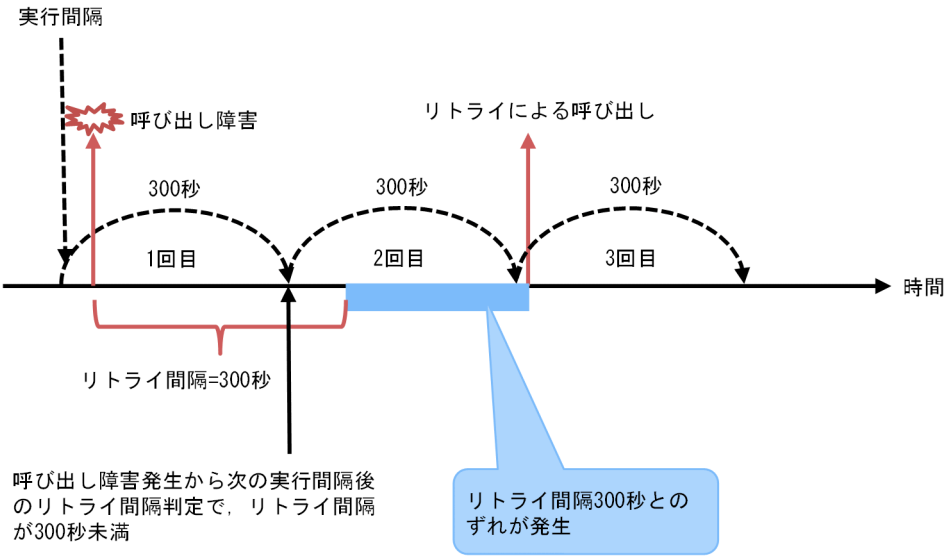
メモ

実際にリトライする間隔は，リトライ間隔の設定値から最大で実行間隔分のずれが発生します。実行間隔を小さくするほど，リトライ間隔の設定値と実際に作業をリトライする間隔とのずれは小さくなります。ただし，呼び出し処理の頻度がその分増えるため，負荷が増加します。

リトライ間隔と実行間隔の関係を次の図に示します。

図の例では，アプリケーション呼び出しサービスは，実行間隔 300 秒ごとに呼び出し処理を行っています。1 回目の実行間隔の途中で呼び出し障害が発生し，呼び出しが失敗した時刻が作業に設定されます。そして，2 回目の実行間隔の最初でリトライ間隔の判定処理が行われます。しかし，判定のタイミングでは，リトライ間隔の 300 秒が経過していません。そのため，リトライによる呼び出しは 3 回目の実行間隔の間に行われ，リトライ間隔の設定値である 300 秒とのずれが発生します。

図 H-6　リトライ間隔と実行間隔のずれ



なお，リトライした回数は，呼び出し元の作業の優先度に設定されます。アプリケーションの呼び出しに失敗した時刻は，呼び出し元の作業の処理期限に設定されます。アプリケーション呼び出しサービスは，

設定したリトライ回数だけリトライします。設定した回数分リトライしても失敗した場合は，アプリケーション呼び出しサービスのメッセージファイルに，エラーメッセージを出力します。また，呼び出し元の作業の状態を「作業者実行」に，優先度を「0」に変更します。これ以降，該当する作業はリトライの対象から外れます。

メモ

リトライ対象の作業がコマンドで「実行開始可能」状態に変更された場合，作業の優先度が「0」に変更されます。このとき，設定したリトライ回数よりも多くアプリケーション呼び出しのリトライが行われます。

リトライの対象から外れた作業を再び呼び出し対象とする場合は，障害の原因を取り除いたあとで，該当する作業の状態を「作業者実行」から「実行開始可能」に変更してください。作業の状態を「作業者実行」から「実行開始可能」に変更する方法については，「付録 H.13　リトライの対象から外れた作業に関する運用」を参照してください。

リトライ回数とリトライ間隔は，ref 識別子単位で設定できます。アプリケーション呼び出し制御情報の「リトライ回数（RetryCount)」および「リトライ間隔（RetryInterval)」に設定してください。アプリケーション呼び出し制御情報の詳細については，マニュアル「uCosminexus Service Coordinator Interactive Workflow コマンド」の「ciwmngap（アプリケーション呼び出し制御情報の管理)」を参照してください。

(2)　アプリケーション呼び出しサービス自身の障害

アプリケーション呼び出しサービス自身が障害になった場合，アプリケーション呼び出しサービスを多重化していれば，ほかのアプリケーション呼び出しサービスが代わりに呼び出し処理を継続できます。

なお，ほかのアプリケーション呼び出しサービスが代わりに呼び出し処理を行うタイミングは，障害復旧間隔が経過したあとになります。

障害復旧間隔は ref 識別子単位に設定できます。アプリケーション呼び出し制御情報の「障害復旧間隔(RecoveryTime)」に設定してください。アプリケーション呼び出し制御情報の詳細については，マニュアル「uCosminexus Service Coordinator Interactive Workflow コマンド」の「ciwmngap（アプリケーション呼び出し制御情報の管理)」を参照してください。

重要

呼び出し先の処理が長時間掛かると，次の図のように障害復旧間隔後にほかのアプリケーション呼び出しサービスが同じ作業の呼び出し処理を行ってしまいます。

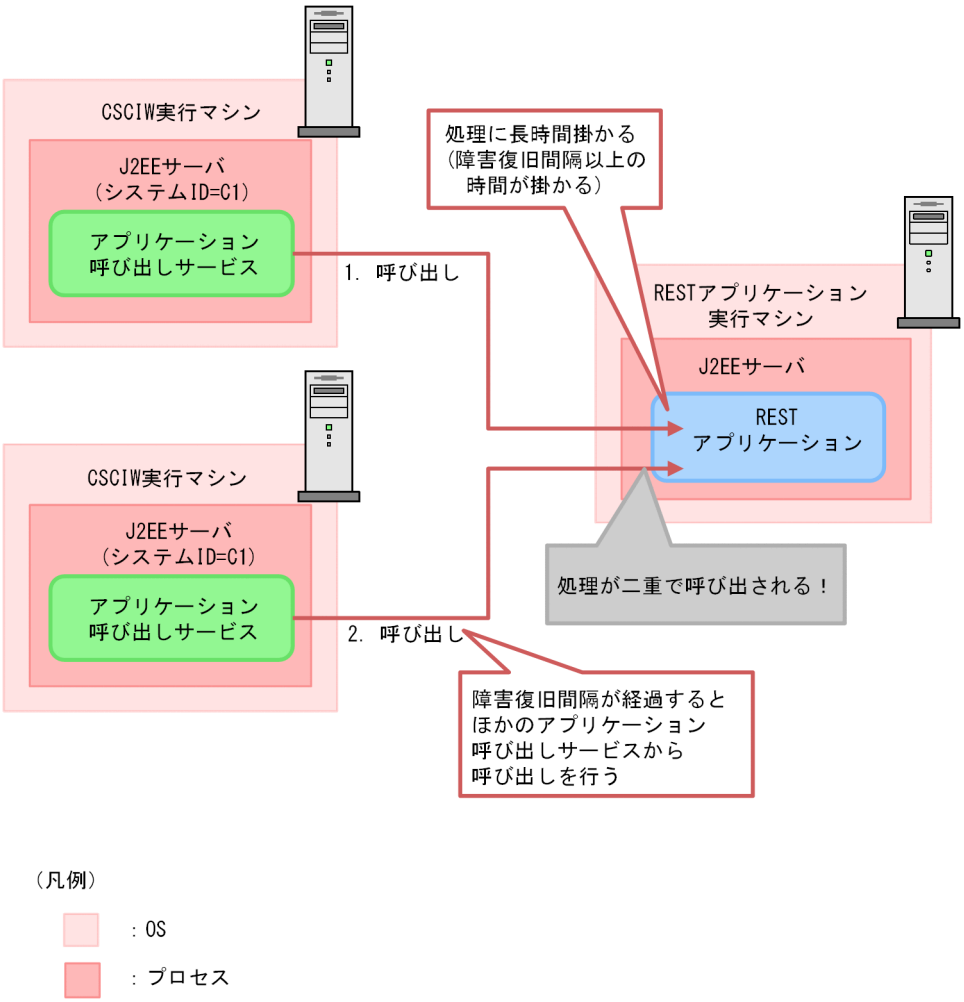
このように，二重に作業が呼び出されてしまうおそれがあるため，呼び出し先のアプリケーションで長時間掛かる処理はしないようにしてください。

障害復旧間隔は，次の式で求められる値よりも大きくしてください。

（〈呼び出し先のアプリケーションの処理時間〉×〈最大作業件数〉）÷〈WorkManagerのスレッド数〉

また，アプリケーション呼び出しの実行中にJ2EEサーバを強制停止した場合，J2EEサーバの再開始後，障害復旧間隔が経過するまでアプリケーションの呼び出しが行われないおそれがあります。このため通常の運用ではJ2EEサーバを強制停止したり，障害復旧間隔を大きくし過ぎたりしないでください。

図 H-7　RESTアプリケーションの処理時間が障害復旧間隔よりも長い場合の動作



付録 H.5　呼び出し処理のタイムアウト

呼び出し先がRESTアプリケーションの場合，アプリケーション呼び出しサービスとRESTアプリケーションの間のRESTful Webサービスの通信タイムアウトを設定できます。

通信タイムアウトの設定方法を，優先順位が高い順に示します。

1. アプリケーション呼び出し情報ファイルでの設定

アプリケーション呼び出し情報ファイルでタイムアウトを設定する場合は，ref 識別子ごとに異なるタイムアウト値を指定できます。アプリケーション呼び出し情報ファイルでの設定方法については，「付録 H.12(9)　アプリケーション呼び出し情報ファイルに指定する内容」の「(f) rest.request.read.timeout」と「(g)　rest.request.connect.timeout」を参照してください。

2. Cosminexus の J2EE サーバ単位の通信タイムアウトの設定

   Cosminexus の J2EE サーバ単位の通信タイムアウトを設定する場合は，J2EE サーバ内の RESTful Web サービスの通信で，最も長くしたいタイムアウト値を設定してください。

   Cosminexus の J2EE サーバ単位の通信タイムアウトの設定方法については，マニュアル「Cosminexus 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)」の「J2EE サーバ単位の通信タイムアウトの設定」およびマニュアル「Cosminexus 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)」の「usrconf.properties（J2EE サーバ用ユーザプロパティファイル)」を参照してください。

注意事項

通信タイムアウト（クライアントソケットの接続タイムアウト，またはクライアントソケットの読み込みタイムアウト）の設定値が，0 または指定なしの場合，REST アプリケーションが無応答になると，アプリケーション呼び出しサービスは応答を待ち続けます。そのため，上記の 1.または 2.の方法で，通信タイムアウトに0 以外の値を設定してください。

付録 H.6　呼び出しの一時抑止

アプリケーション呼び出しサービスは，障害発生時にアプリケーション呼び出しを一時的に抑止し，障害への対処後に一時抑止を解除できます。

アプリケーション呼び出しを一時抑止した場合，REST アプリケーションを呼び出しません。

また，一時抑止すると，解除するまで，リトライした回数がカウントアップされることはありません。

一時抑止と解除は次に示す単位で行えます。

- 作業単位
- アプリケーション呼び出し制御情報単位

(1)　作業単位の一時抑止と解除

作業単位でアプリケーション呼び出しを一時抑止する場合，ciwchgapwork コマンドを使用して，作業の状態を「作業者実行」に変更します。また，一時抑止を解除する場合，作業の状態を「実行開始可能」に戻します。

実行例を次に示します。

一時抑止の実行例

1. ciwchgapwork コマンドを実行して「実行開始可能」状態の作業の情報を取得します。

ciwchgapwork コマンドの指定形式を次に示します。

```
ciwchgapwork -sid システムID -list -ope -s READY > ファイルパス
```

出力結果はファイルにリダイレクトします。

2. 手順 1.でリダイレクトしたファイル中の，アプリケーション呼び出しを一時抑止したい作業の行だけを残して，それ以外の行は削除するか，コメント行に変更します。
3. ciwchgapwork コマンドを実行して，作業を「作業者実行」状態に変更します。

   ciwchgapwork コマンドの指定形式を次に示します。-f オプションには，手順 2.で編集したファイルを指定してください。

```
ciwchgapwork -sid システムID -chg -s PERFORMING -f ファイルパス
```

一時抑止の解除の実行例

1. ciwchgapwork コマンドを実行して，作業を「実行開始可能」状態に戻します。

   ciwchgapwork コマンドの指定形式を次に示します。

```
ciwchgapwork -sid システムID -chg -s READY -f ファイルパス
```

   -f オプションには，「一時抑止の実行例」の手順 2.で編集したファイルを指定してください。

ciwchgapwork コマンドの詳細については，マニュアル「uCosminexus Service Coordinator Interactive Workflow コマンド」を参照してください。

(2) アプリケーション呼び出し制御情報単位の一時抑止と解除

アプリケーション呼び出し制御情報単位の一時抑止と解除は，次に示す単位で適用できます。

- ref 識別子単位
- グループ単位
- ref 識別子共通設定単位

アプリケーション呼び出し制御情報単位でアプリケーション呼び出しを一時抑止する場合，ciwmngap コマンドを使用して最大作業件数に「0」を指定します。また，一時抑止を解除する場合は，最大作業件数を元の値に戻します。

実行例を次に示します。

一時抑止の実行例

1. アプリケーション呼び出し制御情報ファイルを作成します。一時抑止する対象（次の例では ref 識別子またはグループ）の最大作業件数に「0」を指定します。
   - アプリケーション呼び出し制御情報ファイルの例（ref 識別子単位の場合）

```
U,ope,ref01,300,300,0,0,1500
```

- アプリケーション呼び出し制御情報ファイルの例（グループ単位の場合）

```
U,grp,group01,300,300,0,0,1500
```

2. ciwmngap コマンドを実行します。-apdf オプションには，手順 1.のファイルを指定します。

   ciwmngap コマンドの指定形式を次に示します。

```
ciwmngap -sid システムID -chg -apdf アプリケーション呼び出し制御情報ファイル
```

一時抑止の解除の実行例

1. アプリケーション呼び出し制御情報ファイルを作成します。一時抑止を解除する対象（次の例では ref 識別子またはグループ）の最大作業件数に元の値（次の例では10000）を指定します。

   - アプリケーション呼び出し制御情報ファイルの例（ref 識別子単位の場合）

```
U,ope,ref01,300,300,0,10000,1500
```

   - アプリケーション呼び出し制御情報ファイルの例（グループ単位の場合）

```
U,grp,group01,300,300,0,10000,1500
```

2. ciwmngap コマンドを実行します。-apdf オプションには，手順 1.のファイルを指定します。

   ciwmngap コマンドの指定形式を次に示します。

```
ciwmngap -sid システムID -chg -apdf アプリケーション呼び出し制御情報ファイル
```

付録 H.7　使用上の注意事項

(1)　トランザクションの排他待ちについて

アプリケーションが CSCIW の API を発行した延長で，アプリケーション呼び出しサービスが実行される場合，API を発行したアプリケーションがトランザクションを決着するまで，アプリケーション呼び出しサービスは該当する案件の排他待ちとなります。そのため，アプリケーションが API を利用する際は，トランザクションの開始から終了までの期間はできるだけ短くしてください。

(2)　使用できない作業者 ID について

アプリケーション呼び出しサービスを使用する場合，システム ID 内のすべての作業の作業者 ID に「IW」で始まる文字列を指定することはできません。作業者 ID に「IW」で始まる文字列を指定した場合，意図しないアプリケーション呼び出しが発生するおそれがあります。

(3)　作業属性の一部変更禁止について

作業アプリケーション定義（種別がREST）を設定した作業の次に示す属性を変更しないでください。

- 作業者 ID（Participant）
- 優先度（Priority）
- 処理期限（Deadline）

これらの属性を変更した場合，アプリケーション呼び出しが行われないか，または意図しないタイミングでアプリケーション呼び出しが行われるおそれがあります。

付録 H.8　ビジネスプロセスの開発

作業アプリケーション（REST アプリケーション）を呼び出すビジネスプロセスを開発する際に作成するものについて説明します。

- 作業アプリケーション定義（作成は必須）
- アプリケーション呼び出し情報ファイル（作成は必須）
- REST アプリケーション呼び出し用ヘッダファイル（作成は任意）
- REST アプリケーション呼び出しスキーマ変換用スタイルシート（作成は任意）

(1)　作業アプリケーション定義の作成

CSCIW-Definer でビジネスプロセス定義を作成する際，作業アプリケーション定義を作成し，作業定義に設定します。

作業アプリケーション定義の作成

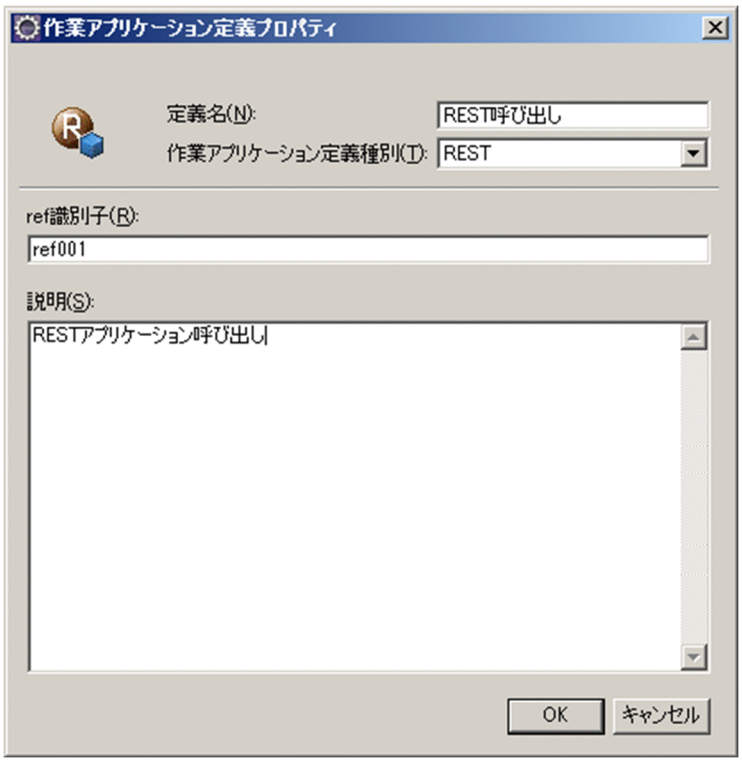
作業アプリケーション定義を作成します。作業アプリケーション定義に指定する内容を次の表に示します。

表 H-1　作業アプリケーション定義の指定内容

定義項目	指定値
定義名	作業アプリケーション定義名
作業アプリケーション定義種別	「REST」を選択
ref 識別子	ref 識別子 使用できる値は，半角英数字とアンダースコア（_）で，25 文字以内です。大文字と小文字だけが異なる同じ名称の ref 識別子は使用しないでください。
説明	任意

作業アプリケーション定義の指定例を次に示します。

図 H-8　作業アプリケーション定義の指定例



作業定義の作成

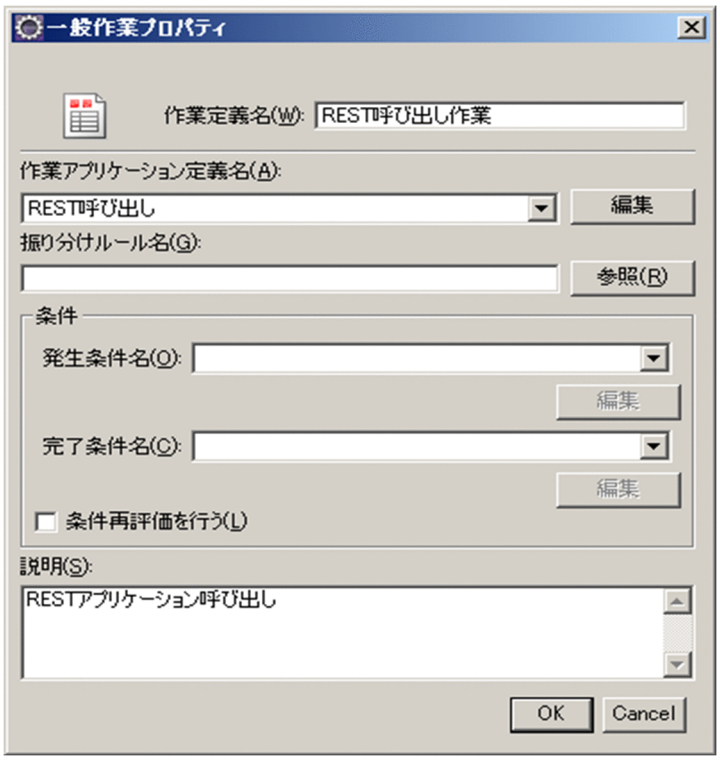
作業定義を作成します。作業定義に指定する内容を次の表に示します。

表 H-2　作業定義の指定内容

定義項目	指定値
作業定義名	任意
作業アプリケーション定義名	「作業アプリケーション定義の作成」で作成した定義名
振り分けルール名	指定しないでください。
発生条件名	任意
完了条件名	任意
条件再評価を行う	任意
説明	任意

作業定義の指定例を次に示します。

図 H-9　作業定義の指定例



(2)　アプリケーション呼び出し情報ファイルの作成

アプリケーション呼び出し情報ファイルは ref 識別子ごとに作成します。

アプリケーション呼び出し情報ファイルには，REST アプリケーションの URL などを設定します。

アプリケーション呼び出し情報ファイルの詳細については，「付録 H.12　アプリケーション呼び出し情報ファイル」を参照してください。

(3)　REST アプリケーション呼び出し用ヘッダファイルの作成

REST アプリケーション呼び出し用ヘッダファイルは，アプリケーション呼び出しサービスからのリクエストに HTTP ヘッダを付けて REST アプリケーションに送信する場合に作成します。

REST アプリケーション呼び出し用ヘッダファイルの詳細については，「付録 H.12(5)　HTTP ヘッダを記述したファイルの指定方法」を参照してください。

(4)　REST アプリケーション呼び出しスキーマ変換用スタイルシートの作成

REST アプリケーション呼び出しスキーマ変換用スタイルシートは，アプリケーション呼び出しサービスからのリクエストのボディデータ形式（XSD）を変換して REST アプリケーションに送信する場合に作成します。

REST アプリケーション呼び出しスキーマ変換用スタイルシートの詳細については，「付録 H.9(5)　XML 形式のボディデータのスキーマ変換」を参照してください。

付録 H.9　REST アプリケーションの開発

(1)　リクエストデータ

REST アプリケーションの開発時に従うリクエストデータの規定について説明します。

(a)　リクエストライン

リクエストラインには，次に示す内容を指定できます。

Method：”GET” | “POST” | “PUT” | “DELETE”

Request-URI：任意の URI

リクエストラインの指定の詳細については，次の項目を参照してください。

- Method：「付録 H.12(9)　アプリケーション呼び出し情報ファイルに指定する内容」の「(b) rest.request.method」
- Request-URI：「付録 H.12(9)　アプリケーション呼び出し情報ファイルに指定する内容」の「(c) rest.request.url」

(b)　リクエストヘッダ

リクエストヘッダには，任意の内容を指定できます。

リクエストヘッダの指定の詳細については，「付録 H.12(9)　アプリケーション呼び出し情報ファイルに指定する内容」の「(d)　rest.request.header.filepath」を参照してください。

(c)　リクエストボディ

リクエストボディは，「付録 H.9(3)　ボディデータスキーマ（XML）」または「付録 H.9(4)　ボディデータスキーマ（JSON）」に示す，XML または JSON のデータスキーマの内容に従って，作成してください。

(2)　レスポンスデータ

REST アプリケーションから返却されるレスポンスデータの規定について説明します。

(a)　ステータスライン

REST アプリケーションから返却されるステータスコードの判定結果を，次の表に示します。

表 H-3　REST アプリケーションから返却されるステータスコードの判定結果

項番	ステータスコード	判定結果
1	Informational 1xx	成功
2	Successful 2xx	成功

項番	ステータスコード	判定結果
3	Redirection 3xx	失敗
4	Client Error 4xx	失敗
5	Server Error 5xx	失敗

ステータスコードの判定結果が失敗の場合のアプリケーション呼び出しサービスの動作については，「付録 H.4　障害時の動作」を参照してください。

(b)　レスポンスヘッダ

レスポンスヘッダには，任意の内容を指定できます。

なお，セッションは継続されないため，Set-Cookie ヘッダの Cookie 情報は保持されません。

(c)　レスポンスボディ

アプリケーション呼び出しサービスは，レスポンスボディを受け取らないため，レスポンスボディを指定する必要はありません。

レスポンスボディを省略した場合，ステータスコードは 204（No Content）を返却してください。

(3)　ボディデータスキーマ（XML）

HTTP ボディ部分のボディデータスキーマの内容

アプリケーション呼び出しサービスの送受信で使用される HTTP ボディ部分のボディデータスキーマの内容を，次に示します。

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema elementFormDefault="qualified" version="1.0"
 xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="csciwRestBody" type="csciwRestBodyType"/>

  <xs:complexType name="csciwRestBodyType">
    <xs:sequence>
      <xs:element name="data" type="dataType" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="dataType">
    <xs:sequence>
      <xs:element name="key" type="xs:string"/>
      <xs:element name="value" type="xs:string" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

