



Synergy Distributed ガイド



IBM Rational Synergy

Distributed

リリース 7.1a

本書をご使用になる前に、171 ページの「特記事項」に記載されている情報をお読みください。

本書は、IBM Rational Synergy（製品番号 5724V66）バージョン 7.1a および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

**©Copyright IBM Corporation 1992, 2009**

---

# 目次

はじめに	1
本書の構成	6
本書で使用する参照資料 6	
ガイドで使用する表記規則	7
<b>DCM の概要</b>	<b>9</b>
はじめに 9	
DCM とは	10
機能	11
用語解説	12
オブジェクト命名規則	17
ロール	19
<b>DCM クラスタの計画</b>	<b>21</b>
はじめに 21	
DCM 手法と複製トポロジの作成	22
マスタとサテライト手法 24	
公開と登録による転送手法 25	
ピア ツー ピアの転送手法 26	
必要ディスク領域の確認	27
ソース データベース 27	
デスティネーション データベース 28	
DCM 用ディスク領域確保計画ワークシート	29
共通データベース プロパティの規定	35
共通データベース パラメータの規定	36
エンジンとサーバーの同期	38
<b>DCM データベースの設定</b>	<b>39</b>
はじめに 39	
手法	40
データベースの初期化	41
プロセスの見直しと選択	42
DCM データベース定義の作成	43
転送セットの作成	44

---

転送セットへのオブジェクトの追加.....	45
<b>DCM の使用</b>	<b>49</b>
はじめに 49	
手法.....	50
生成操作について.....	51
転送パッケージの生成.....	55
受取り操作について.....	57
転送パッケージの手作業受取り.....	60
<b>DCM データベースの管理</b>	<b>61</b>
はじめに 61	
他の DCM アドミニストレータとの通信.....	62
アップグレード後の DCM 設定の変更.....	63
データベース ID の変更.....	64
オブジェクトの更新を伴わない ID の変更 64	
データベース ID の変更と影響を受けたオブジェクトの更新 64	
別のデータベースから受取ったオブジェクトのデータベース ID の変換 66	
DCM 区切り文字の変更.....	67
DCM データベース定義の編集.....	68
DCM データベース定義の削除.....	69
転送セット パラメータの編集.....	69
転送セットの削除.....	70
転送セットへのオブジェクトの追加.....	70
転送セットからオブジェクトを削除.....	70
DCM データベースの複製.....	71
Pack と Unpack を使用した DCM データベースのコピー 71	
データベースの全 DCM メンバーの転送.....	74
「ミラー」データベースの作成と管理.....	76
DCM ディレクトリの管理.....	77
<b>サンプル DCM シナリオ</b>	
<b>マスタとサテライト</b>	<b>79</b>
はじめに 79	
前提条件.....	80
DCM 手法の確立.....	81
データベースの初期化.....	83

マスタの設定 .....	84
サテライトへの初期転送の実行 .....	86
サテライトの設定 .....	87
サテライトの転送セットを初期化 .....	88
定期的なサテライトからマスタへの転送の実行 .....	89
定期的なマスタからサテライトへの転送の実行 .....	90
ソフトウェア開発を続ける .....	91
ベースラインについて .....	92
サテライトの追加 .....	93
まとめ .....	94

## 分散型変更依頼の追跡 95

はじめに 95	
非分散型変更依頼追跡の概要 .....	96
分散型変更依頼追跡の概要 .....	98
分散型変更依頼手法の計画 .....	101
他の分散型変更依頼手法の概要 .....	102
DCM 複製トポロジのタイプ .....	107
Rational Synergy Distributed と Rational Change Distributed (DCS) の概要 .....	108
Rational Change ダイアログと用途の概要 .....	111
Rational Change Distributed モデルパラメータ .....	112

## 上級 DCM トピック 113

はじめに 113	
データベースと管理のハンドオーバ .....	114
管理データベースへの遷移 .....	115
汎用プロセスとリリース固有プロセスの複製 .....	116
DCM データベース定義の複製 .....	121
電子メール通知 .....	122
リコンフィギュア プロパティの全展開 .....	123
関連ベースラインを含む .....	125
転送パッケージの構造 .....	126
転送モード .....	129
DCM パフォーマンス .....	137
転送パッケージの場所 .....	139
自動受取りの使用法 .....	140
転送セットの紛失 .....	143
受取りの失敗 .....	144

---

クラスタ ID .....	147
タイプ定義 .....	148
パラレル検査 .....	149
DCM が生成時刻を格納する方法 .....	150
DCM 受取り時のワークエリアの扱い .....	151
ワークエリア サブディレクトリ テンプレートとパス .....	153
転送セット内のオブジェクト履歴 .....	154
リリース 7.1 以前の Rational Synergy Distributed リリース間の転送 .....	155
DCM 設定 .....	157
DCM イベント ログ .....	159
ブロードキャスト データベースとパッケージ .....	160
<b>用語集</b> .....	<b>161</b>
<b>付録 : 特記事項</b> .....	<b>171</b>
<b>索引</b> .....	<b>174</b>



# 1

## はじめに

この章では、本書を読むための前提条件を説明します。本書では、IBM® Rational® Synergy Distributed の機能を説明します。Rational Synergy Distributed は、分散型構成管理 (Distributed Configuration Management; DCM) ツールです。

### 本書の対象読者

本書は、DCM マネージャおよび構成管理 (Configuration Management; CM) アドミニストレータを対象としています。

DCM マネージャは、DCM クラスタ内のデータベース間で共有しているデータを調整します。特に手法に関する記述、操作の説明、および 79 ページの「サンプル DCM シナリオ マスタとサテライト」を熟読してください。

CM アドミニストレータは、DCM ツールの日々の使用を監督します。特に 21 ページの「DCM クラスタの計画」、61 ページの「DCM データベースの管理」、および [上級 DCM トピック](#) を熟読してください。

### DCM 情報の参照先

DCM 情報は、本書および Synergy Classic ヘルプを参照してください。一般的に、本書では、DCM の概要と操作方法を説明しています。これらの操作を行う具体的な手順については、ヘルプを参照してください。

本書には、出版した時点で最新の情報を記載しています。以後追加された情報は、[IBM Rational Synergy Information Center](#) の Rational Synergy Readme を参照してください。

## IBM Rational ソフトウェア サポートへの問い合わせ

お手持ちのリソースで、問題が解決されない場合は、IBM®Rational® ソフトウェア・サポートに連絡してください。IBM® Rational® ソフトウェア・サポートでは、製品の問題解決に関する支援を行っています。

### 前提条件

IBM Rational ソフトウェア・サポートに問題を送信するには、有効な Passport Advantage® ソフトウェア保守契約が必要です。パスポート・アドバンテージは、IBM の包括的ソフトウェア・ライセンスおよびソフトウェア保守 (製品のアップグレードおよび技術支援) オファリングです。次のサイトからオンラインでパスポート・アドバンテージに登録できます。<http://www.ibm.com/software/lotus/passportadvantage/howtoenroll.htm>

- パスポート・アドバンテージについて詳しくは、パスポート・アドバンテージ FAQ ( [http://www.ibm.com/software/lotus/passportadvantage/brochures\\_faqs\\_quickguides.html](http://www.ibm.com/software/lotus/passportadvantage/brochures_faqs_quickguides.html)) にアクセスしてください。
- さらに支援が必要な場合は、IBM 担当員に連絡してください。

問題をオンラインで (IBM Web サイトから ) IBM Rational ソフトウェア・サポートに送信するには、さらに以下が必要です。

- IBM Support Web サイトの登録ユーザーであること。登録について詳しくは、<http://www-01.ibm.com/software/support/> を参照してください。
- 許可された呼び出し元としてサービス要求ツールにリストされていること。

### 問題報告について

次のようにして、IBM Rational ソフトウェア・サポートに問題を送信します。

1. お客さまの問題のビジネス・インパクトを判別します。IBM へ問題を報告する際は、重大度レベルを問われます。そのため、報告する問題とそのビジネス・インパクトを理解して、評価する必要があります。

重大度のレベルを決めるにあたっては、下表を参照してください。

重大度	説明
1	問題は <b>危機的な</b> ビジネス・インパクトを持ちます。プログラムを使用できず、業務に <b>重大な</b> 影響が出ています。この状況には、即時に解決策が必要とされます。
2	問題は、 <b>重大な</b> ビジネス・インパクトを持ちます。プログラムは使用可能ですが、非常に <b>限定</b> されています。
3	問題は <b>部分的な</b> ビジネス・インパクトを持ちます。プログラムは使用可能ですが、 <b>比較的</b> 重要でない（業務に大きな影響はない）機能が利用できません。
4	問題は <b>わずかな</b> ビジネス・インパクトを持ちます。問題による業務への影響がほとんどないか、問題に対する有効な回避策が実施済みです。

2. 問題を説明して、背景情報を収集します。IBM に問題を説明する際は、なるべく具体的に説明してください。IBM Rational ソフトウェア・サポートの専門家が、問題を解決するために効果的な支援をできるように、関連するすべての背景情報を含めてください。時間を節約するために、以下の質問の答えを用意してください。

- 問題の発生時に実行していたソフトウェア（複数可）のバージョンは何ですか？

次のオプションを使用して、正確な製品名とバージョンを判別することができます。

IBM Installation Manager を始動して、「ファイル」> 「インストール済みパッケージの表示」を選択します。パッケージ・グループを展開し、パッケージを選択して、パッケージ名およびバージョン番号を確認します。

製品を始動して、「ヘルプ」> 「製品情報」をクリックし、オフリング名とバージョン番号を確認します。

- オペレーティング・システムおよびバージョン番号（サービス・パックまたはパッチを含む）は何ですか？
- 問題の症状に関連するログ、トレース、およびメッセージはありますか？
- 問題を再現できますか？再現できる場合は、問題を再現するための手順は何ですか？

- システムに変更を加えましたか？例えば、ハードウェア、オペレーティング・システム、ネットワーク・ソフトウェア、またはその他のシステム・コンポーネントに変更を加えましたか？
  - 現在、問題に対する何らかの回避策を使用していますか？使用している場合は、問題の報告時にその回避策も説明する準備をお願いします。
3. IBM Rational ソフトウェア・サポートに問題を送信します。次の方法で、IBM ソフトウェア・サポートに問題の送信ができます。
- オンラインの場合：IBM Rational ソフトウェア・サポートの Web サイト (<https://www.ibm.com/software/rational/support/>) にアクセスして、Rational サポート・タスク・ナビゲーターで「サービス要求を開く (Open Service Request)」をクリックします。エレクトロニクス問題報告ツールを選択し、「問題管理レコード (PMR) (Problem Management Record (PMR))」を開き、問題についてご自身の言葉で正確に記述してください。
  - サービス要求を開く方法について詳しくは、<http://www.ibm.com/software/support/help.html> にアクセスしてください。
  - IBM Support Assistant を使用してオンラインのサービス要求を開くこともできます。詳しくは、<http://www-01.ibm.com/software/support/isa/faq.html> を参照してください。
  - 電話の場合：国または地域別の電話番号を調べるには、<http://www.ibm.com/planetwide/> の「IBM directory of worldwide contacts」で、お住まいの国名または地域名をクリックします。
  - IBM 担当員に依頼する場合：オンラインまたは電話で IBM Rational ソフトウェア・サポートにアクセスできない場合は、IBM 担当員に連絡してください。必要な場合は、お客さまに代わって、IBM 担当員がサービス要求を開くことができます。<http://www.ibm.com/planetwide/> で、各国への詳しい連絡先情報を検索できます。

送信した問題が、ソフトウェアの障害に関するものか、資料の欠落や不正確な記述によるものである場合は、IBM ソフトウェア・サポートはプログラム診断依頼書 (APAR) を作成します。APAR には、問題の詳細が記述されます。IBM ソフトウェア・サポートは可能な限り、APAR が解決されてフィックスが提供されるまでの間に実施できる回避策を提供します。IBM は、同一の問題を経験している他のユーザーが同じ解決方法を利用できるように、ソフトウェア・サポート Web サイトに解決済みの APAR を公開し、毎日更新しています。

## その他の情報

Rational ソフトウェア製品、ニュース、イベント、その他の情報については、[IBM Rational Software Web site](#) をご覧ください。

## IBM Rational Synergy ドキュメント

IBM Rational Synergy は、すべての製品ドキュメントを公開しています。[IBM Rational Synergy Information Center](#) にアクセスし、必要なドキュメントの電子版を表示するか、ハードコピーを印刷できます。

## 本書の構成

本書の構成は、以下のとおりです。

- 9ページの「DCMの概要」  
DCMの概要を示し、いくつかの機能を簡単に説明します。
- 21ページの「DCMクラスタの計画」  
DCMの仕組みと導入計画の作成方法を説明します。
- 39ページの「DCMデータベースの設定」  
DCMデータベースの設定方法と、他のデータベースへ転送する方法を説明します。
- 49ページの「DCMの使用」  
生成と受信操作を行う方法を説明します。
- 61ページの「DCMデータベースの管理」  
DCMデータベースの管理方法を説明します。
- 79ページの「サンプル DCM シナリオ マスタとサテライト」  
DCMクラスタの設定方法と使用方法の例を示します。
- 95ページの「分散型変更依頼の追跡」  
分散型変更依頼を管理する方法を説明します。
- 113ページの「上級DCMトピック」  
一般的にアドミニストレータが使用する上級 DCM 機能を説明します。
- 161ページの「用語集」  
本書で使用している用語を定義しています。

## 本書で使用する参照資料

ほとんどの Rational Synergy Distributed 操作は Synergy Classic インターフェイスから行う必要があります。本書では、各種の Rational Synergy ダイアログを参照します。特に明記されていないかぎり、ヘルプに含まれる特定のダイアログ、ダイアログ フィールドとオプション、および情報の参照は、Synergy Classic インターフェイスに関するものです。コマンドライン操作に関する詳細は、Classic CLI ヘルプを参照してください。このヘルプは、Synergy Classic CLI の実行中に使用できます。また、[IBM Rational Synergy Information Center](#) からも参照できます。

## ガイドで使用する表記規則

ここでは、本書で使用している表記規則を説明します。

### コマンドとパスの構文

本書では、DCM コマンドオプションを UNIX® のダッシュ (-) 区切り文字を付けて示します。UNIX クライアントを使用している場合は、コマンドオプションではスラッシュを UNIX のダッシュ (-) 区切り文字に、パス名ではバックスラッシュをスラッシュに読み替えてください。UNIX パスと Windows® パスが異なる場合は、両方のパスを示します。

### 書体と記号

下表に、本書で使用している書体と記号の規則を示します。

書体	説明
イタリック	用語に使用されます。ロール ( <i>developer</i> )、状態 ( <i>working</i> )、グループ ( <i>ccm_root</i> ) およびユーザー ( <i>john</i> ) の名前も表します。
太字	ダイアログボックス名とオプション、選択項目、メニューパスを表します。強調にも使用されます。
Courier	コマンド、ファイル名、ディレクトリパスに使用されます。表示どおりに入力するコマンド構文を表します。コンピュータの画面に表示される文字を表します。属性名 ( <i>modify_time</i> )、関数名 ( <i>remote_type</i> )、および型名 ( <i>csrc</i> ) にも使用されます。
<i>Courier Italic</i>	ユーザーが指定するコマンド文字列内の値を示します。たとえば、 <i>drive:¥username¥commands</i> 。

本書ではさらに以下の規則を使用しています。

注記：注意すべき情報を示します。

注意！守らないとデータベースまたはシステムに重大な被害を及ぼす可能性のある情報を示します。





# 2

## DCM の概要

### はじめに

ここでは、DCM の概要を示し、いくつかの機能を簡単に説明します。以下のトピックについて説明します。

- 10 ページの「DCM とは」
- 11 ページの「機能」
- 12 ページの「用語解説」
- 17 ページの「オブジェクト命名規則」
- 19 ページの「ロール」

## DCM とは

DCM は、分散型構成管理ツールです。DCM を使用すると、複数の Rational Synergy データベース間で、世界中の Rational Synergy データベース内で並行して作成および変更されているオブジェクトを共有できます。

DCM は、通常の Rational Synergy データベースを使用してソフトウェア開発オブジェクトを共有します。これらのオブジェクトには、製品の作成に使用するファイル、製品そのもの、プロジェクト、フォルダ、変更依頼、およびタスクがあります。DCM は、変更の論理的なセットを集めて他のデータベースへ送ることが可能なタスクベースの CM 機能を使用します。タスクベースでソフトウェア開発を行っていない場合でも、このツールを使用できます。

DCM を使用する理由には、以下のようなものがあります。

- 複数ファイルをサポートするため
- ファイルのローカライゼーションと移行のため
- データベースを分割するため
- 各種レベルのテストのため
- セキュリティ上の理由

他のデータベースで局所的に行った変更や、他国にいる開発チームが行った変更を共有する場合、DCM でそのプロセスを管理できます。

注記：本書では、読者が Rational Synergy を熟知していることを前提としています。少なくとも、CM の手法とタスクベースの CM ビルド管理作業を行う方法を理解する必要があります。

---

## 機能

DCM は、分散およびパラレル開発を目的として作られています。ここではその主な機能について説明します。

### 透過性

DCM は、開発者の日常業務にはほとんど影響しません。DCM のデータベースは、最低限の管理で済むよう設定できます。

### 既存の変更管理パッケージとの統合

DCM は、Rational Synergy と同じユーザー インターフェイスからアクセスし、コマンドライン操作をサポートしています。いったん DCM を使用するよう Rational Synergy データベースを初期化すると、そのデータは自動的に適切な Rational Synergy ダイアログに表示されるようになります。開発者は、Rational Synergy ダイアログでこのデータを表示しますが、作業習慣を変更する必要はありません。

### 柔軟性

DCM では、DCM を使用するよう初期化された Rational Synergy データベースへ、ソース ファイル、プロジェクト、フォルダ、およびタスクを任意の組み合わせで送ることができます。これらのオブジェクトのグループ化方法に制限はありません。DCM ではタスクベースの手法を使用できます。また多くの分散型手法もサポートしています。

### パラレル（並行型）開発のサポート

DCM は、Rational Synergy のパラレル通知とマージおよび比較機能を使用してパラレル開発をサポートしています。DCM データベースが、パラレルバージョンを持つ新しいオブジェクトバージョンを受取ると、パラレルバージョンを所有する開発者または作成した開発者は電子メールで通知されます（[パラレル検査](#)を参照）。そして Rational Synergy のマージと比較機能によりパラレルバージョンの不具合を解決します。

### タイムゾーンからの独立

DCM は、異なるタイムゾーンにある複数のデータベースをサポートします。データをデスティネーション データベースへ転送するとき、DCM はすべての時刻をグリニッジ標準時刻（GMT）に合わせて記録することで、タイムゾーンからの独立を保証します。

## 用語解説

DCM を使用する前に、以下の用語とその意味を理解する必要があります。

### DCM データベース

DCM データベースは、DCM を使用するよう初期化されている Rational Synergy データベースです（詳細については、14 ページの「DCM 初期化」を参照）。

### ソースデータベースとデスティネーションデータベース

ソースデータベースは、別のデータベースへデータを送る元の DCM データベースです。デスティネーションデータベースは、別のデータベースからデータを受取る DCM データベースです。1 つのデータベースは、ソース データベースかデスティネーション データベースまたはその両方になることができます。

### DCM クラスタとデータベース ID

DCM クラスタは、データを共有する DCM データベースの集まりです。クラスタ内の各データベースには、手作業で固有の [データベース ID](#) (dbid) が与えられます。データベース ID は、ソースとデスティネーション データベースの定義に使用されます。これらは、クラスタ内の各オブジェクト バージョンを固有なものにする目的にも使用されます。たとえば、DCM は、irvine データベース内に作られる main.c オブジェクトに Irvine データベース ID を割り当てます。これにより、このオブジェクトは、san\_jose データベース内に作られる main.c オブジェクトと区別できます。

### デスティネーション データベース定義、転送セット、および転送パッケージ

DCM は、デスティネーション データベース定義を使用してデータをデスティネーション データベースへ送ります。この定義は、デスティネーション データベースに名前を付け、説明し、転送情報を与えます。

転送セットは、デスティネーション データベースへ送ることができる関連するオブジェクトのグループです。

デスティネーション データベースへデータを送るとき、ソース データベースでは次のことが起こります。

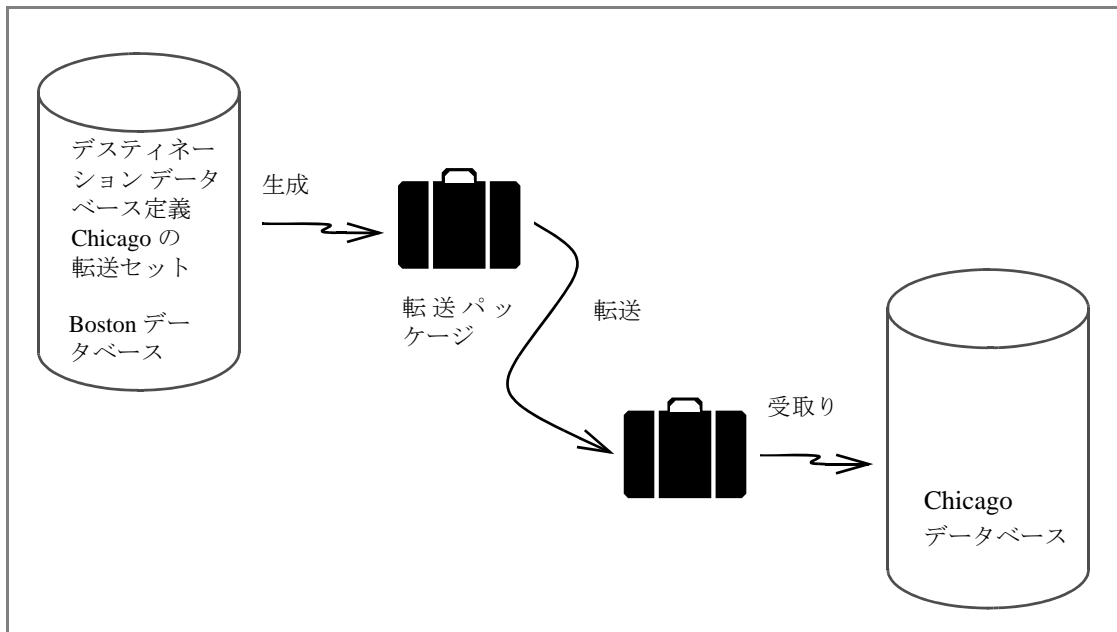
- 1) データを転送する各データベースに対してデスティネーション データベース定義が作成される
- 2) 送られる各オブジェクトのグループに対して転送セットが定義される
- 3) 各転送セットはデスティネーション データベース定義と対にされる（各対は転送パッケージという）
- 4) データが送られる

## 生成、転送、および受取り

データをソース データベースからデスティネーション データベースへ送るには、生成、転送、受取りの 3 つの DCM 操作を行う必要があります。生成操作は、転送パッケージを作成します。このパッケージは、転送操作によりソース データベースからデスティネーション データベースへ送られます。そして、転送されたデータは、受取り操作によりデスティネーション データベースに入れられます。

DCM では、生成、転送、および受取り操作を組み合わせることができます。1 つの動作で、転送パッケージの生成と転送または、転送パッケージの生成、転送、および受取りを行うことができます (140 ページの「自動受取りの使用法」を参照)。また、これらの操作を個別に行うこともできます。

下図は、Boston データベースから Chicago データベースへ送られたパッケージの生成、転送、および受取り処理を示します。



## DCM 初期化

Rational Synergy データベースを DCM で使用できるようにするには、データベースに対して DCM 初期化を行う必要があります。この操作は、データベース、現在データベースにあるオブジェクト、および将来データベースに追加される任意のオブジェクトに対して、DCM プロパティを設定します。詳細については、41 ページの「データベースの初期化」を参照してください。

## DCM オブジェクトに割り当てられるプロパティ

DCM データベース内のすべてのオブジェクトには、自動的に特別のプロパティが割り当てられます。プロパティの例としては、Local To、From、Modifiable In、および Work in DB があります。DCM は、これらの特徴を使用して DCM クラスタ内の全オブジェクトの一意性と履歴を保持します。以下にこれらのプロパティを説明します。

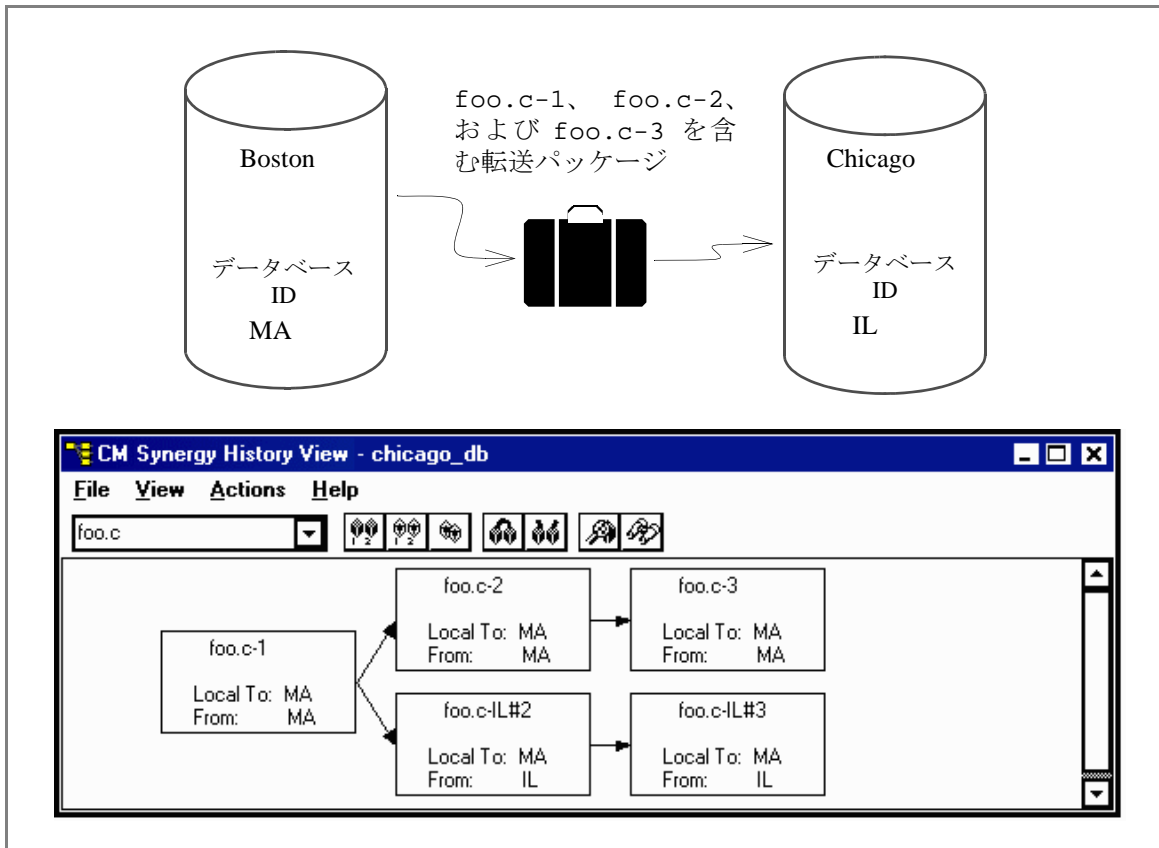
**Local To :** Local To プロパティ (local\_to 属性) の値は、オブジェクトが初期化されているか作成されているデータベースのデータベース ID です。Local To 値は、オブジェクトのすべてのバージョンに残ります。これは、オブジェクトの履歴を通して変わることはありません。DCM オブジェクトのすべての継承者は、新しいバージョンがどこへチェックアウトされたかに関わらずその Local To 値を受け継ぎます。

**From :** From プロパティ (created\_in 属性) の値は、オブジェクトバージョンのチェックアウト元のデータベースの ID です。From 値は、特定のバージョンにのみ割り当てられ、後継バージョンに引き継がれません。

注記: オブジェクトが作成 (ユーザーまたは DCM 初期化により) されると、その From プロパティの値は、そのオブジェクトが作成されたデータベースの ID に設定されます。したがって、DCM オブジェクトの最初のバージョンでは Local To と From 値は常に同じです。

**Modifiable In または Work in DB :** Modifiable In プロパティ (modifiable\_in 属性) の値は、オブジェクトをコントロールするデータベースの ID です。DCM では、DCM クラスタ内の 1 つのデータベースで特定のオブジェクトのみ変更できます。他のすべてのデータベースでは、後の状態への移行を除いてそのオブジェクトを変更することはできません。Modifiable In プロパティがない場合、管理データベースは From (created\_in) プロパティで決まります。通常、ファイル、ディレクトリ、およびプロジェクトはデフォルトで Modifiable In プロパティを持っていません。管理データベースの概念の詳細については、[データベースと管理のハンドオーバー](#)を参照してください。

下図は、foo.c というオブジェクトの Rational Synergy Classic History View を示しています。これは Boston データベースに作られたので、最初のバージョン (foo.c-1) の Local To と From 値は両方とも Boston データベースの ID (MA) に設定されています。詳細については、前述の[注記](#)を参照してください。Chicago データベースから foo.c をチェックアウトすると、From 値は Chicago データベースの ID (IL) に設定されます。Boston データベースから foo.c をチェックアウトすると、From 値は Boston データベースの ID (MA) に設定されます。図に示すように、Local To 値はオブジェクトの全履歴を通して変わりません。



これらのプロパティの他に、すべての DCM オブジェクトにはそれが固有であることを保証する属性が割り当てられます。たとえば、各オブジェクトバージョンは固有の [クラスタ ID](#) にリンクされています。これら、およびオブジェクトの他の特徴は、以下を行うセキュリティ機能を提供しています。

- オブジェクトが管理データベース内で他のデータベースのデータで上書きされるのを防ぐ。
- オブジェクトの制御を1つのデータベースから別のデータベースへ渡すことを許可する。これは、[modifiable in](#) 属性を使用して行います。
- データベース間でコピーされた非静的オブジェクトに対する変更が紛失したり上書きされたりするのを防ぐ。

### マスタ ビルド、マスタ プロジェクト、およびマスタ フォルダ

DCM を使用する前に、データベース間の情報の流れの計画を作成する必要があります。これを手法とよびます。選択肢として多くの標準的な手法があります。たとえば、マスタとサテライトは、1つのデータベース（マスタ）がビルドを行う典型的な DCM 手法です。マスタ ビルドと呼ばれるこれらのビルドは、クラスタ内の他のすべてのデータベース（サテライト）で行われた変更を統合します。

マスタビルドとテストを行うために使用するプロジェクトをマスタ プロジェクトと呼びます。多くの DCM インストレーションには、マスタ統合 *prep* プロジェクトがあります。さらに、ビルドとテストの他のレベルを表すマスタ *prep* プロジェクトを持つこともあります。

タスクベース CM を使用すると、プロジェクトの更新プロパティにフォルダが含まれます。マスタ プロジェクトの更新プロパティに含まれるフォルダをマスタフォルダといいます。

注記：他の DCM 用語の定義は、161 ページの「用語集」を参照してください。



---

## オブジェクト命名規則

DCM は、DCM データベース内で初期化または作成されたオブジェクトに特別の命名規則を使用します。使用する命名規則は、1) オブジェクトの種類と、2) オブジェクトが、作成されるデータベースに局所的であるかどうかに依存します。

以下で説明しているようにオブジェクト属性のバージョン、インスタンス、およびタスク名に DCM 固有命名規則を使用します。

### バージョン

チェックアウト元のデータベース内で局所的ではないオブジェクトをチェックアウトすると、DCM は新しいバージョン番号の前にデータベース ID と DCM 区切り文字を加えます。たとえば、IL データベースにおいて foo.c-1 からチェックアウトすると (foo.c-1 は MA データベースに局所的)、次のバージョンは foo.c-IL#2 となります (IL はデータベース ID で # は DCM 区切り文字)。

注記：データベース ID と DCM 区切り文字は、DCM を使用するようデータベースを初期化したとき設定されます。デフォルトの DCM 区切り文字は「#」です。この他に文字「!」、「~」、および「=」を使用することもできます。

オブジェクトの次のバージョンは、現在のバージョンとオブジェクトのチェックアウト元のデータベースがオブジェクトの[ホーム データベース](#)かどうかによって依存します。下表に、ローカルおよび非ローカルオブジェクトにどのようにバージョンを割り当てるかを示します。

ローカルな場所	初期オブジェクトバージョン	チェックアウト元	新オブジェクトバージョン
MA	foo.c-1	MA (home)	foo.c-2
MA	foo.c-1	IL	foo.c-IL#2
MA	foo.c-IL#2	MA (home)	foo.c-3
MA	foo.c-IL#2	IL	foo.c-IL#3

## インスタンス

DCM は、バージョン管理された DCM オブジェクトのすべてのインスタンスの前にデータベース ID と DCM 区切り文字を加えます。たとえば、MA データベースにローカルなオブジェクトのインスタンス値として MA#1 が可能です(ここで、MA はデータベース ID、# は DCM 区切り文字、1 はインスタンス)。

## タスク名

表示上の理由から DCM は、すべてのタスク番号の前にホーム データベース ID と DCM 区切り文字を加えます。

## プロジェクトとプロジェクト スペック

DCM 用に初期化されたデータベース内のプロジェクトは、DCM データベース ID を含むインスタンスを使用します。たとえば、MA データベース内で作成したプロジェクトのインスタンス値は MA#1 となり、IL データベース内で作成したプロジェクトのインスタンス値は IL#1 となります。

プロジェクトが名前とバージョンでのみ指定されている場合、**Rational Synergy** はデフォルトで現在の DCM データベース ID と DCM 区切り文字と「1」を加えたインスタンスを仮定します。たとえば、MA データベース内に作成されたバージョン 1 の hello という名のプロジェクトは、hello-1 として表示され、これで指定できます。同じプロジェクトは、IL データベースに複製されるとそこでは hello-1:project:MA#1 と表示され、そこではフルオブジェクト名で指定する必要があります。これが必要な理由は、IL データベース内で作成されたバージョン 1 の hello という別の無関係なプロジェクトがあり得るためです。