ボディデータスキーマ定義の要素の階層構造

項番	要素名および属性名※			型	型種別	数
1	csciwRestBody			csciwRestBodyType	混合型	1
2		data		dataType	混合型	0 以上
3			key	xs:string	－	1
4			value	xs:string	－	0～1

（凡例）

－：該当なし

注※

「要素名および属性名」列のインデントは，要素の階層構造を表しています。

ボディデータスキーマ宣言

項番	宣言	属性	値	説明
1	XML 宣言	version	1.0	XML のバージョンを指定します。 1.0 が固定値です。
2		encoding	UTF-8	エンコードを指定します。
3		standalone	yes	外部の markup 宣言の有無について指定します。 スタンドアロン文書のため，yes を指定してください。
4	schema 宣言	elementFormDefault	qualified	ローカル要素にも名前空間接頭辞を付与するために，指定します。
5		version	1.0	XML スキーマのバージョンを指定します。 1.0 が固定値です。
6		xmlns:xs	http://www.w3.org/2001/XMLSchema	名前空間"xs"を宣言するために，指定します。

ボディデータの出力例

上記のスキーマが適用されたボディデータの出力例を次に示します。

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<csciwRestBody>
    <data>
        <key>key1</key>
        <value>value1</value>
    </data>
    <data>
        <key>key2</key>
        <value>2</value>
    </data>
    <data>
        <key>ListData</key>
        <value>AAA</value>
    </data>
```

```
    <data>
        <key>ListData</key>
        <value>BBB</value>
    </data>
</csciwRestBody>
```

(4) ボディデータスキーマ（JSON）

HTTP ボディ部分のボディデータスキーマの内容

アプリケーション呼び出しサービスの送受信で使用される HTTP ボディ部分のボディデータスキーマの内容を，次に示します。

```
{
  "type": "object",
  "properties": {
    "data": {
      "type": "array",
      "items": {
        "properties": {
          "key": {
            "type": "string",
            "required": true
          },
          "value": {
            "type": ["string", "null"]
          }
        }
      }
    }
  }
}
```

ボディデータスキーマ定義の要素の階層構造

項番	要素名および属性名※		型	型種別	数	制約
1	data		array	配列	0 以上	－
2		key	string	文字列	1	null 不可
3		value	string	文字列	0～1	null 不可

（凡例）

－：該当なし

注※

「要素名および属性名」列のインデントは，要素の階層構造を表しています。

ボディデータの出力例

上記のスキーマが適用されたボディデータの出力例を次に示します。

```
{
  "data":[
    { "key":"K1", "value":"V1"},
    { "key":"K2", "value":"V2"}
```

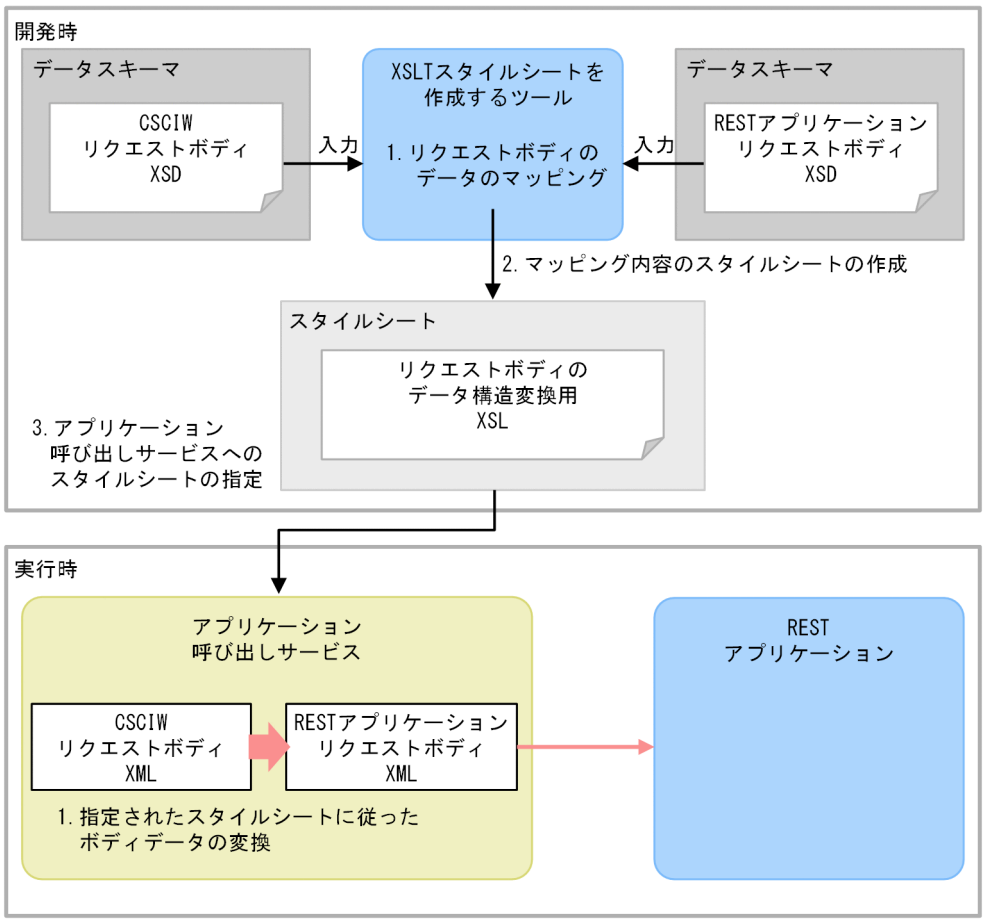
```
  ]
}
```

(5)　XML 形式のボディデータのスキーマ変換

アプリケーション呼び出しサービスのボディデータスキーマと異なるデータスキーマを持つ REST アプリケーションを呼び出したい場合は，各データスキーマにデータ構造変換のためのスタイルシートを指定する必要があります。

ボディデータのスキーマ変換の手順を次の図に示します。

図 H-10　ボディデータのスキーマ変換



＜説明＞

開発時

1. リクエストボディのデータのマッピング

   XSLT スタイルシートを作成するツールを使用して，入力スキーマおよび出力スキーマに次に示すボディデータスキーマを指定して，リクエストボディデータをマッピングします。

   ・入力スキーマ：CSCIW のリクエストのボディデータスキーマ

   ・出力スキーマ：REST アプリケーションのリクエストのボディデータスキーマ

2. マッピング内容のスタイルシートの作成

XSLT スタイルシートを作成するツールを使用して，マッピングした内容のスタイルシートを作成します。

3. アプリケーション呼び出しサービスへのスタイルシートの指定

指定方法の詳細については，「付録 H.12(9)　アプリケーション呼び出し情報ファイルに指定する内容」の「(e)　rest.request.stylesheet.filepath」を参照してください。

実行時

1. 指定されたスタイルシートに従ったボディデータの変換

アプリケーション呼び出しサービスが REST アプリケーションにリクエストを送信する際，指定されたスタイルシートの内容に従って，CSCIW のリクエストボディのデータが REST アプリケーションのリクエストボディのスキーマに変換されます。

(6)　REST アプリケーションの開発時の注意事項

アプリケーション呼び出しサービスは，REST アプリケーションの呼び出しに失敗した場合，リトライによって REST アプリケーションを複数回呼び出します。また，アプリケーション呼び出しサービスを強制停止するなどして直後の作業が完了しなかった場合，REST アプリケーションの呼び出しに成功しても，再度 REST アプリケーションを呼び出します。この場合は，リトライ回数の設定に関係なく再度呼び出します。そのため，REST アプリケーションが複数回呼び出されても正常に動作するように，べき等性を保証した実装をしてください。

(7)　REST アプリケーションの実装例

REST アプリケーションの実装例を示します。

(a)　PartsInfoService.java

PartsInfoService.java は，REST アプリケーションのルートリソースクラスのソースファイルです。「<コンテキストルート> ＋ "/parts/order"」の URL に対する POST リクエストを受信するリソースメソッドorder を持っています。

PartsInfoService.java の実装例

```
package xxx.usrapp;

import java.util.List;

import javax.ws.rs.POST;
import javax.ws.rs.Path;
import javax.ws.rs.core.Response;
import javax.ws.rs.core.Response.Status;

/**
 * RESTアプリケーションを実装したクラスです。
 */
@Path(”/parts”)
public class PartsInfoService {
```

```
    /**
     * 呼び出されるメソッドです。
     *
     * @param reqData リクエストデータ
     * @return レスポンス(204応答)
     */
    @POST
    @Path("order")
    public Response order(CSCIWRestBody reqData) {

        // リクエストデータの取得
        if(reqData != null) {
            List<CSCIWData> list = reqData.getDataList();
            if(list != null) {
                for(CSCIWData d : list) {
                    System.out.println(String.format(
                        "AppInfo: key=%1$s, value=%2$s",
                        d.getKey(),
                        d.getValue()));
                }
            }
        }

        // 204応答を返す
        return Response.status(
            Status.NO_CONTENT).build();
    }
}
```

(b) CSCIWRestBody.java

CSCIWRestBody.java は，REST アプリケーションが送受信するエンティティのcsciwRestBody 要素のソースファイルです。CSCIWRestBody.java は，data 要素を子要素として持っています。data 要素は，任意の数を指定できます。

CSCIWRestBody.java の実装例

```
package xxx.usrapp;

import java.util.ArrayList;
import java.util.List;

import javax.xml.bind.annotation.XmlElement;
import javax.xml.bind.annotation.XmlRootElement;

/**
 * RESTアプリケーション送受信データのクラスです。
 */
@XmlRootElement(name="csciwRestBody")
public class CSCIWRestBody {

  private List<CSCIWData> list = new ArrayList<CSCIWData>();

  @XmlElement(name="data")
  public List<CSCIWData> getDataList() { return list; }
  public void setDataList(List<CSCIWData> l) { list = l; }
```

```
  /**
   * 指定されたデータをリストに追加します。
   * @param data 追加するデータ
   */
  public void add(CSCIWData data) {
    list.add(data);
  }
}
```

(c) CSCIWData.java

CSCIWData.java は，REST アプリケーションが送受信するエンティティのcsciwRestBody 要素の子要素である，data 要素のソースファイルです。data 要素は，子要素としてkey 要素およびvalue 要素を 1 つずつ持っています。key 要素およびvalue 要素のデータ型は，string 型です。

CSCIWData.java の実装例

```
package xxx.usrapp;

import javax.xml.bind.annotation.XmlElement;
import javax.xml.bind.annotation.XmlRootElement;

/**
 * RESTアプリケーション送受信データのクラスです。
 */
@XmlRootElement(name="data")
public class CSCIWData {

  private String key = "";   //キー
  private String value = ""; //値

  /** コンストラクタ */
  public CSCIWData(){}
  public CSCIWData(String k, String v){ key = k; value = v; }

  @XmlElement(name="key")
  public String getKey() { return key; }
  public void setKey(String k) { key = k; }

  @XmlElement(name="value")
  public String getValue() { return value; }
  public void setValue(String v) { value = v; }
}
```

(d) アプリケーション呼び出し情報ファイルの記述例

アプリケーション呼び出し情報ファイルの記述例を示します。この例での ref 識別子の値は，「ope1」とします。

```
type=REST
rest.request.url=http://localhost/CSPWCOMockApp/parts/order
rest.request.method=POST
```

```
rest.request.header.filepath=/home/csciw/ope1header.properties
rest.request.body.key.offset=0
```

アプリケーション呼び出し情報ファイルのファイルパスは，「<BpmnCallInformationFileDirプロパティの指定値>/ope/ope1.properties」です。

なお，アプリケーション呼び出し情報ファイルのrest.request.header.filepath プロパティで設定しているHTTP ヘッダファイルの記述例は，次のとおりです。

```
Content-type: application/xml;charset=UTF-8
```

(e) アプリケーション呼び出し制御情報の設定例

この例では，ref 識別子の値が「ope1」の作業に対して，アプリケーション呼び出し制御情報を設定します。

> **ヒント**
>
> アプリケーション呼び出し制御情報は，環境構築時のデフォルトの動作でも問題ない場合，設定は不要です。環境構築時のデフォルトの動作とは，ref 識別子共通設定のデフォルト値での動作のことです。
>
> アプリケーション呼び出し制御情報および ref 識別子共通設定については，マニュアル「uCosminexus Service Coordinator Interactive Workflow コマンド」の「ciwmngap（アプリケーション呼び出し制御情報の管理)」を参照してください。

この例でのアプリケーション呼び出し制御情報に設定する値，およびciwmngap コマンドを使用したアプリケーション呼び出し制御情報の設定例を次に示します。

アプリケーション呼び出し制御情報に設定する値

- ref 識別子：ope1
- 実行間隔：180（単位：秒）
- リトライ間隔：900（単位：秒）
- リトライ回数：0（単位：回）
- 最大作業件数：10000（単位：件）
- 障害復旧間隔：1500（単位：秒）

アプリケーション呼び出し制御情報の設定例

1. アプリケーション呼び出し制御情報ファイルの作成

   アプリケーション呼び出し制御情報を更新するための入力用ファイルである，アプリケーション呼び出し制御情報ファイルを作成します。

   次に示すアプリケーション呼び出し制御情報ファイルの記入例では，「アプリケーション呼び出し制御情報に設定する値」で示した各値を記入しています。

アプリケーション呼び出し制御情報ファイルの記入例

```
#UPDATEOPTION,REFTYPE,REF,EXECUTEINTERVAL,RETRYINTERVAL,RETRYCOUNT,WORKITEMMAX,RECOVERYTI
ME
U,ope,ope1,180,900,0,10000,1500
```

2. ciwmngap コマンドの実行

作成したアプリケーション呼び出し制御情報ファイルの内容を適用するために，アプリケーション呼び出し制御情報ファイルのファイルパスを指定したciwmngap コマンドを実行します。

ciwmngap コマンドの指定形式

```
ciwmngap -sid 〈システムID〉 -chg -apdf 〈アプリケーション呼び出し制御情報ファイルのファイルパス〉
```

ヒント

アプリケーション呼び出し制御情報ファイルおよびciwmngap コマンドについては，マニュアル「uCosminexus Service Coordinator Interactive Workflow コマンド」の「ciwmngap（アプリケーション呼び出し制御情報の管理)」を参照してください。

付録 H.10　アプリケーション呼び出しサービスの動作を制御する設定項目

アプリケーション呼び出しサービスの動作を制御する設定項目について説明します。

(1)　アプリケーション呼び出し制御情報の設定

アプリケーション呼び出し制御情報は，ciwmngap コマンドを使用して設定します。

アプリケーション呼び出し制御情報の設定の詳細については，マニュアル「uCosminexus Service Coordinator Interactive Workflow コマンド」の「ciwmngap（アプリケーション呼び出し制御情報の管理)」を参照してください。

(2)　アプリケーション呼び出しグループ定義の設定

アプリケーション呼び出しグループ定義は，ciwmngapgrp コマンドを使用して設定できます。ciwmngapgrp コマンドは，アプリケーション呼び出しサービスの性能チューニング時に使用します。

アプリケーション呼び出しグループ定義の設定の詳細については，マニュアル「uCosminexus Service Coordinator Interactive Workflow コマンド」の「ciwmngapgrp（アプリケーション呼び出しグループの管理)」を参照してください。

アプリケーション呼び出しサービスの性能チューニングの詳細については，マニュアル「uCosminexus Service Coordinator Interactive Workflow BPMN 連携機能　使用の手引」の「アプリケーション呼び出しサービスの性能チューニング」を参照してください。

(3)　ポーリング間隔の設定

アプリケーション呼び出しサービスのポーリング間隔は，共通設定ファイルに設定します。

詳細については，「付録 H.11(5)　共通設定ファイルに指定するパラメタ一覧および内容」の「(c) AppCallServicePollingInterval」を参照してください。

(4)　WorkManager の最大スレッド数の設定

WorkManager の最大スレッド数は，J2EE サーバ用ユーザプロパティファイルに設定します。

詳細については，「6.2.8(3)　WorkManager の最大スレッド数を変更する」を参照してください。

(5)　REST 通信に関する設定

アプリケーション呼び出しサービスの REST 通信に関する設定は，Cosminexus アプリケーションサーバで設定します。

Cosminexus アプリケーションサーバでは，次に示す項目などを設定できます。

- 通信タイムアウト（ソケット接続，ソケット読み込み）
- ログ（Cosminexus V11 の V9 互換モードを使用する場合だけ設定できます）

詳細については，マニュアル「Cosminexus 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）」の「J2EE サーバ単位の通信タイムアウトの設定」および「NIO HTTP サーバ」を参照してください。

> **メモ**
>
> プロキシサーバ経由の接続，SSL プロトコルによる接続などの設定はサポート対象外です。

付録 H.11　共通設定ファイル

共通設定ファイルの詳細について説明します。

(1)　共通設定ファイルの概要

共通設定ファイルとは，ポーリング間隔などを設定するファイルを指します。

(2)　共通設定ファイルの設定個所

共通設定ファイルの設定個所について説明します。

共通設定ファイルは，次のパスに格納してください。

- CSCIW_CONF_DIR 環境変数が設定されていない場合

  Windows の場合

  ```
  %CSCIW_HOME%¥bpmn¥conf¥csciwbpmnconf.properties
  ```

  UNIX の場合

  ```
  ${CSCIW_HOME}/bpmn/conf/csciwbpmnconf.properties
  ```

- CSCIW_CONF_DIR 環境変数が設定されている場合

  共通設定ファイルの格納先を，対応するシステム ID のセットアッププロパティファイル（csciwsetup.properties）と同じ格納先にしてください。

  Windows の場合

  ```
  %CSCIW_CONF_DIR%¥csciwbpmnconf.properties
  ```

  UNIX の場合

  ```
  ${CSCIW_CONF_DIR}/csciwbpmnconf.properties
  ```

(3)　共通設定ファイルの読み込みタイミング

共通設定ファイルの読み込みタイミングについて説明します。

読み込みタイミングは，パラメタのカテゴリごとに異なります。各パラメタがどのカテゴリに属するかについては，「付録 H.11(5)　共通設定ファイルに指定するパラメタ一覧および内容」を参照してください。

カテゴリごとの読み込みタイミングを次に示します。

- パラメタのカテゴリが「共通」のとき

  J2EE サーバを起動後，初めてアプリケーション呼び出しサービスを開始した時に読み込まれます（J2EE サーバを起動後，1 回だけ読み込まれます）。

- パラメタのカテゴリが「アプリケーション呼び出しサービス」のとき

  アプリケーション呼び出しサービスを開始した時に読み込まれます。

一度読み込まれた共通設定ファイルのパラメタを変更する方法については，「6.2.8(1)　ファイルに格納した設定情報を変更する」の，共通設定ファイルの内容を変更した場合についての記述を参照してください。

(4)　共通設定ファイルの記述規則

共通設定ファイルの記述時の規則について説明します。

共通設定ファイルは，Java のプロパティファイルの仕様と同じです。値に「¥」を記述する場合，「¥」の前に「¥」をエスケープ文字として記述してください。また，ファイルのエンコーディングは UTF-8 とし，BOM を付けないで保存してください。

CSCIW 02-20 以前で使用していた共通設定ファイル（Unicode 形式のファイルを含む）を，そのまま使用することもできます。

(5)　共通設定ファイルに指定するパラメタ一覧および内容

共通設定ファイルに指定する内容について説明します。

指定するパラメタの一覧（共通設定ファイル）

共通設定ファイルに指定するパラメタの一覧を示します。

表 H-4　指定するパラメタの一覧（共通設定ファイル）

項番	パラメタ名	カテゴリ	指定内容
1	BpmnLogFileDir	共通	ログ出力先ディレクトリ
2	BpmnCallInformationFileDir	共通	アプリケーション呼び出し情報ファイルの格納先ディレクトリ
3	AppCallServicePollingInterval	アプリケーション呼び出しサービス	アプリケーション呼び出しサービスのポーリング間隔
4	AppCallServiceMsgFileNum	アプリケーション呼び出しサービス	アプリケーション呼び出しサービスメッセージファイルの面数
5	AppCallServiceTraceFileNum	アプリケーション呼び出しサービス	アプリケーション呼び出しサービストレースファイルの面数
6	AppCallServiceMsgFileSize	アプリケーション呼び出しサービス	アプリケーション呼び出しサービスメッセージファイルの出力サイズ
7	AppCallServiceTraceFileSize	アプリケーション呼び出しサービス	アプリケーション呼び出しサービストレースファイルの出力サイズ
8	AppCallServiceMsgOutputThreashold	アプリケーション呼び出しサービス	アプリケーション呼び出しサービスメッセージ出力レベル
9	AppCallServiceTraceOutputThreashold	アプリケーション呼び出しサービス	アプリケーション呼び出しサービストレース出力レベル
10	AppCallServiceMsgTimeToDelete	アプリケーション呼び出しサービス	アプリケーション呼び出しサービスメッセージファイルを削除するまでの日数
11	AppCallServiceTraceTimeToDelete	アプリケーション呼び出しサービス	アプリケーション呼び出しサービストレースファイルを削除するまでの日数
12	AppCallServiceTransactionTimeout	アプリケーション呼び出しサービス	アプリケーション呼び出しサービスのトランザクションタイムアウト値
13	AppCallServiceDebugRestBody	アプリケーション呼び出しサービス	アプリケーション呼び出しサービスと REST アプリケーション間で送受信するボディデータのトレースファイル出力の有無
14	AppCallServiceInitialInterval	アプリケーション呼び出しサービス	アプリケーション呼び出しサービスの初回ポーリング間隔

(a) BpmnLogFileDir

共通設定ファイルに指定するBpmnLogFileDirパラメタについて，説明します。

ユーザ指定値

値の範囲	パラメタの指定の省略	指定を省略したときの仮定値
－	可	＜Windowsの場合＞ %CSCIW_HOME%¥log ＜UNIXの場合＞ ${CSCIW_HOME}/log

（凡例）

－：指定できる値の範囲に，制限はありません。

説明

次のログファイルの出力先ディレクトリを指定します。

- アプリケーション呼び出しサービストレースファイル
- アプリケーション呼び出しサービスメッセージファイル

(b) BpmnCallInformationFileDir

共通設定ファイルに指定するBpmnCallInformationFileDirパラメタについて，説明します。

ユーザ指定値

値の範囲	パラメタの指定の省略	指定を省略したときの仮定値
－	可	＜Windowsの場合＞ %CSCIW_HOME%¥bpmn¥callinfo ＜UNIXの場合＞ ${CSCIW_HOME}/bpmn/callinfo

（凡例）

－：指定できる値の範囲に，制限はありません。

説明

アプリケーション呼び出しサービスが呼び出し処理を実行する際に読み込む「アプリケーション呼び出し情報ファイル」の格納先ディレクトリを指定します。

アプリケーション呼び出しサービスは，このパラメタで設定されたディレクトリ配下のopeフォルダに格納された各アプリケーション呼び出し情報ファイルを読み込みます。そのため，このパラメタで指定したディレクトリ直下に，opeフォルダが存在している必要があります。

(c) AppCallServicePollingInterval

共通設定ファイルに指定するAppCallServicePollingIntervalパラメタについて，説明します。

ユーザ指定値

値の範囲	パラメタの指定の省略	指定を省略したときの仮定値
1～86,400 の整数	可	60

説明

アプリケーション呼び出しサービスのポーリング間隔を秒単位で指定します。

このパラメタで指定した秒数ごとに，アプリケーション呼び出しサービスのプロセスを実行します。

このパラメタに短い秒数を指定した場合，データベースの検索が頻繁に発生するため，60 以上の値を指定することを推奨します。

(d) AppCallServiceMsgFileNum

共通設定ファイルに指定するAppCallServiceMsgFileNum パラメタについて，説明します。

ユーザ指定値

値の範囲	パラメタの指定の省略	指定を省略したときの仮定値
1～16 の整数	可	2

説明

アプリケーション呼び出しサービスメッセージファイルの面数を指定します。

ラップアラウンドモードの場合，出力面数を指定してください。シフトモードの場合，バックアップ面数を指定してください。

(e) AppCallServiceTraceFileNum

共通設定ファイルに指定するAppCallServiceTraceFileNum パラメタについて，説明します。

ユーザ指定値

値の範囲	パラメタの指定の省略	指定を省略したときの仮定値
1～16 の整数	可	4

説明

アプリケーション呼び出しサービストレースファイルの面数を指定します。

ラップアラウンドモードの場合，出力面数を指定してください。シフトモードの場合，バックアップ面数を指定してください。

(f) AppCallServiceMsgFileSize

共通設定ファイルに指定するAppCallServiceMsgFileSize パラメタについて，説明します。

ユーザ指定値

値の範囲	パラメタの指定の省略	指定を省略したときの仮定値
100,000～2,147,483,647 の整数	可	2097152

説明

アプリケーション呼び出しサービスメッセージファイルの出力サイズをバイト単位で指定します。

(g) AppCallServiceTraceFileSize

共通設定ファイルに指定するAppCallServiceTraceFileSize パラメタについて，説明します。

ユーザ指定値

値の範囲	パラメタの指定の省略	指定を省略したときの仮定値
100,000～2,147,483,647 の整数	可	16777216

説明

アプリケーション呼び出しサービストレースファイルの出力サイズをバイト単位で指定します。

(h) AppCallServiceMsgOutputThreashold

共通設定ファイルに指定するAppCallServiceMsgOutputThreashold パラメタについて，説明します。

ユーザ指定値

値の範囲	パラメタの指定の省略	指定を省略したときの仮定値
-1～1,000 の整数	可	20

説明

アプリケーション呼び出しサービスメッセージ出力レベルを指定します。

0 以上の値を指定すると，指定した値以下のレベルのアプリケーション呼び出しサービスメッセージが出力されます。

-1 を指定した場合，アプリケーション呼び出しサービスメッセージは出力されません。

(i) AppCallServiceTraceOutputThreashold

共通設定ファイルに指定するAppCallServiceTraceOutputThreashold パラメタについて，説明します。

ユーザ指定値

値の範囲	パラメタの指定の省略	指定を省略したときの仮定値
-1～1,000 の整数	可	20

説明

アプリケーション呼び出しサービストレース出力レベルを指定します。

0 以上の値を指定すると，指定した値以下のレベルのアプリケーション呼び出しサービストレースが出力されます。

-1 を指定した場合，アプリケーション呼び出しサービストレースは出力されません。

(j) AppCallServiceMsgTimeToDelete

共通設定ファイルに指定するAppCallServiceMsgTimeToDelete パラメタについて，説明します。

ユーザ指定値

値の範囲	パラメタの指定の省略	指定を省略したときの仮定値
-1～24,855 の整数	可	-1

説明

アプリケーション呼び出しサービスメッセージファイルを削除するまでの日数を指定します。

このパラメタは，ローテーション種別がラップアラウンドモードの場合だけ有効です。

0 を指定した場合，出力されたアプリケーション呼び出しサービスメッセージファイルはすぐに削除されます。

-1 を指定した場合，日数が経過してもファイルは削除されません。

(k) AppCallServiceTraceTimeToDelete

共通設定ファイルに指定するAppCallServiceTraceTimeToDelete パラメタについて，説明します。

ユーザ指定値

値の範囲	パラメタの指定の省略	指定を省略したときの仮定値
-1～24,855 の整数	可	-1

説明

アプリケーション呼び出しサービストレースファイルを削除するまでの日数を指定します。

このパラメタは，ローテーション種別がラップアラウンドモードの場合だけ有効です。

0 を指定した場合，出力されたアプリケーション呼び出しサービストレースファイルはすぐに削除されます。

-1 を指定した場合，日数が経過してもファイルは削除されません。

(l) AppCallServiceTransactionTimeout

共通設定ファイルに指定するAppCallServiceTransactionTimeout パラメタについて，説明します。

ユーザ指定値

値の範囲	パラメタの指定の省略	指定を省略したときの仮定値
0～2,147,483,647 の整数	可	0

説明

アプリケーション呼び出しサービスのトランザクションタイムアウト値を秒単位で指定します。

0 を指定した場合，J2EE サーバの設定値（J2EE サーバ用ユーザプロパティファイルusrconf.propertiesのejbserver.jta.TransactionManager.defaultTimeOut の設定値）が指定されたと見なされます。

1 以上の値を指定した場合，トランザクションタイムアウト値（秒）が指定されたと見なされます。

(m)　AppCallServiceDebugRestBody

共通設定ファイルに指定するAppCallServiceDebugRestBody パラメタについて，説明します。

ユーザ指定値

値の範囲	パラメタの指定の省略	指定を省略したときの仮定値
true またはfalse	可	false

説明

アプリケーション呼び出しサービスと REST アプリケーションの間で送受信するボディデータ（リクエストボディとレスポンスボディ）の内容を，トレースファイルに出力するかどうかを指定します。

true を指定した場合，ボディデータの内容を XML 形式でトレースファイルに出力します。XSLT スタイルシートを使用してボディデータをスキーマ変換した場合は，変換前と変換後の両方の内容を出力します。

なお，true を指定すると，アプリケーション呼び出しのスループットが低下するおそれがあります。

false を指定した場合，ボディデータの内容をトレースファイルに出力しません。

(n)　AppCallServiceInitialInterval

共通設定ファイルに指定するAppCallServiceInitialInterval パラメタについて，説明します。

ユーザ指定値

値の範囲	パラメタの指定の省略	指定を省略したときの仮定値
0～86,400 の整数	可	0

説明

アプリケーション呼び出しサービスの開始後，最初のポーリングを行うまでの時間を秒単位で指定します。指定された秒数の経過後，アプリケーション呼び出しを定期的に実行します。

このパラメタは，アプリケーション呼び出しサービスの開始後，一定時間経過後にアプリケーション呼び出しを実行したい場合だけ指定してください。

付録 H.12　アプリケーション呼び出し情報ファイル

アプリケーション呼び出し情報ファイルの詳細について説明します。

(1)　アプリケーション呼び出し情報ファイルの概要

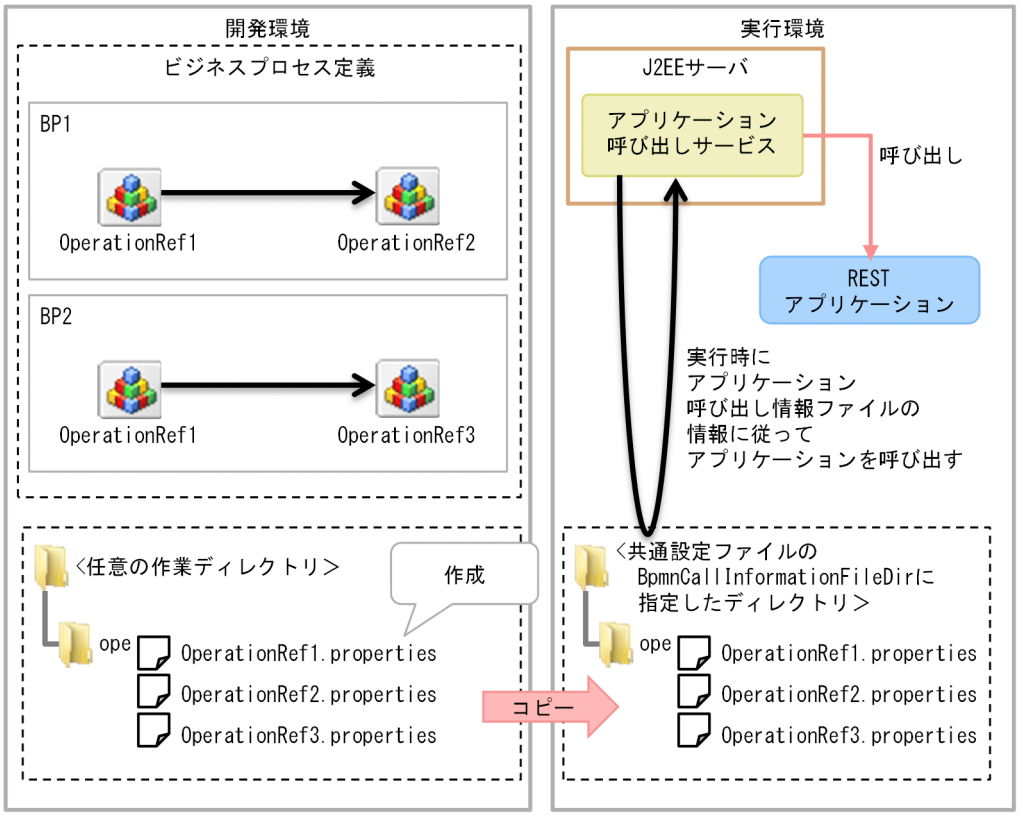
アプリケーション呼び出し情報ファイルには，REST アプリケーションの URL などを設定します。

(2)　アプリケーション呼び出し情報ファイルの設定個所

アプリケーション呼び出し情報ファイルは，ビジネスプロセス定義を作成する開発環境で作成し，実行環境にコピーしてください。

アプリケーション呼び出し情報ファイルの配置例を次の図に示します。

図 H-11　アプリケーション呼び出し情報ファイルの配置例



なお，アプリケーション呼び出し情報ファイルは，ref 識別子単位に作成します。各アプリケーション呼び出し情報ファイルには，呼び出し対象の ref 識別子に対応するファイル名を付けてください。

作成したアプリケーション呼び出し情報ファイルの格納先として，共通設定ファイルの BpmnCallInformationFileDir プロパティで指定したディレクトリ内に，サブディレクトリ（ope）を作成します。

次に示すアプリケーション呼び出し情報ファイルのファイル名を付けて，該当するサブディレクトリにファイルを格納してください。

```
<BpmnCallInformationFileDirの指定値>/ope/<ref識別子の値>.properties
```

重要

- 「図 H-11　アプリケーション呼び出し情報ファイルの配置例」のOperationRef1 のように，複数のビジネスプロセス定義（BP1，BP2）で同じ ref 識別子を定義している場合，共通で使用するOperationRef1.properties ファイルをope フォルダの下に配置してください。
- 大文字と小文字だけが異なる同じ名称の ref 識別子は使用しないでください。

(3)　アプリケーション呼び出し情報ファイルの記述規則

アプリケーション呼び出し情報ファイルの記述規則を次に示します。

- java.util.Properties クラスが扱えるフォーマットで記載してください。
- 文字コードは，UTF-8 を使用してください。ただし，BOM 付き UTF-8 では記述できません。
- プロパティの各キーは，大文字と小文字が区別されます。
- プロパティの各キーは，1 つだけ指定してください（同一のキーは複数指定できません）。同一のキーを複数指定した場合，どの設定値が有効になるかは不定です。
- プロパティの記載順は任意です。また，プロパティが読み込まれる順番は不定です。
- プロパティのキーに，アプリケーション呼び出し情報ファイルで使用できるプロパティで定義されていないキー名を指定した場合，アプリケーションの呼び出し時にエラーが発生します。
- プロパティでファイルパスを指定する場合，パスの区切り文字には"/"を使用してください。
- プロパティの値の先頭に"$$"を記載した場合，"$"に変換されます。
- プロパティの値の先頭に"@@""を記載した場合，"@"に変換されます。
- プロパティの値が空の場合，エラーが発生します。

アプリケーション呼び出し情報ファイルが次に示すケースに該当する場合，アプリケーション呼び出しサービスは，該当する ref 識別子のアプリケーション呼び出しを中止します。

- 指定必須のプロパティのキーが指定されていない
- プロパティに不正な値が指定されている

(4)　アプリケーション呼び出し情報ファイルで使用できる組み込み変数

組み込み変数は，CSCIW で管理する案件や作業の属性値を指定するための変数です。アプリケーション呼び出し情報ファイルに記載した組み込み変数は，呼び出し元の案件や作業の属性値に変換されます。

使用できる組み込み変数

項番	組み込み変数	説明
1	@PIID	案件 ID
2	@PIName	案件名

項番	組み込み変数	説明
3	@PICreator	案件投入者
4	@PIDeadline	案件処理期限
5	@PDefName	ビジネスプロセス定義名
6	@WIID	作業 ID
7	@WDefName	作業定義名
8	@NULL	null 値

組み込み変数の使用規則

アプリケーション呼び出し情報ファイルで組み込み変数を使用する場合は，次の規則に従ってください。

- 組み込み変数名の前後に，ほかの組み込み変数名やその他の文字列を連続して記載することはできません。また，半角空白などで文字列を区切ることもできません。

  誤った記載例（通常の文字列として扱われる）

  ```
  abc@PIName
  ```

  誤った記載例（エラーが発生）

  ```
  @PIName@PIID
  ```

  ただし，rest.request.url プロパティの値の中では，特定の文字で区切ることで，組み込み変数の前後にほかの組み込み変数や文字列を連続して記載できます。

- プロパティの値の先頭に"@"を記載し，上記の表「使用できる組み込み変数」の「組み込み変数」列以外の文字列を指定した場合は，アプリケーションの呼び出し時にエラーが発生します。

  誤った記載例

  ```
  @ABC
  ```

  ```
  @PINameABC
  ```

- プロパティの値の先頭に"@@"を記載した場合は，組み込み変数ではなく通常の文字列として扱われます。
- 組み込み変数の対象となる属性の値が null の場合，null 値として扱われます。ただし，rest.request.url プロパティの値の中に含まれている場合は，空文字として扱われます。
- 案件処理期限は，"1970/01/01 00:00:00 GMT"を起点とした通算秒に置き換えられます。

> **ヒント**
>
> rest.request.url プロパティについては，「付録 H.12(9)　アプリケーション呼び出し情報ファイルに指定する内容」の「(c)　rest.request.url」を参照してください。

(5)　HTTP ヘッダを記述したファイルの指定方法

HTTP ヘッダを記述したファイルとは，REST アプリケーションを呼び出す際のリクエストの HTTP ヘッダに指定する内容を記述するファイルのことです。HTTP ヘッダを記述したファイルを作成した場合は，

HTTP ヘッダを記述したファイルのファイルパスを，アプリケーション呼び出し情報ファイルの rest.request.header.filepath プロパティに指定してください。

> **ヒント**
>
> rest.request.header.filepath プロパティについては，「付録 H.12(9)　アプリケーション呼び出し情報ファイルに指定する内容」の「(d)　rest.request.header.filepath」を参照してください。

HTTP ヘッダを記述したファイルの指定規則を次に示します。

- java.util.Properties クラスが扱えるフォーマットで記載してください。
- 文字コードは UTF-8 を使用してください。ただし，BOM 付き UTF-8 では記述しないでください。
- ファイル中に記載する要素の数や種類は，HTTP ヘッダの仕様に合わせてください。複数回指定できない要素を 2 行以上指定しているなど，HTTP ヘッダの仕様と異なる記述をした場合の動作は保証しません。

HTTP ヘッダの記述例を次に示します。

```
Accept=application/xml
```

(6)　アプリケーション呼び出し情報ファイルの読み込みタイミング

アプリケーション呼び出し情報ファイルは，対応する ref 識別子のアプリケーション呼び出しが最初に実行されるタイミングで，読み込まれます。一度読み込まれたファイルの内容は，アプリケーション呼び出しサービス内で保持されます。

また，アプリケーション呼び出し情報の一部として読み込まれた次のファイルについても同様に，一度読み込まれたファイルの内容は，アプリケーション呼び出しサービス内で保持されます。

- REST アプリケーション用の HTTP ヘッダファイル
- REST アプリケーション用のスキーマ変換に使用されるスタイルシート（リクエストボディ用）

ファイルの再読み込みについては，「6.2.8(1)　ファイルに格納した設定情報を変更する」の，アプリケーション呼び出し情報ファイル，REST アプリケーション呼び出し用ヘッダファイル，または REST アプリケーション呼び出しスキーマ変換用スタイルシートの内容を変更した場合についての説明を参照してください。

(7)　アプリケーション呼び出し情報ファイルとリクエストボディの関係

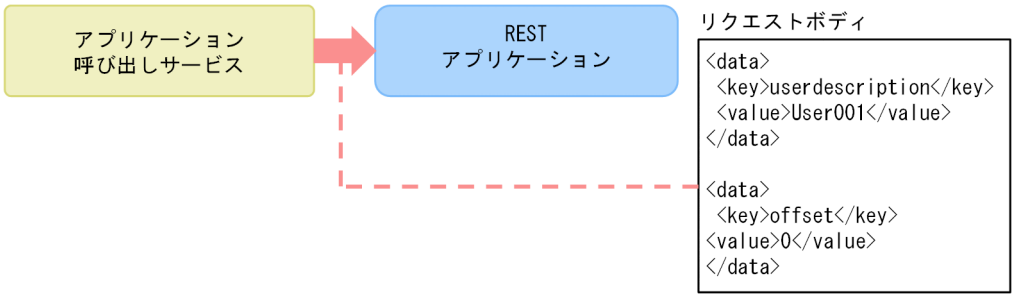
アプリケーション呼び出し情報ファイルに記述した内容に従って，対応する値がリクエストボディに渡されます。リクエストボディは，REST アプリケーションに渡されます。

アプリケーション呼び出し情報ファイルの記述例を次に示します。この例では，1 行目で rest.request.body.key.userdescription の値として案件名を指定しています。

```
rest.request.body.key.userdescription=@PIName
rest.request.body.key.offset=0
```

アプリケーション呼び出し情報ファイルを上記の例のとおり記述した場合の，REST アプリケーションにリクエストボディが渡されるまでの流れを，次の図に示します。

図 H-12　REST アプリケーションにリクエストボディが渡されるまでの流れ



＜説明＞

アプリケーション呼び出し情報ファイルの内容に従って，対応する値がリクエストボディに渡されます。リクエストボディは，REST アプリケーションに渡されます。

(8)　アプリケーション呼び出し情報ファイルに指定するパラメタの一覧

指定するパラメタの一覧（アプリケーション呼び出し情報ファイル）

アプリケーション呼び出し情報ファイルに指定するパラメタの一覧を示します。

表 H-5　指定するパラメタの一覧（アプリケーション呼び出し情報ファイル）

項番	パラメタ名	指定内容
1	type（REST）	アプリケーション呼び出しの種類（REST）
2	rest.request.method	REST アプリケーションのメソッド
3	rest.request.url	REST アプリケーションの URL
4	rest.request.header.filepath	HTTP ヘッダを記述したファイルのファイルパス
5	rest.request.stylesheet.filepath	リクエストボディ用スキーマ変換スタイルシートのファイルパス
6	rest.request.read.timeout	REST アプリケーションの呼び出し時の読み込みタイムアウト値
7	rest.request.connect.timeout	REST アプリケーション呼び出し時の接続タイムアウト値
8	rest.request.body.key.<key 要素値>※	リクエストボディに格納する値
9	rest.request.idempotency	REST アプリケーションがべき等性を保証しているかどうか

注※

key 要素値とは，REST アプリケーションの呼び出し時に使用されるボディデータ（リクエストボディまたはレスポンスボディ）の，data 要素中のkey 要素の値です。

次に示すボディデータの記述例の"aaa"に該当する値です。

```
<data>
    <key>aaa</key>
    <value>bbb</value>
  </data>
```

(9) アプリケーション呼び出し情報ファイルに指定する内容

アプリケーション呼び出し情報ファイルの格納先ファイルパス

REST アプリケーションの呼び出しについて設定する，アプリケーション呼び出し情報ファイルの格納先ファイルパスを，次に示します。

```
<BpmnCallInformationFileDirプロパティの指定値>/ope/<ref識別子の値>.properties
```

アプリケーション呼び出し情報ファイルの記述例

REST アプリケーションの呼び出しについて設定する場合の，アプリケーション呼び出し情報ファイルの記述例を次に示します。

```
type=REST
rest.request.method=POST
rest.request.url=http://hostname/app/function
rest.request.header.filepath= /home/csciw/header.properties
rest.request.body.key.offset=0
rest.request.body.key.userdescription=@PIName
```

(a) type（REST）

キーの説明

アプリケーション呼び出しの種類を指定します。このキーは指定必須です。

指定値の説明

"REST"を指定してください。

REST タイプのアプリケーションにアクセスします。

注意事項

大文字と小文字は区別します。

(b) rest.request.method

キーの説明

REST アプリケーションのメソッドを指定します。このキーは指定必須です。

指定値の説明

呼び出し対象の REST アプリケーションのメソッド名を指定します。次のどれかを指定してください。

- GET：リソースを取得するリクエストの場合
- POST：リソースを追加するリクエストの場合
- PUT：リソースを更新するリクエストの場合
- DELETE：リソースを削除するリクエストの場合

注意事項

大文字と小文字は区別します。

(c) rest.request.url

キーの説明

REST アプリケーションの URL を指定します。このキーは指定必須です。

指定値の説明

呼び出し対象の REST アプリケーションの URL を指定します。

記述例

```
http://host1:80/apppath1/apppath2
```

URL の次の部分には，組み込み変数を記述することもできます。

- パスのディレクトリ名およびファイル名
- クエリパラメタのキーおよび値

記述例