---

## ロール

一部の DCM 操作には *dcm\_mgr* ロールが必要であり、他には *ccm\_admin* ロールが必要です。ユーザーがこのいずれかのロールを持っていないければ、すべての DCM ダイアログは読み取り専用になります。

*dcm\_mgr* ロールは、以下の操作に必要です。

- デスティネーションデータベースの定義
- 転送セットの定義
- 転送セットへのオブジェクトの追加
- 転送パッケージの生成

*ccm\_admin* ロールは、変更不能オブジェクトのプロパティを変える可能性のある操作に必要です。これには以下の操作があります。

- DCM を使用するためにデータベースを初期化：この操作の最中に DCM は変更禁止オブジェクトにプロパティを追加する可能性があります。
- DCM データベース ID または DCM 区切り文字の変更
- 転送セットの受取り：この操作の最中、DCM は変更禁止オブジェクトにプロパティを追加したり、プロパティを変更したりする可能性があります。

実行しようとする操作に必要なロールになっていない場合、ユーザーがデータベース内でそのロールへのアクセス権を持っていれば、DCM は自動的にユーザーを正しいロールに切り替えます。これをダイナミック ロール スイッチングといいます。操作が終了した時点でユーザーを元のロールへ切り替えます。



# 3

## DCM クラスタの計画

### はじめに

ここでは、DCM の導入を計画する方法を説明します。以下のトピックについて説明します。

- 22 ページの「DCM 手法と複製トポロジの作成」
- 27 ページの「必要ディスク領域の確認」
- 29 ページの「DCM 用ディスク領域確保計画ワークシート」
- 35 ページの「共通データベース プロパティの規定」
- 36 ページの「共通データベース パラメータの規定」
- 38 ページの「エンジンとサーバーの同期」

## DCM 手法と複製トポロジの作成

DCM を効果的に使用するには、データベース間での情報の流れを計画する必要があります。これには2つの部分に関連しています。

- DCM 手法は、どのデータベース間でどんな情報をどのような目的で複製するか、そしてそのような共有変更からどのようにソフトウェア コンポーネントやアプリケーションを作成するかを定義します。典型的な DCM 手法には以下のものがあります。

24 ページの「マスタとサテライト手法」

25 ページの「公開と登録による転送手法」

26 ページの「ピア ツー ピアの転送手法」

- 複製トポロジは、データベースが直接相互にデータを複製するか、あるいは仲介ハブデータベースを介して行うかを定義します。典型的な複製トポロジには以下のものがあります。

ポイント ツー ポイント トポロジ

各データベースは、情報の送り先のデータベースに対する転送パッケージを生成します。このトポロジを使用して変更依頼を転送する方法については、[ポイント ツー ポイント トポロジ](#)を参照してください。

ハブとスポーク トポロジ

スポーク データベースでは、複製は相互に直接行われることはありません。代わりに1つまたは複数のハブ データベースを介して複製されます。このトポロジを使用して変更依頼を転送する方法については、[ハブとスポーク トポロジ](#)を参照してください。旧リリースの Rational Synergy で稼動しているデータベースを含む DCM クラスタでは、ハブ データベースはすべて最新のリリースの Rational Synergy にしてください。詳細については、[リリース 7.1 以前の Rational Synergy Distributed リリース間の転送](#)を参照してください。

DCM 手法と複製トポロジは、ともに転送の性質と方向を定義します。以下の質問に答えながら手法とトポロジを定義してください。

- DCM クラスタの目的は何か。
- DCM クラスタ内の各データベースの目的は何か。
- オブジェクト共有の目的は何か (たとえば、テスト済みプロジェクトの分散や製品の共有など)。
- 特定のデータベースにローカルで、共有されていないオブジェクトはどれか。
- 特定のデータベースにローカルで、他のデータベースと共有されているオブジェクトはどれか。

- 
- 特定のデータベースにローカルではないが、そのデータベースで使用されているオブジェクトはどれか。
  - 各データベースはどんな頻度で転送が必要か。
  - DCM クラスタは、オブジェクトの集中管理が必要か。
  - DCM クラスタを管理するのは1人か(たとえば、DCM アドミニストレータ)。

DCM 手法は、転送パッケージの内容には関与しません。転送パッケージは、手法の実施の一部で、各ソース データベースが管理します。

注記 : DCM クラスタ内では、いろいろなソフトウェア コンポーネントがさまざまな手法を使用しています。たとえば、複数のサイトで開発中のアプリケーションでは、マスタとサテライト手法を使用しているかもしれません。一方で、同じクラスタ内で、テスト/リリース済みコンポーネントとして出荷されるソフトウェア機能は、公開と登録手法を使用して転送されるかもしれません。

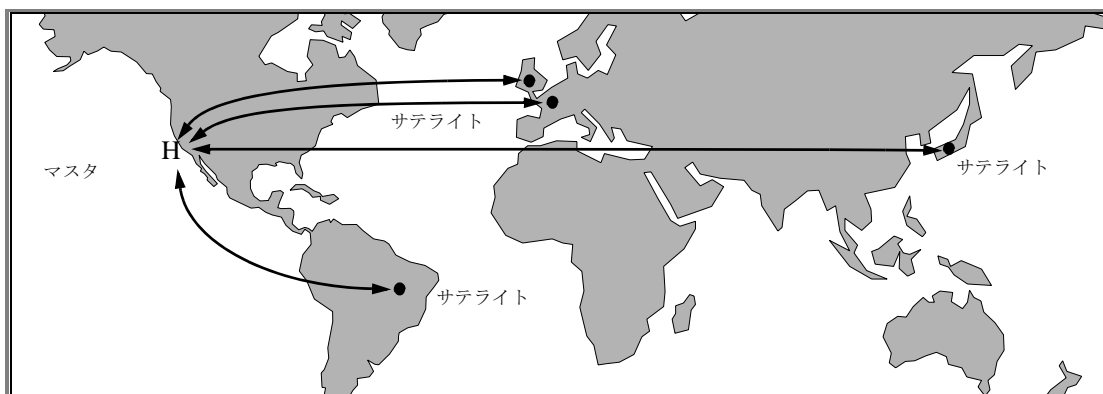
## マスタとサテライト手法

マスタとサテライト手法では、1つのDCMデータベースがマスタに指定されます。クラスタ内の他のすべてのデータベースはサテライトと呼ばれます。マスタは、サテライトで行われた変更を統合するマスタビルドを行います。

マスタとサテライト手法は、アプリケーションのソフトウェアコンポーネントが複数データベース内で開発されていて、それを一緒にビルド、テスト、リリースする必要がある場合に使用します。

以下の図にマスタとサテライト手法を示します。一般的にサテライトは、ポイントツーポイント転送を使用してマスタとの間で直接複製を行います。サテライトは、以下に説明するように、サテライト間で相互に複製を行うことができます。

- ポイントツーポイントトポロジを使用して直接転送  
この方法では、変更がもっとも速くサテライトからサテライトへ転送されます。
- 1つまたは複数のハブデータベースを介して間接的に転送  
多くの場合、マスタデータベースがハブとなり、サテライトが回りのスポークになります。





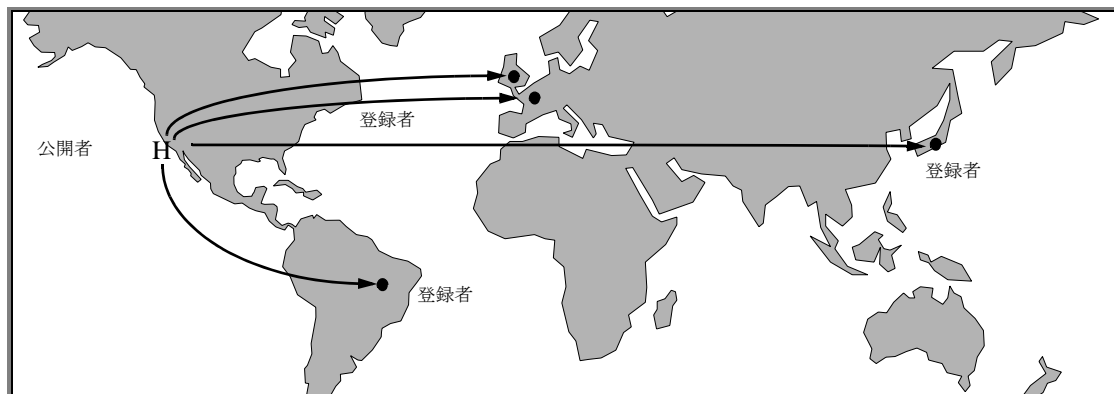
## 公開と登録による転送手法

公開と登録を使用した転送手法では、中央のデータベースが公開者となります。この公開者のデータベースにはすべての共有オブジェクトがあります。登録者は公開者のデータベースから利用可能なデータのサブセットを要求し、公開者がデータを送ります。

データの転送は、公開者データベースから各登録者データベースへの一方行に行われます。しかし、公開者は DCM 操作または Rational Synergy マイグレーション ツールを使用して、他のデータベースから共有データを集めることができます。

公開と登録による転送手法は、多くのデータベースがソフトウェア コンポーネント（ライブラリ、DLL、ヘッダ ファイルなど）を共有する必要があり、共有コンポーネントが登録者のデータベースで開発されていない場合に使用します。

下図に、公開と登録による転送手法を示します。



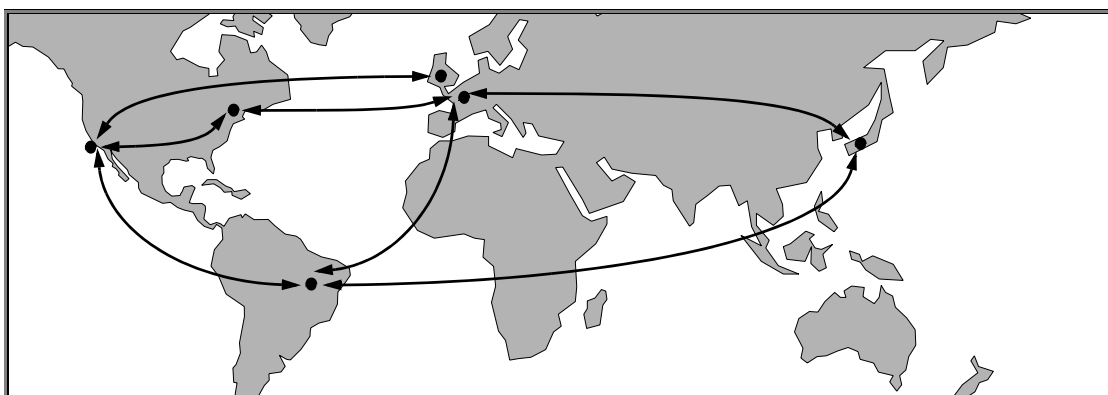
### ピア ツー ピアの転送手法

ピア ツー ピアの転送手法では、マスタ データベースは指定しません。各 DCM データベースは、

- 1) 各ピアへの情報の流れを管理し、
- 2) 統合テスト、SQA テスト、およびリリース用に独自のプロジェクトを維持し、
- 3) ソース データベースから受取ったオブジェクトを使用してプロジェクトをビルドし直し、
- 4) 新しいオブジェクトを要求したデスティネーションデータベースへ送ります。

ピアツーパーの転送手法は、ソフトウェアを分散する柔軟な方法が必要で、中央リポジトリを持つことが不可能か、不要な場合に使用します。

下図に、ピアツーパーの転送手法を示します。



---

## 必要ディスク領域の確認

データを送受信するとき、DCM はソースとデスティネーションデータベースのファイルシステムに書き込みます。したがって、これらのファイルシステムには転送パッケージの生成、転送、受取りのために、十分な空き領域が必要です。以下では、DCM が生成、転送、および受取り操作時にどのようにソースとデスティネーションファイルシステムを使用しているか、および必要なディスク領域を見積もる方法を説明します。

### ソース データベース

DCM 生成操作は、一時ファイルと転送パッケージファイルを [generate\\_directory](#) へ書き出します。以下にその仕組みを説明します。生成操作は、ソース データベースの `generate_directory` に一時エクスポートディレクトリを作成します。転送セットオブジェクトは、一時的にこのエクスポートディレクトリの圧縮 tar ファイルに格納されます。生成操作を正常に行うには、ソース ファイル システムにエクスポートディレクトリおよび圧縮 tar ファイル用に十分なディスク領域が必要です。

生成操作は、転送パッケージのプレビューと情報ファイルも `generate_directory` に書き出します。しかし、これらのファイルは小さいので必要なディスク領域の見積もり時に考慮する必要はありません。

特定の生成操作に必要なディスク領域を見積もるには、29 ページの「DCM 用ディスク領域確保計画ワークシート」を使用してください。

## デスティネーション データベース

DCM の転送と受取り操作は、[receive\\_directory](#) に転送パッケージを書き込みます。以下にこの仕組みを説明します。生成操作時、[generate\\_directory](#) に圧縮 tar ファイルが書き出されます。転送操作は、この圧縮 tar ファイルをデスティネーション データベースの [receive\\_directory](#) 内の一時インポートディレクトリに書き出します。パッケージを ZIP 形式で圧縮する場合、受取り操作は、tar ファイルの解凍と抽出を行い、その後、圧縮ファイルを削除します。受取り操作を正常に行うには、デスティネーション ファイルシステムにインポート ディレクトリおよび圧縮 tar ファイル用に十分なディスク領域が必要です。

受取り操作は、[receive\\_directory](#) に書き出しますが、これらのファイルは小さいので必要ディスク領域を見積もる際には考慮する必要がありません。

転送と受取り操作に必要なディスク領域を見積もるには、29 ページの「DCM 用ディスク領域確保計画ワークシート」を使用してください。

注記：DCM は、生成、転送、または受取り操作を無事に完了すると、一時ファイルと転送パッケージ ファイルを削除しますが、ときにはこれらのファイルが削除されない場合があります。したがって、定期的に [generate\\_directory](#) と [receive\\_directory](#) を調べて残っているファイルがないことを確認します。

## DCM 用ディスク領域確保計画ワークシート

データの送受信を行う前に、以下のワークシートを使用して `generate` と `receive` ディレクトリに必要なディスク領域を見積もります。すでに述べたように、一般的にこれらのディレクトリはそれぞれソースとデスティネーションデータベースにあります。

DCM 用ディスク領域確保計画ワークシート
<b>1. ソースオブジェクト</b>
<p>a. 転送するソースオブジェクトの平均サイズを見積もります。</p> <p>たとえば、プロジェクト階層に複数の 100 KB ソース ファイルがあり、残りのソース ファイルは 30 KB 以下の場合、安全な平均ソース オブジェクト サイズの見積もりとして 80 KB を選ぶことができます。</p> <p style="text-align: right;">平均ソースサイズ = _____ KB</p>
<p>b. <a href="#">通常転送</a></p> <p>に含まれるオブジェクトの数を見積もります。たとえば、それぞれ 200 個のソース オブジェクトを含むプロジェクトを 5 つ転送するとします。毎日の転送時に各プロジェクトにおいて平均 150 のメンバーが変わるとすると、各転送には 750 個のソース オブジェクトが含まれます。</p> <p style="text-align: right;">平均ソース数 = _____ オブジェクト</p> <p>注記：初期転送には、通常、定期的に転送されるオブジェクトよりもはるかに多くのオブジェクトが含まれます。たとえば、200 ファイルを持つプロジェクトが 5 つある場合、初期転送で最低 1,000 個のソース オブジェクトが送られます。オブジェクトが<a href="#">履歴付き</a>で転送され、各ソース ファイルが平均 6 つのバージョンを持つ場合、転送には 6,000 個のソース ファイルが含まれることになります。</p>

DCM 用ディスク領域確保計画ワークシート (Continued)	
<p>c. オブジェクトが履歴付きで転送される場合、変更された各オブジェクトに対して少なくとも2つのオブジェクトが転送されます。したがって、履歴なしで転送する場合は平均ソース オブジェクト数に履歴係数 1 をかけ、履歴付きで転送する場合は 2 をかけます。</p>	<p>履歴係数 = _____ (1 または 2)</p>
<p>d. ソース サイズを見積もります。</p>	<p>ソース サイズ = 平均ソース サイズ (1.a) x 平均ソース数 (1.b) x 履歴係数 (1.c)</p> <p>= _____ KB</p>
<p>e. ソース オブジェクトに関連する他の属性の分として 10% を加えて合計ソース サイズを計算します。</p>	<p>合計ソース サイズ = ソース サイズ (1.d) x 1.10</p> <p>= _____ KB</p>
<b>2. 製品オブジェクト</b>	
<p>a. 転送する製品オブジェクトの平均サイズを見積もります。</p> <p>たとえば、プロジェクト階層に複数の 500 KB 製品ファイルがあり、残りの製品ファイルは 150 KB 以下の場合、安全な平均製品オブジェクトサイズの見積もりとして 300 KB を選ぶことができます。</p>	<p>平均製品サイズ = _____ KB</p>
<p>b. 定期転送に含まれる製品の数を見積もります。</p> <p>たとえば、それぞれ 3 個の製品オブジェクトを含むプロジェクトを 5 つ転送するとします。毎日の転送時に各プロジェクトの製品のうち平均 2 個が変わるとすると、各転送には 10 個の製品オブジェクトが含まれます。</p>	

DCM 用ディスク領域確保計画ワークシート (Continued)

平均製品数 = \_\_\_\_\_

c. オブジェクトを履歴付きで転送する場合、変更された各オブジェクトに対して少なくとも2つのオブジェクトが転送されます。したがって、履歴なしで転送する場合は平均製品数に履歴係数1をかけ、履歴付きで転送する場合は2をかけます。

履歴係数 = \_\_\_\_\_ (1または2)

d. 製品サイズを見積もります。

製品サイズ = 平均製品サイズ (2.a) x  
平均製品数 (2.b) x  
履歴係数 (2.c)  
= \_\_\_\_\_ KB

e. 製品オブジェクトに関連する他の属性の分として10%を加えて合計製品サイズを計算します。

合計製品サイズ = 製品サイズ (2.d) x 1.10  
= \_\_\_\_\_ KB

注記：製品オブジェクトに関連する属性には、10%よりはるかに多くの余裕が必要となる場合があります（たとえば、ヘルプで説明しているように部品構成表を生成する場合）。

DCM 用ディスク領域確保計画ワークシート (Continued)	
3. タスク	
<p>タスクの数に 1 KB をかけて転送内の平均完了タスク数に必要な領域を見積もります。</p> <p style="text-align: center;">合計タスクサイズ = タスク数 x 1 KB = _____ KB</p>	
4. フォルダ	
<p>フォルダを転送する場合、50 個のタスクを含むフォルダには 1 KB のディスク領域が必要だと仮定します。タスク数を 50 で割り、1 KB をかけて転送に含まれる平均フォルダ数に必要な領域を見積もります。</p> <p style="text-align: center;">合計フォルダサイズ = タスク数 / 50 x 1 KB = _____ KB</p> <p style="text-align: center;">注記：クエリベースのフォルダは時間とともにサイズが増えます。見積もりにはこの分を考慮します。</p>	



DCM 用ディスク領域確保計画ワークシート (Continued)

5. プロジェクトと依存関係

a. `ccm export` コマンドを使用して各プロジェクト オブジェクトをエクスポートします。

```
ccm export /t export_directory /h project_name-version:project:dbid#1
```

b. 適切なコマンド (たとえば、Windows の `dir` または UNIX の `du -k`) を使用して、各エクスポート ディレクトリのサイズをキロバイト単位で求めます。

エクスポート ディレクトリ 1  
のサイズ +  
エクスポート ディレクトリ 2  
のサイズ +  
⋮  
エクスポート ディレクトリ N  
のサイズ

\_\_\_\_\_ KB

6. 合計サイズ

必要な合計ディスク領域を計算します。

合計サイズ =  
合計ソース サイズ (1.e) +  
合計製品サイズ (2.e) +  
合計タスク サイズ (3) +  
合計フォルダ サイズ (4) +

\_\_\_\_\_ KB

DCM 用ディスク領域確保計画ワークシート (Continued)
7. 合計 <a href="#">generate_directory</a> サイズ
<p><code>generate_directory</code> には、少なくともステップ6で計算した合計サイズの2.5倍を割り当てます。</p> $\begin{aligned} \text{合計 } \textit{generate\_directory} \text{ サイズ} &= \text{合計サイズ (6)} \times 2.5 \\ &= \underline{\hspace{2cm}} \text{ KB} \end{aligned}$ <p>転送パッケージを次々に送る（つまり転送がキューされている）場合、各転送パッケージに対して領域を割り当てる必要があります。また、失敗した転送操作は成功したものより多くのディスク領域を必要とすることに注意してください。</p>
8. 合計 <a href="#">receive_directory</a> サイズ
<p><code>receive_directory</code> には、少なくともステップ6で計算した合計サイズの2.0倍を割り当てます。</p> $\begin{aligned} \text{合計 } \textit{receive\_directory} \text{ サイズ} &= \text{合計サイズ (6)} \times 2.0 \\ &= \underline{\hspace{2cm}} \text{ KB} \end{aligned}$ <p>転送パッケージを次々に受取る（つまり受取りがキューされている）場合、各転送パッケージに対してディスク領域を割り当てる必要があります。また、失敗した受取り操作は成功したものより多くのディスク領域を必要とすることに注意してください。</p>

注記：データベースを生成と受取り操作の両方に使用する場合、`generate_directory` と `receive_directory` に空き領域が必要です。

---

## 共通データベース プロパティの規定

設計上 DCM クラスタ内のすべてのデータベースは、固有のデータベース ID、同じ DCM 区切り文字、および互換モデルが必要です。以下にこれらのプロパティを説明します。

- 固有のデータベース ID

各 DCM データベースには、手作業で固有の [データベース ID](#) (dbid) を与える必要があります。この ID は、DCM を使用するためにデータベースを初期化 (41 ページの「データベースの初期化」を参照) したとき設定されます。データベース ID は大文字/小文字が区別されたデータベースの論理識別子です。これは、データベース名と同じでもかまいませんが、同じにする必要はありません。

- 同一 DCM 区切り文字

[DCM 区切り文字](#) は、DCM データベース内のすべての非ローカル オブジェクトの名前で使用します。DCM クラスタ内の各データベースは、オブジェクト名を統一するため同じ区切り文字を与える必要があります。この区切り文字は、DCM を使用するためにデータベースを初期化したときに設定されます。ただし、ヘルプで説明しているように、区切り文字は変更できます。DCM 区切り文字の詳細については、17 ページの「オブジェクト命名規則」を参照してください。

- 互換データベースの大文字/小文字設定

ソース データベースとデスティネーション データベースの大文字/小文字設定は、[大文字/小文字設定](#) で説明しているとおり互換性が必要です。

- 同一リリース区切り文字

リリース定義とリリース関連操作の一貫性を保つため、リリース区切り文字は DCM クラスタ内のすべてのデータベースを通して同一である必要があります。

- 互換モデル

すべての共有オブジェクトのライフサイクルとタイプは、DCM クラスタ内のすべてのデータベースを通して互換性が必要です。また、受取り操作時の互換性を保証するため、DCM はまだデスティネーション データベースに定義されていないユーザー定義タイプの定義を自動的に作成します (148 ページの「タイプ定義」を参照)。

## 共通データベース パラメータの規定

DCM を使用する前に、ソース プロジェクトとデスティネーション プロジェクトが同じ更新プロパティを持っていることを確認します。確認しないと、転送したオブジェクトがデスティネーション データベースでプロジェクトに正しく構成されない可能性があります。

更新プロパティは、DCM クラスタ内のすべてのデータベースに対して同じである必要はありません。しかし、ソース データベースとデスティネーション データベースの各ペアは、プロジェクト目的リスト、プラットフォーム、リリース区切り文字、リリース定義、およびタスク属性値の範囲設定が同じ値である必要があります。以下にこれらのプロパティを説明します。

- プロジェクト目的リスト

デスティネーション データベースは、転送された *prep* プロジェクトのプロジェクト目的を認識できる必要があります。このプロパティに使用できる値は、ベース モデル オブジェクトの `project_purpose_list` 属性に格納されています。プロジェクト目的リストを変更するには、**Project Purpose Table** ダイアログまたは **project purpose** コマンドを使用します。このプロパティは、*prep* プロジェクトを作成、チェックアウト、または更新するとき使用します。プロジェクト目的リストの詳細については、ヘルプを参照してください。

- プラットフォーム

デスティネーション データベースは、転送されたオブジェクトのプラットフォーム設定を使用してプロジェクトを更新します。このプロパティに使用できる値は、**Rational Synergy** インストールエリアの `etc\om_hosts.cfg` ファイルに格納されています。

プラットフォーム設定は、タスクベース CM およびオブジェクト ステータスベース CM の両方で使用されます。詳細については、『[IBM Rational Synergy 管理者ガイド UNIX 版](#)』または『[IBM Rational Synergy 管理者ガイド Windows 版](#)』を参照してください。

- リリース区切り文字

リリース区切り文字は、リリース定義のコンポーネント名とコンポーネントリリース部分からリリース値を作成するために使用します。リリース区切り文字は、リリース定義が同じリリース値を生成するため、あるいはリリース値がその対応するリリース定義を参照するためには、DCM クラスタ内のすべてのデータベースで同一である必要があります。

---

- リリース定義

デスティネーションデータベースは、転送されたオブジェクトのリリース設定を使用してプロジェクトのメンバーシップを更新します。したがって、転送されたオブジェクトで使用されている値に対応するリリース定義は同等に定義されている必要があります。これをもっとも簡単に実現するには、リリース定義の複製を許可し、複製に使用する転送セットにリリース定義を含めます。各転送セットは、通常、デフォルトで適格なリリース定義を含みます。詳細については、116 ページの「汎用プロセスとリリース固有プロセスの複製」を参照してください。

- タスク属性値の範囲設定

タスク属性値の範囲設定は、タスク属性に可能な値を定義します。タスクベース CM を使用する場合、デスティネーションデータベースは、デスティネーションデータベースのプロジェクトを更新するために少なくとも一部のタスク属性値を認識できる必要があります。

デフォルトで、タスクベースの更新にはタスクのリリース属性のみが使用されます。`platform` と `task_subsys` 属性も使用できます。デスティネーションデータベースでは、これらの属性をもとにフォルダクエリを設定できます。これを行うため、デスティネーションデータベースの属性値範囲設定は当てはまる場合には `release`、`platform`、および `task_subsys` 属性の値が含まれる必要があります。

タスク属性範囲は、`database_path\pt\attrange.dft` ファイル内で定義されています。`release`、`platform`、または `task_subsys` 属性が他の属性によって定義されている場合、それらの属性はデスティネーションデータベースの `attrange.dft` ファイルに必要です。

少なくとも、転送されたオブジェクトが使用している `release` 値はデスティネーションデータベースで定義されている必要があります。その上、転送されたオブジェクトが使用しているクエリが `platform` あるいは `task_subsys` 値を参照している場合、これらの値もデスティネーションデータベースで定義されている必要があります。

注記：[Rational Change](#) には、ユーザーが属性範囲値を設定できる独自の管理インターフェイスがあります。

## エンジンとサーバーの同期

**Rational Synergy** エンジンとサーバー マシンの時刻は同期している必要があります。デフォルトで、DCM は同じ **Rational Synergy** データベースにアクセスするマシンの時刻が最大 60 秒までずれていることを許可します。あるマシンと他のマシンの時刻のずれが 60 秒を超える場合、DCM 転送パッケージにオブジェクトを含めることはできません。これがどのように起きるか、以下のシナリオで示します。

1. 時刻が正確なサーバーで DCM 生成が行われます。
2. 数秒後、ユーザーはエンジン プロセスが実際の時刻より 100 秒遅れているマシン上で実行している **Rational Synergy** セッションを使用してファイルをチェックインします。
3. 数秒後、正確な時刻のマシン上で別の DCM 生成が行われます。オブジェクトは、チェックインされなかったため、最初の DCM 転送パッケージに入りません。

しかし、最後の生成時刻より古いため、2 番目の DCM 転送パッケージにも入りません。

DCM 生成を行うとき、DCM は現在の時刻からわずかな時間（デフォルトで 60 秒）を引いて生成時刻に使用します。これにより、同じデータベースにアクセスするマシン同士が 60 秒ずれていることができます。つまり、DCM 生成の開始から 60 秒以内に変更されたか転送セットの直接または間接メンバーになるオブジェクトは、その DCM 転送パッケージおよび次の DCM 転送パッケージに入れることができます。

同期処理は、インストラクションごとに異なります。この操作に関する具体的な内容は、システム管理者にお尋ねください。

# 4

## DCM データベースの設定

### はじめに

データベース間でオブジェクトを転送できるようにするには、1) 両方のデータベースを DCM を使用するよう初期化し、2) どちらのデータベースが転送パッケージを送り、どちらが受取るかを決定し、3) 各データベースが他のデータベースへデータを送るとき使用する転送セットを定義する必要があります。ここでは、以下の項目に従って、これらの作業を行う方法を説明します。

- 40 ページの「手法」
- 41 ページの「データベースの初期化」
- 42 ページの「プロセスの見直しと選択」
- 43 ページの「DCM データベース定義の作成」
- 44 ページの「転送セットの作成」
- 45 ページの「転送セットへのオブジェクトの追加」

## 手法

データベースを DCM で使用できるようにするには、データベースに対して DCM 初期化を行う必要があります。これは、**Initialize DCM** ダイアログまたは `ccm dcm /init` コマンドを使用して行います。

注記: データベースの初期化を行う前に 113 ページの「上級 DCM トピック」を参照してください。DCM の使用方法に応じて、上記を参照して DCM パラメータのデフォルト設定を変更する必要があるか判断してください。

データベースをデスティネーションとしてのみ使用する場合は、データベースを初期化し、変更したい DCM 設定を修正すれば設定は終了です。しかし、データベースをソース（またはソースとデスティネーション）として使用する場合は、以下のことを行う必要があります。

- **DCM データベースの定義**  
転送セットは、デスティネーションとして定義されているデータベースへのみ送ることができます。これらのデータベースは、転送前ならいつでも定義できます。
- **転送セットの定義**  
転送セットは、どのオブジェクトをソース データベースから送るかを定義します。転送する必要があるオブジェクトの論理グループそれぞれに対して個別の転送セットを定義できます。
- **転送セットへのオブジェクトの追加**  
いったん転送セットが定義されると、オブジェクトが追加されます。



---

## データベースの初期化

データベースに対して DCM 操作を行う前に、DCM を使用するようデータベースを初期化する必要があります。この手順は、すべての DCM 操作を可能にし、Synergy Classic メニューに DCM オプションを追加し、DCM が DCM クラスタ内で固有のオブジェクト名を保証するバージョン付けを可能にします。

### 作業を開始する前に

DCM を使用するようデータベースを初期化する前に、以下の手順を行います。

1. 前述の[注記](#)を読む。
2. 自分のセッションがデータベースで実行している唯一のセッションであることを確認する。
3. データベースを保護する（『[IBM Rational Synergy 管理者ガイド UNIX 版](#)』または『[IBM Rational Synergy 管理者ガイド Windows 版](#)』の ccm monitor と ccldb protect コマンドの説明を参照）。
4. データベースの有効なバックアップを用意する。

### 初期化操作

この操作は、GUI またはコマンドラインから行うことができます。この操作を行うとき、DCM クラスタ内の各データベースに固有の[データベース ID](#) とすべてのデータベースで同じ [DCM 区切り文字](#) を与える必要があります。また、このデータベースの説明、場所、およびアドミニストレータの何らかの詳細を加えてください。この情報は、クラスタ間で複製できるので、他の人がデータベースの用途、場所、および管理関係の連絡先を知るのに役立ちます。詳細については、121 ページの「DCM データベース定義の複製」を参照してください。

データベースを初期化すると、データベース内の各オブジェクトには自動的に[クラスタ ID](#) が割り当てられます。この属性は、147 ページの「クラスタ ID」で説明しているようにデータベース間にわたって名前の変更を追跡するためにあります。

この操作は、Synergy Classic ヘルプの「[DCM 用にデータベースを初期化](#)」で説明しています。

## プロセスの見直しと選択

DCM 用に初期化されていないデータベースでは、デフォルトで **Rational Synergy** は **Standard** (標準) という名のデフォルトプロセスを使用します。このプロセスには、他のデータベースのタスクをローカルに完了したものと区別しないで処理する、組み込まれた汎用プロセスルールがあります。たとえば、**Integration Testing** (統合テスト) には、ローカルおよび非ローカルタスクを含む、すべての完了タスクが集められます。

データベースを DCM 用に初期化すると、**Rational Synergy** は **Distributed** (分散) という追加プロセスも作成します。このプロセスは 2 階層の開発とテストをサポートする点で **Standard** (標準) プロセスと異なります。

- **Local Collaborative Development** (ローカル共同開発) は、他の開発者のローカルに完了した変更は含むが他のデータベースの完了した変更は含まないという、共同開発に使われます。
- **Local Integration Testing** (ローカル統合テスト) は、コンポーネントがビルド可能か、またローカルに完了した変更を使用して実行可能かをテストするために使われます。これには、他のデータベースからの変更は含まれません。
- **Master Integration Testing** (マスタ統合テスト) は、コンポーネントがビルド可能か、またすべてのデータベースの変更を使用して実行可能かをテストするために使われます。

この 2 階層アプローチには、ビルドを破壊する変更がそのビルドにのみ影響し、リモートサイトの開発者には直接影響しないという利点があります。これは、サイト間でタイムゾーンが大きく異なる場合などに、より管理が可能なプロセスとなります。ただし、**Master Integration Baseline** (マスタ統合ベースライン) が作成されるまで、開発者がプロジェクトを非ローカル変更で更新できないという欠点があります。

DCM 用に初期化したデータベースでは、デフォルトの **Standard** (標準) または **Distributed** (分散) プロセスを使用するか、いずれかのプロセスを修正するか、あるいは独自にデザインしたカスタム プロセスを作成できます。