```
http://host1:80/apppath1/apppath2/@PIName?processid=@PIID&workid=@WIID
```

注意事項

- このプロパティに指定した組み込み変数の値は，次のとおり URI エンコードされます。なお，URI エンコードの文字コードは UTF-8 とします。
  ・半角スペースは"%20"にエンコードされる
  ・半角スペース以外の文字はjava.net.URLEncoder クラスの仕様に従いエンコードされる
- 組み込み変数の値以外は，自動では URI エンコードされません。RFC 2396 で定義されている文字を使用し，必要に応じて URI エンコードした値を指定してください。

(d) rest.request.header.filepath

キーの説明

アプリケーション呼び出しサービスのリクエストに HTTP ヘッダを含める場合に指定します。

指定を省略した場合，ヘッダ部が空のリクエストを送信します。

指定値の説明

HTTP ヘッダを記載したヘッダファイルのパスを，絶対パスで指定します。

ヘッダファイルについては「付録 H.12(5)　HTTP ヘッダを記述したファイルの指定方法」を参照してください。

注意事項

OS が UNIX，かつ，root ユーザ以外の任意の OS ユーザが CSCIW の業務アプリケーションを実行する場合，このプロパティに指定したファイルの所有者，およびグループもあわせて変更してください。

(e)　rest.request.stylesheet.filepath

キーの説明

アプリケーション呼び出しサービスのリクエストボディの，XML スキーマ変換をする場合に指定します。

指定を省略した場合，スキーマ変換をしません。

指定値の説明

XSLT で記述されたスタイルシートのファイルパスを，絶対パスで指定します。

注意事項

OS が UNIX，かつ，root ユーザ以外の任意の OS ユーザが CSCIW の業務アプリケーションを実行する場合，このプロパティに指定したファイルの所有者，およびグループもあわせて変更してください。

(f)　rest.request.read.timeout

キーの説明

REST アプリケーション呼び出し時の読み込みタイムアウト値を変更する場合に指定します。

指定を省略した場合，タイムアウト値は Cosminexus の J2EE サーバ単位の通信タイムアウトを設定する`ejbserver.javaee.jaxrs.config.client.readTimeout` キーの設定に従います。

指定値の説明

REST アプリケーション呼び出し時のクライアントソケットの読み込みタイムアウト値を，0～2,147,483,647 の範囲で指定します。単位はミリ秒です。

タイムアウトしない設定をする場合，`0` を指定してください。

注意事項

REST アプリケーション呼び出しのタイムアウト値を Cosminexus の設定に従わせる場合は，このプロパティを設計および指定する必要はありません。

(g)　rest.request.connect.timeout

キーの説明

REST アプリケーション呼び出し時の接続タイムアウト値を変更する場合に指定します。

指定を省略した場合，タイムアウト値は Cosminexus の J2EE サーバ単位の通信タイムアウトを設定する`ejbserver.javaee.jaxrs.config.client.connectTimeout` キーの設定に従います。

指定値の説明

REST アプリケーション呼び出し時のクライアントソケットの接続タイムアウト値を，0～2,147,483,647 の範囲で指定します。単位はミリ秒です。

タイムアウトしない設定をする場合，0 を指定してください。

注意事項

REST アプリケーション呼び出しのタイムアウト値を Cosminexus の設定に従わせる場合は，このプロパティを設計および指定する必要はありません。

(h) rest.request.body.key.<key 要素値>

キーの説明

アプリケーション呼び出しサービスのリクエストボディに格納する値を指定します。

rest.request.method が POST または PUT の場合に指定できます。

指定を省略した場合，リクエストボディが空のリクエストを送信します。

指定値の説明

リクエストボディに渡すvalue 要素の値を指定してください。指定された<key 要素値>をkey 要素の値に，また指定されたプロパティの値をvalue 要素の値に持つdata 要素を，リクエストボディに追加します。

XML スキーマについては「付録 H.9(3)　ボディデータスキーマ（XML）」を参照してください。

このプロパティの値には組み込み変数を記述できます。

注意事項

プロパティ値に値が null の組み込み変数を指定した場合，value 要素のないリクエストボディが作成されます。

(i) rest.request.idempotency

キーの説明

REST アプリケーションがべき等性を保証していない場合に false を指定します。

指定を省略した場合は ture が指定され，REST アプリケーションがべき等性を保証しているものとして，アプリケーション呼び出しサービスが REST アプリケーションを複数回呼び出すことがあります。

指定値の説明

REST アプリケーションがべき等性を保証しているかどうかを指定します。

true の場合

べき等性を保障している場合に指定します。アプリケーション呼び出しサービスが REST アプリケーションを複数回呼び出すことがあります。

false の場合

べき等性を保障していない場合に指定します。アプリケーション呼び出しサービスが REST アプリケーションを複数回呼び出すことはありません。

注意事項

通常，このプロパティを設定しないでください。このプロパティに false を指定する場合の注意事項を次に示します。

- REST アプリケーションの呼び出しで障害が発生した場合，呼び出し回数が 0 回になることがあります。
- REST アプリケーションの呼び出しで障害が発生した場合，ciwchgapwork コマンドを使用して，作業の状態を変更する必要があります。REST アプリケーションを再実行する場合，作業を実行開始可能状態に戻してください。REST アプリケーションを再実行しない場合，作業を完了させ案件を遷移させてください。再実行するかどうかは REST アプリケーションの実行結果を基に決定してください。
- アプリケーション呼び出し制御情報のリトライ回数とリトライ間隔は有効になりません。

付録 H.13　リトライの対象から外れた作業に関する運用

障害が発生し，作業が「作業者実行」状態になった場合の対処手順について説明します。

(1)　アプリケーション呼び出しを再実行する

「作業者実行」状態の作業を，再びアプリケーション呼び出しの対象に戻すことができます。

アプリケーション呼び出しを再実行する手順を次に示します。

1. **ciwchgapwork コマンドを実行して，「作業者実行」状態の作業の情報を取得する。**

   ciwchgapwork コマンドの指定形式を次に示します。

   ```
   ciwchgapwork -sid システムID -list -ope -ref ref識別子 > ファイルパス
   ```

   -ref オプションには，対象の ref 識別子を指定します。-ref オプションは省略できます。

   出力結果はファイルにリダイレクトします。

   ciwchgapwork コマンドの詳細については，マニュアル「uCosminexus Service Coordinator Interactive Workflow コマンド」を参照してください。

2. **障害の原因を取り除く。**

3. **ciwchgapwork コマンドを実行して，作業を「実行開始可能」状態に戻す。**

   ciwchgapwork コマンドの指定形式を次に示します。

   ```
   ciwchgapwork -sid システムID -chg -s READY -f ファイルパス
   ```

   -f オプションには，手順 1.でリダイレクトしたファイルを指定してください。

(2)　アプリケーション呼び出しをしないで案件を遷移させる

「作業者実行」状態の作業について，アプリケーションの呼び出しをしないで案件を遷移させることができます。

アプリケーション呼び出しをしないで案件を遷移させる手順を次に示します。

1. ciwchgapwork コマンドを実行して，「作業者実行」状態の作業の情報を取得する。

   ciwchgapwork コマンドの指定形式を次に示します。

   ```
   ciwchgapwork -sid システムID -list -ope -ref ref識別子 > ファイルパス
   ```

   -ref オプションには，対象の ref 識別子を指定します。-ref オプションは省略できます。

   出力結果はファイルにリダイレクトします。

   ciwchgapwork コマンドの詳細については，マニュアル「uCosminexus Service Coordinator Interactive Workflow コマンド」を参照してください。

2. 障害の原因を取り除く。

3. ciwchgapwork コマンドを用いて作業を完了し，案件を遷移させる。

   ciwchgapwork コマンドの指定形式を次に示します。

   ```
   ciwchgapwork -sid システムID -chg -s EXECUTED -f ファイルパス
   ```

   -f オプションには，手順 1.でリダイレクトしたファイルのパスを指定してください。

付録 H.14　チューニング

アプリケーション呼び出しサービスのスループットに影響するチューニング項目については，マニュアル「uCosminexus Service Coordinator Interactive Workflow BPMN 連携機能　使用の手引」の「アプリケーション呼び出しサービスの性能チューニング」を参照してください。

付録 I　REST API による運用管理の操作

運用管理 REST サービスの REST API による運用管理の操作について説明します。

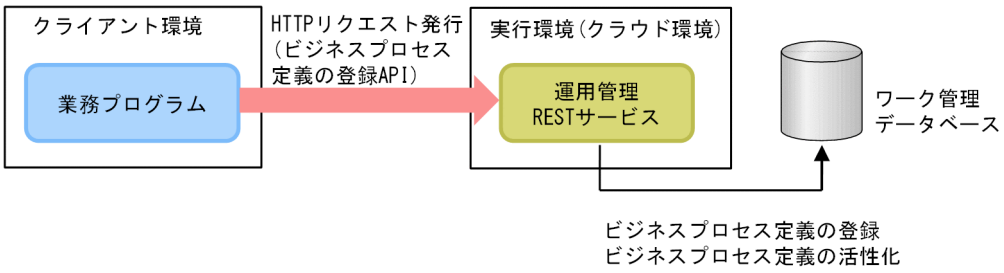
運用管理 REST サービスは，運用管理 REST API を使用してリソースを操作する機能を提供する J2EE アプリケーションです。運用管理 REST サービスは，CSCIW のコマンドで実行できる機能の一部を，RESTful API として提供します。これによって，クラウドサービスやコンテナ環境で動作するシステムとの連携や Web アプリケーションなどの業務プログラムからの運用管理の操作ができます。

運用管理 REST サービスはサーブレットアプリケーションで実装されます。業務プログラムは，リソースの取得やクエリの実行などをしたいリソースがある場合，運用管理 REST サービスが提供する運用管理 REST API を使用してリソースにアクセスできます。運用管理 REST サービスは，HTTP リクエストを受け取り，リソースを操作してレスポンスを返します。

運用管理 REST サービスは，実行環境外からワーク管理システムの運用管理機能を使用したい場合など，必要に応じて使用してください。

運用管理 REST サービスを使用して，実行環境外から実行環境のビジネスプロセス定義の登録およびビジネスプロセス定義の活性化をする場合の，処理の流れを次の図に示します。

図 I-1　実行環境外から運用管理機能を使用する場合の処理の流れ



付録 I.1　運用管理 REST API を使用してできる操作

運用管理 REST API は，HTTP リクエストを受け取り，運用管理 REST サービスへ HTTP リクエストを発行するための RESTful API です。

運用管理 REST API を使用すると，実行環境でのコマンド実行に相当する操作ができます。また，処理に時間が掛かる機能についてはジョブとして提供され，運用管理 REST API でジョブを投入することで機能が実行されます。運用管理 REST API でできる操作と，対応するコマンドを次の表に示します。

表 I-1　運用管理 REST API でできる操作と対応するコマンド

運用管理 REST API でできる操作	ジョブとして提供される機能	対応するコマンド
ビジネスプロセス定義の操作（登録と活性化，および非活性化と削除）	-	ciwmngbp

運用管理 REST API でできる操作	ジョブとして提供される機能	対応するコマンド
振り分けルール定義の操作（登録と活性化，および非活性化と削除）	−	ciwmngcr
案件の一括削除（指定された条件に該当する案件を一括で削除）	○	ciwdelpi

（凡例）

○：機能がジョブとして提供される

−：該当しない

付録 I.2　ジョブを使用した非同期処理の仕組み

ジョブとは，運用管理 REST サービスが実行する機能のうち，運用管理 REST API の呼び出しとは非同期に実行する機能のことです。案件の一括削除のように，処理に時間が掛かる機能をジョブとして提供します。ジョブに対しては次の操作ができます。

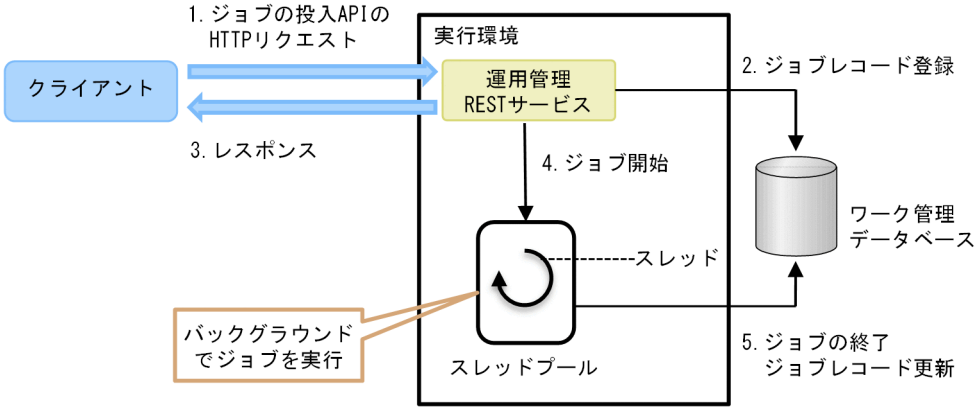
- ジョブの投入
- ジョブレコードの取得（個別と一覧）
- ジョブの自動削除
- ジョブの再投入

(1)　ジョブの投入による非同期処理の実行

ジョブを実行するには，ジョブの投入 API を使用します。

ジョブの投入 API が呼び出された場合の処理の流れを次の図に示します。

図 I-2　ジョブの投入 API が呼び出された場合の処理の流れ



[説明]

1. 実行する処理を，API のリクエストボディに指定します。

2. ジョブの投入 API が呼び出されると，運用管理 REST サービスはジョブに対応するレコード（ジョブレコード）をワーク管理データベースに登録し，ジョブをバックグラウンドで（API の呼び出しとは非同期に）実行します。
3. ジョブを受け付けたことを表すレスポンスが，ジョブの完了を待たずに，API を呼び出したクライアントに返却されます。
4. ジョブが開始されます。

   ジョブは，WorkManager を使用してバックグラウンドで実行されます。
5. ジョブが完了すると，運用管理 REST サービスはジョブの実行結果をジョブレコードに反映します。

ジョブレコードはジョブレコードの取得 API，またはジョブレコードの一覧取得 API を使用して確認できます。

(2) ジョブレコードの取得による非同期処理の実行状況の確認

非同期処理の実行状況は，ジョブレコードの取得 API，またはジョブレコードの一覧取得 API でジョブレコードを取得して確認できます。ジョブレコードは，次の表に示すプロパティで構成されます。

表 I-2 ジョブレコードのプロパティ一覧

項番	プロパティ名	内容
1	ArgumentDataList	ジョブが使用する引数データ
2	ClosedDate	ジョブの終了日時
3	Creator	ジョブの投入者
4	FunctionName	ジョブのファンクション名
5	ID	ジョブのジョブ ID
6	ResultDataList	ジョブの結果データ
7	ReturnCode	ジョブのリターンコード
8	StartDate	ジョブの開始日時
9	StateCode	ジョブの状態

各プロパティの詳細を説明します。

1. ArgumentDataList

   ジョブ実行時の引数データです。
2. ClosedDate

   ジョブが正常終了または異常終了した日時です。
3. Creator

   ジョブの投入者です。
4. FunctionName

ジョブの機能を表すファンクション名です。

ファンクション名とジョブの機能の対応を次に示します。

表 I-3　ファンクション名とジョブの機能の対応

ファンクション名	ジョブの機能
delpi	案件の一括削除

5. ID

ジョブレコードを一意に識別するための識別子です。ジョブ ID は，ジョブ実行前のジョブレコード登録時に，運用管理 REST サービスで採番します。

ジョブレコードの取得 API で取得するレコードを特定する場合に使用します。

6. ResultDataList

ジョブの結果データです。ジョブが正常終了した時に，結果データを格納します。

7. ReturnCode

ジョブのリターンコードです。

8. StartDate

ジョブを開始した日時です。

9. StateCode

ジョブの状態を表すコード値です。コード値ごとの意味を次の表で示します。

表 I-4　コード値ごとの意味

コード値	ジョブの状態	意味
d	実行中	ジョブが実行されている状態を示します。
o	正常終了	ジョブが正常に終了した状態を示します。
v	異常終了	ジョブの実行中に例外が発生し，異常終了した状態を示します。

タイミングごとにジョブレコードのプロパティに設定される値を次の表に示します。

表 I-5　タイミングごとにジョブレコードのプロパティに設定される値

項番	プロパティ名	内容	タイミングごとに設定される値		
			ジョブ投入時	ジョブ正常終了時	ジョブ異常終了時
1	ArgumentDataList	ジョブが使用する引数データ	ジョブの投入 API の ArgumentDataList	なし（ジョブ投入時の値のまま）	なし（ジョブ投入時の値のまま）
2	ClosedDate	終了日時	なし	ジョブが終了した日時	ジョブが終了した日時
3	Creator	ジョブの投入者	ジョブの投入 API の UserDescription	なし（ジョブ投入時の値のまま）	なし（ジョブ投入時の値のまま）

項番	プロパティ名	内容	タイミングごとに設定される値		
			ジョブ投入時	ジョブ正常終了時	ジョブ異常終了時
4	FunctionName	ジョブのファンクション名	ジョブの投入 API の FunctionName	なし（ジョブ投入時の値のまま）	なし（ジョブ投入時の値のまま）
5	ID	ジョブのジョブ ID	ジョブ投入時に採番された一意な ID	なし（ジョブ投入時の値のまま）	なし（ジョブ投入時の値のまま）
6	ResultDataList	結果データ	なし	ジョブの結果データ	なし
7	ReturnCode	リターンコード	なし	ジョブのリターンコード	ジョブのリターンコード
8	StartDate	開始日時	ジョブを開始した日時	なし（ジョブ投入時の値のまま）	なし（ジョブ投入時の値のまま）
9	StateCode	ジョブの状態	実行中を示すコード値（d）	正常終了を示すコード値（o）	異常終了を示すコード値（v）

重要

ジョブ実行時に障害が発生すると，「ジョブ異常終了時」列に示す値に更新できずにジョブが終了することがあります。この場合，状態は実行中（d）のままとなります。

障害発生時にジョブが終了しているかどうかは，メッセージ KDIW05603-E で確認できます。メッセージの詳細については，マニュアル「uCosminexus Service Coordinator Interactive Workflow メッセージ」を参照してください。

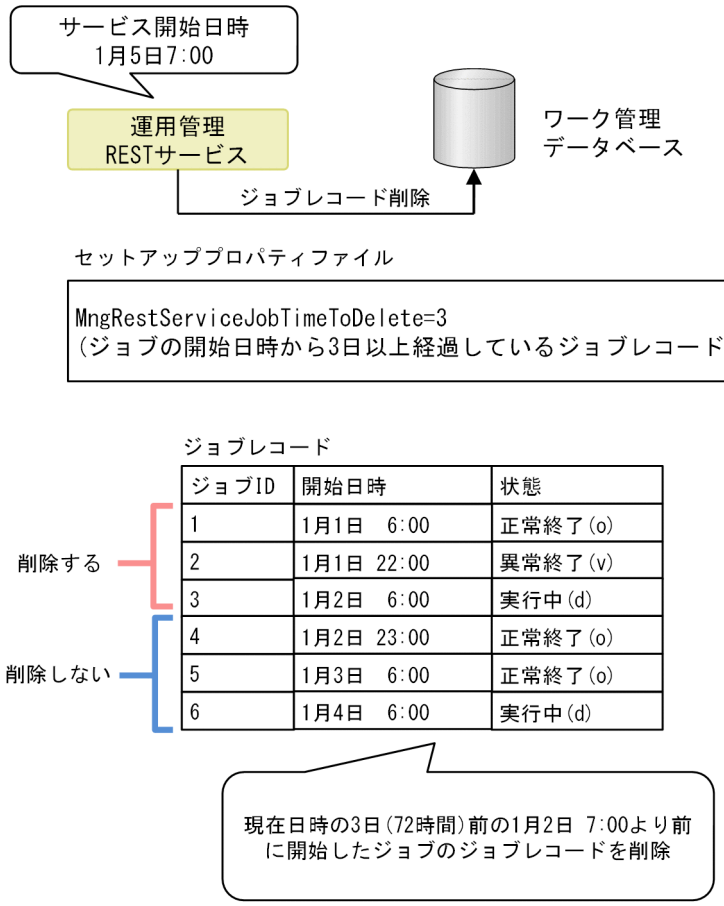
(3) ジョブレコードの自動削除

運用管理 REST サービスは，ジョブの開始日時から一定時間が経過したすべてのジョブレコードを，サービスの開始時に，ジョブの状態に関係なく自動的に削除します。ジョブの開始日時からの経過時間は，セットアッププロパティファイルのMngRestServiceJobTimeToDelete で指定します。ジョブの開始日時から，次に示す計算式の時間が経過しているすべてのジョブレコードが自動削除の対象になります。

<MngRestServiceJobTimeToDeleteの値>×24時間

MngRestServiceJobTimeToDelete の指定と自動削除されるジョブレコードの例を次の図に示します。この例では，MngRestServiceJobTimeToDelete に3 を指定して 1 月 5 日の 7 時に運用管理 REST サービスを開始しているため，1 月 2 日の 7 時より前に開始したジョブのジョブレコードがすべて削除されます。

図 I-3　MngRestServiceJobTimeToDelete の指定と自動削除されるジョブレコードの例



ジョブID	開始日時	状態
1	1月1日　6:00	正常終了(o)
2	1月1日 22:00	異常終了(v)
3	1月2日　6:00	実行中(d)
4	1月2日 23:00	正常終了(o)
5	1月3日　6:00	正常終了(o)
6	1月4日　6:00	実行中(d)

(4)　ジョブの再投入による非同期処理の再実行

ジョブの実行中に障害が発生してジョブが最後まで実行されなかった場合，ジョブを再投入できます。

運用管理 REST サービスは，ジョブを実行する前のジョブレコードの登録時に，メッセージ KDIW70043-I を出力します。ジョブを再投入するには，このメッセージに出力されたファンクション名と引数データを指定して，ジョブの投入 API を実行します。

メッセージの詳細については，マニュアル「uCosminexus Service Coordinator Interactive Workflow メッセージ」を参照してください。

付録 I.3　運用管理 REST サービス使用時の事前準備

ここでは，運用管理 REST サービス使用時の事前準備について説明します。

(1)　データベースコネクション数の見積もり

運用管理 REST サービスが必要とするコネクション数は，次のとおりです。

```
<運用管理REST APIを同時に実行する数>+<ジョブを同時に実行する数>
```

付録 I.4　運用管理 REST サービスの設定

運用管理 REST サービスを使用するには，運用管理 REST サービスの設定をします。運用管理 REST サービスは，運用管理 REST サービスを実行する Cosminexus にだけ組み込んでください。すべての Cosminexus に対して組み込む必要はありません。なお，先に CIWManagementServer を Cosminexus に組み込んでから，運用管理 REST サービスを組み込んでください。

(1)　コンテナ拡張ライブラリへの設定追加

運用管理 REST サービスを使用する場合は，Cosminexus の J2EE サーバ用オプションの add.class.path に，csciwbpmn.jar ファイルを追加してください。すでに csciwbpmn.jar が追加されている場合は，新たに設定を追加する必要はありません。

- Windows の場合

```
add.class.path=<CSCIWインストールフォルダ>¥lib¥csciwbpmn.jar
```

- UNIX の場合

```
add.class.path=/opt/hitachi/CSCIW/lib/csciwbpmn.jar
```

add.class.path については，マニュアル「Cosminexus 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)」の「usrconf.cfg（J2EE サーバ用オプション定義ファイル)」の説明を参照してください。

(2)　運用管理 REST サービスのインポート

Cosminexus を起動したあと，運用管理 REST サービスを組み込みます。

運用管理 REST サービスのファイルの格納パスは，次のとおりです。

- Windows の場合

```
<CSCIWインストールフォルダ>¥lib¥csciwmngws.ear
```

- UNIX の場合

```
/opt/hitachi/CSCIW/lib/csciwmngws.ear
```

cjimportapp コマンドを使用して，インポートします。

インポート方法については，マニュアル「Cosminexus 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」の「J2EE アプリケーションのインポート」の説明を参照してください。

(3)　運用管理 REST サービスのプロパティ設定

データベースと接続するための DB Connector のリソースアダプタの表示名に「DB_Connector_for_CSCIW」を設定した場合，この手順は不要です。

「DB_Connector_for_CSCIW」以外を設定した場合は，データソース表示名をリソースアダプタと同じ表示名に変更してください。変更方法については「付録 I.5(1)　運用管理 REST サービスが使用するデータソース表示名の変更」を参照してください。

(4)　運用管理 REST サービスの開始

cjstartapp コマンドを使用して，運用管理 REST サービスを開始します。

運用管理 REST サービスを開始するためには，CIWManagementServer が開始している必要があります。

開始方法については，マニュアル「Cosminexus 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」の「J2EE アプリケーションの開始」の説明を参照してください。

運用管理 REST サービスの J2EE アプリケーション名は CSCIWManagementRESTService です。

(5)　コンテキストルートの設定

J2EE サーバのフロントエンドにリバースプロキシを配置する場合，コンテキストルート「csciwmngws」のリクエストを転送する設定をしてください。

付録 I.5　運用管理 REST サービスの設定の変更

運用管理 REST サービスに関する設定の変更手順について説明します。

(1)　運用管理 REST サービスが使用するデータソース表示名の変更

運用管理 REST サービスが使用するデータソース表示名には，データベースと接続するための DB Connector のプロパティに設定した DB Connector のリソースアダプタの表示名と同じ名前を指定します。

1. cjimportapp コマンドを使用して，運用管理 REST サービスをインポートする

   運用管理 REST サービスを組み込みます。インポート済みの場合は，この手順は不要です。

2. cjgetappprop コマンドを使用して，XML ファイル（属性ファイル）を取得する

   cjgetappprop コマンドの実行例

   ```
   cjgetappprop <サーバ名称> -name CSCIWManagementRESTService -type all -c <属性ファイルパス>
   ```

   cjgetappropコマンドの詳細については，マニュアル「Cosminexus アプリケーションサーバ　リファレンス コマンド編」を参照してください。

3. 取得した XML ファイル（属性ファイル）を編集する

   取得した XML ファイル（属性ファイル）をテキストエディタで開き，2 か所のresource-ref 要素の linked-to 要素を「DB_Connector_for_CSCIW」から「4.5.3(4)　DB Connector の設定」で設定したリソースアダプタの表示名に変更します。

- Session Bean 属性の設定例

```
<resource-ref>
  <description xml:lang="en"></description>
  <res-ref-name>jdbc/CSCIWManagementRESTServiceDataSource</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
  <res-sharing-scope>Shareable</res-sharing-scope>
  <injection-target>
    <injection-target-class>jp.co.Hitachi.soft.csciw.rest.mng.application.MngRESTService
Bean</injection-target-class>
    <injection-target-name>mDataSource</injection-target-name>
  </injection-target>
  <linked-to>DB_Connector_for_CSCIW</linked-to>
</resource-ref>
```

- WAR 属性の設定例

```
<resource-ref>
  <description xml:lang="en"></description>
  <res-ref-name>jdbc/CSCIWManagementRESTServiceDataSource</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
  <res-sharing-scope>Shareable</res-sharing-scope>
  <linked-to>DB_Connector_for_CSCIW</linked-to>
</resource-ref>
```

4. cjsetappprop コマンドを使用して，編集した XML ファイル（属性ファイル）の情報をプロパティに設定する

   cjsetappprop コマンドの実行例

   ```
   cjsetappprop <サーバ名称> -name CSCIWManagementRESTService -type all -c <属性ファイルパス
   >
   ```

   cjsetappprop コマンドの詳細については，マニュアル「Cosminexus アプリケーションサーバ　リファレンス コマンド編」を参照してください。

5. cjstartapp コマンドを使用して，運用管理 REST サービスを開始する

(2)　WorkManager の最大スレッド数の変更

ジョブの投入 API を，同時に 11 以上実行する場合だけ，WorkManager の最大スレッド数を増やしてください。

運用管理 REST サービスは投入されたジョブ 1 つにつき WorkManager のスレッドを 1 つ使用します。ジョブを同時に並列で実行する必要がなければ，運用管理 REST サービスが使用する WorkManager のスレッドは 1 つで問題ありません。

WorkMaanager の最大スレッド数の変更方法を次に示します。

1. J2EE サーバ用ユーザプロパティファイル（usrconf.properties）に，WorkManager の最大スレッド数を指定する

WorkManager の最大スレッド数の指定例を次に示します。

```
ejbserver.commonj.WorkManager.non_daemon_work_threads=20
```

重要

WorkManager の最大スレッド数は，J2EE サーバ単位で設定します。そのため，設定値の変更は，同じ J2EE サーバ上にインポートされたアプリケーション呼び出しサービスの動作にも影響します。

2. 変更対象の J2EE サーバを再起動する

WorkManager の最大スレッド数に，設定した値が適用されます。

付録 I.6　運用管理 REST サービスの停止および削除

運用管理 REST サービスの停止および削除手順を次に示します。

1. 運用管理 REST サービスの停止

`cjstopapp` コマンドを使用して，運用管理 REST サービスを停止します。

停止方法については，マニュアル「Cosminexus 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」の「J2EE アプリケーションの停止」の説明を参照してください。

2. 運用管理 REST サービスの削除

`cjdeleteapp` コマンドを使用して，運用管理 REST サービスを削除します。

削除方法については，マニュアル「Cosminexus 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」の「J2EE アプリケーションの削除」の説明を参照してください。

付録 I.7　運用管理 REST サービスの開始

運用管理 REST サービスを開始するには，`cjstartapp` コマンドを使用します。なお，運用管理 REST サービスを開始するためには，CIWManagementServer が開始している必要があります。

開始方法の詳細については，マニュアル「Cosminexus 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」の「J2EE アプリケーションの開始」の説明を参照してください。

付録 I.8　運用管理 REST API の構成要素

運用管理 REST API の構成要素について説明します。

(1) HTTP リクエスト

運用管理 REST API のリクエスト URL は，次に示すとおり，ドメイン名，コンテキストルート，API バージョンおよびリソースアクセスパスから構成されます。

```
http(s)://<ドメイン>/<コンテキストルート>/<APIバージョン>/<リソースアクセスパス>
```

ドメインは Web サーバで設定します。運用管理 REST サービスのコンテキストルートは csciwmngws です。API バージョンは，v1，v2 など運用管理 REST API のバージョンが API ごとに固定値として設定されます。残りの部分がリソースにアクセスするためのパスです。

(2) HTTP メソッド

CSCIW の運用管理 REST API で使用する HTTP メソッドを次の表で示します。

表 I-6 HTTP メソッド

HTTP メソッド	内容
GET	リソースを取得します。
POST	リソースを新しく追加します。
PUT	指定されたリソースを修正します。
DELETE	指定されたリソースを削除します。

(3) リクエストパラメタ

業務アプリケーションが運用管理 REST サービスにパラメタを渡す場合，クエリパラメタかリクエストボディを使用します。HTTP メソッドと，使用できるリクエストパラメタの関係を次の表で示します。

表 I-7 リクエストパラメタの送信方法

項番	リクエストパラメタ	GET	POST	PUT	DELETE
1	クエリパラメタ	○	×	×	○
2	リクエストボディ	△	○	○	×

(凡例)

○：利用できる

×：エラーが発生する

△：無視される

リクエストボディにパラメタを指定する場合，XML 形式または JSON 形式で記載します。その際，HTTP ヘッダのコンテンツタイプ（Content-Type）にapplication/xml かapplication/json を指定する必要があります。コンテンツタイプにapplication/xml かapplication/json 以外の値が指定された場合や，未指定の場合はエラーになります。

使用できないリクエストパラメタを送信した場合，ステータスコード 400 のエラーが返されます。

リクエストボディに指定するパラメタをすべて省略する場合は，「付録 I.8(10)　リクエストパラメタ省略時のリクエストボディと HTTP ヘッダ」を参照してください。

(4)　HTTP レスポンス

運用管理 REST API が正常終了した場合，HTTP レスポンスには，2xx のステータスコードが返されます。また，レスポンスボディにはリソースなどの情報が，XML または JSON 形式で返されます。コンテンツタイプはapplication/xml またはapplication/json です。レスポンスで返されるリソースの情報（プロセスデータは除く）については，マニュアル「uCosminexus Service Coordinator Interactive Workflow AP 開発ガイド」の「指定できる属性一覧」の説明を参照してください。

レスポンスの形式は，HTTP ヘッダの"Accept"にapplication/xml またはapplication/json を指定すると切り替わります。application/xml またはapplication/json 以外の値が指定された場合はエラーになります。"Accept"が未指定の場合は XML 形式で返されます。

運用管理 REST API がエラーになった場合，HTTP レスポンスには，3xx，4xx または 5xx のステータスコードが返されます。また，レスポンスボディには次に示すようなエラードキュメントが返されます。エラーを受け取ったクライアントは，返却された HTTP レスポンスと，ステータスコードに従ってエラーに対処できます。HTTP レスポンスのボディに設定しているメッセージにはエラーの詳細情報は付与されません。発生したエラーの詳細情報については，運用管理 REST API のログファイルを確認してください。

エラー時のレスポンスボディ（XML の場合）

```
<?xml version=”1.0” encoding=”UTF-8”?>
<error>
<code>KDIWnnnnn-E</code>
<message>メッセージ本文</message>
</error>
```

エラー時のレスポンスボディ（JSON の場合）

```
{
”code” : ”KDIWnnnnn-E”,
”message” : ”メッセージ本文”
}
```

(5)　ユーザ記述子

運用管理 REST API が内部で Java API を呼び出すとき，運用管理 REST API から Java API にユーザ記述子を渡します。運用管理 REST API から渡すユーザ記述子のデフォルト値は csciwmngws です。デフォルト値とは異なるユーザ記述子を指定する場合は，セットアッププロパティファイルの MngRestServiceUserDescription キーを変更するか，各運用管理 REST API のユーザ記述子のパラメタ（userdescription）を指定します。なお，ユーザ記述子は 32 バイト以下の長さにしてください。

(6) データ型と文字コード

業務アプリケーションと運用管理 REST サービスのリクエストとレスポンスでやり取りされるデータ型は，数値，文字列，配列，日付です。日付は，yyyy-MM-dd'T'HH:mm:ssZ の形式（ISO8601 の拡張形式）で表します（例：2016-06-09T10:32:41+09:00）。レスポンスの文字コードは UTF-8 です。

また，無限遠の日付の場合は日付の形式ではなく定数で表します。無限遠の未来の場合はBEYOND となり，無限遠の過去の場合はORIGIN となります。

(7) オブジェクトの状態

CSCIW のオブジェクトの状態と，それに対応する定数名とコード値の対応を次の表に示します。レスポンスでは，コード値を文字列型で返します。

表 I-8 オブジェクトの状態

状態	定数名（コード値）
ビジネスプロセス定義の状態	ACTIVE(b)，INACTIVE(a)，UNDEFINED(z)
振り分けルール定義の状態	ACTIVE(b)，INACTIVE(a)，UNDEFINED(z)

(8) XML と JSON の相違点

XML の場合

- リクエストボディとレスポンスにルート要素（<Parameter>や<ProcessInstances>など）を含みます。
- タグを省略した場合に null として扱われます。

JSON の場合

- リクエストボディとレスポンスにルート要素を含みません。このため，各運用管理 REST API のリクエストボディとレスポンスの構造に記載されているルート要素は無視されます。
- 要素を省略した場合に null として扱われます。
- Cosminexus アプリケーションサーバでは，値に null が指定された場合は空文字として扱われます。
- レスポンスの型が数値，または boolean 型の場合に，レスポンスはダブルクォーテーション「”」で囲われて返されます。

(9) レスポンスの最大取得件数

運用管理 REST サービスの一覧取得 API の場合，レスポンスで取得できる最大件数を指定できます。最大件数のデフォルト値は 100 件です。最大取得件数は，セットアッププロパティファイルの MngRestServiceResponseMaxCount キーに設定するか，運用管理 REST API のリクエストパラメタに指定できます。

(10)　リクエストパラメタ省略時のリクエストボディと HTTP ヘッダ

一部の運用管理 REST API では，リクエストボディに指定するパラメタをすべて省略できます。パラメタをすべて省略する場合は，リクエストボディを省略できます。リクエストボディを省略するときは，HTTP ヘッダのContent-Type を指定しないでください。リクエストボディを省略してContent-Type を指定した場合の動作はサポートしません。

リクエストボディに指定するパラメタをすべて省略する場合のContent-Type とリクエストボディの指定値を，次の表に示します。

表 I-9　Content-Type とリクエストボディの指定値

項番	Content-Type	リクエストボディ	リクエストボディの例
1	指定しない	指定しない	—
2	application-xml	ルート要素だけ指定する	<Parameter/>
3	application-json	{}だけ指定する	{}

（凡例）

—：該当しない

付録 I.9　運用管理 REST API 一覧

運用管理 REST サービスが対応するリソースの操作と，その URL の一覧を次に示します。

表 I-10　運用管理 REST API 一覧

項番	リソースの操作		URL※
-	ビジネスプロセス定義		
1		ビジネスプロセス定義の登録	POST /v1/process-definition
2		ビジネスプロセス定義の削除	DELETE /v1/process-definition
-	振り分けルール		
3		振り分けルールの登録	POST /v1/casting-rule
4		振り分けルールの削除	DELETE /v1/casting-rule
-	ジョブ		
5		ジョブの投入	POST /v1/job/create
6		ジョブレコードの取得	GET /v1/job/(JOBID)
7		ジョブレコードの一覧取得	GET /v1/job

注※　(JOBID)：ジョブレコードのジョブ ID

付録 I.10　運用管理 REST API の記述形式

運用管理 REST API の URL，リクエスト，レスポンスの仕様を次に示します。ここでは URL スキーム，ドメイン名およびコンテキストルートは省略します。例えば，ドメイン名が restserver の場合，ジョブレコードの一覧取得の API の URL は，http://restserver/csciwmngws/v1/job のようになります。

●メソッドと URL

メソッドと URL の形式を説明しています。

●クエリパラメタまたはリクエストボディ

各クエリパラメタまたはリクエストボディに指定する名前，型，省略可否，および内容について説明しています。API によって，クエリパラメタまたはリクエストボディのどちらかを使用します。

●リクエスト例

リクエスト URL およびリクエストボディの指定例を示しています。

●レスポンス

レスポンスの形式を説明しています。レスポンスが XML 形式または JSON 形式の場合，その構造も説明しています。

●ステータスコード

ステータスコード，メディアタイプおよび内容を説明しています。

●レスポンス例

レスポンス例を示しています。

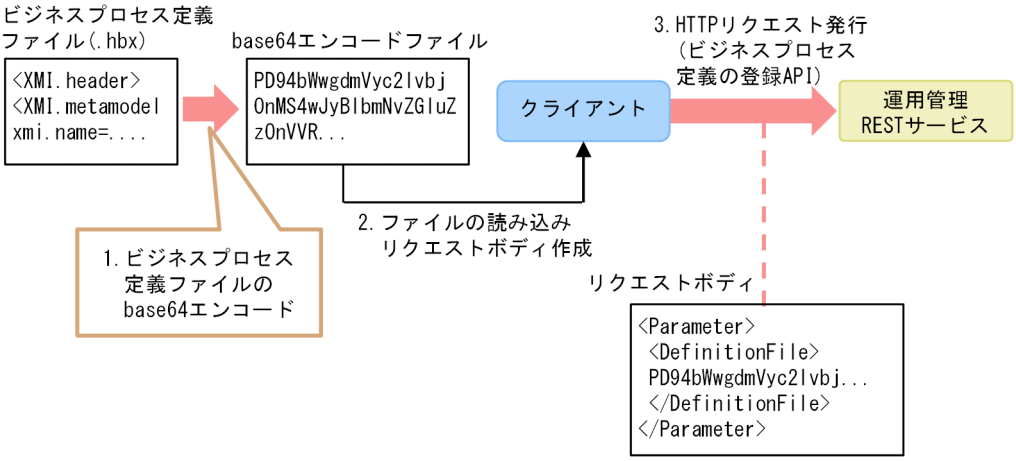
付録 I.11　リクエストパラメタにファイルを指定する方法

運用管理 REST API のリクエストパラメタにファイルを指定する方法を説明します。

運用管理 REST API を使用したビジネスプロセス定義の登録や振り分けルール定義の登録では，リクエストパラメタにファイルの内容を指定します。ファイルの内容を指定する場合，ファイルの内容は base64 エンコードした文字列を使用します。

ビジネスプロセス定義の登録の場合にリクエストパラメタにファイルを指定するときの処理の流れを次の図に示します。

図 I-4　リクエストパラメタにファイルを指定する場合の処理の流れ（ビジネスプロセス定義の登録）



[説明]

1. ビジネスプロセス定義ファイル（.hbx）を base64 エンコードします。
2. base64 エンコードファイルを読み込んでリクエストボディを作成します。
3. 作成したリクエストボディを定義したビジネスプロセス定義の登録 API で，運用管理 REST サービスに HTTP リクエストを発行します。

コマンドラインツールを使用して base64 エンコードとcurl コマンドを実行し，ビジネスプロセス定義を登録する場合の指定例を次に示します。この例では，登録するビジネスプロセス定義ファイル名は「SampleBp#0001.hbx」とします。

コマンドライン	説明
$ base64 -w 0 SampleBp#0001.hbx > EncodedText.txt	hbx ファイルの base64 エンコード
$ APIURL=http://restserver/csciwmngws/v1/process-definition	URL の定義
$ HEADER="Content-Type: application/json"	ヘッダの定義
$ BODYDATA={¥"DefinitionFile¥":¥"$(cat EncodedText.txt)¥"}	リクエストボディの定義
$ curl -X POST ${APIURL} -H "${HEADER}" -d ${BODYDATA}	curl コマンドの実行

ビジネスプロセス定義の登録を行う場合に base64 エンコードした文字列を指定したリクエストボディの例を次に示します。

XML の場合

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Parameter>
  <DefinitionFile>
PD94bWwgdmVyc2lvbj0nMS4wJyBlbmNvZGluZz0nVVRGLTgnPz48IURPQ1RZUEUgWE1JPjxYTUkgeG1pLnZlcnNpb
249JzEuMCc+PC9YTUk+
  </DefinitionFile>
</Parameter>
```

JSON の場合

```
{
  "DefinitionFile":"PD94bWwgdmVyc2lvbj0nMS4wJyBlbmNvZGluZz0nVVRGLTgnPz48IURPQ1RZUEUgWE1JPjxYTUkgeG1pLnZlcnNpb249JzEuMCc+PC9YTUk+"
}
```

付録 I.12　XML スキーマファイル

運用管理 REST API のリクエスト，レスポンスで使用する XML スキーマのファイルの一覧を示します。

運用管理 REST API のリクエストパラメタ，レスポンスデータを XML 形式で定義した，XML スキーマをファイル形式で提供します。

XML スキーマファイルの格納先は，次のとおりです。

```
<CSCIWのインストールディレクトリ>/schema
```

運用管理 REST API と XML スキーマファイルの対応を次に示します。

表 I-11　運用管理 REST API の XML スキーマ

項番	運用管理 REST API		XML スキーマファイル
—	ビジネスプロセス定義		
1		ビジネスプロセス定義の登録	registerProcessDefinitionRequest.xsd getProcessDefinitionResponse.xsd
2		ビジネスプロセス定義の削除	—
—	振り分けルール		
3		振り分けルールの登録	registerCastingRequest.xsd getCastingRuleResponse.xsd
4		振り分けルールの削除	—
—	ジョブ		
5		ジョブの投入	createJobRequest.xsd getJobResponse.xsd
6		ジョブレコードの取得	getJobResponse.xsd
7		ジョブレコードの一覧取得	getJobListResponse.xsd

（凡例）

—：該当しない

付録 I.13　各運用管理 REST API の詳細

各運用管理 REST API の詳細を説明します。

(1)　ビジネスプロセス定義の登録

指定されたビジネスプロセス定義ファイル（.hbx）をワーク管理データベースに登録し，活性化状態に変更します。

ビジネスプロセス定義の登録を反映するためには，該当するシステム ID 内で実行しているすべての CSCIWManagementServer，および CSCIW を使用している Java アプリケーションの再起動が必要です。

メソッドと URL

```
POST /v1/process-definition
```

リクエストボディ

項番	名前	型	指定要否	内容
1	UserDescription	文字列	省略可	ユーザ記述子を指定します。 ビジネスプロセス定義を登録する場合，ユーザ記述子の値がビジネスプロセス定義の作成者として登録されます。 省略した場合はデフォルト値が指定されます。空文字列は指定できません。
2	DefinitionFile	文字列	必須	登録するビジネスプロセス定義ファイルを base64 エンコードした文字列を指定します。 base64 エンコードについては「付録 I.11　リクエストパラメタにファイルを指定する方法」を参照してください。 • BPMN 連携機能を使用する場合 ビジネスプロセス定義ファイルはciwtransbpmn コマンドで変換したファイルを使用してください。 ciwtransbpmn コマンドについては，マニュアル「uCosminexus Service Coordinator Interactive Workflow コマンド」の「ciwtransbpmn（BPMN ビジネスプロセス定義ファイルの CSCIW のビジネスプロセス定義ファイルへの変換）」を参照してください。 • BPMN 連携機能を使用しない場合 ビジネスプロセス定義ファイルは CSCIW-Definer でチェックを実施したファイルを使用してください。 CSCIW-Definer でのチェックについては，マニュアル「uCosminexus Service Coordinator Interactive Workflow ビジネスプロセス開発ガイド」の「ビジネスプロセス定義をチェックする」を参照してください。

リクエストボディの構造を次に示します。

項番	名前	出現回数
1	Parameter	1

項番	名前		出現回数
2		UserDescription	0 または 1
3		DefinitionFile	1

リクエスト例

リクエスト URL

```
POST http://restserver/csciwmngws/v1/process-definition
```

リクエストボディ（XML の場合）

```
<?xml version="1.0" encoding="UTF-8" ?>
<Parameter>
  <DefinitionFile>PD94bWwgdmVyc2lvbj0nMS4wJyBlbmNvZGluZz0nVVRGLTgnPz48IURPQ1RZUEUgWE1JPjx
YTUkgeG1pLnZlcnNpb249JzEuMCc+PC9YTUk+</DefinitionFile>
</Parameter>
```

リクエストボディ（JSON の場合）

```
{
  "DefinitionFile" : "PD94bWwgdmVyc2lvbj0nMS4wJyBlbmNvZGluZz0nVVRGLTgnPz48IURPQ1RZUEUgWE1
JPjxYTUkgeG1pLnZlcnNpb249JzEuMCc+PC9YTUk+"
}
```

レスポンス

登録したビジネスプロセス定義を返します。ビジネスプロセス定義のプロパティを次に示します。

項番	名前	型	内容
1	Author	文字列	ビジネスプロセス定義の作成者
2	CreationDate	日付	ビジネスプロセス定義の作成日
3	Description	文字列	ビジネスプロセス定義の説明
4	ID	数値	ビジネスプロセス定義の ID
5	Name	文字列	ビジネスプロセス定義の名称
6	Responsible	文字列	ビジネスプロセス定義の管理者 ID
7	StateCode	文字列	ビジネスプロセス定義の状態
8	ValidFromDate	日付	ビジネスプロセス定義の有効となる日
9	ValidToDate	日付	ビジネスプロセス定義の無効となる日
10	Version	数値	ビジネスプロセス定義のバージョン

レスポンスの構造を次に示します。

項番	名前		出現回数
1	ProcessDefinition		1
2		Author	1

項番	名前		出現回数
3		CreationDate	1
4		Description	1
5		ID	1
6		Name	1
7		Responsible	1
8		StateCode	1
9		ValidFromDate	1
10		ValidToDate	1
11		Version	1

ステータスコード

項番	ステータスコード	内容
1	201	成功
2	400	リクエストパラメタの不正
3	500	内部処理エラー

レスポンス例

XML の場合

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ProcessDefinition>
  <Author>user1</Author>
  <CreationDate>2016-12-06T15:18:35+09:00</CreationDate>
  <Description></Description>
  <ID>3</ID>
  <Name>販売業務</Name>
  <Responsible>user1</Responsible>
  <StateCode>b</StateCode>
  <ValidFromDate>ORIGIN</ValidFromDate>
  <ValidToDate>BEYOND</ValidToDate>
  <Version>2</Version>
</ProcessDefinition>
```

JSON の場合

```
{
  "Author" : "user1",
  "CreationDate" : "2016-12-06T15:18:35+09:00",
  "Description" : "",
  "ID" : "3",
  "Name" : "販売業務",
  "Responsible" : "user1",
  "StateCode" : "b",
  "ValidFromDate" : "ORIGIN",
  "ValidToDate" : "BEYOND",
```

```
  "Version" : "2"
}
```

(2) ビジネスプロセス定義の削除

指定されたビジネスプロセス定義を非活性化してから削除します。指定されたビジネスプロセス定義がすでに非活性であった場合，ビジネスプロセス定義の削除だけを行います。ただし，削除対象のビジネスプロセス定義に案件が存在するときは，ビジネスプロセス定義を非活性化および削除できません。

ビジネスプロセス定義が削除済み，または削除対象のビジネスプロセス定義が存在しない場合，リクエストは成功し，ステータスコード 204 を返します。

ビジネスプロセス定義の削除を反映するためには，該当するシステム ID 内で実行しているすべての CSCIWManagementServer，および CSCIW を使用している Java アプリケーションの再起動が必要です。

メソッドと URL

```
DELETE /v1/process-definition
```

クエリパラメタ

項番	名前	型	指定要否	内容
1	userdescription	文字列	省略可	ユーザ記述子を指定します。 省略した場合はデフォルト値が指定されます。 空文字列は指定できません。
2	name	文字列	必須	ビジネスプロセス定義名を指定します。
3	version	数値	必須	ビジネスプロセス定義バージョンを指定します。

リクエスト例

リクエスト URL

```
DELETE http://restserver/csciwmngws/v1/process-definition?name=BP001&version=1
```

レスポンス

レスポンスボディは空になります。

ステータスコード

項番	ステータスコード	内容
1	204	成功
2	400	リクエストパラメタの不正
3	500	内部処理エラー

レスポンス例

レスポンスボディは空になります。

(3)　振り分けルール定義の登録

指定された振り分けルール定義ファイル（.hcx）をワーク管理データベースに登録し，活性化状態に変更します。

振り分けルール定義の登録を反映するためには，該当するシステム ID 内で実行しているすべての CSCIWManagementServer，および CSCIW を使用している Java アプリケーションの再起動が必要です。

メソッドと URL

```
POST /v1/casting-rule
```

リクエストボディ

項番	名前	型	指定要否	内容
1	UserDescription	文字列	省略可	ユーザ記述子を指定します。 振り分けルール定義を登録する場合，ユーザ記述子の値が振り分けルール定義の作成者として登録されます。 省略した場合はデフォルト値が指定されます。空文字列は指定できません。
2	DefinitionFile	文字列	必須	登録する振り分けルール定義ファイルを base64 エンコードした文字列を指定します。 base64 エンコードについては，「付録 I.11　リクエストパラメタにファイルを指定する方法」を参照してください。 振り分けルール定義ファイルは CSCIW-Definer でチェックを実行してから登録してください。CSCIW-Definer でのチェックについては，マニュアル「uCosminexus Service Coordinator Interactive Workflow ビジネスプロセス開発ガイド」の「振り分けルール定義をチェックする」を参照してください。

リクエストボディの構造を次に示します。

項目	名前		出現回数
1	Parameter		1
2		UserDescription	0 または 1
3		DefinitionFile	1

リクエスト例

リクエスト URL

```
POST http://restserver/csciwmngws/v1/casting-rule
```

リクエストボディ（XML の場合）

```
<?xml version="1.0" encoding="UTF-8" ?>
<Parameter>
<DefinitionFile>PD94bWwgdmVyc2lvbj0nMS4wJyBlbmNvZGluZz0nU2hpZnRfSklTJz8+PCFET0NUWVBFIFhNST48WE1JIHhtaS52ZXJzaW9uPScxLjAnPjwvWE1JPg==</DefinitionFile>
</Parameter>
```

リクエストボディ（JSON の場合）

```
{
  "DefinitionFile": "PD94bWwgdmVyc2lvbj0nMS4wJyBlbmNvZGluZz0nU2hpZnRfSklTJz8+PCFET0NUWVBFIFhNST48WE1JIHhtaS52ZXJzaW9uPScxLjAnPjwvWE1JPg=="
}
```

レスポンス

登録した振り分けルール定義を返します。振り分けルール定義のプロパティを次に示します。

項目	名前	型	内容
1	Author	文字列	振り分けルール定義の作成者
2	CreationDate	日付	振り分けルール定義の作成日
3	Description	文字列	振り分けルール定義の説明
4	ID	数値	振り分けルール定義の ID
5	Name	文字列	振り分けルール定義の名称
6	Responsible	文字列	振り分けルール定義の管理者 ID
7	StateCode	文字列	振り分けルール定義の状態
8	ValidFromDate	日付	振り分けルール定義の有効となる日
9	ValidToDate	日付	振り分けルール定義の無効となる日
10	Version	数値	振り分けルール定義のバージョン

レスポンスの構造を次に示します。

項目	名前	出現回数
1	CastingRule	1
2	Author	1
3	CreationDate	1
4	Description	1
5	ID	1
6	Name	1
7	Responsible	1
8	StateCode	1
9	ValidFromDate	1

項目	名前		出現回数
10		ValidToDate	1
11		Version	1

ステータスコード

項目	ステータスコード	内容
1	201	成功
2	400	リクエストパラメタの不正
3	500	内部処理エラー

レスポンス例

XML の場合

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<CastingRule>
  <Author>user1</Author>
  <CreationDate>2019-10-06T15:18:35+09:00</CreationDate>
  <Description></Description>
  <ID>12</ID>
  <Name>CastSQL</Name>
  <Responsible>user1</Responsible>
  <StateCode>b</StateCode>
  <ValidFromDate>ORIGIN</ValidFromDate>
  <ValidToDate>BEYOND</ValidToDate>
  <Version>2</Version>
</CastingRule>
```

JSON の場合

```
{
  "Author" : "user1",
  "CreationDate" : "2019-10-06T15:18:35+09:00",
  "Description" : "",
  "ID" : "12",
  "Name" : "CastSQL",
  "Responsible" : "user1",
  "StateCode" : "b",
  "ValidFromDate" : "ORIGIN",
  "ValidToDate" : "BEYOND",
  "Version" : "2"
}
```

(4) 振り分けルール定義の削除

指定された振り分けルール定義を非活性状態にしてから削除します。ただし，削除対象の振り分けルール定義がビジネスプロセス定義で使用されているときは，非活性化および削除することはできません。

指定された振り分けルール定義の状態がすでに非活性であった場合，振り分けルール定義の削除だけを行います。

振り分けルール定義が削除済み，または削除対象の振り分けルール定義が存在しない場合，リクエストは成功し，ステータスコード 204 を返します。

振り分けルール定義の削除を反映するためには，該当するシステム ID 内で実行しているすべての CSCIWManagementServer，および CSCIW を使用している Java アプリケーションの再起動が必要です。

メソッドと URL

```
DELETE /v1/casting-rule
```

クエリパラメタ

項番	名前	型	指定要否	内容
1	userdescription	文字列	省略可	ユーザ記述子を指定します。 省略した場合はデフォルト値が指定されます。空文字列は指定できません。
2	name	文字列	必須	振り分けルール定義名を指定します。
3	version	数値	必須	振り分けルール定義バージョンを指定します。

リクエスト例

リクエスト URL

```
DELETE http://restserver/csciwmngws/v1/casting-rule?name=RULE001&version=1
```

レスポンス

レスポンスボディは空になります。

ステータスコード

項番	ステータスコード	内容
1	204	成功
2	400	リクエストパラメタの不正
3	500	内部処理エラー

レスポンス例

レスポンスボディは空になります。

(5) ジョブの投入

ジョブを投入します。

ジョブは非同期で実行されるため，この API のレスポンスからジョブの実行結果を確認することはできません。投入したジョブの実行結果を確認するには，ジョブレコードの取得 API を実行してください。