既存のリリース定義とプロセスルールは、DCM 初期化時に修正されません。DCM 用に初期化されたデータベースがすでに開発に使われている場合、現在のリリースに使われているプロセスを見直し、現在のプロセスがそれらのリリースに適切であるかを確認する必要があります。リリースに使われているプロセスルールを修正するか、汎用プロセスルールをリリース固有プロセスルールにコピーするか、あるいはその両方を行ってそのリリースの改正プロセスを実装できます。

---

DCM 用に初期化したデータベースにまだリリース定義が設定されていなければ、2つのデフォルトのプロセスを見直す必要があります。いずれもリリースにとって適切ではない場合、いずれかを修正するか新しいカスタム プロセスを作成できます。Rational Synergy (Synergy Classic ではなく) からリリースを作成するとき、使用するプロセスを選ぶことができます。

## DCM データベース定義の作成

転送パッケージを所定のデータベースへ送る前に、その DCM データベース定義を作成する必要があります。DCM は、転送パッケージがソース データベースからデスティネーション データベースへ送られるとき常にこの情報を使用します。この操作は、**Create DCM database Definition** ダイアログを使用して行われます。

データベースを直接デスティネーション データベースとして使用する場合、**Generate Allowed** オプション (デフォルト設定) を選択します。ソース データベースからのデータが 1 つまたは複数のハブ データベースを介してこのデータベースに間接的に複製されていて、オブジェクトのコントロールをソース データベースからデスティネーション データベースへ渡したい場合は、**Handover allowed** オプションを選択します。

デスティネーション データベースが旧リリースの Rational Synergy の場合は、155 ページの「リリース 7.1 以前の Rational Synergy Distributed リリース間の転送」を参照してください。

この操作は、Synergy Classic ヘルプの「[DCM データベース定義の作成](#)」で説明しています。

## 転送セットの作成

転送セットは、デスティネーション データベースへ送ることができる関連するオブジェクト グループの集まりです。転送セットには、転送セットの名前、転送時の電子メール通知の送り先、問題の範囲、[generate directory](#)、転送から除外するオブジェクト、および追加情報を複製あるいは追加するかなどのプロパティがあります (116 ページの「汎用プロセスとリリース固有プロセスの複製」、121 ページの「DCM データベース定義の複製」、123 ページの「リコンフィギュア プロパティの全展開」、および 125 ページの「関連ベースラインを含む」を参照)。この操作は、**Create Transfer Set** ダイアログを使用して行います。

転送セットを作成するときは、そのプロパティ (転送セット名、除外するオブジェクトなど) のみを定義します。後で各オブジェクトの **Show DCM Properties** ダイアログ (45 ページの「転送セットへのオブジェクトの追加」を参照) を使用して、転送セットにオブジェクトを追加します。

この操作は、Synergy Classic ヘルプの[「転送セットの作成」](#)で説明しています。

---

## 転送セットへのオブジェクトの追加

転送セットを作成するときは、そのプロパティ（転送セット名、除外オブジェクトなど）のみを定義します。後で各オブジェクトの **Show DCM Properties** ダイアログまたはコマンドラインを使用して、転送セットにオブジェクトを追加します。

転送セットにプロジェクト、タスク、またはフォルダを追加すると、DCM は自動的にそのオブジェクトのすべてのメンバーを追加します。44 ページの「転送セットの作成」で説明しているように、生成操作時に更新（リコンフィギュア）プロパティを完全に展開できます。

下表に、転送セットに各タイプのオブジェクトを追加し、**Fully Expand Reconfigure Properties** オプションを選択していないとき含まれるものを示します。このオプションを選択したとき追加されるオブジェクトは、123 ページの「リコンフィギュア プロパティの全展開」を参照してください。

オブジェクト タイプ	転送セットに追加されるオブジェクト	履歴
フォルダ	フォルダ自身、そのメンバー タスク、およびタスクに関連するすべてのオブジェクト。	適用外
タスク	タスク自身とタスクに関連するすべてのオブジェクト。タスクがコンポーネントタスクの場合は、関連プロジェクトやプロダクトも含まれます。 <b>Include Associated Baseline</b> オプションを選択している場合、タスクを含むすべてのベースラインも含まれます。詳細については 125 ページの「関連ベースラインを含む」を参照してください。	適用外

オブジェクト タイプ	転送セットに追加されるオブジェクト	履歴
プロジェクト	<p>あるプロジェクトの関連タスクのうち自動タスクを除いたもの。このプロジェクトに関連するコンポーネントタスクまたはユーザー作成タスクがある場合は、それらも含まれます。</p> <p><i>prep</i> プロジェクトなど非静的プロジェクトには、以下のものが追加されます。1) プロジェクトが使用している任意のプロジェクトベースライン、2) プロジェクトの更新プロパティ内のすべてのフォルダとタスク、3) フォルダ内のすべてのタスク</p> <p><b>Include Associated Baseline</b> オプションを選択している場合、プロジェクトを含むすべてのベースラインも含まれます。</p> <p>タスクの関連オブジェクトは、プロジェクト階層のメンバーでない限り含まれません。</p> <p>プロジェクトベースラインのサブプロジェクトは、これらのサブプロジェクトがプロジェクト階層内でベースラインとして使用されていない限り含まれません。</p>	オプション
プロジェクト グルーピング	<p>プロジェクトグルーピング自身とそのメンバープロジェクト、およびそのプロジェクトグルーピングのすべてのタスクと削除されたタスク。</p> <p><i>prep</i> プロジェクトのプロジェクトグルーピングのみ複製できます。他の非静的状態のプロジェクトは複製の対象とならないのでそれらのプロジェクトグルーピングは複製されません。</p>	適用外
ベースライン	<p>ベースラインオブジェクト自身、そのベースラインから作成されたコンポーネントタスク、ベースラインのプロジェクトメンバー、およびベースラインのタスクメンバー。コンポーネントタスク、ベースラインプロジェクトメンバーとタスクメンバーは、上記のようにさらに展開されます。</p>	適用外
プロセスルール	<p>プロセスルール自身、およびプロセスルールのフォルダとフォルダテンプレートメンバー。</p>	適用外
プロセス	<p>プロセス定義自身、およびプロセスが使用する汎用プロセスルール。各汎用プロセスルールは上記のように展開されます。</p>	適用外

オブジェクト タイプ	転送セットに追加されるオブジェクト	履歴
プロダクト	オブジェクト自身とプロダクトに関連する非自動タスク。プロダクトに関連する任意のコンポーネントタスクを含みます。	適用外
ソース	オブジェクト自身とオブジェクトの関連タスク。	オプション

DCM 追加操作を使用して転送セットに明示的に追加するオブジェクトを直接メンバーといいます。直接メンバーと関連しているために DCM が自動的に追加するオブジェクトを間接メンバーといいます。

DCM データベース内の適格なオブジェクトをすべて転送するには、組み込まれている Entire database 転送セットを使用します。

注記：オブジェクトを転送セットに追加しても転送から除外される可能性があります。転送パッケージを生成するとき転送セットから除外されるオブジェクトについては、126 ページの「転送パッケージの構造」を参照してください。

この操作は、Synergy Classic ヘルプの「[転送セットへのオブジェクトの追加](#)」で説明しています。





# 5

## DCM の使用

### はじめに

DCM を使用する前に、前の章の手順を実行する必要があります。その作業を終了すると、以下で説明するようなデータの転送と受取りを行うことができます。

- 50 ページの「手法」
- 51 ページの「生成操作について」
- 55 ページの「転送パッケージの生成」
- 57 ページの「受取り操作について」
- 60 ページの「転送パッケージの手作業受取り」

## 手法

DCM では、データベースの共有データを他のデータベースに送ることと、他のデータベースの共有データを受取ることができます。以下にデータの送受信手順を示します。

### 1. 転送パッケージの生成

[転送セット](#)に新しいオブジェクトを入れる必要があるときは、必ず新しい[転送パッケージ](#)を生成する必要があります。これら同じオブジェクトを別のデータベースへ送るには、同じ転送セットと送り先データベースのデータベース ID を使用して、新しい転送セットを生成する必要があります。

### 2. 転送パッケージの送信

DCM は、Manual Copy [転送モード](#)を選択しない限り、転送パッケージを自動的にデスティネーションデータベースへ送ります。

### 3. 転送パッケージの受取り（オプション）

データを手作業で受取るには、デスティネーションデータベースから受取り操作を開始する必要があります。

自動受取りを行うには、選んだ転送モードが、デスティネーションマシンで DCM がリモート コマンドを実行することを許可し、ソースとデスティネーションマシンが 140 ページの「自動受取りの使用法」で説明しているように正しく設定されている必要があります。

---

## 生成操作について

転送パッケージを生成すると、DCM は次の作業を自動的に行います。

- 転送パッケージの計算と作成
- 転送パッケージのデスティネーション データベースへの送信 (オプション)
- デスティネーション データベースでの転送パッケージの受取り (オプション)

転送セットの作成時に電子メール通知を選択すると、DCM は生成、転送、および受取り操作が終わったとき、その結果を自動的に受信者に送ります。このデータは、DCM イベント ログにも記録されます。

### 転送パッケージの内容

DCM 転送パッケージには、5 種類の情報が含まれます。

- データ部分には、ファイル、タスク、プロジェクトなどユーザー オブジェクトに関する情報があります。
- 省略可能なリリース部分には、リリース定義に関する情報があります。
- 省略可能なテンプレート部分には、フォルダ テンプレートに関する情報があります。
- 省略可能なプロセス部分には、プロセスおよびプロセス ルールに関する情報があります。
- 省略可能なタイプ部分には、タイプ定義に関する情報があります。

### DCM が転送パッケージを計算および作成する方法

DCM は、以下を使用して転送パッケージを作成します。

- 転送パッケージが最後に生成された時刻
- 転送セットに含まれるオブジェクト
- デスティネーション データベース定義

デフォルトで、DCM は転送パッケージの過去 30 回の生成時刻を保存します (詳細については、150 ページの「DCM が生成時刻を格納する方法」を参照してください)。転送セット内の各オブジェクト バージョンの変更時刻が最後の生成時刻と比較されます。そして最後の生成時刻よりも新しいオブジェクトが [転送リスト](#) に追加されます。転送パッケージが最後に生成された時刻は、Generate ダイアログの Last Generated フィールドに表示されます。

変更時刻を比較した後、DCM は除外されているか不適格なオブジェクトを転送リストから削除します。これは、DCM 固有の除外ルールを使用して行われます。Exclude ダイアログでは、転送するオブジェクトをさらに限定できます。最終転送リストは、Preview Generate ダイアログに表示されます。これらのオブジェクトは、すべて DCM 転送パッケージのデータ部分に含まれます。

以下のオブジェクトは、DCM が固有の除外ルールを適用すると転送リストから除外されます。

- 変更禁止状態でも *prep* 状態でもないプロジェクト
- 自動タスク
- *prep* プロジェクト以外のプロジェクト グルーピング
- 管理者オブジェクト、転送セット、およびモデル
- 上記以外の変更可能オブジェクト
- Rational Change Distributed のライセンスがない場合の変更依頼オブジェクト
- 組み込まれている定義済みフォルダ テンプレート
- 組み込まれている定義済み汎用プロセス ルール  
フォルダの転送には制限はありません。

Transfer Set Exclude ダイアログの以下のオプションを使用して転送リストからオブジェクトを除外することもできます。

- Exclude all products (全製品を除く)
- Exclude all imported objects (インポートした全オブジェクトを除く)
- Exclude all objects of specified type (指定タイプの全オブジェクトを除く)  
最終転送リスト内のオブジェクトがデスティネーション データベースへ送られます。転送パッケージの内容の詳細については、126 ページの「転送パッケージの構造」を参照してください。

次のオプションを使用して、転送パッケージのリリース、テンプレート、またはプロセス部分にオブジェクトを含めるか除外することができます。

- Exclude Type Definitions (タイプ定義を除く) : transfer set Exclude ダイアログ内のオプション。
- Exclude database Info (データベース情報を除く) : transfer set Exclude ダイアログ内のオプション。
- Release Scope, Release Query (リリース範囲、リリース クエリ) : Create transfer set および Edit transfer set ダイアログ内のオプション

- 
- Replicate Reconfigure and Folder Templates (リコンフィギュアおよびフォルダ テンプレートを複製) : Create transfer set および Edit transfer set ダイアログ内のオプション。

## DCM が転送パッケージを送る方法

生成操作は、手動コピー転送モードまたは手動コピーが必要なユーザー定義転送モードを使用していない限り、自動的に転送パッケージをデスティネーションデータベースへ送ります。

転送パッケージが無事デスティネーションデータベースに到着した後、DCM はソース データベースから転送パッケージを削除します (ダイレクト転送モードを使用したとき以外)。デスティネーションデータベースを定義したとき電子メール通知を選択していると、DCM は転送ステータス電子メールも送ります。

## DCM が転送パッケージを受取る方法

転送パッケージは、ユーザーが手作業でデスティネーションデータベースに受取るか (60 ページの「転送パッケージの手作業受取り」を参照)、ソースデータベースが自動的に受取りを開始 (140 ページの「自動受取りの使用法」を参照) できます。受取り操作の種類 (Manual または Automatic) は DCM データベース定義の作成または編集を行うときに設定されます。

DCM は、ユーザー *ccm\_root* として自動受取りを行います。これは、Rational Synergy エンジンプロセスが受取りを行い、エンジン プロセスが *ccm\_root* として実行するためです。また、*ccm\_root* ユーザーは、変更禁止オブジェクトの変更に必要な *ccm\_admin* ロールを持っています。

ユーザーが手作業で DCM 受取りを行うには *ccm\_admin* ロールが必要です。UNIX サーバーでは、これは *ccm\_root* ユーザーが行います。Windows サーバーでは、これは CM アドミニストレータ ユーザーが行います。いずれの場合も、このユーザーは常に *ccm\_admin* ロールのアクセス権を持ちます。このようリモート実行には、信頼できるホストまたはログインが必要で、保護された環境では、組織のセキュリティ ポリシーがそのような信頼の設定を禁止することがあります。

Automatic Receive が有効な場合、DCM は生成操作の一部としてデータをデスティネーションデータベースにロードします。DCM は、デスティネーション マシンで Rational Synergy セッションを開始し、データをロードし、Rational Synergy セッションを停止するコマンドを送って自動受取りを行います。DCM がどのようにデータをチェックしてロードするかは、57 ページの「受取り操作について」を参照してください。

転送セット作成時に電子メール通知を選択していると、その転送セットで受取り操作が終了すると DCM は自動的に受信者に受取りステータス情報を送ります。この情報は、送信先データベースの DCM イベント ログにもあります。自動受取りが失敗すると、デスティネーションマシンの `temp_dir\dcm_ui.log` と `temp_dir\dcm_eng.log` ファイルにエラー情報が残ります。Windows では、`temp_dir` は `%CCM_HOME%\temp` です。UNIX では、`temp_dir` は受取り操作を実行しているマシンのオペレーティングシステムが使用している一時ディレクトリです。

---

## 転送パッケージの生成

GUI またはコマンドラインから [転送パッケージ](#) を生成できます。下図に Generate ダイアログを示します。

The screenshot shows the 'Generate' dialog box with the following configuration:

- Destination Database:**
  - ID: djh63nt2
  - Description: Example satellite database
  - Transfer Mode: Direct
- Receive Options:**
  - Automatic Receive
  - Run In Background
  - Wait for receive lock
- Host Name:** (empty)
- OS:** Windows NT
- Path:** \\voceania\ccmdb\djh63nt2
- DCM\_HOME:** (empty)

**Transfer Set:**

- Only show transfer sets previously used with database djh63nt2
- Show all transfer sets
- Name:** test
- E-Mail Address:** (empty)
- Last Generated:** Never

Buttons: Generate, Preview..., Cancel, Help

この操作を始める前に以下のことを行います。

- ソース データベースが DCM を使用するよう初期化されていることを確認する。

- ソース データベースに転送セットと定義済みデスティネーション データベース (44 ページの「転送セットの作成」および 43 ページの「DCM データベース定義の作成」を参照) があることを確認する。
- デスティネーション データベースが転送パッケージを受取るように設定されていることを確認する (たとえば、デスティネーション データベースセキュリティとホスト ファイルは、DCM がユーザー `ccm_root` として転送できるよう設定されている必要があります)。

この操作は、Synergy Classic ヘルプの「[転送パッケージの生成と送信](#)」で説明しています。



---

## 受取り操作について

転送パッケージは、ユーザーが手作業でデスティネーションデータベースに受取るか（60 ページの「転送パッケージの手作業受取り」を参照）、ソースデータベースが自動的に受取りを開始（140 ページの「自動受取りの使用法」を参照）できます。受取り操作の種類（Manual または Automatic）は DCM データベース定義の作成または編集を行うときに設定されます。

注記：ある時点では、特定のデータベースで1つの受取り操作のみ実行できます（つまり、DCM は1つのデータベースで複数の同時受取りを許可しません）。

ここでは、手作業による受取り操作を説明します。ただし、転送パッケージを手作業で受取ろうと自動的に受取ろうと、データは同じようにロードされチェックされます。

転送パッケージの受取りを行うと、DCM は以下の作業を自動的に行います。

- 転送パッケージに含まれ、デスティネーション データベースに定義されていないタイプ定義をすべてロードする。
- 転送済みオブジェクトをロードする。
- 転送済みオブジェクトの平行バージョンがないか調べる。
- ソース データベースとデスティネーション データベースのリリース情報を比較する。

転送セット作成時に電子メール通知を選択した場合、受取り操作が終了すると DCM は自動的に受信者に以下のものを送ります。

- 1) データロードの情報、
  - 2) 並行バージョン検査結果、および
  - 3) スキップしたオブジェクトおよびコンフリクトの追加レポート
- この情報は、[DCM イベント ログ](#)にも含まれます。

### タイプ定義のロード

タイプ定義が転送セットから除外されていない場合、デスティネーションデータベースで定義されておらず、転送パッケージのデータ部分に必要な転送セット内のタイプ定義は、すべてインポートされます。転送セット内のタイプ定義がすでにデスティネーションデータベース内に存在する場合は変更されません。詳細については、148 ページの「タイプ定義」を参照してください。

### リリース定義のロード

リリース定義が転送セットから除外されていない場合、転送パッケージのリリース部分に含まれるリリース定義はインポートされます。詳細については、116 ページの「汎用プロセスとリリース固有プロセスの複製」を参照してください。

### プロセス、プロセスルールおよびフォルダ テンプレートのロード

更新（リコンフィギュア）テンプレートとフォルダ テンプレートが転送セットから除外されていない場合、転送パッケージのプロセスおよびテンプレート部分に含まれるプロセス、プロセスルール、およびフォルダ テンプレートはインポートされます。詳細については、116 ページの「汎用プロセスとリリース固有プロセスの複製」を参照してください。

### 転送済みデータのロード

DCM は、各転送済みオブジェクトを「クローン」化することによってデスティネーションデータベースにロードします。オブジェクトがデスティネーションデータベースに新しく追加された場合、DCM は、そのプロパティが転送済みオブジェクトとまったく同じ新規オブジェクトを作成します。オブジェクトがすでにデスティネーションデータベース内に存在する場合は、DCM は転送済みバージョンと同一になるようにオブジェクトのすべてのプロパティを更新します。これらのプロパティには、履歴関係など他のオブジェクトとの関係がすべて含まれます。

注記：デスティネーションデータベース内にある変更禁止オブジェクトのプロパティは、オブジェクトをロードしたとき変わる可能性があります。しかし、すでに存在する変更禁止オブジェクトのソース コンテンツは更新されません。

### パラレルバージョンの確認

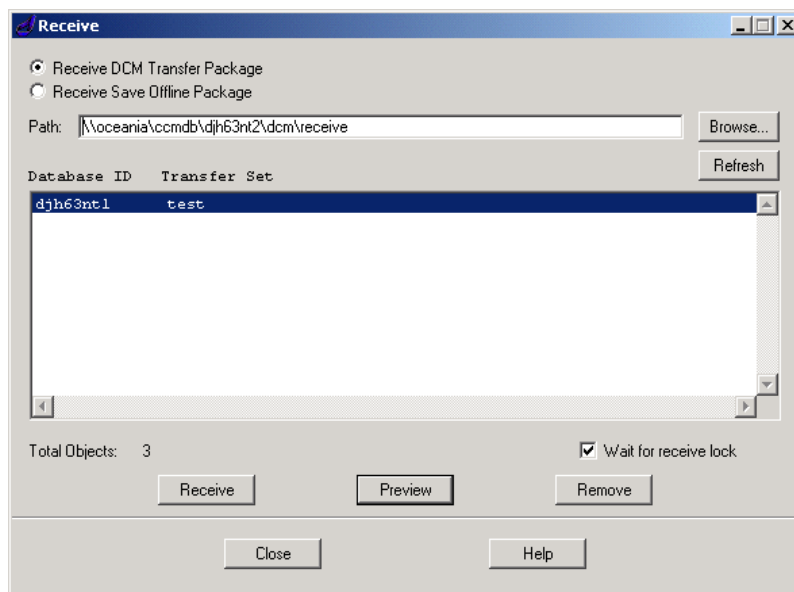
受取り操作時、DCM は新しく追加されたオブジェクトを調べ、デスティネーションデータベース内に同じ祖先オブジェクトを持つオブジェクトがあるか調べます。これは定義済み Entire database 転送セットを使用してパッケージを生成したとき、あるいはオブジェクトがパッケージの生成に使われた転送セットの履歴メンバーである場合にのみ行われます。デスティネーションデータベース内のオブジェクトバージョンは、デスティネーションデータベース内にパラレルバージョンが作成されているか、2 つ以上のデータベースからパラレルバージョンを受取っている場合に、受取り済みオブジェクトバージョンと同じ祖先バージョンを持つことができます。

---

転送セットで **Local Parallel Notification** を指定していると、各受取り済みオブジェクトに対して DCM はデスティネーションデータベース内に作成されたパラレルバージョンのリストを作成します。そのようなバージョンを所有または作成した各ユーザーは電子メールを受取ります。この通知は、受取り済みオブジェクトとそのパラレルバージョンを説明します。それらパラレルバージョンの 1 つを電子メールの受信者が所有しています。詳細については 149 ページの「パラレル検査」を参照してください。

## 転送パッケージの手作業受取り

いったん DCM を使用するデータベースの初期化を行うと、データベースは[転送パッケージ](#)を受取ることができます。GUI またはコマンドラインから手作業受取り操作を行うことができます。下図に Receive ダイアログを示します。



この操作は、Synergy Classic ヘルプの「[転送パッケージの受取り](#)」で説明しています。

# 6

## DCM データベースの管理

### はじめに

DCM アドミニストレータは、クラスタ内の複数のデータベースを監視できます。DCM アドミニストレータは、通常、データ転送の監視と分析、他アドミニストレータへのデータベース変更の通知、およびクラスタ内の1つまたは複数のデータベースの作成と更新を行う責任があります。

DCM アドミニストレータは、各種の操作を行います。たとえば、ftp 転送を受取るディレクトリを変更するには、アドミニストレータはそのディレクトリへ ftp 転送を行うすべてのソース データベースのデスティネーション データベース定義を更新する必要があります。さらに、新しいサイトが既存の DCM データベースのコピーを使用して開始したい場合、アドミニストレータはデータベースを複製し、その ID を変更する必要があります。

このセクションに含まれる以下の項目は、アドミニストレータがこれらの作業および他の作業を行うとき役立ちます。

- 62 ページの「他の DCM アドミニストレータとの通信」
- 63 ページの「アップグレード後の DCM 設定の変更」
- 64 ページの「データベース ID の変更」
- 67 ページの「DCM 区切り文字の変更」
- 68 ページの「DCM データベース定義の編集」
- 69 ページの「DCM データベース定義の削除」
- 69 ページの「転送セット パラメータの編集」
- 70 ページの「転送セットの削除」
- 70 ページの「転送セットへのオブジェクトの追加」
- 70 ページの「転送セットからオブジェクトを削除」
- 71 ページの「DCM データベースの複製」
- 74 ページの「データベースの全 DCM メンバーの転送」
- 76 ページの「「ミラー」 データベースの作成と管理」
- 77 ページの「DCM ディレクトリの管理」

## 他の DCM アドミニストレータとの通信

DCM アドミニストレータは、共有データに影響する問題をお互いに知らせる必要があります。たとえば、*prep* プロジェクトの更新プロパティが変更されたら、プロジェクトのビルド マネージャは DCM クラスタ内のすべてのビルド マネージャに通知する必要があります。以下に、この *prep* プロジェクトまたは同様なローカル *prep* プロジェクトを使用する DCM クラスタ内のすべてのビルド マネージャに知らせる必要がある変更の例を示します。

- いずれかのデータベースへの新しいタイプの追加 (転送パッケージからタイプ定義が除外されている場合)
- リリースまたはプラットフォーム値の変更
- プロジェクトの更新プロパティへのフォルダの追加 (プロセス ルールを使用していないか転送パッケージからプロセス ルールが除外されている場合)
- プロジェクトのベースラインの変更 (プロセス ルールを使用していないか転送パッケージからプロセス ルールが除外されている場合)
- プロセス ルールまたはフォルダ テンプレートの変更 (転送パッケージからテンプレートが除外されている場合)
- フォルダテンプレートを使用していない場合のフォルダのクエリ条件の変更

---

## アップグレード後の DCM 設定の変更

いったんデータベースをより新しいリリースへアップグレードすると、アップグレードしたデータベースへデータを送るすべてのデータベース定義の `CCM_HOME` フィールドを変更する必要があります。また、アップグレードしたデータベース内の DCM データベース定義の設定を調べ、それらのデータベースで使用されている Rational Synergy のリリースと互換性があることを確認する必要があります。

詳しい手順は、[IBM Rational Synergy Information Center](#) にある適切な『Rational Synergy アップグレードガイド』を参照してください。『アップグレードガイド』の「DCM クラスタのアップグレード」で、DCM 機能特有の情報について説明しています。詳細については、155 ページの「リリース 7.1 以前の Rational Synergy Distributed リリース間の転送」も参照してください。

## データベース ID の変更

データベースの ID は、DCM を使用するようデータベースを初期化した後変更できます。この変更は、旧データベース ID で作成したオブジェクトを新データベース ID に変換するために行うこともできます。

### オブジェクトの更新を伴わない ID の変更

この機能は、すでにクラスタにあるデータベースのどれかをコピーして DCM クラスタにデータベースを追加するとき便利です (71 ページの「DCM データベースの複製」を参照してください)。このオプションを使用すると、既存のオブジェクトはすべて以前の名前やその他のプロパティを保持します。変更後作成された新しいオブジェクトは、新しいデータベース ID を使用するため、別のデータベースで作成されたように見えます。

新しいデータベース ID は、DCM クラスタ内で固有であることが重要です。先へ進んだり新しいオブジェクトを作成したりする前にこれを確認してください。

データベース ID の変更は、そのデータベース内のプロジェクトのワークエリアパスに影響する可能性があることに注意してください。詳細については、153 ページの「ワークエリア サブディレクトリ テンプレートとパス」を参照してください。

この操作は、Synergy Classic ヘルプの「[DCM データベース ID の変更](#)」で説明しています。

### データベース ID の変更と影響を受けたオブジェクトの更新

この機能は、以前使用したデータベース ID を変更する必要があり、そのデータベース内に作成されたすべてのオブジェクトを更新して新しいデータベース ID を反映させる必要がある場合に便利です。

注意！データベースが既存の DCM クラスタのメンバーである場合、そのデータベース ID を変更するとクラスタ内の他のデータベースに悪影響を及ぼす可能性があります。

多数のオブジェクトが作成されているデータベースでは、更新を必要とするオブジェクトが多数ある可能性があるため、この操作は時間がかかる可能性があります。新しいデータベース ID は、DCM クラスタ内で固有であることが重要です。先へ進む前に、このことと、データベースの有効なバックアップがあることを確認してください。

データベース ID の変更は、そのデータベース内のプロジェクトのワークエリアパスに影響する可能性があることに注意してください。詳細については、153 ページの「ワークエリア サブディレクトリ テンプレートとパス」を参照してください。



---

クラスタのデータベースのどれかの ID を変更するときは、クラスタのすべての DCM アドミネレータに知らせる必要があります。DCM は、DCM アドミネレータに変更を自動的に通知しません。

データベース ID の変更とオブジェクトの更新を行う手順は以下のとおりです。

- すべての DCM アドミネレータにデータベース ID を変更することを通知し、変更を行うために適した時刻を調整する。
- ID を変更するデータベースとそのデータベースからオブジェクトを受取った他のデータベースですべての DCM 操作を一時停止する。
- 変更するデータベースで、Change Database ID and update affected objects オプションを使用する。
- ID が変わったデータベースからオブジェクトを受取った可能性のある DCM クラスタ内のすべてのデータベースでは、Convert the database ID of objects created in another database オプションを使用する。それぞれの場所のデータベースでこの操作を行うリモートアドミネレータと調整する必要があります。
- 他のすべてのデータベースで、データベースの DCM データベース定義のデータベース ID を変更する。それぞれの場所のデータベースでこの操作を行うリモートアドミネレータと調整する必要があります。
- すべての DCM アドミネレータから、変更を必要とするすべてのデータベースが無事更新されたことが確認できたら、すべてのアドミネレータに変更が済み、DCM 操作を再開できることを知らせる。
- ID が変更されたデータベースから DCM 操作を再開する。

この操作は、以降の DCM 複製に悪影響が出ないようにデータベース内のオブジェクトの変更時刻を保持します。

この操作は、Synergy Classic ヘルプの「[DCM データベース ID の変更](#)」で説明されています。

## 別のデータベースから受取ったオブジェクトのデータベース ID の変換

このオプションは、別のデータベースの ID が変更された後使用し、古いデータベース ID を参照しているすべてのオブジェクトを新しいデータベース ID を反映するよう変換します。データベース ID を変更する詳しい手順は、64 ページの「データベース ID の変更と影響を受けたオブジェクトの更新」を参照してください。この操作は、以降の DCM 複製に悪影響が出ないようにデータベース内のオブジェクトの変更時刻を保持します。

データベース ID の変更は、そのデータベース内のプロジェクトのワークエリアパスに影響する可能性があることに注意してください。詳細については、153 ページの「ワークエリア サブディレクトリ テンプレートとパス」を参照してください。

この操作は、Synergy Classic ヘルプの「DCM データベース ID の変更」で説明しています。

---

## DCM 区切り文字の変更

DCM 区切り文字は、DCM を使用するようデータベースを初期化した後で変更できます。この変更は、DCM クラスタ内のすべてのデータベースに対して行う必要があります。これには多数のオブジェクトの更新が必要になる可能性があるため、更新するデータベースやオブジェクトの数が少ない、クラスタを使用し始めた頃に行うべきです。大きなクラスタやデータベースでは、この操作は時間がかかるので念入りに計画を立てる必要があります。行う前に、クラスタ内の各データベースの最新のバックアップが利用可能なことを確認してください。

DCM 区切り文字の変更手順は以下のとおりです。

- すべての DCM アドミニストレータに DCM 区切り文字を変更することを通知し、変更を行うのに適した時刻を調整する。
- クラスタ内のすべての DCM 操作を一時停止する。
- クラスタ内の各データベースで **Change DCM Delimiter** オプションを使用する。それぞれの場所のデータベースでこの操作を行うリモートアドミニストレータと調整する必要があります。
- すべての DCM アドミニストレータから、変更を必要とするすべてのデータベースが無事更新されたことが確認できたら、すべてのアドミニストレータに変更が済み、DCM 操作を再開できることを知らせる。
- ユーザーのサイトで行っていた DCM 操作を開始する。

データベース ID の変更は、そのデータベース内のプロジェクトのワークエリアパスに影響する可能性があることに注意してください。詳細については、153 ページの「ワークエリア サブディレクトリ テンプレートとパス」を参照してください。

DCM 区切り文字の変更操作は、以降の DCM 複製に悪影響が出ないようデータベース内のオブジェクトの変更時刻を保持します。

## DCM データベース定義の編集

デスティネーションデータベースのユーザーアクセスは、ソースデータベース内の DCM データベース定義で管理されています。デスティネーションデータベースのプロパティが変更されると、ソースデータベース内でその DCM データベース定義を編集する必要があるかもしれません。

たとえば、前は手作業で転送しか行わなかったデータベースに対してリモートコピーができるようになったとします。そのデスティネーションデータベース定義を編集することにより、その転送モードを **Manual Copy** から **Remote Copy** へ変更できます (129 ページの「転送モード」を参照)。

もう 1 つの例として、たとえばデータベースが別のマシンへ移動されたり、ホスト名が変更されたりしたとします。このような場合、データの送信先となるすべてのデータベースのデスティネーションデータベース定義を変更する必要があります。

DCM データベース定義で以下のプロパティを変更できます。

- データベース ID
- データベース説明、ロケーション、管理者連絡先情報
- データベース ID の非表示または表示設定
- デスティネーションデータベースとして使用可能なデータベースの設定
- Handover allowed
- 受取りオプション
- 転送モード (と関連パラメータ)
- Zip 設定

注意！既存のデータベース定義の変更は、一般的にデスティネーションデータベースのアドミニストレータからの変更通知の結果です。データの整合性を確保するため、この人にデスティネーションデータベース定義の変更をすべて知らせてください。

この操作は、Synergy Classic ヘルプの「[DCM データベース定義の編集](#)」で説明しています。

---

## DCM データベース定義の削除

デスティネーション データベース定義が不要になったら削除できます。データベースが存在するがそのデータベース ID を各選択リストから除外したい場合には、それを非表示にできます。詳細については、68 ページの「DCM データベース定義の編集」を参照してください。

この操作は、Synergy Classic ヘルプの「DCM データベース定義の削除」で説明しています。

## 転送セット パラメータの編集

要求が変わるにしたがって、転送セット パラメータを変更したいことがあります。たとえば、特定の転送セットに製品を入れる必要がなくなったら、転送セットを変更して今後の転送からそれらの製品を除外できます。