メソッドと URL

```
POST /v1/job/create
```

リクエストボディ

項番	名前	型	指定要否	内容
1	UserDescription	文字列	省略可	ユーザ記述子を指定します。 省略した場合はデフォルト値が指定されます。空文字列は指定できません。
2	FunctionName	文字列	必須	ジョブの機能を表すファンクション名を指定します。 機能ごとのファンクション名の詳細は，「付録 I.14　投入するジョブの機能ごとの指定値と戻り値」を参照してください。
3	ArgumentDataList	配列	省略可	ジョブの機能に渡す引数データを指定します。 機能ごとの引数データの詳細は，「付録 I.14　投入するジョブの機能ごとの指定値と戻り値」の各機能の引数データを参照してください。
4	Data	オブジェクト	必須	データオブジェクトを指定します。 次の 2 つの要素（Key および Value）から構成されます。
5	Key	文字列	必須	データのキー名を指定します。 空文字列は指定できません。
6	Value	文字列	省略可	データのキーに対応する値を指定します。 省略した場合，キーに対応する値は設定されずに null が指定されます。空文字列は指定できません。

リクエストボディの構造を次に示します。

項番	名前	出現回数
1	Parameter	1
2	UserDescription	0 または 1
3	FunctionName	1
4	ArgumentDataList	0 または 1
5	Data	1 以上
6	Key	1
7	Value	0 または 1

リクエスト例

リクエスト URL

```
POST http://restserver/csciwmngws/v1/job/create
```

リクエストボディ（XML の場合）

```
<?xml version="1.0" encoding="UTF-8" ?>
<Parameter>
  <FunctionName>delpi</FunctionName>
  <ArgumentDataList>
```

```
    <Data>
      <Key>DaysFromClosedDate</Key>
      <Value>3</Value>
    </Data>
    <Data>
      <Key>ProcessDefinitionName</Key>
      <Value>販売業務</Value>
    </Data>
    <Data>
      <Key>ProcessDefinitionVersion</Key>
      <Value>5</Value>
    </Data>
    <Data>
      <Key>IsAll</Key>
      <Value>true</Value>
    </Data>
    <Data>
      <Key>Name</Key>
    </Data>
    <Data>
      <Key>Unit</Key>
      <Value>30</Value>
    </Data>
  </ArgumentDataList>
</Parameter>
```

リクエストボディ（JSON の場合）

```
{
  "FunctionName" : "delpi",
  "ArgumentDataList" : {
    "Data" : [
      {
        "Key" : "DaysFromClosedDate",
        "Value" : "3"
      },
      {
        "Key" : "ProcessDefinitionName",
        "Value" : "販売業務"
      },
      {
        "Key" : "ProcessDefinitionVersion",
        "Value" : "5"
      },
      {
        "Key" : "IsAll",
        "Value" : "true"
      },
      {
        "Key" : "Name"
      },
      {
        "Key" : "Unit",
        "Value" : "30"
      }
    ]
```

```
  }
}
```

レスポンス

投入したジョブのジョブレコードを返します。ジョブレコードの詳細については，「付録 I.2(2)　ジョブレコードの取得による非同期処理の実行状況の確認」を参照してください。

ジョブレコードのプロパティを次に示します。

項番	名前	型	内容
1	ArgumentDataList	配列	ジョブが使用する引数データ
2	ClosedDate	日付	ジョブの終了日時
3	Creator	文字列	ジョブの投入者
4	FunctionName	文字列	ジョブのファンクション名
5	ID	数値	ジョブのジョブ ID
6	ResultDataList	配列	ジョブの結果データ
7	ReturnCode	文字列	ジョブのリターンコード
8	StartDate	日付	ジョブの開始日時
9	StateCode	文字列	ジョブの状態

レスポンスの構造を次に示します。

項番	名前	出現回数
1	Job	1
2	ArgumentDataList	1
3	Data	0 または 1 以上
4	Key	1
5	Value	0 または 1
6	ClosedDate	1
7	Creator	1
8	FunctionName	1
9	ID	1
10	ResultDataList	1
11	Data	0 または 1 以上
12	Key	1
13	Value	0 または 1
14	ReturnCode	1

項番	名前		出現回数
15		StartDate	1
16		StateCode	1

ジョブの実行結果を確認するには，レスポンスとして返却されるジョブのジョブ ID（ID）を使用して，ジョブレコードの取得 API を実行してください。

ステータスコード

項番	ステータスコード	内容
1	202	成功（Accepted）
2	400	リクエストパラメタの不正
3	500	内部処理エラー

レスポンス例

XML の場合

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Job>
  <ArgumentDataList>
    <Data>
      <Key>DaysFromClosedDate</Key>
      <Value>3</Value>
    </Data>
    <Data>
      <Key>ProcessDefinitionName</Key>
      <Value>販売業務</Value>
    </Data>
    <Data>
      <Key>ProcessDefinitionVersion</Key>
      <Value>5</Value>
    </Data>
    <Data>
      <Key>IsAll</Key>
      <Value>true</Value>
    </Data>
    <Data>
      <Key>Name</Key>
    </Data>
    <Data>
      <Key>Unit</Key>
      <Value>30</Value>
    </Data>
  </ArgumentDataList>
  <ClosedDate></ClosedDate>
  <Creator>csciwmngws</Creator>
  <FunctionName>delpi</FunctionName>
  <ID>1481005115000</ID>
  <ResultDataList></ResultDataList>
  <ReturnCode></ReturnCode>
  <StartDate>2016-12-06T15:18:35+09:00</StartDate>
```

```
  <StateCode>d</StateCode>
</Job>
```

JSON の場合

```
{
  "ArgumentDataList" : {
    "Data" : [
      {
        "Key" : "DaysFromClosedDate",
        "Value" : "3"
      },
      {
        "Key" : "ProcessDefinitionName",
        "Value" : "販売業務"
      },
      {
        "Key" : "ProcessDefinitionVersion",
        "Value" : "5"
      },
      {
        "Key" : "IsAll",
        "Value" : "true"
      },
      {
        "Key" : "Name"
      },
      {
        "Key" : "Unit",
        "Value" : "30"
      }
    ]
  },
  "ClosedDate" : "2016-12-06T15:19:53+09:00",
  "Creator" : "csciwmngws",
  "FunctionName" : "delpi",
  "ID" : "1481005115000",
  "ResultDataList" : "",
  "ReturnCode" : "",
  "StartDate" : "2016-12-06T15:18:35+09:00",
  "StateCode" : "d"
}
```

(6) ジョブレコードの取得

指定されたジョブ ID のジョブレコードを取得します。

メソッドと URL

```
GET /v1/job/<ジョブID>※
```

注※　<ジョブ ID>：取得するジョブレコードのジョブのID（必須）

クエリパラメタ

項番	名前	型	指定要否	内容
1	userdescription	文字列	省略可	ユーザ記述子を指定します。 省略した場合はデフォルト値が指定されます。空文字列は指定できません。

リクエスト例

リクエスト URL

```
GET http://restserver/csciwmngws/v1/job/1481005115000
```

レスポンス

指定されたジョブレコードを返します。レスポンスのジョブレコードの詳細については,「付録 I.2(2)　ジョブレコードの取得による非同期処理の実行状況の確認」を参照してください。

ジョブレコードのプロパティを次に示します。

項番	名前	型	内容
1	ArgumentDataList	配列	ジョブが使用する引数データ
2	ClosedDate	文字列	ジョブの終了日時
3	Creator	文字列	ジョブの投入者
4	FunctionName	日付	ジョブのファンクション名
5	ID	文字列	ジョブのジョブ ID
6	ResultDataList	配列	ジョブの結果データ
7	ReturnCode	文字列	ジョブのリターンコード
8	StartDate	文字列	ジョブの開始日時
9	StateCode	文字列	ジョブの状態

レスポンスの構造を次に示します。

項番	名前	出現回数
1	Job	1
2	ArgumentDataList	1
3	Data	0 または 1 以上
4	Key	1
5	Value	0 または 1
6	ClosedDate	1
7	Creator	1
8	FunctionName	1

項番	名前	出現回数
9	ID	1
10	ResultDataList	1
11	Data	0 または 1 以上
12	Key	1
13	Value	0 または 1
14	ReturnCode	1
15	StartDate	1
16	StateCode	1

ジョブの結果データ（ResultDataList）およびジョブのリターンコード（ReturnCode）の詳細は，「付録 I.14　投入するジョブの機能ごとの指定値と戻り値」を参照してください。

ステータスコード

項番	ステータスコード	内容
1	200	成功
2	400	リクエストパラメタの不正
3	404	データが見つからなかった
4	500	内部処理エラー

レスポンス例

XML の場合

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Job>
  <ArgumentDataList>
    <Data>
      <Key>DaysFromClosedDate</Key>
      <Value>3</Value>
    </Data>
  </ArgumentDataList>
  <ClosedDate>2016-12-06T15:19:53+09:00</ClosedDate>
  <Creator>csciwmngws</Creator>
  <FunctionName>delpi</FunctionName>
  <ID>1481005115000</ID>
  <ResultDataList>
    <Data>
      <Key>NumOfDeletedPI</Key>
      <Value>2078</Value>
    </Data>
    <Data>
      <Key>NumOfDeletedChildPI</Key>
      <Value>122</Value>
    </Data>
  </ResultDataList>
  <ReturnCode>0</ReturnCode>
```

```
  <StartDate>2016-12-06T15:18:35+09:00</StartDate>
  <StateCode>o</StateCode>
</Job>
```

JSON の場合

```
{
  "ArgumentDataList" : {
    "Data" : [
      {
        "Key" : "DaysFromClosedDate",
        "Value" : "3"
      }
    ]
  },
  "ClosedDate" : "2016-12-06T15:19:53+09:00",
  "Creator" : "csciwmngws",
  "FunctionName" : "delpi",
  "ID" : "1481005115000",
  "ResultDataList" : {
    "Data" : [
       {
         "Key" : "NumOfDeletedPI",
         "Value" : "2078"
       },
     {
        "Key" : "NumOfDeletedChildPI",
        "Value" : "122"
      }
    ]
  },
  "ReturnCode" : "0",
  "StartDate" : "2016-12-06T15:18:35+09:00",
  "StateCode" : "o"
}
```

(7) ジョブレコードの一覧取得

指定された条件に該当するジョブレコードの一覧を取得します。ジョブレコードの一覧は開始時刻日時でソートされます。

メソッドと URL

```
GET /v1/job
```

クエリパラメタ

項番	名前	型	指定要否	内容
1	userdescription	文字列	省略可	ユーザ記述子を指定します。 省略した場合はデフォルト値が指定されます。デフォルト値はセットアッププロパティファイルの MngRestServiceUserDescription の値が使用されます。空文字列は指定できません。

項番	名前	型	指定要否	内容
2	functionname	文字列	省略可	検索条件としてファンクション名を指定します。 省略した場合は検索条件としてファンクション名を使用しません。
3	statecode	文字列	省略可	検索条件としてジョブレコードの状態を指定します。 複数の状態を指定する場合は，状態のコード値を同時に指定します。例えば「正常終了（o)」,「異常終了（v)」のジョブレコードを取得する場合は"ov"と指定します。ジョブの状態のコード値については， 「付録 I.2(2)　ジョブレコードの取得による非同期処理の実行状況の確認」を参照してください。 省略した場合は検索条件としてジョブの状態を使用しません。 空文字列は指定できません。
4	creator	文字列	省略可	検索条件として投入者を指定します。 省略した場合は検索条件として投入者を使用しません。
5	sort	文字列	省略可	取得するジョブレコード一覧のソート順を指定します。 • ASC：開始日時の昇順にソートします。 • DESC：開始日時の降順にソートします。 省略した場合は昇順（ASC）でソートされます。空文字列は指定できません。
6	offset	数値	省略可	取得するジョブレコード一覧のオフセットを指定します。 先頭は 0 になります。省略した場合は 0 が指定されます。0 未満の値は指定できません。
7	maxcount	数値	省略可	取得するジョブレコード一覧の最大取得数を指定します。 省略した場合はデフォルト値が指定されます。デフォルト値はセットアッププロパティファイルの MngRestServiceResponseMaxCount の値が使用されます。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。

リクエスト例

リクエスト URL

```
GET http://restserver/csciwmngws/v1/job?functionname=delpi&statecode=do&creator=csciwmngws&sort=ASC
```

レスポンス

検索条件に該当するジョブレコードの一覧を返します。ジョブレコードの詳細については，「付録 I.2(2)　ジョブレコードの取得による非同期処理の実行状況の確認」を参照してください。

ジョブレコードのプロパティを次に示します。

項番	名前	型	内容
1	ArgumentDataList	配列	ジョブが使用する引数データ

項番	名前	型	内容
2	ClosedDate	日付	ジョブの終了日時
3	Creator	文字列	ジョブの投入者
4	FunctionName	文字列	ジョブのファンクション名
5	ID	数値	ジョブのジョブ ID
6	ResultDataList	配列	ジョブの結果データ
7	ReturnCode	文字列	ジョブのリターンコード
8	StartDate	日付	ジョブの開始日時
9	StateCode	文字列	ジョブの状態

レスポンスの構造を次に示します。

項番	名前	出現回数
1	Jobs	1
2	Job	0 または 1 以上
3	ArgumentDataList	1
4	Data	0 または 1 以上
5	Key	1
6	Value	0 または 1
7	ClosedDate	1
8	Creator	1
9	FunctionName	1
10	ID	1
11	ResultDataList	1
12	Data	0 または 1 以上
13	Key	1
14	Value	0 または 1
15	ReturnCode	1
16	StartDate	1
17	StateCode	1

ジョブの結果データ（ResultDataList）およびジョブのリターンコード（ReturnCode）の詳細は，「付録 I.14　投入するジョブの機能ごとの指定値と戻り値」を参照してください。

ステータスコード

項番	ステータスコード	内容
1	200	成功（データが０件の場合も成功）
2	400	リクエストパラメタの不正
3	500	内部処理エラー

レスポンス例

XML の場合

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Jobs>
  <Job>
    <ArgumentDataList>
      <Data>
        <Key>DaysFromClosedDate</Key>
        <Value>3</Value>
      </Data>
    </ArgumentDataList>
    <ClosedDate></ClosedDate>
    <Creator>csciwmngws</Creator>
    <FunctionName>delpi</FunctionName>
    <ID>1482412942222</ID>
    <ResultDataList></ResultDataList>
    <ReturnCode></ReturnCode>
    <StartDate>2016-12-22T22:22:22+09:00</StartDate>
    <StateCode>d</StateCode>
  </Job>
  <Job>
    <ArgumentDataList>
      <Data>
        <Key>DaysFromClosedDate</Key>
        <Value>10</Value>
      </Data>
    </ArgumentDataList>
    <ClosedDate>2016-12-06T15:19:53+09:00</ClosedDate>
    <Creator>csciwmngws</Creator>
    <FunctionName>delpi</FunctionName>
    <ID>1481005115000</ID>
    <ResultDataList>
      <Data>
        <Key>NumOfDeletedPI</Key>
        <Value>2078</Value>
      </Data>
      <Data>
        <Key>NumOfDeletedChildPI</Key>
        <Value>122</Value>
      </Data>
    </ResultDataList>
    <ReturnCode>0</ReturnCode>
    <StartDate>2016-12-06T15:18:35+09:00</StartDate>
    <StateCode>o</StateCode>
  </Job>
</Jobs>
```

JSON の場合


```
{
  "Job" : [
    {
      "ArgumentDataList" : {
        "Data" : [
          {
            "Key" : "DaysFromClosedDate",
            "Value" : "3"
          }
        ]
      },
      "ClosedDate" : "",
      "Creator" : "csciwmngws",
      "FunctionName" : "delpi",
      "ID" : "1482412942222",
      "ResultDataList" : "",
      "ReturnCode" : "0",
      "StartDate" : "2016-12-22T22:22:22+09:00",
      "StateCode" : "d"
    },
    {
      "ArgumentDataList" : {
        "Data" : [
          {
            "Key" : "DaysFromClosedDate",
            "Value" : "10"
          }
        ]
      },
      "ClosedDate" : "2016-12-06T15:19:53+09:00",
      "Creator" : "csciwmngws",
      "FunctionName" : "delpi",
      "ID" : "1481005115000",
      "ResultDataList" : {
        "Data" : [
          {
            "Key" : "NumOfDeletedPI",
            "Value" : "2078"
          },
          {
            "Key" : "NumOfDeletedChildPI",
            "Value" : "122"
          }
        ]
      },
      "ReturnCode" : "0",
      "StartDate" : "2016-12-06T15:18:35+09:00",
      "StateCode" : "o"
    }
  ]
}
```

付録 I.14　投入するジョブの機能ごとの指定値と戻り値

ここでは，ジョブの機能ごとに，リクエストパラメタのファンクション名と引数データに指定する値，およびレスポンスボディの結果データとリターンコードに設定される値の詳細を説明します。

(1)　案件一括削除の場合

指定された条件に該当する案件を一括削除する場合は，ファンクション名（FunctionName）に「delpi」を指定します。案件一括削除は，ciwdelpi コマンドと同様の処理を実行します。ciwdelpi コマンドについては，マニュアル「uCosminexus Service Coordinator Interactive Workflow コマンド」の「ciwdelpi (案件の削除)」を参照してください

(a)　案件一括削除の引数データの指定値

引数データ（ArgumentDataList）のKey とValue に指定する値は，ciwdelpi コマンドのオプションとオプション引数に対応しています。

引数データのKey とValue に指定する値と，それに対応するciwdelpi コマンドのオプションの対応を次の表に示します。

表 I-12　引数データに指定する値および ciwdelpi コマンドのオプションの対応

項番	Key に指定する値	型	Key の指定要否	Value に指定する値	対応する ciwdelpi コマンドのオプション
1	DaysFromClosedDate	数値	次のどちらかが必須 • DaysFromClosedDate • DaysFromStartDate DaysFromStartDate と同時には指定できません。	削除対象の案件の終了日時からの経過日数を指定します。	-dfc
2	DaysFromStartDate	数値	次のどちらかが必須 • DaysFromClosedDate • DaysFromStartDate DaysFromClosedDate と同時には指定できません。	削除対象の案件の開始日時からの経過日数を指定します。	-dfs
3	UseTimeOffset	文字列	省略可	経過日数の区切りを，true または false で指定します。 • true：経過日数の区切りを「00（時）：00（分）：00（秒）」にします。 • false：DaysFromClosedDate または	-ofs

項番	Key に指定する値	型	Key の指定要否	Value に指定する値	対応する ciwdelpi コマンドのオプション
3	UseTimeOffset	文字列	省略可	DaysFromStartDateで指定した経過日数×24時間以上経過した案件を削除対象とします。 省略した場合は，false が指定されます。	-ofs
4	IsAll	文字列	省略可	「完了」状態以外の案件を削除する場合に指定します。 • true：「完了」状態以外の案件も削除します。 • false：「完了」状態の案件だけを削除します。 省略した場合は，false が指定されます。	-all
5	ProcessDefinitionName	文字列	省略可	ビジネスプロセス定義名を指定します。 指定したビジネスプロセス定義名を使用して生成したすべての案件が削除対象になります。	-bpn
6	ProcessDefinitionVersion	数値	省略可 ProcessDefinitionName を指定した場合にだけ指定できます。	ビジネスプロセス定義のバージョンを指定します。 指定したビジネスプロセス定義の，指定したバージョンのビジネスプロセス定義を使用して生成した，すべての案件が削除対象になります。	-bpv
7	Name	文字列	省略可	案件名を指定します。 指定した案件名を使用して生成したすべての案件が削除対象になります。案件名が設定されていない案件を削除対象にする場合には，値を指定しないでください。また，案件名はワイルドカードが指定できます。 例：案件 A%	-pin

項番	Key に指定する値	型	Key の指定要否	Value に指定する値	対応する ciwdelpi コマンドのオプション
8	Escape	文字列	省略可 Name を指定した場合にだけ指定できます。	Name を指定した場合に案件名で使用するエスケープ文字を指定します。	-esc
9	Unit	文字列	省略可	削除処理のトランザクションの処理単位を指定します。 省略した場合は，デフォルト値の 100 が指定されます。	-unt

(b)　案件一括削除のジョブの取得時のリターンコード

案件の一括削除のジョブが完了すると，ciwdelpi コマンドの戻り値と同じ値がジョブレコードのリターンコード（ReturnCode）に設定されます。また，リターンコードの値に対応してジョブレコードの状態（StateCode）も設定されます。

リターンコードと状態の対応を次の表に示します。

表 I-13　リターンコードと状態の対応

リターンコード	状態
0（正常終了）	o（正常終了）
20（条件に指定した案件が 1 件も存在しない）	o（正常終了）
それ以外	v（異常終了）

(c)　案件一括削除の結果データに設定される値

案件の一括削除が完了すると，ジョブレコードの結果データ（ResultData）のKey とValue に次の表に示す内容が設定されます。

表 I-14　結果データの Key と Value に設定される内容

項番	Key の内容	Value の内容
1	NumOfDeletedPI	削除した案件の数
2	NumOfDeletedChildPI	削除した子案件の数 BPMN 連携機能を使用しない場合は 0 になります。

付録 J　J2EE サーバを V9 互換モードで使用する場合の設定

Cosminexus J2EE サーバを V9 互換モードで動作させる場合の設定について説明します。

付録 J.1　互換モード用のファイルへの入れ替え

Cosminexus J2EE サーバを V9 互換モードで動作させる場合，推奨モードで動作させる場合とは異なる CSCIW のear ファイルを使用します。CSCIW のインストール時には推奨モード用のファイルが配備されるため，実行環境を構築する前に，次に示すコピー元のフォルダに格納されているすべてのファイルをコピー先のフォルダに上書きコピーしてください。

コピー元のフォルダ

- Windows の場合

```
<CSCIWインストールディレクトリ>¥lib¥compatible
```

- UNIX の場合

```
/opt/hitachi/CSCIW/lib/compatible
```

コピー先のフォルダ

- Windows の場合

```
<CSCIWインストールディレクトリ>¥lib
```

- UNIX の場合

```
/opt/hitachi/CSCIW/lib
```

互換モード用の CSCIW のear ファイルを推奨モード用のファイルに戻す場合は，次に示すフォルダに格納されているすべてのファイルをコピー先のフォルダに上書きコピーしてください。そのあと，実行環境を構築し直してください。

- Windows の場合

```
<CSCIWインストールディレクトリ>¥lib¥standard
```

- UNIX の場合

```
/opt/hitachi/CSCIW/lib/standard
```

付録 J.2　コンテナ拡張ライブラリに取り込むライブラリの変更

Cosminexus J2EE サーバを V9 互換モードで動作させる場合，推奨モードで動作させる場合とは異なるライブラリをコンテナ拡張ライブラリに取り込みます。

Cosminexus の J2EE サーバ用オプションのadd.class.path に，コンテナ拡張ライブラリ用の次に示すファイルを設定してください。

- CSCIW ライブラリ（csciw.jar）
- JAX-RS ライブラリ（cjjaxws.jar およびcjjaxrs.jar）

  JAX-RS ライブラリは，Cosminexus に同梱されているものを使用してください。別に入手した Jersey などのライブラリは使用できません。
- JDBC ドライバのライブラリ

  JDBC ドライバは，HiRDB の場合は「HiRDB Type4 JDBC Driver」，ORACLE の場合は「Oracle JDBC Thin Driver」，SQL Server の場合は「SQL Server の JDBC ドライバ」を追加してください。

Windows の場合

```
add.class.path=<CSCIWインストールディレクトリ>¥lib¥csciw.jar
add.class.path=<cosminexus.home>¥jaxws¥lib¥cjjaxws.jar
add.class.path=<cosminexus.home>¥jaxrs¥lib¥cjjaxrs.jar
add.class.path=<JDBCドライバのライブラリのパス>
```

UNIX の場合

```
add.class.path=/opt/hitachi/CSCIW/lib/csciw.jar
add.class.path=/opt/Cosminexus/jaxws/lib/cjjaxws.jar
add.class.path=/opt/Cosminexus/jaxrs/lib/cjjaxrs.jar
add.class.path=<JDBCドライバのライブラリのパス>
```

アプリケーション呼び出しサービスを使用する場合は，add.class.path にcsciwbpmn.jar ファイルを追加してください。

- Windows の場合

```
add.class.path=<CSCIWインストールフォルダ>¥lib¥csciwbpmn.jar
```

- UNIX の場合

```
add.class.path=/opt/hitachi/CSCIW/lib/csciwbpmn.jar
```

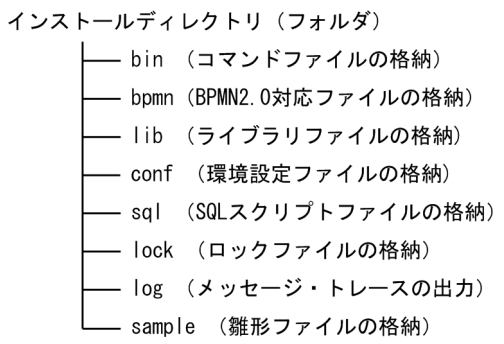
add.class.path については，マニュアル「Cosminexus アプリケーションサーバ リファレンス 定義編(サーバ定義)」の「usrconf.cfg（J2EE サーバ用オプション定義ファイル)」の説明を参照してください。

csciw.jar ファイルはコンテナ拡張ライブラリへ取り込むと，アプリケーションサーバのプロセス内で有効になります。このため，各アプリケーションのear にcsciw.jar ファイルを組み込まないでください。

付録 K　インストールディレクトリ（フォルダ）

CSCIW をインストールした場合のディレクトリ（フォルダ）構成を次の図に示します。

図 K-1　インストール時に作成するディレクトリ（フォルダ）



●bin

CSCIW の各機能の実行形式ファイルおよびコマンドの実行形式ファイルが格納されています。

●bpmn

BPMN2.0 対応のファイルが格納されています。

●lib

CSCIW として共通に使用するライブラリが格納されています。

●conf

環境構築時に作成されるコンフィグレーション情報などが格納されています。

●sql

CSCIW の SQL スクリプトファイルが格納されています。

●lock

CSCIW の排他制御用のファイルが格納されています。

●log

CSCIW の各機能が出力したメッセージやトレース情報を記録したファイルが格納されています。

●sample

ひな形となるファイルが格納されています。

付録 L　各バージョンの変更内容

付録 L.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。
- 使用できる Web ブラウザから Internet Explorer を削除しました。また，Microsoft Edge を追加しました。
- 性能解析トレースとして出力する情報に，次のメソッドを追加しました。
  - getActivityDefinition
  - getActivityDefinitionsList
  - startAndAdhocInstantiateActivityInstanceByADID
  - startAndAdhocInstantiateActivityInstanceByADName
- JP1 によるログファイルの監視の記述を追加しました。
- ワーク管理システムのバージョンアップの概念図を追加しました。
- ワーク管理データベースを移行する手順について，CSCIW03-10 から CSCIW03-11 へ移行する場合の手順を追加しました。
- トレース容量の見積もりに次を追加しました。
  - getActivityDefinition
  - getActivityDefinitionsList
- 共通設定ファイルに指定するパラメタに次のパラメタを追加しました。
  - AppCallServiceDebugRestBody
  - AppCallServiceInitialInterval
- アプリケーション呼び出し情報ファイルに指定するパラメタに rest.request.idempotency を追加しました。
- リトライの対象から外れた作業に関する運用での ciwchgapwork コマンドの実行例と手順を変更しました。
- マニュアル「JP1 Version 12 JP1/Base 運用ガイド」を追加しました。

付録 L.2　03-10 の変更内容

- 次の図を変更しました。
  - ワーク管理システムのプログラム構成

  - ワーク管理システムの基本的なシステム構成モデル
- システムを構成する上での注意の説明を変更しました。
- 「HiRDB の SQL 最適化について」の記述を追加しました。
- 2 とおりの手順で場合分けしました。
  - 運用管理ポータルを使用して設定する場合
  - 定義ファイルおよびサーバ管理コマンドを使用して設定する場合
- CSCIW ライブラリのファイルに「csciwbpmn.jar」を追加しました。
- アプリケーション呼び出し情報ファイルに次のパラメタを追加しました。
  - rest.request.connect.timeout
- Cosminexus の起動の手順を追加しました。それに伴い，既存の手順の中に含まれていた Cosminexus の起動の説明を削除しました。
- BPMN 連携機能を使用する場合の記述を削除しました。
- 2 とおりの手順で場合分けしました。
  - 運用管理ポータルを使用する場合
  - サーバ管理コマンドを使用する場合
- 運用管理ポータルを使用した場合の記述を追加しました。
- 付録に記載していたバージョンアップ手順の説明を 9 章に移動し，内容を修正しました。
- セットアッププロパティファイルに次のキーを追加しました。
  - AdmAcDispMaxCount
  - AdmAgDispMaxCount
- リトライ時の複数回呼び出しに関する説明を修正しました。

付録 L.3　03-00 の変更内容

- Cosminexus V11 に対応しました。
- CSCIW-Definer の動作環境として使用するプログラムを「uCosminexus Business Process Developer」に変更しました。
  また，プログラムの変更に伴い「CIW プラグイン」に関する説明を，削除または変更しました。
- 対象となる J2EE の起動モードについての注意書きを削除しました。
- 案件運用操作のデータベースコネクション数の見積もりに，案件一括削除時の見積もり数を追加しました。
- Cosminexus J2EE サーバを V9 互換モードで動作させる場合の設定方法を追加しました。
- コンテナ拡張ライブラリに取り込む JAR ファイルを変更しました。

- REST アプリケーション呼び出しの通信タイムアウトの設定についての説明を変更しました。
- DB Connector のリソースアダプタの表示名に CSCIW のデフォルトの値を設定することで，各アプリケーションのデータソース表示名の設定手順を省略できるようにしました。
- CSCIWManagementServer および案件運用操作のコンテキストルートの設定の説明を，リバースプロシキを配置する設定の説明に変更しました。
- 案件運用操作のセキュリティロールの設定で，ユーザ名とパスワードにマルチバイト文字は使用できないという注意を追加しました。
- Windows の場合に業務プログラムの実行権限を設定するときの注意事項から OS のバージョンごとの操作方法を削除しました。
- コマンド用環境設定ファイルの CSCIWCMD_JVM_ARCHITECTURE キーにデフォルト値として「64」を設定し，ユーザによる設定を不要としました。それに伴い，「Cosminexus の 64 ビットコンポーネントを使用する場合について」の説明を削除しました。
- CSCIW のコマンドで実行できる機能の一部を，RESTful API で実行できるようにしました。
- CSCIW-Definer のインストール先フォルダおよびログの出力先フォルダを変更しました。
- セットアッププロパティファイルに次のキーを追加しました。
  - AdmDelPIResultsDispMaxCount
  - AdmDelPIUnit
  - AdmDelPIUseTimeOffset

  また，次のキーのデフォルト値を変更しました。
  - TraceFileNum
  - CmdTraceFileNum
  - AdmTraceFileNum
  - TraceFileSize
  - CmdTraceFileSize
  - AdmTraceFileSize
  - MsgTimeToDelete
  - TraceTimeToDelete
  - AdmMsgTimeToDelete
  - AdmTraceTimeToDelete

  また，次のキーの最小値を変更しました。
  - AdmBpCrDispMaxCount
  - AdmPIDispMaxCount
  - AdmAIDispMaxCount

- AdmWIDispMaxCount

- CSCIWCMD_JVM_CLPATH に JDBC のクラスパスを設定する指定例から次の指定例を削除しました。
  - Oracle（JDK5 使用）の場合
  - Oracle（JDK6 使用）の場合
  - Oracle（JDK7 使用）の場合
  - SQL Server（JDK5 使用）の場合
  - SQL Server（JDK6 使用）の場合

  また，Oracle の場合の指定例の jar ファイルを変更しました。
- <SYSTEMID>_APPLICATION_LOCK_INFO の項番 2～8 の，ORACLE のデータ型が誤っていたため修正しました。
- ワーク管理データベースを移行する手順について，CSCIW 02-30 から CSCIW 03-00 へ移行する場合の手順を追加しました。
- アプリケーション呼び出しサービス自身の障害の注意事項として，障害復旧間隔の求め方の説明を追加しました。
- アプリケーション呼び出しサービスの REST 通信に関する設定の説明を変更しました。
- アプリケーション呼び出しサービスで使用する共通設定ファイルの次のパラメタの指定を省略したときの仮定値を変更しました。
  - AppCallServiceTraceFileNum
  - AppCallServiceTraceFileSize
- アプリケーション呼び出し情報ファイルで使用できる組み込み変数「案件処理期限」の説明に，代入先についての説明を追加しました。
- アプリケーション呼び出し情報ファイルの次のキーに，注意事項を追加しました。
  - rest.request.header.filepath
  - rest.request.stylesheet.filepath
- アプリケーション呼び出し情報ファイルの「rest.request.body.key.<key 要素値>」は，rest.request.method が DELETE の場合には指定できないように変更しました。

付録 L.4　02-30 の変更内容

- BPMN 連携機能を使用しないでアプリケーション呼び出しサービスを使用する方法を追加しました。
- Firefox および Chrome に対応しました。
- Cosminexus 09-00 以前で使用していた Server Plug-in の操作の代替手段の説明を削除しました。
- アプリケーションサーバの J2EE サーバ用オプション定義ファイルへの設定方法を変更しました。

- アプリケーションサーバのセキュリティロールの設定で，ユーザ名とパスワードに指定できない文字の説明を追加しました。
- CSCIWManagementServer のプロパティ設定のセキュリティロール名の登録で，設定先を WAR 属性ファイルに変更しました。
- CSCIWManagementServer のコンテキストルートの説明を追加しました。
- データベースへのアクセス権の削除が不要な場合の説明を追加しました。
- 次のファイルのエンコーディングについての説明を変更しました。
  - セットアッププロパティファイル
  - 共通設定ファイル
- ciwdelpi コマンド実行時の排他要求数の見積もり（HiRDB 限定）に，次のテーブルを追加しました。
  - サブプロセス用マルチインスタンス管理テーブル
  - アドホック・サブプロセス管理テーブル
- ワーク管理データベースの移行に SQL スクリプトファイルを「02-20 から 02-30 へ移行する場合」を追加しました。
- バージョンアップ時の環境設定に，コンテナ拡張ライブラリへの jar の取り込みについての説明を追加しました。また，CSCIWManagementServer へのリダイレクトについての説明を追加しました。

付録 L.5　02-20 の変更内容

- 条件に一致する作業の作業者割り当てと着手の機能を追加しました。
- ワーク管理システムの前提となるアプリケーションサーバのプログラムに uCosminexus Service Platform を追加しました。
- ワーク管理システムの前提となる uCosminexus 開発環境のプログラムに uCosminexus Service Architect を追加しました。
- CIWServer クラスの allocateWorkItemEx メソッド，および CIWWorkItem クラスの free メソッドを追加しました。
- セットアッププロパティファイルの設定内容に RangeSkipCleanup キーを追加しました。
- 次に示すテーブルの分類に，BPMN 連携機能を使用する場合だけ追加されるインデクスを追加しました。
  - <SYSTEMID>_PROCESS_INSTANCE テーブルの「ビジネスプロセス定義名」，および「状態」
  - <SYSTEMID>_WORK_ITEM テーブルの「案件 ID」，「作業定義 ID」，「状態」，および「作業者 ID」
- 次に示すメソッドを追加しました。
  - CIWProcessInstance クラス
    adhocInstantiateActivityInstance メソッド

- CIWActivityInstance クラス
  adhocMakeTransition メソッド

付録 L.6　02-10 の変更内容

- 組み込み作業の案件の実行時に子作業生成ルールが評価されるタイミングの説明を追加しました。
- アプリケーションサーバ（Cosminexus）上で CSCIW を使用するための設定手順から，保護区リストファイルの設定の作業を削除しました。
- 次のコマンドを追加しました。
  - ciwmngap
  - ciwmngapgrp
  - ciwtransbpmn
- SQL スクリプトファイルの編集に「02-00 から 02-10 へ移行する場合」を追加しました。

付録 L.7　02-00 の変更内容

- BPMN 連携機能を追加しました。
- アドホック API を追加しました。
- 次に示すワーク管理システムの前提となるアプリケーションサーバのプログラムを削除しました。
  - uCosminexus Application Server Enterprise
  - uCosminexus Application Server Standard
  - uCosminexus Service Platform
- 次に示すワーク管理システムの前提となる uCosminexus 開発環境のプログラムを削除しました。
  - uCosminexus Developer Professional
  - uCosminexus Service Architect
- Windows の場合にインストール先のフォルダに「&」を含むパスを指定できないことを追加しました。
- UNIX の場合の mount の実行方法の説明を変更しました。
- コマンド用環境設定ファイルに設定する JDBC のクラスパスについて，HiRDB および Oracle を使用している場合の指定例を変更しました。
- SQL スクリプトファイルの編集に「01-95 から 02-00 へ移行する場合」を追加しました。
- 完全フロー定義の説明を追加しました。

付録 L.8　01-95 の変更内容

- Windows の場合のインストール先フォルダを指定する手順に，インストール先フォルダに指定できないパスの記述を追加しました。また，デフォルトのインストール先の記述を削除しました。
- HNTRLib2 の 64 ビット版の場合は，hntrlib2j64.jar および hntrlibMj64.jar へのパスを追加する必要があることを記述しました。
- snapshot ログの収集についての説明を追加しました。
- SQL スクリプトファイルの編集に「01-90 から 01-95 へ移行する場合」を追加しました。

付録 L.9　01-90 の変更内容

- 「Hitachi Web Server」の記述を「Hitachi Web Server または Cosminexus HTTP Server」に変更しました。
- Cosminexus の 64 ビットコンポーネントを使用している場合に，監査ログを使用するときの注意事項を新規追加しました。
- ローテーション種別についての説明を追加しました。
- トレースファイルの自動削除機能についての説明を変更しました。
- セットアッププロパティファイルの設定内容に次を追加しました。
  - 次に示すキーの説明を追加しました。
    HiRDBTransientSQLState
    OracleTransientSQLState
    SQLServerTransientSQLState
    LogRotationMode
    RotationTime
  - 次に示すキーの説明にローテーション種別がシフトモードの場合の説明を追加しました。
    MsgFileNum
    TraceFileNum
    CmdMsgFileNum
    CmdTraceFileNum
    AdmMsgFileNum
    AdmTraceFileNum
  - 次に示すキーの説明にローテーション種別がラップアラウンドモードの場合の説明を追加しました。
    MsgTimeToDelete
    TraceTimeToDelete
    CmdMsgTimeToDelete
    CmdTraceTimeToDelete

AdmMsgTimeToDelete
AdmTraceTimeToDelete

- SQL スクリプトファイルの編集に「01-70 から 01-90 へ移行する場合」を追加しました。

付録 L.10　01-70 の変更内容

- Cosminexus V8 に対応しました。
- 条件およびルールに Java オブジェクト呼び出し機能を追加しました。
- 作業アプリケーションの呼び出し機能を追加しました。
- クラスタ構成を設定する場合の注意事項を追加しました。
- ciwchgdef（条件・ルール・作業アプリケーションの変更）コマンドを追加しました。
- SQL 文を付与するパラメタに次のパラメタを追加しました。
  案件投入者，案件開始日，案件処理期限，案件優先度，業務ステップ開始日，業務ステップ処理期限，業務ステップ優先度，作業生成日，作業開始日，作業処理期限，作業優先度，ユーザ ID
- テーブル一覧に次の分類を追加しました。
  条件 Java 定義，生成ルール Java 定義，作業アプリケーション Java 定義，振り分けルール Java 定義
- インデクス一覧に次のテーブル名を追加しました。
  <SYSTEM>_APPLICATION_DEF，<SYSTEM>_APPLICATION_JMS_DEF，<SYSTEM>_APPLICATION_JAVA_DEF，<SYSTEMID>_CASTINGRULE_JAVA_DEF，<SYSTEMID>_CONDITION_JAVA_DEF，<SYSTEMID>_CREATIONRULE_JAVA_DEF
- CSCIW 01-50 または 01-60 から CSCIW 01-70 へのワーク管理データベースの移行手順を追加しました。
- ワーク管理データベースの移行手順に SQLServer を使用している場合を追加しました。
- 環境設定ファイルに次のファイルを追加した。
  ビジネスプロセス管理用環境設定ファイル，EJB クライアント用ユーザプロパティファイル，クラス名設定プロパティファイル

付録 L.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。
- 案件の引き戻しと差し戻しの説明を追加しました。
- 使用できる DBMS として Microsoft(R) SQL Server を追加しました。
- ワーク管理システムの前提となるアプリケーションサーバのプログラムに，uCosminexus Application Server Standard を追加しました。

- 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の 64bit 版に関する注意事項を追加しました。
- 業務プログラムの実行権限に関する注意事項を追加しました。
- 監査ログの取得によるシステム監視についての説明を追加しました。
- トレースファイルに出力されるメソッドに，makeBackwardTransition メソッドを追加しました。
- メッセージの確認で，ビジネスプロセス管理（CSCIW-Definer）で出力されるメッセージ ID を追加しました。
- セットアッププロパティファイルの値に「¥」を記述する場合と，値にマルチバイトコードが含まれる場合の説明を追加しました。
- トレースファイルに出力されるメソッド complete，および performAndComplete のトレース量の概算（単位：KB）の値を変更しました。
- 監査ログのメッセージの出力個所を追加しました。
- CSCIW 01-50 から CSCIW 01-60 への移行方法を追加しました。

付録 M　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 M.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 AP 開発ガイド（3020-3-M81）
- 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 運用管理ポータル操作ガイド（3021-3-J14）
- 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 アプリケーションサーバ Web サービス開発ガイド（3021-3-J23）
- Cosminexus V11 アプリケーションサーバ メッセージ（構築／運用／開発用）（3021-3-J27）
- 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。

(4) JP1 関連

- JP1 Version 12 JP1/Base 運用ガイド（3021-3-D65）

なお，このマニュアルでは，JP1 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 M.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

表記	製品名
Cosminexus	uCosminexus Application Server
CSCIW	uCosminexus Service Coordinator Interactive Workflow
Firefox	Mozilla Firefox
HiRDB	HiRDB Server Version 10
Linux※	Red Hat(R) Enterprise Linux(R) Server 7.1（64-bit x86_64）以降
	Red Hat(R) Enterprise Linux(R) Server 8.1（64-bit x86_64）以降
ORACLE	Oracle(R) Database 19*c*
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 M.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

表記		製品名
Windows	Windows 11	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 M.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
BPMN	Business Process Modeling and Notation
EJB	Enterprise JavaBeans
J2EE	Java 2 Platform, Enterprise Edition，および J2EE
JAR	Java ARchive
JavaVM	Java Virtual Machine
JAX-RS	Java API for RESTful Web Service
JDBC	Java Database Connectivity，および JDBC
JDK	Java Development Kit，および JDK
LOB	Large Object Block
REST	Representational State Transfer
WAR	Web ARchive

付録 M.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。

付録 N　用語解説

CSCIW のマニュアルで使用する用語とその解説を次に示します。

(英字)

BPMN 連携機能

CSCIW で BPMN2.0 をサポートするための機能です。BPMN 連携機能を使用すると，BPMN2.0 の仕様に従ったビジネスプロセスを，CSCIW のビジネスプロセスに変換して CSCIW で使用できます。

CSCIW-Definer

ビジネスプロセスを定義，管理および運用するビジネスプロセス管理ツールのことです。CSCIW-Definer 用のプラグインを Eclipse に組み込むことでビジネスプロセス管理ツールとして使用できます。

CSCIWManagementServer

アプリケーションサーバでの CSCIW の初期化・終了処理を提供している J2EE アプリケーションです。

Eclipse

Eclipse Foundation が提供するオープンソースの統合開発環境です。Eclipse は，CSCIW-Definer の前提プログラムです。CSCIW-Definer 用のプラグインを Eclipse に組み込むことでビジネスプロセス管理ツールとして使用できます。

Java オブジェクト呼び出し

CSCIW が提供する Java インタフェースを使用した Java オブジェクトを，同期を取って実行する機能です。条件，生成ルール，振り分けルール，および作業アプリケーションの種別が指定できます。

ref 識別子

アプリケーション呼び出しサービスによる呼び出しの定義を識別する名称です。BPMN 連携機能を使用しない場合，ref 識別子として次の種類があります。

- 作業アプリケーション定義（種別が REST）の属性

REST アプリケーション

ビジネスプロセスの内容に従って，アプリケーション呼び出しサービスから REST 形式で呼び出される業務アプリケーションのことです。

WorkManager の最大スレッド数

WorkManager の非デーモン（短寿命）Work スレッドプールの最大スレッド数です。

（ア行）

アドホック API

フロー遷移定義の存在しない遷移や，フロー遷移定義と関係ない業務ステップの生成を実行する API です。アドホック API を使用する設定のビジネスプロセス定義だけで使用できます。

アプリケーション呼び出しサービス

ビジネスプロセス定義で定義された REST アプリケーションの呼び出しなどを行うサービスのことです。

アロー

業務ステップおよびノード間の処理の流れを規定するためのものです。アローは，遷移元や遷移先となる業務ステップの情報を持ちます。

案件

ビジネスプロセスの開始から終了までを一つの単位とする，業務の実例（ビジネスプロセスインスタンス）のことです。

案件 ID

同一ビジネスプロセスの案件を識別するための識別子です。ワーク管理および CSCIW が付与します。

案件運用操作

案件の運用状況を監視したり，必要に応じて案件，業務ステップ，および作業の状態を操作したりする J2EE アプリケーションです。

案件開始

案件の開始処理のことです。

案件名

同一ビジネスプロセスの案件を識別するための識別子です。ユーザが設定できます。案件キーともいいます。

一時停止

案件単位または業務ステップ単位で，作業の状態を一時的に停止することです。

（カ行）

開始日時

開始日時は，案件，業務ステップ，および作業の属性です。案件，業務ステップ，および作業が，「running（実行中）」状態へ遷移したときの日時を指します（作業の場合，「running（実行中）」状態に含まれる「performing（作業者実行）」状態，または「executing（自動実行）」状態に遷移したときの日時を指します）。

CSCIW の API を利用することで「running（実行中）」状態へ遷移できます。

環境設定ファイル

CSCIW を動作させるために必要な情報を設定するためのプロパティ形式のファイルのことです。CSCIW の環境設定ファイルは次の 5 種類です。

- システム設定プロパティファイル
- セットアッププロパティファイル
- コマンド用環境設定ファイル
- ビジネスプロセス管理用環境設定ファイル
- クラス名設定プロパティファイル

完全フロー定義

次の条件を満たしたビジネスプロセス定義のことです。

- すべての業務ステップ定義，階層定義に遷移元と遷移先が定義されている
- すべての業務ステップ定義内に作業が定義されている
- すべての業務ステップ定義，階層定義，制御ノードがソースノードからシンクノードまでの経路に含まれている

完了条件

作業または業務ステップを「完了」状態にする条件です。作業または業務ステップの評価時（作業の発生時を含む）にこの条件を満足していた場合，作業または業務ステップを「完了」状態にします。

キャンセル

案件の実行を取り消すことです。

業務ステップ

ビジネスプロセス定義の構成要素の一つです。業務ステップは，業務の状態を表し，ある状態での作業の集合として定義されます。

業務データ

業務ステップの完了条件や制御ノードの分岐条件として参照されるデータです。業務データは，ワーク管理データベースの外部の業務データベースで管理，運用する必要があります。

業務データベース

業務データが格納されているデータベースのことです。

業務プログラム

CSCIW で作業を実行するために利用するプログラムのことです。CSCIW は，業務プログラムを作成するのに有効なツールとしての API を提供しています。

組み込み作業

CSCIW の基本的な機能を組み合わせることで実現するような複雑な処理を 1 つの作業としてまとめたものです。CSCIW の組み込み作業には並列作業があります。

コマンド

CSCIW の環境構築や運用をするための機能で，シェルやコマンドプロンプトから実行します。

（サ行）

作業