転送セットの以下のプロパティを変更できます。

- 転送セット名
- 電子メール通知の電子メール アドレス
- 電子メール ポリシー
- 問題範囲
- 累積問題範囲
- [generate directory](#)
- リリース 範囲とリリース クエリ
- ローカル パラレル通知
- 関連ベースラインの取り入れ
- 除外オブジェクト

この操作は、Synergy Classic ヘルプの「[転送セット パラメータの編集](#)」で説明しています。

## 転送セットの削除

転送セットが不要になったら削除できます。

注意！いったん転送セットを削除すると、そのすべてのメンバーとの関連付けがなくなり、**Last Generated** 情報も削除されるので、転送パッケージの生成や送信はできなくなります。

この操作は、Synergy Classic ヘルプの「[転送セットの削除](#)」で説明しています。

## 転送セットへのオブジェクトの追加

転送セットを変更する 1 つの方法は、新しいオブジェクトを追加することです。この操作は、45 ページの「[転送セットへのオブジェクトの追加](#)」で説明しています。

## 転送セットからオブジェクトを削除

オブジェクトは、転送セットの[直接メンバー](#)である場合にのみ転送セットから削除できます。[間接メンバー](#)は、転送セットから明示的に削除できません。転送セットからプロジェクト、タスク、フォルダ、またはベースラインなどの直接メンバーを削除すると、その転送セットは再計算が必要と指定されます。この時点で間接メンバーは削除されません。後に転送セットの間接メンバーが再計算されると、転送セットの他で使用されていない限り間接メンバーが自動的に削除されます。転送セットの間接メンバーの再計算は明示的に行われるか、転送セットが次に DCM 生成に使われるとき自動的に行われます。

注意！[履歴付き](#)で追加したオブジェクトを削除すると、そのオブジェクトの全履歴が転送セットから削除されます。

この操作は、Synergy Classic ヘルプの「[転送セットからオブジェクトを削除](#)」で説明しています。

---

## DCM データベースの複製

既存の DCM データベースを複製して新しい DCM データベースを作成できます。これは、定義済みの Entire Database 転送セットを使用して複製したい場合には、DCM クラスタにデータベースを追加する簡単な方法です。

DCM データベースを複製する方法は 2 つあります。

- pack と unpack コマンドを使用したデータベースのコピー
- データベースのすべての DCM メンバーの転送

### Pack と Unpack を使用した DCM データベースのコピー

Rational Synergy の pack コマンドと unpack コマンドを使用して、既存の DCM データベースを複製できます。パックするデータベースは、DCM を使用するよう初期化されている必要があります。また、アクティブなサーバーと、データベースのアンパック先に十分なディスク領域が必要です。

データベースを Windows サーバーでパックして UNIX サーバーでアンパックすることはできません。

パック操作は、[generate directory](#) と [receive directory](#) を含みます。したがって、すべての未解決の転送と受取りは、データベースをパックする前に解決しておく必要があります。

**注意！** Rational Synergy データベースをアンパックまたはコピーしたとき、新しいデータベースのプロジェクトワークエリアパスは元のデータベースのものと同じです。衝突する可能性があるため、これらのパスは新しいデータベースを使用する前に変更する必要があります。

ワークエリアパスを変更する方法は、Synergy Classic ヘルプの「[ワークエリアプロパティの設定](#)」を参照してください。

pack と unpack コマンドを使用して新しい DCM データベースを作成する方法は以下のとおりです。

1. パックするデータベースで実行しているセッションがないことを確認します。
2. パックするデータベースを保護します（『[IBM Rational Synergy 管理者ガイド UNIX 版](#)』または『[IBM Rational Synergy 管理者ガイド Windows 版](#)』の ccm monitor と ccmdb protect コマンドの説明を参照）。

3. DCM データベースをバックします。  

```
> ccldb pack database_path /to packfile
```

Windows の場合、`database_path` は UNC パスである必要があります。`database_name.cpk` というバック ファイルが作成されます。
4. DCM データベースをアンパックして新しい名前を付けます。  

```
> ccldb unpack database_name.cpk /t  
new_database_path¥new_database_name /s server_name
```

Windows の場合、`new_database_path` は UNC パスである必要があります。
5. ワークエリア パスを変更します。
6. DCM データベース ID を変更します。
  - a. 新しいデータベースで `ccm_admin` としてセッションを開始します。
  - b. 新しいデータベースを保護します。
  - c. Change DCM Database ID or Delimiter ダイアログを開きます。
  - d. Change Database ID without updating any objects オプションを選択します。
  - e. データベース ID を DCM クラスタ内で固有の ID に変更します。OK をクリックします。
  - f. セッションを終了します。  
詳細については 64 ページの「データベース ID の変更」を参照してください。
7. 新しいデータベースへ転送パッケージを送るように元のデータベースを設定します。
  - a. 元のデータベースの保護を解除します（『IBM Rational Synergy 管理者ガイド UNIX 版』または『IBM Rational Synergy 管理者ガイド Windows 版』の `ccldb unprotect` コマンドの説明を参照）。
  - b. 元のデータベースでセッションを開始します。
  - c. 他の人がセッションを開始できないよう元のデータベースを保護します。
  - d. 新しいデータベースのデスティネーションデータベース定義を作成します。詳細については 43 ページの「DCM データベース定義の作成」を参照してください。



- 
- e. 元のデータベースから新しいデータベースへ送られるすべての転送パッケージに対して転送セットを作成してオブジェクトを追加します。詳細については 44 ページの「転送セットの作成」を参照してください。
  - f. 新しい転送セットを初期化します。
  - g. **Last Generated Time** を **Current Time** に設定して新しい転送セットにそれぞれ転送パッケージを生成します。これは不必要な転送を防ぎます（つまり、新しくアンパックしたデータベースのオブジェクトは、元のデータベースへ送られません）。その上、元のデータベースの保護を解除してから変更されたオブジェクトのみ新しいデータベースへ転送できます。
  - h. セッションを終了します。
8. 新しいデータベースの保護を解除します。
  9. 元のデータベースへ転送パッケージを送るよう新しいデータベースを設定します（オプション）。
    - a. 新しいデータベースでセッションを開始します。
    - b. 他の人がセッションを開始しないよう新しいデータベースを保護します。
    - c. 元のデータベースのデスティネーションデータベースを作成します。詳細については 43 ページの「DCM データベース定義の作成」を参照してください。
    - d. 新しいデータベースから元のデータベースへ送られるすべての転送パッケージに対して転送セットを作成してオブジェクトを追加します。詳細については 44 ページの「転送セットの作成」を参照してください。
    - e. 新しい転送セットを初期化します。
    - f. **Last Generated Time** を **Current Time** に設定して新しい転送セットにそれぞれ転送パッケージを生成します。これは不必要な転送を防ぎます（つまり、元のデータベースから得たオブジェクトは、元のデータベースへ送り返されません）。その上、新しいデータベースの保護を解除してから変更されたオブジェクトのみ元のデータベースへ転送できます。
    - g. セッションを終了します。
  10. 新しいデータベースの保護を解除します。
  11. 元のデータベースの保護を解除します。

サーバー コマンドの詳細については、『[IBM Rational Synergy 管理者ガイド UNIX 版](#)』または『[IBM Rational Synergy 管理者ガイド Windows 版](#)』を参照してください。

## データベースの全 DCM メンバーの転送

Entire database という組み込まれている転送セットを使用して、すべての適格オブジェクトを 1 つの DCM データベースから別の DCM データベースへコピーできます。デスティネーションデータベースは、新しくなくても、空でなくてもかまいません。DCM データベース内のすべての適格オブジェクトは、Entire Database 転送セットのメンバーです。したがって、これ以上オブジェクトを追加する必要はありません。

Entire Database 転送セットは、データベース内のすべてのオブジェクトを複製しないことに注意してください。作業状態のオブジェクトとデータベースの正常動作に不可欠なモデルに関連するオブジェクトは除外できます。したがって、この転送セットをデータベースの定期的バックアップの代わりに使用するべきではありません。DCM のみを使用してデータベースを完全にリストアすることはできません。

Entire Database 転送セットは、アプリケーションまたはコンポーネントを開発している小規模なチームが、複数の場所にまたがって分散開発を行う必要があるときの小さいデータベースに便利です。しかし、大きなデータベースや複雑な開発では、新しい静的オブジェクトバージョンがそれぞれ他のデータベースへ複製されるので、この転送セットを使用すると大量のデータが複製され、データベースが急速に肥大化します。また、しばらくソースデータベースを使用していると、最初の DCM 転送パッケージは非常に大きくなり、生成と受取りに時間がかかる可能性があります。

この手順を開始する前に以下のことを行います。

- ソースとデスティネーションデータベースを、DCM を使用するよう初期化する。
- デスティネーション データベースを、転送パッケージを受け付けるよう設定する（たとえば、デスティネーション データベースのセキュリティとホスト ファイルは、ユーザー `ccm_root` が転送できるように設定しておく必要がある）。

すべての適格オブジェクトをソース データベースからデスティネーション データベースへ転送するには以下のことを行います。

1. デスティネーション データベース定義を作成します。  
この操作は、Synergy Classic ヘルプの「[デスティネーション データベース定義の追加](#)」で説明しています。
2. 転送パッケージを生成します。  
デスティネーション データベースと Entire Database 転送セットを選択します。  
この操作は、Synergy Classic ヘルプの「[転送パッケージの生成と送信](#)」で説明しています。
3. 転送パッケージを受取ります。

---

この操作は、Synergy Classic ヘルプの「[転送セットの受取り](#)」で説明しています。

## 「ミラー」データベースの作成と管理

2 つの DCM データベースを同じデータで同期するには、まず別のデータベースから「ミラー」データベースを作成します。そして、それらの Entire Database 転送セットを相互に転送してデータベースを同期させます。

この操作を始める前に以下のことを行います。

- パックするデータベースが、DCM を使用するよう初期化されていることを確認する。
- データベースをパックする前にすべての未解決の転送と受取りを解決する。これは、パック操作にはそれぞれデフォルトの [generate\\_directory](#) と [receive\\_directory](#) に使用されている `database_path¥dcm¥generate` と `database_path¥dcm¥receive` パスが含まれているため、行う必要があります。
- アクティブなサーバーと、データベースのアンパック先に十分なディスク領域があることを確認する。

ミラー データベースを作成するには、以下のことを行います。

1. 71 ページの「Pack と Unpack を使用した DCM データベースのコピー」に従ってデータベースを設定します。ただし、2 つのステップを以下のように置き換えます。
  - a. [ステップ 7.f](#) の代わりに、次のように元のデータベースの転送セットを初期化します。

ミラー データベースの Entire Database 転送セットを使用し、Last Generated Time を Current Time に設定して生成を行います。したがって、元のデータベースの保護を解除してから変更されたオブジェクトのみ Entire Database 転送セットを使用してミラー データベースへ転送できます。
  - b. [ステップ 9.f](#) の代わりに、以下のようにミラー データベースの転送セットを初期化します。

元のデータベースの Entire Database 転送セットを使用し、Last Generated Time を Current Time に設定して生成を行います。したがって、ミラー データベースの保護を解除してから変更されたオブジェクトのみ Entire Database 転送セットを使用して元のデータベースへ転送できます。
2. 定期的にデータベースの Entire Database 転送セットを相互に転送してデータベースを同期します。

---

## DCM ディレクトリの管理

DCM は、一時ファイルを使用して転送パッケージ情報の送受信を行います。27 ページの「ソース データベース」と 28 ページの「デスティネーション データベース」で説明しているように、これらのファイルは [generate\\_directory](#) と [receive\\_directory](#) に格納されています。

DCM は、データベースで最初に生成操作が行われるとき [generate\\_directory](#) を作成します。DCM は、データベースが転送を受取れるようデータベースの DCM 初期化時に [receive\\_directory](#) を作成します。通常の操作では、DCM はこれらのディレクトリを転送パッケージの一時的な作成と抽出に使用します。DCM は操作を完了すると、一時ファイルを削除します。時折、転送パッケージが完全に削除されないことがあります。したがって、[generate\\_directory](#) と [receive\\_directory](#) を定期的に調べ、不要なファイルを削除してください。

注意！ディレクトリ

`database_path\%dcm%\receive\%history`

とその内容は削除しないでください。DCM は、143 ページの「転送セットの紛失」で説明しているように、このデータを使用して不足している転送セットを確認します。

DCM が [generate\\_directory](#) と [receive\\_directory](#) に格納するファイルの詳細については、126 ページの「転送パッケージの構造」を参照してください。



# 7

## サンプル DCM シナリオ マスタとサテライト

### はじめに

ここでは、サンプル DCM シナリオを説明します。ユーザーが使用する手法が異なる場合でも、この例はガイドラインとして役立ちます。ここに示す情報は、DCM の導入時に考慮すべき項目のチェックリストに使用してください。以下のトピックについて説明します。

- 80 ページの「前提条件」
- 81 ページの「DCM 手法の確立」
- 83 ページの「データベースの初期化」
- 84 ページの「マスタの設定」
- 86 ページの「サテライトへの初期転送の実行」
- 87 ページの「サテライトの設定」
- 89 ページの「定期的なサテライトからマスタへの転送の実行」
- 90 ページの「定期的なマスタからサテライトへの転送の実行」
- 91 ページの「ソフトウェア開発を続ける」
- 92 ページの「ベースラインについて」
- 93 ページの「サテライトの追加」
- 94 ページの「まとめ」

## 前提条件

このシナリオの手順を行うには、以下の要件を満足する必要があります。

- 転送の送信先マシンはそれぞれ、リモート コマンドを実行できるよう設定されている。たとえば、UNIX では、送信先マシンは、その `hosts.equiv` ファイルまたは `ccm_root` の `.rhosts` ファイルに送信マシンのエントリを持つ必要があり、Windows では、送信先マシンは `ccm_remd` を実行している必要があります。
- DCM クラスタ内のすべてデータベースは、互換モデルを持っている。
- タスクベース CM の実用的な知識を持っている。
- `build_mgr`、`dcm_mgr`、および `ccm_admin` ロールを持っている。詳細については 19 ページの「ロール」を参照してください。



---

## DCM 手法の確立

このシナリオには 2 つの個別のデータベースがあります。1 つはマセチューセッツ州 (MA) のボストンにあり、もう 1 つはイリノイ州 (IL) のシカゴにあります。この 2 つのサイトは、並行して「Hello world」と表示するソフトウェアアプリケーションのバージョン 2.0 を開発します。これは、リリース値 hello/2.0 を使用します。このアプリケーションの最初のリリースは、ボストンのサイトで開発されリリースされました。このリリース済みベースラインは hello/1.0 のリリースを使用して作成され、これには hello-1.0 のメンバープロジェクトがあります。

各サイトがアプリケーションの一部を開発し、独自の統合 *prep* プロジェクトを管理します。マスタとサテライト手法を使用します。MA がマスタで、マスタの統合とリリース済みプロジェクトを管理します。IL がサテライトです。この手法の詳細については、24 ページの「マスタとサテライト手法」を参照してください。

以下を前提とします。

- 両サイトは、同じかほぼ同じモデルを使用する。
- 両サイトは、デフォルトの DCM 区切り文字 ("#") を使用する。
- 両サイトは、デフォルトのリリース区切り文字 ("/") を使用する。
- 両サイトは、タスクベース CM を使用する。
- 両サイトは、分散型プロセスを使用してリリース値 hello/2.0 を使用する。
- 両サイトは、自動受取りを使用し、セキュリティは各サイトで自動受取りをサポートするように設定されている。
- DCM は、転送パッケージを生成するか受取るときボストンとシカゴ DCM マネージャに電子メール通知を送る。
- 両サイトは、リモート コピー コマンドを使用してデータを転送できるワイドエリア ネットワークで接続されている。

ボストンとシカゴに 1 人ずつ、合計 2 人の DCM マネージャがいます。各マネージャは、両方のデータベースに関連した責任があります。各サイトのビルド マネージャもこのシナリオにかかわります。

### マスタの責任

ボストンのビルド マネージャには、以下の責任があります。

- マスタ統合プロジェクトの確立
- マスタ プロジェクトのビルド
- ボストンのローカル統合プロジェクトの確立
- ローカル統合プロジェクトのビルド

ボストンの DCM マネージャには、以下の責任があります。

- 最初の転送の開始
- サテライト データベースからのデータの収集
- 変更のサテライトへの分散

### サテライトの責任

シカゴのビルド マネージャは、以下の責任があります。

- シカゴのローカル統合プロジェクトの確立
- シカゴのローカル統合プロジェクトのビルド

シカゴの DCM マネージャには、以下の責任があります。

- シカゴの変更のマスタへの送り返し

このサンプル シナリオでは以下のトピックについて説明します。

1. データベースの初期化
2. マスタの設定
3. サテライトへの初期転送の実行
4. サテライトの設定
5. 定期的サテライトツーマスタ転送の実行
6. 定期的マスタツーステライト転送の実行
7. ソフトウェア開発の継続
8. サテライトの追加

---

## データベースの初期化

このシナリオでは、データベースはまだ DCM を使用するように初期化されていません。データベースを初期化するには以下のことを行います。

1. MA データベースで、リリース値 `hello/1.0` が定義されていて、DCM 転送が許可されていることを確認します。これらのリリースは、IL データベースでは定義しないでください。
2. MA データベースで、`hello/1.0` リリースのリリース済みベースラインが作成されていて `hello-1.0` プロジェクトを含むことを確認します。
3. ボストン データベースを MA に初期化します。

**DCM Initialization** ダイアログを使用してデータベース ID を MA に設定し、[DCM 区切り文字](#)を「#」に設定します。

4. MA データベースで、**Rational Synergy** を使用して (**Synergy Classic** ではなく)、新リリースに分散型プロセスを選択して `hello/1.0` リリースを新しい `hello/2.0` リリースへコピーします。`hello/2.0` リリースが DCM 転送を許可されていることを確認します。データベースが DCM 用に初期化される前にすでにリリース `hello/2.0` が定義されている場合、**Rational Synergy** を使用して (**Synergy Classic** ではなく)、リリース `hello/2.0` のプロパティを表示し、プロセスルールタブで追加ボタンをクリックして、分散型プロセスから汎用プロセスルールを追加します。
5. シカゴデータベースを IL に初期化します。

**DCM Initialization** ダイアログを使用してデータベース ID を IL に設定し、DCM 区切り文字を「#」に設定します。

## マスタの設定

MA データベースがマスタです。このデータベースを以下のように設定します。

注記：以下の手順は、プロジェクトがデフォルトでプロセスルール（更新またはリコンフィギュア テンプレート）を使用することを前提にしています。プロセスルールを使用しない場合、適切なフォルダを手動で作成して追加する必要があります。

1. マスタ統合プロジェクトを作成します。

マスタ統合プロジェクトは、2つのデータベースの完了タスクの統合ビルドを結合するために必要です。マスタ統合 *prep* プロジェクトを `hello-1.0` からチェックアウトします。バージョンを `MasterInt` に設定し、**Check Out Project** ダイアログの **Purpose** リストから **Master Integration** を選択します。リリース値を `hello/2.0` に設定します。新しいプロジェクトは、`hello-MasterInt` です。

新しいプロジェクトの更新プロパティは、プロセスルールを使用して自動的に設定されます。プロセスルールの詳細については、『[ビルドマネージャガイド](#)』を参照してください。

2. MA のローカル統合プロジェクトを作成します。

MA のローカル統合ビルドに 1つのローカル統合テストプロジェクトが必要です。バージョンを `MAInt` に設定し、**Check Out Project** ダイアログの **Purpose** リストから **Integration Testing** を選択して `hello-1.0` プロジェクトから MA 統合 *prep* プロジェクトをチェックアウトします。リリース値を `hello/2.0` に設定します。新しいプロジェクトは、`hello-MAInt` です。

新しいプロジェクトの更新プロパティは、プロセスルールを使用して自動的に設定されます。

3. IL の DCM データベース定義を作成します。

**Create DCM database Definition** ダイアログを使用してデスティネーションデータベース ID (IL)、その説明、およびそのプロパティを追加します。転送モードに **Remote Copy** を選択し、**Automatic Receive** チェックボックスを選択します。

4. マスタからサテライトへの転送セットを作成します。

- a. 転送セットのプロパティを設定します。

**Create Transfer Set** ダイアログを使用して転送セットに名前を付け (**Master Transfer of hello Project**)、生成と受取り通知の電子メールアドレスを設定します。**Release Scopes** が **Releases and Templates** に設定されていることを確認します。転送セットから DCM データベース定義を除外していないことを確認します。

- 
- b. **Admin > Browse Baselines** を選択して **Browse Baselines** ダイアログを開きます。**DCM Properties** をクリックし、転送セットに以前のリリースのリリース済みベースライン hello/1.0 を追加します。これには自動的に hello-1.0 プロジェクトとそのメンバーが含まれます。
  - c. 転送セットにフォルダを追加します。

**Tools > Folder > Browse** を選択して **Browse Folders** ダイアログを開きます。**DCM Properties** をクリックし、転送セットに **All completed tasks for release hello/2.0** フォルダと **Master Integration Tested Tasks for hello/2.0** フォルダを追加します。これは自動的に全データベースのすべての完了済み変更と hello/2.0 のマスタ統合テスト済みタスクを含みます。

転送セットにマスタプロジェクトを追加する必要はありません。すべてのタスクと関連するソース変更は、転送セットのメンバーであるフォルダに複製されます。

## サテライトへの初期転送の実行

サテライトに統合テストプロジェクトを作成する前に、マスタの `hello` プロジェクトと関連データをサテライトへ送る必要があります。これは、各サテライトに対して行う最初の転送です。

注記：この操作には、`dcm_mgr` ロールが必要です。DCM は、この操作および他のすべての DCM 操作で [ダイナミックロールスイッチング](#) を行います。

以下のように **Generate** ダイアログを使用して、初期転送パッケージの生成、送信、および受取りを行います。

1. 生成オプションを選択します。

IL デスティネーション データベースと Master Transfer of hello Project 転送セットを選択します。

2. **Preview** ボタンをクリックして転送リストに正しいオブジェクトがあることを確認します (オプション)。
3. 転送パッケージを生成します。

この操作は、転送パッケージの自動受取りを生成、転送、および開始します。

生成と送信操作が完了すると、IL は自動的に転送セットを受取ります。すると IL は、ベースラインのコピー、`hello-1.0` プロジェクトのコピー、`All completed tasks for release hello/2.0` フォルダと `All completed tasks for hello/2.0 from database MA` フォルダのコピーを持ちます。

`hello/2.0` リリースのプロセス ルール、および `hello/1.0` と `hello/2.0` のリリース定義は転送パッケージに含まれているので、IL データベースはこれらも持ちます。このことは、**Browse Reconfigure Templates** および **Browse Releases** ダイアログを使用して確認できます。

IL データベースには、MA のスケルトン DCM データベース定義も設定されます。これは、**Browse DCM Database Definitions** ダイアログを使用して確認できます。

---

## サテライトの設定

IL は、このシナリオではサテライトです。これは、マスタの設定とサテライトへの初期転送を行った後で設定します。

サテライトの設定は以下のように行います。

1. IL のローカル統合 *prep* プロジェクトを作成します。

IL のローカル統合ビルドに 1 つの *prep* プロジェクトが必要です。 *prep* プロジェクトを `hello-1.0` プロジェクトからチェックアウトします。バージョンを `IL#Int` に設定し、**Check Out Project** ダイアログの **Purpose** リストから **Integration Testing** を選択します。新しいプロジェクトは、`hello-IL#Int:project:MA#1` です。

2. サテライトの MA 用 DCM データベース定義を編集します。

MA 用 DCM データベース定義は、DCM データベース定義が転送パッケージから除外されていないので、すでに DCM 受取りによって作成されているはずです。 **Edit Destination database** ダイアログを使用して残りのプロパティを設定します。 **Generate Allowed** を選択し、転送モードに **Remote Copy** を選択し、**Automatic Receive** を選択します。

3. サテライトからマスタへの転送セットを作成します（この転送セットにはサテライトデータベースで完了したタスクがあります）。

注記：この操作には、*dcm\_mgr* ロールが必要です。DCM は、この操作および他のすべての DCM 操作で [ダイナミックロールスイッチング](#) を行います。

- a. 転送セットのプロパティを設定します。

**Create transfer set** ダイアログを使用して転送セットに名前を付け（`IL Transfer of Completed Tasks for hello Project`）、生成と受取り通知の電子メールアドレスを設定します。

- b. データベースにローカルではないオブジェクトを除外します。

非ローカルオブジェクトを除外することにより、転送パッケージは IL がマスタから受取ったオブジェクトを含みません。 **Create Transfer Set** ダイアログから **Exclude** を選択します。 **Exclude Imported Objects** を選択します。 **OK** をクリックします。

- c. サテライトのフォルダを追加します。

Rational Synergy Classic の **Project View** の **Tools** メニューから、**Folder > Browse** を選択して **Browse Folder** ダイアログを開きます。 **All completed tasks for Release hello/2.0 from Database IL** フォルダを探して選択し、**DCM Properties** ボタンをクリックします。 **Show DCM Properties** ダイアログを使用してフォルダを転送セットに追加します。

## サテライトの転送セットを初期化

マスタへは、マスタからサテライトへの最初の転送が受取られてから変更されたオブジェクトのみを送る必要があります。つまり、マスタがサテライトへ最初に送った「シード」オブジェクトを送る必要はありません（86 ページの「サテライトへの初期転送の実行」を参照）。したがって、**Last Generated** リストで **Current Time** を選択します。次にサテライトで転送セットを生成するときは、最後の生成時刻以降変更されたオブジェクトのみがその転送セットに入れられます。

サテライトの転送セットを「初期化」するには、**Generate** ダイアログから生成操作を行います。

1. 生成オプションを選択します。

デスティネーション データベースに **MA** を選択し、**IL Transfer of Completed Tasks for hello Project** 転送セットを選択します。

2. 転送パッケージの生成時刻を選択します。

**Last Generated** を **Current Time** に設定します。これによって空の転送セットが作成され、初期生成時刻が設定されます。次の転送パッケージはこの時刻をもとにして作成されます。

3. サテライトデータベースにローカルではないオブジェクトを除外します。

**Exclude Imported Objects** を選択します。

4. **Preview** をクリックし、転送リストにオブジェクトがないことを確認します（オプション）。

5. 転送パッケージを生成します。

この操作は空の転送パッケージを作成し、データはマスタへ転送されません。



---

## 定期的なサテライトからマスタへの転送の実行

ILがマスタから初期転送を受取った後、開発者は転送済みプロジェクトの個別開発作業中バージョンをチェックアウトして通常の開発作業を行うことができます。

ビルドマネージャは、ローカル統合テストプロジェクトを更新してコンフリクトの検出を行います。使用可能な製品が正しくビルドされたら、ビルドマネージャはリリース hello/2.0 の新しい統合テストベースラインを作成します。これにより、開発者はこれらのテスト済み変更を新しいベースラインとして使用できます。開発者のプロジェクトは、更新されるとき自動的にこの最新のベースラインを使用します。

統合テスト *prep* プロジェクトの再ビルドとテスト後、DCM マネージャは、IL Transfer of Completed Tasks for hello Project 転送セットを使用して、完了タスクをマスタへ送ることができます。この転送セットには、All completed tasks for Release hello/2.0 from Database IL フォルダのタスクがあります。

注記：この操作には、*dcm\_mgr* ロールが必要です。DCM は、この操作および他のすべての DCM 操作で [ダイナミックロールスイッチング](#) を行います。

**Generate** ダイアログを使用して、転送パッケージの生成、送信、および受取りを行います。

1. 生成オプションを選択します。

MA デスティネーションデータベースと IL Transfer of Completed Tasks for hello Project 転送セットを選択します。

2. **Preview** ボタンをクリックして、転送リストに正しいオブジェクトがあることを確認します (オプション)。
3. 転送パッケージを生成します。

この操作は、転送パッケージの自動受取りを生成、転送、および開始します。

## 定期的なマスタからサテライトへの転送の実行

MA がサテライトから転送を受取った後、マスタ プロジェクト hello-MasterInt を更新する必要があります。この操作の最中、Rational Synergy は All Completed Tasks for Release hello/2.0 from Database MA フォルダを見直します。ローカル完了タスクのみならず、サテライトから転送されたすべての完了タスクもフォルダのクエリによってフォルダに追加されます。

そして、All Completed Tasks for Release hello/2.0 フォルダからすべての完了タスクをフォルダへコピーすることで hello-MasterInt のテストを完了し、Master Integration Tested Tasks for Release hello/2.0 フォルダを更新します。

この後、DCM マネージャは Master Transfer of hello Project 転送セットを使用して、更新されたマスタ統合テスト済みタスク フォルダをサテライトへ送ります。

注記：この操作には、`dcm_mgr` ロールが必要です。DCM は、この操作および他のすべての DCM 操作で [ダイナミックロールスイッチング](#) を行います。

複数のサテライトへデータを送る場合、各サテライトに対して転送パッケージを再生成します。その理由は、転送パッケージをデータベースへ送る前に再生成すると、DCM は転送パッケージが最後にデータベースへ送られた後に変更されたか転送セットの新しいメンバーになったオブジェクトのみ生成して送るからです。

更新した Master Integration Tested Tasks for Release hello/2.0 フォルダを定期的に各サテライトへ送ります。これは、通常、マスタ統合テストが成功するたびに行います。

Generate ダイアログを使用して、各転送パッケージまたは各データベースの生成、送信、および受け取りを行います。

1. 生成オプションを選択します。

IL デスティネーションデータベースと Master Transfer of hello Project 転送セットを選択します。

2. Preview ボタンをクリックして、転送リストに正しいオブジェクトがあることを確認します (オプション)。

3. 転送パッケージを生成します。

この操作は、転送パッケージの自動受取りを生成、転送、および開始します。

生成と送信操作が終了すると IL は自動的に転送セットを受取ります。この時点で IL はマスタの最新の統合テスト済みタスクを持ち、それらを hello-IL#Int:project:MA#1 prep プロジェクトに構成できます。

---

## ソフトウェア開発を続ける

サテライトデータベースに個別開発作業中プロジェクトを作成するには、開発者はリリース値 `hello/2.0` と目的 `Insulated Development` を使用して `hello-1.0:project:MA#1` からチェックアウトします。更新プロパティは、MA マスタ データベースから受取ったプロセスルール `hello/2.0:Insulated Development` を使用して自動的に設定されます。

ビルド マネージャは、定期的に `hello-IL#Int:project:MA#1` 統合テスト *prep* プロジェクトを更新してビルドします (たとえば、毎晩)。更新 (リコンフィギュア) 操作は、開発者の最新のチェックインオブジェクトとタスクを集め、`Master Integration Tested Tasks for Release hello/2.0` フォルダを介してマスタ統合テスト済みタスクも含めます。

ビルド マネージャは、統合テストプロジェクトをビルドし直してテストを無事終了した後で、リリース `hello/2.0` および目的 `Integration Testing` の新しいベースラインを作成します。すると DCM マネージャは、`IL Transfer of Completed Tasks for hello Project` 転送セットを使用してサテライト データベースで完成したタスクをマスタ データベースへ転送します。

## ベースラインについて

DCM でベースラインを複製することはできますが、フォルダを転送して各データベースに独自のローカルベースラインを作成させるのが、最も一般的な DCM 手法です。フォルダは、ベースラインより軽いので、DCM 転送はフォルダを使用した方が速くなります。

ベースラインを転送する場合、サテライトデータベースのプロジェクトは、それがどのデータベースのものかに関わらず、最新のベースラインをもとにメンバーを更新することに注意してください。したがって、リモートデータベースから、自分のローカルタスクを含まない、より新しいベースラインを受取らないよう注意する必要があります。

ベースラインを転送する場合、マスタデータベースにすべてのベースラインを作成し、サテライトデータベースのプロセスルールを最新のベースラインとすべてのローカル完了タスクをベースにするよう構成するのが最善の方法です。

---

## サテライトの追加

データベースは、新しいサテライトを作成してマスタとサテライト DCM クラスタに追加できます。たとえば、CA ID を使用して Irvine、California に新しいデータベースを追加できます。このサテライトを既存の DCM クラスタに追加するには、マスタのデスティネーションデータベース定義と最初のサテライトを設定したときと同じ手順を使用します。

1. Irvine データベースを CA に初期化します。

詳細については 83 ページの「データベースの初期化」を参照してください。

2. CA のマスタのデスティネーションデータベース定義を作成します。

MA の Create DCM database Definition ダイアログで、デスティネーションデータベース ID (CA)、その説明、およびそのプロパティを追加します。

3. CA サテライトへの初期転送を行います。

IL データベースに使用したのと同じ転送セットを使用し、新しいデスティネーションの転送パッケージを生成します。

詳細については 86 ページの「サテライトへの初期転送の実行」を参照してください。

## まとめ

このセクションを終了すると、DCM データベースの初期化、データベース間の転送の設定、およびマスタとサテライト手法を使用した定期更新を行う方法がわかります。この例で使用した方法に限定する必要はありません。各種の手法を組み合わせ、必要に応じて転送セットを作成できます。

注記 : DCM では、コマンドラインから転送パッケージの生成、転送および受取りを行うこともできます。したがって、DCM 操作は、UNIX または Windows のスクリプトとスケジュール機能を使用して自動化できます。

# 8

## 分散型変更依頼の追跡

### はじめに

ここでは、分散型構成管理および変更管理ツールを使用して変更依頼を管理する方法を説明します。以下のトピックについて説明します。

- 96 ページの「非分散型変更依頼追跡の概要」
- 98 ページの「分散型変更依頼追跡の概要」
- 101 ページの「分散型変更依頼手法の計画」
- 102 ページの「他の分散型変更依頼手法の概要」
- 107 ページの「DCM 複製トポロジのタイプ」
- 108 ページの「Rational Synergy Distributed と Rational Change Distributed (DCS) の概要」
- 111 ページの「Rational Change ダイアログと用途の概要」
- 112 ページの「Rational Change Distributed モデル パラメータ」