ビジネスプロセス定義の構成要素の一つです。作業は，ある業務ステップで実行する具体的な処理です。作業では，処理を実行する作業者の情報を振り分けルール名として持ちます。

作業一覧

作業の内容を一覧で表示したものです。

作業者

作業の処理を担当するユーザまたはユーザグループのことです。作業者は，ワーク管理データベースの外部の業務データベースで管理，運用する必要があります。

作業者データ

作業を処理するユーザに関するデータのことです。

作業者データベース

作業者に関する情報を格納したデータベースのことです。

差し戻し

遷移元の実行中の業務ステップを強制終了して，すでに終了している遷移先の業務ステップを追加生成して実行中状態にすることです。差し戻し処理では，遷移元（差し戻し元）となる現

在実行中の業務ステップが，動作を要求します。ユーザまたは業務プログラムは，すでに完了した業務ステップを検索して遷移先（差し戻し先）を決定します。

システム共通環境情報

同一のシステム ID によって CSCIW を運用するシステムに共通に適用する情報です。マルチマシンを構成する複数の異なるマシンにも共通して適用されます。複数の異なるマシンに共通して適用するために，情報はワーク管理データベースに格納しています。

事前生成

業務ステップの生成方法の一つです。ビジネスプロセス定義で事前生成することを指定した業務ステップは，案件の開始要求を受け付けたときに生成されます。

終了日時

終了日時は，案件，業務ステップ，および作業の属性です。案件，業務ステップ，および作業が，「closed（終了）」状態へ遷移したときの日時を指します。CSCIW の API を利用することで「closed（終了）」状態へ遷移できます。

処理期限

案件に対して処理期限を日数単位で指定できます。

シンクノード

案件の終了を意味します。推進された案件は，シンクノードに状態が遷移することで完了します。

制御ノード

ビジネスプロセス定義の構成要素の一つです。制御ノードは，案件の遷移を制御するノードです。制御ノードには，分岐ノード，分業ノード，先着ノード，および待合ノードがあります。

先着ノード

前の業務ステップとして複数の業務ステップが存在しており，どれか最初の業務ステップからの遷移時に次の業務ステップに遷移します。

ソースノード

案件の開始を意味します。案件が開始されると，ソースノードから次の業務ステップまたは制御ノードに状態が遷移します。

（タ行）

データ条件

分岐ノードでの案件の流れを規定したり，業務ステップや作業の状態を制御したりするための条件のことです。

（ハ行）

発生条件

作業を生成する条件です。業務ステップの評価時にこの条件を満足していた場合，作業を生成します。

発生日時

発生日時は作業の属性です。作業が，「ready（実行開始可能）」状態へ遷移したときの日時を指します。通常，作業の「ready（実行開始可能）」状態への遷移は，案件投入時，または前の業務ステップの完了時に自動的に実行されます。

引き戻し

遷移元の実行中の業務ステップを強制終了して，すでに終了している遷移先の業務ステップを追加生成して実行中状態にすることです。引き戻し処理では，遷移先（引き戻し先）となる，すでに完了した業務ステップが動作を要求します。ユーザまたは業務プログラムは，現在実行中の業務ステップを検索して遷移元（引き戻し元）を決定します。

ビジネスプロセス

業務を実行するための作業の流れです。

ビジネスプロセス管理

業務の流れをワーク管理システムで使用できるように，ビジネスプロセスを定義したり，それらの運用状態を管理したりすることです。CSCIW では，ビジネスプロセス管理に CSCIW-Definer を使用します。

ビジネスプロセス管理ツール

業務の流れをワーク管理システムで使用できるように，ビジネスプロセスを定義したり，それらの運用状態を操作したりするツールのことです。

ビジネスプロセス定義

業務の流れ（ビジネスプロセス）をビジネスプロセス管理ツール（CSCIW-Definer）で定義したものです。

ビジネスプロセス定義機能

ビジネスプロセス管理ツール（CSCIW-Definer）で定義できる機能の一つです。次の操作ができます。

- ビジネスプロセスの業務ステップ，作業，制御ノード，および遷移を必要に応じて定義します。
- ビジネスプロセス定義の各作業に振り分けルール定義を設定して，業務が運用されたときに各作業の作業者が特定されるようにします。

振り分けルール定義

作業の作業者を決定するためのルールです。ビジネスプロセス定義で各作業に指定した振り分けルール定義を適用することで，作業者を決定します。

振り分けルール定義機能

ビジネスプロセス管理ツール（CSCIW-Definer）で定義できる機能の一つです。

CSCIW では，ビジネスプロセスの作業者の情報を外部のデータベースで管理しています。このため，そのデータベースからどのような条件で作業者を検索し，決定すればいいのかを振り分けルール定義として定義します。

分岐ノード

制御ノードの一つです。次の業務ステップとしてあらかじめ定義された複数の業務ステップから，条件に従って 1 つの業務ステップを選択し開始します。

分業ノード

制御ノードの一つです。次の業務ステップとして，あらかじめ定義された複数の業務ステップをすべて開始します。

並列作業

組み込み作業の一つで，並列作業をビジネスプロセスに定義しておくことで，案件の実行時に複数の作業が自動的に生成されます。生成される作業数は，案件の実行時に決まります。

（マ行）

待合ノード

制御ノードの一つです。直前の業務ステップとしてあらかじめ定義された複数の業務ステップのうち，すべてが完了した時点で次の業務ステップを開始します。

マルチインスタンス

1 台のマシンで，複数ディレクトリに実行環境を構築することで，複数の異なるシステム ID で CSCIW を運用するシステム構成のことです。この場合の，CSCIW の実行環境は 1 つです。

マルチプロセス

1 台のマシンで，同一のシステム ID の設定となっている CSCIW を使用したプロセス（J2EE サーバおよび Java アプリケーション）を複数運用するシステム構成のことです。

マルチマシン

複数の異なるマシン上で，同一のシステム ID によって CSCIW を運用するシステム構成のことです。

（ヤ行）

ユーザ

ワーク管理の利用者のことです。

ユーザグループ

ユーザが所属するグループのことです。ユーザの集合がユーザグループとなります。

優先度

案件，業務ステップ，および作業に設定できる優先度を示す整数値です。案件，業務ステップ，および作業の属性なので，検索の条件として利用できます。優先度の値を基に案件の流れは制御されません。入力指定範囲は，-2,147,483,648～2,147,483,647 です。

（ラ行）

リソースマネージャ

分散処理システムでの，資源を管理する機能の総称です。DBMS もリソースマネージャに含まれます。

（ワ行）

ワーク管理

CSCIW の中でビジネスプロセス定義に従い遷移するワークフローのエンジンのことです。

ワーク管理システム

業務の進行を「完成に至るまでの業務の状態変化」としてとらえ，状態変化を統括的に管理することで業務の進行状況を管理する，ワークフロー型汎用アプリケーションの構築基盤です。

CSCIW で構築したワークフローシステムを指します。

ワーク管理データベース

CSCIW が扱う情報を管理するデータベースです。

索引

記号

<SYSTEMID>_ACTIVITY_DEF の内容　284
<SYSTEMID>_ACTIVITY_INSTANCE の内容　302
<SYSTEMID>_APPLICATION_DEF の内容　294
<SYSTEMID>_APPLICATION_JAVA_DEF の内容　296
<SYSTEMID>_APPLICATION_JMS_DEF の内容　295
<SYSTEMID>_APPLICATION_LOCK_GROUP の内容　309
<SYSTEMID>_APPLICATION_LOCK_INFO の内容　308
<SYSTEMID>_ASYNC_JOB の内容　310
<SYSTEMID>_CASTINGRULE_DEF の内容　298
<SYSTEMID>_CASTINGRULE_JAVA_DEF の内容　300
<SYSTEMID>_CASTINGRULE_SQL_DEF の内容　299
<SYSTEMID>_CONDITION_DEF の内容　289
<SYSTEMID>_CONDITION_JAVA_DEF の内容　290
<SYSTEMID>_CONDITION_SQL_DEF の内容　290
<SYSTEMID>_CONTROLNODE_DEF の内容　286
<SYSTEMID>_CONTROLNODE_INSTANCEMNG の内容　305
<SYSTEMID>_CONTROLNODE_INSTANCE の内容　304
<SYSTEMID>_CREATIONRULE_DEF の内容　291
<SYSTEMID>_CREATIONRULE_JAVA_DEF の内容　292
<SYSTEMID>_CREATIONRULE_SQL_DEF の内容　292
<SYSTEMID>_FLOW_DEF の内容　287
<SYSTEMID>_FLOW_POLYGONALLINE_INFO の内容　297
<SYSTEMID>_HIERARCHY_DEF の内容　296
<SYSTEMID>_ID_RANGE_RESERVE_LOG の内容　308
<SYSTEMID>_PARALLEL_WORK_DEF の内容　288
<SYSTEMID>_PROCESS_DEF の内容　283
<SYSTEMID>_PROCESS_INSTANCE の内容　300
<SYSTEMID>_SERVER_PROC_INFO の内容　306
<SYSTEMID>_TIMERULE_DATETIME_DEF の内容　294
<SYSTEMID>_TIMERULE_DEF の内容　293
<SYSTEMID>_UNUSED_ID_RANGE の内容　307
<SYSTEMID>_USER_SETTING_INFO の内容　306
<SYSTEMID>_WORK_DEF の内容　285
<SYSTEMID>_WORK_ITEM の内容　303

数字

1 行に出力されるメッセージのバイト数　220

A

API　59

B

BPMN 連携機能を使用しない場合のアプリケーション呼び出しサービス　360
BPMN 連携機能〔用語解説〕　460

C

ciwdelpi コマンド実行時の排他要求数の見積もり(HiRDB 限定)　322
Cosminexus 性能解析トレースの取得　221
Cosminexus の起動　118
Cosminexus の設定　254, 259
CSCIW　20
CSCIWManagementServer に関係する監査ログ　325
CSCIWManagementServer のインポート　123
CSCIWManagementServer の開始　124
CSCIWManagementServer の開始時期　208
CSCIWManagementServer の監査事象　195
CSCIWManagementServer の再起動が必要なコマンド　168
CSCIWManagementServer の設定　122

CSCIWManagementServer のプロパティ設定 124
CSCIWManagementServer〔用語解説〕 460
CSCIW　案件運用操作 26
CSCIW が出力する情報 214
CSCIW が性能解析トレースで出力するイベント ID およびメソッドの概要 222
CSCIW のアンインストール 156
CSCIW のインストール 95
CSCIW のインストール（上書きインストール） 246
CSCIW のインストール（新規インストール） 256
CSCIW の出力情報の取得 214
CSCIW の障害時の回復方法 236
CSCIW の性能解析トレースの取得ポイント 221
CSCIW の前提プログラム 91
CSCIW の停止および削除 244
CSCIW を使用している Java アプリケーションの再起動が必要なコマンド 168
CSCIW をマルチインスタンス構成にする場合のセットアッププロパティファイルの設定 112
CSCIW-Definer 26
CSCIW-Definer〔用語解説〕 460

D

DB Connector の設定 119
DBMS の格納領域の変更（HiRDB の場合） 161
DBMS の格納領域の変更（ORACLE の場合） 163
DB サーバの前提プログラム 92

E

Eclipse〔用語解説〕 460

I

IP アドレスの変更 160

J

J2EE サーバの停止 146
J2EE サーバを V9 互換モードで使用する
　互換モード用のファイル 444
　コンテナ拡張ライブラリに取り込むライブラリ 444
J2EE サーバを V9 互換モードで使用する場合の設定 444
Java アプリケーションの設定 131
Java オブジェクト呼び出し機能を使用した，Java オブジェクトの J2EE アプリケーションへの組み込みについて 211
Java オブジェクト呼び出し〔用語解説〕 460
Java オブジェクト呼び出しを利用する場合の設定 132
JP1 によるログファイルの監視 239

O

OS からのアンインストール 156

R

RDB 検索 76
RDB 検索による生成ルール定義 76
RDB 検索によるデータ条件の評価 76
RDB 検索による振り分けルール定義 76
ref 識別子〔用語解説〕 460
REST API による運用管理の操作 404
REST アプリケーション〔用語解説〕 460

S

SQL インジェクション 210
SQL 文の排他要求数を算出する場合に必要な値 323

U

Unicode の補助文字 220
UNIX からのアンインストール 157

W

Windows からのアンインストール 156
WorkCoordinator Definer で定義したビジネスプロセス定義の機能の登録可否 357
WorkCoordinator Definer で定義した振り分けルール定義の機能の登録可否 359
WorkManager の最大スレッド数〔用語解説〕 461

X

XML スキーマファイル 420

あ

アドホック API　461
アプリケーションサーバの設定　113
アプリケーション呼び出しサービスのインポート　129
アプリケーション呼び出しサービスの設定　129
アプリケーション呼び出しサービスのプロパティ設定　130
アプリケーション呼び出しサービス〔用語解説〕　461
アロー　37
アロー〔用語解説〕　461
案件　32
案件 ID〔用語解説〕　461
案件運用操作　26
案件運用操作に関係する監査ログ　345
案件運用操作のインポート　127
案件運用操作の開始　128
案件運用操作の監査事象　197
案件運用操作の設定　125
案件運用操作のプロパティ設定　128
案件運用操作〔用語解説〕　461
案件開始〔用語解説〕　461
案件投入可能期間　182
案件に関する運用　188
案件に対する運用　188
案件の運用状況の確認　233
案件の運用・管理を支援する機能　26
案件の開始　50, 188
案件の基本的な進み方　49
案件の強制終了　189
案件の再開　189
案件の削除　189
案件の終了　59
案件の処理期限の設定　190
案件の遷移依頼　65
案件の中断　189
案件の動的な制御　59
案件の優先度の更新　190
案件の優先度の設定　190
案件名〔用語解説〕　461
案件〔用語解説〕　461
案件を中断した場合の業務ステップの状態遷移　189

い

移行元環境の CSCIW の停止　256
一時停止〔用語解説〕　461
インスタンステーブルのレコード数　313
インストール　91
インストール時に作成するディレクトリ（フォルダ）　446
インストールディレクトリ（フォルダ）　446
インデクス一覧　310

う

運用管理 REST API　404
運用管理 REST サービス　404
運用管理 REST サービスのインポート　410
運用管理 REST サービスの開始　411
運用管理 REST サービスの設定　410
運用管理 REST サービスのプロパティ設定　410
運用上の注意事項　208
運用中のバックアップ　202

え

エラーメッセージの確認　232

か

改行文字の扱い　220
開始日時〔用語解説〕　462
開発環境での環境削除　148
開発環境での環境設定　107
各機能の概要　26
各種トレースファイルの切り替えの流れ　218
格納領域の変更（SQL Server の場合）　165
活性　182, 185
環境設定ファイル　263
環境設定ファイル〔用語解説〕　462
環境変数の削除〔Cosminexus Management Server で設定した環境変数〕　146
環境変数の削除〔CSCIW の環境変数〕　147

環境変数の設定　110
環境変数の取り込み　117
環境変数の変更　169
監査対象が CSCIWManagementServer の場合の監査事象　195
監査対象が案件運用操作の場合の監査事象　197
監査対象がコマンドの場合の監査事象　196
監査ログの出力ポイントと出力項目　197
監査ログの取得によるシステム監視　195
監査ログの詳細　325
完全フロー定義　462
完了条件〔用語解説〕　462
完了通知　193

き

キャッシュする ID 範囲の見積もり　321
キャンセル〔用語解説〕　462
強制終了　189, 192, 193
業務アプリケーションが異常終了した場合や強制終了した（終了処理をしなかった）場合の影響　209
業務システムの変更に伴う運用　204
業務状態　35
業務ステップ　35
業務ステップが完了するタイミング　52
業務ステップに対する運用　190
業務ステップの活性化　191
業務ステップの完了　47, 191
業務ステップの完了条件が評価されるタイミング　52
業務ステップの強制終了　192
業務ステップの再開　191
業務ステップの状態遷移　50
業務ステップの処理期限の設定　192
業務ステップの中断　191
業務ステップの評価要求　63
業務ステップの優先度の更新　192
業務ステップ〔用語解説〕　462
業務ステップを中断した場合の作業の状態遷移　191
業務データベース　46
業務データベース〔用語解説〕　463
業務データ〔用語解説〕　463
業務の開始　179
業務の停止　180
業務プログラムの実行権限を設定する場合の注意事項　138
業務プログラム〔用語解説〕　463

く

組み込み作業　68
組み込み作業での案件の進み方　68
組み込み作業〔用語解説〕　463
組み込み変数　76
クラスタ構成　87
クラスタ構成を設定する場合の注意事項　138

こ

後続停止　56
子作業　68
子作業生成ルール　68
子作業生成ルールが評価されるタイミング　69
コマンドに関係する監査ログ　331
コマンドによる実行環境の削除　147
コマンドによる実行環境の初期化　113
コマンドの監査事象　196
コマンド用環境設定ファイル　278
コマンド用環境設定ファイルの設定内容　279
コマンド用環境設定ファイルの編集　112
コマンド〔用語解説〕　463
コンテナ拡張ライブラリからの削除　146
コンテナ拡張ライブラリへの設定追加　410
コンテナ拡張ライブラリへの取り込み　114

さ

作業　35
作業アプリケーションの呼び出し　71
作業一覧〔用語解説〕　463
作業が完了するタイミング　54
作業者 ID　45
作業者データベース　41

作業者データベース〔用語解説〕 463
作業者データ〔用語解説〕 463
作業者の再割り当て 205
作業者〔用語解説〕 463
作業者を決定する仕組み 41
作業者を変更する 204
作業に対する運用 192
作業の完了 48
作業の完了条件が評価されるタイミング 54
作業の完了通知 193
作業の強制終了 193
作業の再開 193
作業の準備 193
作業の状態遷移 52
作業の処理期限の設定 194
作業の着手 193
作業の中断 193
作業の発生 47
作業の発生条件が評価されるタイミング 54
作業の評価要求 64
作業の優先度の更新 194
作業〔用語解説〕 463
差し戻し 60
差し戻し〔用語解説〕 463

し

システム共通環境情報〔用語解説〕 464
システム構成モデル 84
システム削除の流れ 141
システム設定の流れ 90
システム設定プロパティファイル 263
システム設定プロパティファイルの設定内容 264
システム設定プロパティファイルの変更 169
システムの構成変更 160
事前生成 50
事前生成〔用語解説〕 464
実行開始可能 193
実行環境での環境削除 142
実行環境での環境設定 108
実行環境の構築 110
実行環境の削除 146
実行環境の設定情報の変更 168
実行環境マシンの IP アドレスの変更 160
終了日時〔用語解説〕 464
出力情報の設定 217
出力する情報およびファイル名 214
出力ファイルのローテーションの設定 216
出力ファイル名 215
出力レベルおよび出力内容 218
巡回部分 61
障害情報の取得 214
障害の回復 236
障害発生時の対処の流れ 213
障害への対処 235
障害要因の特定 232
条件に一致する作業の作業者割り当てと着手 66
状態 182, 185
状態遷移型モデル 22
ジョブの投入 428
ジョブレコード 406
ジョブレコードの一覧取得 436
ジョブレコードの取得 433
処理期限〔用語解説〕 464
シンクノード 59
シンクノード〔用語解説〕 464

せ

制御ノード 36, 55
制御ノードでの案件遷移 55
制御ノード〔用語解説〕 464
生成ルール記述形式 77
性能解析トレースとして出力する情報 221
性能解析トレースの取得ポイント 221
セキュリティロールの設定 122, 126
接続先 DBMS の変更 160
設定 98
セットアップコマンドによるバージョンアップ（既存のワーク管理システムを引き継ぐ場合） 253

セットアップコマンドによるバージョンアップ（ワーク管理システムを新規に構築する場合） 259
セットアッププロパティファイル 264
セットアッププロパティファイルの設定内容 265
セットアッププロパティファイルの変更 169
セットアッププロパティファイルの編集 111
遷移可 192
先着ノード 36
先着ノードでの案件遷移 56
先着ノード〔用語解説〕 464
前提プログラムのインストール 91

そ

ソースノード 50
ソースノード〔用語解説〕 464
そのほかのテーブルのレコード数 316

ち

中断 191, 193

て

定義テーブルのレコード数 314
データ移行だけを繰り返し行う手順 260
データ条件 38, 46
データ条件記述 48
データ条件記述形式 76
データ条件種別 48
データ条件と業務データ 46
データ条件による案件制御の例 46
データ条件による案件遷移の制御 46
データ条件の構成 48
データ条件評価の RDB 検索に関連するデータを変更する 207
データ条件〔用語解説〕 464
データ条件を変更する 205
データベースコネクション数の見積もり 108
データベースコネクションプール数の見積もり 108
データベース使用時のセキュリティ 210
データベースの管理 210
データベースの再編成 203
データベースへのアクセス権限の削除（HiRDB の場合） 150
データベースへのアクセス権限の削除（ORACLE の場合） 152
データベースへのアクセス権限の削除（SQL Server の場合） 154
データベースへのアクセス権限の付与（HiRDB の場合） 100
データベースへのアクセス権限の付与（ORACLE の場合） 103
データベースへのアクセス権限の付与（SQL Server の場合） 105
テーブル一覧 281
テーブル容量の見積もり 281
適用可能期間 185
デフォルトトレースレベルで出力されるトレース量の概算 318

と

動的 59
トラブルシューティング 237
トレースファイル取得時の注意事項 219
トレースファイルの自動削除機能 219
トレース容量の見積もり 318

は

バージョン 182, 185
バージョンアップの流れ 241
バックアップする内容 200
バックアップとリストア 200
バックアップの流れ 201
発生条件〔用語解説〕 465
発生日時〔用語解説〕 465

ひ

非活性 182, 185
引き戻し 59
引き戻しおよび差し戻しに関する注意事項 61
引き戻しと差し戻し 59
引き戻し〔用語解説〕 465

ビジネスプロセス　20, 30
ビジネスプロセス管理ツール（CSCIW-Definer）　26
ビジネスプロセス管理ツール〔用語解説〕　465
ビジネスプロセス管理〔用語解説〕　465
ビジネスプロセス定義　32
ビジネスプロセス定義開発環境の前提プログラム　94
ビジネスプロセス定義機能〔用語解説〕　465
ビジネスプロセス定義削除時の注意事項　177
ビジネスプロセス定義と案件の流れ　32
ビジネスプロセス定義に関する運用　182
ビジネスプロセス定義の案件投入可能期間　182
ビジネスプロセス定義の管理　40
ビジネスプロセス定義の構成　34
ビジネスプロセス定義の削除　39, 184, 424
ビジネスプロセス定義の作成　39
ビジネスプロセス定義の状態　182
ビジネスプロセス定義の登録　39, 183, 421
ビジネスプロセス定義のバージョン　40, 182
ビジネスプロセス定義の変更　176
ビジネスプロセス定義ファイル　357
ビジネスプロセス定義〔用語解説〕　465
ビジネスプロセス〔用語解説〕　465
ビジネスプロセスを変更する　207
日付・時刻の整合性　208
必要ファイルの退避　156

ふ

複数バージョンを持つ振り分けルール定義の特定　185
振り分けルール　41
振り分けルール記述　42
振り分けルール記述形式　76
振り分けルール種別　42
振り分けルール定義　41
振り分けルール定義機能〔用語解説〕　466
振り分けルール定義に関する運用　185
振り分けルール定義による作業者の特定　41
振り分けルール定義の管理　43, 185
振り分けルール定義の構成　42
振り分けルール定義の削除　43, 187, 427
振り分けルール定義の作成　43
振り分けルール定義の状態　185
振り分けルール定義の登録　43, 186, 425
振り分けルール定義のバージョン指定　185
振り分けルール定義の変更　176, 204
振り分けルール定義の有効期間　185
振り分けルール定義評価の RDB 検索に関連するデータを変更する　207
振り分けルール定義ファイル　357
振り分けルール定義〔用語解説〕　466
振り分けルール名　37
分岐　46
分岐ノード　36
分岐ノードでの案件遷移　55
分岐ノード〔用語解説〕　466
分業ノード　36
分業ノードでの案件遷移　55
分業ノード〔用語解説〕　466

へ

並列作業　36, 68
並列作業〔用語解説〕　466
変更後の接続先情報の登録　160
変更前の接続先情報の削除　160

ま

前処理の排他要求数を算出する場合に必要な値　322
待合ノード　36
待合ノードでの案件遷移　58
待合ノード〔用語解説〕　466
マルチインスタンス構成　85
マルチインスタンス構成の環境を削除する場合の注意事項　158
マルチインスタンス構成を設定する場合の注意事項　136
マルチインスタンス〔用語解説〕　466
マルチプロセス構成　86
マルチプロセス構成を設定する場合の注意事項　137
マルチプロセス〔用語解説〕　466
マルチマシン構成　86

マルチマシン構成の環境を削除する場合の注意事項 158
マルチマシン構成を設定する場合の注意事項 136
マルチマシン〔用語解説〕 466

め

メッセージ ID の概要 232

ゆ

有効期間 185
ユーザ 44
ユーザおよびユーザグループに関する情報の変更 204
ユーザグループ 44
ユーザグループ識別子 45
ユーザグループ〔用語解説〕 467
ユーザとユーザグループの扱い 45
ユーザとユーザグループの関係 44
ユーザの操作内容の確認 233
ユーザ〔用語解説〕 467
優先度〔用語解説〕 467

り

リストアの流れ 201
リソースマネージャ〔用語解説〕 467

れ

レコード数の概算式 313

わ

ワーク管理システム 20
ワーク管理システムによる業務システム開発の流れ 27
ワーク管理システムによる業務の流れ 30
ワーク管理システムの運用状況の確認 233
ワーク管理システムの監査対象 195
ワーク管理システムの関連プログラム 82
ワーク管理システムの基本的なシステム構成モデル 84
ワーク管理システムの前提プログラム 81
ワーク管理システムの適用業務 23
ワーク管理システムの適用方法 23
ワーク管理システムの特長 21
ワーク管理システムのバージョンアップ（既存のワーク管理システムを引き継ぐ場合） 244
ワーク管理システムのバージョンアップ（ワーク管理システムを新規に構築する場合） 255
ワーク管理システムのプログラム構成 81
ワーク管理システムの目的 20
ワーク管理システム〔用語解説〕 467
ワーク管理データベースの環境削除（HiRDB の場合） 149
ワーク管理データベースの環境削除（ORACLE の場合） 151
ワーク管理データベースの環境削除（SQL Server の場合） 153
ワーク管理データベースの環境設定（HiRDB の場合） 99
ワーク管理データベースの環境設定（ORACLE の場合） 101
ワーク管理データベースの環境設定（SQL Server の場合） 104
ワーク管理データベースの構築（ワーク管理システムの設定） 98
ワーク管理データベースの構築（ワーク管理システムのバージョンアップ） 257
ワーク管理データベースの削除 149
ワーク管理データベースの削除（HiRDB の場合） 149
ワーク管理データベースの削除（ORACLE の場合） 151
ワーク管理データベースの削除（SQL Server の場合） 153
ワーク管理データベースの作成（HiRDB の場合） 99
ワーク管理データベースの作成（ORACLE の場合） 101
ワーク管理データベースの作成（SQL Server の場合） 104
ワーク管理データベースの障害時の回復方法 236
ワーク管理データベースのデータ移行 257
ワーク管理データベースのバージョンアップ（既存のワク管理システムを引き継ぐ場合） 246
ワーク管理データベースのバージョンアップ（ワーク管理システムを新規に構築する場合） 257
ワーク管理データベースへの接続情報の決定 98
ワーク管理データベース〔用語解説〕 467

ワーク管理データベース容量の算出　98
ワーク管理〔用語解説〕　467

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