## 非分散型変更依頼追跡の概要

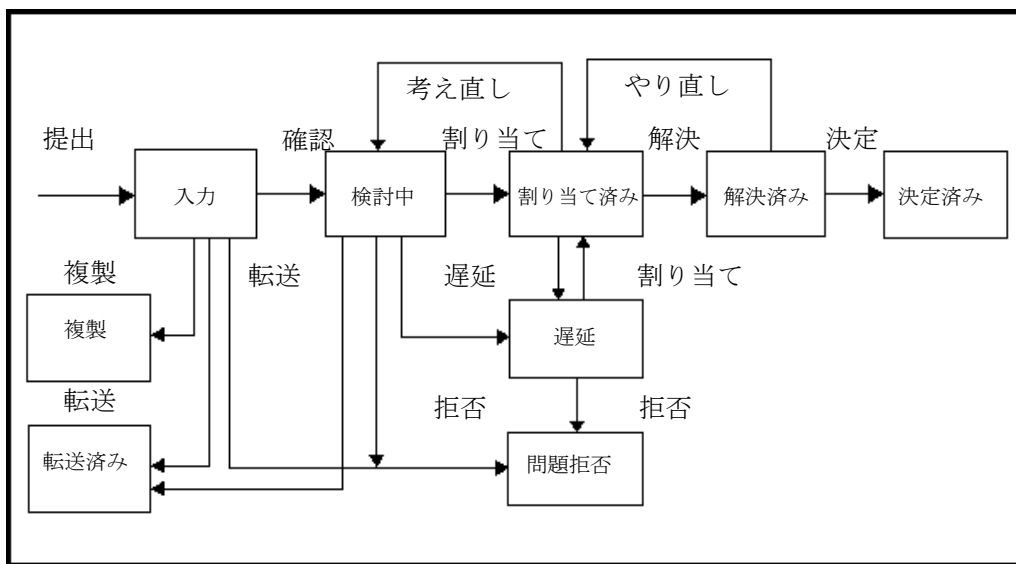
ここでは、非分散型変更依頼ツールである IBM® Rational® Change の概要を説明します。

### Rational Change

Rational Change は、変更依頼の提出と追跡を行うブラウザをベースにしたウェブ型アプリケーションです。このツールは、Rational Synergy Distributed に統合されており、Synergy Classic の Tools メニューからアクセスします。Rational Change を使用すると、以下の作業が実行できます。

- 変更依頼の提出
- 変更依頼の追跡
- レポートの実行

Rational Change は、変更依頼という特別のオブジェクトを使用してユーザーが提出した問題や懸案事項を把握します。変更依頼オブジェクトは、具体的な変更依頼を記述します。その属性には、要求が行われた理由と時刻、およびそれを提出した人などがあります。下図は、Rational Change のライフサイクルのサンプルです。ここでは、変更依頼の遷移と状態の例を示しています。





---

Rational Change は、変更依頼データを複製する方法は提供しません。  
Rational Change の詳細については、[IBM Rational Information Center](#) を参照してください。

## 分散型変更依頼追跡の概要

ここでは、分散型変更依頼追跡パッケージである Distributed Rational Change (DCS) の概要を説明します。

### サポートしている機能

DCS は、以下のように Rational Synergy Distributed および Rational Change の機能を拡張します。

- 任意の状態の変更依頼およびタスクの複製が可能。
- 特定の変更依頼またはタスクは、DCM クラスタ内では 1 時点で 1 つのデータベースでのみ修正または遷移が可能。これは、`modifiable_in` 属性とセキュリティルールで管理されます。
- 特定の変更依頼またはタスクを修正または昇格させる管理データベースの許可は、DCM クラスタ内の別のデータベースへ渡すことが可能。
- 変更依頼およびタスク担当者は、Rational Change ファイル内にリストされる外部データベース ユーザーを含むリストから選択可能。

- 
- 転送セットは、変更依頼クエリを含む変更依頼適用範囲の定義が可能。変更依頼適用範囲は、変更依頼を含めるべきか、含める場合は、変更依頼が関連タスクを含むか、そして、含む場合は、各関連タスクの関連オブジェクトを定義します。
  - 転送セットの変更依頼適用範囲と変更依頼クエリは、GUI または CLI から定義することが可能。
  - 転送セットは、変更依頼適用範囲が累積されるか否かを定義することが可能。これが無効の場合、転送セットの間接変更依頼メンバーは変更依頼クエリと適用範囲で正確に定義され、クエリに一致しないメンバーは削除されます。有効の場合、適用範囲クエリは既存の間接変更依頼メンバーに追加し、すべての既存変更依頼メンバーは保持されます。このオプションは、一般的に変更依頼が選択的に複製されたとき、特に中間ハブ データベースを使用して複製（特定の変更依頼クエリを使用して）されたときに有効になります。これにより、一度データベースへ複製したオブジェクトが、すでに転送セットに定義されている変更依頼クエリ定義で見つからない場合でも更新し続けることができます。
  - 開発者に割り当てられ、現在のデータベースで変更可能なタスクのみ開発者のカレントタスクとして、あるいはオブジェクトのチェックアウトまたはチェックインに使用することが可能。そのようなタスクは、別のデータベースで作成して割り当ててから現在のデータベースへ複製されている可能性があります。
  - DCM 用に初期化されたデータベースでタスクを選択すると、それは `dbid#nnnn` 形式で表示される。ここで `dbid` は [データベース ID](#)、`#` は [DCM 区切り文字](#)、そして `nnnn` はタスク番号。このデータは、Rational Synergy タスク ダイアログと Rational Change の特定のフィールドに表示されます。

## DCS の構成

DCS は、以下に説明するように構成できます。

- 転送セットは、変更禁止タスクを除外できます。デフォルトで、任意の状態のタスクを転送パッケージに入れることができます。必要に応じて、転送セット定義はすべての未完了タスクを除外できます。
- 転送セットは、インポートしたオブジェクトを除外できます。デフォルトで、転送セットはデータベースが他のデータベースから受取ったオブジェクトを含む転送パッケージを生成することを可能にします。これにより、そのようなデータベースの連鎖にわたって複製が可能です。このオプションが無効な場合、オブジェクトが生成されたデータベースは常にオブジェクトの送信先データベースへ複製を行う必要があります。
- DCM モデル パラメータにより、DCS の振る舞いを複雑な DCS 手法に合わせることができます。これらの手法は、101 ページの「分散型変更依頼手法の計画」で説明しています。

---

## 分散型変更依頼手法の計画

DCS を効果的に使用するためには、データベース間の情報の流れを計画する必要があります。この計画は、**Rational Change** ライフサイクル（状態間の遷移）と、対応する DCM 変更管理フロー（1 つのデータベースから別のデータベースへの管理の受け渡し）を扱う必要があります。

まず以下の質問を考えます。答えが DCS 手法とトポロジの決定に役立ちます。101 ページの「分散型変更依頼手法の計画」にサンプル手法を示し、107 ページの「DCM 複製トポロジのタイプ」でサンプルトポロジを説明します。

- 変更依頼はどのデータベースに入れられるか。
- 変更依頼は確認され、入力されたのと同じデータベースで遷移されるか。
- 変更依頼はリモート データベース内の人員に割り当てられるか。
- 変更依頼のタスクはその変更依頼の 1 つの管理データベースから作成され割り当てられるか。
- タスクは同じ変更依頼に対して複数のデータベースで作成され割り当てられるか。
- 変更依頼はどのデータベースで解決するか。
- 変更依頼はどのデータベースで決定するか。
- レポートおよびクエリ要件は、他のデータベースから変更依頼が見えることを要求するか。そうならば、これらの要件は関連タスクの内訳を含むか。
- 関連オブジェクトのレポートは必要か。
- 変更依頼の複製が必要な場合、それは選択的である必要があるか。そうならば、それは製品名をベースにしますか、あるいは他の条件をベースにするか。
- *registered* または *task\_assigned* タスクの複製は必要か。あるいは、*completed* 状態のタスクの複製で十分か。
- リモートデータベース内の構成管理データの営業上の機密に関する問題があるか。それにはどのように対処するか。
- どのような電子メールトリガが必要か。どのように各種のデータベースに対する電子メール通知を行うか。

## 他の分散型変更依頼手法の概要

ここでは、DCS を使用して変更依頼を管理するさまざまな方法を説明します。ここで示すものがすべてではありません。ユーザーの手法はここに示す例とは異なる可能性があります。

### サンプル手法 1 — 作成データベースでのみ変更依頼を管理

もっとも簡単な手法は、変更依頼とタスクを作成したのと同じデータベース内でそれぞれ遷移させることです。そうすると、DCM を使用して変更依頼および/またはタスクをレポートで使用するために他のデータベースへ複製できます。この計画は **Rational Change** ワークフローを使用します。DCM ワークフローは不要です。DCM 複製は、変更依頼および/またはタスク データを読み取り専用クエリおよび/またはレポートで使用することを可能にします。したがって、デフォルトの DCS モデル パラメータを使用します。複製は、一方向の公開と登録手法（詳細については、25 ページの「公開と登録による転送手法」を参照）としてまとめることができます。

### サンプル手法 2 — すべての変更依頼を 1 つのデータベースに入力

別の手法は、すべての変更依頼を 1 つのデータベースに入れることです。これは、企業がいくつかのアプリケーションの変更を管理するメインの製品管理データベースかもしれません。以下にこの手法の典型的な **Rational Change** および DCM ワークフローを示します。

1. 変更依頼がメイン製品管理データベースに入力されます。
2. 変更依頼はそのデータベース内で確認されるか、拒否されるか、複製としてマークされます。
3. 変更依頼は製品開発チーム リーダーに割り当てられ、対応する開発データベースへ渡されます。
4. すべての関連タスクは、1 つの開発データベースに入力され割り当てられます。
5. すべてのタスクを終了すると、製品開発チーム リーダーは変更依頼を解決してメイン製品管理データベースへ渡します。
6. メイン製品管理データベース内の決定者は、変更依頼を決定するか、開発データベースでやり直すために送り返します。

タスクの開発データベースからメイン製品管理データベースへの複製も必要になることがあります。これによりレポートを作成し、製品マネージャにタスクの進行状況を知らせることができます。別のケースでは、*completed* タスクの詳細で十分で、転送セットを未完了タスクから除外できることもあります。さらに別のケースでは、タスクを開発データベースからメイン製品管理データベースへ転送する必要がないかもしれません。

---

選択的複製が必要な場合、変更依頼クエリは通常、製品名または／および製品サブシステム（あるいは変更依頼の状態）をベースにしています。このような場合、転送セットはインポートしたオブジェクトを除外すべきではありません。除外すると、インポートしたオブジェクトがメイン製品管理データベースに戻されることがありません。

#### 特殊なケース

サンプル手法 2 を使用する場合、ときとして特別の処理が必要になります。以下にそのようなケースを説明します。

**Rational Change** を使用して、割り当て済み変更依頼を延期する場合、関連する割り当て済みタスクもすべて延期されます。変更依頼を再び割り当てると、関連する延期タスクももう一度割り当てられます。

上記のシナリオでは、割り当て済みで、開発データベース内で処理している変更依頼はチームリーダーが延期してメイン製品管理データベースへ戻すことができます。後にその変更依頼を再度割り当てることができます。しかし、関連タスクはメイン製品管理データベースにない可能性があります。あったとしても、そこで変更することはできません。

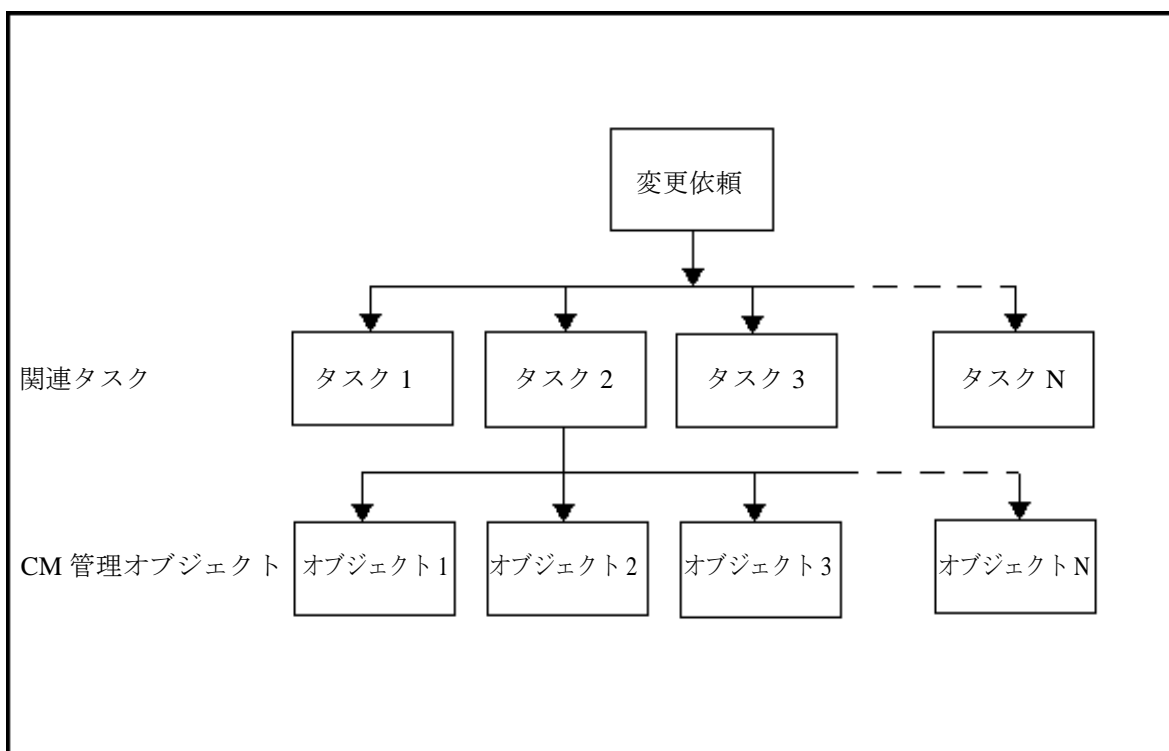
これを扱うため、DCS は受取り操作時に変更依頼をチェックします。したがって、この例では、DCS は、開発データベースに受取られる割り当て済み状態の変更依頼を探し、関連する遅延タスクを調べて再割り当てを行います。このような自動遷移タスクの詳細は、DCM 電子メール通知に入れられます。デフォルトの DCS モデル パラメータは、以下のいずれか、または両方を行う限りほとんどの場合に十分です。

- タスクの複製を行わないか、一部のタスクを複製する場合、変更依頼が割り当て済み状態にある場合にのみ開発データベースへ複製し、変更依頼が解決状態のときのみメイン製品管理データベースへ複製して戻します。
- すべてのタスクを関連オブジェクトとともに、あるいは関連オブジェクトなしで複製します。

つまり、デフォルトで、オブジェクトを受取るとそのオブジェクトから他のオブジェクトへの関係が受取りデータベースで再現されます。また、受取りデータベースに存在するが転送されたデータにはないその他の適切な関係は削除されます。

下図に示すように、Rational Change の主な関係は以下のとおりです。

- 変更依頼からタスクへ (associated\_task)
- タスクからオブジェクトへ (associated\_cv)



上図に示すオブジェクト関係により、一部のタスクがメイン製品管理データベースに複製して戻されると以下の問題が発生します。つまり、変更依頼が開発データベースに複製されると、変更依頼は適切なタスクすべてではなくその一部にのみ関係付けられます。デフォルトで、DCM 受取り操作時、他のタスクは変更依頼から関係が解かれます。

この問題を避けるには、1) Rational Change ライフサイクルに関連する正しい DCM ワークフローを使用するか、2) すべての関連タスクが変更依頼オブジェクトが存在するデータベースにあることを確認するか、3) 変更依頼を受取ったときイメージ処理を無効にします（これは上記の導入上の制限をすべて取り除きますが、タスクの削除または関係の削除は複製されません）。（イメージ処理の詳細については、112 ページの「Rational Change Distributed モデルパラメータ」およびそこで参照しているヘルプを参照してください）。



---

注記：デフォルトで、タスクとその関連オブジェクトでは上記の問題はありません。DCS のデフォルトの動作は、送信先データベースで変更可能なタスクの受取りをスキップします。この動作は変更することができますが、変更する前に 112 ページの「Rational Change Distributed モデル パラメータ」をお読みください。

### サンプル手法 3 — 中央 CCB データベース / サテライト開発データベースの使用

この手法では、各変更依頼は中央変更管理ボード (CCB) データベースへ送られます。CCB は、問題、案件、および変更依頼を見直して処理するチームです。これは、以下のいずれかの方法で行うことができます。

- 各変更依頼を直接 CCB データベースに入力する。このような問題は、親変更依頼で作成されます。  
または
- 変更依頼をローカルデータベースに入力するとき **Work in database** フィールドを CCB データベースに設定する。したがって、変更依頼は CCB データベースへ複製されそこで処理できます。このような問題は親変更依頼として作成されます。

いったん CCB データベース内に [親変更依頼](#) を作成すると、CCB は、その親に関連する 1 つまたは多数の子変更依頼を作成し、各 [子変更依頼](#) の **Work in database** フィールドを、サテライト開発データベースに対応するよう設定します。そのサテライト開発データベースの中ではその子に対する操作が行われます。

すべての開発作業は、サテライト開発データベース内で行われます。したがって、すべてのタスクと関連設定ファイル項目は、適切な開発データベースにあります。子変更依頼が完了するかまたは拒否されると、CCB に電子メールが送られます。これは、電子メールトリガを使用して実現できます。電子メールメッセージは、CCB に親変更依頼の状態と進捗を見直すよう指示します。CCB は、すべての子変更依頼が終末状態 (たとえば、決定など) になるまで親変更依頼を完了できません。

一般的に、変更依頼のみ CCB データベースへ複製して戻されます。CCB は通常、各子変更依頼のタスクへの分類を見たり、タスクから構成項目への分類を表示する必要はありません。ユーザーは、関連する開発データベースからそのような詳細を入手できます。

#### サンプル手法 4 — 関連タスク用のマルチ データベース / シングル データベースの使用

この手法では、変更依頼を任意のデータベースに入力して別のデータベースへ渡して処理できます。この方法は、サンプル手法 3 と同じような導入上の問題があります。しかし、この方法はハブ（製品管理データベース）とスポーク（開発データベース）トポロジの代わりにポイント ツー ポイント通信を使用するため、サンプル手法 3 よりも複雑である可能性があります。詳細については、107 ページの「ポイント ツー ポイント トポロジ」と 107 ページの「ハブとスポーク トポロジ」を参照してください。

---

## DCM 複製トポロジのタイプ

DCM は、ソース データベースとデスティネーション データベース間のリアルタイムの双方向フローに依存しません。したがって、DCS にデータベース間で変更依頼の管理を複製して渡す方法を知らせる必要があります。各種の複製トポロジを使用できます。DCS が安全に変更依頼の管理を渡すために、ハンドオーバーが行われる 1 つの固有の定義済み複製プロセスが必要です。ここでは、ポイント ツー ポイントおよびハブとスポーク複製トポロジに注目します。

### ポイント ツー ポイント トポロジ

ポイント ツー ポイント トポロジでは、各データベースは以下のいずれかの転送パッケージを生成します。

- 情報が送られるデータベース
- 変更依頼の管理が渡されるデータベース

たとえば、互いに変更依頼情報を複製し、場合によって管理を渡したい 3 つのデータベース (A、B、C) があると仮定します。ポイント ツー ポイント トポロジを使用して、データベース A は B と C に対して送信するか管理を渡す任意の変更依頼を生成できます。同じようにデータベース B は A と C に対する変更依頼を生成し、データベース C は A と B に対する変更依頼を生成できます。2、3 のデータベースのみが関わる場合、これがもっとも簡単な DCM 複製トポロジです。

詳細については、114 ページの「データベースと管理のハンドオーバー」を参照してください。

### ハブとスポーク トポロジ

ハブとスポーク トポロジでは、スポーク データベース同士は互いに直接複製を行いません。代わりに 1 つまたは複数のハブ データベースを介して複製を行います。したがって、DCS はデフォルトでどの複製パスがデータを 1 つのスポークから他のスポークへ渡すかわかりません。

構成のより詳しい例は、114 ページの「データベースと管理のハンドオーバー」を参照してください。

## Rational Synergy Distributed と Rational Change Distributed (DCS) の概要

ここでは、DCS を使用して以下の作業をどのように行うかについて説明します。

- Rational Synergy Distributed ダイアログを有効にする。
- Rational Synergy Distributed の用途を拡張する。

### Rational Synergy Distributed ダイアログ

DCS は、特定の DCM データベースに対する Rational Synergy Distributed Transfer Set ダイアログと Task ダイアログの機能を拡張します。ダイアログに応じて、DCS は以下のことを行います。

- Problem Scope フィールドと Problem Query フィールドを使用可能にする。
- Show Problems ボタンを使用可能にする。
- ダイアログを拡張して強調表示されているタスクを変更できるデータベースを使用可能にする。
- 未完了タスク（完了状態ではないタスク）が転送セットから除外されているかどうかを示す。

### 問題の範囲とクエリ

DCS は、転送セット定義を拡張して問題範囲と問題クエリを含めます。これらの定義は、転送セットが自動的にクエリをベースにした変更依頼を含めることを可能にします。以下の問題範囲をサポートしています。

- None — 変更依頼は含まれません。
- Problems Only — 特定のクエリで見つかった変更依頼は含まれますが、その関連タスクは含まれません。
- Problems and Tasks — 特定のクエリで見つかった変更依頼が関連タスクとともに含まれます。転送セットの除外プロパティウィンドウを使用して未完了タスクを除外できます。
- Problems and Tasks and Objects — 特定のクエリで見つかった変更依頼が関連タスクとともに含まれます。関連タスクが含まれると、その関連オブジェクトも静的であれば含まれます。タスクの非静的関連オブジェクトは、DCM 除外ルールで除外されます。

None 以外の問題範囲では、転送セットは、変更依頼クエリ式を定義できます。この式は、問題タイプのクエリ句を含む必要はありません。クエリ句は、クエリ式を使用するとき自動的に追加されます。したがって、ブランクのクエリ文字列は、すべての変更依頼オブジェクトを返します。クエリ式は、標準のクエリ構文を使用して任意の問題属性を含むことができます。たとえば、製品名をベースにしたクエリ `"product_name='foobar'"` が可能です。

問題クエリ文字列はタイプ入力する必要があります。ただし、問題クエリダイアログを使用して転送セットに指定する適切なクエリ文字列を決定できます。下表に、転送セット定義の問題クエリ文字列に使用できる標準キーワードの例を示します。

キーワード	結果
<code>%to_db_id</code>	これは、生成が行われる DCM データベースの <code>dbid</code> で置き換わります。これにより、同じ転送セット定義を複数のデータベースに適用できます。
<code>%from_db_id</code>	これは、転送パッケージの生成元となる現在のデータベースの <code>dbid</code> で置き換わります。

## DCM 初期化

DCM を使用するようデータベースを初期化すると、DCS はこの操作を拡張して `created_in` と `local_to` 属性を持つすべての変更依頼とタスク オブジェクトに、現在の DCM の初期値が `dbid` の `modifiable_in` 属性も持たせます。さらに、そのようなデータベースで変更依頼またはタスクを作成すると、`modifiable_in` 属性も設定することになります。詳細については 14 ページの「DCM 初期化」を参照してください。

## リモート担当者と電子メールトリガ

ローカルデータベースの認定ユーザーではない担当者を選ぶことができます。これは、DCS がユーザー リストを拡張して `extusers.dft` からユーザー定義可能な名前を入れるために可能です。このファイルには、外部ユーザー名のリスト (1 行に 1 つの名前) があります。ブランクの行とコメント文字 (#) で始まる行は無視します。外部ユーザー名は、他のデータベースで定義されているユーザーのユーザー ID です。

Rational Change では、変更依頼とタスクは常にローカル担当者に割り当てられます。したがって、電子メールトリガはローカルユーザー名のみを扱い、ローカルユーザー名用の電子メール別名を設定します。

しかし、DCS では、変更依頼またはタスク担当者はリモート データベースにある可能性があります。さらに、このユーザーはローカル サイトにいない可能性があります。したがって、そのようなリモート ユーザーには電子メール別名を設定するか、対象 dbid を渡すようトリガ スクリプトを直す必要があります。このことは、電子メールアドレスを作成するとき考慮する必要があります。

電子メール通知テキストは、変更依頼またはタスクが変更可能なデータベースの DCM dbid を含める必要があるかもしれません。デフォルトで、通知テキストは割り当てを行ったデータベースのパスを含みます。しかし、変更依頼またはタスクは別のデータベースでのみ変更可能な可能性があります。トリガ スクリプトは、自動的にリモート データベースのパスを決定できません。したがって、通知は DCM dbid のみを含むか、あるいはトリガ スクリプトが dbid をテストしてそれを適切なリモート データベース パスに対応付ける必要があるかもしれません。

---

## Rational Change ダイアログと用途の概要

ここでは、DCS が以下の作業をどのように行うかについて説明します。

- Rational Change ダイアログを使用する。
- Rational Change の用途を拡張する。

### ボタンとフィールド — 特殊動作

一部の DCS ダイアログには、対象オブジェクトが現在のデータベース内で変更可能かどうかによって有効または無効になるボタンがあります。デフォルトでは、属性の保存および/または既存オブジェクトの遷移を行うボタンはすべて、対象オブジェクトが変更可能な場合にのみ有効です。

読みやすくするため、Rational Change フィールドはグレー表示になることはありません（対象オブジェクトが変更可能ではない場合でも）。

### 現在のデータベース内の変更

**Modify** ボタンは、表示オブジェクトを変更できる場合にのみ **Rational Change** ダイアログに表示されます。**Rational Synergy Distributed** では、**Modify** ボタンをクリックすると、DCS がオブジェクトの `modifiable_in` 属性を調べます。オブジェクトが現在のデータベース内で変更可能ではない場合、確認ダイアログが表示されます。このダイアログは警告で、現在のデータベース内で変更を行っても良いか考える機会を与えます。

オブジェクトが別のデータベースで変更可能な場合は、そこで変更を行ってセキュリティルールを破らないことを推奨します。しかし、オブジェクトが現在のデータベースで変更可能で別のデータベースへ渡される場合は、そのオブジェクトを変更できます（ただし、DCM 転送に入れられていないことが前提です。なぜならば、いったんオブジェクトを転送パッケージに入れると、そのパッケージが受取られた可能性があり、オブジェクトが他で変更された可能性があるからです）。

この例では、警告を無視して続けると、別のデータベースで行われた変更が現在のデータベースの変更を上書きする可能性があります。そのような変更は、警告なしで失われます。明らかに、これらの変更を行うために必要な DCS の `pt_admin` ロールには大きな責任があります。

## Rational Change Distributed モデル パラメータ

DCS モデル パラメータは、DCS の振る舞いを制御します。そして、これらのパラメータはモデル オブジェクトの属性として設定できるようになっています。以下に DCS モデルパラメータの一覧を示します。

- `dcm_noimage_import_types`
- `dcm_allow_assoc_tasks_across_dbs`
- `dcm_problem_types`
- `dcm_relation_import_types`
- `dcm_problem_receive_actions`
- `dcm_task_assigner_role`

DCM クラスタ内のすべてのデータベースは、これらのパラメータに同じ設定を使用する必要があります。`ccm_admin` ロールのユーザーは、以下のコマンドを使用してこれらの設定を変更できます。

```
ccm query /t model /n base
```

```
ccm attr /m attribute_name /v "new_value" @1
```

DCS モデルパラメータの詳細については、Synergy Classic ヘルプの「[デフォルト オプション](#)」を参照してください。



# 9

## 上級 DCM トピック

### はじめに

このセクションは、DCM 操作のより詳しい説明を必要とするアドミニストレータを対象にしています。以下のトピックについて説明します。

- 114 ページの「データベースと管理のハンドオーバー」
- 115 ページの「管理データベースへの遷移」
- 116 ページの「汎用プロセスとリリース固有プロセスの複製」
- 121 ページの「DCM データベース定義の複製」
- 122 ページの「電子メール通知」
- 123 ページの「リコンフィギュア プロパティの全展開」
- 125 ページの「関連ベースラインを含む」
- 129 ページの「転送モード」
- 137 ページの「DCM パフォーマンス」
- 139 ページの「転送パッケージの場所」
- 140 ページの「自動受取りの使用法」
- 125 ページの「関連ベースラインを含む」
- 144 ページの「受取りの失敗」
- 147 ページの「クラスタ ID」
- 148 ページの「タイプ定義」
- 149 ページの「パラレル検査」
- 150 ページの「DCM が生成時刻を格納する方法」
- 151 ページの「DCM 受取り時のワークエリアの扱い」
- 153 ページの「ワークエリア サブディレクトリ テンプレートとパス」
- 154 ページの「転送セット内のオブジェクト履歴」
- 155 ページの「リリース 7.1 以前の Rational Synergy Distributed リリース間の転送」
- 157 ページの「DCM 設定」
- 159 ページの「DCM イベント ログ」
- 160 ページの「ブロードキャスト データベースとパッケージ」

## データベースと管理のハンドオーバー

DCM は、特定のオブジェクトに対して、どの時点でも DCM クラスタ内の 1 つのデータベース内でのみ変更可能なポリシーを実施してオブジェクトを保護します。他のすべてのデータベースでは、`ccm_admin` ロールを持たない限りオブジェクトを変更することができません。唯一の例外として、オブジェクトをより後の状態へ遷移できます。

オブジェクトの管理データベースは、その `modifiable_in` 属性または `created_in` 属性で決まります。`modifiable_in` 属性があれば、オブジェクトの管理データベースは、明示的に定義され、属性値は管理データベースの DCM データベース ID を指定します。`modifiable_in` 属性がない場合、管理データベースは `created_in` 属性の値で決まります。

オブジェクトの管理は、`modifiable_in` 属性の値を現在の DCM データベース ID から新しい管理データベースの ID に変更することで、現在の管理データベースから他のデータベースへ渡すことができます。これは、オブジェクトを管理のハンドオーバー保留状態にし、管理が別のデータベースへ無事渡されたことを確認するまでオブジェクトを更新から保護します。この確認は、オブジェクトを別のデータベースから受取ったとき検査することで可能にしています。管理のハンドオーバー機能は、通常、変更依頼、タスク、リリース定義、プロセス、およびプロセスルールとフォルダ テンプレートにのみ提供されます。しかし、これは特定の種類のオブジェクトに限定されません。

DCM は、以下のように、いくつかの方法でオブジェクトを保護します。

- セキュリティ ルールは、`ccm_admin` ロール以外のユーザーが別のデータベースで管理されているオブジェクト、あるいは管理のハンドオーバーが保留されているオブジェクトを変更するのを防ぎます。
- DCM 受取りは、インポート時にオブジェクトをスキップし、オブジェクトが送信先データベースで管理されているか、管理のハンドオーバーが保留されている場合には更新しません。DCM は、スキップしたオブジェクトのレポートを DCM イベント ログと送信した DCM 受取り電子メールに入れます。これは、オブジェクトの管理コピーが、別のデータベースからの自身のより古いコピーで上書きされるのを防ぎます。
- チェックイン コンポーネントでオブジェクトをチェックインするとき、他のデータベースで管理されているオブジェクトのコメントは更新されません。

デフォルトでは、DCM データベース定義が作成される時 `Handover allowed` 設定は `FALSE` で、そのデータベースへ管理を渡すことはできません。オブジェクトの管理を別のデータベースへ渡したい場合は、この設定を `TRUE` にします。

---

## 管理データベースへの遷移

デフォルトで、DCM はオブジェクトの管理コピーが他のデータベースからの自身のコピーで上書きされることを許可しません。詳細については、114 ページの「データベースと管理のハンドオーバー」を参照してください。この保護は、遷移の複製にも適用されます。たとえば、オブジェクトがデータベース A で作成され、*integrate* 状態に遷移し、データベース B に複製されるケースを見てみましょう。そのオブジェクトはデフォルトでデータベース A で管理されます。データベース B では、そのオブジェクトは *integrate* から *released* へ遷移します。デフォルトで、オブジェクトがデータベース B から A へ送られると、A のオブジェクトは更新されず、*integrate* 状態を保持します。この動作は、データベース A がオブジェクトを「所有」するので意図的なものです。このような保護がないと、データベース A でそのオブジェクトが *released* プロジェクトに含まれているにも関わらず、別のデータベースのビルドマネージャがオブジェクトを *rejected* 状態に遷移する可能性があります。

1 つのコンポーネントまたはアプリケーションが複数のデータベースにまたがって開発されているとき、非管理データベースで行われる一部の遷移を管理データベースに適用できるようにすることが望ましい場合があります。ただし、そのような遷移すべてが望ましいとは限りません。たとえば、*integrate* から *released* への遷移は複製して管理データベースへ戻りたいが、*integrate* から *rejected* または *released* から *rejected* は戻したくないかもしれません。Receive Control Transitions DCM 設定を変更して、このようなポリシーを導入できます。

DCM 受取りでは、転送パッケージ内のオブジェクトが送信先データベースで管理されている場合、DCM は既存のオブジェクトの状態と他のデータベースから受取ったオブジェクトの状態を調べます。Receive Control Transitions DCM 設定を調べ、遷移が許可されていればオブジェクトを新しい状態へ遷移します。しかし、オブジェクトの他のプロパティは更新されず、そのオブジェクトはスキップしたオブジェクトのレポートに残ります。status\_log 属性は、遷移を行った元のデータベースを示し、その遷移が前に管理データベースで適用されたときのエントリも含みます。

## 汎用プロセスとリリース固有プロセスの複製

プロセスは、一連の目的に対応する汎用プロセス ルールの集まりです。更新（更新メンバーおよびリコンフィギュアともいいます）操作がどのようにプロジェクトまたはプロジェクト グルーピング内の変更を含み、それらの変更がどのように開発から 1 つまたは複数のテストステージを通してリリースへ流れるかを、これらのプロセスルールが管理します。汎用プロセスルールは、リリースから独立しており、更新操作を行ったときどのように変更が集められるかを定義するパターンです。

リリース定義が作られると、汎用プロセスルールを使用して、そのリリースの特定のプロセスルールがどのように形成されるかを定義します。結果として得られるリリース特有プロセスルールが、そのリリースのプロセスの管理、およびそのリリースの更新時にプロジェクトまたはプロジェクト グルーピングにどの変更を含めるかを管理するために使用されます。リリース定義とそれが使用するプロセスルールの集まりが、リリースに使用される有効な目的、およびそのリリースと有効な目的を持つプロジェクトまたはプロジェクト グルーピングがどのように更新されるかを定義します。

Rational Synergy Distributed 6.5a 以降の DCM は、プロセスと汎用プロセスルールの複製、およびリリースとその関連するリリース固有プロセスルールの複製をサポートしています。以下のように、2つの個別の仕組みをサポートしています。

1. プロセスおよび/またはプロセスルールは DCM 転送セットに明示的に追加できます。プロセスを追加すると、そのプロセスが使用するプロセスルールも含まれます。プロセスルールを追加した場合、直接追加したかプロセスの追加によって間接的に追加されたかに関わらず、プロセスルールのフォルダとフォルダテンプレートも間接的に追加されます。
2. リリース定義と関連プロセスルールは、Release Scope および Release Query の使用により、間接的なクエリメンバーとして自動的に追加されることがあります。適用範囲とクエリはどのリリースが自動的に追加されるべきかを定義し、追加される各リリースに対してもそのプロセスルールが自動的に追加されます。

組み込まれている定義済み汎用プロセスルールとフォルダテンプレートは、常に、固有の除外ルールによって DCM 転送パッケージから除外されます。これらの組み込まれているプロセスルールとフォルダテンプレートは、変更も削除もできません。常に各データベースに等しく存在します。したがって、複製する必要はありません。

注記：プロセスルールに従ってフォルダが転送セットに間接的に追加される場合、そのフォルダのタスクは追加されません。フォルダのタスクを含めるには、そのフォルダを明示的に転送セットの直接メンバーとして追加する必要があります。

---

## ローカル管理と中央管理

DCM では、プロセス、プロセスルール、フォルダ テンプレート、およびリリース定義を、ローカル管理または中央管理、あるいはその両方の組み合わせで管理できます。

ローカル管理では、オブジェクトはローカルで管理され更新されます。そのオブジェクトがそのデータベースへの DCM 転送パッケージに含まれる場合、それは DCM 受取りのインポート フェーズではスキップされます。ローカル管理の利点は、各データベースがオブジェクトのプロパティを個別に管理でき、他のデータベースで使用されるプロパティの影響を受けないようにできる点です。ただし、データベース間でプロパティに一貫性を持たせたい場合には、各データベースで個別に管理し更新しなければならないため、メンテナンスにかかる労力が大きくなるという欠点があります。

一方、中央管理では、各オブジェクトに対して定義されたデータベースから管理される 1 つのマスタ兼管理コピーが存在します。オブジェクトの変更は、管理データベース内でのみ行われます。DCM クラスタ内の他のデータベースにオブジェクトのコピーが複製され、管理コピーから行われる変更によって更新されます。DCM は、オブジェクトの管理を現在の管理データベースから他の指定データベースへ渡すことを許可します。管理を渡した後は、そのオブジェクトは新しい管理データベースでのみ管理されます。異なるオブジェクトは異なるデータベースで管理できます。しかし、より標準的な使用形態は、リリース定義（あるいは少なくとも指定コンポーネントのリリース定義）などすべてのオブジェクトを、1 つのデータベースから管理する方法です。中央管理の利点は、DCM クラスタ間で一貫性を保つことができ、ビルド マネージャのメンテナンス労力を軽減するという点です。ただし、変更は管理データベースでのみ可能で、データベースごとにオブジェクトのプロパティを変えることができないという欠点があります。

DCM は複合的な管理手法もサポートしています。一部のリリースとそのプロセスルールを中央管理し、他をローカルに管理することが可能です。また、1 つのリリースを中央管理し、そのリリースのプロセスルールの一部をローカルに管理して、特定のデータベースで異なるプロセスを実装することもできます。ユーザーは、リリース固有プロセスルールをローカルに管理して、そのデータベースにローカルな変更を行うことができます。プロセスルールは、リリースの有効な目的として引き続きそのリリースに関連付けられています。

DCM では、ユーザーはローカル管理と中央管理モデルを切り替えて使うことができます。オブジェクトが複数のデータベースで定義されている場合、それは各データベースでローカルに管理されます。オブジェクトを中央管理したい場合には、以下のことを行う必要があります。

- どのデータベースがそのオブジェクトを管理するか決定し、オブジェクトがそのデータベースでローカルに管理される設定になっていることを確認する。
- オブジェクトがある他のすべてのデータベースで、管理データベースから管理を受取るよう管理データベースを設定する。
- そのオブジェクトを含む転送セットを使用して、管理データベースから他のデータベースへ DCM 複製を行う。オブジェクトがすでに転送セットのメンバーである場合、それが生成済み転送パッケージに含まれるよう「Never」以降生成を行う必要があるかもしれません。

中央管理をやめてローカル管理オブジェクトに戻るには、**Select Controlling database** ダイアログまたはコマンドライン オプションを使用してローカル管理を指定します。オブジェクトの詳細は変わりません。しかし、オブジェクトには、これが別のオブジェクトであり、DCM 受取りで更新すべきではないことを示すため、新しい固有のクラスタ ID が与えられます。

## リリース定義と関連プロセス ルールの複製

リリース定義の複製では、ユーザーがリリース定義とその関連するプロセスルールを中央管理し、DCM クラスタ内の他のデータベースを自動的に更新できます。

DCM クラスタがリリース値を使用してプロジェクトに共有データを構成する場合、すべての転送済みオブジェクトのリリース値はデスティネーションデータベースでリリース定義として定義されている必要があります。

リリース定義と情報の複製は、以下のパラメータと設定で管理されます。

- 各リリース定義には、**Allow DCM Transfer** という設定があります。デフォルトでこのオプションは、新しいコンポーネントのリリースが作成される時選択されるか、あるいは前のリリースをベースにしたリリースが作成されたとき前のリリースに設定されます。このオプションを選択解除することは、そのリリースの詳細を常にすべての DCM 転送パッケージから除外することを意味します。
- 各転送セットには、**Release Scope** と **Release Query** という設定があります。リリース範囲には以下の3つの設定があります。

**Releases and Templates** : リリース クエリと関連プロセスルールで探したリリース定義を複製します。これは、新しい転送セットを作成するときのデフォルトです。

**Releases** : リリース クエリで探したリリース定義を複製しますが、関連プロセス ルールは複製しません。この適用範囲を使用した場合、他のデータベースに複製されたリリース定義は、リモートデータベース内にそのリリースのプロセス ルールがすでに存在していないと、目的がないように見える可能性があります。

**None** : リリース定義のクエリを行いません。転送セットに明示的に追加されたプロセス ルール、あるいは転送セットにプロセス定義が追加されたことで間接的に追加されたプロセス ルールのみが複製されます。

リリース クエリは、一致するリリース定義を転送セットの間接的クエリ メンバーとして自動的に追加するために使用するクエリ式を定義します。空白の文字列はすべてのリリース定義のクエリを意味します。これは、新しい転送セットを作成するときのデフォルトです。

- 各受取りデータベースでは、**Update Release Definitions DCM** 設定が、受取る転送パッケージ内のリリース情報を処理するために必要な動作を管理します。設定と対応する動作は、下表で説明しています。

リリース定義の更新	動作
none	DCM 転送パッケージ内のすべてのリリース情報を無視します。
active	パッケージ内のリリース定義を使用して受取りデータベース内のリリース定義を作成および/または更新します。ただし、リリース定義はアクティブ リリースの場合にのみ作成されます。非アクティブ リリースの新しいリリース定義は作成されません。非アクティブで前に受取った既存のリリース定義は引き続き更新されます。 これはデフォルト設定です。
inactive	パッケージ内のリリース定義を使用して受取りデータベース内のリリース定義を作成および/または更新します。アクティブおよび非アクティブ リリースのリリース定義が作成または更新されます。

DCM 生成を行うと、DCM はその転送セットとデータベースの最後の生成時刻よりも後で変更されたか転送セットのメンバーとなったリリース定義、プロセスルールまたはフォルダ テンプレートのみを含めます。したがって、Never 以降 DCM 生成を行い、転送セットのリリース範囲が **Releases and Templates** に設定されており、かつリリース クエリが空白の場合、すべてのリリース定義と関連プロセスルールおよびユーザー定義フォルダ テンプレートが、DCM 転送パッケージに含まれます。

リリース定義のチェックまたは手動変更を行うには、**Browse Releases** ダイアログまたは `ccm release` コマンドを使用します。



---

## DCM データベース定義の複製

デフォルトで、DCM では DCM 転送パッケージのソース データベースに DCM データベース定義の多少の詳細があります。これは、DCM 受取り時に送信先データベース内で対応する DCM データベース定義を作成または更新するために使用します。これは、DCM データベース定義を設定する手順を自動化し、クラスタのいたる所で説明、ロケーション、および管理情報を一貫して使用できるようにするのに役立ちます。

DCM データベース定義内のすべての情報が複製されるわけではありません。転送モード、パス、自動受取り設定などは、異なるデータベース内の同じデータベース定義では異なる必要があるかもしれません。たとえば、ニューヨークのデータベース NY1 と NY2、およびロンドンのデータベース LO1 と LO2 からなるクラスタを想定します。NY2 の DCM データベース定義は、NY1 から直接転送モードを使用し、LO1 と LO2 からは ftp を使用する可能性があります。

データベースの DCM データベース ID を変更すると、この変更は自動的に複製されません。各データベース内の対応する DCM データベース定義が更新されるように、他のすべてのデータベースの DCM アドミニストレータに変更した識別子を通知する必要があります。詳細については、64 ページの「データベース ID の変更」を参照してください。

## 電子メール通知

Rational Synergy Distributed は、DCM 電子メール通知にデフォルトのメールツールを使用します。Rational Synergy Distributed のメイラー以外のメイラーを使用するには、`ccm.ini` ファイルの `Options` セクションに以下の行を入力します。

```
mail_cmd = user-defined mail command
```

`user-defined mail` コマンドの構文は、使用しているメイラーによって異なります。しかし、メイラーは一般的に受取人、件名、および内容オプションと引数を必要とします。

たとえば、以下に `ccmail` の `mail_cmd` 定義を示します。

```
mail_cmd = C:¥ccmail¥mailer.exe -r %recipients -s  
%subject -f %content
```

`%recipients`、`%subject`、および `%content` 引数は、ユーザーがダイアログで与える情報をもとに Rational Synergy Distributed が自動的に展開します。

## リコンフィギュア プロパティの全展開

ソース データベースの転送セットで **Fully Expand Reconfigure Properties** (リコンフィギュア プロパティを完全に展開) を選択すると、DCM 生成操作時にプロジェクトは完全に展開されます。これはプロジェクトが以下を含むことを意味します。

- すべてのタスクの関連オブジェクト — これらのオブジェクトがプロジェクト階層のメンバーではない場合でも。
- プロジェクトの更新プロパティ内のすべてのフォルダとタスク
- プロジェクトの更新プロパティで参照されているベースラインのすべてのサブプロジェクト
- 上記すべて — プロジェクトが変更禁止または静的状態 (たとえば、*released*、*integrate*、*test*、*sql* など) にある場合でも。

下表に、転送セットに各タイプのオブジェクトを追加し、**Fully Expand Reconfigure Properties** オプションを選択した場合に含まれるものを示します。

オブジェクト タイプ	転送セットに追加されるオブジェクト	履歴
フォルダ	フォルダ自身、そのメンバー タスク、およびタスクに関連するすべてのオブジェクト。	適用外
タスク	タスク自身とタスクに関連するすべてのオブジェクト。	適用外
プロジェクト	プロジェクト自身とプロジェクト階層のすべてのメンバー オブジェクト。 <i>prep</i> プロジェクトなど非静的プロジェクトには、以下のものが追加されます。 --プロジェクトが使用している任意のプロジェクトベースライン --プロジェクトの更新プロパティ内のすべてのフォルダとタスク、および --フォルダ内のすべてのタスク すべてのプロジェクト ( <i>released</i> プロジェクトなどの静的プロジェクトを含む) に対して以下も追加されます。 --プロジェクトが使用している任意のプロジェクトベースラインとそのベースラインのすべてのサブプロジェクト --プロジェクトの更新プロパティ内のすべてのフォルダとタスク --フォルダ内のすべてのタスク、および --そのようなタスクに関連するすべてのオブジェクト (プロジェクト階層のメンバーではない場合でも)。	オプション

オブジェクト タイプ	転送セットに追加されるオブジェクト	履歴
ベースライン	ベースライン オブジェクト自身、ベースラインのプロジェクト メンバー、およびベースラインのタスク メンバー。ベースライン プロジェクト メンバーとタスク メンバーは、上記のようにさらに展開されます。	適用外
ソース	オブジェクト自身とオブジェクトの関連タスク。	オプション

デフォルトを使用する場合（つまり、Fully Expand Reconfigure Properties が選択されていない場合）は、以下のようになります。

- 静的プロジェクトはフォルダ、タスク、およびタスクの関連オブジェクトの展開を含まない。
- Prep プロジェクトは、フォルダとタスクを含むが、タスクに関連するオブジェクトは含まない。プロジェクト階層で使用されるオブジェクトのみ含まれる。

Fully Expand Reconfigure Properties オプションの詳細については、44 ページの「転送セットの作成」および 45 ページの「転送セットへのオブジェクトの追加」を参照してください。

---

## 関連ベースラインを含む

**Include Associated Baselines** 転送セットオプションは、タスクとプロジェクトが転送セットのメンバーである場合、それらのタスクまたはプロジェクトに関連するベースラインも転送セットのメンバーとして含まれるかどうかを管理します。デフォルト設定ではこのオプションは選択されません。ベースラインは転送セットの直接メンバーとして追加できますが、これは **Include Associated Baselines** オプションとは無関係です。

転送セットに関連ベースラインを含める利点は、プロジェクトまたはタスクを別のデータベースへ複製するときその関連ベースラインもデスティネーションデータベースへ一緒に付いていく点です。これによりユーザーは、**Find Use** を行ってどのベースラインにプロジェクトまたはタスクが含まれているか調べることができます。

複数のデータベースで開発された、異なるコンポーネントを組み合わせて作成するアプリケーションの場合、関連ベースラインを含めると更新（リコンフィギュア）に問題が発生する可能性があります。これは、どのデータベースでもベースラインはそのデータベースで開発されたコンポーネントに関連するプロジェクトのサブセットがあり、それはすべてのデータベースにわたってすべてのプロジェクトを表さないためです。そのベースラインを別のデータベースへ送ると、更新はそれを最新のベースラインとして選択する可能性があります。しかし、それはベースラインプロジェクトの選択に必要なすべてのプロジェクトメンバーを持たず、更新は一部のプロジェクトの適切な候補バージョンを選択できない可能性があります。

定義済みの **Entire database** 転送セットを使用している場合は、すべてのベースラインが各データベースへ複製され、**Include Associated Baselines** オプションの設定は意味がありません。

アプリケーションに固有のリリース値を使用してデータベース内でアプリケーションを開発していて、そのアプリケーションのすべてのプロジェクトを複製する場合、**Include Associated Baselines** の選択は開発者にメリットがあります。しかし、各データベースに一部のプロジェクトしか複製しない場合、このオプションの選択は有害なのでデフォルト設定を使用すべきです。

## 転送パッケージの構造

転送パッケージには、DCM がバイナリとして読み書きする3つのオブジェクト、情報ファイル、プレビューファイル、およびコンテンツファイルまたはディレクトリがあります。DCM は、ソース データベースが転送パッケージを生成するとき、これらのオブジェクトをソース データベースの [generate\\_directory](#) に作成します。DCM が無事ファイルをデスティネーションディレクトリへ転送した確認をソース データベースが受取るまで、ファイルは [generate\\_directory](#) に残ります。転送が終了すると、転送ファイルはデスティネーションデータベースの [receive\\_directory](#) にあります。例外は、[ダイレクトモード](#) 転送モードを使用する（つまりソース データベースが直接デスティネーションデータベースの [receive\\_directory](#) にオブジェクトを生成する）ときです。ファイルは、DCM がそれらが無事デスティネーションデータベースへロードするまで [receive\\_directory](#) に残ります。

パッケージ コンテンツ ファイルまたはディレクトリは、最大5つの部分を含むことができます。

- 転送セットのメンバーシップで定義されているファイル、ディレクトリ、タスクなどのユーザーデータ。これは [data](#) ディレクトリの下に置かれます。
- [releases](#) ディレクトリの下オプションのリリース部分。ここにはリリース定義に関する情報があります。
- [templates](#) ディレクトリの下オプションのテンプレート部分。ここにはフォルダテンプレートに関する情報があります。
- [process](#) ディレクトリの下オプションのプロセス部分。ここにはプロセスルールおよびプロセスルールに関する情報があります。
- [types](#) ディレクトリの下オプションのタイプ部分。ここにはタイプ定義に関する情報があります。

転送ファイル名の構文は以下のとおりです。

```
<FROM_DB_ID><TSET_NUMBER><TO_DB_ID><TIME><name>
```

ここで、*name* は DCM 情報ファイルでは `dcm_info.txt`、DCM プレビューファイルでは `dcm_preview.txt`、圧縮パッケージコンテンツでは `.tar.gz`、そして非圧縮コンテンツでは `.dcm` です。*name* の前にある名前の部分は、特定の転送パッケージの一部であるすべてのファイルで同一です。*name* の前にある情報の説明は、以下のセクションを参照してください。

---

## DCM 情報ファイル

dcm\_info.txt ファイルには、デスティネーションデータベースが使用する DCM データベース パラメータと転送セット パラメータがあります。このほとんどの情報は、Generate ダイアログにあります。

## DCM プレビュー ファイル

dcm\_preview.txt ファイルには、転送パッケージのユーザー データ部分内にあるオブジェクトのリストがあります。Preview Transfer ダイアログは、デスティネーションデータベースでこのリストを使用します。このファイルには、以降の行に含まれるフィールドのラベルとなるヘッダ行があり、その後転送セット内の各オブジェクトのレコードが続きます。プレビュー ファイルは、DCM 転送パッケージの管理部分に含まれる情報の詳細は含まないことに注意してください。たとえば、パッケージに含まれるタイプ定義やリリース定義はありません。

以下に DCM プレビューファイルの例を示します。

```
displayname@@@name@@@version@@@cvtype@@@subsystem@@@owner@@@status@@@release@@@created_in@@@local_to@@@create_time@@@modify_time@@@task_synopsis@@@problem_synopsis@@@description
```

example-

```
1@@@example@@@1@@@project@@@A#1@@@ccm_root@@@prep@@@1.0@@@A@@@A@@@1161698530@@@1161698548@@@@@@@@
```

example-

```
1@@@example@@@1@@@dir@@@A#1@@@ccm_root@@@integrate@@@1.0@@@A@@@A@@@1161698533@@@1161698555@@@@@@@@
```

```
All 1.0 Integration Testing Projects from database  
A@@@1.0%003aintegrate@@@1@@@project_grouping@@@A#1@@@ccm_root@@@prep@@@1.0@@@A@@@A@@@1161698535@@@1161698537@@@@@@@@
```

```
A#1@@@1@@@1@@@folder@@@A@@@ccm_root@@@prep_folder@@@1.0@@@A@@@A@@@1161698537@@@1161698537@@@@@@@@All
```

Completed Tasks for Release 1.0

example.txt-

```
1@@@example.txt@@@1@@@ascii@@@A#1@@@ccm_root@@@integrate@@@1.0@@@A@@@A@@@1161698548@@@1161698555@@@@@@@@
```

## パッケージの内容

.tar.gz ファイルまたは .dcm ディレクトリには、転送されるオブジェクトがあります。生成操作時、DCM は以下のことを行ってこのファイルまたはディレクトリを作成します。

1. タイプ定義が除外されていない場合、DCM は `generate_directory` `named <FROM_DB_ID><TSET_NUMBER><TO_DB_ID><TIME>¥types` という一時ディレクトリを作成し、ユーザーが定義した各タイプ定義をエクスポートします。形式は、`ccm typedef /export` コマンドを使用した場合と同じです。
2. リリース定義が除外されていない場合、DCM は `generate_directory` に `<FROM_DB_ID><TSET_NUMBER><TO_DB_ID><TIME>¥releases` という一時ディレクトリを作成し、含めるリリース定義をエクスポートします。また、Rational Synergy Distributed の旧リリースと互換性を保つため、これらリリース定義用の擬似テーブルを作成し、`<FROM_DB_ID><TSET_NUMBER><TO_DB_ID><TIME>¥rel_tbl.txt` ファイルを作成します。
3. テンプレートが除外されていない場合、DCM は `generate_directory` に `<FROM_DB_ID><TSET_NUMBER><TO_DB_ID><TIME>¥templates` という一時ディレクトリを作成し、含めるテンプレートをエクスポートします。
4. 転送パッケージのデータ部分に含めるオブジェクトを決定します。
5. `generate_directory` に一時サブディレクトリを作成します。ディレクトリは、`<FROM_DB_ID><TSET_NUMBER><TO_DB_ID><TIME>¥data` です。
6. 転送リスト内の各オブジェクトを一時データ サブディレクトリへエクスポートします。
7. パッケージを圧縮する場合、`ccm_gzip` とパイプした `ccm_tar` を使用して `<FROM_DB_ID><TSET_NUMBER><TO_DB_ID><TIME>` 一時ディレクトリのもとのすべてのファイルとディレクトリを1つのファイルファイルに集め、一時サブディレクトリを削除します。パッケージを圧縮しない場合、パッケージコンテンツディレクトリは、`<FROM_DB_ID><TSET_NUMBER><TO_DB_ID><TIME>.dcm` と名前変更されます。



---

## 転送モード

転送モードは、DCM がデスティネーション データベースへ転送パッケージを送る方法です。

DCM は、以下の転送モードを提供します。

- 手動コピー
- ダイレクト
- ローカル コピー
- リモート コピー
- ファイル転送プロトコル
- ユーザー定義

これらの方法は、以下で詳細に説明します。UNIX データベースと Windows データベース間の転送には、クロスプラットフォーム情報が含まれます。

転送モードの選択方法は、43 ページの「DCM データベース定義の作成」または 55 ページの「転送パッケージの生成」を参照してください。

### 手動コピーモード

ユーザー提供オプション：転送パス（オプション）

手動コピーモードを使用して転送パッケージを生成すると、DCM は転送パッケージを `generate_directory` に入れますが、それを送信しません。手動コピーモードは、転送パッケージを自動的に送信したくないとき、または自動的に転送する方法がないときに使用します。

このモードでは、ユーザーは `generate_directory` からデスティネーションデータベースの `receive_directory` へ、手作業で移動する必要があります。一般的に、これは、磁気テープまたは書き込み可能 CD を使用して行います。

転送ファイルの詳細については、125 ページの「関連ベースラインを含む」を参照してください。

### ダイレクトモード

ユーザー提供オプション：OS、Path、および Transfer Path

ダイレクトモードを使用して転送パッケージを生成すると、転送パッケージは直接デスティネーションデータベースの転送パスに作られます。したがって、ダイレクトモードは、ソース データベースとデスティネーションデータベースが同じ LAN 上の相互に参照可能なファイルシステム上にある場合にのみ使用できます。

## ローカル コピーモード

### ユーザー提供オプション : OS と Path

ローカル コピーモードを使用して転送パッケージを生成すると、DCM はコピー コマンドを使用して転送パッケージを `generate_directory` からデスティネーションデータベースの `receive_directory` へ送ります。転送パッケージを生成する Rational Synergy Distributed エンジンがコピーを行います。

ローカル コピー モードは、ソース データベースのエンジンがデスティネーション データベースの `receive_directory` にユーザー `ccm_root` として書き込み可能な場合に適しています。そのような転送は、同じファイル システムにアクセスできる 2 つの DCM データベース間でよく行われます。

DCM は、OS フィールドを使用してどのコピー コマンドを使用するか判断します。ソース データベースが UNIX ベースの場合、DCM は UNIX の `cp -r` コマンドを使用します。ソース データベースが Windows ベースの場合、DCM はファイルに Windows の `copy` コマンドを使用し、ディレクトリに `xcopy` コマンドを使用します。

DCM は、Path フィールドを使用してデスティネーション データベースの `receive_directory` へのパスを完成します。ローカル コピー モードでは、DCM は自動的に Path 値に `dcm¥receive` ディレクトリを付加します。

注記: Windows-to-UNIX ローカル コピー モードを使用する転送では、デスティネーション データベースを含む UNIX ファイル システムが Windows ホストから参照できる必要があります。これは、Windows 用のサードパーティの NFS クライアント パッケージあるいは Samba などのゲートウェイ サーバー ソリューションを使用して実現できます。

下表に、UNIX および Windows 転送のローカル コピー モードのパス値の例を示します。

ソース	デスティネーション	デスティネーション データベース パス
UNIX	UNIX	/ccm71a/ccmdb/ESX
UNIX	Windows	/ccm71a/ccmdb/ESX1
Windows	Windows	¥¥server_machine¥ccm71a¥ccmdb¥ESX2
Windows	UNIX	x:¥ccm71a¥ccmdb¥ESX

---

## リモート コピーモード

ユーザー提供オプション : Host Name、OS、および Path

リモート コピー モードを使用して転送パッケージを生成すると、DCM は `rcp` コマンドを使用して `generate_directory` からデスティネーション データベースの `receive_directory` へ転送パッケージを送ります。転送パッケージを生成する Rational Synergy Distributed エンジンがコピーを行います。

リモート コピー モードは、ソース データベースのエンジンがデスティネーション データベースのマシンにログインしてデスティネーション データベースの `receive_directory` にユーザー `ccm_root` として書き込み可能な場合に適していますが、共通のファイルシステムはありません。

DCM は、Host Name フィールドをデスティネーション データベースマシンの名前として使用します。

DCM は、OS フィールドを内部的に使用します。ソースデータ ベースが UNIX ベースの場合、DCM は以下の `rcp` コマンドを使用します。

```
rcp -r info_file preview_file path_to_data
host_name:receive_directory
```

ソースデータベースが Windows ベースの場合、DCM は以下の `rcp` コマンドを使用します。

```
rcp -b -r info_file preview_file path_to_data
host_name:receive_directory
```

注記 : `rcp` サービスは、UNIX-to-Windows および Windows-to-Windows リモート コピー転送では、Windows ホストで実行している必要があります。ただし、`rcp` サービスは Windows とともに出荷されません。サードパーティの商用パッケージまたはシェアウェア パッケージを使用して、`rcp` サービスを利用できます。これらのパッケージは、通常、Windows マシン用の `rcp` と `rsh` 機能をサポートします。パッケージによっては、`d:¥ccm71a¥ccmdb¥ESX` の `d:` のようにパス名でドライブ文字のマッピングをサポートしています。

詳細については、使用する `rcp` ソフトウェア パッケージのマニュアルを参照してください。

特定のサードパーティ `rcp` サービス パッケージは保証あるいはサポートされません。ユーザーの環境でそのようなパッケージを使用してソースとデスティネーション ホスト間のファイルとディレクトリのリモート コピーが可能か確認してください。

DCM は、Host Name と Path フィールドを使用してデスティネーションデータベースの *receive\_directory* へのパスを完成します。リモートコピーモードでは、DCM は自動的に Path 値に `dcm¥receive` ディレクトリを付加します。

注記：rcp をテストして設定とセキュリティを確認してください。Host Name と Path フィールドに自分の値を使用して、DCM 外でコマンドをテストします。

下表に、UNIX および Windows 転送のリモートコピーモードのパス値の例を示します。

ソース	デスティネーション	デスティネーションデータベースパス
UNIX	UNIX	/ccm71a/ccmdb/ESX
UNIX	Windows	¥¥server_machine¥ccm71a¥ccmdb¥ESX1
Windows	Windows	¥¥server_machine¥ccm71a¥ccmdb¥ESX1
Windows	UNIX	/ccm71a/ccmdb/ESX

### ファイル転送プロトコル

ユーザー提供オプション：Host Name、OS、および Path

ファイル転送プロトコル (ftp) モードを使用して転送パッケージを生成すると、DCM は ftp コマンドを使用して転送パッケージを *generate\_directory* からリモート デスティネーションデータベースの ftp ディレクトリへ送ります。転送パッケージを生成する Rational Synergy Distributed エンジンがコピーを行います。

ファイル転送プロトコルモードは、ソース データベースのエンジンがデスティネーションデータベースのファイルシステムにアクセスできない場合に適しています。ファイル転送プロトコルモードは、未認証ユーザーが匿名ログインを使用して保護されている施設の特定のディレクトリへデータを転送することを許可します。このモードは、認証されたユーザーが自分のログイン ID を使用してデータを転送することも許可します。

DCM は、ftp サーバーの名前に Host Name フィールドを使用します。デスティネーションマシンで、ftp サービスが実行している必要があります。

注記：Windows エンジンからの ftp 転送には、Microsoft® が BackOffice internet suite for Windows 2000 Server を提供しています。サードパーティのサーバーもあります。詳細については、使用するサービスのマニュアルを参照してください。

---

DCM は、OS フィールドを内部目的にのみ使用します。

DCM は、転送パスを使用して ftp ディレクトリへのパスを指定します。転送パスにブランクを使用すると、デスティネーション データベースの `receive_directory` へのデフォルトのパスが使用されます。

注記:UNIX サーバに接続されている Windows クラスタから ftp 転送を開始しないでください。この操作では転送は失敗します。これは、クライアントからサーバーのファイル システムにある DCM 転送パッケージが見えないためです。

サイトによってはセキュリティ上の制限があります。その場合、ステージング エリアへのみデータを送ることができます。このステージング エリアを `receive_directory` に指定できます。

注記:これらを DCM で使用する前に手動で ftp コマンドを試してみてください。以下を参考にしてください。

## UNIX エンジンからの FTP

UNIX マシンから ftp 転送を開始する場合、エンジンまたはクライアントプロセスはユーザーの介入なしにデスティネーションマシンにログインする必要があります。つまり、ftp コマンドにホスト名、ユーザー名、およびパスワード ログイン引数を与える必要があります。これは、ソース マシンの `.netrc` ファイルにログイン引数を入力して行います。

たとえば、以下の `.netrc` ファイルエントリは、ESX ftp サーバーへの匿名転送のログイン引数を与えます。

```
machine ftp.ESX.com
login anonymous
password dcm_transfer
```

セキュリティを強化するため、`.netrc` ファイルに他のログインとパスワードの組み合わせを入れることができます。

DCM は、まず `ccm_root` のホーム ディレクトリの `.netrc` ファイルを使用して、エンジンから転送を試みます。エンジンからの転送が失敗すると、DCM は転送を行っている人のホーム ディレクトリにある `.netrc` ファイルを使用して、クラスタから転送を試みます。

UNIX エンジンの ftp 転送では、Korn シェル (`ksh -c`) を使用して以下の ftp コマンドを実行します。

```
ftp @host < @infile
```

ここで、`@infile` には一時的に作成したコマンド ファイルを代入し、`@host` には Host Name フィールドの値を代入します。

コマンド ファイルの作成には、まず以下のいずれかの内容を採用します。

- `<database_path>/etc/UNIXftp.in` ファイル。デフォルトでこのファイルには以下のコマンドがあります。

```
binary
```

- 上記のファイルがない場合は、`<database_path>/dcm/UNIXftp.<dbid>` ファイル。ここで `<dbid>` はデスティネーション DCM データベース ID です。

上記のファイルの内容に以下のコマンドが追加されます。

```
put <data_file> <data_file>
put <preview_file> <preview_file>
put <info_file> <info_file>
get <info_file> <temp_file>
quit
```

最後の `get` コマンドは、DCM が DCM 情報ファイルをリモート マシンから取得したものと比較するためにあります。これは、多くの `ftp` の実装が、転送が失敗したり接続が無事行われない場合でもゼロの正常終了状態コードを返すために行います。

注記：一部の UNIX `ftp` サーバーは、転送パッケージファイル名で文字「#」の使用をサポートしていません。したがって、DCM は UNIX-to-UNIX `ftp` 転送でこれらの文字を自動的に「@」に変換します。受取り操作は、いずれの形式の転送パッケージ ファイル名も扱うことができます。また、`.netrc` ファイルは、`rwX-----` である必要があります（つまり、所有者のみ読み取り／書込み／実行可能）。

## Windows マシンからの FTP

Windows マシンから `ftp` 転送を開始するとき、エンジンまたはクライアントプロセスはユーザーの介入なしにデスティネーション マシンにログインする必要があります。組み込まれている Windows `ftp` コマンドが、無人 `ftp` 転送をサポートしています。

組み込まれているコマンドは以下のとおりです。

```
ftp -n -s:@infile @host
```

ここで、`@infile` には一時的に作成したコマンドファイルを代入し、`@host` には Host Name フィールドの値を代入します。

`ccm.ini` ファイルの Options セクションに以下のような行を挿入して別の `ftp` コマンドを使用できます。

```
DCM_NT_FTP_CMD=your_command
```

コマンドファイルの作成には、まず以下のいずれかの内容を採用します。

- `<database_path>/etc/UNIXftp.in` ファイル。デフォルトでこのファイルには以下のコマンドがあります。

---

```
user anonymous dcm_transfer
binary
```

このファイルのユーザー名とパスワードを修正して、データベースからのすべての Windows ftp 転送のログインコマンドを変更できます。

- 上記のファイルがない場合は、<database\_path>/dcm/UNIXftp.<dbid> ファイル。ここで、<dbid> はデスティネーション DCM データベース ID です。そのデスティネーション データベースに固有のログイン コマンドを入れることができます。

ファイルの内容に以下のコマンドが追加されます。

```
put <data_file> <data_file>
put <preview_file> <preview_file>
put <info_file> <info_file>
get <info_file> <temp_file>
quit
```

最後の get コマンドは、DCM が DCM 情報ファイルをリモート マシンから取得したものと比較するためにあります。これは、多くの ftp の実装が、転送が失敗したり接続が無事行われない場合でもゼロの正常終了状態コードを返すために行います。

注記：一部の UNIX ftp サーバーは、DCM 転送パッケージ ファイル名で文字「#」の使用をサポートしていません。したがって、DCM は Windows-to-UNIX ftp 転送でこれらの文字を自動的に「@」に変換します。受取り操作は、いずれの形式の転送パッケージファイル名もサポートしています。

## ユーザー定義モード

ユーザー提供オプション：Host Name、OS、および Path

ユーザー定義モードでは、スクリプトまたはバッチ ファイルを使用して転送を行うことができます。これは、たとえば、転送する前にデータを暗号化する場合に必要です。

このモードを使用すると、DCM は内部コマンドの代わりに外部コマンド スクリプトを実行します。このスクリプトは、<database\_path>\bin ディレクトリになければならず、UNIX エンジンの場合は dcm\_transfer、Windows エンジンの場合は dcm\_transfer.bat という名前になければなりません。このスクリプトは、ゼロ以外の値を返してエラーを報告します。ゼロはすべてのコマンドが正常に終わったことを意味します。

このスクリプトは、以下の順に 10 個の引数を受け付けます。

1. デスティネーション データベース ID
2. デスティネーション サーバー OS (UNIX または Windows)
3. デスティネーション サーバー ホスト名
4. デスティネーション サーバー CCM\_HOME インストール エリア
5. ソース データベース ID
6. デスティネーション データベース パス
7. 転送パス
8. DCM データ ファイルまたはディレクトリへのパス
9. DCM プレビュー ファイルへのパス
10. DCM 情報ファイルへのパス

上記の引数を使用して、転送が完了した後で自動受取りを行うことができます。ユーザー定義転送スクリプトで自動受取りを行うには、43 ページの「DCM データベース定義の作成」で説明しているように、デスティネーション データベース定義で **Automatic Receive** をオンにします。



---

## DCM パフォーマンス

DCM のパフォーマンスは、多数の要因に左右されます。ここでは、これらの要因を説明し、DCM のパフォーマンスを改善するヒントを提供します。

### 転送セット

転送セットが[直接メンバー](#)を含む場合、転送セットが「要再計算」とマークされることがあります。たとえば、以下のような場合です。

- 転送セットがプロジェクトを含み、そのプロジェクトのメンバーシップが変更されたとき。たとえば、プロジェクトのメンバーシップは、Update Members、Use、または Unuse 操作時に変わることがあります。
- 転送セットがフォルダを含み、そのフォルダのタスク メンバーシップが変更されたとき。たとえば、フォルダのタスク メンバーシップは、Update Folder、Add Task、または Remove Task 操作時に変わることがあります。

生成操作を行ったとき、転送セットが再計算を必要とする場合、すべての間接メンバーを調べるためすべての直接メンバーが展開されます。多数のプロジェクトとフォルダを含む複雑な転送セットでは、これは多くの処理時間とデータベースクエリを使用します。

多くの場合、1 つの大きな転送セットよりも複数の転送セットを使用する方がパフォーマンスと効率が向上します。大きな転送セットを再計算する代わりに、小さい転送セットの再計算で済むかもしれません。

転送セット定義には、Fully expand reconfigure properties オプションがあります。このオプションは、デフォルトで選択解除されています。一般的に、デフォルト設定を使用した方がより小さい転送セットの間接メンバーシップになります。

Fully Expand Reconfigure Properties オプションを選択すると、転送セットはその転送セットのメンバー プロジェクトすべての更新プロパティを展開します。その上、含まれるタスクのすべての関連オブジェクトも転送セットの一部になります。これは多くの場合、大きな転送セット メンバーシップとなり、そのメンバーシップの再計算には時間がかかります。

### ローカル データベース

デスティネーション データベースがローカルの場合、適切なデータベース定義オプションを使用することでパフォーマンスを向上できます。

ローカル ファイル システムを介して参照可能なデスティネーション データベースでは、以下のようにします。

- [ダイレクトモード](#)転送モードを使用する。これは、転送パッケージを圧縮せず、直接デスティネーション データベースの転送パスに生成します。

ローカル ファイル システムを介して参照できないが、高速ネットワーク接続を使用しており、信頼できるホスト（つまり、パスワードを入力せずに `rcp` および `rsh` コマンドを使用できる）上にデスティネーション データベースがある場合、以下のようにします。

- 非圧縮パッケージおよびリモート コピー転送モードを使用する。また、転送セットの [generate\\_directory](#) に NFS ファイル システムではなくローカル ファイル システムを指定した方が、パフォーマンスが向上します。

### リモート データベース

デスティネーション データベースが信頼できるホストにない（つまり、パスワードを入力しないと `rcp` および `rsh` コマンドを使用できない）か、またはデスティネーション ホストとの接続に遅いネットワーク回線を使用している場合、以下のようにします。

- 圧縮パッケージを使用し、ファイル転送プロトコル、ユーザー定義、または手動コピー転送モードを使用する。

---

## 転送パッケージの場所

DCM では、1 つのデスティネーション データベース定義に 2 つのパスを指定できます。

- Path は、デスティネーション データベースへのパスです。
- Transfer Path は、DCM がデスティネーション データベース用に生成された転送パッケージを置く場所です。

Transfer Path がブランクの場合（これは新しいデスティネーション データベースを追加するときのデフォルト）、デフォルトの転送パスはデスティネーション データベースの `dcm\receive` ディレクトリです。

DCM がデスティネーション データベースのファイル システムに転送パッケージを書き込む方法の詳細については、28 ページの「デスティネーション データベース」を参照してください。

## 自動受取りの使用法

自動受取りは、生成操作の一部としてソース データベースで開始する操作です。自動受取りでは、GUI またはコマンドラインからの 1 つの操作を使用して生成、転送、および受取りを行うことができます。

自動受取りを伴う生成操作を行うと、DCM は `ccm_receive` コマンドを使用してデスティネーション マシンで受取りを開始します。このコマンドは、Rational Synergy Distributed セッションを開始し、引数を DCM 受取り操作に渡し、セッションを停止します。

DCM は、ユーザー `ccm_root` として自動受取りを行います。これは、Rational Synergy Distributed エンジン プロセスが受取りを行い、エンジン プロセスが `ccm_root` として実行するために発生します。その上、`ccm_root` ユーザーは、DCM がデータをロードするとき変更禁止オブジェクトの変更に必要な `ccm_admin` ロールを持っています。

以下では、自動受取りを可能にするようソース データベースとデスティネーション データベースを設定する方法を説明します。デスティネーション データベース定義で **Automatic Receive** オプションを選択する前に、使用するプラットフォームに当てはまる箇所をお読みください。

### UNIX-to-UNIX 自動受取り

UNIX-to-UNIX 転送では、DCM はソース マシンで `ccm_remexec` を使用してデスティネーション マシンで `ccm_receive` を起動します。

#### 現在のマシンでの受取り

デスティネーション データベース定義内のホスト名がソース (現在の) マシンの名前と一致する場合、生成、転送、および受取り操作は現在のマシン上で行われます。そのような場合、DCM は `ccm_remexec` を使用せずに現在のマシンで `ccm_receive` を実行します。

ソースとデスティネーションのデータベース ホスト名が同じなのに DCM が `ccm_remexec` の実行を試みる場合、Host Name フィールドに長いホスト名を使用します。たとえば、ホスト名を `mymachine` ではなく `mymachine.mydomain.com` に設定します。

#### 別のマシンでの受取り

別のソースおよびデスティネーション マシンで UNIX-to-UNIX 自動受取りを行うには、以下の条件を満足する必要があります。

- `ccm_receive` を `ccm_root` として実行する必要があるため、DCM は、デスティネーション マシンにユーザー `ccm_root` としてログインできる必要があります。

- 
- `ccm_root` ユーザーがパスワードなしにデスティネーション マシンにログインできるように、ソース マシンで `.rhosts` ファイルを設定する必要があります。そうしないと、システムがパスワードを要求するため受取りは自動的に行われません。

## Windows-to-Windows 自動受取り

Windows-to-Windows 転送では、DCM はソース マシンで `ccm_rem` を使用して、デスティネーション マシンの `ccm_remd` にアクセスします。その後 `ccm_remd` アプリケーションがデスティネーション マシンで `ccm_receive` を起動します。

### 現在のマシンでの受取り

デスティネーション データベース定義内のホスト名がソース (現在の) マシンの名前と一致する場合、生成、転送、および受取り操作は現在のマシン上で行われます。そのような場合、DCM は `ccm_rem` を使用せずに現在のマシンで `ccm_receive` を実行します。`ccm_receive` が開始するセッションは、ユーザー `ccm_root` によって開始されますが、現在のマシンに `ccm_root` としてログインする必要はありません。

ソースとデスティネーションのデータベース ホスト名が同じなのに DCM が `ccm_rem` の実行を試みる場合、Host Name フィールドに長いホスト名を使用します。たとえば、ホスト名を `mymachine` ではなく `mymachine.mydomain.com` に設定します。

### 別のマシンでの受取り

Windows-to-Windows 自動受取りを行うには、デスティネーション マシンを以下のように設定する必要があります。

1. Windows にユーザー `ccm_root` としてログインしている必要があります。そうしないと、受取りが失敗し、エラーメッセージ「User is not admin user (ccm\_root)」が表示されます。
2. デスティネーション マシンで、`ccm_remd` アプリケーションが実行している必要があります。そうしないと、受取りが失敗し、エラーメッセージ「Cannot establish connection」が表示されます。

## UNIX-to-Windows 受取り

UNIX-to-Windows 自動受取りは、現時点ではサポートしていません。しかし、別の方法としてデスティネーションデータベースで Windows の `at` コマンドを使用してスケジュールジョブを設定できます。`at` コマンドを使用すると、必要に応じて自動受取りを行うことができます。

デスティネーションデータベースで受取りをスケジュールするには、以下の `ccm_receive` コマンドとともに `at` コマンドを使用します。

```
ccm_receive -h engine_host -d
destination_database_path -dbid source_database_ID -ts
transfer_set_name -ccm_home Windows_CCM_install_area -
dir receive_directory
```

すべての転送セットを指定データベースから受取りたい場合は、`/ts` オプションを省略できます。デフォルトの `receive_directory` を使用したい場合は、`-dir` オプションを省略できます。

## Windows-to-UNIX 受取り

Windows-to-UNIX 自動受取りは、現時点ではサポートしていません。しかし、別の方法としてデスティネーションデータベースで、`ccm_receive` コマンドを実行する `cron` ジョブを用意できます。

---

## 転送セットの紛失

転送セットを受取るとき、パッケージ情報ファイルの最終生成時刻が最後に受取ったパッケージの生成時刻と比較されます。それらが一致しない場合、転送パッケージが紛失している可能性があります。そのような場合、DCM 警告を読んで以下のいずれかを行います。

- 紛失パッケージを無視する：これは 145 ページの「転送パッケージの紛失」で説明しているように、後にエラーを引き起こす可能性があります。  
または
- 受取ろうとしている転送パッケージを破棄して再生成する。

注記：受取りパッケージの生成時刻は

`database_path\dcm\receive\history`

ディレクトリにあります。このディレクトリとその内容は削除しないでください。削除すると、転送パッケージを受取るときに DCM 警告が表示される可能性があります。

## 受取りの失敗

### 失敗の理由

転送パッケージを抽出する前に以下のチェックが行われます。

- データベース パラメータは、DCM データベースになければならない (36 ページの「共通データベース パラメータの規定」を参照)。DCM は、重要なパラメータがソース データベースとデスティネーション データベースで一致していることを確認します。DCM が不一致を見つけるとチェックは失敗します。
- ソース データベースとデスティネーション データベースは同じ DCM 区切り文字を使用する必要がある。そうしないと、ソース データベースから生成した転送パッケージはデスティネーション データベースから認識されません。DCM マネージャは、DCM クラスタ全体で同じ区切り文字が使用されることを確認する必要があります。
- ソース データベースとデスティネーション データベースは同じリリース区切り文字を使用する必要がある。
- ソース データベースとデスティネーション データベースは、互換性のある Rational Synergy Distributed のバージョンを使用する必要がある。このチェックが失敗すると、DCM は使用できるバージョンを示すエラーメッセージを表示します。

注意！このチェックが失敗し、使用しているものより新しいリリースで生成されたパッケージを受取る場合、そのパッケージを受取るパッチをインストールする必要があるかもしれません。いずれにしても、IBM Rational Synergy サポートに連絡してサポートを依頼してください。そうしないと、重要なデータ変換が行われない可能性があります。

- ソース データベースとデスティネーション データベースの間で大文字／小文字設定の互換性が必要。Lower Case に設定されているデータベースからの転送パッケージは、Preserve Case に設定されているデータベースで受取ることができます。しかし、Preserve Case に設定されているデータベースからの転送パッケージは、オブジェクト名に大文字が含まれる場合には、Lower Case に設定されているデータベースと互換性がありません。

注記：転送パッケージのファイル名が小文字のみの場合には、大文字／小文字の不一致に関するメッセージは無視してかまいません。



- 
- ソース データベースとデスティネーション データベースは同じバージョン区切り文字を使用する必要がある。同じでないと、名前にデスティネーション データベースのバージョン区切り文字を含むオブジェクトが転送パッケージに含まれる可能性があります。ユーザーは、そのようなオブジェクトにアクセスできません。

注意！このエラーメッセージが表示された場合、ユーザーが確実にオブジェクトにアクセスできる場合にのみ先に進んでください。

### 転送パッケージの紛失

転送パッケージを受取る場合、DCM は紛失パッケージがないか調べます。紛失パッケージが報告される場合は、ソース データベースの DCM マネージャに連絡して原因を調べてください。一般的に、もっとも安全な方法は、転送パッケージを削除して、ソース データベースの DCM マネージャにそれより前の最新生成時刻を持つ新しい転送パッケージを生成してもらうことです。

注意！このエラーを無視すると、空のディレクトリ エントリあるいは関係が紛失する可能性があります。

### 紛失タイプ定義

デフォルトで、DCM は転送パッケージにすべてのユーザー定義タイプ定義を入れます。受取り操作時、DCM はまだデスティネーション データベース内に定義されていないユーザー定義タイプ定義を、デスティネーション データベース内に自動的に作成します。

転送セットがタイプ定義を除外する場合、転送パッケージはタイプがデスティネーション データベース内で定義されていないオブジェクトを含む可能性があります。DCM はタイプ定義の紛失を見つけると、受取り操作の実行を許可しません。そのような場合、*type\_developer* ロールのユーザーに紛失したタイプ定義の作成を依頼するか、転送パッケージを削除してソース データベースの DCM マネージャにタイプ定義を除外せずに生成を依頼します。

### オブジェクト名のコンフリクト

DCM クラスタ内の各オブジェクトは、固有のクラスタ ID 属性を持っています。詳細については、147 ページの「クラスタ ID」を参照してください。DCM 転送パッケージ内のオブジェクトが、送信先データベース内のオブジェクトと同じ 4 部構成のオブジェクト名を持つがクラスタ ID が異なる場合、これはオブジェクト名のコンフリクトとして報告されます。要するに、同じオブジェクト名を持つ 2 つのオブジェクトがまったく無関係である可能性があります。

オブジェクト名のコンフリクトは、いくつかの場合に発生します。

1. ソース データベースでオブジェクトが作られ、デスティネーション データベースへ送られます。それがソース データベースから削除されます。後に、ソース データベースで同じ 4 部構成のオブジェクト名を持つ別のオブジェクトが作られ、デスティネーション データベースへ送られます。これのもっとも一般的なケースは、*prep* 状態のプロジェクトで発生します。
2. ソース データベースでオブジェクトが作られ、デスティネーション データベースへ送られます。ソース データベースが、そのオブジェクトが存在する前に作られたバックアップからリストアされます。後に、ソース データベースで同じ4部構成のオブジェクト名を持つ別のオブジェクトが作られ、デスティネーション データベースへ送られます。
3. ソース データベースでオブジェクトが作られ、デスティネーション データベースへ送られます。オブジェクトがソース データベース内で名前変更され、オブジェクトの前の名前と同じ 4 部構成のオブジェクト名を持つ新しいオブジェクトが作成されます。新しいオブジェクトは、名前変更された前のオブジェクトを伴わずにデータベースへ複製されます。DCM は、名前変更されたオブジェクトがデスティネーション データベースへ複製される場合にのみ、自動的にオブジェクトの名前を変更します。

オブジェクト名のコンフリクトが発生した場合、まずソース データベースとデスティネーション データベースの両方でオブジェクトをチェックします。特に作成時刻と所有者を比較し、ファイルならソース コンテンツも比較します。

ケース (1) と (2) で、両方のオブジェクトが必要な場合は、新しい方のオブジェクトのバージョン属性を変更し、転送パッケージを削除し、再生成して受取りをやり直します。ケース (3) では、デスティネーション データベース内のオブジェクトの名前を変更するか、パッケージを削除し、名前を変更したオブジェクトが転送セット内にあることを確認し、転送パッケージを再生成して受取りをやり直します。

両方のオブジェクトが同じでなければならない場合、データベースから古いオブジェクトを削除して受取りをやり直します。DCM 受取りが使用するインポート処理は、以前受取った静的オブジェクトのソースの更新は行わないことに注意してください。cluster\_id 属性を削除して受取りをやり直したときに 2 つのオブジェクト間でソース属性が異なる場合、受取りによってそれは新しいコンテンツに更新されません。

---

## クラスタ ID

DCM データベース内のすべてのオブジェクトは、自動的にクラスタ ID が付けられます。属性は、全 **Rational Synergy Distributed** リポジトリにわたってオブジェクトを一意に識別し、オブジェクトが作成されたデータベースに直接関連しています。この属性には以下の目的があります。

- データベースにまたがって名前の変更を追跡する。
- オブジェクト名のコンフリクト、つまり同じ 4 部構成オブジェクト名を持つ異なるオブジェクトをチェックする。

オブジェクトをインポートすると、DCM はその名前とクラスタ ID を調べます。このチェックによって、名前変更されたオブジェクトを確認します。たとえば、静的オブジェクトが別のデータベースへ送られ、名前変更されてそのデータベースへ送り返されたとします。そのような場合、クラスタ ID を使用して名前の変更を検出します。したがって、デスティネーションデータベースには名前変更されたオブジェクトが 1 つ存在します。

## タイプ定義

DCM が特定のタイプのオブジェクトを受取るには、デスティネーションデータベースにそのタイプの定義が必要です。デフォルトで、DCM は転送パッケージにすべてのユーザー定義タイプ定義を入れます。受取り操作では、DCM は自動的にデスティネーションデータベース内にそのデータベースでまだ定義されていないユーザー定義タイプ定義を作成します。

手法によっては、ローカルアドミニストレータがすべてのタイプ定義を定義する必要があります。そのような場合、転送セットからタイプ定義を除外して、自動タイプ定義複製をやめることができます (51 ページの「省略可能なタイプ部分には、タイプ定義に関する情報があります。」を参照)。転送パッケージ内のタイプ定義がデスティネーションデータベースで定義されていない場合、受取り操作は放棄され、エラーメッセージが表示されます。

デフォルトで、DCM 受取り操作が無事終了した後で、転送パッケージに含まれるタイプ定義は、送信先データベースのファイルシステムから削除されます。しかし、ときにはタイプ定義を残しておく方がよい場合があります。たとえば、タイプ定義を残しておく、`ccm typedef /import /force` コマンドを使用してすべてのユーザー定義タイプ定義を中央データベースで管理できます。

**Keep Type Definition after Receive DCM** 設定は、DCM 受取りが無事行われた後でタイプ定義データを残しておくかどうかを管理します。デフォルト値は `FALSE` で、データを送信先データベースのファイルシステムから削除することを意味します。値が `TRUE` の場合、そのデータは以下に説明する `types` 受取りディレクトリに残ります。

転送パッケージは、パス `receive_directory¥package_ident` のもとに抽出されます。ここで `package_ident` は、データベース、転送セット番号、および時刻値で構成されています (例、`M#22#UM#962984701`)。 `types` ディレクトリは、`package_ident` ディレクトリの下にあり、エクスポートオブジェクトの `data` ディレクトリもそこにあります。

注記：タイプ定義が不要になったら、すべてのファイルを `types` と `data` 受取りディレクトリから削除してディスク領域を空けてください。

---

## パラレル検査

パラレル開発では、リリースに含める、同じオブジェクトに対するパラレル変更をマージすることが重要です。単一データベース環境では、ユーザー オブジェクトをチェックアウトまたはチェックインするときにパラレルバージョンに関する警告を受けることがあります。分散環境では、パラレルバージョンは異なるデータベースにある可能性があり、したがって、チェックアウトまたはチェックインするとき参照できない可能性があります。そのため、分散データベース環境ではパラレル開発を無効にすることはできません。

デフォルトで、受取り操作時、パッケージが定義済み **Entire database** 転送セットを使用して生成されている場合、あるいはオブジェクトが履歴メンバーである場合は、**DCM** は転送パッケージに受取られた各新規作成オブジェクトのパラレルバージョンをチェックします（プロジェクトと製品は例外）。ただし、履歴メンバーではなく、ユーザー定義転送セットであったオブジェクトについては、パラレル検査は行われません。これは、オブジェクトの履歴が完全である保証がないため、まとまりのない履歴によって大量の誤ったパラレル レポートが出される可能性があるからです。

この動作は、**Parallel Checking DCM** 設定を使用して、すべての受取り済みオブジェクト（プロジェクトと製品オブジェクトを除く）を含むよう変更できます。これらパラレルバージョンの詳細は、転送セットで定義されている受信者へ送られる **DCM** 受取り電子メールにあります。

**DCM** は、そのようなパラレルバージョンを所有するか作成した開発者に直接パラレル電子メール通知を送ることもできます。ソース データベースの転送セットの **Local Parallel Notification** オプションが選択されていると、パラレルバージョンの所有者である各ユーザーに電子メールが送られます（パラレル検査時）。電子メールは、ユーザーに関連する各受取り済みオブジェクトとそのパラレルバージョンのリストを示します。ここでも、この機能は、**Entire database** 転送セットを使用しているとき、あるいは転送セットの履歴メンバーであったオブジェクトにのみ適用されます。

電子メールの前文テキストは、

`database_path/etc/dcm_local_parallel_intro.txt`

ファイルに定義されています。このテキストには、キーワード `%database` があります。電子メールが送られると、このキーワードはデスティネーション データベースへのフルパスで置き換えられます。

**Entire database** 転送セットを使用しておらず、履歴メンバーを使用していない場合、**DCM** 受取りのパラレル検査は行われません。たとえば、ユーザー定義転送セットを使用しており、リリースのタスクの複製方法として転送セットにフォルダを追加するような場合がこれに当てはまります。このような場合、定期的にコンフリクト検出 (`ccm conflicts` コマンドまたは、**Rational Synergy Distributed** (**Synergy Classic** ではなく) のメンバーシップ コンフリクトの検出操作) を使用して別にパラレルチェックを行う必要があります。

## DCM が生成時刻を格納する方法

デフォルトで、DCM は 50 個の生成時刻を保存します。そのうち 20 個は「古い」時刻です。DCM が格納する生成時刻の数は、No. of Generate Times DCM 設定で定義されています。「古い」生成時刻の数は、No. of Old Generate Times DCM 設定で定義されています。

上記のデフォルト値に基づき、DCM は転送パッケージが生成された最新の 30 回の時刻を保存します。この値は、式  $50 - 20 = 30$  から求められます。つまり、No. of Generate Times から No. of Old Generate Times を引いて 30 が得られます。

DCM は、この 30 個の時刻を生成操作が行われた順に管理します。ユーザーが前の時刻からの転送パッケージを再生成しなければ、DCM が格納する生成時刻のリストは最新の時刻を先頭に時間順に並んでいます。

DCM は、Old Generate Times Resolution DCM 設定に基づいて 20 個の「古い」生成時刻を管理しています。このパラメータは、「古い」生成時刻の間隔を定義します。デフォルト値は、1 日 (1.0) です。

たとえば、毎時生成操作が行われると仮定します。また、パラメータ No. of Generate Times、No. of Old Generate Times、および Old Generate Times Resolution にデフォルト値が使用されているとします。

上記のシナリオでは、DCM は、最終的に最新の 30 個の生成時刻を格納します。DCM は、少なくとも 1 日 (1.0) 離れている 20 個の「古い」生成時刻も格納します。したがって、20 日前の紛失転送パッケージの再生成が可能です。

---

## DCM 受取り時のワークエリアの扱い

DCM は現在受取りデータベースにないプロジェクトを受取ると、デフォルトでソース データベースと同じワークエリア プロパティを持つプロジェクトを作成します。例外としてワークエリア パスは複製されません。これは、送信先データベースでワークエリア パスが存在しないか有効ではない可能性があるため、あるいはデータベースが同じ LAN 上にある場合、ソースデータベース内の同じプロジェクトが使用しているのと同じワークエリアパスを使用する可能性があり、これは常に失敗するためです。代わりに、受取られるプロジェクトのワークエリア管理が有効になっていると、そのプロジェクトは現在のユーザーのデフォルト ワークエリア パス テンプレートに基づいた新しいデフォルトのワークエリア パスを与えられます。

新しい受取りプロジェクトのためにワークエリアを作成できない場合、DCM は自動的にプロジェクトのワークエリア管理を無効にし、管理ワークエリアなしでプロジェクトを作成しようとします。また、DCM は、それを行ったことを示す警告を DCM イベント ログと送信するすべての DCM 受取り電子メールに入れます。その理由は、これにより DCM 受取りを進めてプロジェクトにメンバーを作成し、その後ユーザーは失敗した DCM 受取りをやり直さずにワークエリア問題を修正できるからです。

デフォルトの動作は、常に望ましい結果をもたらすとは限りません。

- デフォルトのプロジェクト パスが空き領域の少ないファイル システムにある場合、新たに受取ったプロジェクトがその領域を使用する可能性があります。これは、コピーベースのワークエリアでは特に顕著です。
- プロジェクトは、受取ったクライアントとは異なるクライアント用のものかもしれません。たとえば、UNIX クライアントで受取った Windows アプリケーション用のプロジェクトは、UNIX ファイルシステムにデフォルトのワークエリアを持っています。Windows クライアントユーザーは、後でワークエリアパスを Windows ファイルシステムの場所に変更する必要があります。
- *prep* プロジェクトでは、デフォルトのワークエリア パスは、そこからビルドを行う必要があるビルドマネージャが参照および変更できない可能性があります。

必要に応じて以下のいずれかを行うことができます。

- 受取りに使用される `ccm.ini` ファイルでワークエリア テンプレート定義を次のように変更する。Options セクションで `wa_path_template` オプションをより適切なワークエリア パスに変更する。

または

- すべての受取りプロジェクトに対してワークエリア管理を無効にする（これには `ccm_admin` ロールが必要）。後で、ユーザーは適切なワークエ

リアパスを定義できます。詳細については、158 ページの「Ignore Work Area Maintenance」を参照してください。

DCM が、すでに送信先データベースに存在するプロジェクトを受取るとき、プロジェクトのワークエリア プロパティは何も変更されません。作成後、プロジェクトのワークエリア プロパティはローカル設定とみなされます。これによりユーザーは、変更が後の DCM 受取りで上書きされる心配をせずに、相対または絶対ワークエリアなどのワークエリア プロパティを変更できます。



---

## ワークエリア サブディレクトリ テンプレートとパス

異なるデータベースで作成されるプロジェクトが、同じプロジェクト名とバージョンを使用することがあります。Rational Synergy Distributed が正しく管理するには、各プロジェクトが異なるワークエリア パスを使用する必要があります。このため、デフォルトで Rational Synergy Distributed は以下のようにプロジェクトサブディレクトリ テンプレートを使用します。

```
%project_name%optional_project_instance%delimiter%project_version
```

%optional\_project\_instance キーワードは、インスタンス値 `dbid#1` を使用してローカルプロジェクトの空の文字列に展開されます。非ローカルプロジェクトまたはローカルプロジェクトの 2 番目以降のインスタンスでは、DCM 区切り文字とインスタンス値に展開されます。

下表では、データベース A では、プロジェクトはプロジェクトの最終ワークエリア パスで使用される以下の展開サブディレクトリ テンプレートを持つ可能性があります。

プロジェクトオブジェクト名	展開サブディレクトリ テンプレート
local-1:project:A#1	local-1
local-1:project:A#2	local#A#2-1
nonlocal-1:project:B#1	nonlocal#B#1-1
nonlocal-1:project:C#1	nonlocal#C#1-1

したがって、DCM データベース ID または DCM 区切り文字の変更は、既存のワークエリア パスに影響する可能性があることに注意してください。

## 転送セット内のオブジェクト履歴

デフォルトでは、転送セットにオブジェクトが追加される時は履歴なしで追加されます。デフォルトを変更する方法は2つあります。

- 自分の個人用 `ccm.ini` ファイルの `Options` セクションの下に以下の設定を定義する。

```
dcm_dflt_add_history = TRUE
```

または

- データベース全般のデフォルト値を定義する（これには `ccm_admin` ロールが必要）。これは、**DCM Settings** ダイアログまたは `dcm` コマンドを使用して行います。詳細については、157 ページの「Default Add History」を参照してください。

ユーザーの `ccm.ini` ファイルで `dcm_dflt_add_history` を設定していない場合、デフォルト値は Default Add History DCM 設定で決まります。

---

## リリース 7.1 以前の Rational Synergy Distributed リリース間の転送

Rational Synergy Distributed 7.1 では、6.4SP1、6.5SP2、6.5a、6.6a、7.0、および 7.1 のデータベースとの転送をサポートしています。7.1 より古いデータベースには、DCM 互換パッチを適用する必要があります。Rational Synergy Distributed 7.1 は、以前のリリースではサポートされていなかったコンポーネントタスク機能を導入します。詳細については、Rational Synergy Readme を参照してください。

それ以前のデータベース (6.1、6.2、6.3、6.3a など) を持っていてデータを交換したい場合、以下のいずれかを行います。

- それらのデータベースを Rational Synergy Distributed 6.4 以降にアップグレードする。
- リリース 6.4SP1 または 6.5SP2 のハブデータベースを経由して複製する。

CCM45SP2 export format オプションおよび map project instances オプションは Rational Synergy Distributed 6.5 でなくなりました。

Rational Synergy Distributed 7.1 の DCM は、6.5 以降のデータベースでプロセス、プロセスルール、フォルダ テンプレート、およびリリース定義の複製をサポートしています。これらプロセス定義およびプロセスルールのすべての機能は、他の 6.5 以降のデータベースで複製する場合にのみ使用できます。したがって、できるだけ早くコンポーネントの開発に使用するマスタおよびサテライトデータベースを更新することを推奨します。旧リリースのハブ データベースを介して複製するスポーク データベースは、ハブ データベースを更新するまで機能が限定されます。

Rational Synergy Distributed 7.1 データベースを 6.5 より前の Rational Synergy Distributed データベースに複製すると、以下のようになります。

- プロセス定義とプロセスルールは複製されない。プロセス定義は Synergy Distributed 6.5 の新機能であり、旧リリースには同等の機能がありません。プロセスルールは、プロセスと使用するため 6.5 からデザインが大幅に変わり、旧リリースのリコンフィギュア/更新テンプレートに置き換わる新しい種類のオブジェクトです。
- フォルダ テンプレートは複製できる。
- クエリベースのフォルダおよびフォルダ テンプレートのクエリは、旧リリースの Synergy Classic では常に Custom クエリとなる。
- DCM 転送パッケージのプレビューは、日付/時刻値を日付文字列ではなく整数時刻値として表示し、名前が切り捨てられない。これは、プレビュー ファイルの形式が、場所やクライアントに影響されないよう変更されたためです。

Rational Synergy Distributed 6.5 より前のデータベースを Rational Synergy Distributed 7.1 データベースに複製すると、以下のようになります。

- リコンフィギュア/更新テンプレートは無視される。*recon\_temp* タイプは Rational Synergy Distributed 7.0 からはなくなり、より強力で柔軟なプロセスとプロセスルールデザインで置き換われました。
- フォルダテンプレートは複製できる。
- クエリベースのフォルダおよびフォルダテンプレートのクエリは、Synergy Classic では常に Custom クエリとなる。
- リリース定義を受取ったとき、リリースの有効な目的の1つに対応するプロセスルールが存在しない場合、そのリリースと目的用にリリース固有のプロセスルールが自動的に作成される。これは、旧 Rational Synergy Distributed データベースで使用されたどのリコンフィギュアテンプレートとも異なる設定を持つ可能性があります。

## DCM 設定

DCM Settings ダイアログと `ccm dcm /settings` コマンドでは、現在のデータベースでの DCM 操作に影響するもっとも一般的な DCM 設定を変更できます。下表に、使用可能な DCM 設定を示します。

DCM 設定	デフォルト値	説明
Description	空白文字列	現在のデータベースの説明。これは DCM 初期化時に指定できます。
Location	空白文字列	現在のデータベースの地理的位置を記述する自由形式テキスト。これは DCM 初期化時に指定できます。
Admin Info	空白文字列	現在のデータベースの DCM 管理者の名前と連絡先を示す自由形式テキスト。これは DCM 初期化時に指定できます。
Event Log Size	100	DCM イベント ログの最大エントリ数。詳細については、159 ページの「DCM イベント ログ」を参照してください。
Default Add History	FALSE	オブジェクトを転送セットに追加するときの履歴のデフォルト設定を決定します。
Default Include Associated Baselines	FALSE	新しい転送セットを作成するときの <b>Include Associated Baselines</b> 設定のデフォルトを決定します。
No. of Generate Times	50	すべての転送セットとデスティネーションデータベース ペアの最新生成時刻の最大数を指定します。詳細については、150 ページの「DCM が生成時刻を格納する方法」を参照してください。
No. of Old Generate Times	20	すべての転送セットとデスティネーションデータベースペアの、少なくとも <b>Old Generation Time Resolution</b> 日離れた最新生成時刻の最大数を指定します。詳細については、150 ページの「DCM が生成時刻を格納する方法」を参照してください。
Old Generate Time Resolution	1	古い生成時刻を保管する間隔を指定します。詳細については、150 ページの「DCM が生成時刻を格納する方法」を参照してください。

DCM 設定	デフォルト値	説明
Ignore Work Area Maintenance	FALSE	ソース データベースに管理ワークエリアを持っていたプロジェクトを、受取り時にワークエリア管理を有効にして作成するべきかを決定します。詳細については、151 ページの「DCM 受取り時のワークエリアの扱い」を参照してください。
Update Release Definitions	active	DCM 受取り時にリリース定義に対して行う操作を決定します。詳細については、116 ページの「汎用プロセスとリリース固有プロセスの複製」を参照してください。
Update DCM Database Definitions	TRUE	DCM 受取り時に自動的に DCM データベース定義を作成または更新するかどうかを決定します。詳細については、121 ページの「DCM データベース定義の複製」を参照してください。
Update Reconfigure and Folder Templates	TRUE	DCM 受取り時に転送パッケージ内のプロセス、プロセス ルールおよびフォルダ テンプレートを処理するべきかどうかを決定します。詳細については、116 ページの「汎用プロセスとリリース固有プロセスの複製」を参照してください。
Parallel Checking	created	DCM 受取り時に行うパラレル検査を決定します。詳細については、149 ページの「パラレル検査」を参照してください。
Keep Type Definitions after Receive	FALSE	DCM 転送パッケージのタイプ定義を、DCM 受取り操作が完了した後 <a href="#">receive_directory</a> に残すべきかどうかを決定します。
Receive Control Transitions	空のリスト	オブジェクトに、非管理データベースから管理データベースへの転送を適用できるかを決定します。詳細については、115 ページの「管理データベースへの遷移」を参照してください。

---

## DCM イベント ログ

DCM イベント ログは、以下の操作の詳細を記録します。

- DCM 初期化と再初期化
- DCM データベース ID の変更
- DCM 区切り文字の変更
- DCM 生成
- DCM 転送
- DCM 受取り
- Save Offline and Delete 機能からの DCM パッケージの作成

デフォルトで、イベント ログは最大 100 個のイベントを記録できます。イベント ログが最大サイズに達すると、もっとも古いイベントを削除して新しいイベントが記録されます。最大 DCM イベント ログ サイズは、DCM Settings ダイアログまたは dcm コマンドで増やすことができます。

各イベントに対して DCM は 2 種類の情報を取り込みます。

- イベント情報 — イベント詳細のまとめです。DCM 生成、転送、および受取り操作では、これは送信される電子メールに含まれる情報と同じです。
- メッセージ — その操作のメッセージ表示またはログ ファイルに書き出されたメッセージをすべて取り込みます。操作が失敗した場合、そのようなメッセージを調べて詳しい内容を確認してください。

## ブロードキャスト データベースとパッケージ

デフォルトで、DCM は Any ブロードキャスト データベースの定義済みデータベース定義を作成します。ブロードキャスト データベース定義は、実際のデータベースを表すわけではありません。これは、どのデータベースでも受取れる DCM 転送パッケージを作成できるようにすることが目的です。これは、リリース済みアプリケーションまたはコンポーネントを発行するとき便利です。DCM 転送パッケージを生成し、そのパッケージを FTP サーバーなどの周知の場所におくことができます。他のユーザーは、そのパッケージをダウンロードして自分のデータベースに受取ることができます。



# 用語集

<b>associated_cv</b>	<i>associated_cv</i> は、タスクに関連する特定のオブジェクトバージョンを示す関係です。
<b>associated_task</b>	<i>associated_task</i> は、変更依頼に関連するタスクを示す関係です。
<b>ccm_admin</b>	<i>ccm_admin</i> ロールは、変更不能オブジェクトのプロパティを変える可能性のある操作に必要です。これには、DCMを使用するためにデータベースを初期化、転送セットの受取りなどの操作があります。
<b>created_in</b>	<i>created_in</i> 属性は、特定のオブジェクトバージョンが作成されたデータベースを示します。
<b>dbid</b>	データベース ID を参照。
<b>DCM</b>	Rational Synergy Distributed の省略形。分散型変更管理ツールです。Synergy Classic メニューから DCM にアクセスするには、別ライセンスが必要です。DCM を使用すると、世界中の Synergy データベース内で並行して作成および変更されているオブジェクトを、複数の Synergy データベース間で共有できます。
<b>dcm_mgr</b>	<i>dcm_mgr</i> ロールは、デスティネーションデータベースや転送セットの定義、転送セットへのオブジェクトの追加、および転送パッケージの生成に必要です。
<b>DCM アドミニストレータ</b>	DCM アドミニストレータは、DCM 操作を行う人です。ほとんどの DCM 操作に必要なロールは <i>dcm_mgr</i> で、 <i>ccm_admin</i> ロールは DCM 初期化または受取り操作を行うときのみ必要です。
<b>DCM 区切り文字</b>	DCM 区切り文字は、1) オブジェクト参照でデータベース ID とオブジェクトインスタンス間、2) デスティネーションデータベースとチェックアウトされたインポート済みオブジェクトでデータベース ID とオブジェクトバージョン間、3) タスク番号が表示される一部のダイアログでデータベース ID とタスク番号間で、それぞれを区切るために使用される文字です。  デフォルトの DCM 区切り文字は「#」です。これ他に「!」、「~」、および「=」などの文字も使

	用できます。
	DCM 区切り文字は、特定の DCM クラスタ内のすべてのデータベースで同じである必要があります。
<b>DCM クラスタ</b>	DCM クラスタは、共有開発環境を構成し、DCM 手法に基いて相互に交信するデータベースの集まりです。
<b>DCM 手法</b>	DCM 手法は、どのデータベース間でどのような情報をどのような目的で複製するか、そしてそのような共有変更からどのようにソフトウェアコンポーネントやアプリケーションを作成するかを定義する計画です。典型的な DCM 手法には、マスタとサテライト、公開と登録、およびピア ツー ピアがあります。
<b>DCM データベース</b>	DCM データベースは、DCM を使用するよう初期化されている Rational Synergy データベースです。
<b>DCS</b>	Distributed Rational Change の省略形。
<b>direct history</b>	直接メンバーが履歴付きで追加されるときそのメンバータイプは <i>direct history</i> です。
<b>Entire Database</b>	
転送セット	Entire Database 転送セットは、データベース内のすべての適格オブジェクトを含む内蔵転送セットです。
<b>from</b>	Properties ダイアログの From フィールドは、オブジェクトバージョンが作成されたデータベースの dbid を示します。
<b>generate_directory</b>	<i>generate_directory</i> は、DCM 生成操作が一時ファイルと転送パッケージファイルを書き出す場所です。デフォルトで DCM はこれらのファイルをソース データベースの <i>database_path\%dcm%\generate</i> ディレクトリに入れます。しかし、転送セットの作成時に別のディレクトリを指定することもできます。
<b>indirect history</b>	履歴付き転送セットに直接メンバーを追加すると、その間接メンバーは <i>indirect history</i> メンバーとして追加されます。

---

<b>local to</b>	<b>Properties</b> ダイアログの <b>Local To</b> フィールドは、オブジェクトの初期バージョンが作成されたデータベースの ID を示します。 <b>Local To</b> データベース ID は、オブジェクトの履歴を通してオブジェクトに対応し、変化しません。
<b>modifiable_in</b>	<i>modifiable_in</i> は、オブジェクトを管理するデータベースを示すオブジェクト属性です。この属性は、オブジェクトが管理データベース内で他のデータベースのデータで上書きされるのを防ぎます。たとえば、オブジェクトが現在の dbid 以外の dbid に設定されている <i>modifiable_in</i> 属性を持つ場合、それは DCM 変更可能ではありません。また、オブジェクトが <i>modifiable_in</i> 属性を持たない場合、それは <i>created_in</i> 属性が現在の dbid に設定されている場合にのみ DCM 変更可能です。
<b>receive_directory</b>	<i>receive_directory</i> は、DCM 転送および受取り操作が転送パッケージファイルを書き出す場所です。デフォルトで、DCM はこれらのファイルをデスティネーションデータベースの <i>database_path\%dcm\%receive</i> ディレクトリに入れます。ただし、ユーザーはデフォルト以外のパスを指定できます。指定するパスは、エンジンプロセスから参照および書き出し可能である必要があります。
大文字／小文字区別	大文字／小文字区別は、大文字と小文字を区別します。
受取り	受取り操作は、転送パッケージで送られた新しいオブジェクトバージョンをデータベースに取り入れます。転送パッケージは、そのデスティネーションデータベースで手動で受取るか、生成操作の一部として自動的に受取ることができます。
受取りデータベースリスト	受取りデータベース リストは、現在のデータベースが転送パッケージを受取ったデータベースのリストです。
オブジェクト名の コンフリクト	オブジェクト名のコンフリクトは、2つの異なるオブジェクトが同じ 4 部構成のオブジェクト名を持つとき発生します。

親変更依頼	親変更依頼は、他の変更依頼の論理グルーピングです。たとえば、変更依頼は複数のデータベースの修正を必要とする可能性があります。これを行う 1 つの方法としては、変更依頼を 1 つまたは複数の変更依頼（ <a href="#">子変更依頼</a> ）に分けて、修正を行う適切なデータベースに子変更依頼を配布します。
カレント タスク	カレント タスクは、現在作業を行っているタスクです。この用語は Rational Synergy インターフェイスで使用しています。デフォルト タスクも参照してください。
間接メンバー	<p>転送セットの間接メンバーは、DCM 追加操作を使用して明示的に追加されるオブジェクトに関連しているために DCM が含めるオブジェクトです。間接メンバーは、転送セットから明示的に削除できません。</p> <p>転送セットにフォルダを追加すると、DCM はそのフォルダのメンバー タスクとタスクに関連するすべてのオブジェクトを間接メンバーとして含めます。</p> <p>転送セットにタスクを追加すると、DCM はそのタスクに関連するすべてのオブジェクトを間接メンバーとして含めます。</p> <p>転送セットにプロジェクトを追加すると、DCM は以下を間接メンバーとして含めます。</p> <ul style="list-style-type: none"> <li>-- そのプロジェクトのすべてのメンバー オブジェクト</li> <li>-- プロジェクトが使用している任意のベースライン</li> <li>-- プロジェクトの更新プロパティ内のすべてのフォルダ</li> <li>-- 更新プロパティ内のすべてのタスク</li> </ul> <p>転送セットにソース オブジェクトを追加すると、DCM はそのオブジェクトに関連するすべてのタスクを間接メンバーとして含めます。</p>
管理データベース	オブジェクトの管理データベースは、オブジェクトを管理するデータベースで、その中で変更を行うことができるデータベースです。他のすべてのデータベースでは、オブジェクトを変更できません。

---

管理のハンドオーバー	管理のハンドオーバーは、オブジェクトの管理データベースが現在のデータベースから別のデータベースへ変わると起きます。
クラスタ ID	クラスタ ID は、DCM データベース内の各オブジェクトに割り当てられる属性です。この属性は、すべての Synergy リポジトリにわたってオブジェクトを一意に識別し、オブジェクトが作成されたデータベースに直接関連しています。この属性は、データベースにまたがって名前の変更を管理することを目的としています。
公開者	公開者は、DCM クラスタ内のすべてのデータベースに共有データを公開するデータベースです。公開者は、登録者と呼ばれるデスティネーションデータベースへデータを送ります。
子変更依頼	子変更依頼は、親を持つ変更依頼です ( <a href="#">親変更依頼</a> を参照)。
サテライトデータベース	サテライト データベースは、中央データベースまたはマスタ データベースから共有データの更新を受取り、そのアドレスをマスタへ送るデータベースです。
スポーク データベース	スポーク データベースは、中央データベースまたはハブ データベースから共有データのアップデートを受取り、その更新をハブへ送るデータベースです。
生成	生成操作は、転送パッケージを作成してデスティネーションデータベースへ送ります。転送パッケージは、手動コピー転送モードを使用していない限り自動的に送られます。自動受取りを選択している場合は、DCM はデスティネーションデータベースで転送パッケージを自動的に受取ります。
送信	転送を参照してください。
ソース オブジェクト	ソース オブジェクトは、C のソース ファイルや実行形式ファイルの ASCII またはバイナリ オブジェクトバージョンです。
ソース データベース	ソース データベースは、別のデータベースへ転送パッケージを送るデータベースです。

ダイナミック ロール スイッチング	ダイナミック ロール スイッチングは、ユーザーが、データベース内の正しいロールにアクセスできるにも関わらず誤ったロールで DCM 操作を行おうとしたとき、自動的にロールを変更するものです。操作が終わると、DCM はユーザーを元のロールに戻します。
チェックアウト	Rational Synergy のデータベース内に格納されている既存バージョンからオブジェクトの新規バージョンを作成する操作。開発者は作業ができるようにオブジェクトをチェックアウトします。
チェックイン	他のユーザーも利用できる開発者のオブジェクト バージョンを作成するために使用される操作。
データベース ID	データベース ID (dbid) は、DCM クラスタ内の各データベースを一意に識別する文字列です。これは、DCM 初期化の設定プロセスで手作業で割り当てます。データベース ID は、DCM オブジェクトの version と instance 属性で使用します。これは、オブジェクト名が DCM クラスタ内で一意であることを保証します。
直接メンバー	転送セットの直接メンバーは、DCM 追加操作を使用して明示的に追加されるオブジェクトです。直接メンバーのみ明示的に転送セットから削除することができます。直接メンバーを削除すると、そのすべての間接メンバーも削除されます（転送セットの他の場所で使用されていない限り）。  オブジェクトを履歴付きで追加すると、そのすべての前後のバージョンがすべて直接メンバーとして追加されます。したがって、履歴付きで追加したオブジェクトを削除すると、そのオブジェクトのすべてのバージョンが削除されます。
通常転送	通常転送は、初期転送の後で送られる特定の転送パッケージです。通常転送は、変更だけが送られるため、一般的に最初の転送より少ないオブジェクトで構成されています。
デスティネーション データベース	デスティネーション データベースは、転送パッケージを受取るすべてのデータベースです。

---

デスティネーション データベース定義	デスティネーションデータベース定義は、変更セットを送信する送信先データベース用の転送情報の名前付け、説明、提供を行うパラメータセットです。DCM は、転送パッケージを生成するために、転送セットとこの定義をペアで使用します。
デフォルト タスク	デフォルト タスクは、現在作業を行っているタスクです。この用語は <b>Synergy Classic</b> インターフェイスで使用しています。カレントタスクも参照してください。
転送	転送操作は、転送パッケージをソースデータベースからデスティネーションデータベースへ移動します。転送は、転送パッケージを生成したとき自動的に行われます（手動コピーの転送を行っていない限り）。
転送セット	転送セットは、デスティネーションデータベースへ送ることができる関連するオブジェクトのグループです。DCM は、転送セットをデスティネーションデータベース定義と組み合わせて転送パッケージを生成します。
転送パッケージ	転送パッケージは、ソースデータベースで転送セットとデスティネーションデータベース定義のペアから生成されるファイルの集合です。転送パッケージは、デスティネーションデータベースへ送られ、DCM 情報ファイル、プレビュー ファイル、および転送するオブジェクトを含むデータ ファイルを含みます。
転送モード	<p>           転送モードは、転送パッケージをデスティネーションデータベースへ送るときに使用する手法です。データベースの転送モードは、データベースをデスティネーションデータベースリストに追加するとき選択されます。転送モードは、<b>Generate</b> ダイアログを使用して変更できます。         </p> <p>           転送モードには、<b>Manual Copy</b>、<b>Local Copy</b>、<b>Remote Copy</b>、<b>File Transfer Protocol</b>、および <b>User Defined</b> の 5 種類があります         </p> <p> <b>Manual Copy</b> モードを使用する場合、転送パッケージをデフォルトディレクトリ以外のディレクトリ（つまり、<code>database_path¥dcm¥generate</code>）に置くほう         </p>

	が都合が良いかもしれません。そのときには、別の <code>generate_directory</code> を指定できます。
転送リスト	転送リストは、生成操作で計算した、転送パッケージでデスティネーションデータベースへ送られるオブジェクトのリストです。
登録者	登録者は、公開者と呼ばれる中央データベースから共有データを要求して受取るデータベースです。
ノード データベース	ノード データベースは、DCM クラスタ内のすべてのデータベースから更新を受け付けるデータベースです。最初からノード データベースであるデータベースはありません。データベースは、マスタ/ノード DCM 手法（マスタとサテライトなど）を定義するときノードに指定されます。
ハブ データベース	ハブ データベースは、共有データの DCM クラスタ内のすべてのデータベースへの転送を受け持つデータベースです。ハブ データベースは、他のデータベースから共有データの変更も受取ります。
ピア	ピアは、DCM クラスタ内のデータベースへの情報の独自の流れを管理し、独自の <i>prep</i> プロジェクトを維持するデータベースです。ピア データベースは、DCM クラスタ内の任意のデータベースとデータの転送および受取りを行うことができます。
複製トポロジ	複製トポロジは、データベースが相互に直接データを複製するか、仲介のハブ データベースを介して行うかを定義します。典型的な複製トポロジにはポイント ツー ポイントおよびハブとスポークの 2 つがあります。
変更依頼	変更依頼は、以下のいずれかを使用して入力した問題または課題です。Rational Change（標準変更管理ツール）、または Distributed Rational Change（DCS）（分散型変更依頼管理パッケージ）。
変更依頼オブジェクト	変更依頼オブジェクトは、変更依頼を記述するデータベース オブジェクトです。変更依頼の属性には、変更依頼の理由と日時、および依頼を提出した人が含まれます。
ホーム データベース	ホーム データベースは、オブジェクトバージョンが初期化または作成された DCM データベースです。オブジェクトのホーム データベース ID は、その Local To プロパティと同じで、オブジェクトがどこへ送られても一緒に行きます。



---

ポイント ツー ポイント	ポイント ツー ポイント複製トポロジでは、各データベースは情報を送るデータベースまたは / および変更依頼の管理を渡すデータベースの転送パッケージを生成します。
マスタ データベース	マスタ データベースは、DCM クラスタ内のすべてのデータベースの更新を受け持つデータベースです。最初からマスタ データベースであるデータベースはありません。データベースは、マスタ / ノード DCM 手法 (マスタとサテライトなど) を定義するときマスタに指定されます。
マスタ統合 プロジェクト	マスタ統合プロジェクトは、DCM クラスタ内のすべてのデータベースで行われた統合変更をビルドしてテストするプロジェクトです。
マスタ ビルド	マスタ ビルドは、DCM クラスタ内のすべてのデータベースで行われた変更を統合するビルドです。
マスタ フォルダ	マスタ フォルダは、DCM クラスタ内のすべてのデータベースから変更を集めるために使用するフォルダです。マスタ フォルダは、マスタ <i>prep</i> プロジェクトの更新プロパティに含まれています。
履歴付き	履歴付きは、オブジェクトをその前のバージョンとともに転送することを意味します。



## 付録：特記事項

© Copyright 1992, 2009

本書は米国 IBM が提供する製品およびサービスについて作成したものであり、本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権（特許出願中のものを含む）を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒 106-8711

東京都港区六本木 3-2-12

日本アイ・ビー・エム株式会社

法務・知的財産

知的財産権ライセンス渉外

以下の保証は、国または地域の法律に沿わない場合は、適用されません。: IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示 もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、製造元に連絡してください。

Intellectual Property Dept. for Rational Software  
IBM Corporation  
1 Rogers Street  
Cambridge, Massachusetts 02142  
U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性がありますが、その測定値が、一般に利用可能なシステムのもと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者にお願いします。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

---

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

## 商標

IBM および関連の商標については、[www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml) をご覧ください。

Microsoft、Windows、Windows 2003、Windows XP、Windows Vista、および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

UNIX は、The Open Group の米国およびその他の国における登録商標です。他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。

## 索引

## A

associated\_cv、説明 161  
 associated\_task、説明 161  
 at、自動受取りに使用 142

## C

ccm.ini ファイル  
   電子メール通知 122  
 ccm\_admin、説明 161  
 ccm\_remd ファイル 80  
 Change Distributed  
   概要 98  
 created\_in、説明 161  
 cron、自動受取りに使用 142

## D

dbid  
   説明 161  
   転送パッケージで 121  
   変更 64  
 DCM 121  
   DCM が必要とする通知 62  
   Rational Synergy との使用 11  
   機能 11  
   サポートされているバージョン 155  
   サンプルプレビュー ファイル 127  
   使用する理由 10  
   セキュリティ 16  
   説明 161  
   ディスク領域のワークシート 29  
   トラブルシューティング 143, 144  
   パラレル開発のサポート 11  
   プロジェクト スペック 18  
   変更依頼 98  
   用語 12  
   ロール 19  
 DCM ID、変更 121  
 dcm\_eng.log ファイル 54  
 dcm\_info.txt ファイル 127

dcm\_mgr、説明 161  
 dcm\_preview.txt ファイル 127  
 dcm\_ui.log ファイル 54  
 DCM アドミニストレータ、説明 161  
 DCM イベント ログの内容 159  
 DCM 区切り文字  
   使用可能な文字 161  
   説明 161  
   ID の変更 67  
 DCM クラスタ、説明 162  
 DCM クラスタの計画 21  
 DCM コマンド、記述方法 7  
 DCM 手法  
   公開と登録 25  
   考慮点 22  
   サンプル シナリオ 81  
   説明 162  
   ピア ツー ピア 26  
   マスタとサテライト 24  
 DCM 情報の検索 1  
 DCM 初期化、説明 41  
 DCM 設定、アップグレード後の変更 63  
 DCM 操作に必要なディレクトリ 77  
 DCM ディレクトリ、管理 77  
 DCM データベース  
   generate オプションの定義 43  
   ID の変更 64  
   共通プロパティ 35  
   初期化 41  
   説明 162  
   全メンバーの転送 74  
   定義の作成 43  
   複製 71  
 DCM 転送セット、作成 44  
 DCM の機能 11  
 DCM ハードウェア、同期 38  
 DCM 複製  
   手法 107  
   トポロジ 107  
 DCM プロジェクト インスタンス 18  
 DCM プロパティ  
   DCM 区切り文字 35  
   大文字／小文字設定 35  
   オブジェクトの 14  
   タスク属性 36, 37

プラットフォーム 36  
プロジェクト目的リスト 36  
リリース 37  
リリース区切り文字 35, 36  
例 15

DCM 命名規則 17

DCS

説明 162

分散型 Change も参照

direct history、説明 162

## E

Entire Database 転送セット

メンバー 74

指定 47

使用 76

説明 162

メンバー 74

## F

From プロパティ、説明 162

ftp 転送用コマンド

UNIX 133

Windows 134

ftp モード 132

## G

*generate\_directory*、説明 162

## H

hosts.equiv ファイル 80

## I

IBM Rational Change、概要 96

indirect history、説明 162

## L

Local To プロパティ、説明 163

## M

modifiable\_in、説明 163

## O

om\_hosts.cfg ファイル 36

## P

project\_purpose\_list 36

## R

*receive\_directory*、説明 163

## T

tar ファイル 128

## U

UNIX 区切り文字 7

## あ

新しい転送セットで使用されるリリース  
クエリ 119

アップグレード、後の DCM 設定の変更  
63

## い

一時ファイル、管理 77

インスタンス、デフォルト プロジェクト  
18

## う

受取り操作

オブジェクトの所有 115

失敗 144

自動 140

詳細説明 57

設定 53

説明 163

遷移のチェック 115  
 必要なロール 53  
 プレビュー ファイルの表示 28  
 方法 60  
 ワークエリアの処理 151  
 受取りデータベースリスト、説明 163

## え

エンジンとサーバーの同期 38

## お

大文字／小文字区別、説明 163  
 大文字／小文字設定、互換性 35  
 オブジェクト
 

- dbid 変更の影響 64
- DCM 内での命名方法 17
- 管理の切り替え 118
- 修正プロパティの変更 114
- 集中管理 117
- 状態の複製 115
- すべて転送 74
- 転送セットから削除 70
- 転送パッケージから除外 52
- 転送パッケージに追加される方法 51
- 転送プロパティ 45
- 名前のコンフリクト 145
- パラレルバージョンの検査 149
- ハンドオーバーコントロール 43
- 履歴付き転送 154
- ローカルコントロール 117

 オブジェクト名のコンフリクト 145  
 親変更依頼、説明 164

## か

カスタマ サポート 2  
 カレント タスク、説明 164  
 間接メンバー、説明 164  
 管理データベース
 

- 使用している属性 114
- 遷移の複製 115
- ハンドオーバー管理 114

 管理、ローカル対中央 117

## き

技術サポート 2  
 共有コンポーネント、使用 25

## く

区切り文字の変更 67  
 クラスタ ID
 

- 説明 165
- 目的 147

## け

現在の時刻設定 88

## こ

公開者、説明 165  
 公開と登録による転送手法 25  
 更新プロパティ
 

- 設定 36
- 転送方法 45, 123

 子変更依頼、説明 165  
 コマンド構文 7

## さ

削除
 

- デスティネーション データベース定義 69
- 転送セット 70

 作成
 

- データベースの複製 71
- 転送セット 44
- 転送パッケージ 51
- ミラー データベース 76

 サテライト データベース
 

- 説明 165
- マスタへ転送 89

 サテライトへの初期転送 86  
 サーバー上の時間差 38  
 サーバー、同期 38



## し

## 自動受取り

- at の使用 142
- cron を使用 142
- エラー 54
- 制限 140
- 設定と使用 140
- ユーザー ID 53

## 自動受取りのユーザー ID 53

## 手動コピー モード 129

## 情報ファイル 127

## 使用方法 149

## 初期化

- DCM データベース 41
  - スポーク転送セット 88
  - データベース、サンプルシナリオ 83
- 信頼されるホスト、DCM との使用 138

## す

## スクリプト

- DCM との使用 135

## スポーク データベース

- クラスタへの追加 93
- 設定 87
- 説明 165
- 転送セットの初期化 88

## せ

## 生成時刻

- DCM が格納する数を設定する方法 151
- DCM が格納する方法 150
- ずれ 38

## 生成操作

- 詳細説明 51
- 説明 165
- 含まれるプロセス ルール 121
- 含まれるリリース 120
- 方法 55

## 設定

- DCM データベース 40
- DCM ファイルシステム 27

- 更新プロパティ 36
- スポーク データベース 87

## そ

- ソース オブジェクト、説明 165
- ソース データベース、説明 165

## た

## ダイナミック ロール スイッチング、説明 166

## タイプ定義

- 説明 148
- 紛失 145
- ロード 57

## タイムゾーン、差の処理 11

## ダイレクトコピー モード 129

## タスク、転送セットから削除 70

## タスク属性

- DCM での使用 36, 37
- 定義 37

## 他の DCM ドキュメント 1

## ち

## チェックアウト、説明 166

## チェックイン、説明 166

## チェックの失敗 144

## 注意、説明 7

## 注記、説明 7

## 直接メンバー、説明 166

## つ

## 追加

- スポーク データベース 93
- デスティネーション データベース 43
- 転送セットにオブジェクトを 45
- フォルダ内のタスク 116

## 追加操作、説明 45

## 通常転送、説明 166

## て

## 定義

- デスティネーション データベース 43
- 転送セット 44
- ディスク領域の割り当て 27
- デスティネーション データベース
  - zip 設定の変更 68
  - 受取りオプションの変更 68
  - 追加 43
  - 定義の削除 69
  - データベース名の変更 68
  - 転送時のアクセス可能度 74
  - 転送モードの変更 68
  - 定義の変更 68
- デスティネーションデータベース、説明 166
- デスティネーション データベース定義、説明 167
- データの整合性 16
- データ ファイル 128
- データ複製プロセス 22
- データベース
  - オブジェクトの管理方法 114
  - 共通プロパティの設定 35
  - 高速パフォーマンス用の定義 137
  - 初期化 41
  - 信頼されるホストの使用 138
  - すべて転送 74
- データベース ID
  - dbid を参照
- データベース管理用の属性 114
- データベース プロパティ、必要な 36
- デフォルト タスク、説明 167
- 電子メール通知 57
- 転送
  - 旧リリース 155
  - サテライトからマスタへ 89
  - マスタからサテライトへ 90
- 転送済みデータ、ロード 58
- 転送セット
  - Entire Database 162
  - 受取り 60
  - オブジェクトの削除 70
  - オブジェクトの追加 45

- オブジェクトの履歴の追加 154
- 削除 70
- 作成 44
- 除外オブジェクトの変更 69
- 生成ディレクトリの変更 69
- 説明 167
- 電子メールアドレスの変更 69
- 名前の変更 69
- プロセス ルールの追加 116
- 紛失している場合 143
- ベースラインを含める 125
- 問題範囲の変更 69
- リリース情報を含む 118
- リリース複製の変更 69
- ローカルパラレル通知 69

## 転送操作

- 旧バージョン 155
- スクリプトの使用 135
- 説明 167
- 方法 55

## 転送パッケージ

- 受取り 53
- 受取り時の場所 139
- オブジェクトを除外する方法 52
- 構成方法 126
- 構造 125
- 作成 51
- 生成、詳細説明 51
- 説明 167
- 送信 53
- 内容 51
- 紛失 143

## 転送パッケージの受取り 53

## 転送パッケージの送信 53

## 転送パッケージの汎用プロセス ルール 116

## 転送モード

- 設定の詳細 129
- 説明 167

## 転送リスト、説明 168

## テンプレート、サブディレクトリ 153

## と

- 登録者、説明 168

トポロジの例 22  
トラブルシューティング 143, 144

## な

名前のコンフリクト 145

## の

ノードデータベース、説明 168

## は

バージョン、DCM での使用方法 17  
パス名、書き方 7  
パフォーマンス  
  影響する要因 137  
  向上 137  
ハブデータベース、説明 168  
ハブとスポークトポロジ 22  
パラレル開発 149  
パラレル開発のサポート 11  
パラレルバージョンの確認 58  
パラレルバージョンのチェック 149, 150  
ハンドオーバー  
  オブジェクトの管理 114  
  オブジェクトのコントロールの設定  
    43

## ひ

ピア、説明 168  
ピア ツー ピア転送手法 26  
必要ディスク領域の決定 27  
標準対分散プロセス 42  
ビルド  
  DCM の影響 42  
ビルドマネージャ  
  DCM ドキュメントの使用 1  
  DCM 責任のサンプル 81  
  ビルドマネージャ間の通信 62

## ふ

ファイル

.netrc 133  
.rhosts 80  
dcm\_eng.log 54  
dcm\_info.txt 127  
dcm\_preview.txt 127  
dcm\_ui.log 54  
om\_hosts.cfg 36  
tar 128  
  自動受取りエラー 54  
  情報 127  
  データ 128  
  プレビュー 127  
フォルダ  
  転送セットから削除 70  
  転送セットへの追加 116  
フォルダ テンプレート  
  転送セットへの追加 116  
複製  
  プロセスルール 69  
  リリース情報 69  
  リリース名 69  
複製トポロジ、説明 168  
不適格オブジェクト 52  
プレビューファイル 127  
ブロードキャストデータベースとパッ  
  ケージ 160  
プロジェクト  
  転送セットから削除 70  
  変更の連絡 62  
  メンバーシップの更新 37  
  ワークエリア パスの使用 153  
プロセス  
  データベース定義 116  
  転送セットの追加 46  
  標準 42  
  分散 42  
プロセスルール  
  Local Collaborative Development  
    (ローカル共同開発) 42  
  Local Integration Testing (ローカル統  
    合テスト) 42  
  Master Integration Testing (マスタ統  
    合テスト) 42  
  データベース定義 116  
  転送セットの追加 46

プロパティ、転送セットの定義 44  
 プロパティのリコンフィギュア、転送方法 123  
 分散型 Change  
   構成 100  
   サポートしている機能 98  
   手法 101  
   モデルパラメータ 112  
 分散型変更依頼手法 101  
 紛失  
   タイプ定義 145  
   転送パッケージ 145

## へ

ベースライン  
   関連の取り入れ 69  
   転送すべきかどうか 92  
   転送セットに含める 125  
   メンバー更新の影響 125  
 変更  
   除外オブジェクト 69  
   生成ディレクトリ 69  
   デスティネーション データベース説明 68  
   デスティネーション データベース転送モード 68  
   デスティネーション データベースの zip 設定 68  
   デスティネーション データベースの受取りオプション 68  
   デスティネーション データベース名 68  
   データベースの管理 114  
   電子メール通知アドレス 69  
   転送セット名 69  
   問題範囲 69  
   ローカルパラレル通知 69  
 変更依頼オブジェクト、説明 168  
 変更依頼、説明 168

## ほ

ポイント ツー ポイント トポロジ、説明 169

ホームデータベース、説明 168  
 本書で正在している記号 7  
 本書で正在している書体 7

## ま

マスタ  
   prep (準備) プロジェクト 16  
   データベース 169  
   ビルド 16  
   フォルダ 16  
 マスタ データベース  
   サテライトへ転送 90  
   サテライトへの初期転送 86  
 マスタ統合プロジェクト、説明 169  
 マスタとサテライト手法 24  
 マスタ ビルド、説明 169  
 マスタ フォルダ、説明 169

## み

ミラー データベース 76

## め

命名規則 17  
 メイラー、自分のものを使用 122

## ゆ

ユーザー定義モード 135

## り

リモート コピー モード 130  
 リモート実行 80  
 リリース区切り文字  
   DCM での使用 36  
   要件 35  
 リリース定義  
   DCM での使用 37, 118  
   旧リリースとの互換性 128  
   追加 116  
   転送パッケージに含める方法 128  
   複製 116

履歴、オブジェクトの追加 154  
履歴付き、説明 169

## ろ

ローカル コピー モード 130  
ローカル データベース  
パフォーマンスの向上 137  
ログ、イベントの表示 159  
ロード  
タイプ定義 57  
転送済みオブジェクト 58  
ロール  
ccm\_admin 19  
dcm\_mgr 19  
受取り操作 53  
ダイナミック ロール スイッチング 19

## わ

ワークエリア管理 151  
ワークエリア管理を無視 158  
ワークエリア プロパティ、設定方法 151